ALIBABA CLOUD

Alibaba Cloud

E-MapReduce

SmartData

Document Version: 20220424

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	Danger: Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Overview 15
2.Overview and usage of JindoFS 17
3.SmartData 3.8.x 22
3.1. SmartData 3.8.X overview 22
3.2. JindoFS in block storage mode 22
3.2.1. Use JindoFS in block storage mode 22
3.2.2. Use RocksDB to store metadata 26
3.2.3. Use Raft-RocksDB-Tablestore to store metadata 30
3.2.4. Use Jindo AuditLog 38
3.2.5. Access the web UI of JindoFS 42
3.2.6. Manage JindoFS permissions 44
3.2.7. Data management policies 48
3.2.8. Analyze metadata offline 50
3.2.9. Use a JindoFS credential provider 53
3.2.10. Use JindoFS in block storage mode to store encrypted... 56
3.3. JindoFS in cache mode 61
3.3.1. Use JindoFS in cache mode 61
3.3.2. Use the password-free feature of JindoFS SDK 66
3.3.3. Use Jindo AuditLog 69
3.3.4. Use Jindo Job Committer 73
3.3.5. Use JindoFS OSS credential providers 76
3.3.6. Access the web UI of JindoFS 83
3.3.7. Manage JindoFS permissions 85
3.4. JindoTable 89
3.4.1. Enable query acceleration based on a native engine 89
3.4.2. Use JindoTable 93

3.4.3. Archive and unarchive data in SDK mode 98
3.4.4. Use the MoveTo command of JindoTable 101
3.4.5. Use JindoTable to collect access frequency statistics on... 106
3.4.6. Use JindoTable to collect infrequent-access statistics of... 109
3.5. Toolset 113
3.5.1. Use Jindo SQL 113
3.5.2. Use JindoFS FUSE 122
3.5.3. Use Jindo DistCp 124
3.5.4. Use Jindo DistCp in specific scenarios 134
3.5.5. Use tiered-storage commands of JindoFS 141
4.SmartData 3.7.x 144
4.1. SmartData 3.7.X overview 144
4.2. JindoFS in block storage mode 145
4.2.1. Use JindoFS in block storage mode 145
4.2.2. Use RocksDB to store metadata 148
4.2.3. Use Raft-RocksDB-Tablestore to store metadata 153
4.2.4. Use Jindo AuditLog 159
4.2.5. Access the web UI of JindoFS 164
4.2.6. Manage JindoFS permissions 166
4.2.7. Data management policies 170
4.2.8. Analyze metadata offline 172
4.2.9. Use a JindoFS credential provider 175
4.2.10. Use JindoFS in block storage mode to store encrypted... 178
4.3. JindoFS in cache mode 183
4.3.1. Use JindoFS in cache mode 183
4.3.2. Use the password-free feature of JindoFS SDK 188
4.3.3. Use Jindo AuditLog 191
4.3.4. Use Jindo Job Committer 195

4.3.5. Use JindoFS OSS credential providers 198
4.3.6. Access the web UI of JindoFS 203
4.3.7. Manage JindoFS permissions 205
4.4. JindoTable 209
4.4.1. Enable query acceleration based on a native engine 209
4.4.2. Use JindoTable 213
4.4.3. Archive and unarchive data in SDK mode 218
4.4.4. Use the MoveTo command of JindoTable 221
4.4.5. Use JindoTable to collect access frequency statistics on... 226
4.4.6. Use JindoTable to collect infrequent-access statistics of... 229
4.5. Toolset 233
4.5.1. Use Jindo SQL 233
4.5.2. Use JindoFS FUSE 242
4.5.3. Use Jindo DistCp 244
4.5.4. Use Jindo DistCp in specific scenarios 254
4.5.5. Use tiered-storage commands of JindoFS 261
5.SmartData 3.6.x 264
5.1. SmartData 3.6.X overview 264
5.2. JindoFS in block storage mode 266
5.2.1. Use JindoFS in block storage mode 266
5.2.2. Use RocksDB to store metadata 269
5.2.3. Use Raft-RocksDB-Tablestore to store metadata 274
5.2.4. Use Jindo AuditLog 280
5.2.5. Access the web UI of JindoFS 285
5.2.6. Manage JindoFS permissions 287
5.2.7. Data management policies 291
5.2.8. Analyze metadata offline 293
5.2.9. Use a JindoFS credential provider 296

5.2.10. Use JindoFS in block storage mode to store encrypted... 299
5.3. JindoFS in cache mode 304
5.3.1. Use JindoFS in cache mode 304
5.3.2. Use the password-free feature of JindoFS SDK 309
5.3.3. Use Jindo AuditLog 312
5.3.4. Use Jindo Job Committer 316
5.3.5. Use JindoFS OSS credential providers 319
5.3.6. Access the web UI of JindoFS 324
5.3.7. Manage JindoFS permissions 326
5.4. JindoTable 330
5.4.1. Enable query acceleration based on a native engine 330
5.4.2. Use JindoTable 334
5.4.3. Archive and unarchive data in SDK mode 339
5.4.4. Use the MoveTo command of JindoTable 343
5.4.5. Use JindoTable to collect access frequency statistics on... 347
5.4.6. Use JindoTable to collect infrequent-access statistics of... 350
5.5. Toolset 354
5.5.1. Use Jindo SQL 354
5.5.2. Use JindoFS FUSE 363
5.5.3. Use Jindo DistCp 365
5.5.4. Use Jindo DistCp in specific scenarios 375
5.5.5. Use tiered storage commands of JindoFS 382
6.SmartData 3.5.x 385
6.1. SmartData 3.5.X 385
6.2. JindoFS in block storage mode 385
6.2.1. Use JindoFS in block storage mode 385
6.2.2. Use RocksDB to store metadata 389
6.2.3. Use Raft-RocksDB-Tablestore to store metadata 393

6.2.4. Use Jindo AuditLog 400
6.2.5. Access the web UI of JindoFS 405
6.2.6. Manage JindoFS permissions 407
6.2.7. Data management policies 411
6.2.8. Analyze metadata offline 413
6.2.9. Use a JindoFS credential provider 416
6.2.10. Use JindoFS in block storage mode to store encrypted... 419
6.3. JindoFS in cache mode 424
6.3.1. Use JindoFS in cache mode 424
6.3.2. Use the password-free feature of JindoFS SDK 429
6.3.3. Use Jindo AuditLog 432
6.3.4. Use Jindo Job Committer 436
6.3.5. Use JindoFS OSS credential providers 439
6.3.6. Access the web UI of JindoFS 444
6.3.7. Manage JindoFS permissions 446
6.4. JindoTable 450
6.4.1. Enable query acceleration based on a native engine 450
6.4.2. Use JindoCube 453
6.4.3. Use JindoTable 464
6.4.4. Use JindoTable to collect access frequency statistics on... 468
6.4.5. Use JindoTable to collect infrequent-access statistics of... 471
6.5. Toolset 475
6.5.1. Use JindoFS FUSE 475
6.5.2. Use Jindo DistCp 477
6.5.3. Use Jindo DistCp in specific scenarios 487
6.5.4. Use tiered storage commands of JindoFS 494
7.SmartData 3.4.x 497
7.1. SmartData 3.4.X 497

7.2. JindoFS in block storage mode 497
7.2.1. Use JindoFS in block storage mode 497
7.2.2. Use RocksDB to store metadata 501
7.2.3. Use Raft-RocksDB-Tablestore to store metadata 505
7.2.4. Use Jindo AuditLog 512
7.2.5. Access the web UI of JindoFS 517
7.2.6. Manage JindoFS permissions 519
7.2.7. Data management policies 523
7.2.8. Analyze metadata offline 525
7.2.9. Use a JindoFS credential provider 528
7.2.10. Use JindoFS in block storage mode to store encrypted... 531
7.3. JindoFS in cache mode 536
7.3.1. Use JindoFS in cache mode 536
7.3.2. Use the password-free feature of JindoFS SDK 541
7.3.3. Use Jindo AuditLog 544
7.3.4. Use Jindo Job Committer 548
7.3.5. Use JindoFS OSS credential providers 551
7.3.6. Access the web UI of JindoFS 556
7.3.7. Manage JindoFS permissions 558
7.4. JindoTable 562
7.4.1. Enable query acceleration for ORC files 562
7.4.2. Use JindoTable 564
7.4.3. Use JindoCube 569
7.4.4. Use JindoTable to collect access frequency statistics on... 580
7.5. Toolset 582
7.5.1. Use JindoFS FUSE 582
7.5.2. Use Jindo DistCp 584
7.5.3. Use Jindo DistCp in specific scenarios 594

7.5.4. Use tiered storage commands of JindoFS 601
8.SmartData 3.2.X 604
8.1. SmartData 3.2.X 604
8.2. JindoFS in block storage mode 604
8.2.1. Use JindoFS in block storage mode 604
8.2.2. Use RocksDB to store metadata 608
8.2.3. Use Raft-RocksDB-Tablestore to store metadata 612
8.2.4. Use Jindo AuditLog 619
8.2.5. Access the web UI of JindoFS 624
8.2.6. Manage JindoFS permissions 626
8.2.7. Data management policies 630
8.2.8. Analyze metadata offline 632
8.3. JindoFS in cache mode 635
8.3.1. Use JindoFS in cache mode 635
8.3.2. Use the password-free feature of JindoFS SDK 641
8.3.3. Use Jindo AuditLog 643
8.3.4. Use Jindo Job Committer 647
8.3.5. Use a credential provider 650
8.3.6. Access the web UI of JindoFS 655
8.3.7. Manage JindoFS permissions 657
8.4. JindoTable 661
8.4.1. Enable query acceleration for ORC files 661
8.4.2. Use JindoTable 663
8.4.3. Use JindoCube 668
8.4.4. Use JindoTable to collect access frequency statistics on... 679
8.5. Toolset 681
8.5.1. Use JindoFS FUSE 681
8.5.2. Use Jindo DistCp 683

8.5.3. Use Jindo DistCp in specific scenarios 693
8.5.4. Use tiered storage commands of JindoFS 700
9.SmartData 3.1.x 703
9.1. SmartData 3.1.x 703
9.2. JindoFS in block storage mode 705
9.2.1. Use JindoFS in block storage mode 705
9.2.2. Use RocksDB to store metadata 708
9.2.3. Use Raft-RocksDB-Tablestore to store metadata 713
9.2.4. Use Jindo AuditLog 719
9.2.5. Access the web UI of JindoFS 724
9.2.6. Manage JindoFS permissions 726
9.2.7. Data management policies 730
9.2.8. Analyze metadata offline 732
9.3. JindoFS in cache mode 735
9.3.1. Use JindoFS in cache mode 735
9.3.2. Use the password-free feature of JindoFS SDK 741
9.3.3. Use Jindo AuditLog 743
9.3.4. Use Jindo Job Committer 747
9.3.5. Use a credential provider 750
9.3.6. Access the web UI of JindoFS 755
9.3.7. Manage JindoFS permissions 757
9.4. JindoTable 761
9.4.1. Use JindoTable 761
9.4.2. Use JindoTable to collect access frequency statistics of ... 766
9.4.3. Use EMR JindoCube 768
9.5. Toolset 779
9.5.1. Use JindoFS FUSE 779
9.5.2. Use Jindo DistCp 780

9.5.3. Use Jindo DistCp in specific scenarios 790
9.5.4. Use tiered storage commands of JindoFS 798
10.SmartData 3.0.x 800
10.1. SmartData 3.0.x 800
10.2. JindoFS in block storage mode 801
10.2.1. Use JindoFS in block storage mode 801
10.2.2. Use RocksDB to store metadata 805
10.2.3. Use Raft-RocksDB-Tablestore to store metadata 809
10.2.4. Access the web UI of JindoFS 816
10.2.5. Manage JindoFS permissions 818
10.2.6. Use Jindo AuditLog 822
10.2.7. Analyze metadata offline 827
10.3. JindoFS in cache mode 830
10.3.1. Use JindoFS in cache mode 830
10.3.2. Use the password-free feature of JindoFS SDK 835
10.3.3. Access the web UI of JindoFS 838
10.3.4. Manage JindoFS permissions 840
10.3.5. Use Jindo Job Committer 843
10.3.6. Use Jindo AuditLog 846
10.3.7. Use a credential provider 851
10.4. JindoTable 854
10.4.1. Use JindoTable 854
10.4.2. Use EMR JindoCube 859
10.5. Toolset 870
10.5.1. Use JindoFS FUSE 870
10.5.2. Use tiered storage commands of JindoFS 871
10.5.3. Use Jindo DistCp 873
10.5.4. Use Jindo DistCp in specific scenarios 883

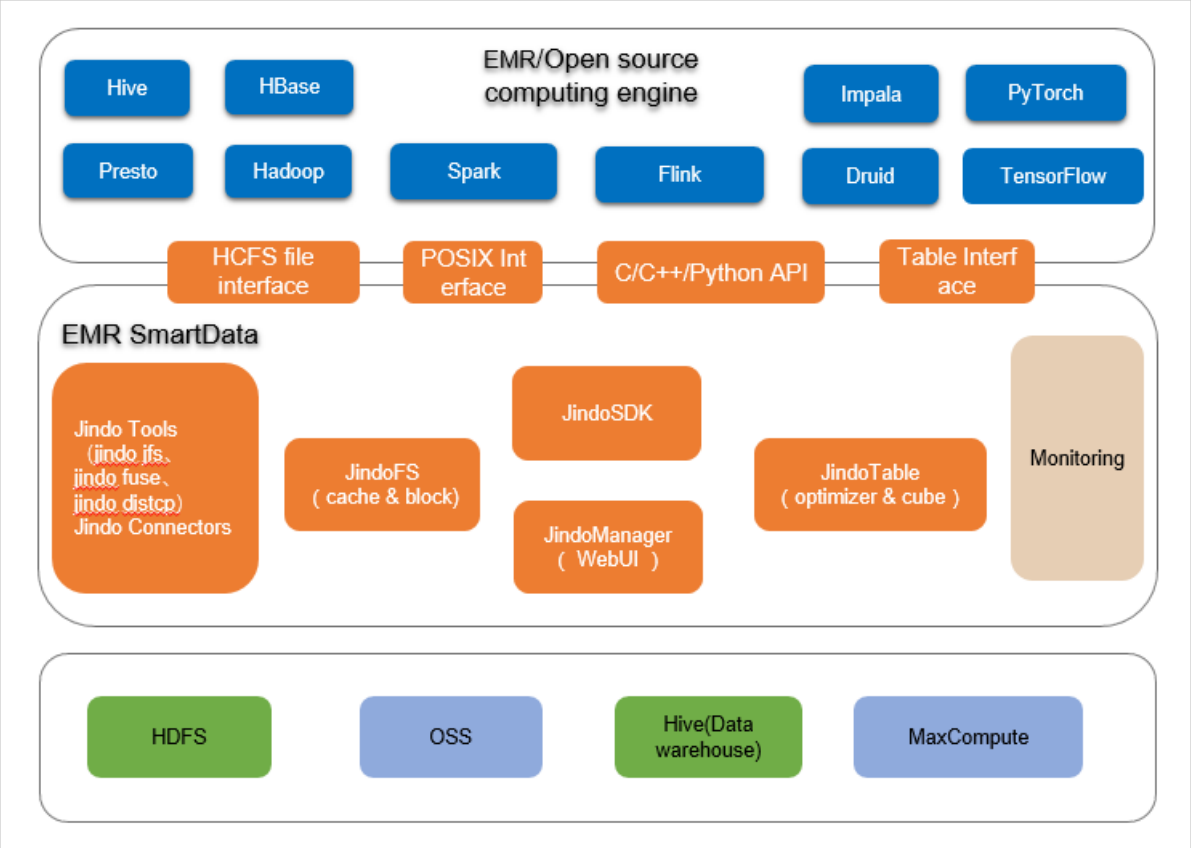
11.SmartData 2.7.301 891
11.1. JindoFS in block storage mode 891
11.1.1. Use JindoFS in block storage mode 891
11.1.2. Use Tablestore instances to store metadata 894
11.1.3. Use RocksDB to store metadata 897
11.1.4. Use Raft-RocksDB-Tablestore to store metadata 899
11.1.5. Manage JindoFS permissions 905
11.1.6. Use Jindo AuditLog 909
11.2. JindoFS in cache mode 913
11.2.1. Use JindoFS in cache mode 913
11.2.2. Use the password-free feature of JindoFS SDK 918
11.2.3. Use Jindo Job Committer 920
11.2.4. Manage JindoFS permissions 923
11.2.5. Use Jindo AuditLog 927
11.3. JindoTable 931
11.3.1. Use EMR JindoCube 931
11.4. Toolset 942
11.4.1. Use Jindo DistCp 942
11.4.2. Use JindoFS FUSE 952
12.SmartData 2.6.0-2.7.2 954
12.1. SmartData 2.6.X 954
12.2. JindoFS in block storage mode 954
12.2.1. Use JindoFS in block storage mode 954
12.2.2. Use Tablestore instances to store metadata 957
12.2.3. Use RocksDB to store metadata 961
12.2.4. Use Raft-RocksDB-Tablestore to store metadata 962
12.2.5. Manage JindoFS permissions 968
12.3. JindoFS in cache mode 972

12.3.1. Use JindoFS in cache mode 972
12.3.2. Manage JindoFS permissions 977
12.3.3. Use Jindo Job Committer 981
12.4. JindoTable 984
12.4.1. Use JindoCube 984
12.5. Toolset 995
12.5.1. Use Jindo DistCp 995
13.SmartData 2.2.x and earlier 1005
13.1. Use JindoFS in EMR V3.20.0 to V3.22.0 (V3.22.0 excluded) 1005
13.2. Use JindoFS in EMR V3.22.0 to V3.26.3 1012
13.3. Use the block storage mode 1019
13.4. Use the cache mode 1024
13.5. Use the password-free feature of JindoFS SDK 1027
13.6. Use the external client 1029
14.Best Practices 1031
14.1. Migrate data from HDFS to JindoFS 1031
14.2. Use MapReduce to process data in JindoFS 1031
14.3. Use Hive to query data in JindoFS 1032
14.4. Use Spark to process data in JindoFS 1035
14.5. Use Flink to process data in JindoFS 1035
14.6. Use Impala or Presto to query data in JindoFS 1036
14.7. Use JindoFS as the storage backend of HBase 1037
14.8. Store the logs of YARN MapReduce and Spark jobs 1040
14.9. Import data from Kafka to JindoFS 1044
14.10. Access JindoFS across clusters 1045
14.11. Change the HDFS client path for the Jindo system 1046
15.SmartData FAQ 1050

1.Overview

SmartData is a core self-developed component of E-MapReduce (EMR). SmartData optimizes storage, caching, and computing for various EMR computing engines in a centralized manner and extends storage features. SmartData is used in data access, data governance, and data security scenarios.

The following figure shows the position of SmartData in EMR.



Composition of SmartData:

- JindoFS core subsystem: provides caching and cache-based acceleration features for various remote storage systems. For more information, see Overview and usage of JindoFS.
- JindoTable core subsystem: provides table- and partition-level optimization and governance for data sources, such as a Hive warehouse. For more information, see Use JindoTable.
- JindoManager: provides a web UI to manage JindoFS and JindoTable services and features. For example, you can view the metrics of the cached data of files and tables.
- JindoSDK: provides a unified SDK for various open source computing engines of EMR. It supports Java, C, C++, and Python programming languages and provides a variety of access interfaces and APIs, such as HCFS interfaces, Portable Operating System Interface (POSIX) interfaces, and table-related interfaces.
- Toolset: includes Jindo tools and the data copy tool Jindo DistCp.
- Various connectors: include the Hadoop connector, Flink connector, and TensorFlow connector. Kite SDK, Apache Beams, Flume, Sqoop, and Kafka are supported.

The data sources that are supported by JindoFS and JindoTable include Alibaba Cloud OSS, Apache Hadoop HDFS, Hive, and Alibaba Cloud MaxCompute.

SmartData is independently developed and released. For more information about SmartData versions, see Overview.

For more information about how to use SmartData, see the following topics:

- SmartData 3.5.X
- SmartData 3.4.X
- SmartData 3.2.X
- SmartData 3.1.x
- SmartData 3.0.x
- SmartData 2.7.3 to 2.7.4
- SmartData 2.6.X
- Use JindoFS in EMR V3.22.0 to V3.26.3
- Use JindoFS in EMR V3.20.0 to V3.22.0 (V3.22.0 excluded)

2.Overview and usage of JindoFS

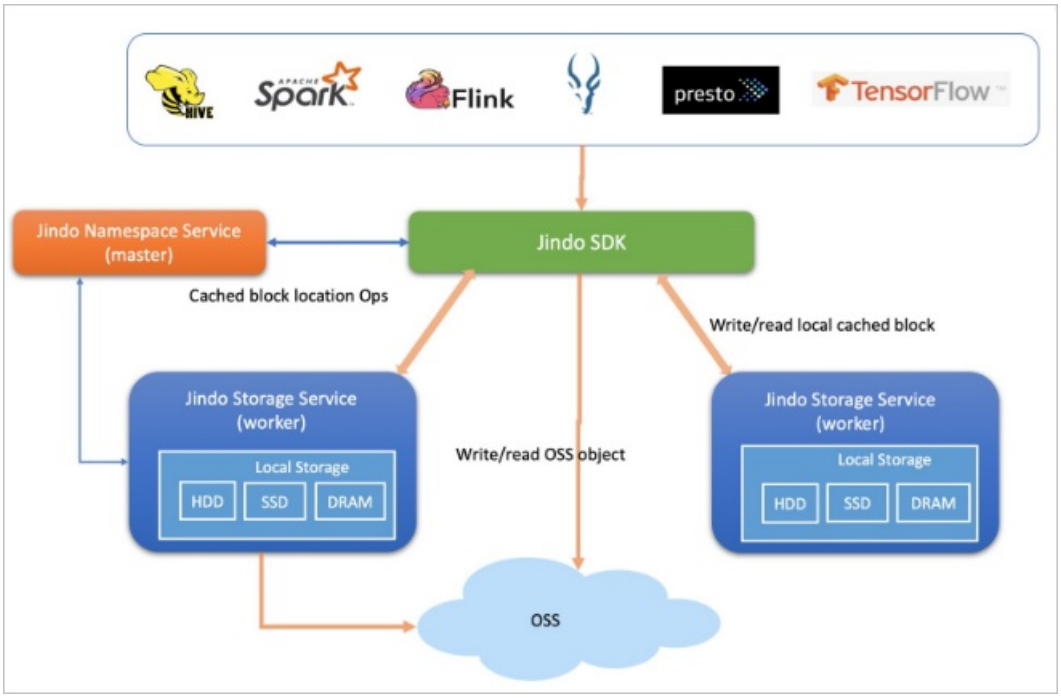
JindoFS is a Hadoop-compatible file system (HCFS) built for open source big data ecosystems based on Alibaba Cloud Object Storage Service (OSS). JindoFS provides three storage modes to store data in OSS: client-only mode (SDK), cache mode, and block storage mode. JindoFS in client-only mode or cache mode optimizes the access to OSS from computing engines of Hadoop and Spark ecosystems. JindoFS in block storage mode provides a tremendous storage capacity by using OSS as the storage backend and supports efficient metadata queries.

Client-only mode (SDK)

In this mode, JindoFS provides features similar to OSS FileSystem and S3A FileSystem in the Hadoop community. JindoFS optimizes the access to Alibaba Cloud OSS and various operations on data for computing frameworks such as Hive and Spark. This mode does not change the way files or objects are organized in OSS. Files are still stored in OSS. JindoFS provides only client connection, extension, adaptation, and optimized access for the Hadoop ecosystem. To use JindoFS in this mode, you need only to upload the JAR package of JindoFS SDK to the *classpath* directory. This mode is simple and easy to use and requires no deployment of distributed services.

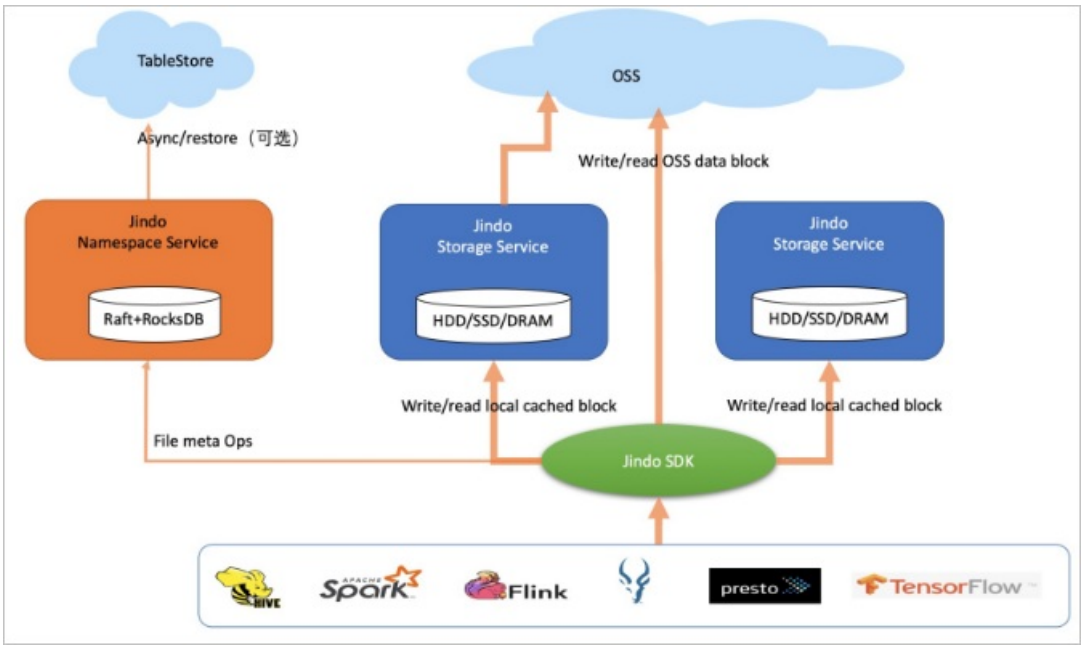
Cache mode

This mode is compatible with the client-only mode (SDK) and accelerates data caching for OSS by using the distributed data caching capability of Jindo. This helps meet large-scale data analysis requirements and throughput-related requirements for training. On the basis of the client-only mode (SDK), the cache mode supports metadata caching and distributed data caching and maintains data compatibility and synchronization with OSS. Data can be cached in memory, SSDs, and basic disks to suit different computing scenarios.



Block storage mode

In this mode, JindoFS provides features similar to Apache Hadoop HDFS. JindoFS can cache data to accelerate data access. It can also organize and store data and manage file metadata. In this mode, JindoFS serves as an independent storage system, but files are stored as blocks in OSS.



Comparison between the cache mode and block storage mode

Both modes allow JindoFS to store data in OSS and determine whether to cache the data in local clusters to accelerate data access based on the usage of local storage space.

The essential difference between the two modes lies in the file storage methods in OSS. JindoFS in block storage mode manages directories and file metadata and stores files as blocks in OSS. JindoFS in cache mode stores files as objects in OSS.

Comparison among the three modes

The following table describes the three modes in multiple dimensions.

Dimension	Client-only mode (SDK)	Cache mode	Block storage mode
Storage cost	• Stores full data in OSS. • Supports the Archive storage class.	• Stores full data in OSS. • Caches hot data, which accounts for 20% of the total amount of data. • Supports the Archive storage class.	• Stores full data in OSS. • Caches warm data and hot data, which account for 60% of the total amount of data. • Supports the Archive storage class. • Supports transparent compression.
Scalability	High	Relatively high	Medium
Throughput	Depends on the bandwidth occupied by OSS.	Depends on the bandwidth occupied by OSS and the bandwidth consumed for caching hot data.	Depends on the bandwidth occupied by OSS and the bandwidth consumed for caching warm data and hot data.
Metadata	• Simulates HDFS to manage metadata and does not support directory-based storage and file semantics. • Supports exabytes of data.	• Simulates HDFS to manage metadata and does not support directory-based storage and file semantics. JindoFS can cache file data. • Supports exabytes of data.	• Provides the highest performance. The compatibility of JindoFS in this mode is close to that of HDFS. • Supports more than 1 billion files.
Maintenance workload	Low	Medium Requires the O&M of the cache system.	Relatively high Requires the O&M of the Namespace Service and Storage Service.

Dimension	Client-only mode (SDK)	Cache mode	Block storage mode
Security	• Supports AccessKey pair-based authentication. • Supports RAM authentication. • Supports OSS access logs. • Supports encryption of OSS data.	• Supports AccessKey pair-based authentication. • Supports RAM authentication. • Supports OSS access logs. • Supports encryption of OSS data.	• Supports AccessKey pair-based authentication. • Allows you to run UNIX commands or use Ranger to manage permissions of JindoFS in this mode. • Supports audit logs generated by AuditLog. • Supports data encryption.
Usage	Only allows you to specify an OSS directory in the *oss://<oss_bucket>/<oss_dir>/* format to access files. Cross-service access to the OSS directory is supported.	• Allows you to specify an OSS directory in the *oss://<oss_bucket>/<oss_dir>/* format to access files. Cross-service access to the OSS directory is supported. The caching feature can be enabled. This is the default method. • Allows you to specify a JindoFS directory in the *jfs://<your_namespace>/<path_of_file>* format for one of the deployed namespaces to access data. Cross-service access to the JindoFS directory is not supported. The caching feature can be enabled. **Note** For more information about how to use JindoFS in cache mode, see the documentation of JindoFS in cache mode.	Only allows you to specify a JindoFS directory in the *jfs://<your_namespace>/<path_of_file>* format for one of the deployed namespaces to access data. Cross-service access to the JindoFS directory is not supported. The caching feature can be enabled. **Note** For more information about how to use JindoFS in block storage mode, see the documentation of JindoFS in block storage mode.

FAQ

- Q: What mode is recommended for typical data lake scenarios?

  A: The client-only mode (SDK) and the cache mode are fully compatible with object storage semantics of OSS and provide complete compute-storage separation and flexible scalability. We recommend that you use the client-only mode (SDK) or the cache mode for big data analysis and AI training acceleration in typical data lake scenarios.

- Q: Why does JindoFS in block storage mode provide higher performance than HDFS?

  A:

  - JindoFS in block storage mode can process more than 1 billion files. However, HDFS can process only a maximum of 0.4 billion files. In addition, the performance of JindoFS in block storage mode is more stable at peak hours of cluster business.
  - JindoFS in block storage mode has no limits on on-heap memory in Java and memory usage and can process data at a larger scale than HDFS. HDFS has limits on on-heap memory in Java.
  - JindoFS in block storage mode requires lightweight O&M. You do not need to worry about damaged disks or anomalous nodes. Data has one backup on OSS, and nodes can be connected or disconnected.
  - JindoFS in block storage mode can transparently compress and archive cold data. It uses various means to optimize costs and connects to OSS to support exabytes of data.
  - JindoFS in block storage mode supports some important features of HDFS, such as HDFS AuditLog, integration with Ranger, and data encryption.

- Q: What are the special advantages of JindoFS in block storage mode?

  A:

  - JindoFS in block storage mode can manage file metadata and organize file data. Therefore, it can fully meet the requirements of various big data engines on storage interfaces. These interfaces include but are not limited to the interface to implement the atomicity and transaction processing of rename operations, the interface to implement high-performance local data writing, the interface to implement transparent compression, and the truncate, append, flush, sync, and snapshot interfaces. These high-level storage interfaces are required to achieve complete POSIX and are used to connect more big data engines, such as Flink, HBase, Kafka, and Kudu, to OSS. JindoFS in client-only mode (SDK) or JindoFS in cache mode can also use some interfaces to access OSS. However, the capabilities and advantages of these two modes are insufficient.
  - The block storage mode is more cost-effective than the other two modes. This is because in block storage mode, warm data and hot data, which account for 60% of the total amount of data, are cached in local clusters. Therefore, you can read a large amount of data from your local cluster instead of OSS.

3.SmartData 3.8.x

3.1. SmartData 3.8.X overview

SmartData is a storage service for the E-MapReduce (EMR) Jindo engine. SmartData provides centralized storage, caching, and computing optimization for EMR computing engines. This service also provides additional features. SmartData consists of JindoFS, JindoTable, and related tools. This topic describes the updates in SmartData 3.8.X.

JindoSDK

Feature	Description
Support for AssumeRoleStsCredentialsProvider	This credential provider is suitable for scenarios in which a temporary AccessKey pair that assumes a RAM role is obtained to access Alibaba Cloud Object Storage Service (OSS). For more information, see Use JindoFS OSS credential providers.
Support for JindoRangerCredentialsProvider	This credential provider is suitable for scenarios in which you configure parameters in Ranger to control access to OSS. For more information, see Use JindoFS OSS credential providers.

3.2. JindoFS in block storage mode

3.2.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read and write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

Context

The block storage mode of JindoFS has the following features:

- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.
- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.

i.

ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

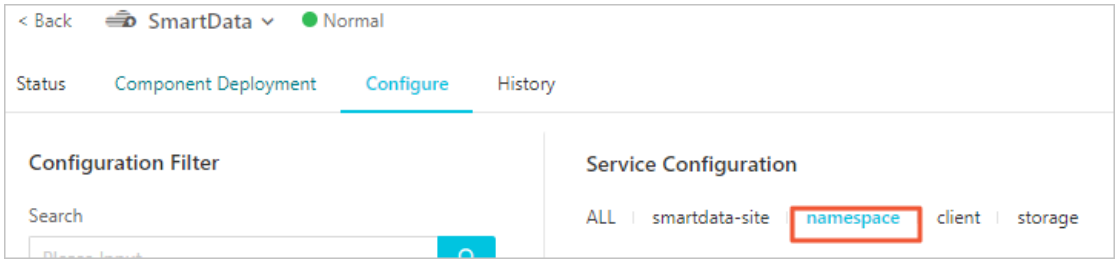
iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **namespace** tab for the SmartData service.

   i. Click the **Configure** tab.

   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

   If you configure multiple namespaces, separate them with commas (,).

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

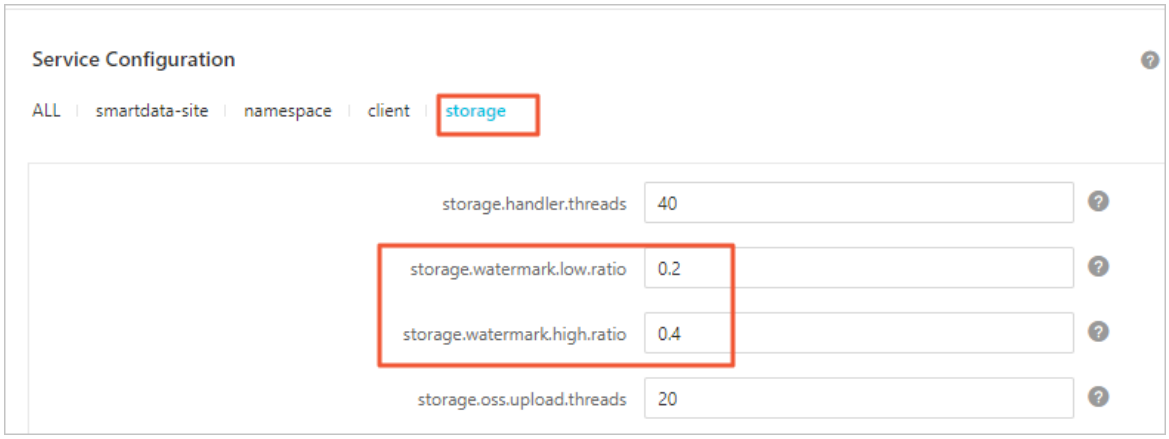
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Choose **Actions > Restart Jindo Storage Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

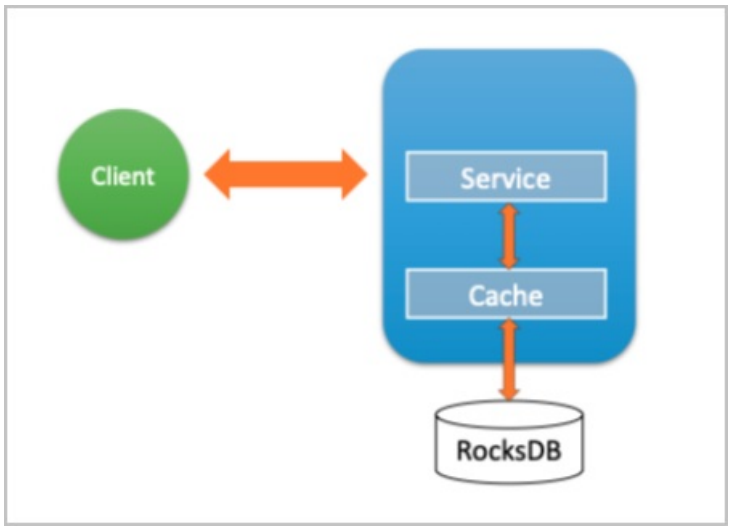
3.2.2. Use RocksDB to store metadata

Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

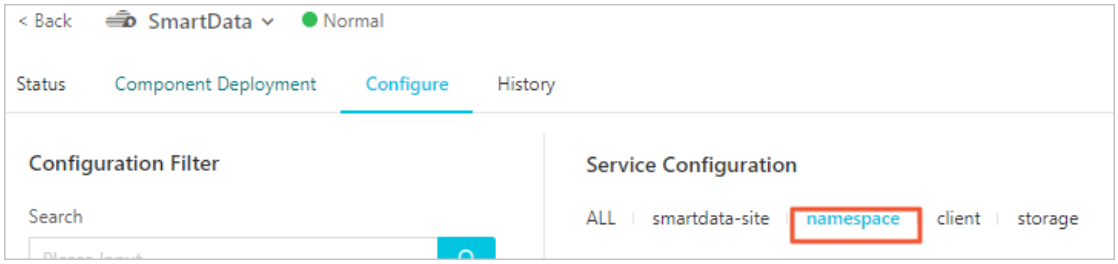
RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage backend in high availability mode, we recommend that you use Raft instances. For more information, see Use Raft-RocksDB-Tablestore to store metadata.

The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
6. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

7. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

8. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. Prepare for the recovery.

   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

   ```
   hadoop fs -count jfs://test/
   ```

   Information similar to the following example is returned:

   ```
    1596      1482809                 25 jfs://test/
   ```

   The number of folders is 1596, and the number of files is 1,482,809.

   ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1`, the latest metadata is synchronized to the Tablestore instance.

   ```
   jindo jfs -metaStatus
   ```

   Information similar to that shown in the following figure is returned.

   ```
   [hadoop@emr-header-1 ~]$ jindo jfs -metaStatus
   ==== emr-header-1:8101 ====
   OtsUploader: _synced=1
   [RocksDB Row Counts of each CF(Table)]
   ```

   iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

   ```
   jindo jfs -metaStatus
   ```

   If the state is FINISH, the recovery is completed, as shown in the following figure.

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

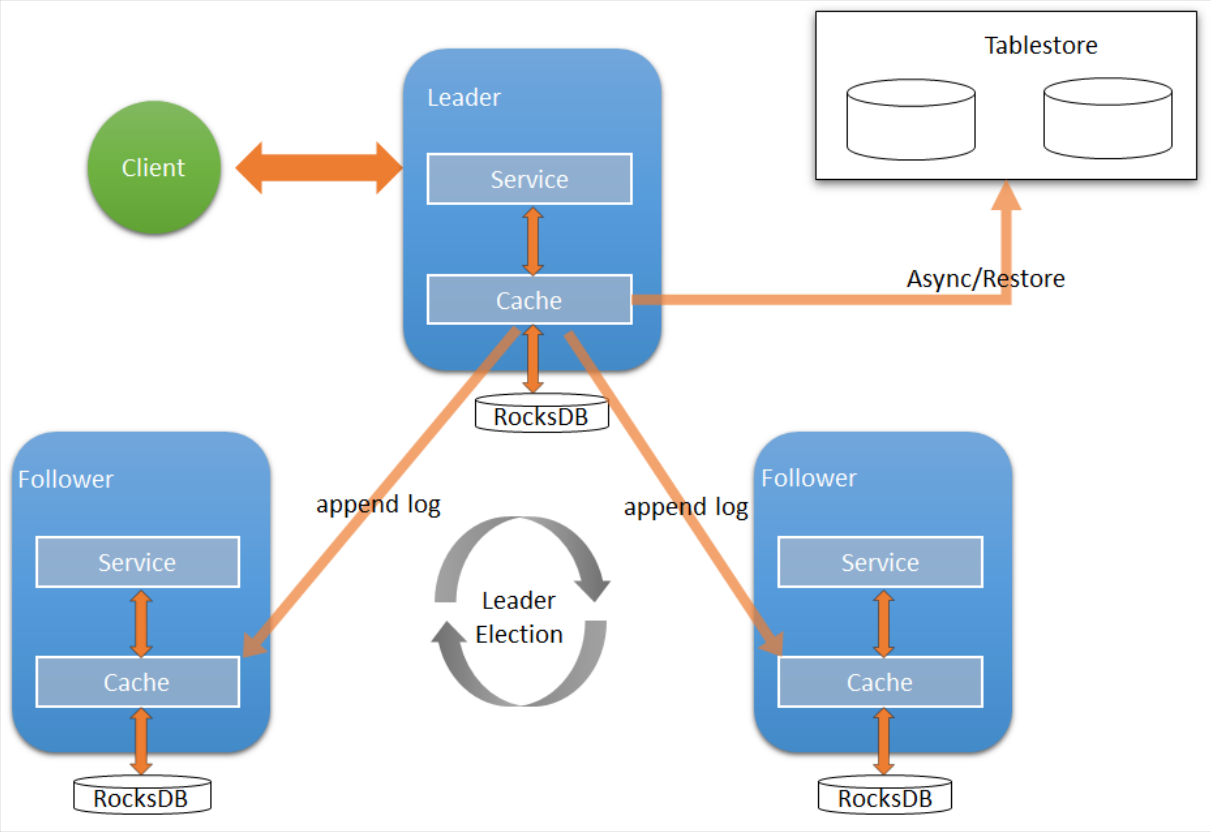
3.2.3. Use Raft-RocksDB-Tablestore to store metadata

In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata. The Raft instance provides a built-in disaster recovery capability. If necessary, you can also back up the metadata in a Tablestore instance for disaster recovery.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.

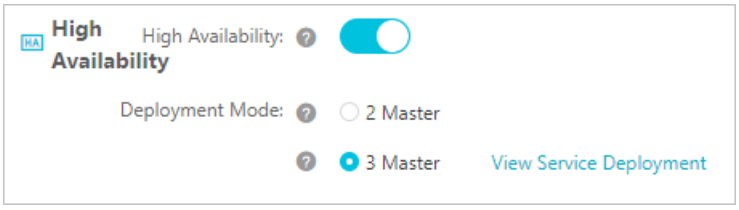


Prerequisites

- (Optional)A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  > **Note** The transaction feature provided by the Tablestore instance is enabled.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.



Note If no deployment modes are available, .

Configure a Raft instance as the local storage backend

1. Go to the SmartData service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane of the page that appears, choose **Cluster Service > SmartData**.
2. Suspend all SmartData components.
   i. In the upper-right corner of the SmartData service page, choose **Actions > Stop All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.
3. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
4. Go to the **namespace** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **namespace** tab.
5. Configure the parameters that are described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. Valid values: ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft

Parameter	Description	Example
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

> **Note** If you do not need to use Tablestore for remote storage, perform only Step 7 and Step 9. If you need to use Tablestore for remote storage, perform Step 6 to Step 9.

6. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   Configure the parameters that are described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not complete. **Note** If the initialization is complete, the setting does not take effect because metadata has been generated in the local RocksDB.	true

7. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
8. Start all SmartData components.
   i. In the upper-right corner of the SmartData service page, choose **Actions > Start All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.
9. Log on to your cluster in SSH mode and run the following command to view the metadata status:

   ```
   jindo jfs -metaStatus -detail
   ```

   If the returned information contains raft, the configuration takes effect.

Recover metadata from a Tablestore instance

> **Notice** Make sure that a Tablestore instance is configured as the remote asynchronous storage backend for your cluster.

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. View the metadata information of the original EMR cluster. The metadata information includes the numbers of files and folders.

   ```
   hadoop fs -count jfs://test/
   ```

   The following information is returned:

   ```
   1596      1482809                 25 jfs://test/
   ```

   > **Note** 1596 is the number of folders. 1482809 is the number of files.

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all the metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```

```
[RaftPeerImpl]
peer id: [illegible]
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [illegible]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 2335ms)
snapshot_timer: timeout(3600000ms) SCHEDULING(in 150305ms)
storage: [1, 624625]
disk_index: 624625
known_applied_index: 624625
last_log_id: (index=624625,term=2)
first_index_pinned: 624625
state_machine: Idle
last_committed_index: 624625
last_snapshot_index: 0
last_snapshot_term: 0
snapshot_status: IDLE
replicator_25769803789@1[illegible] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=624261 ic=0
replicator_329853488332[illegible] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=623564 ic=0

OtsUploader: _lastStopIndex=624624, _synced=1
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   Configure the parameters that are described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Start all SmartData components.

   i. In the upper-right corner of the SmartData service page, choose **Actions > Start All Components**.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.

```
[RaftPeerImpl]
peer_id: [redacted]:8103:0
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 3382ms)
snapshot_timer: timeout(600000ms) SCHEDULING(in 474855ms)
storage: [1, 153]
disk_index: 153
known_applied_index: 153
last_log_id: (index=153,term=2)
first_index_pinned: 1
state_machine: Idle
last_committed_index: 153
last_snapshot_index: 1
last_snapshot_term: 2
snapshot_status: IDLE
replicator_1116691496965@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0
replicator_3311419785217@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0

[Recovery From OTS Status]
state: FINISH
total rows: 1484409
table `jfs_block_test` 2 rows.
table `jfs_namespace_cache_ns` 1 rows.
table `jfs_namespace_test` 1484406 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   - Run the following command to check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster:

```
hadoop fs -count jfs://test/
```

The following information is returned:

```
1596      1482809                 25 jfs://test/
```

- Run the following command to check whether data can be read from the files:

```
hadoop fs -cat jfs://test/testfile
```

- Run the following command to view file directories:

```
hadoop fs -ls jfs://test/
```

The following information is returned:

```
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.clus
ter-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.
txt
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
```

- Run the following command to check whether a file can be removed:

```
hadoop fs -rm jfs://test/testfile
```

The following information is returned:

```
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode,
and is read-only.
```

> **Note** The new EMR cluster is in recovery mode and is read-only. Files cannot be removed.

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters that are described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster and choose **More > Restart** in the Actions column.

iii. In the **Restart Cluster** dialog box, click **OK**.

3.2.4. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

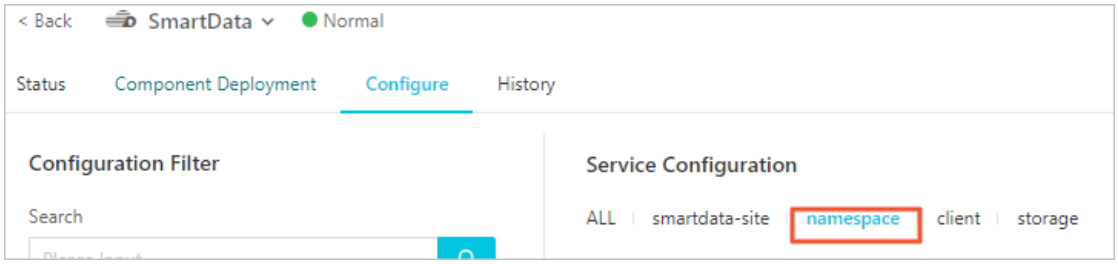
Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

3.2.5. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

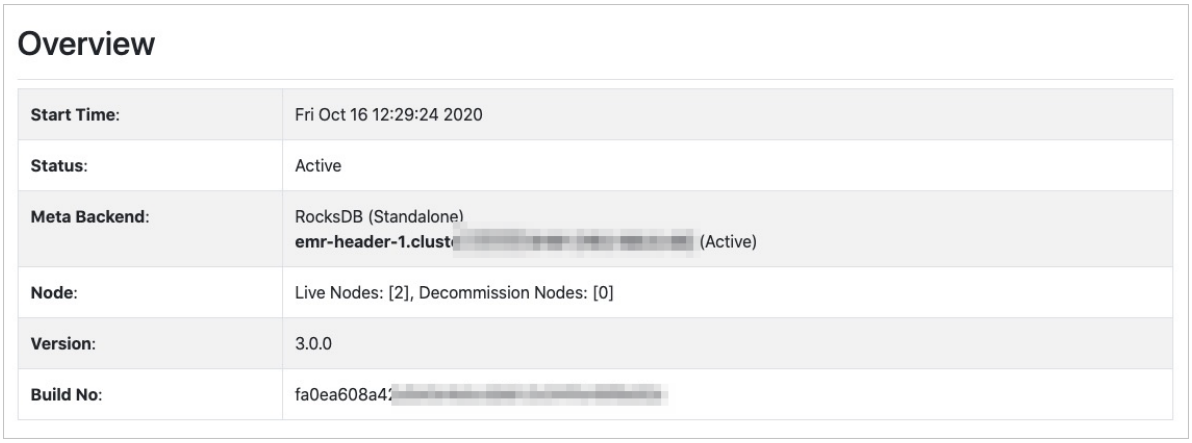
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

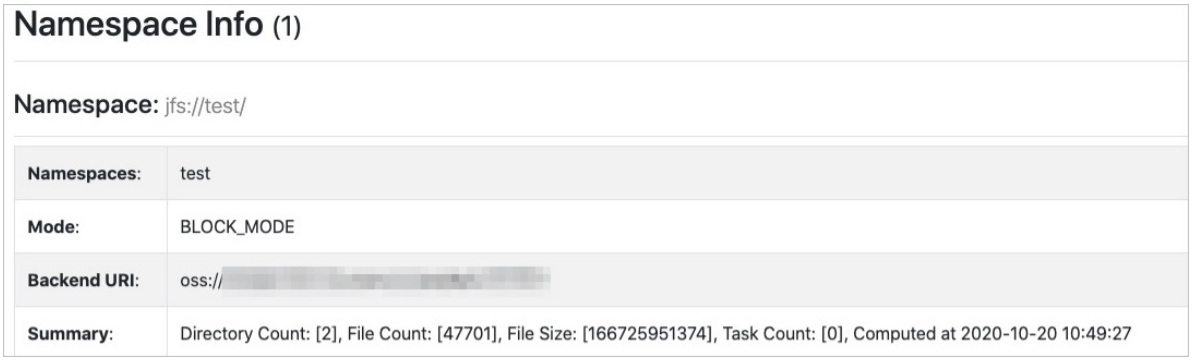


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



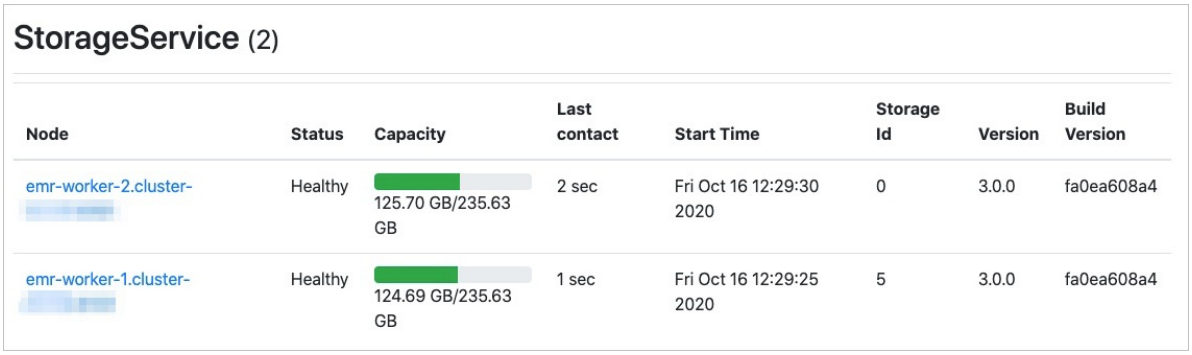
Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

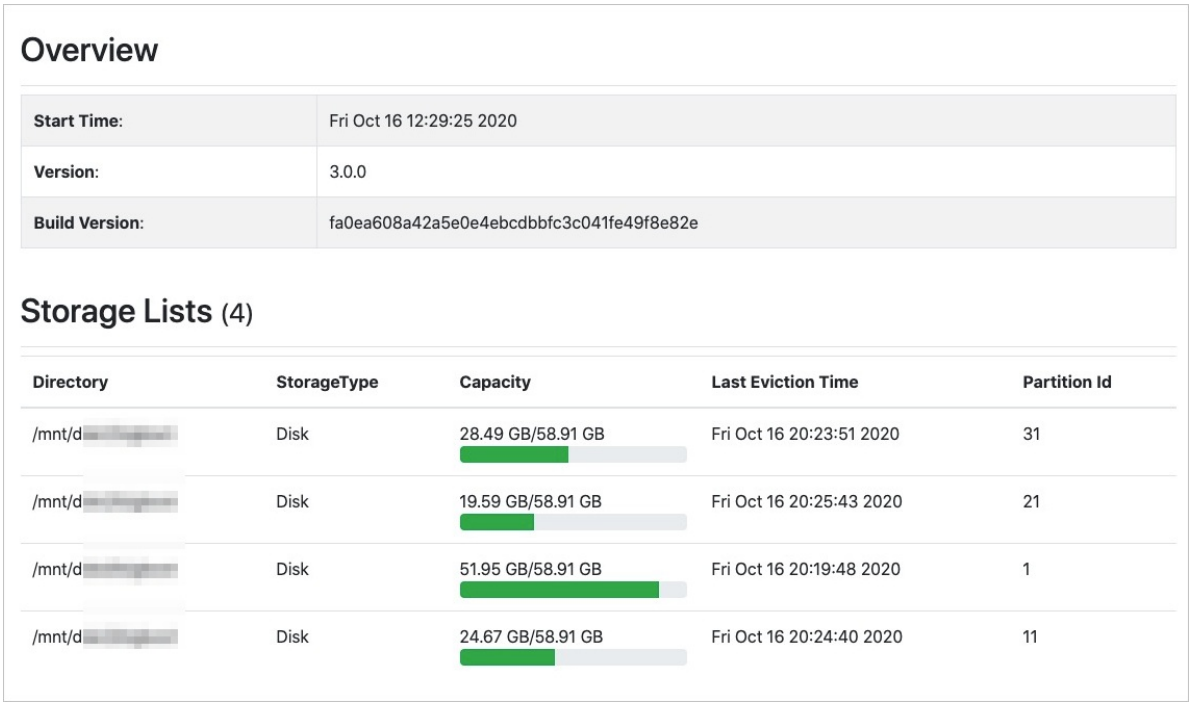
  This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.



StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.



Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

3.2.6. Manage JindoFS permissions

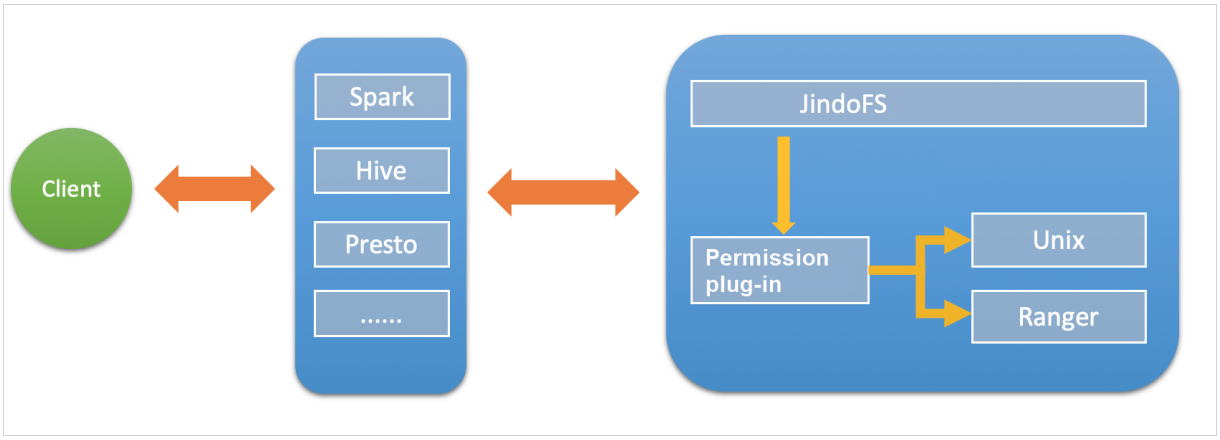
This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

Permission management based on the storage mode:

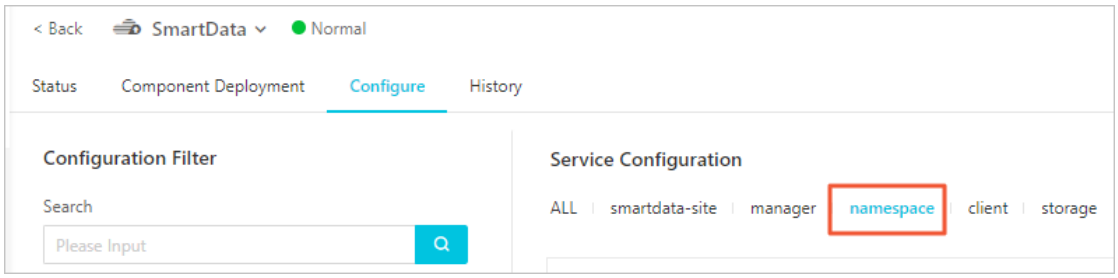
- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.
  - Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

3.2.7. Data management policies

JindoFS in block storage mode provides advanced data management policies to meet storage requirements in different scenarios. The advanced policies include storage policies and compression policies. This topic describes the policies in detail and provides examples to show how to use the policies.

Storage policies

JindoFS provides the five storage policies described in the following table to support flexible storage.

Policy	Description
AR	Data has only one backup. The backup is stored as an Archive object in OSS.
IA	Data has only one backup. The backup is stored as an Infrequent Access (IA) object in OSS.

Policy	Description
COLD	Data has only one backup. The backup is stored as a Standard object in OSS.
WARM	Data has one backup in OSS and one local backup. The local backup can effectively accelerate data read operations. By default, this storage policy is used.
HOT	Data has one backup in OSS and one local backup. The local backup is forcibly locked. This way, it is not deleted when the cache is automatically cleared. This policy can achieve a better acceleration effect than the WARM policy and is suitable for the hottest data.

For more information about the storage classes of OSS, see Overview.

A newly added file is stored based on the storage policy specified for its parent directory. Example:

- Run the following command to configure a storage policy:

```
jindo jfs -setStoragePolicy [-R] <StoragePolicy>(AR/IA/COLD/WARM/HOT) <path> ...
```

  where:

  - `[-R]`: specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.
  - `<path>`: the name of the directory for which the storage policy is configured.

- Run the following command to obtain the storage policy configured for a directory:

```
jindo jfs -getStoragePolicy <path>
```

Compression policies

JindoFS allows you to compress data blocks before you store them. This feature effectively reduces the storage space occupied by data blocks and improves data read and write efficiency. This feature is suitable for files that have a high compression ratio. The following table describes the supported compression policies.

Policy	Description
NONE	Data blocks are not compressed. By default, this compression policy is used.
ZSTD	The Zstandard compression algorithm is used to compress data blocks.

A newly added file is compressed based on the compression policy specified for its parent directory before it is stored. Example:

- Run the following command to configure a compression policy:

```
jindo jfs -setCompressionPolicy [-R] <CompressionPolicy>(NONE/ZSTD) <path> ...
```

where:

- `[-R]`: specifies that a recursive operation is performed to configure the same compression policy for all sub-directories of the directory.
- `<path>`: the name of the directory for which the compression policy is configured.

- Run the following command to obtain the compression policy configured for a directory:

```
jindo jfs -getCompressionPolicy <path> ...
```

3.2.8. Analyze metadata offline

In E-MapReduce (EMR) V3.30.0 or later, JindoFS in block storage mode allows you to dump the metadata of an entire namespace to Object Storage Service (OSS) and use Jindo SQL to analyze the metadata.

Context

In Hadoop Distributed File System (HDFS), the metadata is stored in a snapshot file named fsimage. The snapshot file contains the following information about metadata: namespace, file that stores the metadata, blocks, and file system quota. In HDFS, you can run a command to download the fsimage file in the XML format to your on-premises machine and view this file to analyze metadata offline. JindoFS does not require you to download metadata to your on-premises machine.

Upload HDFS metadata to OSS

Run the following command to upload metadata of a specific namespace to OSS:

```
jindo jfs -dumpMetadata <nsName>
```

`<nsName>` indicates the name of the namespace in block storage mode.

For example, upload the metadata of the test-block namespace to OSS and analyze the metadata offline.

```
jindo jfs -dumpMetadata test-block
```

If the following information appears, data is uploaded to OSS and stored as a file in the JSON format.

```
Sucessfully upload namespace metadata to OSS.
```

Metadata upload path

The metadata upload path is the metadataDump sub-directory under *sysinfo* in JindoFS.

For example, if `namespace.sysinfo.oss.uri` is set to `oss://abc/`, metadata is uploaded to the `oss://abc/metadataDump` directory.

Parameter	Description
namespace.sysinfo.oss.uri	The path of the OSS bucket.
namespace.sysinfo.oss.endpoint	The endpoint. An endpoint specified in a different region can also be used.
namespace.sysinfo.oss.access.key	The AccessKey ID of your Alibaba Cloud account.
namespace.sysinfo.oss.access.secret	The AccessKey secret of your Alibaba Cloud account.

Batch information: Metadata from HDFS changes based on the usage. Each time you want to analyze metadata, you must run commands to view the snapshot file of the metadata. Each time you run a Jindo command to upload metadata, a batch number that is generated based on the upload time is used as the root directory to store the metadata. This ensures that the data you uploaded each time is not overwritten. You can delete historical data based on your requirements.

- 1 indicates the configuration path of system information in OSS.
- 2 indicates a namespace.
- 3 indicates a batch number.

Metadata schema

HDFS metadata that is uploaded to OSS is stored as a file in the JSON format. Schema information:

```
{
  "type":"string",          /*Inode type, FILE DIRECTORY*/
  "id": "string",            /*INode id*/
  "parentId" :"string",         /*Parent node ID*/
  "name":"string",         /*Inode name*/
  "size": "int",         /*Inode size, BIGINT*/
  "permission":"int",         /*Permission, INT*/
  "owner":"string",          /*Owner name*/
  "ownerGroup":"string",     /*Owner group name*/
  "mtime":"int",              /*Inode modification time, BIGINT*/
  "atime":"int",              /*The time when the inode was last accessed, BIGINT*/
  "attributes":"string",       /*File-related attributes*/
  "state":"string",            /*Inode status*/
  "storagePolicy":"string",    /*Storage policy*/
  "etag":"string"           /*etag*/
}
```

Use Jindo SQL to analyze metadata

1. Run the command shown in the following figure to start Jindo SQL.

```
[root@emr-header-1 ~]# jindo sql
Spark master: yarn, Application Id: application_1603081647416_0050
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.33 seconds, Fetched 3 row(s)
```

2. Query the tables whose data can be analyzed by using Jindo SQL.
   - Run the `show tables` command to view the tables whose data can be analyzed. Jindo SQL has two built-in modules audit_log and fs_image for audit and metadata analysis.
   - Run the `show partitions fs_image` command to view the partition information of the fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

     The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

3. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287249    1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       1546468482017665002     1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The namespace and datetime columns are added and indicate the namespace name and the timestamp when metadata is uploaded.

Example: Obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

Use Hive to analyze metadata

1. Create a table schema in Hive.

   You can use the following command to create a table schema of the metadata that is used for queries in Hive:

```
CREATE EXTERNAL TABLE `table_name`
(`type` string,
 `id` string,
 `parentId` string,
 `name` string,
 `size` bigint,
 `permission` int,
 `owner` string,
 `ownerGroup` string,
 `mtime` bigint,
 `atime` bigint,
 `attr` string,
 `state` string,
 `storagePolicy` string,
 `etag` string)
 ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
 STORED AS TEXTFILE
 LOCATION 'The OSS path to which file data is uploaded';
```

2. Use Hive to analyze data offline.

   After you create a Hive table, you can use Hive SQL to analyze metadata.

```
select * from table_name limit 200;
```

   The following figure shows an example.

```
hive> select * from inode_metadata_test0 limit 100;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_202009[redacted]
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1[redacted], Tracking URL = http://emr-heade[redacted]:20888/proxy/applica[redacted]
Kill Command = /usr/lib/hadoop-current/bin/hadoop job  -kill job_1599[redacted]
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-09-08 14:57:26,112 Stage-1 map = 0%,  reduce = 0%
2020-09-08 14:57:31,263 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.22 sec
MapReduce Total cumulative CPU time: 1 seconds 220 msec
Ended Job = job_[redacted]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.22 sec   HDFS Read: 6867 HDFS Write: 1524 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 220 msec
OK
Directory       1127433438684721971 4    1127433438684721971 3    /uttest/oss     0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Directory       1127433438684721971 9    1127433438684721971 3    /uttest/oss2    0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
Directory       1127433438684721971 6    1127433438684721971 4    /uttest/oss/dir 0       511     caojie  staff   1599545017636   1599545017636           Finalized       WARM
File    1127433438684721971 5    1127433438684721971 4    /uttest/oss/file1       0       420     caojie  staff   1599545017632   1599545017632           Finalized       WARM
File    1127433438684721971 7    1127433438684721971 6    /uttest/oss/dir/file2   0       420     caojie  staff   1599545017642   1599545017642           Finalized       WARM
File    1127433438684721971 8    1127433438684721971 6    /uttest/oss/dir/file3   0       420     caojie  staff   1599545017651   1599545017651           Finalized       WARM
Directory       1127433438684721972 0    1127433438684721971 9    /uttest/oss2/dir        0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
File    1127433438684721972 1    1127433438684721972 0    /uttest/oss2/dir/file2  0       420     caojie  staff   1599545017658   1599545017658           Finalized       WARM
File    1127433438684721972 2    1127433438684721972 0    /uttest/oss2/dir/file3  0       420     caojie  staff   1599545017666   1599545017666           Finalized       WARM
Directory       1127433438684721971 3    1767282355743385190 5    /uttest 0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Time taken: 10.734 seconds, Fetched: 10 row(s)
hive>
```

3.2.9. Use a JindoFS credential provider

The data of JindoFS is stored in Object Storage Service (OSS). If you want to access the data of JindoFS, you must provide an AccessKey pair that can be used to access OSS. SmartData V3.4.0 and later support JindoFS credential providers. You can use a JindoFS credential provider to save an encrypted AccessKey pair into a file. This prevents the AccessKey pair from being leaked.

Configure a JindoFS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **smartdata-site** tab.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **smartdata-site** tab.
3. Add configuration information.
   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

Parameter	Description
fs.jfs.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

   iii. Click **OK**.
4. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Credential provider types

- TemporaryAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a validity period and a security token with a validity period are used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this credential provider, you must configure the parameters described in the following table in environment variables.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**.
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- JindoCommonCredentialsProvider

This is a common credential provider. After you configure the parameters described in the following table, you can use this credential provider in both JindoOSS and JindoFS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.JindoCommonCredentialsProvider**.
jindo.common.accessKeyId	The AccessKey ID used to access OSS.
jindo.common.accessKeySecret	The AccessKey secret used to access OSS.
jindo.common.securityToken	The temporary security token used to access OSS.

- EcsStsCredentialsProvider

  This credential provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EcsStsCredentialsProvider**.

3.2.10. Use JindoFS in block storage mode to store encrypted files

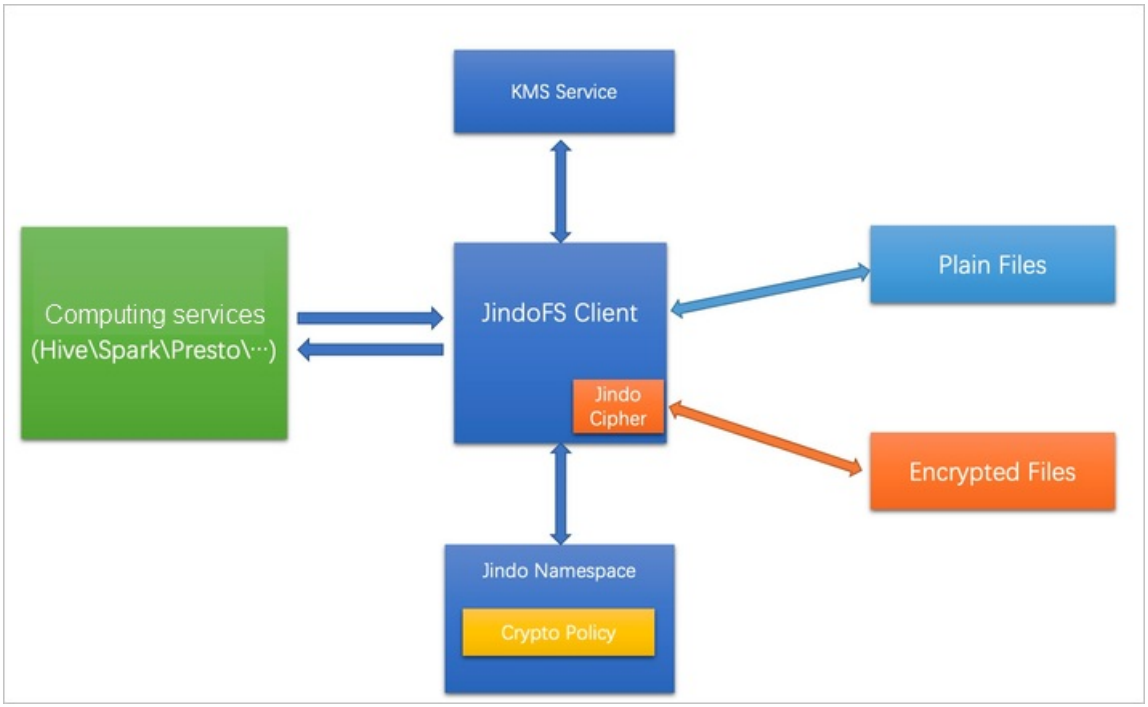
JindoFS in block storage mode supports file encryption. The encryption mechanism and usage method are similar to those of encryption zones for Apache HDFS. Keys are managed by Key Management Service (KMS). You can configure encryption policies for a directory that contains sensitive data. This way, you can transparently encrypt the data that you want to write to the directory and transparently decrypt the data that you want to read from the directory. In this process, you do not need to change your code.

Prerequisites

- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.
- KMS is activated. For more information, see Activate KMS.

Context

The following figure shows the architecture of JindoFS in block storage mode with file encryption supported.



Configure KMS-related parameters for JindoFS

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **namespace** tab.
3. Add configuration information.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description
crypto.provider.type	The type of the provider. Set this parameter to **ALIYUN**.
crypto.provider.endpoint	The public endpoint of KMS. For more information, see Make API requests.
crypto.provider.kms.accessKeyId	The AccessKey ID used to access KMS.
crypto.provider.kms.accessKeySecret	The AccessKey secret used to access KMS.

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Use JindoFS KeyProvider

JindoFS KeyProvider is used to connect JindoFS to KMS. The keys for JindoFS are stored in KMS. You can use JindoFS KeyProvider to create, query, and rotate keys based on KMS.

- Create a key: You can run the following command to create a key:

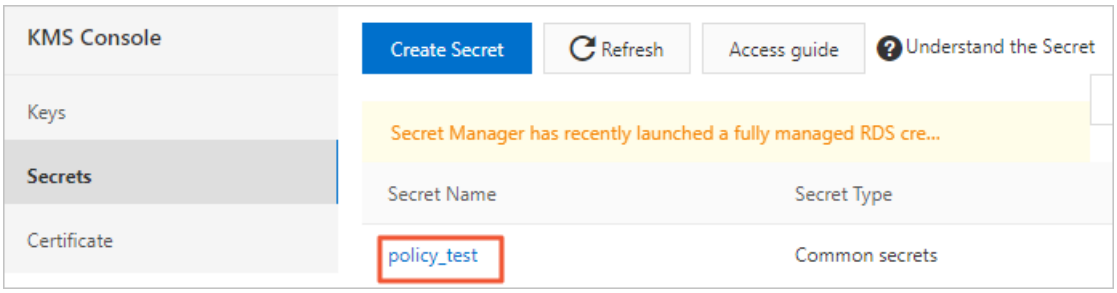
```
jindo key -create -keyIdName <keyIdName>
```

> **Note** In this topic, <keyIdName> in commands indicates the name of the key that you created.

For example, run the following command to create a key named *policy_test*:

```
jindo key -create -keyIdName policy_test
```

After the *policy_test* key is created, it is displayed on the Secrets page of the KMS console.



- Query keys: You can run the following command to query the names of all existing keys for JindoFS:

```
jindo key -list
```

The following information is returned:

```
Listing Keys:
        policy_test
        policy_test2
```

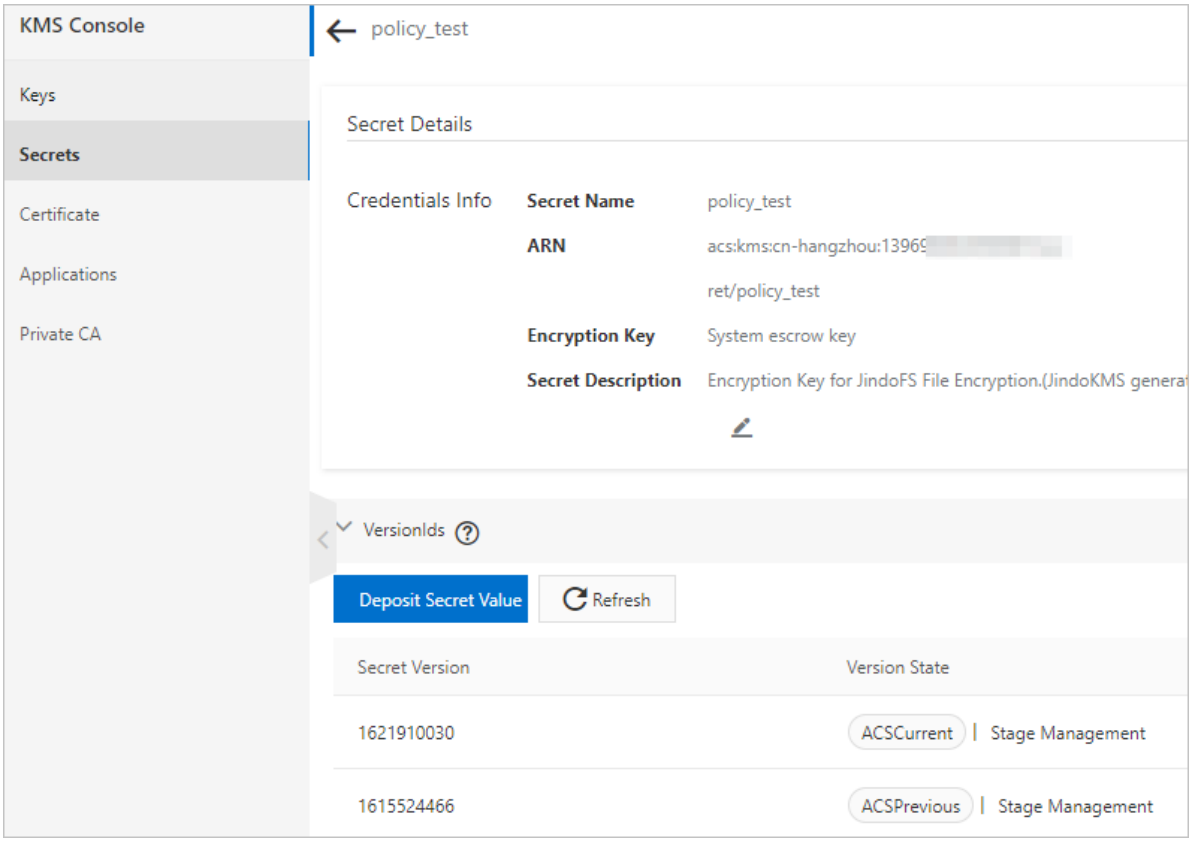
- Rotate a key: You can run the following command to periodically rotate a key based on its ID. After the key is rotated, the key version changes. If a file is encrypted after the rotation, the latest key version is used. If a file is decrypted after the rotation, the key version that corresponds to the file is used.

```
jindo key -roll -keyIdName <keyIdName>
```

For example, run the following command to rotate the *policy_test* key:

```
jindo key -roll -keyIdName policy_test
```

After the *policy_test* key is rotated, the version of the key is updated. The original version is in the **ACSPrevious** state and the new version is in the **ACSCurrent** state.



Manage encryption policies for JindoFS

You can run the following commands to configure and query encryption policies.

- Configure an encryption policy:

```
jindo jfs -setCryptoPolicy -keyIdName <keyIdName> <path>
```

> **Note** `<path>` in the preceding example indicates the path of a file on JindoFS, for example, *jfs://test/*.

- Query an encryption policy:

```
jindo jfs -getCryptoPolicy <path>
```

Example:

1. Query the encryption policy of the *jfs://test/* path:

```
jindo jfs -getCryptoPolicy jfs://test/
```

`{NONE}` is returned.

2. Configure an encryption policy for the *jfs://test/* path:

```
jindo jfs -setCryptoPolicy -keyIdName policy_test jfs://test/
```

3. Go to the *bigboot* directory and query the encryption policy of the *jfs://test/* path again:

```
jindo jfs -getCryptoPolicy jfs://test/
```

The following information is returned:

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/b2jindosdk/3.4.0-hadoop3.1/pack
age/b2jindosdk-3.4.0-hadoop3.1/lib/jindo-distcp-3.4.0.jar!/org/slf4j/impl/StaticLoggerB
inder.class]
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/hadoop/3.2.1-1.0.1/package/hado
op-3.2.1-1.0.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticL
oggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
21/03/12 13:52:34 WARN: Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable
21/03/12 13:52:35 INFO: Jboot log name is /var/log/bigboot/jboot-20210312-135234-12953.
LOG
21/03/12 13:52:35 INFO: Write buffer size 1048576, logic block size 134217728
21/03/12 13:52:35 INFO: cmd=getFileStatus, src=jfs://test/, dst=null, size=0, parameter
=null, time-in-ms=7, version=3.4.0
21/03/12 13:52:35 INFO: cmd=getCryptoPolicy, src=jfs://test/, dst=null, size=0, paramet
er=, time-in-ms=2, version=3.4.0
The crypto policy of path: jfs://test/ is {cipherSuite: AES_CTR_NOPADDING_256, keyIdNam
e: policy_test2, keyIdVersion: null, edek: , iv: }
21/03/12 13:52:35 INFO: Read total statistics: oss read average <none>, cache read aver
age <none>, read oss percent <none>
```

After you configure the encryption policy, you can read data from or write data to the files under this path.

- Copy a local file to HDFS:

```
hadoop fs -put test.log jfs://test/
```

- Display the content of a file:

```
hadoop fs -cat jfs://test/test.log
```

3.3. JindoFS in cache mode

3.3.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in Object Storage Service (OSS) based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

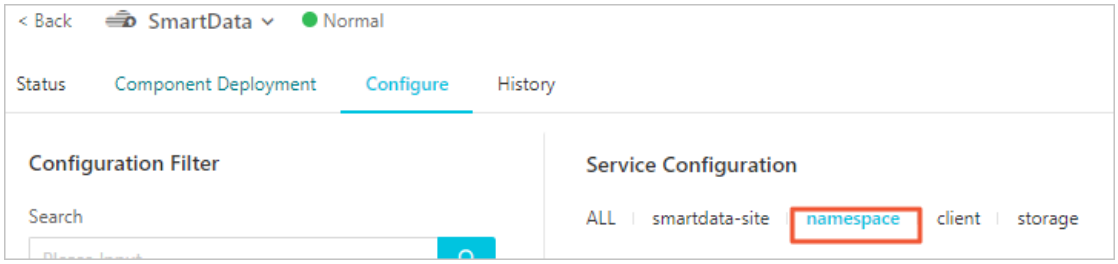
  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. Click **OK**.
5. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
6. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **client** tab.
2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   You do not need to restart the SmartData service to apply the new configurations.
3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

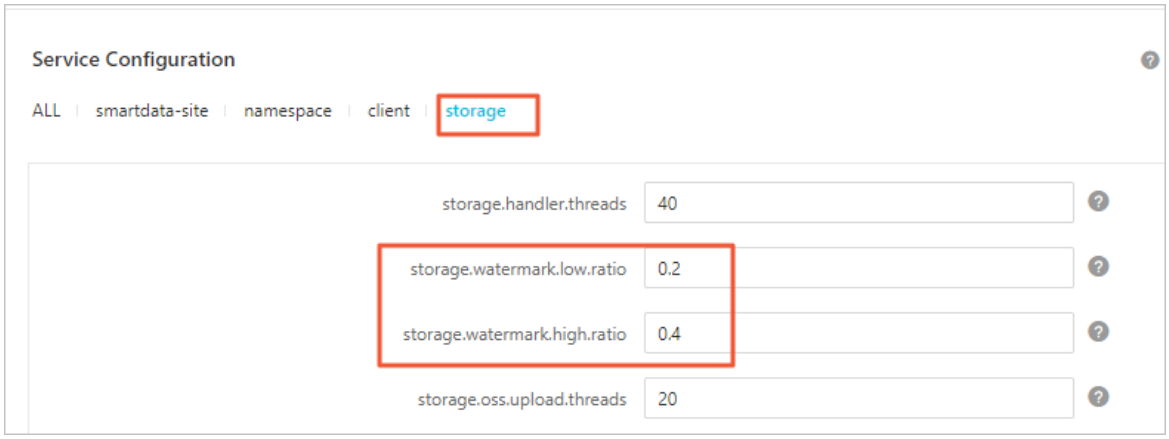
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you specify. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to specify an AccessKey pair. In other cases, you must specify an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you choose for accessing files in OSS:

- OSS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.
  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss.accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.endpoint	The endpoint of the OSS bucket that serves as the storage backend.

> **Note** You can set these parameters to the values of the parameters in an EMR version earlier than EMR V3.30.0.

- JFS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **namespace** tab.
  ii. Set **jfs.namespaces** to **test**.
  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize the cache performance. After you configure the parameters, you do not need to restart the SmartData service to apply the new configurations.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. You can use this parameter to prevent upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If an excessive amount of memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

3.3.2. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

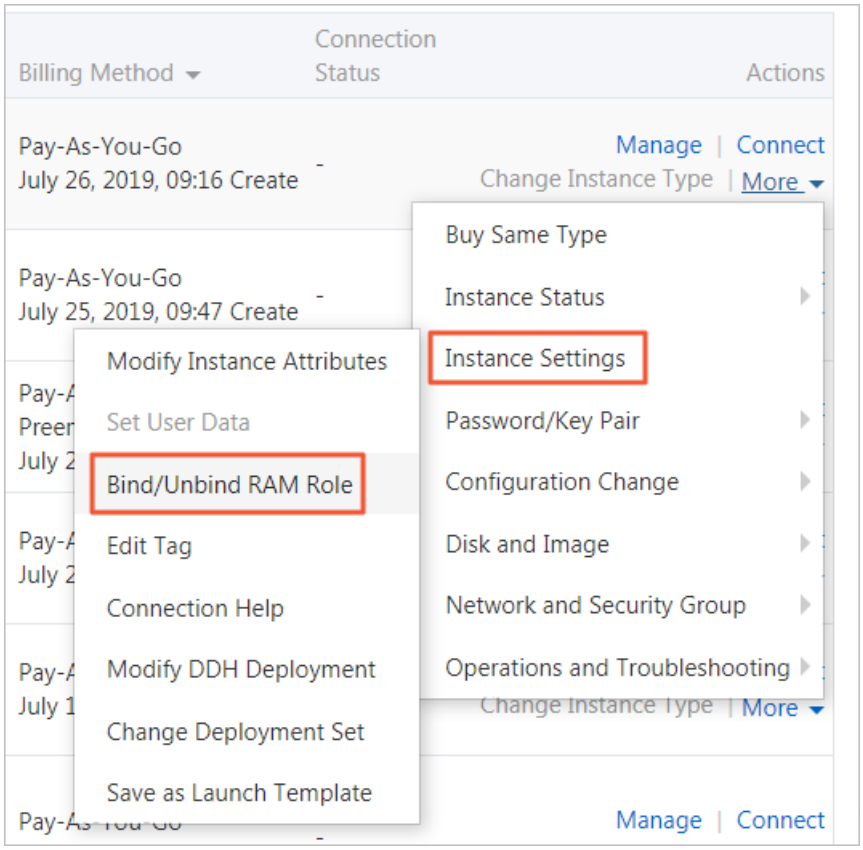
1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.
4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

   ```
   hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
   ```

2. Access OSS by using the FileSystem interface of Hadoop.

   Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

3.3.3. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false

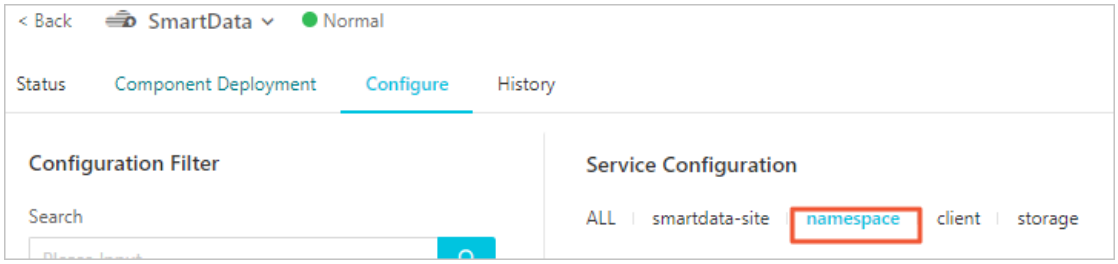
Parameter	Description
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-

right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

3.3.4. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

> **Notice**
>
> - The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
> - After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
> - Hive and Presto jobs do not use Hadoop Job Committer.
> - In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.

   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.
6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

   The default value is 8.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

3.3.5. Use JindoFS OSS credential providers

SmartData 3.4.0 and later support JindoFS OSS credential providers. You can use a JindoFS OSS credential provider to save an encrypted AccessKey pair in a file. This prevents the AccessKey pair from being leaked.

Limits

JindoRangerCredentialsProvider and AssumeRoleStsCredentialsProvider are available only in SmartData 3.8.0 and later.

Configure a JindoFS OSS credential provider

1. Go to the SmartData service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.

iii. Click the **Cluster Management** tab.

iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.

v. In the left-side navigation pane of the page that appears, choose **Cluster Service > SmartData**.

vi. Click the **Configure** tab.

2. Add or modify configurations based on the configuration method.

Configuration method	Description
Global configuration (for all buckets)	In the Configuration Filter section, search for the **fs.jfs.cache.oss.credentials.provider** parameter and append the required implementation class of AliyunCredentialsProvider to the end of the parameter value. If you want to add multiple classes, separate them with commas (,). The system reads credential values in sequence until a valid credential value is found. For more information about the configuration of this parameter, see Global configuration. Example: com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider, com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider, com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider.

Configuration method	Description
Bucket-level configuration	Perform the following steps to add configurations: i. In the upper-right corner of the **smartdata-site** tab, click Custom Configuration. **Notice** If you use JindoRangerCredentialsProvider, you must add configurations on the **namespace** tab. ii. In the **Add Configuration Item** dialog box, set **Key** to fs.jfs.cache.oss.bucket.XXX.credentials.provider and Value to the required implementation class of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. If you want to add multiple classes, separate them with commas (,). The system reads credential values in sequence until a valid credential value is found. For more information about the configuration of this parameter, see Bucket-level configuration. Example: com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider, com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider, com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider. **Note** XXX in fs.jfs.cache.oss.bucket.XXX.credentials.provider indicates the name of an OSS bucket. iii. Click **OK**.

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Global configuration

You can configure different credential providers based on your business requirements. The following table describes the supported credential providers.

Type	Description
TemporaryAliyunCredentialsProvider	This credential provider is suitable for scenarios in which an AccessKey pair with a validity period and a security token with a validity period are used to access OSS. To use this credential provider, you must append com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider to the value of the fs.jfs.cache.oss.credentials.provider parameter and add the following configuration items on the **smartdata-site** tab: • fs.jfs.cache.oss.accessKeyId: the AccessKey ID that is used to access your OSS bucket. • fs.jfs.cache.oss.accessKeySecret: the AccessKey secret that is used to access your OSS bucket. • fs.jfs.cache.oss.securityToken: the temporary security token that is used to access your OSS bucket.
SimpleAliyunCredentialsProvider	This credential provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS. To use this credential provider, you must append com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider to the value of the fs.jfs.cache.oss.credentials.provider parameter and add the following configuration items on the **smartdata-site** tab: • fs.jfs.cache.oss.accessKeyId: the AccessKey ID that is used to access your OSS bucket. • fs.jfs.cache.oss.accessKeySecret: the AccessKey secret that is used to access your OSS bucket.
EnvironmentVariableCredentialsProvider	To use this credential provider, you must configure the following parameters in the environment variable file: • fs.jfs.cache.oss.credentials.provider: Set this parameter to com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider. • ALIYUN_ACCESS_KEY_ID: the AccessKey ID that is used to access your OSS bucket. • ALIYUN_ACCESS_KEY_SECRET: the AccessKey secret that is used to access your OSS bucket. • ALIYUN_SECURITY_TOKEN: the temporary security token that is used to access your OSS bucket. **Note** This parameter is required only if you configure a token that has a validity period.

Type	Description
JindoCommonCredentialsProvider	This is a common credential provider. After you configure the required parameters, you can use this credential provider in JindoOSS and JindoFS. To use this credential provider, you must append com.aliyun.emr.fs.auth.JindoCommonCredentialsProvider to the value of the fs.jfs.cache.oss.credentials.provider parameter and add the following configuration items on the **smartdata-site** tab: • jindo.common.accessKeyId: the AccessKey ID that is used to access your OSS bucket. • jindo.common.accessKeySecret: the AccessKey secret that is used to access your OSS bucket. • jindo.common.securityToken: the temporary security token that is used to access your OSS bucket.
EcsStsCredentialsProvider	This credential provider does not require an AccessKey pair. You can access OSS in password-free mode. To use this credential provider, you must append com.aliyun.emr.fs.auth.EcsStsCredentialsProvider to the value of the fs.jfs.cache.oss.credentials.provider parameter.
JindoRangerCredentialsProvider	This credential provider is suitable for scenarios in which you configure parameters in Ranger to control access to OSS. To use this credential provider, you must append com.aliyun.emr.fs.auth.JindoRangerCredentialsProvider to the value of the fs.jfs.cache.oss.credentials.provider parameter and add a custom parameter whose name is namespace.oss.permission.method and value is ranger on the **namespace** tab. **Note** You must restart Jindo Namespace Service after you add the custom parameter. For more information, see Restart Jindo Namespace Service.

Type	Description
AssumeRoleStsCredentialsProvider	This credential provider is suitable for scenarios in which a temporary AccessKey pair that assumes a RAM role is obtained to access OSS. To use this credential provider, you must append com.aliyun.emr.fs.auth.AssumeRoleStsCredentialsProvider to the value of the fs.jfs.cache.oss.credentials.provider parameter and add the following configuration items on the **smartdata-site** tab: • assume.role.sts.accessKeyId: the AccessKey ID that is obtained from Security Token Service (STS). • assume.role.sts.accessKeySecret: the AccessKey secret that is obtained from STS. • assume.role.sts.endpoint: the endpoint of STS. For more information, see Endpoints. • assume.role.roleArn: the Alibaba Cloud Resource Name (ARN) of the RAM role. The ARN is specified in the format of acs:ram::$accountID:role/$roleName. For information about how to view an ARN, see How do I view the ARN of a RAM role?. • assume.role.roleSessionName: the name of the role session. This is a custom configuration item. You can set this configuration item based on your requirements. For example, you can set this configuration item to a username.

Bucket-level configuration

You can configure different credential providers based on your business requirements. The following table describes the supported credential providers.

Type	Description
TemporaryAliyunCredentialsProvider	This credential provider is suitable for scenarios in which an AccessKey pair with a validity period and a security token with a validity period are used to access OSS. To use this credential provider, you must set the fs.jfs.cache.oss.bucket.XXX.credentials.provider parameter to com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider and add the following configuration items on the **smartdata-site** tab: • fs.jfs.cache.oss.bucket.XXX.accessKeyId: the AccessKey ID that is used to access your OSS bucket. • fs.jfs.cache.oss.bucket.XXX.accessKeySecret: the AccessKey secret that is used to access your OSS bucket. • fs.jfs.cache.oss.bucket.XXX.securityToken: the temporary security token that is used to access your OSS bucket.

Type	Description
SimpleAliyunCredentialsProvider	This credential provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS. To use this credential provider, you must set the fs.jfs.cache.oss.bucket.XXX.credentials.provider parameter to com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider and add the following configuration items on the **smartdata-site** tab: • fs.jfs.cache.oss.bucket.XXX.accessKeyId: the AccessKey ID that is used to access your OSS bucket. • fs.jfs.cache.oss.bucket.XXX.accessKeySecret: the AccessKey secret that is used to access your OSS bucket.
EnvironmentVariableCredentialsProvider	To use this credential provider, you must configure the following parameters in the environment variable file: • fs.jfs.cache.oss.bucket.XXX.credentials.provider: Set this parameter to com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider. • ALIYUN_ACCESS_KEY_ID: the AccessKey ID that is used to access your OSS bucket. • ALIYUN_ACCESS_KEY_SECRET: the AccessKey secret that is used to access your OSS bucket. • ALIYUN_SECURITY_TOKEN: the temporary security token that is used to access your OSS bucket. **Note** This parameter is required only if you configure a token that has a validity period.
JindoCommonCredentialsProvider	This is a common credential provider. After you configure the required parameters, you can use this credential provider in JindoOSS and JindoFS. To use this credential provider, you must set the fs.jfs.cache.oss.bucket.XXX.credentials.provider parameter to com.aliyun.emr.fs.auth.JindoCommonCredentialsProvider and add the following configuration items on the **smartdata-site** tab: • jindo.common.accessKeyId: the AccessKey ID that is used to access your OSS bucket. • jindo.common.accessKeySecret: the AccessKey secret that is used to access your OSS bucket. • jindo.common.securityToken: the temporary security token that is used to access your OSS bucket.
EcsStsCredentialsProvider	This credential provider does not require an AccessKey pair. You can access OSS in password-free mode. To use this credential provider, you must set the fs.jfs.cache.oss.bucket.XXX.credentials.provider parameter to com.aliyun.emr.fs.auth.EcsStsCredentialsProvider.

Type	Description
JindoRangerCredentialsProvider	This credential provider is suitable for scenarios in which you configure parameters in Ranger to control access to OSS. To use this credential provider, you must append com.aliyun.emr.fs.auth.JindoRangerCredentialsProvider to the value of the fs.jfs.cache.oss.bucket.XXX.credentials.provider parameter and add a custom parameter whose name is namespace.oss.permission.method and value is ranger on the **namespace** tab. **Note** You must restart Jindo Namespace Service after you add the custom parameter. For more information, see Restart Jindo Namespace Service.
AssumeRoleStsCredentialsProvider	This credential provider is suitable for scenarios in which a temporary AccessKey pair that assumes a RAM role is obtained to access OSS. To use this credential provider, you must append com.aliyun.emr.fs.auth.AssumeRoleStsCredentialsProvider to the value of the fs.jfs.cache.oss.bucket.XXX.credentials.provider parameter and add the following configuration items on the **smartdata-site** tab: • assume.role.sts.accessKeyId: the AccessKey ID that is obtained from STS. • assume.role.sts.accessKeySecret: the AccessKey secret that is obtained from STS. • assume.role.sts.endpoint: the endpoint of STS. For more information, see Endpoints. • assume.role.roleArn: the ARN of the RAM role. The ARN is specified in the format of acs:ram::$accountID:role/$roleName. For information about how to view an ARN, see How do I view the ARN of a RAM role?. • assume.role.roleSessionName: the name of the role session. This is a custom configuration item. You can set this configuration item based on your requirements. For example, you can set this configuration item to a username.

Restart Jindo Namespace Service

If you use JindoRangerCredentialsProvider, you must restart Jindo Namespace Service after you add the required configuration items.

1. In the upper-right corner of the Configure tab on the SmartData service page, choose **Actions > Restart Jindo Namespace Service**.
2. In the **Cluster Activities** dialog box, specify Description and click **OK**.
3. In the **Confirm** message, click **OK**.

3.3.6. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

  Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.

  Namespace Info (1)

  Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.

StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.

Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

3.3.7. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

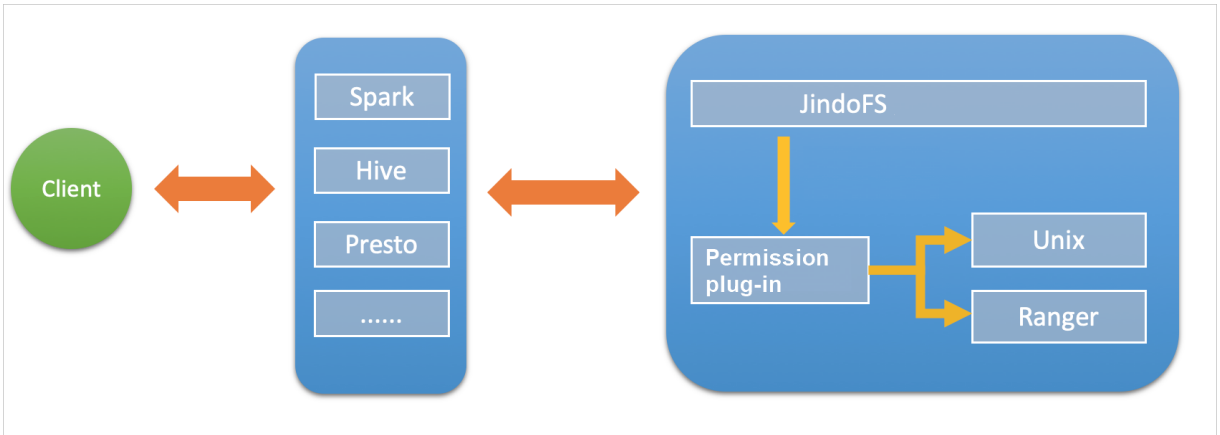
Permission management based on the storage mode:

- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.

- Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.

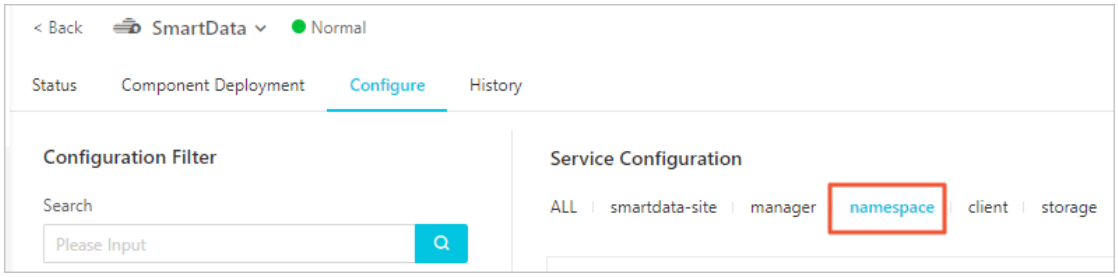
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

   After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

   If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

3.4. JindoTable

3.4.1. Enable query acceleration based on a native engine

JindoTable provides a native engine for you to accelerate the speed of using Spark, Hive, or Presto to query files in the ORC or Parquet format. This topic describes how to enable query acceleration based on the native engine to improve the performance of Spark, Hive, and Presto.

Prerequisites

An E-MapReduce (EMR) cluster is created, and a file in the ORC or Parquet format is stored in JindoFS or Object Storage Service (OSS). For more information about how to create a cluster, see Create a cluster.

Limits

- Binary files are not supported.

- Partitioned tables whose values of partition key columns are stored in files are not supported.
- EMR clusters of V5.X.X or later are not supported.
- spark.read.schema (userDefinedSchema) is not supported.
- Data of the DATE type must be in the YYYY-MM-DD format and range from 1400-01-01 to 9999-12-31.
- If a table contains two columns that have the same column name with different letter cases, such as ID and id, queries on the columns cannot be accelerated.
- The following table lists the supported Spark, Hive, and Presto engines and the file formats supported by each engine.

Engine	ORC	Parquet
Spark2	Supported	Supported
Spark3	Supported	Supported
Presto	Supported	Supported
Hive2	Not supported	Supported
Hive3	Not supported	Supported

- The following table lists the supported Spark, Hive, and Presto engines and the file systems supported by each engine.

Engine	OSS	JindoFS	HDFS
Spark2	Supported	Supported	Supported
Presto	Supported	Supported	Supported
Hive2	Supported	Supported	Not supported
Hive3	Supported	Supported	Not supported

Improve the performance of Spark

1. Enable query acceleration for ORC or Parquet files in JindoTable.

> **Note**
>
> - Query acceleration consumes off-heap memory. We recommend that you add `--conf spark.executor.memoryOverhead=4g` to a Spark task to apply for additional resources for query acceleration.
> - If you use Spark to read data from an ORC or Parquet file, the DataFrame API or Spark SQL is required.

   - Global configuration

a. Go to the Cluster Overview page of your cluster.
   a. Log on to the Alibaba Cloud EMR console.
   b. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   c. Click the **Cluster Management** tab.
   d. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
b. Modify the related parameter.
   a. In the left-side navigation pane, choose **Cluster Service > Spark**.
   b. On the Spark service page, click the **Configure** tab.
   c. Search for the spark.sql.extensions parameter and change its value to io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sql.JindoTableExtension.
c. Save the configuration.
   a. Click **Save** in the upper-right corner.
   b. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
d. Restart ThriftServer.
   a. Choose **Actions > Restart ThriftServer** in the upper-right corner.
   b. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   c. In the **Confirm** message, click **OK**.

- Job-level configuration

  When you configure a Spark Shell or Spark SQL job, add the following configuration to the code:

  ```
  --conf spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sq
  l.JindoTableExtension
  ```

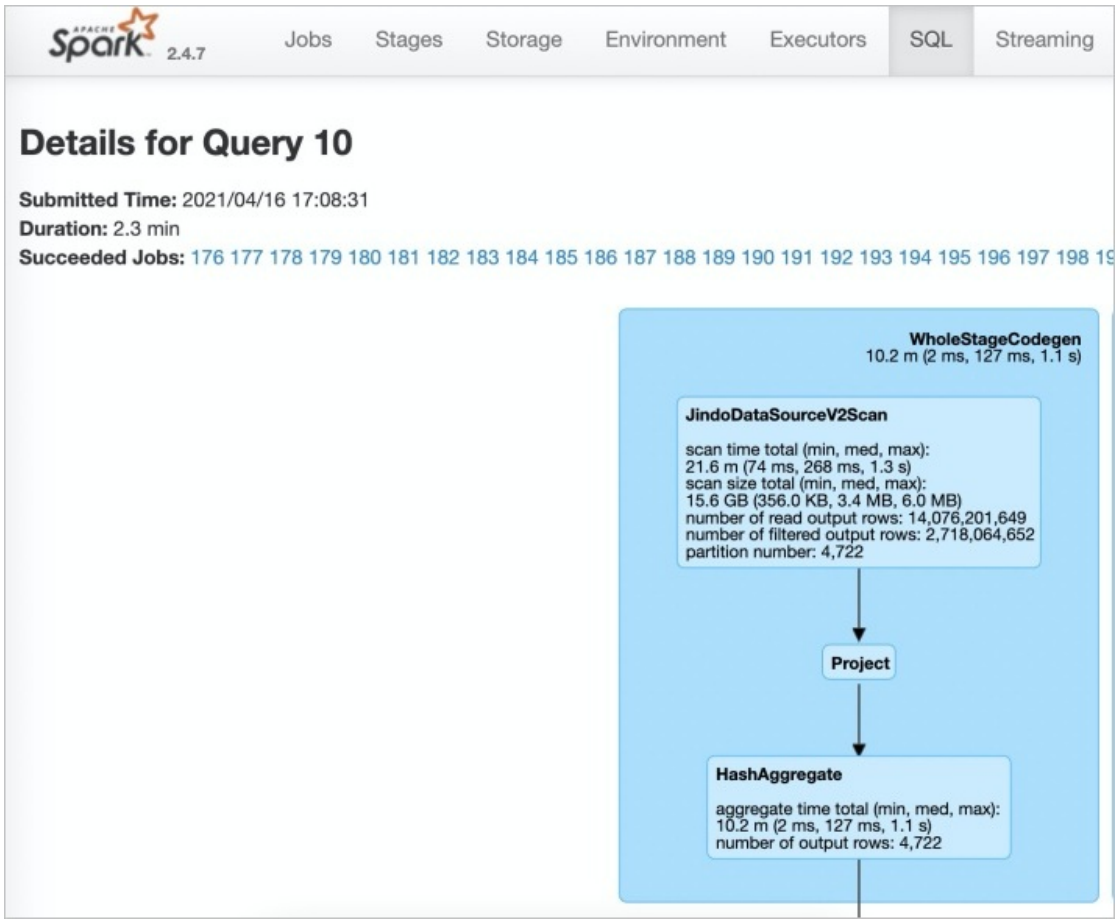
  For more information about how to configure a Spark job, see Configure a Spark Shell job or Configure a Spark SQL job.

2. Check whether query acceleration is enabled.
   i. Access the web UI of Spark History Server.

      For more information, see Access the web UIs of open source components.

ii. On the SQL tab of Spark, view information about the related task.

If JindoDataSourceV2Scan appears, query acceleration is enabled. Otherwise, check the configurations in Step 1.



Improve the performance of Presto

> **Notice** Presto has high query concurrency and uses off-heap memory for query acceleration. To use the query acceleration feature, make sure that the memory is greater than 10 GB.

By default, `catalog: hive-acc` of the native engine is built in the Presto service. You can use `catalog: hive-acc` to enable query acceleration.

Example:

```
presto --server https://emr-header-1.cluster-xxx:7778/ --catalog hive-acc --schema default
```

`emr-header-1.cluster-xxx` indicates the hostname of the emr-header-1 node.

> **Note** When you use this feature in Presto, complex data types such as MAP, STRUCT, and ARRAY are not supported.

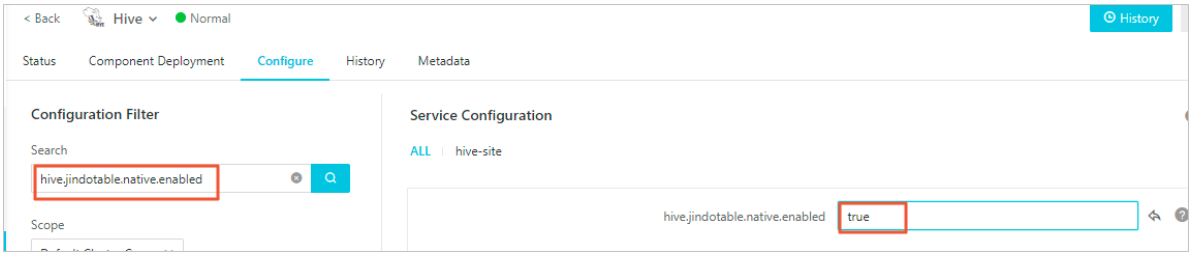
Improve the performance of Hive

> **Notice** If you have high requirements for job stability, we recommend that you do not enable query acceleration.

You can use one of the following methods to improve the performance of Hive:

- Use the EMR console

  On the **Configure** tab of the Hive service page, search for the **hive.jindotable.native.enabled** parameter and change its value to **true**. Then, save the configuration and restart the Hive service. This method is suitable for Hive on MapReduce and Hive on Tez jobs.



- Use the CLI

  Set `hive.jindotable.native.enabled` to `true` in the CLI to enable query acceleration. By default, the query acceleration plug-in for Parquet files is deployed in JindoTable in EMR V3.35.0 and later. You can directly set this parameter in EMR V3.35.0 and later.

```
set hive.jindotable.native.enabled=true;
```

> **Note** When you use this feature in Hive, complex data types such as MAP, STRUCT, and ARRAY are not supported.

3.4.2. Use JindoTable

JindoTable is used to implement tiered storage, optimize table files, and collect data statistics based on the popularity of tables or partitions. This topic describes how to use JindoTable.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your on-premises machine.
- An EMR cluster of the 3.30.0 version or later is created. For more information about how to create a cluster, see Create a cluster.

Use JindoTable

Common commands:

- -accessStat
- -cache
- -archive
- -unarchive
- -uncache
- -status
- -optimize

- -showTable
- -showPartition
- -listTables
- -dumpmc

> **Notice** Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

-accessStat

- Syntax

  jindo table -accessStat {-d} *<days>* **{-n}** <topNums>

- Description

  This command is used to query the access records in which tables or partitions are visited most in a specified time range.

  <days> and <topNums> must be positive integers. If <days> is 1, all the access records generated from 00:00 (local time) on the current day to the current time are queried.

- Example: Query the 20 access records of tables or partitions that are most frequently visited within the last seven days.

```
jindo table -accessStat -d 7 -n 20
```

-cache

- Syntax

  jindo table -cache {-t} <dbName.tableName> **[-p]** <partitionSpec> **[-pin]**

- Description

  This command is used to cache data of specified tables or partitions to local disks.

  The data of the tables or partitions must be stored in Object Storage Service (OSS) or JindoFileSystem (JindoFS). Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`. When you specify `-pin`, if the cache space is insufficient, do not delete related data if possible.

- Example: Cache data of the db1.t1 table that is generated on March 16, 2020 to local disks.

```
jindo table -cache -t db1.t1 -p date=2020-03-16
```

-uncache

- Syntax

  jindo table -uncache {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to delete cached data of specified tables or partitions from local disks.

  The data of the tables or partitions must be stored in OSS or JindoFS. Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

- Examples:
  - Delete cached data of the db1.t2 table from local disks.

    ```
    jindo table -uncache -t db1.t2
    ```

  - Delete cached data of the db1.t1 table from local disks.

    ```
    jindo table -uncache -t db1.t1 -p date=2020-03-16,category=1
    ```

-archive

- Syntax

  jindo table -archive {-a|i} {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to lower the level of the policy that is used to store data of specified tables or partitions. By default, the Archive storage class is used.

  To use the Infrequent Access (IA) storage class, add -i to the command. Specify tables in the format of database.table. Specify partitions in the format of 'partitionCol1=1,partitionCol2=2,...'.

- Example: Cache data of the db1.t1 table to local disks.

  ```
  jindo table -archive -t db1.t1 -p date=2020-10-12
  ```

-unarchive

- Syntax

  jindo table -unarchive [-o|-i] {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to change the storage class from Archive to Standard.

  If `-o` is added to the command, an Archived object is temporarily restored. If `-i` is added to the command, an Archived object is changed to an IA object.

- Example:

  ```
  jindo table -unarchive -o -t db1.t1 -p date=2020-03-16,category=1
  ```

  ```
  jindo table -unarchive -i -t db1.t2
  ```

-status

- Syntax

  jindo table -status {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to view the data storage status of specified tables or partitions.

- Examples:
  - View the data storage status of the db1.t2 table.

    ```
    jindo table -status -t db1.t2
    ```

- View the data storage status of the db1.t1 table on March 16, 2020.

```
jindo table -status -t db1.t1 -p date=2020-03-16
```

-optimize

- Syntax

  jindo table -optimize {-t} <dbName.tableName>

- Description

  This command is used to optimize the data organization of tables at the storage layer.

- Example: Optimize the data organization of the db1.t1 table at the storage layer.

```
jindo table -optimize -t db1.t1
```

-showTable

- Syntax

  jindo table -showTable {-t} <dbName.tableName>

- Description

  This command is used to display all partitions in a partitioned table or display the data storage of a non-partitioned table.

- Example: Display all partitions in the db1.t1 partitioned table.

```
jindo table -showTable -t db1.t1
```

-showPartition

- Syntax

  jindo table -showPartition {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to display the data storage of partitions.

- Example: Display the data storage of all partitions in the db1.t1 partitioned table on October 12, 2020.

```
jindo table -showPartition -t db1.t1 -p date=2020-10-12
```

-listTables

- Syntax

  jindo table -listTables [-db] <dbName>

- Description

  This command is used to display all tables in a specified database. If you do not specify `[-db]`, tables in the default database are displayed.

- Examples:

- Display tables in the default database.

```
jindo table -listTables
```

- Display tables in the db1 database.

```
jindo table -listTables -db db1
```

-dumpmc

- Syntax

jindo table -dumpmc {-i} <accessId> **{-k}** <accessKey> **{-m}** <numMaps> **{-t}** <tunnelUrl> **{-project}** <projectName> **{-table}** <tablename> **{-p}** <partitionSpec> **{-f}** <csv|tfrecord> **{-o}** <outputPath>

Parameter	Description	Required
-i	The AccessKey ID of your Alibaba Cloud account.	Yes
-k	The AccessKey secret of your Alibaba Cloud account.	Yes
-m	The number of map tasks.	Yes
-t	The VPC Tunnel endpoint of MaxCompute.	Yes
-project	The name of the MaxCompute project.	Yes
-table	The name of the MaxCompute table.	Yes
-p	The partition information. Example: `pt=xxx`. Separate multiple partitions with commas (,), such as `pt=xxx,dt=xxx`.	No
-f	The file format. Valid values: ○ tfrecord ○ csv	Yes
-o	The destination path.	Yes

- Description

  This command is used to dump MaxCompute tables to an EMR cluster or OSS. The formats CSV and TFRECORD are supported.

- Examples:

- Dump a MaxCompute table in the TFRECORD format to an EMR cluster.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o /tmp/outputtf1 -f tfrecord
```

- Dump a MaxCompute table in the CSV format to OSS.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o oss://bucket1/tmp/outputcsv -f csv
```

3.4.3. Archive and unarchive data in SDK mode

JindoTable allows you to run the archiveTable and unarchiveTable commands in SDK mode to archive and unarchive data in Object Storage Service (OSS). The commands do not rely on the Jindo Namespace Service component of SmartData. This topic describes how to use the archiveTable and unarchiveTable commands.

Prerequisites

- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.
- The partitioned table or non-partitioned table that you want to archive is stored in OSS. Only table data can be archived.

Context

You can use the original archive and unarchive commands of JindoTable to archive or unarchive tables or partitions in OSS. However, these commands rely on the Jindo Namespace Service component of SmartData. The new commands archiveTable and unarchiveTable do not rely on the Jindo Namespace Service component.

The archiveTable and unarchiveTable commands have the following advantages over the archive and unarchive commands:

- You can run the archiveTable and unarchiveTable commands even if the SmartData service is not deployed in your cluster. For example, you can run the commands on a self-managed cluster.
- You can configure filter parameters in the archiveTable or unarchiveTable command to archive or unarchive a large number of partitions on multiple threads at the same time. If local multithreading cannot meet your business requirements, you can run MapReduce tasks on the entire cluster to archive or unarchive data.

For more information about the original archive and unarchive commands, see Use JindoTable.

Limits

The archiveTable and unarchiveTable commands are supported only in EMR V3.36.0 and later minor versions, and EMR V5.2.0 and later minor versions.

archiveTable

You can use the archiveTable command to archive tables or partitions in OSS.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to obtain help information:

```
jindo table -help archiveTable
```

archiveTable syntax:

```
-archiveTable -t <dbName.tableName> \
-i/-a/-ca \
[-c "<condition>" | -fullTable] \
[-b/-before <before days>] \
[-p/-parallel <parallelism>] \
[-mr/-mapReduce] \
[-e/-explain] \
[-w/-workingDir <working directory>] \
[-l/-logDir <log directory>]
```

Parameter	Description	Required
-t <dbName.tableName>	The name of the table that you want to archive. You must configure this parameter in the `Database name.Table name` format. Separate the database name and table name with a period (.). The table can be a partitioned table or a non-partitioned table.	Yes
-i/-a/-ca	The storage class in which you want to archive data. You can use one of the following options to specify a storage class: ○ -i: Infrequent Access (IA) ○ -a: Archive ○ -ca: Cold Archive If you use the -i option in the command, the files whose storage class is Archive or Cold Archive are skipped. If you use the -a option in the command, the files whose storage class is Cold Archive are skipped.	Yes
-c "<condition>" | -fullTable	You must specify either `-fullTable` or `-c "<condition>"`. ○ If you specify `-fullTable`, the entire partitioned or non-partitioned table is archived. ○ If you specify `-c "<condition>"`, only the partitions that meet the filter condition are archived. Common operators, such as greater-than signs (>), are supported. For example, if the partition key column is the ds column whose data type is String and you want to archive partitions whose partition names are greater than 'd', use `-c " ds > 'd' "`.	Yes
-b/before <before days>	Only the tables or partitions that were created at least the specified days ago can be archived.	No
-p/-parallel <parallelism>	The parallelism of archiving operations.	No

Parameter	Description	Required
-mr/-mapReduce	Hadoop MapReduce instead of local multithreading is used to archive data.	No
-e/-explain	The explain mode is used. In explain mode, the list of partitions to be archived is displayed, but no data is archived.	No
-w/-workingDir	The working directory of a MapReduce job. This option is used only when you use a MapReduce job to archive data. You must have read and write permissions on the directory. The directory can be empty or not. Temporary files are created when you run the MapReduce job and are automatically deleted after the job is completed.	No
-l/-logDir <log directory>	The directory in which log files are stored.	No

unarchiveTable

The syntax of the unarchiveTable command is similar to the syntax of the archiveTable command. You can use the unarchiveTable command to unarchive tables or partitions in OSS.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to obtain help information:

   ```
   jindo table -help unarchiveTable
   ```

   unarchiveTable syntax:

   ```
   -archiveTable -t <dbName.tableName> \
   -i/-a/-o/-cr \
   [-notWait] \
   [-c "<condition>" | -fullTable] \
   [-b/-before <before days>] \
   [-p/-parallel <parallelism>] \
   [-mr/-mapReduce] \
   [-e/-explain] \
   [-w/-workingDir <working directory>] \
   [-l/-logDir <log directory>]
   ```

The parameters of the unarchiveTable command differ from the parameters of the archiveTable command in the following aspects:

- The optional parameter -i/-a/-o/-cr is used instead of the required parameter -i/-a/-ca.
- The optional parameter -notWait is used.

Parameter	Description	Required

Parameter	Description	Required
-i/-a/-o/-cr	The conversion of the storage class for data in the Cold Archive storage class. • If you do not specify -i/-a/-o/-cr, the storage class of the data is changed from Cold Archive to Standard. Note The storage class of the data can be changed from Cold Archive to Standard, IA, or Archive only when the data is completely unarchived. • If you specify -i/-a/-o/-cr, the following descriptions prevail: ○ If you specify the -i option, the storage class of the data is changed from Cold Archive to IA. Files whose storage class is Standard are skipped. ○ If you specify the -a option, the storage class of the data is changed from Cold Archive to Archive. Files whose storage class is Standard or IA are skipped. ○ If you specify the -o option, data is only temporarily unarchived and its storage class is retained. Files whose storage class is Standard or IA are skipped. Files that are previously unarchived are also skipped. This way, these files are not repeatedly unarchived. ○ You can specify the -cr option to check whether the files stored to Archive or Cold Archive storage are unarchived.	No
-notWait	This parameter is valid only when you unarchive data. This parameter is used to unarchive data that is stored to Cold Archive storage. If you specify this parameter, the system only sends an unarchive command, but does not wait for the completion of the unarchiving operation.	No

3.4.4. Use the MoveTo command of JindoTable

JindoTable allows you to run the MoveTo command to migrate data in tables or partitions. This topic describes how to use the MoveTo command.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

The MoveTo command can automatically update metadata after the command copies the underlying data. This way, data in a table or partitions can be fully migrated to the destination path. You can configure filter conditions for the MoveTo command to migrate a large number of partitions at the same time. JindoTable also provides some protective measures to ensure data integrity and security when the MoveTo command is used to migrate data.

Limits

The MoveTo command is supported only in EMR V3.36.0 and later minor versions, and in EMR V5.2.0 and later minor versions.

Use the MoveTo command

> **Notice** Only one MoveTo process can be run at a time in an EMR cluster. If you attempt to start a MoveTo process on an EMR cluster on which another MoveTo process is running, your request is rejected because the configuration lock is unavailable. A message that contains information about the running MoveTo process is also displayed. In this case, you can terminate the running MoveTo process and start a new MoveTo process, or wait for the running MoveTo process to end.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to obtain help information:

   ```
   jindo table -help moveTo
   ```

   Information similar to the following output is returned:

```
<dbName.tableName>      The table to move.
<destination path>      The destination base directory which is always at the
                          same level of a 'table location', where the moved
                          partitions or un-partitioned data would located in.
<condition>/-fullTable  A filter condition to determine which partitions should
                          be moved, supporting common operators (like '>') and
                          built-in UDFs (like to_date) (UDFs not supported
                          yet...), while -fullTable means that all partitions (or
                          a whole un-partitioned table) should be moved. One but
                          only one option must be specified among -c
                          "<condition>" and -fullTable.
<before days>           Optional, saying that table/partitions should be moved
                          only when they are created (not updated or modified)
                          more than some days before from now.
<parallelism>           The maximum concurrency when copying partitions, 1 by
                          default.
        <OSS storage policy>: Storage policy for OSS destination, which can be Standard
  (by default), IA, Archive, or ColdArchive. Not applicable for destinations other
  than OSS. NOTE: if you are willing to use ColdArchive storage policy, please
  make sure that Cold Archive has been enabled for your OSS bucket.
-o/-overWrite     Overwriting the final paths where the data would be moved.
                    For partitioned tables this overwrites partitions' locations
                    which are subdirectories of <destination path>; for
                    un-partitioned table this overwrites the <destination path>
                    itself.
-r/-removeSource  Let the source data be removed when the corresponding
                    table/partition is successfully moved to the new destination.
                    Otherwise (by default), the source data would be left as it
                    was.
-skipTrash        Applicable only when [-r/-removeSource] is enabled. If
                    present, source data would be immediately deleted from the
                    file system, bypassing the trash.
-e/-explain       If present, the command would not really move data, but only
                    prints the table/partitions that would be moved for given
                    conditions.
<log directory>   A directory to locate log files, '/tmp/<current user>/' by
                    default.
```

MoveTo syntax:

```
jindo table -moveTo \
  -t <dbName.tableName> \
  -d <destination path> \
  [-c "<condition>" | -fullTable] \
  [-b/-before <before days>] \
  [-p/-parallel <parallelism>] \
  [-s/-storagePolicy <OSS storage policy>] \
  [-o/-overWrite] \
  [-r/-removeSource] \
  [-skipTrash] \
  [-e/-explain] \
  [-l/-logDir <log directory>]
```

Parameter	Description	Required
-t <dbName.tableName>	The name of the table that you want to migrate. You must specify this parameter in the `Database name.Table name` format. Separate the database name and table name with a period (.). The table can be a partitioned table or a non-partitioned table.	Yes
-d <destination path>	The destination path. No matter whether you want to migrate a specific partition or an entire non-partitioned table, this parameter specifies a table-level path. If you want to migrate a partition, the complete path of the partition is composed of the value of this parameter and the name of the partition, such as `<destination path>/p1=v1/p2=v2/` .	Yes
-c "<condition>" | -fullTable	You must specify either `-c "<condition>"` or `-fullTable` . ○ If you specify `-fullTable` , the entire partitioned or non-partitioned table is archived. ○ If you specify `-c "<condition>"` , only the partitions that meet the filter condition are archived. Common operators, such as greater-than signs (>), are supported. For example, if the partition key column is the ds column whose data type is String and you want to archive partitions whose partition names are greater than 'd', use `-c " ds > 'd' "` .	No
-b/before <before days>	Only the tables or partitions that were created at least the specified days ago can be migrated.	No
-p/-parallel <parallelism>	The parallelism among migration operations.	No

Parameter	Description	Required
-s/-storagePolicy <OSS storage policy>	The storage class that you want to use after data is migrated to Object Storage Service (OSS). Valid values: ○ Standard ○ IA ○ Archive ○ ColdArchive **Note** Make sure that the storage class you want to use is enabled on the destination OSS bucket.	No
-o/-overWrite	The destination path is forcibly cleared. For a partitioned table, only the destination path of the partition that you want to migrate is cleared.	No
-r/-removeSource	After data is migrated and metadata is updated, the source path is cleared. For a partitioned table, only the source path of the partition that is migrated is cleared.	No
-skipTrash	The trash is skipped when the source path is cleared. **Note** You can specify this option only if -r/-removeSource is specified.	No
-e/-explain	The explain mode is used. In explain mode, the list of partitions to be migrated is displayed, but no data is migrated.	No
-l/-logDir <log directory>	The directory in which log files are stored.	No

Configure a lock directory

The MoveTo command supports process locks. You must use an HDFS path to store lock files. The default path is *hdfs:///tmp/jindotable-lock/*.

Notice The path must be an HDFS path. If you have no permission on the default path, you can perform the following steps to customize a path.

1. Go to the HDFS service page.
   i.

ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.

iii. Click the **Cluster Management** tab.

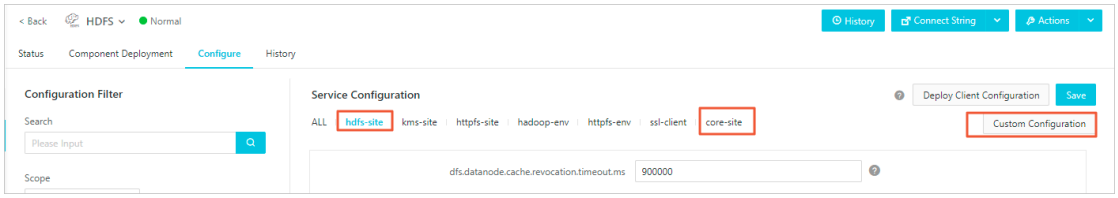
iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.

v. In the left-side navigation pane of the Cluster Overview page, choose **Cluster Service > HDFS**.

2. Add a custom item.

   i. Click the **Configure** tab. Then, click the **hdfs-site** or **core-site** tab in the Service Configuration section.

   ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**.



   iii. In the **Add Configuration Item** dialog box, add the **jindotable.moveto.tablelock.base.dir** parameter and set it to an existing HDFS path.

> **Notice** When you customize a lock directory, make sure that no MoveTo process is running on the nodes of the cluster. Otherwise, the MoveTo process may fail, which may even cause data pollution.

3. Save the configuration.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and click **OK**.

3.4.5. Use JindoTable to collect access frequency statistics on tables and partitions

You can use JindoTable to collect access frequency statistics on tables and partitions, and separate cold and hot data based on these statistics. This helps you reduce storage costs and improve cache usage efficiency.

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

JindoTable can be used to collect access frequency statistics on Hive tables. The collected data is saved in the namespaces of the SmartData service of a cluster.

In SmartData 3.2.X and later, JindoTable can be used to collect access frequency statistics on Spark, Hive, and Presto tables. This feature is enabled for Spark and Presto by default. If you want to disable the feature for Spark or Presto, perform the operations described in Disable the collection of access frequency statistics. The feature is disabled for Hive by default. If you want to enable the feature for Hive, perform the operations described in Enable the collection of access frequency statistics for Hive.

Query access frequency statistics

You can run a command to query access frequency statistics.

- Syntax

```
jindo table -accessStat <-d [days]> <-n [topNums]>
```

Set `days` and `topNums` to positive integers. For example, if days is set to 1 and topNums is not specified, the access frequency statistics of all the tables or partitions that are visited on the current day (from 00:00 to the current time) are queried.

- Description

This command is used to query the first N tables or partitions that are visited the most frequently within a specific period of time, and the number of times each of them is visited.

- For example, you can run the following command to query the first 20 tables or partitions that have been visited the most frequently within the last seven days, and the number of times each of them has been visited:

```
jindo table -accessStat -d 7 -n 20
```

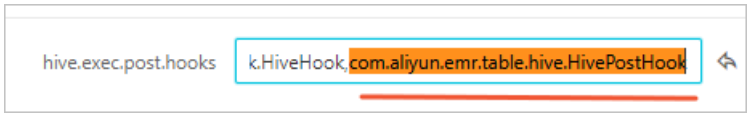
For more information about how to use JindoTable, see Use JindoTable.

Enable the collection of access frequency statistics for Hive

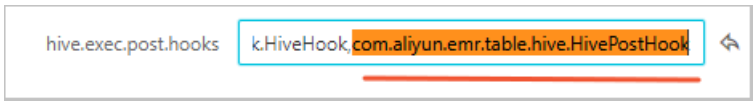
1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of a Hive parameter.
   i. In the left-side navigation pane, choose **Cluster Service > Hive**.
   ii. On the Hive service page, click the **Configure** tab.
   iii. Search for the hive.exec.post.hooks parameter and add **com.aliyun.emr.table.hive.HivePostHook** to the end of the existing parameter value. Separate the hooks with a comma (,).
6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   i. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
   ii. In the **Cluster Activities** dialog box, specify **Description**.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Disable the collection of access frequency statistics

1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of the related parameter.
   - Hive:
     a. In the left-side navigation pane, choose **Cluster Service > Hive**.
     b. On the Hive service page, click the **Configure** tab.
     c. Search for the hive.exec.post.hooks parameter and delete **com.aliyun.emr.table.hive.HivePostHook** from the parameter value.



   - Spark:
     a. In the left-side navigation pane, choose **Cluster Service > Spark**.
     b. On the Spark service page, click the **Configure** tab.
     c. Search for the spark.sql.queryExecutionListeners parameter and delete **com.aliyun.emr.table.spark.SparkSQLQueryListener** from the parameter value.



   - Presto:
     a. In the left-side navigation pane, choose **Cluster Service > Presto**.
     b. On the Presto service page, click the **Configure** tab.
     c. Search for the event-listener.name parameter and delete the parameter value.
6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   - Hive:
     a. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
     b. In the **Cluster Activities** dialog box, specify **Description**.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.
   - Spark:

a. In the upper-right corner of the Spark service page, choose **Actions > Restart ThriftServer**.
b. In the **Cluster Activities** dialog box, specify **Description**.
c. Click **OK**.
d. In the **Confirm** message, click **OK**.

- Presto:
  a. In the upper-right corner of the Presto service page, choose **Actions > Restart All Components**.
  b. In the **Cluster Activities** dialog box, specify **Description**.
  c. Click **OK**.
  d. In the **Confirm** message, click **OK**.

3.4.6. Use JindoTable to collect infrequent-access statistics of tables and partitions

JindoTable can be used to collect infrequent-access statistics of tables and partitions. It enables you to find the data that is not accessed recently based on the last access time of tables and partitions. This helps optimize data storage and reduce costs. For example, in data analytics scenarios, you can use this feature to move infrequently used partitions to lower-cost storage media.

Prerequisites

A cluster is created in E-MapReduce (EMR) V3.35.0 or later V3.X.X, or in EMR V4.9.0 or later V4.X.X. For more information, see Create a cluster.

Context

In SmartData 3.5.X and later, JindoTable can be used to collect infrequent-access statistics of Hive, Spark, and Presto tables. This feature is disabled by default. For more information about how to enable this feature, see Enable the collection of infrequent-access statistics for Spark, Enable the collection of infrequent-access statistics for Hive, and Enable the collection of infrequent-access statistics for Presto.

> **Note** The same hooks and listeners are used to collect frequent-access and infrequent-access statistics. Therefore, if you enable the collection of infrequent-access statistics, the collection of frequent-access statistics is also enabled. For more information about the collection of frequent-access statistics, see Use JindoTable to collect access frequency statistics on tables and partitions.

Limits

- Data Lake Formation (DLF) metadata is not supported.
- You must have the permissions to access the underlying metadata storage, such as MySQL or ApsaraDB RDS, from the IP addresses of the nodes on which the Hive CLI, HiveServer2, Spark SQL CLI, Spark Thrift Server, and Presto services reside.
- You can use JindoTable to collect infrequent-access statistics only of Hive, Spark, and Presto tables.

Query infrequent-access statistics

You can run a command to query infrequent-access statistics.

- Syntax

```
jindo table -leastUseStat -n <num> [-i/-ignoreNever]
```

num indicates the number of tables or partitions that you want to query. Set this parameter to a positive integer. -i/-ignoreNever is an optional parameter. If you specify this parameter, the tables or partitions that have never been accessed are filtered out.

- Description

This command is used to query the tables or partitions that have not been accessed for the longest time.

- Example: Query the first 20 tables or partitions that have not been accessed for the longest time.

```
jindo table -leastUseStat -n 20
```

The following figure shows the output.

```
tdb.t1      pid=20/qid=ten      2021-03-26 13:53:52
tdb.t2                          2021-03-26 13:53:58
```

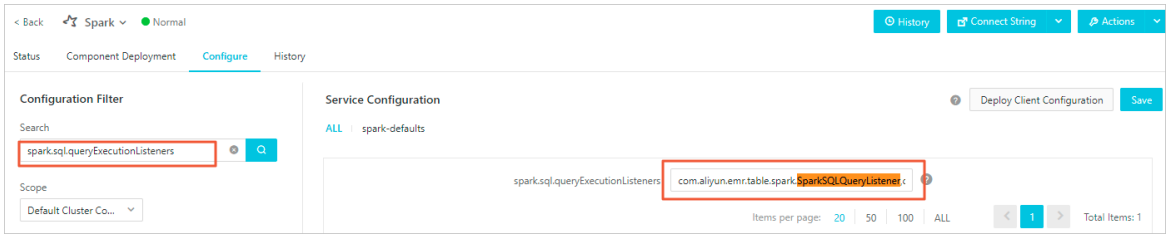
  - The first column lists tables in the Database name.Table name format.
  - The second column lists partitions in the First partition key column=Column value/Second partition key column=Column value/... format. For a non-partitioned table, this column is empty.
  - The third column displays the last access time of the tables or partitions in the yyyy-MM-dd HH:mm:ss format.

> **Note** For a partitioned table, only partition-level access statistics are displayed. Table-level access statistics are not displayed.

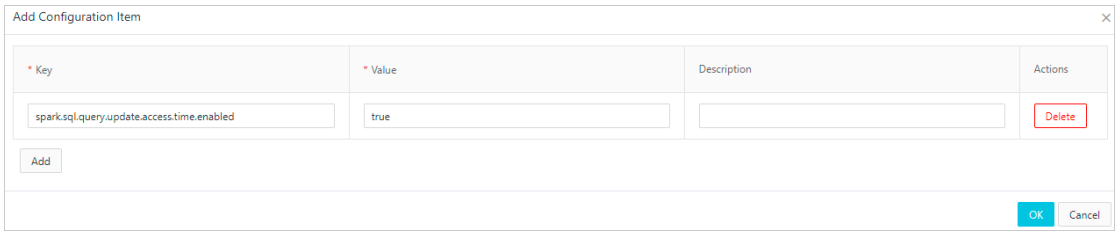
For more information about how to use JindoTable, see Use JindoTable.

Enable the collection of infrequent-access statistics for Spark

1. Go to the Spark service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Spark**.
2. On the Spark service page, click the **Configure** tab.
3. Search for the spark.sql.queryExecutionListeners parameter and check whether the value contains com.aliyun.emr.table.spark.SparkSQLQueryListener. If the value does not contain this listener, add the listener to the value. Separate multiple listeners with comma(,).



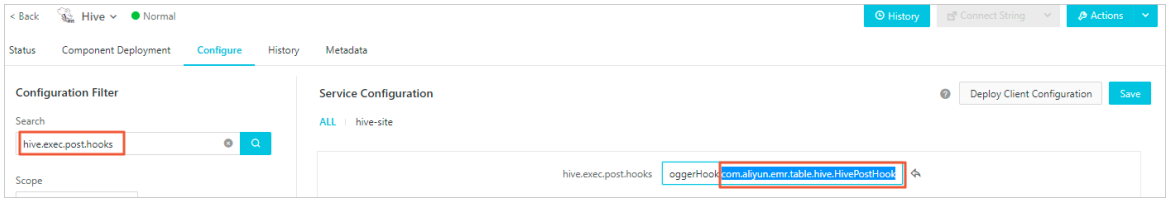
4. Add custom configurations.
   i. In the **Service Configuration** section, click the **spark-defaults** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **spark.sql.query.update.access.time.enabled** parameter and set it to **true**.



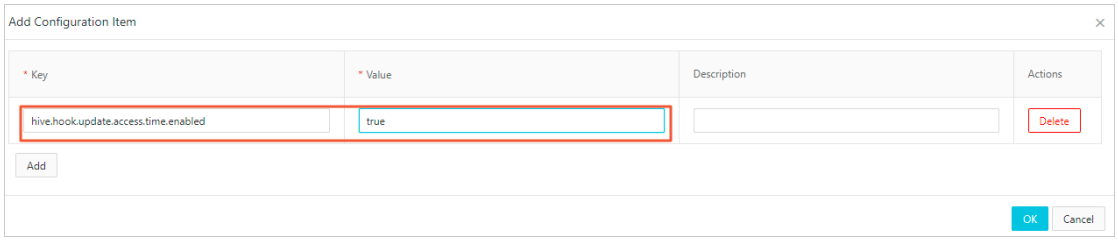
   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

Enable the collection of infrequent-access statistics for Hive

1. Go to the Hive service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Hive**.
2. On the Hive service page, click the **Configure** tab.
3. Search for the hive.exec.post.hooks parameter and check whether the value contains com.aliyun.emr.table.hive.HivePostHook. If the value does not contain this hook, add the hook to the value. Separate multiple hooks with comma(,).



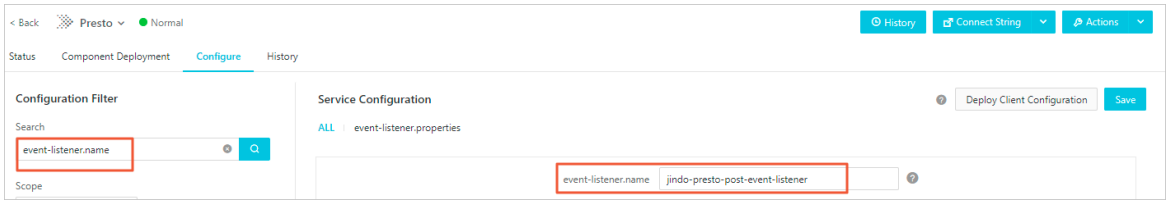
4. Add custom configurations.
   i. In the **Service Configuration** section, click the **hive-site** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **hive.hook.update.access.time.enabled** parameter and set it to **true**.



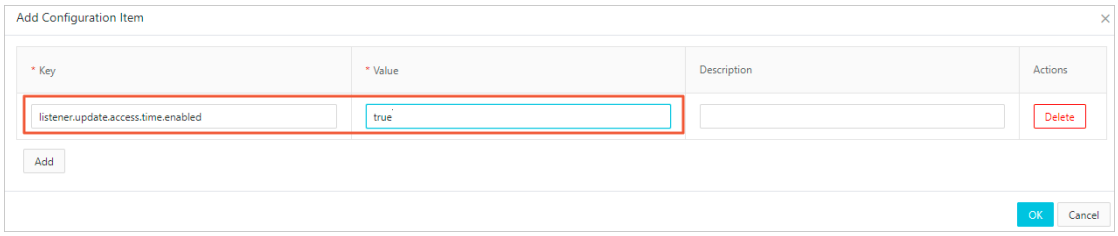
   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

Enable the collection of infrequent-access statistics for Presto

1. Go to the Presto service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Presto**.
2. On the Presto service page, click the **Configure** tab.
3. Search for the event-listener.name parameter and check whether the value contains jindo-presto-post-event-listener. If the value does not contain this listener, add the listener to the value.



4. Add custom configurations.
   i. In the **Service Configuration** section, click the **event-listener.properties** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **listener.update.access.time.enabled** parameter and set it to **true**.



   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

3.5. Toolset

3.5.1. Use Jindo SQL

Jindo SQL is a tool provided by JindoFS. You can use Jindo SQL to analyze metadata, JindoFS access logs, and Object Storage Service (OSS) access logs. This topic describes how to use Jindo SQL to analyze metadata, JindoFS access logs, and OSS access logs.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

You can use Jindo SQL to perform the following operations:

- Analyze JindoFS access logs
- Analyze metadata
- Analyze OSS access logs

Limits

Jindo SQL is supported only in EMR V3.36.0 and later minor versions, and EMR V5.2.0 and later minor versions.

Start Jindo SQL

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to start Jindo SQL:

```
jindo sql
```

The following table describes the common parameters that can be used in the jindo sql command.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically initializes the SQL script after the jindo sql command is run.
-d	Specifies parameters in key-value pairs, such as `-d A=B`.

Structures of built-in tables in Jindo SQL

- audit_log_source (partitioned table)

  The audit_log_source table is used to store raw access logs of JindoFS.

Parameter	Description
datetime	The time format is yyyy-MM-dd HH:mm:ss.
allowed	Indicates whether the current operation is allowed. Valid values: ○ true: The operation is allowed. ○ false: The operation is not allowed.
ugi	The user who performs the operation. Information about the authentication method is provided.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dst	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Parameter	Description
date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

- audit_log

  The audit_log table is used to store JindoFS access logs. You can use the partition key column to filter data in the table.

Parameter	Description
datetime	The time format is yyyy-MM-dd HH:mm:ss.
allowed	Indicates whether the current operation is allowed. Valid values: ○ true: The operation is allowed. ○ false: The operation is not allowed.
ugi	The user who performs the operation. Information about the authentication method is provided.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dst	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.
date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

- fs_image (partitioned table)

  The fs_image table is used to dump image information.

Parameter	Description
atime	The time when the current inode is last accessed.
attr	File-related attributes.
etag	The ETag value of OSS.

Parameter	Description
id	The ID of the inode.
mtime	The time when the inode is modified.
name	The name of the inode.
owner	The name of the owner.
ownerGroup	The name of the group to which the owner belongs.
parentId	The ID of the parent node.
permission	The operation permissions on the file.
size	The size of the inode.
state	The status of the inode.
type	The type of the inode.
storagePolicy	The storage policy.
namespace (partition key column)	The name of the namespace.
datetime (partition key column)	The time when the information is dumped.

- oss_access_log_source

  If the table partitioning mode is enabled, the oss_access_log_source table is a partitioned table. This table is used to store raw access logs of OSS.

Parameter	Description
line	A raw log entry.
bucket (partition key column)	The name of the bucket.
partition_date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

- oss_access_log

  If the table partitioning mode is enabled, you can use the partition key columns to filter data in the oss_access_log table. This table is used to store OSS access logs.

Parameter	Description
Remote_IP	The IP address of the requester.
Reserved	A reserved field. Fixed value: -.
Reserved1	A reserved field. Fixed value: -.

Parameter	Description
Time	The time when OSS receives the request.
Request_URI	The request URL. It contains a query string. OSS ignores the query string parameter that starts with x-. However, this parameter is recorded in access logs. Therefore, you can use the query string parameter that starts with x- as the tag of a request. Then, you can use this tag to query the log that corresponds to the request.
HTTP_Status	The HTTP status code that is returned by OSS.
SentBytes	The downstream traffic generated by the request. Unit: bytes.
RequestTime	The time required to process the request. Unit: milliseconds.
Referer	The HTTP referer of the request.
User_Agent	The user-agent header in the HTTP request.
HostName	The destination domain name in the request.
Request_ID	The ID of the request.
LoggingFlag	Indicates whether log dump is enabled.
Requester	The user ID of the requester. A value of - indicates anonymous access.
Operation	The request type.
Bucket	The name of the destination bucket.
Key	The name of the destination object.
ObjectSize	The size of the destination object. Unit: bytes.
Server_Cost_Time	The time required by OSS to process the request. Unit: milliseconds.
ErrorCode	The error code that is returned by OSS. A value of - indicates that no error code is returned.
RequestLength	The size of the request. Unit: bytes.
UserID	The ID of the bucket owner.
Delta_DataSize	The variation of the bucket size. A value of - indicates that this request does not involve write operations on objects.

Parameter	Description
SyncRequest	Indicates whether the request is an Alibaba Cloud CDN back-to-origin request. Valid values: ○ cdn: The request is a CDN back-to-origin request. ○ -: The request is not a CDN back-to-origin request.
StorageClass	The storage class of the destination object. Valid values: ○ Standard ○ IA ○ Archive ○ Cold Archive ○ -: no storage class obtained
TargetStorageClass	The storage class to which the object is converted after a lifecycle rule is triggered or the CopyObject operation is called. Valid values: ○ Standard ○ IA ○ Archive ○ Cold Archive ○ -: no storage class conversion involved
TransmissionAccelerationAccessPoint	The accelerate endpoint that is used to access the destination bucket based on the transfer acceleration feature. A value of - indicates that no accelerate endpoint is used or the accelerate endpoint is in the same region as the destination bucket. For example, if the requester accesses the destination bucket by using the accelerate endpoint that corresponds to the China (Hangzhou) region, cn-hangzhou is used as the value of this parameter.
AccessKeyID	The AccessKey ID that is used to access the bucket.
bucket (partition key column)	The name of the bucket.
partition_date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

Analyze JindoFS access logs

JindoFS allows you to use SQL queries to analyze the JindoFS access logs that are stored in OSS. You can use SQL queries to analyze the most active commands or IP addresses in the related tables. You can use the `jindo sql` command to analyze the data.

> **Note** The AuditLog feature is enabled. For more information, see Use Jindo AuditLog.

You can use the following commands related to Jindo SQL:

- Run the following command to query tables:

```
show tables;
```

> **Note** For more information about the table structures, see Structures of built-in tables in Jindo SQL.

The information shown in the following figure is returned.

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to query partitions:

```
show partitions audit_log_source;
```

Information similar to the output in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

Information similar to the output in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted] =
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted] =
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted] =
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

Information similar to the output in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics about the use frequencies of different commands that were run on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

Analyze metadata

JindoFS allows you to use SQL queries to analyze the metadata files that are stored in JindoFS. You can use SQL queries to analyze the related tables. You can use the `jindo sql` command to analyze the metadata files.

> **Note** The AuditLog feature is enabled. For more information, see Use Jindo AuditLog.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to start Jindo SQL:

   ```
   jindo sql
   ```

3. Query the tables that can be analyzed by using Jindo SQL.
   - Run the `show tables` command to query the tables that can be analyzed. Jindo SQL provides two built-in modules: audit_log and fs_image, which are used for auditing and metadata analysis.

     The following figure shows an example.

     ```
     [root@emr-header-1 ~]# jindo sql
     Spark master: yarn, Application Id: application_1603081647416_0050
     jindo-sql> show tables;
     database        tableName       isTemporary
     default audit_log       false
     default audit_log_source        false
     default fs_image        false
     Time taken: 0.33 seconds, Fetched 3 row(s)
     ```

   - Run the `show partitions fs_image` command to view the partition information of the

fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

4. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287249    1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       15464684820176650002    1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The following figure shows an example on how to obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

   Note The namespace and datetime columns are added. The namespace column contains the namespace name, and the datetime column contains the timestamp when metadata is uploaded.

Analyze OSS access logs

> **Notice** To analyze OSS access logs, you must specify a log directory and whether to enable the table partitioning mode. If you enable the table partitioning mode, OSS archives logs in the partitioned table by bucket or partition_date, and you can execute statements to query logs in a specific partition. This improves query efficiency. If you enable the partitioning table mode, you must use this mode each time you query data in the table. Otherwise, some files may be archived to the specified log directory, which causes the failure to query some data.

JindoFS allows you to use SQL queries to analyze the OSS access log files that are stored in OSS. You can use SQL queries to analyze the related tables. You can run the `jindo sql` command to analyze the log files.

> **Note** Make sure that log dump is enabled. For more information, see Logging.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to start Jindo SQL:

   ```
   jindo sql
   ```

3. Run the following command to specify the storage path of the OSS access logs and whether the table partitioning mode is enabled.

   ```
   jindo sql -d access_log_path=oss://test-sh/oss-accesslog -d partition.table.enabled=tru
   e
   ```

   `access_log_path` specifies the storage path, and `partition.table.enabled` specifies whether the table partitioning mode is enabled.

FAQ

- Q: How do I modify the startup parameters of Jindo SQL?

  `Jindo SQL` is developed based on Spark. Therefore, the initial resources of Jindo SQL may be insufficient. If you want to increase the resources of Jindo SQL, you can modify the startup parameters of Jindo SQL by using the environment variable `JINDO_SPARK_OPTS`.

  ```
  export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
  --conf spark.executor.cores=5 --conf spark.executor.memory=20G"
  ```

- Q: How do I use Hive to analyze tables?

  Tables in the default database are invisible to Hive jobs. This prevents Hive metadata from being polluted. If you want to use Hive to analyze the tables, you can execute the `show create table {table_name}` statement to query the statement that is executed to create a specific table or execute an SQL statement to create a table. In addition, you must use Hive to load external tables.

3.5.2. Use JindoFS FUSE

This topic describes how to access JindoFileSystem (JindoFS) from a Filesystem in Userspace (FUSE) client. FUSE allows you to use JindoFS in block storage mode and cache mode (only JFS Scheme supported).

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

FUSE is a method provided by the Linux kernel to mount a file system. You can use a FUSE client to map files in a JindoFS cluster to a local disk. Then, you can directly access data in the JindoFS cluster without the need to use the `hadoop fs -ls jfs://<namespace>/` command.

Mount a directory

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs
   ```

3. Run the following command to mount the directory:

   ```
   jindofs-fuse /mnt/jfs
   ```

   In this example, the */mnt/jfs/* directory is mounted.

Read and write files

1. Run the following command to query all the sub-directories under the /mnt/jfs/ directory:

   ```
   ls /mnt/jfs/
   ```

   All the namespaces that you have configured on the server are returned.

   ```
   test  testcache
   ```

2. Run the following command to query all the files in the test namespace:

   ```
   ls /mnt/jfs/test/
   ```

3. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs/test/dir1
   ls /mnt/jfs/test/
   ```

4. Run the following command to write data into a file:

   ```
   echo "hello world" > /tmp/hello.txt
   cp /tmp/hello.txt /mnt/jfs/test/dir1/
   ```

5. Run the following command to read data from a file:

   ```
   cat /mnt/jfs/test/dir1/hello.txt
   ```

   The following information is returned:

   ```
   hello world
   ```

The following example shows how to read data from and write data into files in Python:

1. Write data into a file in Python. Create the *write.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'w',encoding = 'utf-8') as f:
   f.write("my first file\n")
   f.write("This file\n\n")
   f.write("contains three lines\n")
```

2. Read data from a file in Python. Create the *read.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'r',encoding = 'utf-8') as f:
    lines = f.readlines()
    [print(x, end = '') for x in lines]
```

Run the following command to read the data that is written into the *test.txt* file:

```
[hadoop@emr-header-1 ~]$ ./read.py
```

The following information is returned:

```
my first file
This file
```

Unmount FUSE

Note Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to unmount FUSE:

```
umount jindofs-fuse
```

If the `target is busy` error is returned, switch to another directory, stop all the programs that are reading data from and writing data into FUSE files, and then unmount FUSE.

3.5.3. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain help information:

```
jindo distcp --help
```

The following information is returned:

```
     --help           - Print help text
     --src=VALUE          - Directory to copy files from
     --dest=VALUE              - Directory to copy files to
     --parallelism=VALUE         - Copy task parallelism
     --outputManifest=VALUE       - The name of the manifest file
     --previousManifest=VALUE   -   The path to an existing manifest file
     --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
if specified
     --copyFromManifest   -   Copy from a manifest instead of listing a directory
     --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
     --srcPattern=VALUE   -   Include only source files matching this pattern
     --deleteOnSuccess   -   Delete input files after a successful copy
     --outputCodec=VALUE   -   Compression codec for output files
     --groupBy=VALUE   -   Pattern to group input files by
     --targetSize=VALUE   -   Target size for output files
     --enableBalancePlan   -   Enable plan copy task to make balance
     --enableDynamicPlan   -   Enable plan copy task dynamically
     --enableTransaction   -   Enable transation on Job explicitly
     --diff   -   show the difference between src and dest filelist
     --ossKey=VALUE   -   Specify your oss key if needed
     --ossSecret=VALUE   -   Specify your oss secret if needed
     --ossEndPoint=VALUE   -   Specify your oss endPoint if needed
     --policy=VALUE   -   Specify your oss storage policy
     --cleanUpPending   -   clean up the incomplete upload when distcp job finish
     --queue=VALUE   -   Specify yarn queuename if needed
     --bandwidth=VALUE   -   Specify bandwidth per map/reduce in MB if needed
     --s3Key=VALUE   -   Specify your s3 key
     --s3Secret=VALUE   -   Specify your s3 Sercet
     --s3EndPoint=VALUE   -   Specify your s3 EndPoint
```

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-
01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03
/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03
/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017
-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-0
1/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-0
1/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_tab
le/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/
03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/
03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromMani
fest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

Note You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

> **Note** If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

--queue

--queue specifies the name of the YARN queue where the current DistCp task resides.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<your_bucket>/hourly_table --qu
eue yarnqueue
```

--bandwidth

--bandwidth specifies the bandwidth allocated to the current DistCp task. This prevents the task from using excessive bandwidth. Unit: MB/s.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.c
om --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy archive --parallelism 20
  ```

- Example of the IA storage class (`--ia`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy ia --parallelism 20
  ```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

Use Amazon S3 as a data source

You can use the --s3Key, --s3Secret, and --s3EndPoint parameters in a command to specify information related to Amazon S3.

Example:

```
jindo distcp jindo-distcp-2.7.3.jar --src s3a://yourbucket/ --dest oss://<your_bucket>/hour
ly_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com
```

You can configure the s3Key, s3Secret, and s3EndPoint parameters in the *core-site.xml* file of Hadoop. This way, you do not need to specify an AccessKey pair each time.

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

Example:

```
jindo distcp /tmp/jindo-distcp-2.7.3.jar --src s3://smartdata1/ --dest s3://smartdata1/tmp
--s3EndPoint  s3-us-west-1.amazonaws.com
```

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

> **Note** If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

3.5.4. Use Jindo DistCp in specific scenarios

This topic describes how to use Jindo DistCp in specific scenarios.

Prerequisites

- An E-MapReduce (EMR) cluster of a required version is created. For more information, see Create a cluster.
- Java Development Kit (JDK) 1.8 is installed.
- *jindo-distcp-<version>.jar* is downloaded based on your Hadoop version.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 2.7 and later 2.X versions.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 3.X.

Use scenarios

Common use scenarios of Jindo DistCp:

- Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?
- Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?
- Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?
- Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?
- Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file size?
- Scenario 8: What parameters are required if Amazon S3 is used as a data source?
- Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?
- Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?
- Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?
- Scenario 12: What parameters are required if I want to delete original files after a copy operation?
- Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?

If you are not using EMR, the following conditions must be met before you can import HDFS data to Object Storage Service (OSS):

- You can read data from HDFS.
- The AccessKey ID, AccessKey secret, and endpoint of OSS are obtained. You can write data to a destination OSS bucket.
- The storage class of the OSS bucket is not Archive.
- You can submit MapReduce tasks.
- The JAR package of Jindo DistCp is downloaded.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 10
```

> **Note** For more information about the parameters, see Use Jindo DistCp.

To import a large amount of data or a large number of files, such as millions or tens of millions of files, to OSS, you can set parallelism to a large value to increase the concurrency. You can also set `enableBatch` to true for optimization. Optimization command:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 500 --enableBatch
```

Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?

You can use one of the following methods to verify data integrity:

- Jindo DistCp Counters

  Check DistCp Counters in the counter information of a MapReduce task.

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

where:

- Bytes Destination Copied specifies the size of the file that you copy to a destination directory, in bytes.
- Bytes Source Read specifies the size of the file that you read from the source directory, in bytes.
- Files Copied specifies the number of copied files.

- Jindo DistCp --diff

  Use `--diff` to compare the information of files in the source and destination directories. The information includes file name and file size. Missing files and the files that fail to be copied are recorded in a manifest file. The manifest file is generated in the directory where you perform the comparison.

  Add `--diff` to the command described in Scenario 1. Example:

  ```
  hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
  -hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
  aliyuncs.com --diff
  ```

  If all files are copied, the following information is returned:

  ```
  INFO distcp.JindoDistCp: distcp has been done completely
  ```

Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?

If your DistCp task fails, and you want to perform resumable upload to copy only the files that fail to be copied, perform the following operations on the basis of the command described in Scenario 1:

1. Add `--diff` to check whether all files are copied.

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
   hou.aliyuncs.com --diff
   ```

   If all the files are copied, the following information is returned. Otherwise, a manifest file is generated. In this case, go to the next step.

   ```
   INFO distcp.JindoDistCp: distcp has been done completely.
   ```

2. Use `--copyFromManifest` and `--previousManifest` to copy the remaining files. Example:

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --dest oss://yang-hhht/hourly_table --previousManifest=file:///opt
   /manifest-2020-04-17.gz --copyFromManifest --parallelism 20
   ```

   `file:///opt/manifest-2020-04-17.gz` is the path where the manifest file is stored.

Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?

1. If no information is generated for the files that were copied last time, specify parameters to

generate a manifest file to record the information about the files that are copied.

Add `--outputManifest=manifest-2020-04-17.gz` and `--requirePreviousManifest=false` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

where:

- `--outputManifest` generates a manifest file whose name can be customized and must have a file name extension of gz, such as *manifest-2020-04-17.gz*. This file is stored in the directory specified by `--dest`.
- `--requirePreviousManifest` specifies whether to require a previous manifest file.

2. After a DistCp task is complete, copy the new files that may be generated in the source directory.

   Add `--outputManifest=manifest-2020-04-17.gz` and `--previousManifest=oss://yang-hhht/hourly_table/manifest-2020-04-17.gz` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

```
hadoop jar jindo-distcp-2.7.3.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --outputManifest=manifest-2020-04-18.gz --previousManifest=oss://yang-hhht
/hourly_table/manifest-2020-04-17.gz --parallelism 10
```

3. Repeat Step 2 to continuously copy incremental files.

Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?

Add the following two parameters to the command described in Scenario 1. These two parameters can be used together or separately.

- `--queue`: the name of the YARN queue.
- `--bandwidth`: the size of the specified bandwidth , in MB.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --queue yarnqueue --bandwidth 6 --parallelism 10
```

Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?

- To write data to OSS Archive storage, add `--archive` to the command described in Scenario 1.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --archive --parallelism 20
```

- To write data to OSS Infrequent Access (IA) storage, add `--ia` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --ia --parallelism 20
```

Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file sizes?

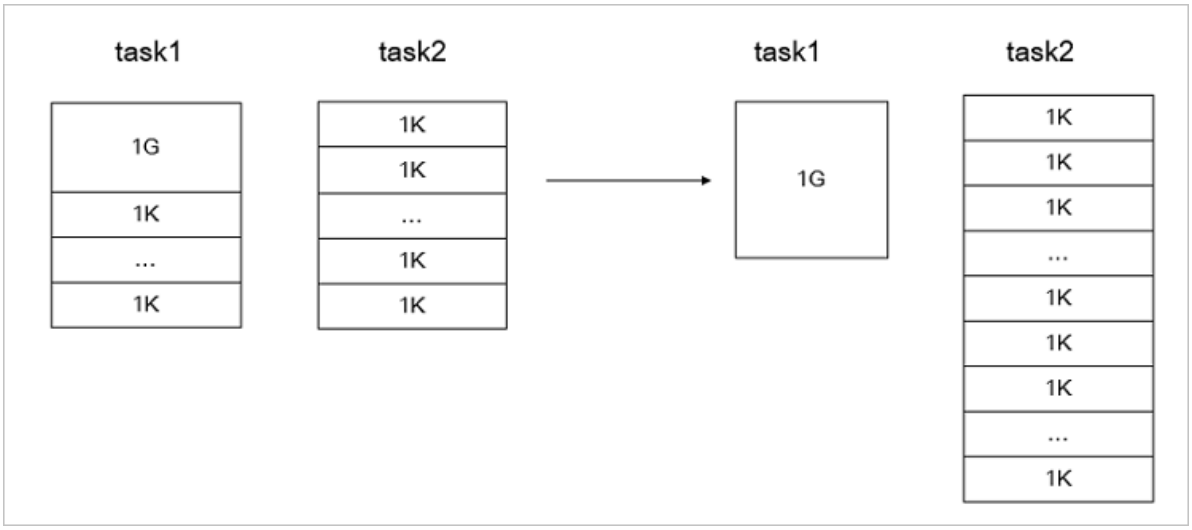
- A large number of small files and large data volume for a single large file

  If most of the files that you want to copy are small files, and the data volume of a single large file is large, the general solution is to randomly allocate the files to be copied. In this case, if you do not optimize the job allocation plan, few large files and many small files may be allocated to the same copy process. This cannot achieve excellent copy performance.

  Add `--enableDynamicPlan` to the command described in Scenario 1 to enable the optimization. --enableDynamicPlan cannot be used with `--enableBalancePlan`. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableDynamicPlan --parallelism 10
```

  The following figure shows the data copy before and after the plan is optimized.

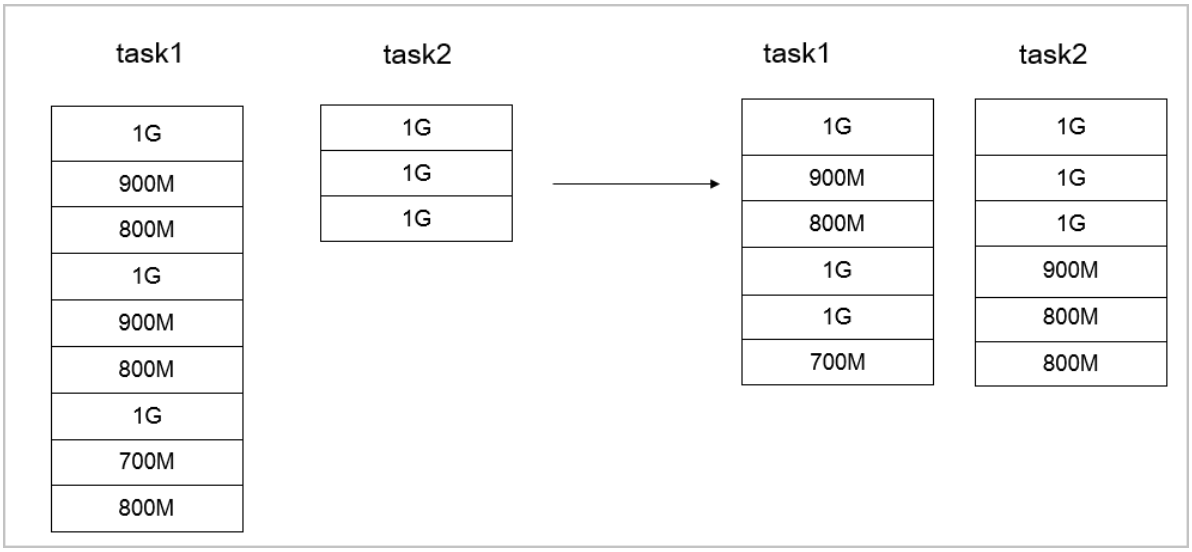


- No significant differences in sizes of files

  If the sizes are not significantly different among the files that you want to copy, use `--enableBalancePlan` to optimize the job allocation plan. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableBalancePlan --parallelism 10
```

The following figure shows the data copy before and after the plan is optimized.



Scenario 8: What parameters are required if Amazon S3 is used as a data source?

Replace the parameters that specify the AccessKey ID, AccessKey secret, and endpoint of OSS in the command described in Scenario 1 with the following parameters of Amazon S3:

- `--s3Key` : the AccessKey ID for Amazon S3
- `--s3Secret` : the AccessKey secret for Amazon S3
- `--s3EndPoint` : the endpoint for Amazon S3

Example:

```
hadoop jar jindo-distcp-<version>.jar --src s3a://yourbucket/ --dest oss://yang-hhht/hourly
_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com --para
llelism 10
```

Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?

You can use `--outputCodec` to compress copied files in a format such as LZO or GZ to reduce the space that is used to store the files.

Add `--outputCodec` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --outputCodec=gz --parallelism 10
```

You can set --outputCodec to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files are compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change to the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?

- If you want to copy the files that meet specific rules, add `--srcPattern` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPattern .*\.log --parallelism 10
```

  `--srcPattern`: specifies a regular expression that filters files for the copy operation.

- If you want to copy the files stored in some sub-directories of the same parent directory, add `--srcPrefixesFile` to the command described in Scenario 1.

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

  `--srcPrefixesFile`: enables Jindo DistCp to copy files in multiple folders under the same parent directory at a time.

  Content of the *folders.txt* file:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?

Add the following parameters to the command described in Scenario 1:

- `--targetSize`: the maximum size of the merged file, in MB
- `--groupBy`: the merging rule, which is a regular expression

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Scenario 12: What parameters are required if I want to delete original files after a copy operation?

Add `--deleteOnSuccess` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --deleteOnSuccess --parallelism 10
```

Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Jindo DistCp allows you to save the AccessKey ID, AccessKey secret, and endpoint of OSS in the *core-site.xml* file so that you do not need to repeatedly specify the information.

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of OSS, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.jfs.cache.oss-accessKeyId</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-accessKeySecret</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-endpoint</name>
        <value>oss-cn-xxx.aliyuncs.com</value>
    </property>
</configuration>
```

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of Amazon S3, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

3.5.5. Use tiered-storage commands of JindoFS

In E-MapReduce (EMR) V3.30, JindoFS provides a tiered storage feature. This feature allows you to store cold and hot data in different storage media. This helps reduce the data storage costs and accelerate data access.

Use jindo jfs

Run the following command to obtain help information:

```
jindo jfs -help archive
```

Tiered-storage commands in JindoFS are asynchronous and start only related tasks.

Common commands:

- Cache
- Uncache
- Archive
- Unarchive
- Status
- ls2

Cache

You can use this command to back up data stored in a specific path to local disks. Then, you can read the data from local disks, without the need to read data from Object Storage Service (OSS).

```
jindo jfs -cache -p <path>
```

-p can be used to ensure that local data is not cleared based on disk usage.

Uncache

You can use this command to delete backup data from local disks and store data only to OSS Standard storage.

```
jindo jfs -uncache  <path>
```

Archive

You can use this command to delete backup data from local disks and store data to OSS Infrequent Access (IA) or Archive storage. For information about the storage classes, see Overview.

```
jindo jfs -archive -i|-a|-c <path>
```

- If you specify the -i option, data is stored to OSS IA storage.
- If you specify the -a option, data is stored to OSS Archive storage.
- If you specify the -c option, data is stored to OSS Cold Archive storage.

Unarchive

You can use this command to convert the storage class of data from Archive to IA or Standard. You can temporarily unarchive data that is stored to Archive storage to allow the data to be readable for one day.

```
jindo jfs -unarchive -i/-o <path>
```

By default, this command can be used to store data to OSS Standard storage.

- If you specify the -i option, data is stored to OSS IA storage.
- If you specify the -o option, data stored to Archive storage can be temporarily unarchived and becomes readable.

Status

You can use this command to view task details. By default, the number of files for which you want to use tiered storage in a specific directory and the data to which tiered storage has been applied are measured.

```
jindo jfs -status -detail/-sync <path>
```

- If you specify the -detail option, the storage progress of file data can be viewed.
- If you specify the -sync option, the command exits only after a tiered-storage task is completed.

ls2

JindoFS provides the ls2 command that allows you to view the file storage status on the basis of Hadoop ls commands.

```
hadoop fs -ls2 <path>
```

Example of command output, which includes the file storage class:

```
drwxrwxrwx  - -         0    2020-06-05 04:27 oss://xxxx/warehouse
-rw-rw-rw-  1 Archive   1484 2020-09-23 16:40 oss://xxxx/wikipedia_data.csv
-rw-rw-rw-  1 Standard  1676 2020-06-07 20:04 oss://xxxx/wikipedia_data.json
```

4.SmartData 3.7.x

4.1. SmartData 3.7.X overview

SmartData is a storage service for the E-MapReduce (EMR) Jindo engine. SmartData provides centralized storage, caching, and computing optimization for EMR compute engines. This service also provides additional features. SmartData consists of JindoFS, JindoTable, and related tools. This topic describes the updates in SmartData 3.7.X.

JindoFS

The following table describes the updates to JindoFS.

Feature	Description
Display of statistics	SmartData 3.7.2 and later versions support this feature. JindoFS collects statistics on important metrics, such as Object Storage Service (OSS) read/write throughput, cache read/write throughput, and cache usage. You can connect JindoFS to Prometheus to monitor metrics in a visualized manner.
Cold Archive storage class supported for tiered storage of data	SmartData 3.7.3 and later versions support this storage class. You can specify the -c option in the Archive command to store data to Cold Archive storage. This helps reduce the storage costs of cold data. For more information, see Use tiered-storage commands of JindoFS.

JindoSDK

The following table describes the updates to JindoSDK.

Feature	Description
Support for access to Amazon Simple Storage Service (S3)	The JindoFS client supports Amazon S3. You can access data in Amazon S3 by using the JindoFS client.
OSS PrefixLink feature	The JindoFS client supports the OSS PrefixLink feature. You can use this feature to improve the rename performance of JindoFS. This effectively accelerates Hive jobs and has a significant optimization effect in extract, transform, load (ETL) scenarios of Hive jobs.
Support for atomicity of the rename operation	SmartData 3.7.2 and later versions support this feature. The JindoFS client supports atomicity of the rename operation based on the configuration of the x-oss-forbid-overwrite request header. This is suitable for scenarios in which Delta Lake is supported.

JindoTable

The following table describes the updates to JindoTable.

Feature	Description
Conversion of the storage class to Cold Archive and unarchiving of data stored to Cold Archive storage	SmartData 3.7.2 and later versions support this feature. JindoTable allows you to run commands in SDK mode to convert the storage class to Cold Archive and to unarchive data stored to Cold Archive storage. For more information, see Archive and unarchive data in SDK mode.

4.2. JindoFS in block storage mode

4.2.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read and write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

Context

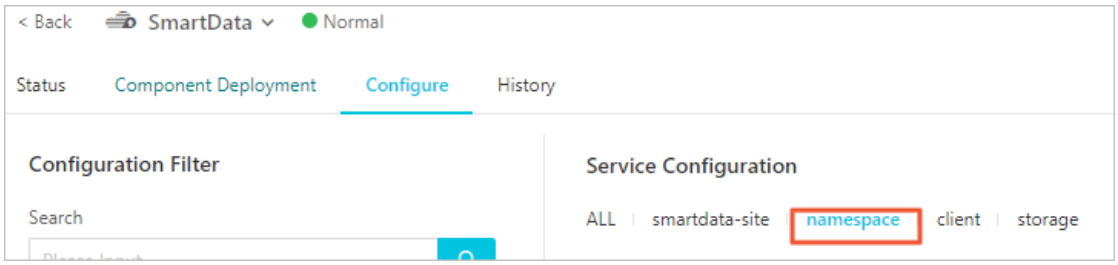
The block storage mode of JindoFS has the following features:

- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.
- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.

ii. Click the **namespace** tab in the Service Configuration section.



3. Configure required parameters.

JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

i. Set **jfs.namespaces** to **test**.

If you configure multiple namespaces, separate them with commas (,).

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block

Parameter	Description	Example
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

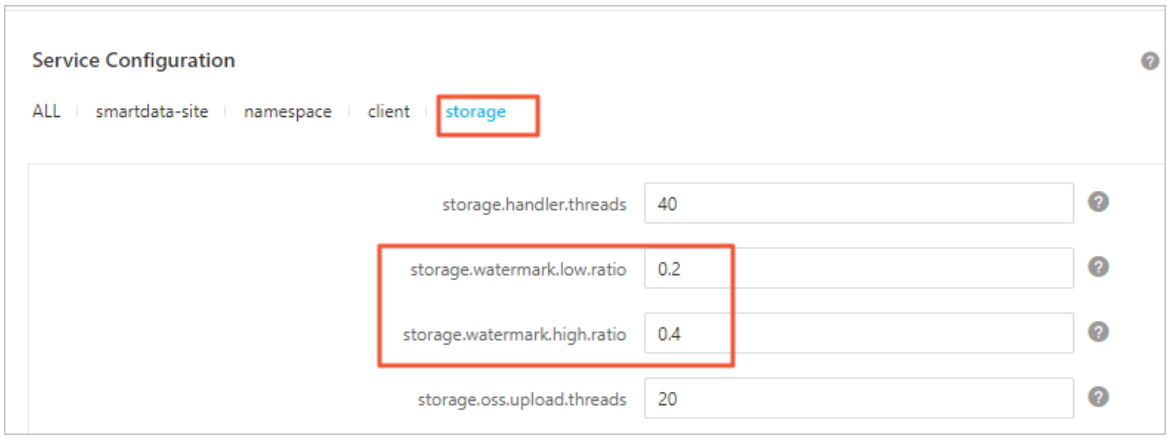
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

Note You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Choose **Actions > Restart Jindo Storage Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

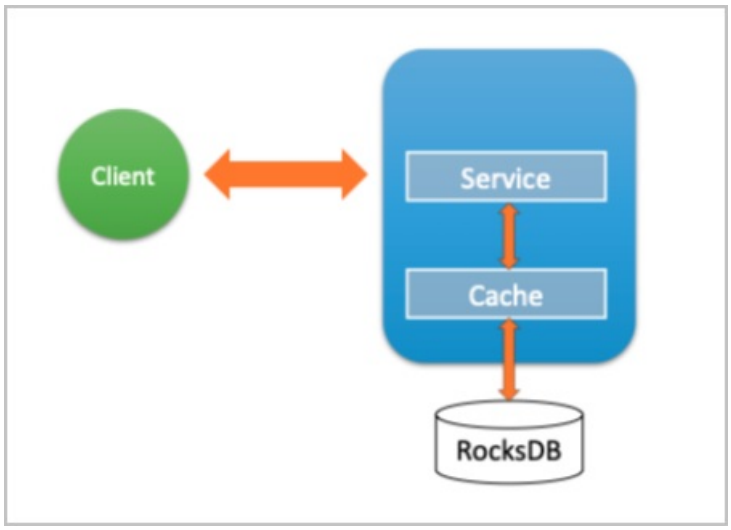
4.2.2. Use RocksDB to store metadata

Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

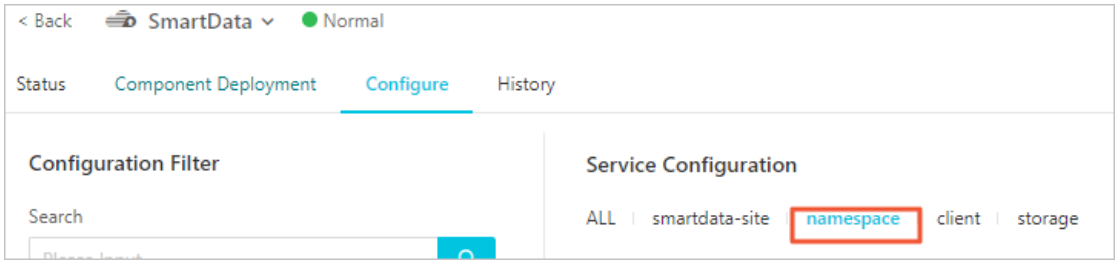
RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage backend in high availability mode, we recommend that you use Raft instances. For more information, see Use Raft-RocksDB-Tablestore to store metadata.

The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

6. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

7. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
8. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

      ```
      hadoop fs -count jfs://test/
      ```

      Information similar to the following example is returned:

      ```
       1596      1482809                25 jfs://test/
      ```

      The number of folders is 1596, and the number of files is 1,482,809.

   ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1`, the latest metadata is synchronized to the Tablestore instance.

      ```
      jindo jfs -metaStatus
      ```

      Information similar to that shown in the following figure is returned.

      ```
      [hadoop@emr-header-1 ~]$ jindo jfs -metaStatus
      ==== emr-header-1:8101 ====
      OtsUploader: _synced=1
      [RocksDB Row Counts of each CF(Table)]
      ```

   iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.
2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components.
3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus
```

If the state is FINISH, the recovery is completed, as shown in the following figure.

```
==== emr-header-1:8101 ====
[Recovery From OTS Status]
  state: FINISH
  total 22855 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

4.2.3. Use Raft-RocksDB-Tablestore to store metadata

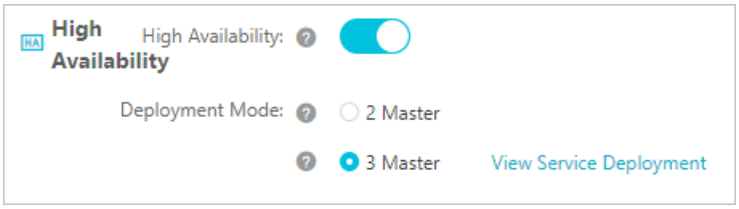
In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata.

Prerequisites

- A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  > **Note** You must enable the transaction feature.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.

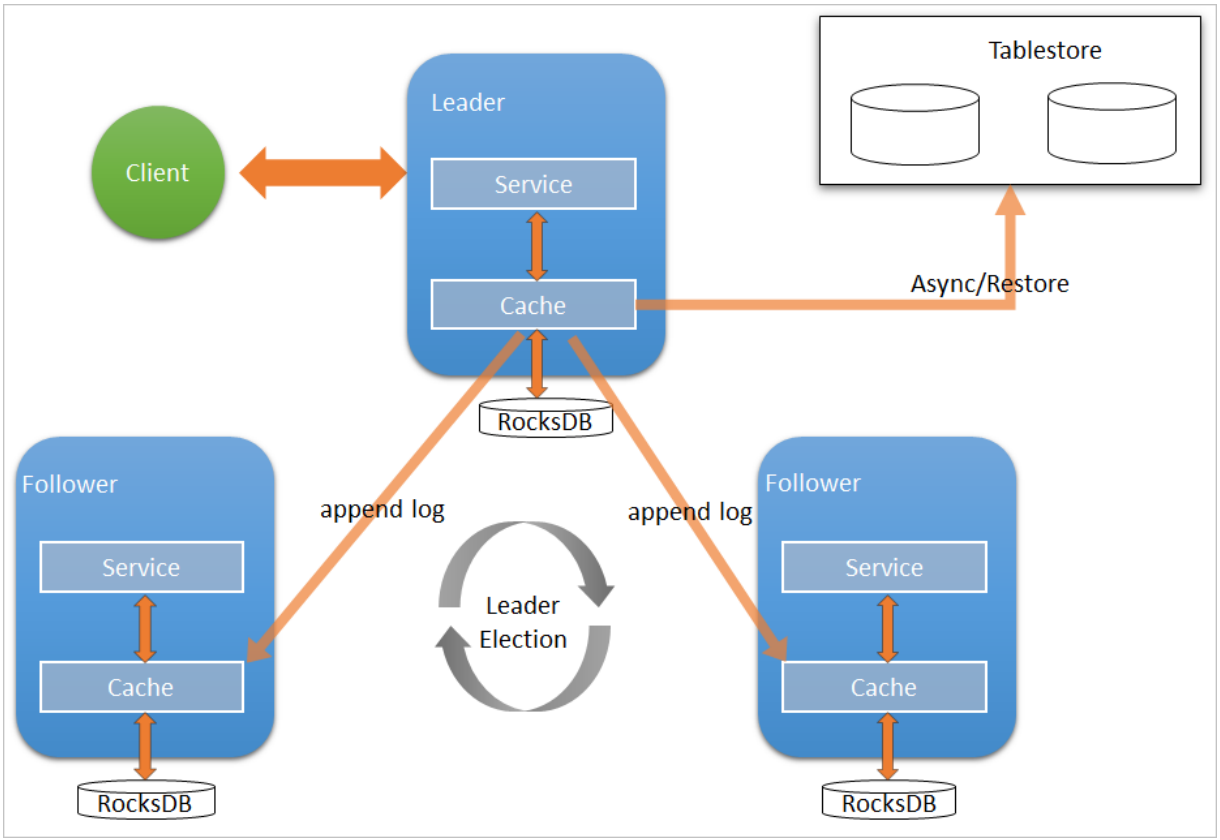


> **Note** If no deployment modes are available, submit a ticket.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.



Configure a Raft instance as the local storage backend

1. Suspend all the SmartData components of an EMR cluster.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   vi. Select **Stop All Components** from the **Actions** drop-down list in the upper-right corner.
2. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
3. Go to the **namespace** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **namespace** tab.

4. Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. Valid values: ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

Note If you do not need to use Tablestore for remote storage, skip Step 5.

5. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

```
hadoop fs -count jfs://test/
      1596      1482809                 25 jfs://test/
  (Number of folders) (Number of files)
```

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```

```
[RaftPeerImpl]
peer id: [redacted]
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 2335ms)
snapshot_timer: timeout(3600000ms) SCHEDULING(in 150305ms)
storage: [1, 624625]
disk_index: 624625
known_applied_index: 624625
last_log_id: (index=624625,term=2)
first_index_pinned: 624625
state_machine: Idle
last_committed_index: 624625
last_snapshot_index: 0
last_snapshot_term: 0
snapshot_status: IDLE
replicator_25769803789@[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=624261 ic=0
replicator_329853488332[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=623564 ic=0

OtsUploader: _lastStopIndex=624624, _synced=1
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.

```
[RaftPeerImpl]
peer_id: [redacted]:8103:0
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 3382ms)
snapshot_timer: timeout(600000ms) SCHEDULING(in 474855ms)
storage: [1, 153]
disk_index: 153
known_applied_index: 153
last_log_id: (index=153,term=2)
first_index_pinned: 1
state_machine: Idle
last_committed_index: 153
last_snapshot_index: 1
last_snapshot_term: 2
snapshot_status: IDLE
replicator_1116691496965@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0
replicator_3311419785217@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0

[Recovery From OTS Status]
state: FINISH
total rows: 1484409
table `jfs_block_test` 2 rows.
table `jfs_namespace_cache_ns` 1 rows.
table `jfs_namespace_test` 1484406 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

4.2.4. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see

Create a cluster.

- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.

ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

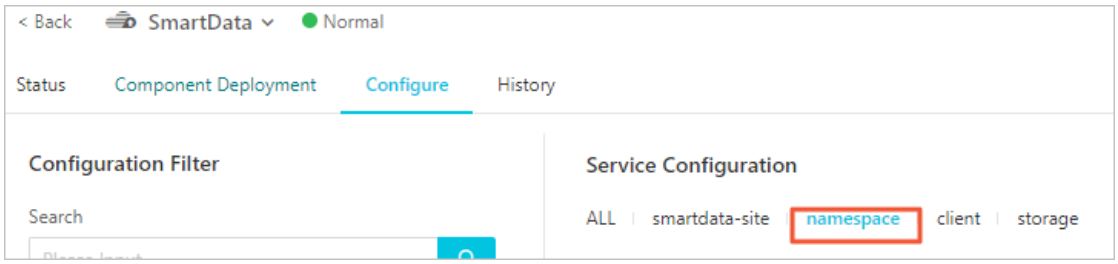
iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **namespace** tab for the SmartData service.

   i. Click the **Configure** tab.

   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:

   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

4.2.5. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

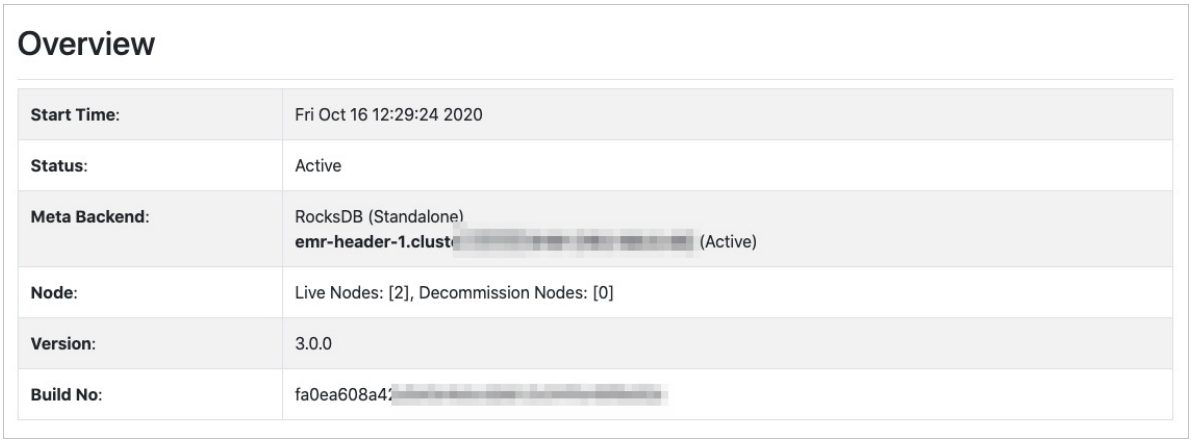
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

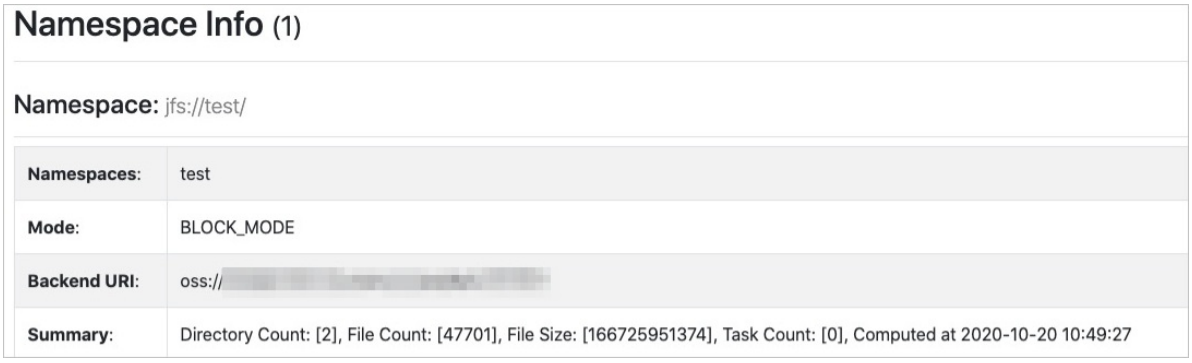


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



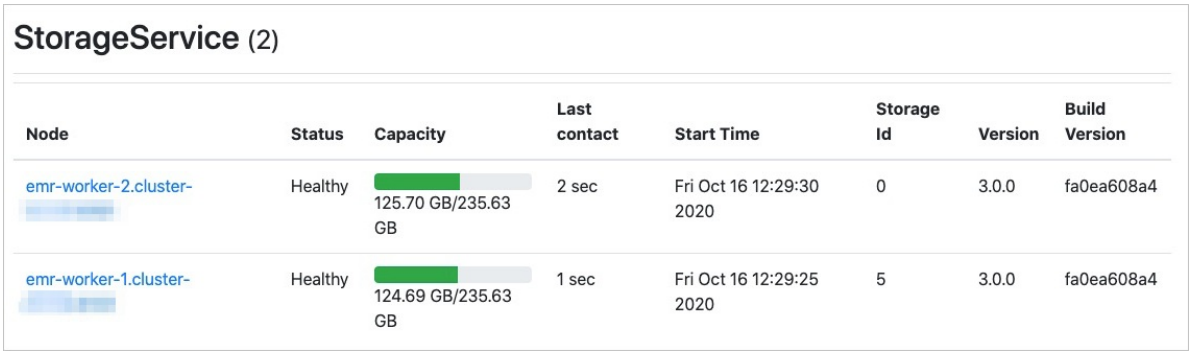
Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

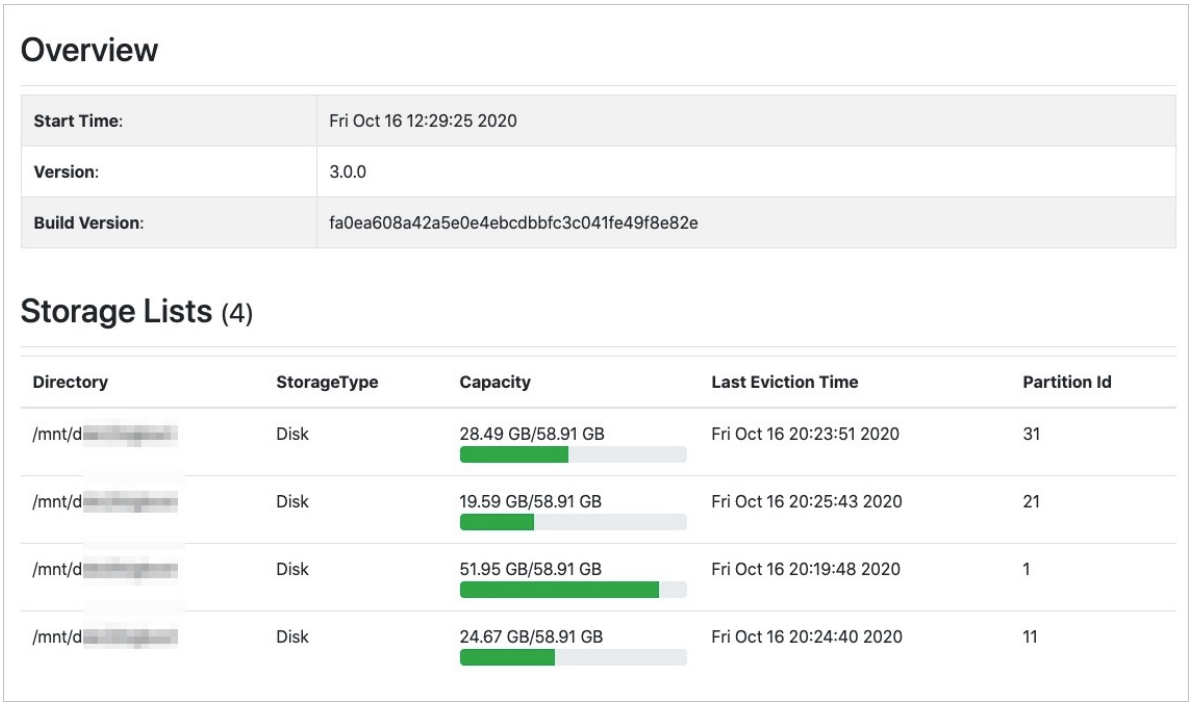
  This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.



StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.



Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

4.2.6. Manage JindoFS permissions

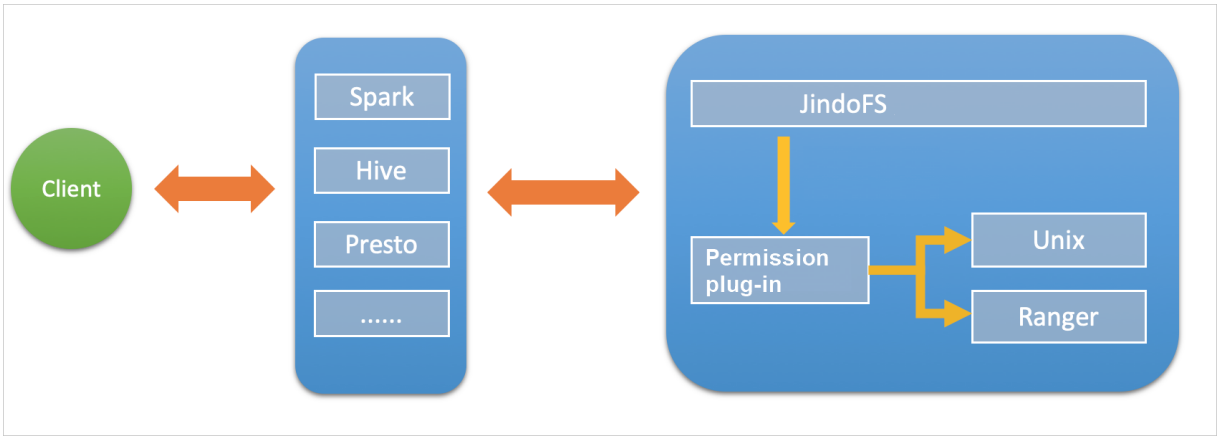
This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

Permission management based on the storage mode:

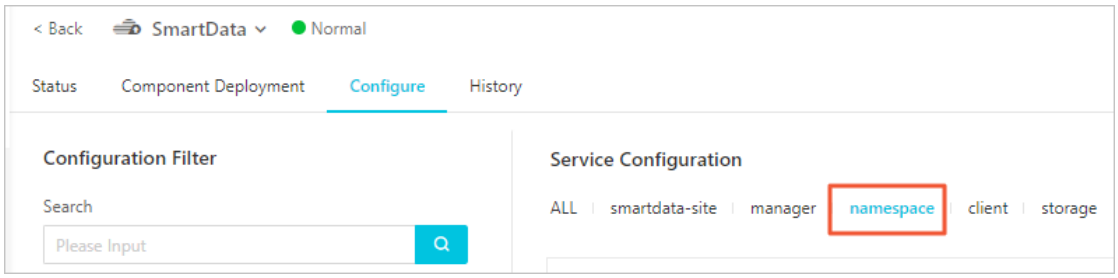
- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.
  - Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

4.2.7. Data management policies

JindoFS in block storage mode provides advanced data management policies to meet storage requirements in different scenarios. The advanced policies include storage policies and compression policies. This topic describes the policies in detail and provides examples to show how to use the policies.

Storage policies

JindoFS provides the five storage policies described in the following table to support flexible storage.

Policy	Description
AR	Data has only one backup. The backup is stored as an Archive object in OSS.
IA	Data has only one backup. The backup is stored as an Infrequent Access (IA) object in OSS.

Policy	Description
COLD	Data has only one backup. The backup is stored as a Standard object in OSS.
WARM	Data has one backup in OSS and one local backup. The local backup can effectively accelerate data read operations. By default, this storage policy is used.
HOT	Data has one backup in OSS and one local backup. The local backup is forcibly locked. This way, it is not deleted when the cache is automatically cleared. This policy can achieve a better acceleration effect than the WARM policy and is suitable for the hottest data.

For more information about the storage classes of OSS, see Overview.

A newly added file is stored based on the storage policy specified for its parent directory. Example:

- Run the following command to configure a storage policy:

```
jindo jfs -setStoragePolicy [-R] <StoragePolicy>(AR/IA/COLD/WARM/HOT) <path> ...
```

  where:

  - `[-R]`: specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.
  - `<path>`: the name of the directory for which the storage policy is configured.

- Run the following command to obtain the storage policy configured for a directory:

```
jindo jfs -getStoragePolicy <path>
```

Compression policies

JindoFS allows you to compress data blocks before you store them. This feature effectively reduces the storage space occupied by data blocks and improves data read and write efficiency. This feature is suitable for files that have a high compression ratio. The following table describes the supported compression policies.

Policy	Description
NONE	Data blocks are not compressed. By default, this compression policy is used.
ZSTD	The Zstandard compression algorithm is used to compress data blocks.

A newly added file is compressed based on the compression policy specified for its parent directory before it is stored. Example:

- Run the following command to configure a compression policy:

```
jindo jfs -setCompressionPolicy [-R] <CompressionPolicy>(NONE/ZSTD) <path> ...
```

where:

- `[-R]`: specifies that a recursive operation is performed to configure the same compression policy for all sub-directories of the directory.
- `<path>`: the name of the directory for which the compression policy is configured.

- Run the following command to obtain the compression policy configured for a directory:

```
jindo jfs -getCompressionPolicy <path> ...
```

4.2.8. Analyze metadata offline

In E-MapReduce (EMR) V3.30.0 or later, JindoFS in block storage mode allows you to dump the metadata of an entire namespace to Object Storage Service (OSS) and use Jindo SQL to analyze the metadata.

Context

In Hadoop Distributed File System (HDFS), the metadata is stored in a snapshot file named fsimage. The snapshot file contains the following information about metadata: namespace, file that stores the metadata, blocks, and file system quota. In HDFS, you can run a command to download the fsimage file in the XML format to your on-premises machine and view this file to analyze metadata offline. JindoFS does not require you to download metadata to your on-premises machine.

Upload HDFS metadata to OSS

Run the following command to upload metadata of a specific namespace to OSS:

```
jindo jfs -dumpMetadata <nsName>
```

`<nsName>` indicates the name of the namespace in block storage mode.

For example, upload the metadata of the test-block namespace to OSS and analyze the metadata offline.

```
jindo jfs -dumpMetadata test-block
```

If the following information appears, data is uploaded to OSS and stored as a file in the JSON format.

```
Sucessfully upload namespace metadata to OSS.
```

Metadata upload path

The metadata upload path is the metadataDump sub-directory under *sysinfo* in JindoFS.

For example, if `namespace.sysinfo.oss.uri` is set to `oss://abc/`, metadata is uploaded to the `oss://abc/metadataDump` directory.

Parameter	Description
namespace.sysinfo.oss.uri	The path of the OSS bucket.
namespace.sysinfo.oss.endpoint	The endpoint. An endpoint specified in a different region can also be used.
namespace.sysinfo.oss.access.key	The AccessKey ID of your Alibaba Cloud account.
namespace.sysinfo.oss.access.secret	The AccessKey secret of your Alibaba Cloud account.

Batch information: Metadata from HDFS changes based on the usage. Each time you want to analyze metadata, you must run commands to view the snapshot file of the metadata. Each time you run a Jindo command to upload metadata, a batch number that is generated based on the upload time is used as the root directory to store the metadata. This ensures that the data you uploaded each time is not overwritten. You can delete historical data based on your requirements.

- 1 indicates the configuration path of system information in OSS.
- 2 indicates a namespace.
- 3 indicates a batch number.

Metadata schema

HDFS metadata that is uploaded to OSS is stored as a file in the JSON format. Schema information:

```
{
  "type":"string",          /*Inode type, FILE DIRECTORY*/
  "id": "string",            /*INode id*/
  "parentId" :"string",         /*Parent node ID*/
  "name":"string",         /*Inode name*/
  "size": "int",         /*Inode size, BIGINT*/
  "permission":"int",         /*Permission, INT*/
  "owner":"string",          /*Owner name*/
  "ownerGroup":"string",     /*Owner group name*/
  "mtime":"int",              /*Inode modification time, BIGINT*/
  "atime":"int",              /*The time when the inode was last accessed, BIGINT*/
  "attributes":"string",       /*File-related attributes*/
  "state":"string",            /*Inode status*/
  "storagePolicy":"string",    /*Storage policy*/
  "etag":"string"           /*etag*/
}
```

Use Jindo SQL to analyze metadata

1. Run the command shown in the following figure to start Jindo SQL.

```
[root@emr-header-1 ~]# jindo sql
Spark master: yarn, Application Id: application_1603081647416_0050
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.33 seconds, Fetched 3 row(s)
```

2. Query the tables whose data can be analyzed by using Jindo SQL.
   - Run the `show tables` command to view the tables whose data can be analyzed. Jindo SQL has two built-in modules audit_log and fs_image for audit and metadata analysis.
   - Run the `show partitions fs_image` command to view the partition information of the fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

     The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

3. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287249    1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       1546468482017665002     1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The namespace and datetime columns are added and indicate the namespace name and the timestamp when metadata is uploaded.

Example: Obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

Use Hive to analyze metadata

1. Create a table schema in Hive.

   You can use the following command to create a table schema of the metadata that is used for queries in Hive:

```
CREATE EXTERNAL TABLE `table_name`
(`type` string,
 `id` string,
 `parentId` string,
 `name` string,
 `size` bigint,
 `permission` int,
 `owner` string,
 `ownerGroup` string,
 `mtime` bigint,
 `atime` bigint,
 `attr` string,
 `state` string,
 `storagePolicy` string,
 `etag` string)
 ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
 STORED AS TEXTFILE
 LOCATION 'The OSS path to which file data is uploaded';
```

2. Use Hive to analyze data offline.

   After you create a Hive table, you can use Hive SQL to analyze metadata.

```
select * from table_name limit 200;
```

   The following figure shows an example.

```
hive> select * from inode_metadata_test0 limit 100;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_202009[redacted]
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1[redacted], Tracking URL = http://emr-heade[redacted]:20888/proxy/applica[redacted]
Kill Command = /usr/lib/hadoop-current/bin/hadoop job  -kill job_1599[redacted]
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-09-08 14:57:26,112 Stage-1 map = 0%,  reduce = 0%
2020-09-08 14:57:31,263 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.22 sec
MapReduce Total cumulative CPU time: 1 seconds 220 msec
Ended Job = job_[redacted]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.22 sec   HDFS Read: 6867 HDFS Write: 1524 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 220 msec
OK
Directory       1127433438684721971 4    1127433438684721971 3    /uttest/oss     0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Directory       1127433438684721971 9    1127433438684721971 3    /uttest/oss2    0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
Directory       1127433438684721971 6    1127433438684721971 4    /uttest/oss/dir 0       511     caojie  staff   1599545017636   1599545017636           Finalized       WARM
File    1127433438684721971 5    1127433438684721971 4    /uttest/oss/file1       0       420     caojie  staff   1599545017632   1599545017632           Finalized       WARM
File    1127433438684721971 7    1127433438684721971 6    /uttest/oss/dir/file2   0       420     caojie  staff   1599545017642   1599545017642           Finalized       WARM
File    1127433438684721971 8    1127433438684721971 6    /uttest/oss/dir/file3   0       420     caojie  staff   1599545017651   1599545017651           Finalized       WARM
Directory       1127433438684721972 0    1127433438684721971 9    /uttest/oss2/dir        0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
File    1127433438684721972 1    1127433438684721972 0    /uttest/oss2/dir/file2  0       420     caojie  staff   1599545017658   1599545017658           Finalized       WARM
File    1127433438684721972 2    1127433438684721972 0    /uttest/oss2/dir/file3  0       420     caojie  staff   1599545017666   1599545017666           Finalized       WARM
Directory       1127433438684721971 3    1767282355743385190 5    /uttest 0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Time taken: 10.734 seconds, Fetched: 10 row(s)
hive>
```

4.2.9. Use a JindoFS credential provider

The data of JindoFS is stored in Object Storage Service (OSS). If you want to access the data of JindoFS, you must provide an AccessKey pair that can be used to access OSS. SmartData V3.4.0 and later support JindoFS credential providers. You can use a JindoFS credential provider to save an encrypted AccessKey pair into a file. This prevents the AccessKey pair from being leaked.

Configure a JindoFS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **smartdata-site** tab.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **smartdata-site** tab.
3. Add configuration information.
   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

Parameter	Description
fs.jfs.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

   iii. Click **OK**.
4. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Credential provider types

- TemporaryAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a validity period and a security token with a validity period are used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this credential provider, you must configure the parameters described in the following table in environment variables.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**.
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- JindoCommonCredentialsProvider

This is a common credential provider. After you configure the parameters described in the following table, you can use this credential provider in both JindoOSS and JindoFS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.JindoCommonCredentialsProvider**.
jindo.common.accessKeyId	The AccessKey ID used to access OSS.
jindo.common.accessKeySecret	The AccessKey secret used to access OSS.
jindo.common.securityToken	The temporary security token used to access OSS.

- EcsStsCredentialsProvider

  This credential provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EcsStsCredentialsProvider**.

4.2.10. Use JindoFS in block storage mode to store encrypted files

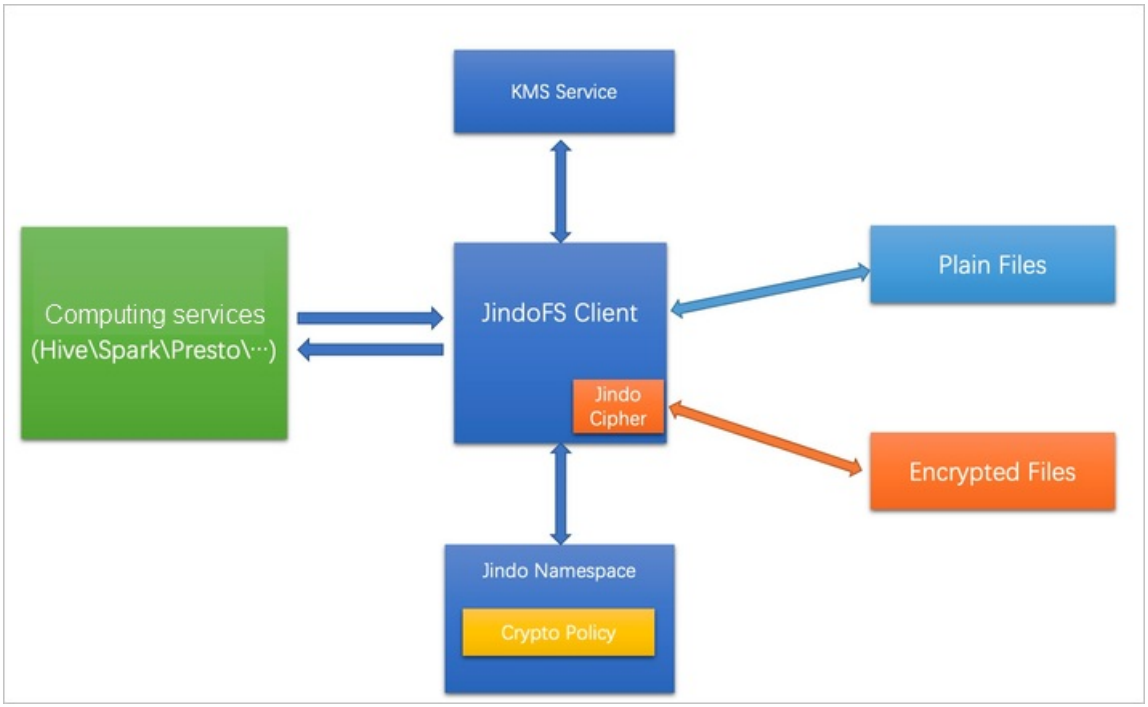
JindoFS in block storage mode supports file encryption. The encryption mechanism and usage method are similar to those of encryption zones for Apache HDFS. Keys are managed by Key Management Service (KMS). You can configure encryption policies for a directory that contains sensitive data. This way, you can transparently encrypt the data that you want to write to the directory and transparently decrypt the data that you want to read from the directory. In this process, you do not need to change your code.

Prerequisites

- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.
- KMS is activated. For more information, see Activate KMS.

Context

The following figure shows the architecture of JindoFS in block storage mode with file encryption supported.



Configure KMS-related parameters for JindoFS

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **namespace** tab.
3. Add configuration information.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description
crypto.provider.type	The type of the provider. Set this parameter to **ALIYUN**.
crypto.provider.endpoint	The public endpoint of KMS. For more information, see Make API requests.
crypto.provider.kms.accessKeyId	The AccessKey ID used to access KMS.
crypto.provider.kms.accessKeySecret	The AccessKey secret used to access KMS.

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Use JindoFS KeyProvider

JindoFS KeyProvider is used to connect JindoFS to KMS. The keys for JindoFS are stored in KMS. You can use JindoFS KeyProvider to create, query, and rotate keys based on KMS.

- Create a key: You can run the following command to create a key:

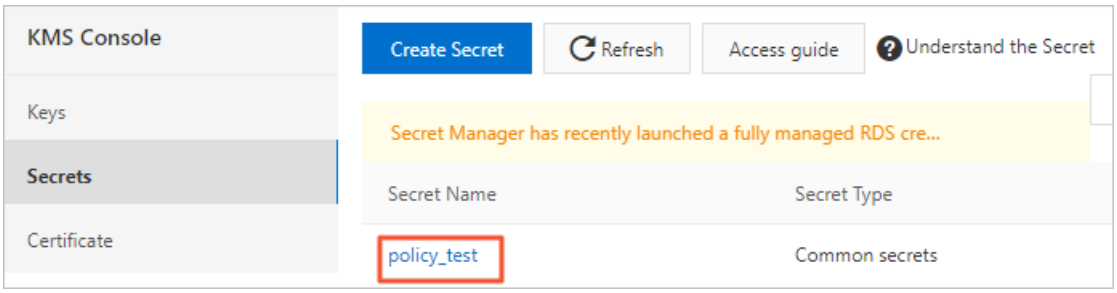
```
jindo key -create -keyIdName <keyIdName>
```

> **Note** In this topic, <keyIdName> in commands indicates the name of the key that you created.

For example, run the following command to create a key named *policy_test*:

```
jindo key -create -keyIdName policy_test
```

After the *policy_test* key is created, it is displayed on the Secrets page of the KMS console.



- Query keys: You can run the following command to query the names of all existing keys for JindoFS:

```
jindo key -list
```

The following information is returned:

```
Listing Keys:
        policy_test
        policy_test2
```

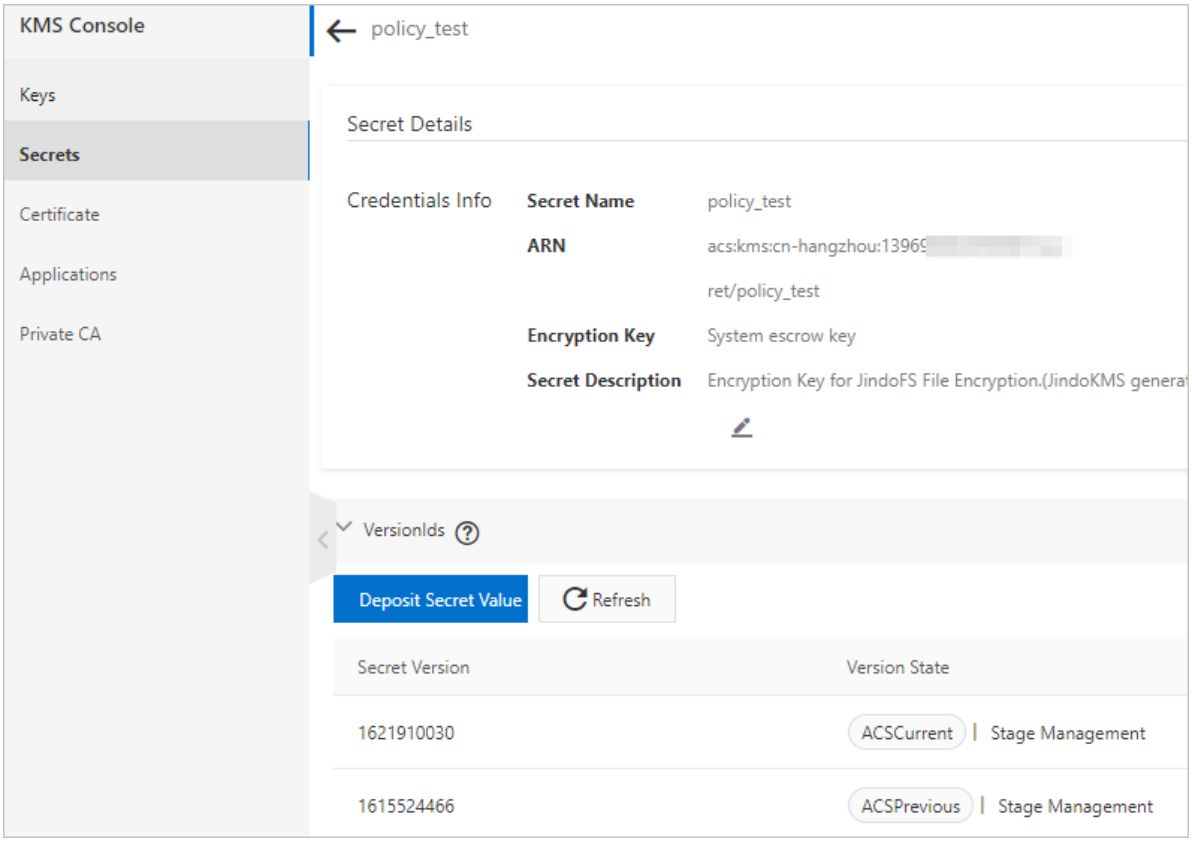
- Rotate a key: You can run the following command to periodically rotate a key based on its ID. After the key is rotated, the key version changes. If a file is encrypted after the rotation, the latest key version is used. If a file is decrypted after the rotation, the key version that corresponds to the file is used.

```
jindo key -roll -keyIdName <keyIdName>
```

For example, run the following command to rotate the *policy_test* key:

```
jindo key -roll -keyIdName policy_test
```

After the *policy_test* key is rotated, the version of the key is updated. The original version is in the **ACSPrevious** state and the new version is in the **ACSCurrent** state.



Manage encryption policies for JindoFS

You can run the following commands to configure and query encryption policies.

- Configure an encryption policy:

```
jindo jfs -setCryptoPolicy -keyIdName <keyIdName> <path>
```

> **Note** `<path>` in the preceding example indicates the path of a file on JindoFS, for example, *jfs://test/*.

- Query an encryption policy:

```
jindo jfs -getCryptoPolicy <path>
```

Example:

1. Query the encryption policy of the *jfs://test/* path:

```
jindo jfs -getCryptoPolicy jfs://test/
```

`{NONE}` is returned.

2. Configure an encryption policy for the *jfs://test/* path:

```
jindo jfs -setCryptoPolicy -keyIdName policy_test jfs://test/
```

3. Go to the *bigboot* directory and query the encryption policy of the *jfs://test/* path again:

```
jindo jfs -getCryptoPolicy jfs://test/
```

The following information is returned:

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/b2jindosdk/3.4.0-hadoop3.1/pack
age/b2jindosdk-3.4.0-hadoop3.1/lib/jindo-distcp-3.4.0.jar!/org/slf4j/impl/StaticLoggerB
inder.class]
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/hadoop/3.2.1-1.0.1/package/hado
op-3.2.1-1.0.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticL
oggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
21/03/12 13:52:34 WARN: Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable
21/03/12 13:52:35 INFO: Jboot log name is /var/log/bigboot/jboot-20210312-135234-12953.
LOG
21/03/12 13:52:35 INFO: Write buffer size 1048576, logic block size 134217728
21/03/12 13:52:35 INFO: cmd=getFileStatus, src=jfs://test/, dst=null, size=0, parameter
=null, time-in-ms=7, version=3.4.0
21/03/12 13:52:35 INFO: cmd=getCryptoPolicy, src=jfs://test/, dst=null, size=0, paramet
er=, time-in-ms=2, version=3.4.0
The crypto policy of path: jfs://test/ is {cipherSuite: AES_CTR_NOPADDING_256, keyIdNam
e: policy_test2, keyIdVersion: null, edek: , iv: }
21/03/12 13:52:35 INFO: Read total statistics: oss read average <none>, cache read aver
age <none>, read oss percent <none>
```

After you configure the encryption policy, you can read data from or write data to the files under this path.

- Copy a local file to HDFS:

```
hadoop fs -put test.log jfs://test/
```

- Display the content of a file:

```
hadoop fs -cat jfs://test/test.log
```

4.3. JindoFS in cache mode

4.3.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in Object Storage Service (OSS) based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

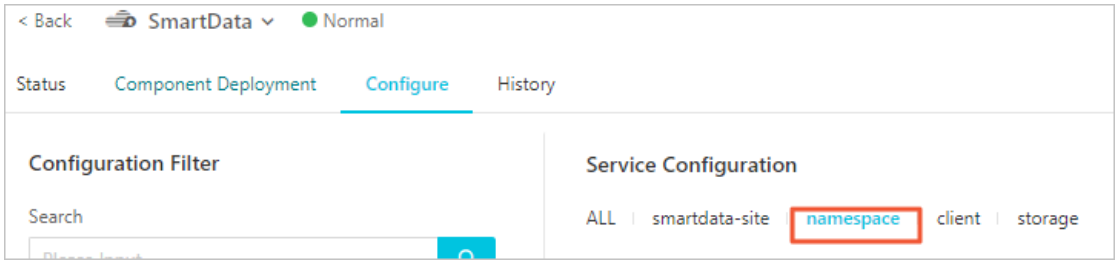
  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. Click **OK**.
5. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
6. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **client** tab.
2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   You do not need to restart the SmartData service to apply the new configurations.
3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

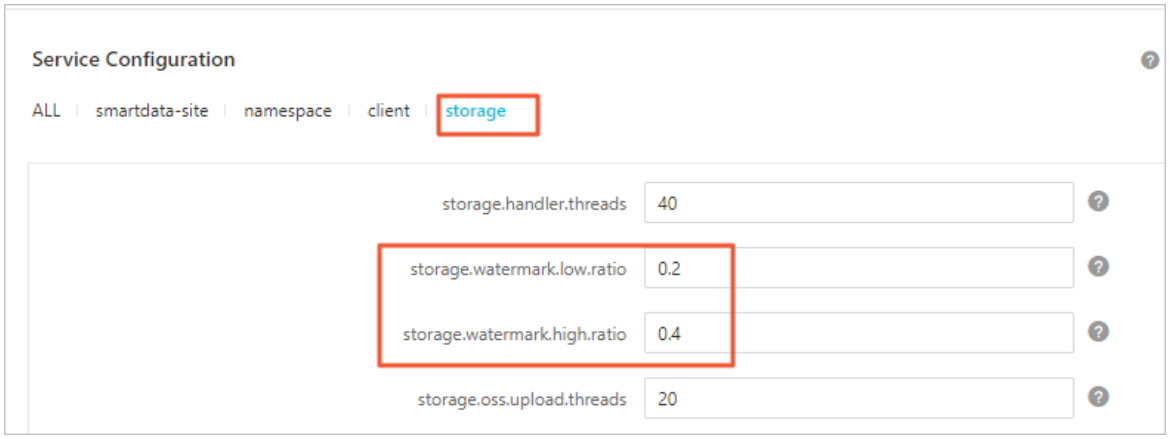
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you specify. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to specify an AccessKey pair. In other cases, you must specify an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you choose for accessing files in OSS:

- OSS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.
  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss.accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.endpoint	The endpoint of the OSS bucket that serves as the storage backend.

> **Note** You can set these parameters to the values of the parameters in an EMR version earlier than EMR V3.30.0.

- JFS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **namespace** tab.
  ii. Set **jfs.namespaces** to **test**.
  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize the cache performance. After you configure the parameters, you do not need to restart the SmartData service to apply the new configurations.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. You can use this parameter to prevent upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If an excessive amount of memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

4.3.2. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

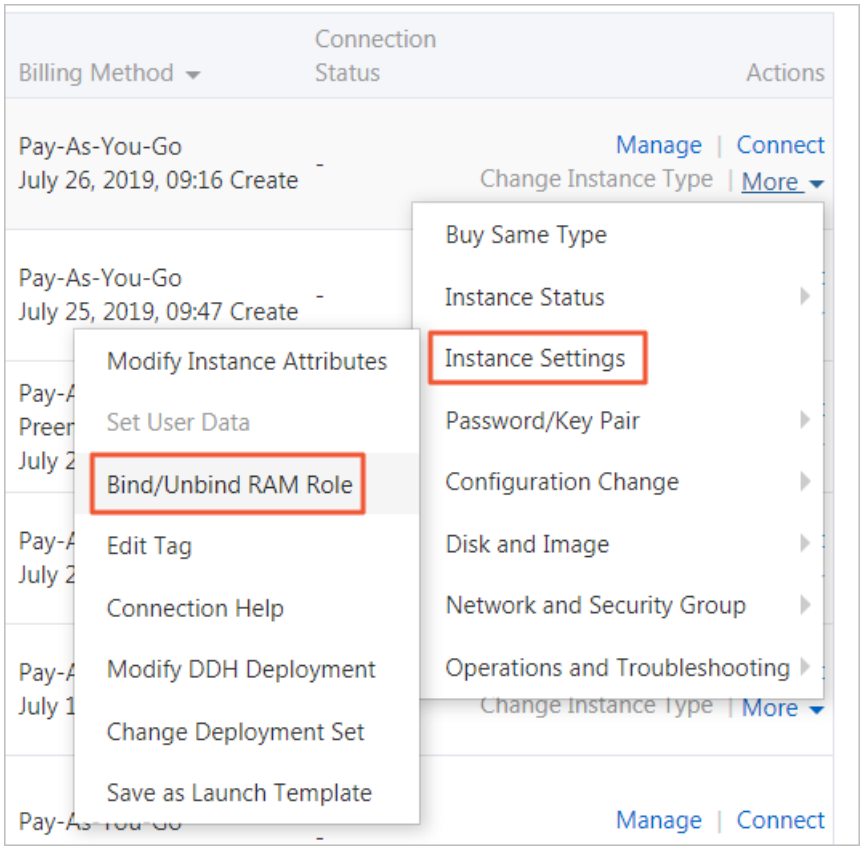
1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.
4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

   ```
   hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
   ```

2. Access OSS by using the FileSystem interface of Hadoop.

   Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

4.3.3. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false

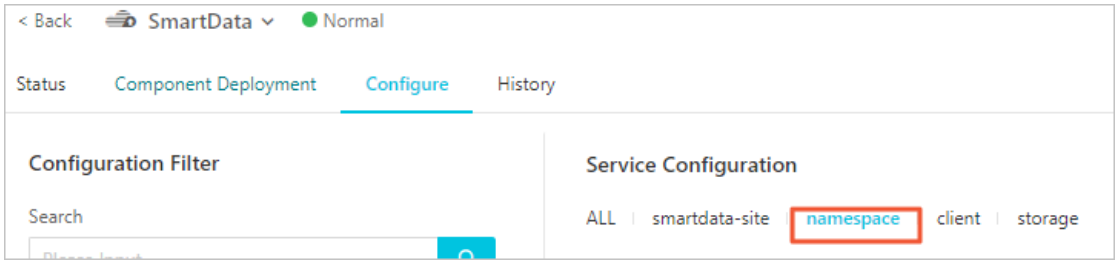
Parameter	Description
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-

right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

4.3.4. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

> **Notice**
>
> - The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
> - After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
> - Hive and Presto jobs do not use Hadoop Job Committer.
> - In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.

i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
ii. Click the **Configure** tab.
iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.
6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.

3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

   The default value is 8.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

4.3.5. Use JindoFS OSS credential providers

You can use a credential provider to save an encrypted AccessKey pair into a file. This prevents the leak of the AccessKey pair.

Context

You can use a Hadoop credential provider to save an encrypted AccessKey pair into a file. This prevents the issue that the AccessKey pair is transmitted in plaintext. You can select an appropriate JindoOSS credential provider based on your business requirements.

Configure a JindoOSS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **smartdata-site** tab.

   i. Click the **Configure** tab.

   ii. In the **Service Configuration** section, click the **smartdata-site** tab.

3. Add configuration information.

   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

   - Global configuration (for all buckets)

Parameter	Description
fs.jfs.cache.oss.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

- Bucket-level configuration

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types. **Note** XXX indicates the name of an Object Storage Service (OSS) bucket.

Credential provider types

You can select different credential providers based on your business requirements. The following providers are supported:

- Global configuration
  - TemporaryAliyunCredentialsProvider

    This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.cache.oss.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

- Bucket-level configuration

- TemporaryAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.
fs.jfs.cache.oss.bucket.XXX.securityToken	The temporary security token used to access the OSS bucket.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access an OSS bucket.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access the OSS bucket.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access the OSS bucket. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

Use a Hadoop credential provider to store AccessKey pair information

> **Note** For more information about Hadoop credential providers, see CredentialProvider API Guide.

Use a command that is provided by Hadoop to store AccessKey pair and security token information into a credential file. Syntax:

```
hadoop credential <subcommand> [options]
```

For example, in global configuration mode, store AccessKey pair and token information into a JCEKS file. You can protect the file by using file permissions or you can specify a password to encrypt the information you want to store. If you do not specify a password, the default string is used for encryption.

```
hadoop credential create fs.jfs.cache.oss.accessKeyId -value AAA -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.accessKeySecret -value BBB -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.securityToken -value  CCC -provider jceks://file/root/oss.jceks
```

After a credential file is generated, you must configure the parameter described in the following table to specify the provider type and location.

Parameter	Description
fs.jfs.cache.oss.security.credential.provider.path	The path used to store the credential file that stores AccessKey pair information. For example, you can set this parameter to *jceks://file/${user.home}/oss.jceks*, which indicates that the *oss.jceks* file is stored in the home directory.

4.3.6. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

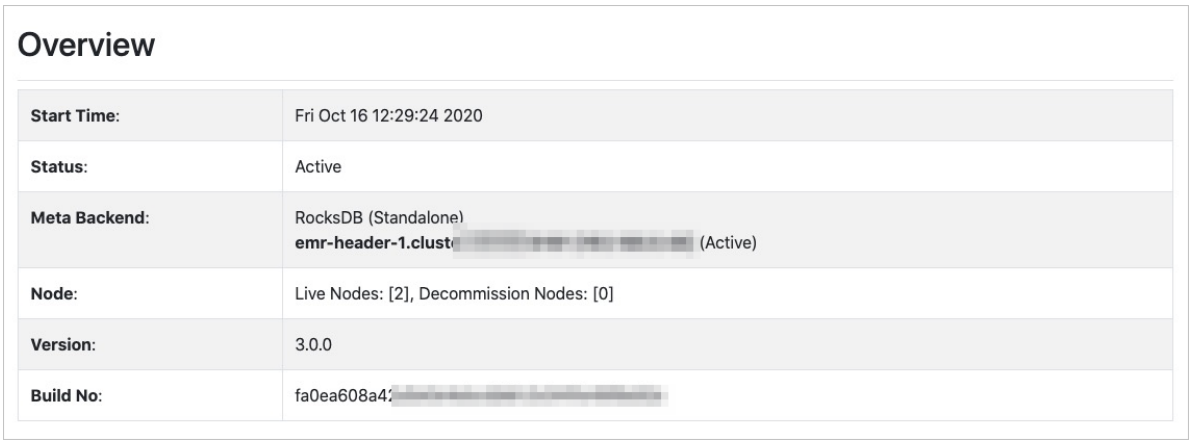
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

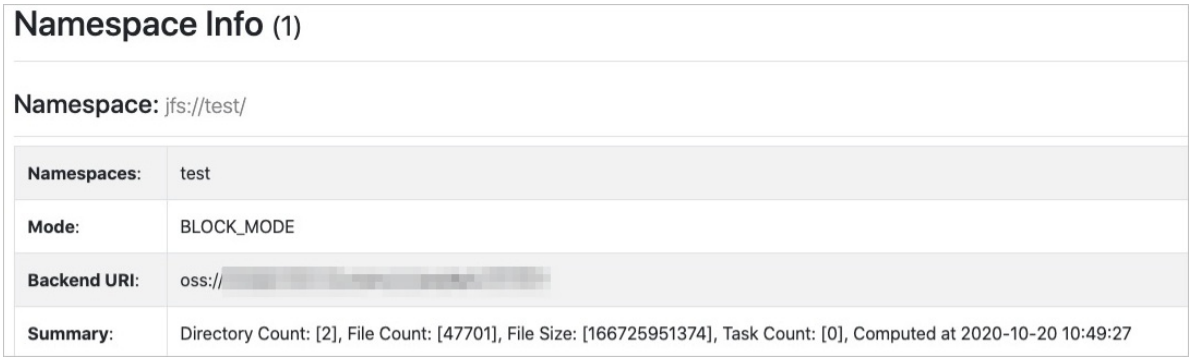
  

  Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.

  Namespace Info (1)

  Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.

StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.

Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

4.3.7. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

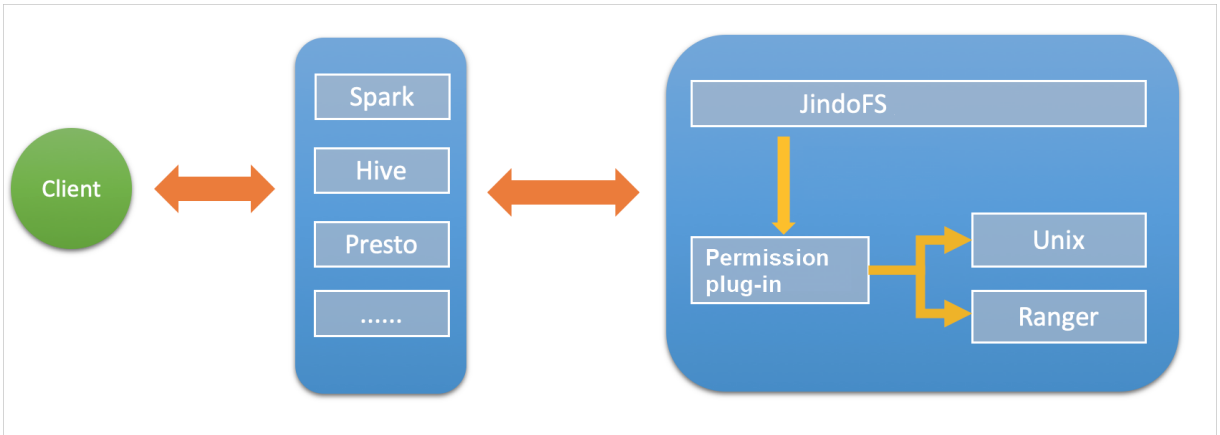
Permission management based on the storage mode:

- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.

- Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.

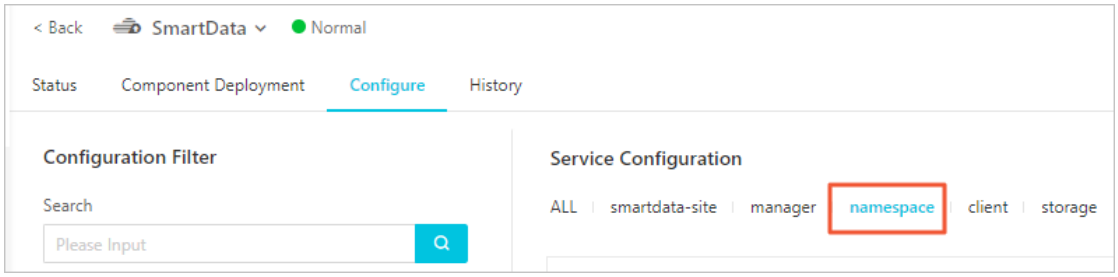
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

   After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

   If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

4.4. JindoTable

4.4.1. Enable query acceleration based on a native engine

JindoTable provides a native engine for you to accelerate the speed of using Spark, Hive, or Presto to query files in the ORC or Parquet format. This topic describes how to enable query acceleration based on the native engine to improve the performance of Spark, Hive, and Presto.

Prerequisites

An E-MapReduce (EMR) cluster is created, and a file in the ORC or Parquet format is stored in JindoFS or Object Storage Service (OSS). For more information about how to create a cluster, see Create a cluster.

Limits

- Binary files are not supported.

- Partitioned tables whose values of partition key columns are stored in files are not supported.
- EMR clusters of V5.X.X or later are not supported.
- spark.read.schema (userDefinedSchema) is not supported.
- Data of the DATE type must be in the YYYY-MM-DD format and range from 1400-01-01 to 9999-12-31.
- If a table contains two columns that have the same column name with different letter cases, such as ID and id, queries on the columns cannot be accelerated.
- The following table lists the supported Spark, Hive, and Presto engines and the file formats supported by each engine.

Engine	ORC	Parquet
Spark2	Supported	Supported
Spark3	Supported	Supported
Presto	Supported	Supported
Hive2	Not supported	Supported
Hive3	Not supported	Supported

- The following table lists the supported Spark, Hive, and Presto engines and the file systems supported by each engine.

Engine	OSS	JindoFS	HDFS
Spark2	Supported	Supported	Supported
Presto	Supported	Supported	Supported
Hive2	Supported	Supported	Not supported
Hive3	Supported	Supported	Not supported

Improve the performance of Spark

1. Enable query acceleration for ORC or Parquet files in JindoTable.

> **Note**
>
> - Query acceleration consumes off-heap memory. We recommend that you add `--conf spark.executor.memoryOverhead=4g` to a Spark task to apply for additional resources for query acceleration.
> - If you use Spark to read data from an ORC or Parquet file, the DataFrame API or Spark SQL is required.

   - Global configuration

a. Go to the Cluster Overview page of your cluster.
   a. Log on to the Alibaba Cloud EMR console.
   b. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   c. Click the **Cluster Management** tab.
   d. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
b. Modify the related parameter.
   a. In the left-side navigation pane, choose **Cluster Service > Spark**.
   b. On the Spark service page, click the **Configure** tab.
   c. Search for the spark.sql.extensions parameter and change its value to io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sql.JindoTableExtension.
c. Save the configuration.
   a. Click **Save** in the upper-right corner.
   b. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
d. Restart ThriftServer.
   a. Choose **Actions > Restart ThriftServer** in the upper-right corner.
   b. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   c. In the **Confirm** message, click **OK**.

- Job-level configuration

  When you configure a Spark Shell or Spark SQL job, add the following configuration to the code:

  ```
  --conf spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sq
  l.JindoTableExtension
  ```

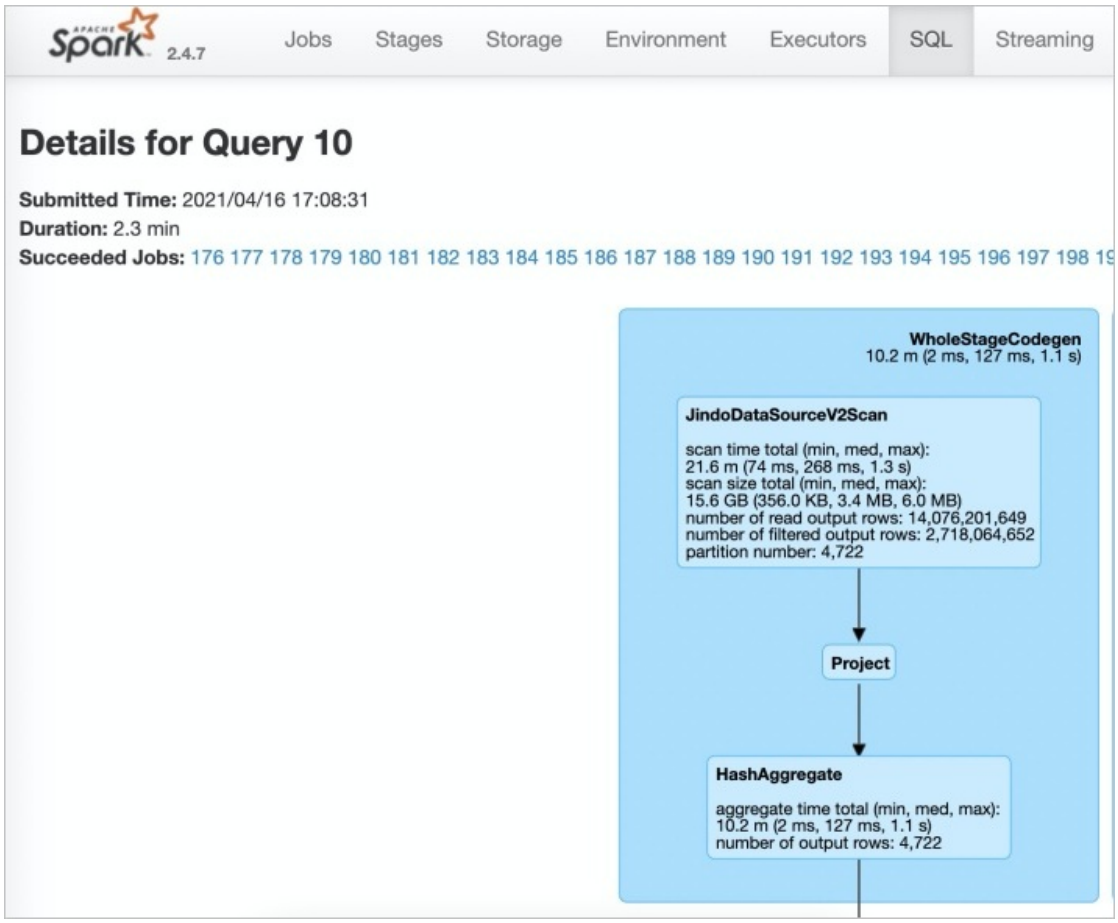
  For more information about how to configure a Spark job, see Configure a Spark Shell job or Configure a Spark SQL job.

2. Check whether query acceleration is enabled.
   i. Access the web UI of Spark History Server.

      For more information, see Access the web UIs of open source components.

ii. On the SQL tab of Spark, view information about the related task.

If JindoDataSourceV2Scan appears, query acceleration is enabled. Otherwise, check the configurations in Step 1.



Improve the performance of Presto

> **Notice** Presto has high query concurrency and uses off-heap memory for query acceleration. To use the query acceleration feature, make sure that the memory is greater than 10 GB.

By default, `catalog: hive-acc` of the native engine is built in the Presto service. You can use `catalog: hive-acc` to enable query acceleration.

Example:

```
presto --server https://emr-header-1.cluster-xxx:7778/ --catalog hive-acc --schema default
```

`emr-header-1.cluster-xxx` indicates the hostname of the emr-header-1 node.

> **Note** When you use this feature in Presto, complex data types such as MAP, STRUCT, and ARRAY are not supported.

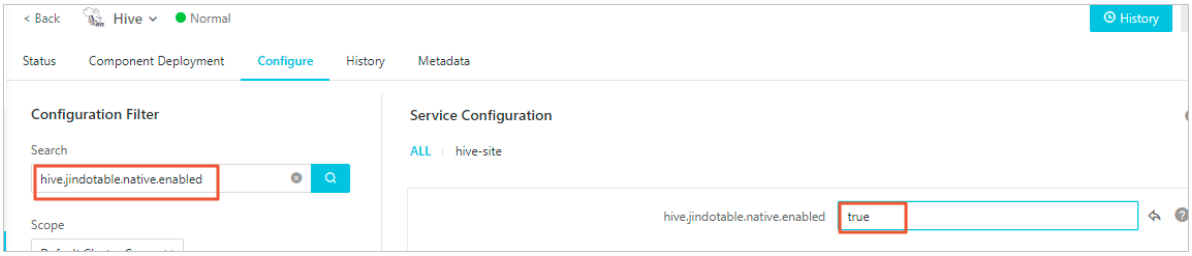
Improve the performance of Hive

> **Notice** If you have high requirements for job stability, we recommend that you do not enable query acceleration.

You can use one of the following methods to improve the performance of Hive:

- Use the EMR console

  On the **Configure** tab of the Hive service page, search for the **hive.jindotable.native.enabled** parameter and change its value to **true**. Then, save the configuration and restart the Hive service. This method is suitable for Hive on MapReduce and Hive on Tez jobs.



- Use the CLI

  Set `hive.jindotable.native.enabled` to `true` in the CLI to enable query acceleration. By default, the query acceleration plug-in for Parquet files is deployed in JindoTable in EMR V3.35.0 and later. You can directly set this parameter in EMR V3.35.0 and later.

```
set hive.jindotable.native.enabled=true;
```

> **Note** When you use this feature in Hive, complex data types such as MAP, STRUCT, and ARRAY are not supported.

4.4.2. Use JindoTable

JindoTable is used to implement tiered storage, optimize table files, and collect data statistics based on the popularity of tables or partitions. This topic describes how to use JindoTable.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your on-premises machine.
- An EMR cluster of the 3.30.0 version or later is created. For more information about how to create a cluster, see Create a cluster.

Use JindoTable

Common commands:

- -accessStat
- -cache
- -archive
- -unarchive
- -uncache
- -status
- -optimize

- -showTable
- -showPartition
- -listTables
- -dumpmc

> **Notice** Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

-accessStat

- Syntax

  jindo table -accessStat {-d} *<days>* **{-n}** <topNums>

- Description

  This command is used to query the access records in which tables or partitions are visited most in a specified time range.

  <days> and <topNums> must be positive integers. If <days> is 1, all the access records generated from 00:00 (local time) on the current day to the current time are queried.

- Example: Query the 20 access records of tables or partitions that are most frequently visited within the last seven days.

```
jindo table -accessStat -d 7 -n 20
```

-cache

- Syntax

  jindo table -cache {-t} <dbName.tableName> **[-p]** <partitionSpec> **[-pin]**

- Description

  This command is used to cache data of specified tables or partitions to local disks.

  The data of the tables or partitions must be stored in Object Storage Service (OSS) or JindoFileSystem (JindoFS). Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`. When you specify `-pin`, if the cache space is insufficient, do not delete related data if possible.

- Example: Cache data of the db1.t1 table that is generated on March 16, 2020 to local disks.

```
jindo table -cache -t db1.t1 -p date=2020-03-16
```

-uncache

- Syntax

  jindo table -uncache {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to delete cached data of specified tables or partitions from local disks.

  The data of the tables or partitions must be stored in OSS or JindoFS. Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

- Examples:
  - Delete cached data of the db1.t2 table from local disks.

    ```
    jindo table -uncache -t db1.t2
    ```

  - Delete cached data of the db1.t1 table from local disks.

    ```
    jindo table -uncache -t db1.t1 -p date=2020-03-16,category=1
    ```

-archive

- Syntax

  jindo table -archive {-a|i} {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to lower the level of the policy that is used to store data of specified tables or partitions. By default, the Archive storage class is used.

  To use the Infrequent Access (IA) storage class, add -i to the command. Specify tables in the format of database.table. Specify partitions in the format of 'partitionCol1=1,partitionCol2=2,...'.

- Example: Cache data of the db1.t1 table to local disks.

  ```
  jindo table -archive -t db1.t1 -p date=2020-10-12
  ```

-unarchive

- Syntax

  jindo table -unarchive [-o|-i] {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to change the storage class from Archive to Standard.

  If `-o` is added to the command, an Archived object is temporarily restored. If `-i` is added to the command, an Archived object is changed to an IA object.

- Example:

  ```
  jindo table -unarchive -o -t db1.t1 -p date=2020-03-16,category=1
  ```

  ```
  jindo table -unarchive -i -t db1.t2
  ```

-status

- Syntax

  jindo table -status {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to view the data storage status of specified tables or partitions.

- Examples:
  - View the data storage status of the db1.t2 table.

    ```
    jindo table -status -t db1.t2
    ```

- View the data storage status of the db1.t1 table on March 16, 2020.

```
jindo table -status -t db1.t1 -p date=2020-03-16
```

-optimize

- Syntax

  jindo table -optimize {-t} <dbName.tableName>

- Description

  This command is used to optimize the data organization of tables at the storage layer.

- Example: Optimize the data organization of the db1.t1 table at the storage layer.

```
jindo table -optimize -t db1.t1
```

-showTable

- Syntax

  jindo table -showTable {-t} <dbName.tableName>

- Description

  This command is used to display all partitions in a partitioned table or display the data storage of a non-partitioned table.

- Example: Display all partitions in the db1.t1 partitioned table.

```
jindo table -showTable -t db1.t1
```

-showPartition

- Syntax

  jindo table -showPartition {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to display the data storage of partitions.

- Example: Display the data storage of all partitions in the db1.t1 partitioned table on October 12, 2020.

```
jindo table -showPartition -t db1.t1 -p date=2020-10-12
```

-listTables

- Syntax

  jindo table -listTables [-db] <dbName>

- Description

  This command is used to display all tables in a specified database. If you do not specify `[-db]`, tables in the default database are displayed.

- Examples:

- Display tables in the default database.

```
jindo table -listTables
```

- Display tables in the db1 database.

```
jindo table -listTables -db db1
```

-dumpmc

- Syntax

jindo table -dumpmc {-i} <accessId> **{-k}** <accessKey> **{-m}** <numMaps> **{-t}** <tunnelUrl> **{-project}** <projectName> **{-table}** <tablename> **{-p}** <partitionSpec> **{-f}** <csv|tfrecord> **{-o}** <outputPath>

Parameter	Description	Required
-i	The AccessKey ID of your Alibaba Cloud account.	Yes
-k	The AccessKey secret of your Alibaba Cloud account.	Yes
-m	The number of map tasks.	Yes
-t	The VPC Tunnel endpoint of MaxCompute.	Yes
-project	The name of the MaxCompute project.	Yes
-table	The name of the MaxCompute table.	Yes
-p	The partition information. Example: `pt=xxx`. Separate multiple partitions with commas (,), such as `pt=xxx,dt=xxx`.	No
-f	The file format. Valid values: ○ tfrecord ○ csv	Yes
-o	The destination path.	Yes

- Description

  This command is used to dump MaxCompute tables to an EMR cluster or OSS. The formats CSV and TFRECORD are supported.

- Examples:

- Dump a MaxCompute table in the TFRECORD format to an EMR cluster.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o /tmp/outputtf1 -f tfrecord
```

- Dump a MaxCompute table in the CSV format to OSS.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o oss://bucket1/tmp/outputcsv -f csv
```

4.4.3. Archive and unarchive data in SDK mode

JindoTable allows you to run the archiveTable and unarchiveTable commands in SDK mode to archive and unarchive data in Object Storage Service (OSS). The commands do not rely on the Jindo Namespace Service component of SmartData. This topic describes how to use the archiveTable and unarchiveTable commands.

Prerequisites

- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.
- The partitioned table or non-partitioned table that you want to archive is stored in OSS. Only table data can be archived.

Context

You can use the original archive and unarchive commands of JindoTable to archive or unarchive tables or partitions in OSS. However, these commands rely on the Jindo Namespace Service component of SmartData. The new commands archiveTable and unarchiveTable do not rely on the Jindo Namespace Service component.

The archiveTable and unarchiveTable commands have the following advantages over the archive and unarchive commands:

- You can run the archiveTable and unarchiveTable commands even if the SmartData service is not deployed in your cluster. For example, you can run the commands on a self-managed cluster.
- You can configure filter parameters in the archiveTable or unarchiveTable command to archive or unarchive a large number of partitions on multiple threads at the same time. If local multithreading cannot meet your business requirements, you can run MapReduce tasks on the entire cluster to archive or unarchive data.

For more information about the original archive and unarchive commands, see Use JindoTable.

Limits

The archiveTable and unarchiveTable commands are supported only in EMR V3.36.0 and later minor versions, and EMR V5.2.0 and later minor versions.

archiveTable

You can use the archiveTable command to archive tables or partitions in OSS.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to obtain help information:

```
jindo table -help archiveTable
```

archiveTable syntax:

```
-archiveTable -t <dbName.tableName> \
-i/-a/-ca \
[-c "<condition>" | -fullTable] \
[-b/-before <before days>] \
[-p/-parallel <parallelism>] \
[-mr/-mapReduce] \
[-e/-explain] \
[-w/-workingDir <working directory>] \
[-l/-logDir <log directory>]
```

Parameter	Description	Required
-t <dbName.tableName>	The name of the table that you want to archive. You must configure this parameter in the `Database name.Table name` format. Separate the database name and table name with a period (.). The table can be a partitioned table or a non-partitioned table.	Yes
-i/-a/-ca	The storage class in which you want to archive data. You can use one of the following options to specify a storage class: ○ -i: Infrequent Access (IA) ○ -a: Archive ○ -ca: Cold Archive If you use the -i option in the command, the files whose storage class is Archive or Cold Archive are skipped. If you use the -a option in the command, the files whose storage class is Cold Archive are skipped.	Yes
-c "<condition>" | -fullTable	You must specify either `-fullTable` or `-c "<condition>"`. ○ If you specify `-fullTable`, the entire partitioned or non-partitioned table is archived. ○ If you specify `-c "<condition>"`, only the partitions that meet the filter condition are archived. Common operators, such as greater-than signs (>), are supported. For example, if the partition key column is the ds column whose data type is String and you want to archive partitions whose partition names are greater than 'd', use `-c " ds > 'd' "`.	Yes
-b/before <before days>	Only the tables or partitions that were created at least the specified days ago can be archived.	No
-p/-parallel <parallelism>	The parallelism of archiving operations.	No

Parameter	Description	Required
-mr/-mapReduce	Hadoop MapReduce instead of local multithreading is used to archive data.	No
-e/-explain	The explain mode is used. In explain mode, the list of partitions to be archived is displayed, but no data is archived.	No
-w/-workingDir	The working directory of a MapReduce job. This option is used only when you use a MapReduce job to archive data. You must have read and write permissions on the directory. The directory can be empty or not. Temporary files are created when you run the MapReduce job and are automatically deleted after the job is completed.	No
-l/-logDir <log directory>	The directory in which log files are stored.	No

unarchiveTable

The syntax of the unarchiveTable command is similar to the syntax of the archiveTable command. You can use the unarchiveTable command to unarchive tables or partitions in OSS.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to obtain help information:

   ```
   jindo table -help unarchiveTable
   ```

   unarchiveTable syntax:

   ```
   -archiveTable -t <dbName.tableName> \
   -i/-a/-o/-cr \
   [-notWait] \
   [-c "<condition>" | -fullTable] \
   [-b/-before <before days>] \
   [-p/-parallel <parallelism>] \
   [-mr/-mapReduce] \
   [-e/-explain] \
   [-w/-workingDir <working directory>] \
   [-l/-logDir <log directory>]
   ```

The parameters of the unarchiveTable command differ from the parameters of the archiveTable command in the following aspects:

- The optional parameter -i/-a/-o/-cr is used instead of the required parameter -i/-a/-ca.
- The optional parameter -notWait is used.

Parameter	Description	Required

Parameter	Description	Required
-i/-a/-o/-cr	The conversion of the storage class for data in the Cold Archive storage class. • If you do not specify -i/-a/-o/-cr, the storage class of the data is changed from Cold Archive to Standard. Note The storage class of the data can be changed from Cold Archive to Standard, IA, or Archive only when the data is completely unarchived. • If you specify -i/-a/-o/-cr, the following descriptions prevail: ○ If you specify the -i option, the storage class of the data is changed from Cold Archive to IA. Files whose storage class is Standard are skipped. ○ If you specify the -a option, the storage class of the data is changed from Cold Archive to Archive. Files whose storage class is Standard or IA are skipped. ○ If you specify the -o option, data is only temporarily unarchived and its storage class is retained. Files whose storage class is Standard or IA are skipped. Files that are previously unarchived are also skipped. This way, these files are not repeatedly unarchived. ○ You can specify the -cr option to check whether the files stored to Archive or Cold Archive storage are unarchived.	No
-notWait	This parameter is valid only when you unarchive data. This parameter is used to unarchive data that is stored to Cold Archive storage. If you specify this parameter, the system only sends an unarchive command, but does not wait for the completion of the unarchiving operation.	No

4.4.4. Use the MoveTo command of JindoTable

JindoTable allows you to run the MoveTo command to migrate data in tables or partitions. This topic describes how to use the MoveTo command.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

The MoveTo command can automatically update metadata after the command copies the underlying data. This way, data in a table or partitions can be fully migrated to the destination path. You can configure filter conditions for the MoveTo command to migrate a large number of partitions at the same time. JindoTable also provides some protective measures to ensure data integrity and security when the MoveTo command is used to migrate data.

Limits

The MoveTo command is supported only in EMR V3.36.0 and later minor versions, and in EMR V5.2.0 and later minor versions.

Use the MoveTo command

> **Notice** Only one MoveTo process can be run at a time in an EMR cluster. If you attempt to start a MoveTo process on an EMR cluster on which another MoveTo process is running, your request is rejected because the configuration lock is unavailable. A message that contains information about the running MoveTo process is also displayed. In this case, you can terminate the running MoveTo process and start a new MoveTo process, or wait for the running MoveTo process to end.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to obtain help information:

   ```
   jindo table -help moveTo
   ```

   Information similar to the following output is returned:

```
<dbName.tableName>      The table to move.
<destination path>      The destination base directory which is always at the
                          same level of a 'table location', where the moved
                          partitions or un-partitioned data would located in.
<condition>/-fullTable  A filter condition to determine which partitions should
                          be moved, supporting common operators (like '>') and
                          built-in UDFs (like to_date) (UDFs not supported
                          yet...), while -fullTable means that all partitions (or
                          a whole un-partitioned table) should be moved. One but
                          only one option must be specified among -c
                          "<condition>" and -fullTable.
<before days>           Optional, saying that table/partitions should be moved
                          only when they are created (not updated or modified)
                          more than some days before from now.
<parallelism>           The maximum concurrency when copying partitions, 1 by
                          default.
        <OSS storage policy>: Storage policy for OSS destination, which can be Standard
  (by default), IA, Archive, or ColdArchive. Not applicable for destinations other
  than OSS. NOTE: if you are willing to use ColdArchive storage policy, please
  make sure that Cold Archive has been enabled for your OSS bucket.
-o/-overWrite     Overwriting the final paths where the data would be moved.
                    For partitioned tables this overwrites partitions' locations
                    which are subdirectories of <destination path>; for
                    un-partitioned table this overwrites the <destination path>
                    itself.
-r/-removeSource  Let the source data be removed when the corresponding
                    table/partition is successfully moved to the new destination.
                    Otherwise (by default), the source data would be left as it
                    was.
-skipTrash        Applicable only when [-r/-removeSource] is enabled. If
                    present, source data would be immediately deleted from the
                    file system, bypassing the trash.
-e/-explain       If present, the command would not really move data, but only
                    prints the table/partitions that would be moved for given
                    conditions.
<log directory>   A directory to locate log files, '/tmp/<current user>/' by
                    default.
```

MoveTo syntax:

```
jindo table -moveTo \
  -t <dbName.tableName> \
  -d <destination path> \
  [-c "<condition>" | -fullTable] \
  [-b/-before <before days>] \
  [-p/-parallel <parallelism>] \
  [-s/-storagePolicy <OSS storage policy>] \
  [-o/-overWrite] \
  [-r/-removeSource] \
  [-skipTrash] \
  [-e/-explain] \
  [-l/-logDir <log directory>]
```

Parameter	Description	Required
-t <dbName.tableName>	The name of the table that you want to migrate. You must specify this parameter in the `Database name.Table name` format. Separate the database name and table name with a period (.). The table can be a partitioned table or a non-partitioned table.	Yes
-d <destination path>	The destination path. No matter whether you want to migrate a specific partition or an entire non-partitioned table, this parameter specifies a table-level path. If you want to migrate a partition, the complete path of the partition is composed of the value of this parameter and the name of the partition, such as `<destination path>/p1=v1/p2=v2/` .	Yes
-c "<condition>" | -fullTable	You must specify either `-c "<condition>"` or `-fullTable` . ○ If you specify `-fullTable` , the entire partitioned or non-partitioned table is archived. ○ If you specify `-c "<condition>"` , only the partitions that meet the filter condition are archived. Common operators, such as greater-than signs (>), are supported. For example, if the partition key column is the ds column whose data type is String and you want to archive partitions whose partition names are greater than 'd', use `-c " ds > 'd' "` .	No
-b/before <before days>	Only the tables or partitions that were created at least the specified days ago can be migrated.	No
-p/-parallel <parallelism>	The parallelism among migration operations.	No

Parameter	Description	Required
-s/-storagePolicy <OSS storage policy>	The storage class that you want to use after data is migrated to Object Storage Service (OSS). Valid values: ○ Standard ○ IA ○ Archive ○ ColdArchive **Note** Make sure that the storage class you want to use is enabled on the destination OSS bucket.	No
-o/-overWrite	The destination path is forcibly cleared. For a partitioned table, only the destination path of the partition that you want to migrate is cleared.	No
-r/-removeSource	After data is migrated and metadata is updated, the source path is cleared. For a partitioned table, only the source path of the partition that is migrated is cleared.	No
-skipTrash	The trash is skipped when the source path is cleared. **Note** You can specify this option only if -r/-removeSource is specified.	No
-e/-explain	The explain mode is used. In explain mode, the list of partitions to be migrated is displayed, but no data is migrated.	No
-l/-logDir <log directory>	The directory in which log files are stored.	No

Configure a lock directory

The MoveTo command supports process locks. You must use an HDFS path to store lock files. The default path is *hdfs:///tmp/jindotable-lock/*.

Notice The path must be an HDFS path. If you have no permission on the default path, you can perform the following steps to customize a path.

1. Go to the HDFS service page.
   i.

ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.

iii. Click the **Cluster Management** tab.

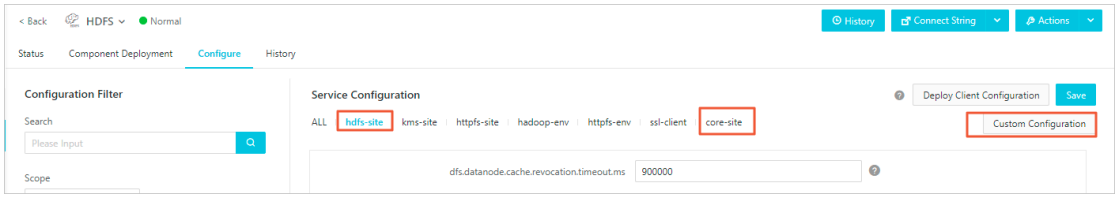
iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.

v. In the left-side navigation pane of the Cluster Overview page, choose **Cluster Service > HDFS**.

2. Add a custom item.

   i. Click the **Configure** tab. Then, click the **hdfs-site** or **core-site** tab in the Service Configuration section.

   ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**.



   iii. In the **Add Configuration Item** dialog box, add the **jindotable.moveto.tablelock.base.dir** parameter and set it to an existing HDFS path.

> **Notice** When you customize a lock directory, make sure that no MoveTo process is running on the nodes of the cluster. Otherwise, the MoveTo process may fail, which may even cause data pollution.

3. Save the configuration.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and click **OK**.

4.4.5. Use JindoTable to collect access frequency statistics on tables and partitions

You can use JindoTable to collect access frequency statistics on tables and partitions, and separate cold and hot data based on these statistics. This helps you reduce storage costs and improve cache usage efficiency.

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

JindoTable can be used to collect access frequency statistics on Hive tables. The collected data is saved in the namespaces of the SmartData service of a cluster.

In SmartData 3.2.X and later, JindoTable can be used to collect access frequency statistics on Spark, Hive, and Presto tables. This feature is enabled for Spark and Presto by default. If you want to disable the feature for Spark or Presto, perform the operations described in Disable the collection of access frequency statistics. The feature is disabled for Hive by default. If you want to enable the feature for Hive, perform the operations described in Enable the collection of access frequency statistics for Hive.

Query access frequency statistics

You can run a command to query access frequency statistics.

- Syntax

```
jindo table -accessStat <-d [days]> <-n [topNums]>
```

Set `days` and `topNums` to positive integers. For example, if days is set to 1 and topNums is not specified, the access frequency statistics of all the tables or partitions that are visited on the current day (from 00:00 to the current time) are queried.

- Description

This command is used to query the first N tables or partitions that are visited the most frequently within a specific period of time, and the number of times each of them is visited.

- For example, you can run the following command to query the first 20 tables or partitions that have been visited the most frequently within the last seven days, and the number of times each of them has been visited:

```
jindo table -accessStat -d 7 -n 20
```

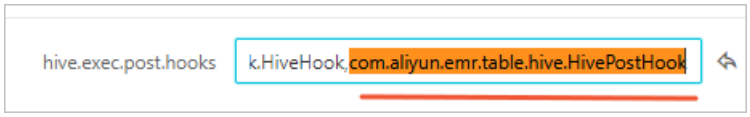
For more information about how to use JindoTable, see Use JindoTable.

Enable the collection of access frequency statistics for Hive

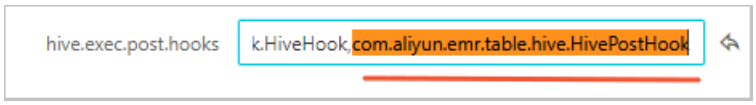
1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of a Hive parameter.
   i. In the left-side navigation pane, choose **Cluster Service > Hive**.
   ii. On the Hive service page, click the **Configure** tab.
   iii. Search for the hive.exec.post.hooks parameter and add **com.aliyun.emr.table.hive.HivePostHook** to the end of the existing parameter value. Separate the hooks with a comma (,).
6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   i. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
   ii. In the **Cluster Activities** dialog box, specify **Description**.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Disable the collection of access frequency statistics

1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of the related parameter.
   - Hive:
     a. In the left-side navigation pane, choose **Cluster Service > Hive**.
     b. On the Hive service page, click the **Configure** tab.
     c. Search for the hive.exec.post.hooks parameter and delete **com.aliyun.emr.table.hive.HivePostHook** from the parameter value.



   - Spark:
     a. In the left-side navigation pane, choose **Cluster Service > Spark**.
     b. On the Spark service page, click the **Configure** tab.
     c. Search for the spark.sql.queryExecutionListeners parameter and delete **com.aliyun.emr.table.spark.SparkSQLQueryListener** from the parameter value.



   - Presto:
     a. In the left-side navigation pane, choose **Cluster Service > Presto**.
     b. On the Presto service page, click the **Configure** tab.
     c. Search for the event-listener.name parameter and delete the parameter value.
6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   - Hive:
     a. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
     b. In the **Cluster Activities** dialog box, specify **Description**.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.
   - Spark:

a. In the upper-right corner of the Spark service page, choose **Actions > Restart ThriftServer**.
b. In the **Cluster Activities** dialog box, specify **Description**.
c. Click **OK**.
d. In the **Confirm** message, click **OK**.

- Presto:
  a. In the upper-right corner of the Presto service page, choose **Actions > Restart All Components**.
  b. In the **Cluster Activities** dialog box, specify **Description**.
  c. Click **OK**.
  d. In the **Confirm** message, click **OK**.

4.4.6. Use JindoTable to collect infrequent-access statistics of tables and partitions

JindoTable can be used to collect infrequent-access statistics of tables and partitions. It enables you to find the data that is not accessed recently based on the last access time of tables and partitions. This helps optimize data storage and reduce costs. For example, in data analytics scenarios, you can use this feature to move infrequently used partitions to lower-cost storage media.

Prerequisites

A cluster is created in E-MapReduce (EMR) V3.35.0 or later V3.X.X, or in EMR V4.9.0 or later V4.X.X. For more information, see Create a cluster.

Context

In SmartData 3.5.X and later, JindoTable can be used to collect infrequent-access statistics of Hive, Spark, and Presto tables. This feature is disabled by default. For more information about how to enable this feature, see Enable the collection of infrequent-access statistics for Spark, Enable the collection of infrequent-access statistics for Hive, and Enable the collection of infrequent-access statistics for Presto.

> **Note** The same hooks and listeners are used to collect frequent-access and infrequent-access statistics. Therefore, if you enable the collection of infrequent-access statistics, the collection of frequent-access statistics is also enabled. For more information about the collection of frequent-access statistics, see Use JindoTable to collect access frequency statistics on tables and partitions.

Limits

- Data Lake Formation (DLF) metadata is not supported.
- You must have the permissions to access the underlying metadata storage, such as MySQL or ApsaraDB RDS, from the IP addresses of the nodes on which the Hive CLI, HiveServer2, Spark SQL CLI, Spark Thrift Server, and Presto services reside.
- You can use JindoTable to collect infrequent-access statistics only of Hive, Spark, and Presto tables.

Query infrequent-access statistics

You can run a command to query infrequent-access statistics.

- Syntax

```
jindo table -leastUseStat -n <num> [-i/-ignoreNever]
```

num indicates the number of tables or partitions that you want to query. Set this parameter to a positive integer. -i/-ignoreNever is an optional parameter. If you specify this parameter, the tables or partitions that have never been accessed are filtered out.

- Description

This command is used to query the tables or partitions that have not been accessed for the longest time.

- Example: Query the first 20 tables or partitions that have not been accessed for the longest time.

```
jindo table -leastUseStat -n 20
```

The following figure shows the output.

```
tdb.t1      pid=20/qid=ten      2021-03-26 13:53:52
tdb.t2                          2021-03-26 13:53:58
```

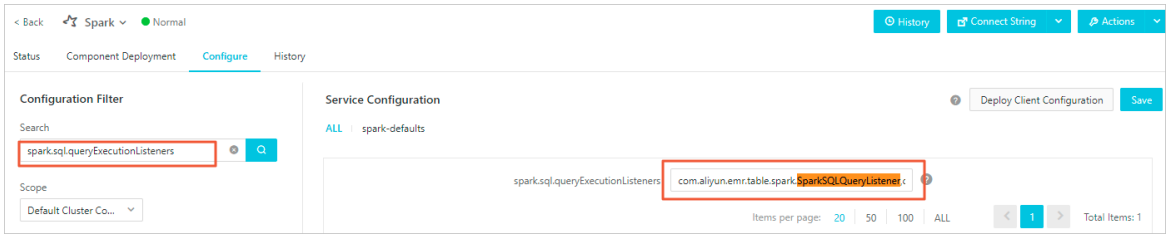
  - The first column lists tables in the Database name.Table name format.
  - The second column lists partitions in the First partition key column=Column value/Second partition key column=Column value/... format. For a non-partitioned table, this column is empty.
  - The third column displays the last access time of the tables or partitions in the yyyy-MM-dd HH:mm:ss format.

> **Note** For a partitioned table, only partition-level access statistics are displayed. Table-level access statistics are not displayed.

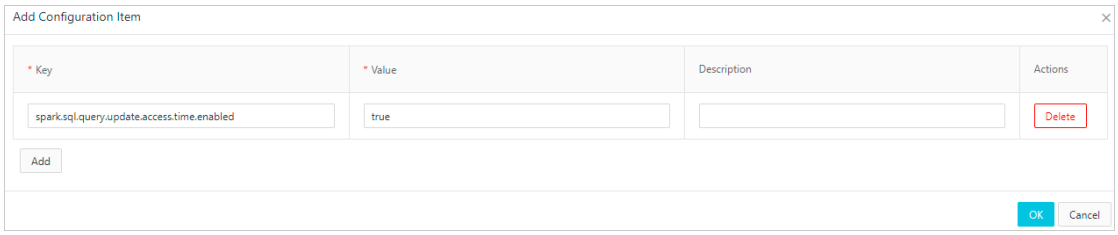
For more information about how to use JindoTable, see Use JindoTable.

Enable the collection of infrequent-access statistics for Spark

1. Go to the Spark service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Spark**.
2. On the Spark service page, click the **Configure** tab.
3. Search for the spark.sql.queryExecutionListeners parameter and check whether the value contains com.aliyun.emr.table.spark.SparkSQLQueryListener. If the value does not contain this listener, add the listener to the value. Separate multiple listeners with comma(,).



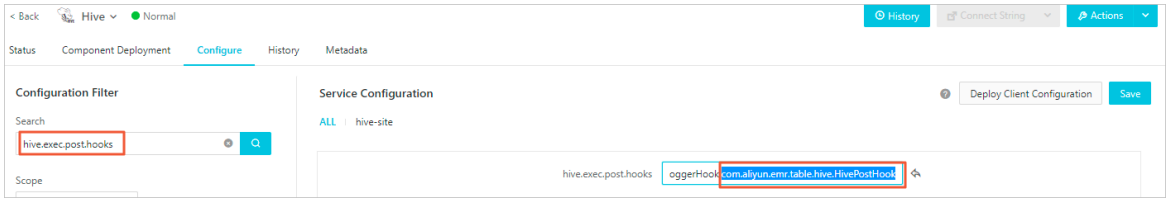
4. Add custom configurations.
   i. In the **Service Configuration** section, click the **spark-defaults** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **spark.sql.query.update.access.time.enabled** parameter and set it to **true**.



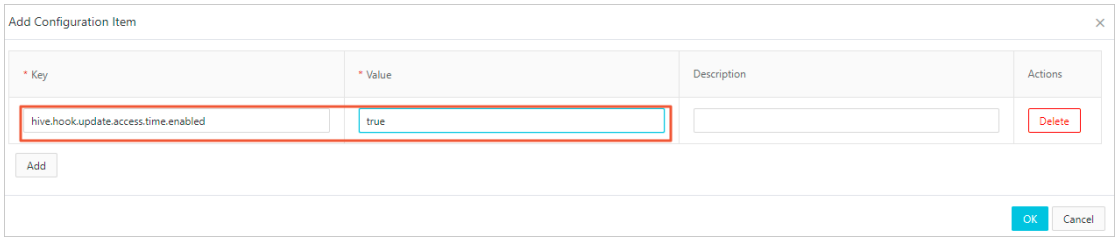
   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

Enable the collection of infrequent-access statistics for Hive

1. Go to the Hive service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Hive**.
2. On the Hive service page, click the **Configure** tab.
3. Search for the hive.exec.post.hooks parameter and check whether the value contains com.aliyun.emr.table.hive.HivePostHook. If the value does not contain this hook, add the hook to the value. Separate multiple hooks with comma(,).



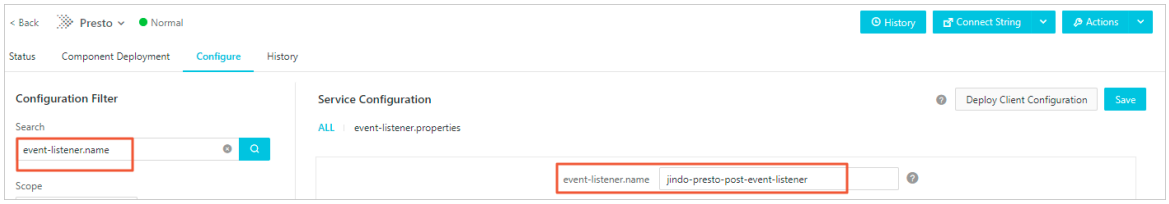
4. Add custom configurations.
   i. In the **Service Configuration** section, click the **hive-site** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **hive.hook.update.access.time.enabled** parameter and set it to **true**.



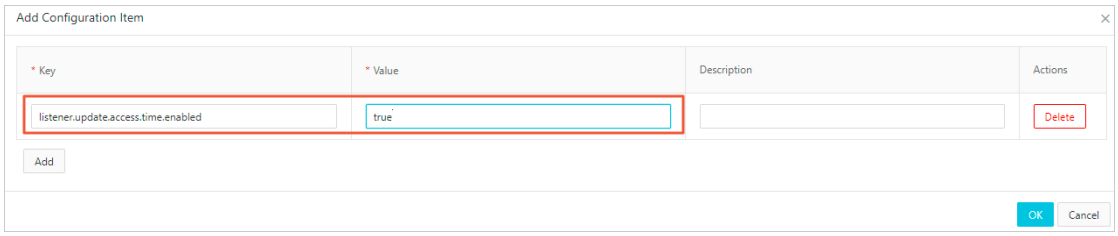
   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

Enable the collection of infrequent-access statistics for Presto

1. Go to the Presto service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Presto**.
2. On the Presto service page, click the **Configure** tab.
3. Search for the event-listener.name parameter and check whether the value contains jindo-presto-post-event-listener. If the value does not contain this listener, add the listener to the value.



4. Add custom configurations.
   i. In the **Service Configuration** section, click the **event-listener.properties** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **listener.update.access.time.enabled** parameter and set it to **true**.



   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

4.5. Toolset

4.5.1. Use Jindo SQL

Jindo SQL is a tool provided by JindoFS. You can use Jindo SQL to analyze metadata, JindoFS access logs, and Object Storage Service (OSS) access logs. This topic describes how to use Jindo SQL to analyze metadata, JindoFS access logs, and OSS access logs.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

You can use Jindo SQL to perform the following operations:

- Analyze JindoFS access logs
- Analyze metadata
- Analyze OSS access logs

Limits

Jindo SQL is supported only in EMR V3.36.0 and later minor versions, and EMR V5.2.0 and later minor versions.

Start Jindo SQL

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to start Jindo SQL:

```
jindo sql
```

The following table describes the common parameters that can be used in the jindo sql command.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically initializes the SQL script after the jindo sql command is run.
-d	Specifies parameters in key-value pairs, such as `-d A=B`.

Structures of built-in tables in Jindo SQL

- audit_log_source (partitioned table)

  The audit_log_source table is used to store raw access logs of JindoFS.

Parameter	Description
datetime	The time format is yyyy-MM-dd HH:mm:ss.
allowed	Indicates whether the current operation is allowed. Valid values: ○ true: The operation is allowed. ○ false: The operation is not allowed.
ugi	The user who performs the operation. Information about the authentication method is provided.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dst	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Parameter	Description
date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

- audit_log

  The audit_log table is used to store JindoFS access logs. You can use the partition key column to filter data in the table.

Parameter	Description
datetime	The time format is yyyy-MM-dd HH:mm:ss.
allowed	Indicates whether the current operation is allowed. Valid values: ○ true: The operation is allowed. ○ false: The operation is not allowed.
ugi	The user who performs the operation. Information about the authentication method is provided.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dst	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.
date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

- fs_image (partitioned table)

  The fs_image table is used to dump image information.

Parameter	Description
atime	The time when the current inode is last accessed.
attr	File-related attributes.
etag	The ETag value of OSS.

Parameter	Description
id	The ID of the inode.
mtime	The time when the inode is modified.
name	The name of the inode.
owner	The name of the owner.
ownerGroup	The name of the group to which the owner belongs.
parentId	The ID of the parent node.
permission	The operation permissions on the file.
size	The size of the inode.
state	The status of the inode.
type	The type of the inode.
storagePolicy	The storage policy.
namespace (partition key column)	The name of the namespace.
datetime (partition key column)	The time when the information is dumped.

- oss_access_log_source

  If the table partitioning mode is enabled, the oss_access_log_source table is a partitioned table. This table is used to store raw access logs of OSS.

Parameter	Description
line	A raw log entry.
bucket (partition key column)	The name of the bucket.
partition_date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

- oss_access_log

  If the table partitioning mode is enabled, you can use the partition key columns to filter data in the oss_access_log table. This table is used to store OSS access logs.

Parameter	Description
Remote_IP	The IP address of the requester.
Reserved	A reserved field. Fixed value: -.
Reserved1	A reserved field. Fixed value: -.

Parameter	Description
Time	The time when OSS receives the request.
Request_URI	The request URL. It contains a query string. OSS ignores the query string parameter that starts with x-. However, this parameter is recorded in access logs. Therefore, you can use the query string parameter that starts with x- as the tag of a request. Then, you can use this tag to query the log that corresponds to the request.
HTTP_Status	The HTTP status code that is returned by OSS.
SentBytes	The downstream traffic generated by the request. Unit: bytes.
RequestTime	The time required to process the request. Unit: milliseconds.
Referer	The HTTP referer of the request.
User_Agent	The user-agent header in the HTTP request.
HostName	The destination domain name in the request.
Request_ID	The ID of the request.
LoggingFlag	Indicates whether log dump is enabled.
Requester	The user ID of the requester. A value of - indicates anonymous access.
Operation	The request type.
Bucket	The name of the destination bucket.
Key	The name of the destination object.
ObjectSize	The size of the destination object. Unit: bytes.
Server_Cost_Time	The time required by OSS to process the request. Unit: milliseconds.
ErrorCode	The error code that is returned by OSS. A value of - indicates that no error code is returned.
RequestLength	The size of the request. Unit: bytes.
UserID	The ID of the bucket owner.
Delta_DataSize	The variation of the bucket size. A value of - indicates that this request does not involve write operations on objects.

Parameter	Description
SyncRequest	Indicates whether the request is an Alibaba Cloud CDN back-to-origin request. Valid values: ○ cdn: The request is a CDN back-to-origin request. ○ -: The request is not a CDN back-to-origin request.
StorageClass	The storage class of the destination object. Valid values: ○ Standard ○ IA ○ Archive ○ Cold Archive ○ -: no storage class obtained
TargetStorageClass	The storage class to which the object is converted after a lifecycle rule is triggered or the CopyObject operation is called. Valid values: ○ Standard ○ IA ○ Archive ○ Cold Archive ○ -: no storage class conversion involved
TransmissionAccelerationAccessPoint	The accelerate endpoint that is used to access the destination bucket based on the transfer acceleration feature. A value of - indicates that no accelerate endpoint is used or the accelerate endpoint is in the same region as the destination bucket. For example, if the requester accesses the destination bucket by using the accelerate endpoint that corresponds to the China (Hangzhou) region, cn-hangzhou is used as the value of this parameter.
AccessKeyID	The AccessKey ID that is used to access the bucket.
bucket (partition key column)	The name of the bucket.
partition_date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

Analyze JindoFS access logs

JindoFS allows you to use SQL queries to analyze the JindoFS access logs that are stored in OSS. You can use SQL queries to analyze the most active commands or IP addresses in the related tables. You can use the `jindo sql` command to analyze the data.

> **Note** The AuditLog feature is enabled. For more information, see Use Jindo AuditLog.

You can use the following commands related to Jindo SQL:

- Run the following command to query tables:

```
show tables;
```

> **Note** For more information about the table structures, see Structures of built-in tables in Jindo SQL.

The information shown in the following figure is returned.

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to query partitions:

```
show partitions audit_log_source;
```

Information similar to the output in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

Information similar to the output in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted] =
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted] =
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted] =
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

Information similar to the output in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics about the use frequencies of different commands that were run on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

Analyze metadata

JindoFS allows you to use SQL queries to analyze the metadata files that are stored in JindoFS. You can use SQL queries to analyze the related tables. You can use the `jindo sql` command to analyze the metadata files.

> **Note** The AuditLog feature is enabled. For more information, see Use Jindo AuditLog.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to start Jindo SQL:

   ```
   jindo sql
   ```

3. Query the tables that can be analyzed by using Jindo SQL.
   - Run the `show tables` command to query the tables that can be analyzed. Jindo SQL provides two built-in modules: audit_log and fs_image, which are used for auditing and metadata analysis.

     The following figure shows an example.

     ```
     [root@emr-header-1 ~]# jindo sql
     Spark master: yarn, Application Id: application_1603081647416_0050
     jindo-sql> show tables;
     database        tableName       isTemporary
     default audit_log       false
     default audit_log_source        false
     default fs_image        false
     Time taken: 0.33 seconds, Fetched 3 row(s)
     ```

   - Run the `show partitions fs_image` command to view the partition information of the

fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

4. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287249    1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       15464684820176650002    1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The following figure shows an example on how to obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

> **Note** The namespace and datetime columns are added. The namespace column contains the namespace name, and the datetime column contains the timestamp when metadata is uploaded.

Analyze OSS access logs

> **Notice** To analyze OSS access logs, you must specify a log directory and whether to enable the table partitioning mode. If you enable the table partitioning mode, OSS archives logs in the partitioned table by bucket or partition_date, and you can execute statements to query logs in a specific partition. This improves query efficiency. If you enable the partitioning table mode, you must use this mode each time you query data in the table. Otherwise, some files may be archived to the specified log directory, which causes the failure to query some data.

JindoFS allows you to use SQL queries to analyze the OSS access log files that are stored in OSS. You can use SQL queries to analyze the related tables. You can run the `jindo sql` command to analyze the log files.

> **Note** Make sure that log dump is enabled. For more information, see Logging.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to start Jindo SQL:

   ```
   jindo sql
   ```

3. Run the following command to specify the storage path of the OSS access logs and whether the table partitioning mode is enabled.

   ```
   jindo sql -d access_log_path=oss://test-sh/oss-accesslog -d partition.table.enabled=tru
   e
   ```

   `access_log_path` specifies the storage path, and `partition.table.enabled` specifies whether the table partitioning mode is enabled.

FAQ

- Q: How do I modify the startup parameters of Jindo SQL?

  `Jindo SQL` is developed based on Spark. Therefore, the initial resources of Jindo SQL may be insufficient. If you want to increase the resources of Jindo SQL, you can modify the startup parameters of Jindo SQL by using the environment variable `JINDO_SPARK_OPTS`.

  ```
  export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
  --conf spark.executor.cores=5 --conf spark.executor.memory=20G"
  ```

- Q: How do I use Hive to analyze tables?

  Tables in the default database are invisible to Hive jobs. This prevents Hive metadata from being polluted. If you want to use Hive to analyze the tables, you can execute the `show create table {table_name}` statement to query the statement that is executed to create a specific table or execute an SQL statement to create a table. In addition, you must use Hive to load external tables.

4.5.2. Use JindoFS FUSE

This topic describes how to access JindoFileSystem (JindoFS) from a Filesystem in Userspace (FUSE) client. FUSE allows you to use JindoFS in block storage mode and cache mode (only JFS Scheme supported).

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

FUSE is a method provided by the Linux kernel to mount a file system. You can use a FUSE client to map files in a JindoFS cluster to a local disk. Then, you can directly access data in the JindoFS cluster without the need to use the `hadoop fs -ls jfs://<namespace>/` command.

Mount a directory

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs
   ```

3. Run the following command to mount the directory:

   ```
   jindofs-fuse /mnt/jfs
   ```

   In this example, the */mnt/jfs/* directory is mounted.

Read and write files

1. Run the following command to query all the sub-directories under the /mnt/jfs/ directory:

   ```
   ls /mnt/jfs/
   ```

   All the namespaces that you have configured on the server are returned.

   ```
   test  testcache
   ```

2. Run the following command to query all the files in the test namespace:

   ```
   ls /mnt/jfs/test/
   ```

3. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs/test/dir1
   ls /mnt/jfs/test/
   ```

4. Run the following command to write data into a file:

   ```
   echo "hello world" > /tmp/hello.txt
   cp /tmp/hello.txt /mnt/jfs/test/dir1/
   ```

5. Run the following command to read data from a file:

   ```
   cat /mnt/jfs/test/dir1/hello.txt
   ```

   The following information is returned:

   ```
   hello world
   ```

The following example shows how to read data from and write data into files in Python:

1. Write data into a file in Python. Create the *write.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'w',encoding = 'utf-8') as f:
   f.write("my first file\n")
   f.write("This file\n\n")
   f.write("contains three lines\n")
```

2. Read data from a file in Python. Create the *read.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'r',encoding = 'utf-8') as f:
    lines = f.readlines()
    [print(x, end = '') for x in lines]
```

Run the following command to read the data that is written into the *test.txt* file:

```
[hadoop@emr-header-1 ~]$ ./read.py
```

The following information is returned:

```
my first file
This file
```

Unmount FUSE

Note Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to unmount FUSE:

```
umount jindofs-fuse
```

If the `target is busy` error is returned, switch to another directory, stop all the programs that are reading data from and writing data into FUSE files, and then unmount FUSE.

4.5.3. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain help information:

```
jindo distcp --help
```

The following information is returned:

```
     --help           - Print help text
     --src=VALUE          - Directory to copy files from
     --dest=VALUE              - Directory to copy files to
     --parallelism=VALUE         - Copy task parallelism
     --outputManifest=VALUE       - The name of the manifest file
     --previousManifest=VALUE   -   The path to an existing manifest file
     --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
if specified
     --copyFromManifest   -   Copy from a manifest instead of listing a directory
     --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
     --srcPattern=VALUE   -   Include only source files matching this pattern
     --deleteOnSuccess   -   Delete input files after a successful copy
     --outputCodec=VALUE   -   Compression codec for output files
     --groupBy=VALUE   -   Pattern to group input files by
     --targetSize=VALUE   -   Target size for output files
     --enableBalancePlan   -   Enable plan copy task to make balance
     --enableDynamicPlan   -   Enable plan copy task dynamically
     --enableTransaction   -   Enable transation on Job explicitly
     --diff   -   show the difference between src and dest filelist
     --ossKey=VALUE   -   Specify your oss key if needed
     --ossSecret=VALUE   -   Specify your oss secret if needed
     --ossEndPoint=VALUE   -   Specify your oss endPoint if needed
     --policy=VALUE   -   Specify your oss storage policy
     --cleanUpPending   -   clean up the incomplete upload when distcp job finish
     --queue=VALUE   -   Specify yarn queuename if needed
     --bandwidth=VALUE   -   Specify bandwidth per map/reduce in MB if needed
     --s3Key=VALUE   -   Specify your s3 key
     --s3Secret=VALUE   -   Specify your s3 Sercet
     --s3EndPoint=VALUE   -   Specify your s3 EndPoint
```

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-
01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03
/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03
/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017
-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-0
1/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-0
1/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_tab
le/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/
03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/
03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromMani
fest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

Note You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

> **Note** If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

--queue

--queue specifies the name of the YARN queue where the current DistCp task resides.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<your_bucket>/hourly_table --qu
eue yarnqueue
```

--bandwidth

--bandwidth specifies the bandwidth allocated to the current DistCp task. This prevents the task from using excessive bandwidth. Unit: MB/s.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.c
om --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy archive --parallelism 20
  ```

- Example of the IA storage class (`--ia`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy ia --parallelism 20
  ```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

Use Amazon S3 as a data source

You can use the --s3Key, --s3Secret, and --s3EndPoint parameters in a command to specify information related to Amazon S3.

Example:

```
jindo distcp jindo-distcp-2.7.3.jar --src s3a://yourbucket/ --dest oss://<your_bucket>/hour
ly_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com
```

You can configure the s3Key, s3Secret, and s3EndPoint parameters in the *core-site.xml* file of Hadoop. This way, you do not need to specify an AccessKey pair each time.

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

Example:

```
jindo distcp /tmp/jindo-distcp-2.7.3.jar --src s3://smartdata1/ --dest s3://smartdata1/tmp
--s3EndPoint  s3-us-west-1.amazonaws.com
```

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

> **Note** If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

4.5.4. Use Jindo DistCp in specific scenarios

This topic describes how to use Jindo DistCp in specific scenarios.

Prerequisites

- An E-MapReduce (EMR) cluster of a required version is created. For more information, see Create a cluster.
- Java Development Kit (JDK) 1.8 is installed.
- *jindo-distcp-<version>.jar* is downloaded based on your Hadoop version.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 2.7 and later 2.X versions.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 3.X.

Use scenarios

Common use scenarios of Jindo DistCp:

- Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?
- Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?
- Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?
- Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?
- Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file size?
- Scenario 8: What parameters are required if Amazon S3 is used as a data source?
- Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?
- Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?
- Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?
- Scenario 12: What parameters are required if I want to delete original files after a copy operation?
- Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?

If you are not using EMR, the following conditions must be met before you can import HDFS data to Object Storage Service (OSS):

- You can read data from HDFS.
- The AccessKey ID, AccessKey secret, and endpoint of OSS are obtained. You can write data to a destination OSS bucket.
- The storage class of the OSS bucket is not Archive.
- You can submit MapReduce tasks.
- The JAR package of Jindo DistCp is downloaded.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 10
```

> **Note** For more information about the parameters, see Use Jindo DistCp.

To import a large amount of data or a large number of files, such as millions or tens of millions of files, to OSS, you can set parallelism to a large value to increase the concurrency. You can also set `enableBatch` to true for optimization. Optimization command:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 500 --enableBatch
```

Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?

You can use one of the following methods to verify data integrity:

- Jindo DistCp Counters

  Check DistCp Counters in the counter information of a MapReduce task.

  ```
  Distcp Counters
          Bytes Destination Copied=11010048000
          Bytes Source Read=11010048000
          Files Copied=1001
  Shuffle Errors
          BAD_ID=0
          CONNECTION=0
          IO_ERROR=0
          WRONG_LENGTH=0
          WRONG_MAP=0
          WRONG_REDUCE=0
  ```

where:

- Bytes Destination Copied specifies the size of the file that you copy to a destination directory, in bytes.
- Bytes Source Read specifies the size of the file that you read from the source directory, in bytes.
- Files Copied specifies the number of copied files.

- Jindo DistCp --diff

  Use `--diff` to compare the information of files in the source and destination directories. The information includes file name and file size. Missing files and the files that fail to be copied are recorded in a manifest file. The manifest file is generated in the directory where you perform the comparison.

  Add `--diff` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --diff
```

  If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?

If your DistCp task fails, and you want to perform resumable upload to copy only the files that fail to be copied, perform the following operations on the basis of the command described in Scenario 1:

1. Add `--diff` to check whether all files are copied.

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --diff
```

   If all the files are copied, the following information is returned. Otherwise, a manifest file is generated. In this case, go to the next step.

```
INFO distcp.JindoDistCp: distcp has been done completely.
```

2. Use `--copyFromManifest` and `--previousManifest` to copy the remaining files. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --dest oss://yang-hhht/hourly_table --previousManifest=file:///opt
/manifest-2020-04-17.gz --copyFromManifest --parallelism 20
```

   `file:///opt/manifest-2020-04-17.gz` is the path where the manifest file is stored.

Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?

1. If no information is generated for the files that were copied last time, specify parameters to

generate a manifest file to record the information about the files that are copied.

Add `--outputManifest=manifest-2020-04-17.gz` and `--requirePreviousManifest=false` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

where:

- `--outputManifest` generates a manifest file whose name can be customized and must have a file name extension of gz, such as *manifest-2020-04-17.gz*. This file is stored in the directory specified by `--dest`.
- `--requirePreviousManifest` specifies whether to require a previous manifest file.

2. After a DistCp task is complete, copy the new files that may be generated in the source directory.

   Add `--outputManifest=manifest-2020-04-17.gz` and `--previousManifest=oss://yang-hhht/hourly_table/manifest-2020-04-17.gz` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

```
hadoop jar jindo-distcp-2.7.3.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --outputManifest=manifest-2020-04-18.gz --previousManifest=oss://yang-hhht
/hourly_table/manifest-2020-04-17.gz --parallelism 10
```

3. Repeat Step 2 to continuously copy incremental files.

Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?

Add the following two parameters to the command described in Scenario 1. These two parameters can be used together or separately.

- `--queue`: the name of the YARN queue.
- `--bandwidth`: the size of the specified bandwidth , in MB.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --queue yarnqueue --bandwidth 6 --parallelism 10
```

Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?

- To write data to OSS Archive storage, add `--archive` to the command described in Scenario 1.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --archive --parallelism 20
```

- To write data to OSS Infrequent Access (IA) storage, add `--ia` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --ia --parallelism 20
```

Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file sizes?

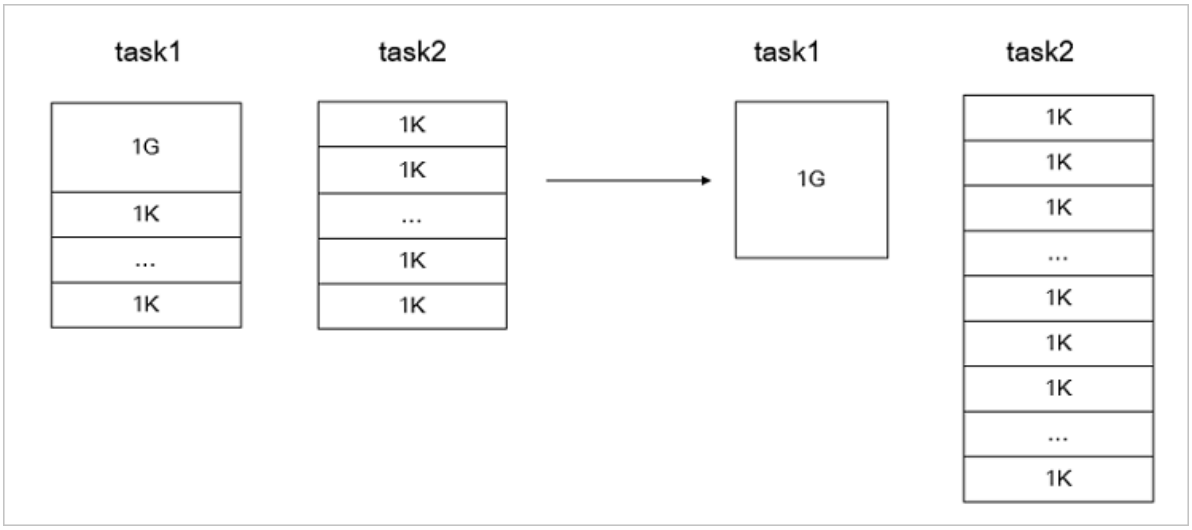
- A large number of small files and large data volume for a single large file

  If most of the files that you want to copy are small files, and the data volume of a single large file is large, the general solution is to randomly allocate the files to be copied. In this case, if you do not optimize the job allocation plan, few large files and many small files may be allocated to the same copy process. This cannot achieve excellent copy performance.

  Add `--enableDynamicPlan` to the command described in Scenario 1 to enable the optimization. --enableDynamicPlan cannot be used with `--enableBalancePlan`. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableDynamicPlan --parallelism 10
```

  The following figure shows the data copy before and after the plan is optimized.

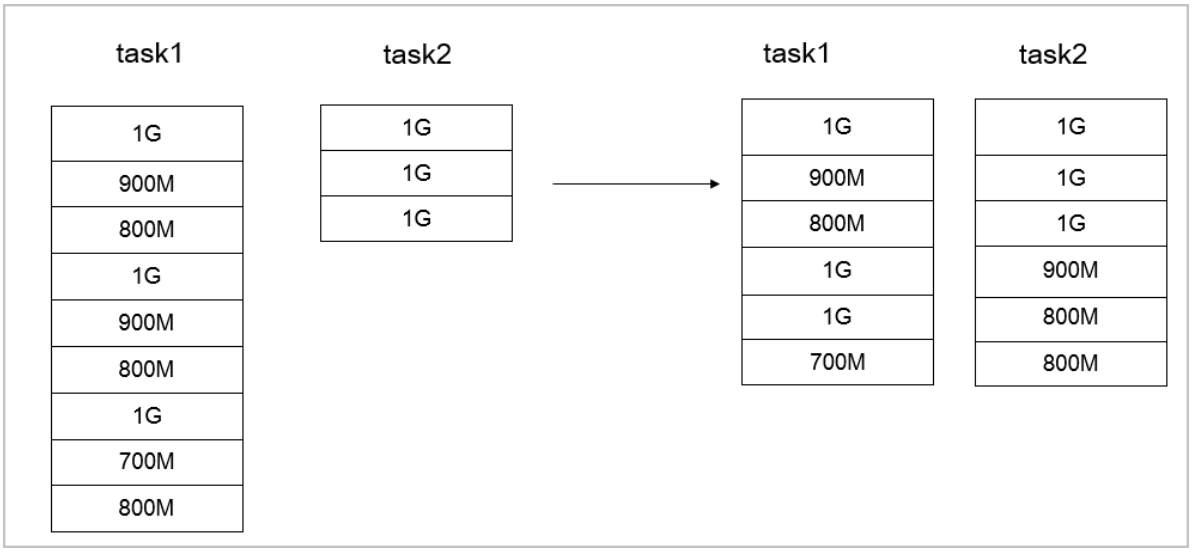


- No significant differences in sizes of files

  If the sizes are not significantly different among the files that you want to copy, use `--enableBalancePlan` to optimize the job allocation plan. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableBalancePlan --parallelism 10
```

The following figure shows the data copy before and after the plan is optimized.



Scenario 8: What parameters are required if Amazon S3 is used as a data source?

Replace the parameters that specify the AccessKey ID, AccessKey secret, and endpoint of OSS in the command described in Scenario 1 with the following parameters of Amazon S3:

- `--s3Key` : the AccessKey ID for Amazon S3
- `--s3Secret` : the AccessKey secret for Amazon S3
- `--s3EndPoint` : the endpoint for Amazon S3

Example:

```
hadoop jar jindo-distcp-<version>.jar --src s3a://yourbucket/ --dest oss://yang-hhht/hourly
_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com --para
llelism 10
```

Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?

You can use `--outputCodec` to compress copied files in a format such as LZO or GZ to reduce the space that is used to store the files.

Add `--outputCodec` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --outputCodec=gz --parallelism 10
```

You can set --outputCodec to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files are compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change to the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?

- If you want to copy the files that meet specific rules, add `--srcPattern` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPattern .*\.log --parallelism 10
```

  `--srcPattern` : specifies a regular expression that filters files for the copy operation.

- If you want to copy the files stored in some sub-directories of the same parent directory, add `--srcPrefixesFile` to the command described in Scenario 1.

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

  `--srcPrefixesFile` : enables Jindo DistCp to copy files in multiple folders under the same parent directory at a time.

  Content of the *folders.txt* file:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?

Add the following parameters to the command described in Scenario 1:

- `--targetSize` : the maximum size of the merged file, in MB
- `--groupBy` : the merging rule, which is a regular expression

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Scenario 12: What parameters are required if I want to delete original files after a copy operation?

Add `--deleteOnSuccess` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --deleteOnSuccess --parallelism 10
```

Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Jindo DistCp allows you to save the AccessKey ID, AccessKey secret, and endpoint of OSS in the *core-site.xml* file so that you do not need to repeatedly specify the information.

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of OSS, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.jfs.cache.oss-accessKeyId</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-accessKeySecret</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-endpoint</name>
        <value>oss-cn-xxx.aliyuncs.com</value>
    </property>
</configuration>
```

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of Amazon S3, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

4.5.5. Use tiered-storage commands of JindoFS

In E-MapReduce (EMR) V3.30, JindoFS provides a tiered storage feature. This feature allows you to store cold and hot data in different storage media. This helps reduce the data storage costs and accelerate data access.

Use jindo jfs

Run the following command to obtain help information:

```
jindo jfs -help archive
```

Tiered-storage commands in JindoFS are asynchronous and start only related tasks.

Common commands:

- Cache
- Uncache
- Archive
- Unarchive
- Status
- ls2

Cache

You can use this command to back up data stored in a specific path to local disks. Then, you can read the data from local disks, without the need to read data from Object Storage Service (OSS).

```
jindo jfs -cache -p <path>
```

-p can be used to ensure that local data is not cleared based on disk usage.

Uncache

You can use this command to delete backup data from local disks and store data only to OSS Standard storage.

```
jindo jfs -uncache  <path>
```

Archive

You can use this command to delete backup data from local disks and store data to OSS Infrequent Access (IA) or Archive storage. For information about the storage classes, see Overview.

```
jindo jfs -archive -i|-a|-c <path>
```

- If you specify the -i option, data is stored to OSS IA storage.
- If you specify the -a option, data is stored to OSS Archive storage.
- If you specify the -c option, data is stored to OSS Cold Archive storage.

Unarchive

You can use this command to convert the storage class of data from Archive to IA or Standard. You can temporarily unarchive data that is stored to Archive storage to allow the data to be readable for one day.

```
jindo jfs -unarchive -i/-o <path>
```

By default, this command can be used to store data to OSS Standard storage.

- If you specify the -i option, data is stored to OSS IA storage.
- If you specify the -o option, data stored to Archive storage can be temporarily unarchived and becomes readable.

Status

You can use this command to view task details. By default, the number of files for which you want to use tiered storage in a specific directory and the data to which tiered storage has been applied are measured.

```
jindo jfs -status -detail/-sync <path>
```

- If you specify the -detail option, the storage progress of file data can be viewed.
- If you specify the -sync option, the command exits only after a tiered-storage task is completed.

ls2

JindoFS provides the ls2 command that allows you to view the file storage status on the basis of Hadoop ls commands.

```
hadoop fs -ls2 <path>
```

Example of command output, which includes the file storage class:

```
drwxrwxrwx  - -         0    2020-06-05 04:27 oss://xxxx/warehouse
-rw-rw-rw-  1 Archive   1484 2020-09-23 16:40 oss://xxxx/wikipedia_data.csv
-rw-rw-rw-  1 Standard  1676 2020-06-07 20:04 oss://xxxx/wikipedia_data.json
```

5.SmartData 3.6.x

5.1. SmartData 3.6.X overview

SmartData is a storage service for the E-MapReduce (EMR) Jindo engine. SmartData provides centralized storage, caching, and computing optimization for EMR computing engines. This service also provides additional features. SmartData consists of JindoFS, JindoTable, and related tools. This topic describes the updates in SmartData 3.6.X.

JindoFS

The following table describes the updates to JindoFS.

Feature	Description
Cross-cloud access to Amazon Simple Storage Service (S3) and other services that use the S3 protocol, and cache-based acceleration for access to these services	JindoFS supports the S3 protocol and can access Amazon S3 and other systems that use the S3 protocol. You can use the cache-based acceleration feature to accelerate access to these systems.
Cache-based acceleration for access to HDFS	You can use the cache-based acceleration feature to accelerate access to HDFS.
Transactional loading options MetaSync and Data Cache	Two transactional loading options MetaSync and Data Cache are added to JindoFS. This ensures the transactional properties of preloading tasks and also ensures that intermediate status is not displayed in a metadata view during the loading process.
Optimization of the cache preloading mechanism	• The mechanism of preloading the cache of metadata and data is optimized. This way, the preloading efficiency is improved. • The progress feedback feature is optimized. • Tasks on super-large directories are supported.

JindoSDK

The following table describes the updates to JindoSDK.

Feature	Description
Local caching	JindoSDK allows you to configure a local caching policy, which enables local data caching even if the SmartData service is not deployed. This speeds up access to Object Storage Service (OSS) data.

Feature	Description
Object Store API at the same level as the FileSystem API	The Object Store API is added to JindoSDK. It is at the same level as the existing FileSystem API. The Object Store API is better suited to storage systems such as OSS and makes access to these systems easier. The Object Store API also improves the efficiency of copy and rename operations.
Optimization of OSS server-side caching	An OSS access accelerator is provided. After you start the accelerator, you can customize the bandwidth for accessing OSS data based on the capacity of the accelerator.
OSS second-level domain	Access to OSS by using a second-level domain is supported. After you enable this feature, you can use a second-level domain or an IP address to access OSS in some special environments. This feature is disabled by default.

JindoTable

The following table describes the updates to JindoTable.

Feature	Description
Tiered storage and archiving of HDFS data in OSS	You can run the MoveTo command to migrate data in multiple tables or partitions to OSS at a time. The command also automatically updates metadata. You can specify filter conditions to filter partitions and configure storage policies to migrate data to OSS. You can archive a large volume of data that is already in OSS at a time.
Unarchiving and retrieval of data in OSS	You can run the unarchiveTable command to unarchive and retrieve a large volume of data that is archived in OSS.
Acceleration of the speed at which Presto queries files in the Parquet format	You can use a native engine provided by JindoTable to accelerate the speed at which Presto queries files in the Parquet format.
Acceleration of the speed at which Spark 3 queries files in the Parquet or ORC format	You can use a native engine provided by JindoTable to accelerate the speed at which Spark 3 queries files in the Parquet or ORC format.
Acceleration of the speed at which Spark or Presto queries data in HDFS	You can use a native engine provided by JindoTable to accelerate the speed at which Spark or Presto queries data in HDFS.
Analysis of OSS access logs	You can execute SQL statements to analyze OSS access logs.

JindoFuse

The following table describes the updates to JindoFuse.

Feature	Description
Training and online mounting scenarios	You can mount a file system to a specific JindoFS namespace or an OSS directory. You can specify an OSS directory in SDK mode.

5.2. JindoFS in block storage mode

5.2.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read and write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

Context

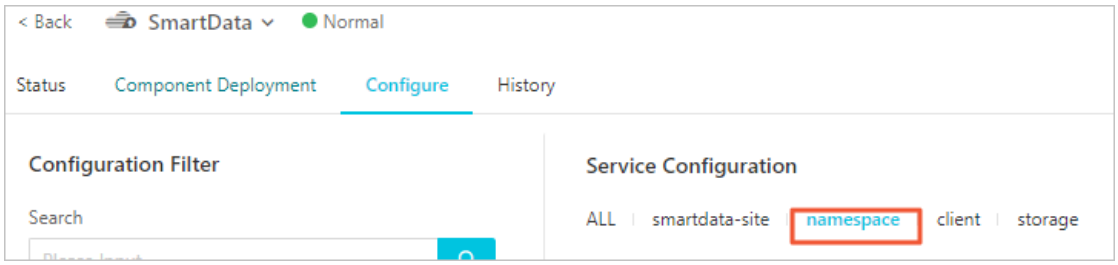
The block storage mode of JindoFS has the following features:

- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.
- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.

ii. Click the **namespace** tab in the Service Configuration section.



3. Configure required parameters.

JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

i. Set **jfs.namespaces** to **test**.

If you configure multiple namespaces, separate them with commas (,).

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block

Parameter	Description	Example
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

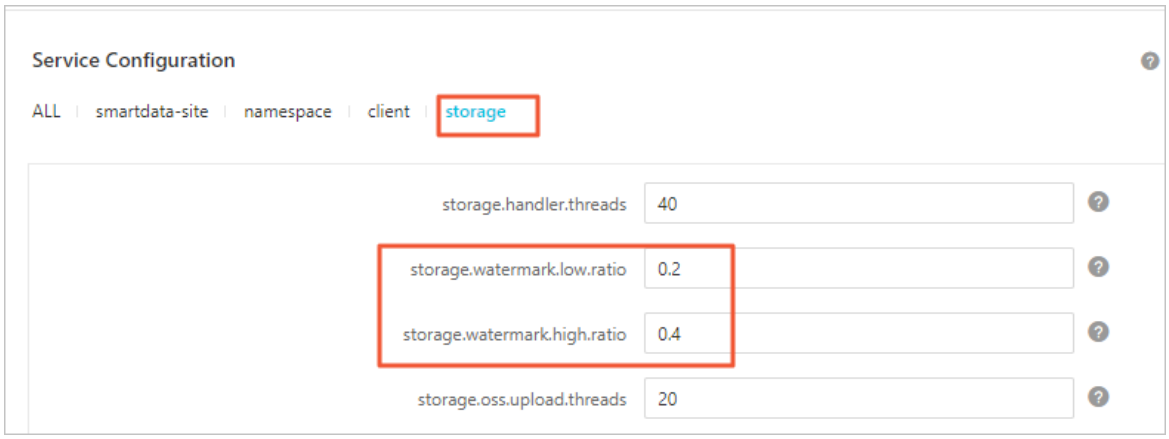
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

Note You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Choose **Actions > Restart Jindo Storage Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

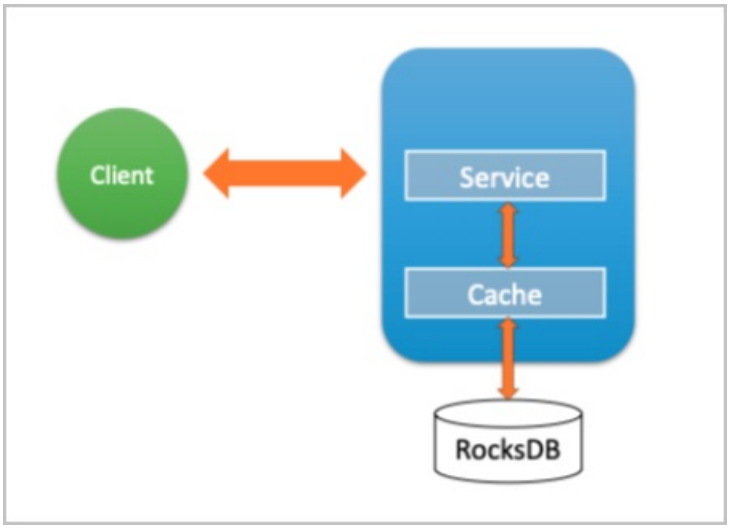
5.2.2. Use RocksDB to store metadata

Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

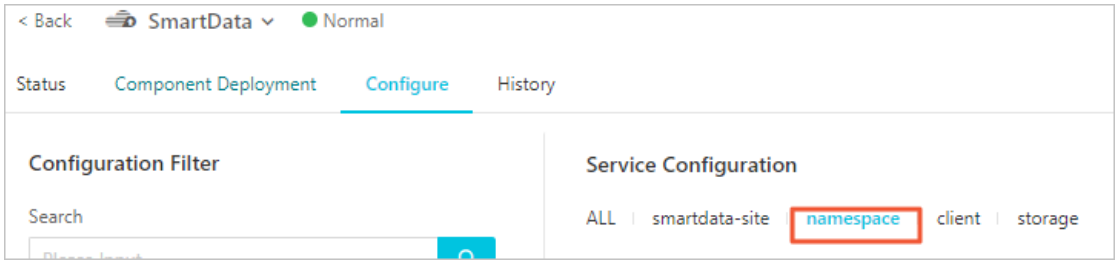
RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage backend in high availability mode, we recommend that you use Raft instances. For more information, see Use Raft-RocksDB-Tablestore to store metadata.

The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

6. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

7. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
8. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

      ```
      hadoop fs -count jfs://test/
      ```

      Information similar to the following example is returned:

      ```
       1596      1482809                25 jfs://test/
      ```

      The number of folders is 1596, and the number of files is 1,482,809.

   ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1`, the latest metadata is synchronized to the Tablestore instance.

      ```
      jindo jfs -metaStatus
      ```

      Information similar to that shown in the following figure is returned.

      ```
      [hadoop@emr-header-1 ~]$ jindo jfs -metaStatus
      ==== emr-header-1:8101 ====
      OtsUploader: _synced=1
      [RocksDB Row Counts of each CF(Table)]
      ```

   iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.
2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components.
3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus
```

If the state is FINISH, the recovery is completed, as shown in the following figure.

```
==== emr-header-1:8101 ====
[Recovery From OTS Status]
  state: FINISH
  total 22855 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

5.2.3. Use Raft-RocksDB-Tablestore to store metadata

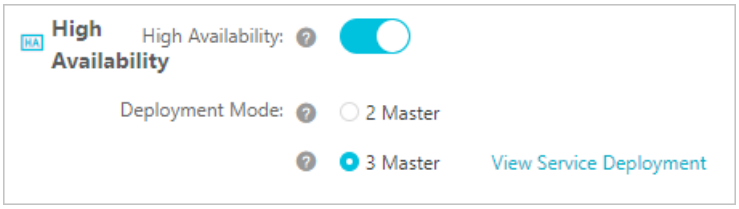
In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata.

Prerequisites

- A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  > **Note** You must enable the transaction feature.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.

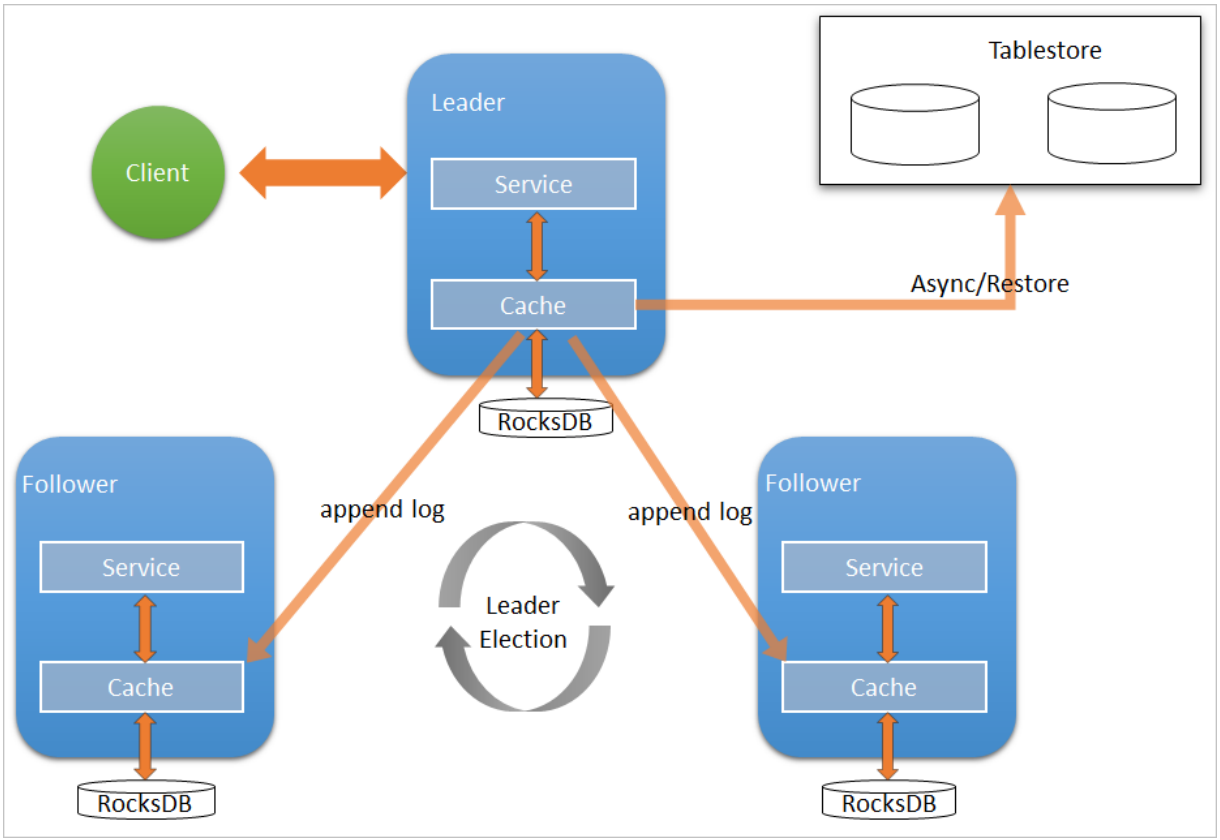


> **Note** If no deployment modes are available, submit a ticket.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.



Configure a Raft instance as the local storage backend

1. Suspend all the SmartData components of an EMR cluster.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   vi. Select **Stop All Components** from the **Actions** drop-down list in the upper-right corner.
2. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
3. Go to the **namespace** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **namespace** tab.

4. Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. Valid values: ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

Note If you do not need to use Tablestore for remote storage, skip Step 5.

5. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

```
hadoop fs -count jfs://test/
      1596      1482809                 25 jfs://test/
  (Number of folders) (Number of files)
```

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```

```
[RaftPeerImpl]
peer id: [illegible]
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [illegible]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 2335ms)
snapshot_timer: timeout(3600000ms) SCHEDULING(in 150305ms)
storage: [1, 624625]
disk_index: 624625
known_applied_index: 624625
last_log_id: (index=624625,term=2)
first_index_pinned: 624625
state_machine: Idle
last_committed_index: 624625
last_snapshot_index: 0
last_snapshot_term: 0
snapshot_status: IDLE
replicator_25769803789@[illegible] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=624261 ic=0
replicator_329853488332[illegible] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=623564 ic=0

OtsUploader: _lastStopIndex=624624, _synced=1
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.

```
[RaftPeerImpl]
peer_id: [redacted]:8103:0
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 3382ms)
snapshot_timer: timeout(600000ms) SCHEDULING(in 474855ms)
storage: [1, 153]
disk_index: 153
known_applied_index: 153
last_log_id: (index=153,term=2)
first_index_pinned: 1
state_machine: Idle
last_committed_index: 153
last_snapshot_index: 1
last_snapshot_term: 2
snapshot_status: IDLE
replicator_1116691496965@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0
replicator_3311419785217@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0

[Recovery From OTS Status]
state: FINISH
total rows: 1484409
table `jfs_block_test` 2 rows.
table `jfs_namespace_cache_ns` 1 rows.
table `jfs_namespace_test` 1484406 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

5.2.4. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see

Create a cluster.

- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.

ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

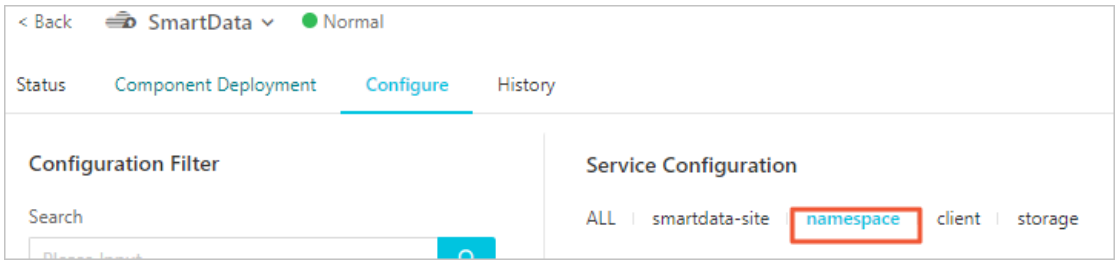
iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **namespace** tab for the SmartData service.

   i. Click the **Configure** tab.

   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:

   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

5.2.5. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

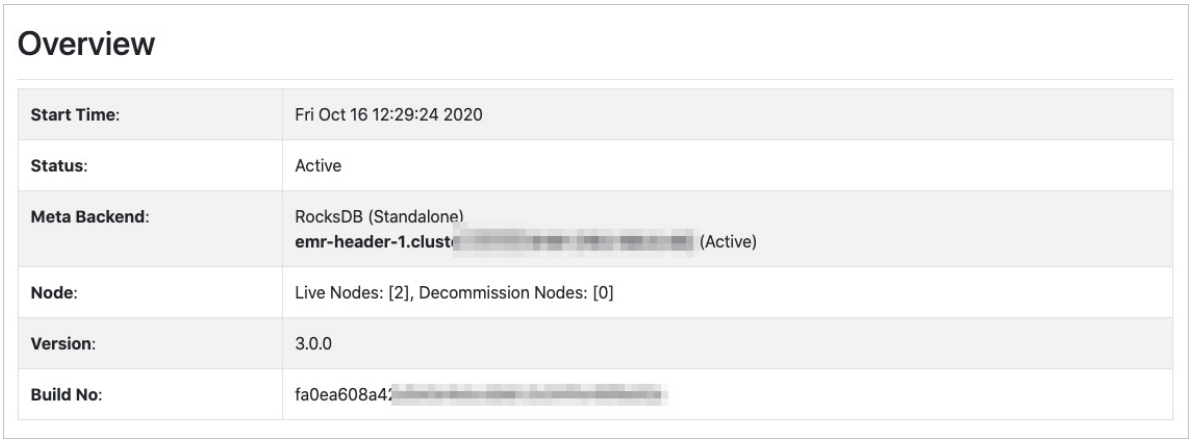
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

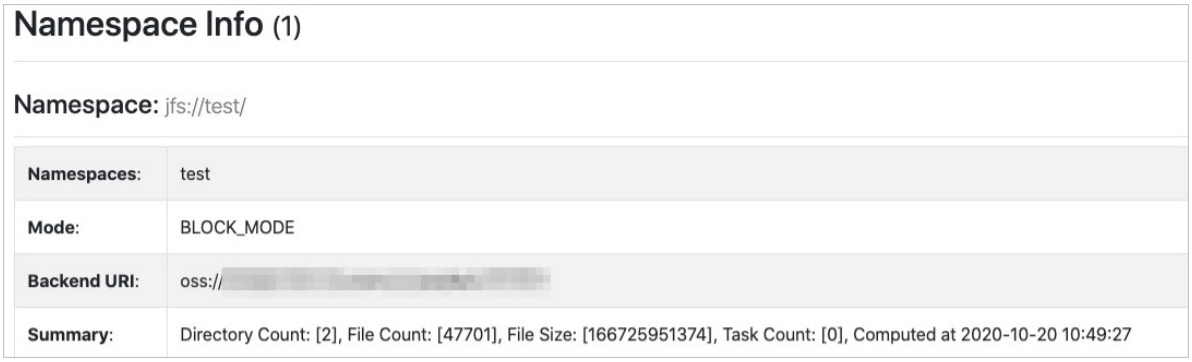


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



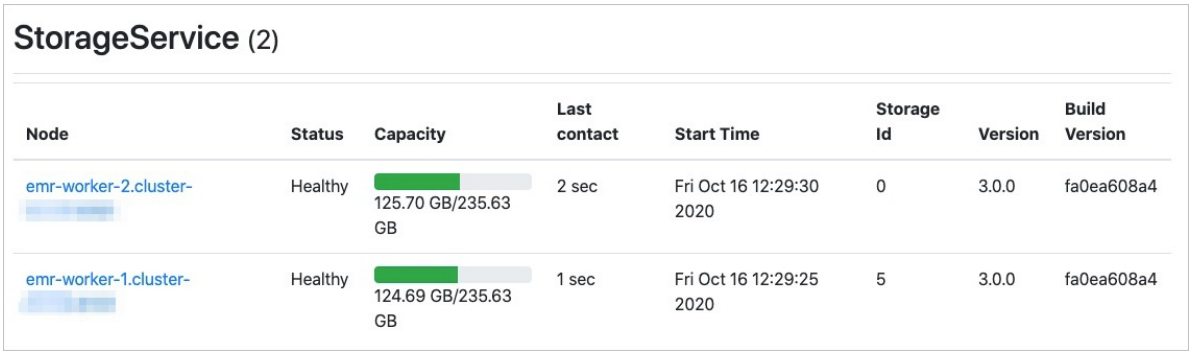
Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

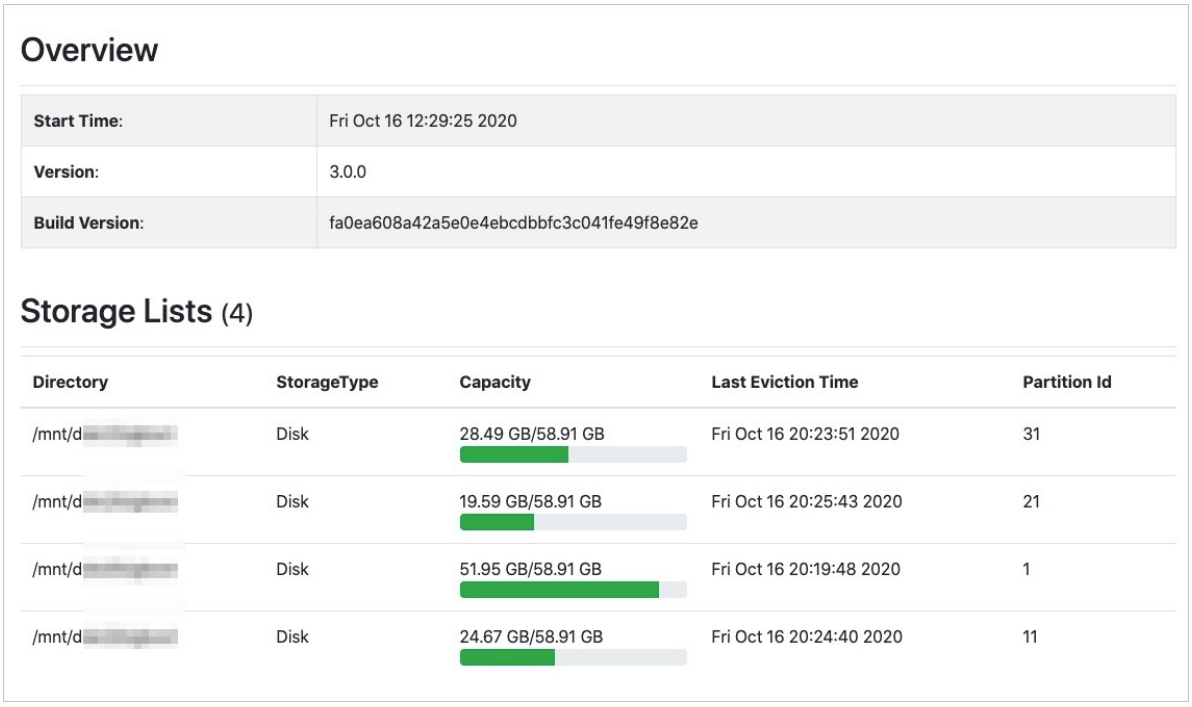
  This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.



StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.



Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

5.2.6. Manage JindoFS permissions

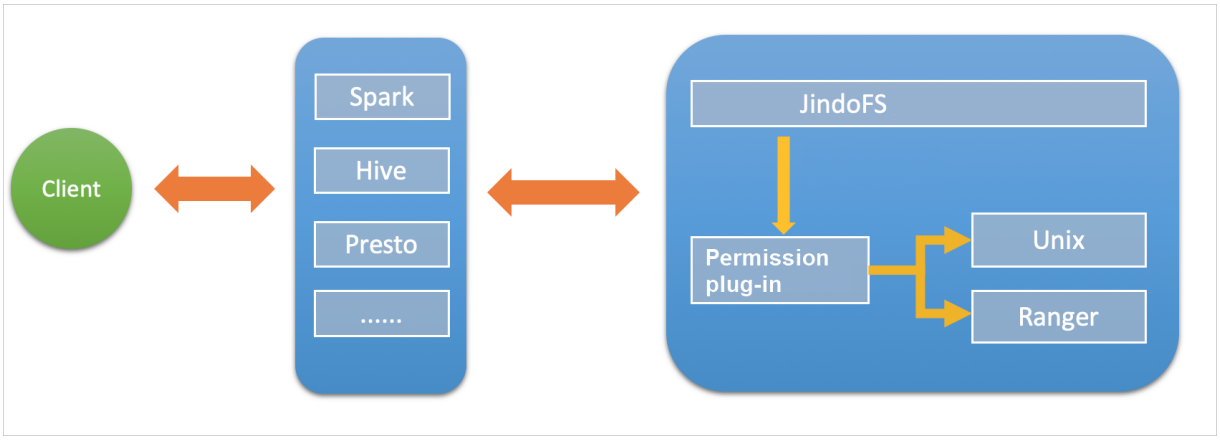
This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

Permission management based on the storage mode:

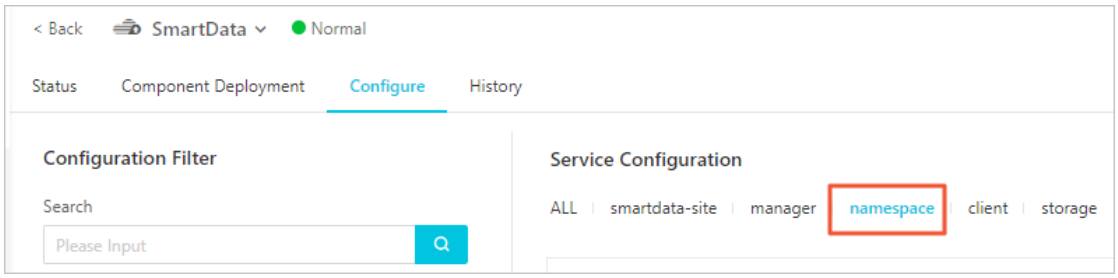
- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.
  - Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

5.2.7. Data management policies

JindoFS in block storage mode provides advanced data management policies to meet storage requirements in different scenarios. The advanced policies include storage policies and compression policies. This topic describes the policies in detail and provides examples to show how to use the policies.

Storage policies

JindoFS provides the five storage policies described in the following table to support flexible storage.

Policy	Description
AR	Data has only one backup. The backup is stored as an Archive object in OSS.
IA	Data has only one backup. The backup is stored as an Infrequent Access (IA) object in OSS.

Policy	Description
COLD	Data has only one backup. The backup is stored as a Standard object in OSS.
WARM	Data has one backup in OSS and one local backup. The local backup can effectively accelerate data read operations. By default, this storage policy is used.
HOT	Data has one backup in OSS and one local backup. The local backup is forcibly locked. This way, it is not deleted when the cache is automatically cleared. This policy can achieve a better acceleration effect than the WARM policy and is suitable for the hottest data.

For more information about the storage classes of OSS, see Overview.

A newly added file is stored based on the storage policy specified for its parent directory. Example:

- Run the following command to configure a storage policy:

```
jindo jfs -setStoragePolicy [-R] <StoragePolicy>(AR/IA/COLD/WARM/HOT) <path> ...
```

  where:

  - `[-R]`: specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.
  - `<path>`: the name of the directory for which the storage policy is configured.

- Run the following command to obtain the storage policy configured for a directory:

```
jindo jfs -getStoragePolicy <path>
```

Compression policies

JindoFS allows you to compress data blocks before you store them. This feature effectively reduces the storage space occupied by data blocks and improves data read and write efficiency. This feature is suitable for files that have a high compression ratio. The following table describes the supported compression policies.

Policy	Description
NONE	Data blocks are not compressed. By default, this compression policy is used.
ZSTD	The Zstandard compression algorithm is used to compress data blocks.

A newly added file is compressed based on the compression policy specified for its parent directory before it is stored. Example:

- Run the following command to configure a compression policy:

```
jindo jfs -setCompressionPolicy [-R] <CompressionPolicy>(NONE/ZSTD) <path> ...
```

where:

- `[-R]`: specifies that a recursive operation is performed to configure the same compression policy for all sub-directories of the directory.
- `<path>`: the name of the directory for which the compression policy is configured.

- Run the following command to obtain the compression policy configured for a directory:

```
jindo jfs -getCompressionPolicy <path> ...
```

5.2.8. Analyze metadata offline

In E-MapReduce (EMR) V3.30.0 or later, JindoFS in block storage mode allows you to dump the metadata of an entire namespace to Object Storage Service (OSS) and use Jindo SQL to analyze the metadata.

Context

In Hadoop Distributed File System (HDFS), the metadata is stored in a snapshot file named fsimage. The snapshot file contains the following information about metadata: namespace, file that stores the metadata, blocks, and file system quota. In HDFS, you can run a command to download the fsimage file in the XML format to your on-premises machine and view this file to analyze metadata offline. JindoFS does not require you to download metadata to your on-premises machine.

Upload HDFS metadata to OSS

Run the following command to upload metadata of a specific namespace to OSS:

```
jindo jfs -dumpMetadata <nsName>
```

`<nsName>` indicates the name of the namespace in block storage mode.

For example, upload the metadata of the test-block namespace to OSS and analyze the metadata offline.

```
jindo jfs -dumpMetadata test-block
```

If the following information appears, data is uploaded to OSS and stored as a file in the JSON format.

```
Sucessfully upload namespace metadata to OSS.
```

Metadata upload path

The metadata upload path is the metadataDump sub-directory under *sysinfo* in JindoFS.

For example, if `namespace.sysinfo.oss.uri` is set to `oss://abc/`, metadata is uploaded to the `oss://abc/metadataDump` directory.

Parameter	Description
namespace.sysinfo.oss.uri	The path of the OSS bucket.
namespace.sysinfo.oss.endpoint	The endpoint. An endpoint specified in a different region can also be used.
namespace.sysinfo.oss.access.key	The AccessKey ID of your Alibaba Cloud account.
namespace.sysinfo.oss.access.secret	The AccessKey secret of your Alibaba Cloud account.

Batch information: Metadata from HDFS changes based on the usage. Each time you want to analyze metadata, you must run commands to view the snapshot file of the metadata. Each time you run a Jindo command to upload metadata, a batch number that is generated based on the upload time is used as the root directory to store the metadata. This ensures that the data you uploaded each time is not overwritten. You can delete historical data based on your requirements.

- 1 indicates the configuration path of system information in OSS.
- 2 indicates a namespace.
- 3 indicates a batch number.

Metadata schema

HDFS metadata that is uploaded to OSS is stored as a file in the JSON format. Schema information:

```
{
  "type":"string",          /*Inode type, FILE DIRECTORY*/
  "id": "string",            /*INode id*/
  "parentId" :"string",         /*Parent node ID*/
  "name":"string",         /*Inode name*/
  "size": "int",         /*Inode size, BIGINT*/
  "permission":"int",         /*Permission, INT*/
  "owner":"string",          /*Owner name*/
  "ownerGroup":"string",     /*Owner group name*/
  "mtime":"int",              /*Inode modification time, BIGINT*/
  "atime":"int",              /*The time when the inode was last accessed, BIGINT*/
  "attributes":"string",       /*File-related attributes*/
  "state":"string",            /*Inode status*/
  "storagePolicy":"string",    /*Storage policy*/
  "etag":"string"           /*etag*/
}
```

Use Jindo SQL to analyze metadata

1. Run the command shown in the following figure to start Jindo SQL.

```
[root@emr-header-1 ~]# jindo sql
Spark master: yarn, Application Id: application_1603081647416_0050
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.33 seconds, Fetched 3 row(s)
```

2. Query the tables whose data can be analyzed by using Jindo SQL.
   - Run the `show tables` command to view the tables whose data can be analyzed. Jindo SQL has two built-in modules audit_log and fs_image for audit and metadata analysis.
   - Run the `show partitions fs_image` command to view the partition information of the fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

     The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

3. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287249    1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       15464684820176650002    1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The namespace and datetime columns are added and indicate the namespace name and the timestamp when metadata is uploaded.

Example: Obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

Use Hive to analyze metadata

1. Create a table schema in Hive.

   You can use the following command to create a table schema of the metadata that is used for queries in Hive:

```
CREATE EXTERNAL TABLE `table_name`
(`type` string,
 `id` string,
 `parentId` string,
 `name` string,
 `size` bigint,
 `permission` int,
 `owner` string,
 `ownerGroup` string,
 `mtime` bigint,
 `atime` bigint,
 `attr` string,
 `state` string,
 `storagePolicy` string,
 `etag` string)
 ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
 STORED AS TEXTFILE
 LOCATION 'The OSS path to which file data is uploaded';
```

2. Use Hive to analyze data offline.

   After you create a Hive table, you can use Hive SQL to analyze metadata.

```
select * from table_name limit 200;
```

   The following figure shows an example.

```
hive> select * from inode_metadata_test0 limit 100;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_202009[redacted]
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1[redacted], Tracking URL = http://emr-heade[redacted]:20888/proxy/applic[redacted]
Kill Command = /usr/lib/hadoop-current/bin/hadoop job  -kill job_1599[redacted]
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-09-08 14:57:26,112 Stage-1 map = 0%,  reduce = 0%
2020-09-08 14:57:31,263 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.22 sec
MapReduce Total cumulative CPU time: 1 seconds 220 msec
Ended Job = job_[redacted]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.22 sec   HDFS Read: 6867 HDFS Write: 1524 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 220 msec
OK
Directory       1127433438684721971 4    1127433438684721971 3    /uttest/oss     0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Directory       1127433438684721971 9    1127433438684721971 3    /uttest/oss2    0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
Directory       1127433438684721971 6    1127433438684721971 4    /uttest/oss/dir 0       511     caojie  staff   1599545017636   1599545017636           Finalized       WARM
File    1127433438684721971 5    1127433438684721971 4    /uttest/oss/file1       0       420     caojie  staff   1599545017632   1599545017632           Finalized       WARM
File    1127433438684721971 7    1127433438684721971 6    /uttest/oss/dir/file2   0       420     caojie  staff   1599545017642   1599545017642           Finalized       WARM
File    1127433438684721971 8    1127433438684721971 6    /uttest/oss/dir/file3   0       420     caojie  staff   1599545017651   1599545017651           Finalized       WARM
Directory       1127433438684721972 0    1127433438684721971 9    /uttest/oss2/dir        0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
File    1127433438684721972 1    1127433438684721972 0    /uttest/oss2/dir/file2  0       420     caojie  staff   1599545017658   1599545017658           Finalized       WARM
File    1127433438684721972 2    1127433438684721972 0    /uttest/oss2/dir/file3  0       420     caojie  staff   1599545017666   1599545017666           Finalized       WARM
Directory       1127433438684721971 3    1767282355743385190 5    /uttest 0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Time taken: 10.734 seconds, Fetched: 10 row(s)
hive>
```

5.2.9. Use a JindoFS credential provider

The data of JindoFS is stored in Object Storage Service (OSS). If you want to access the data of JindoFS, you must provide an AccessKey pair that can be used to access OSS. SmartData V3.4.0 and later support JindoFS credential providers. You can use a JindoFS credential provider to save an encrypted AccessKey pair into a file. This prevents the AccessKey pair from being leaked.

Configure a JindoFS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **smartdata-site** tab.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **smartdata-site** tab.
3. Add configuration information.
   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

Parameter	Description
fs.jfs.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

   iii. Click **OK**.
4. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Credential provider types

- TemporaryAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a validity period and a security token with a validity period are used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this credential provider, you must configure the parameters described in the following table in environment variables.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**.
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- JindoCommonCredentialsProvider

This is a common credential provider. After you configure the parameters described in the following table, you can use this credential provider in both JindoOSS and JindoFS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.JindoCommonCredentialsProvider**.
jindo.common.accessKeyId	The AccessKey ID used to access OSS.
jindo.common.accessKeySecret	The AccessKey secret used to access OSS.
jindo.common.securityToken	The temporary security token used to access OSS.

- EcsStsCredentialsProvider

  This credential provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EcsStsCredentialsProvider**.

5.2.10. Use JindoFS in block storage mode to store encrypted files

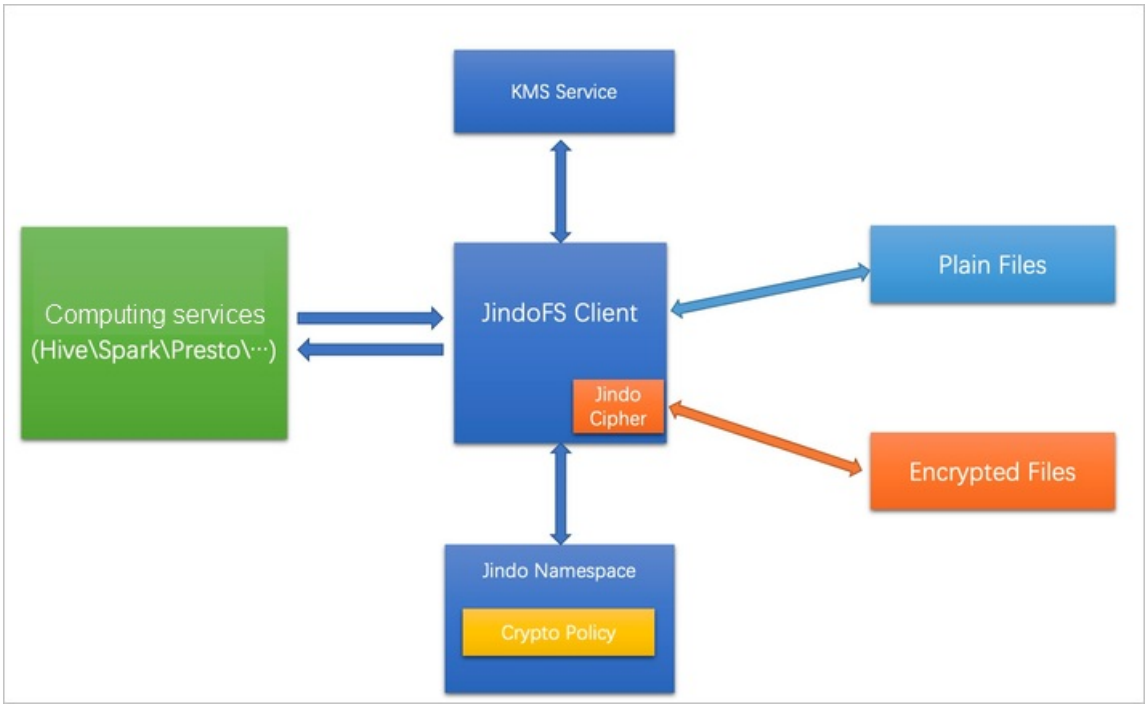
JindoFS in block storage mode supports file encryption. The encryption mechanism and usage method are similar to those of encryption zones for Apache HDFS. Keys are managed by Key Management Service (KMS). You can configure encryption policies for a directory that contains sensitive data. This way, you can transparently encrypt the data that you want to write to the directory and transparently decrypt the data that you want to read from the directory. In this process, you do not need to change your code.

Prerequisites

- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.
- KMS is activated. For more information, see Activate KMS.

Context

The following figure shows the architecture of JindoFS in block storage mode with file encryption supported.



Configure KMS-related parameters for JindoFS

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **namespace** tab.
3. Add configuration information.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description
crypto.provider.type	The type of the provider. Set this parameter to **ALIYUN**.
crypto.provider.endpoint	The public endpoint of KMS. For more information, see Make API requests.
crypto.provider.kms.accessKeyId	The AccessKey ID used to access KMS.
crypto.provider.kms.accessKeySecret	The AccessKey secret used to access KMS.

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Use JindoFS KeyProvider

JindoFS KeyProvider is used to connect JindoFS to KMS. The keys for JindoFS are stored in KMS. You can use JindoFS KeyProvider to create, query, and rotate keys based on KMS.

- Create a key: You can run the following command to create a key:

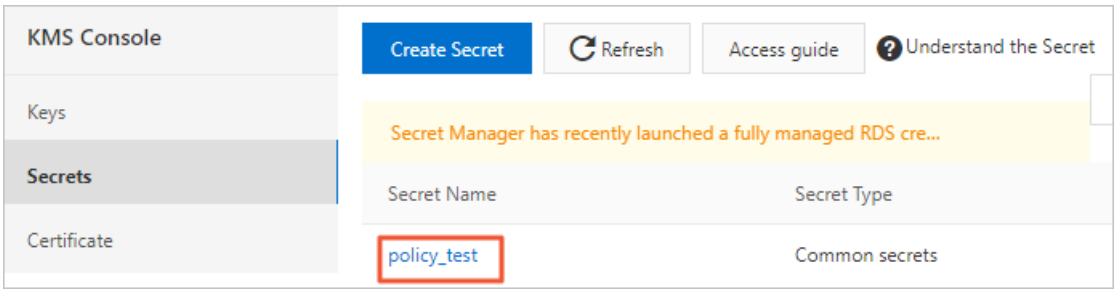
```
jindo key -create -keyIdName <keyIdName>
```

> **Note** In this topic, <keyIdName> in commands indicates the name of the key that you created.

For example, run the following command to create a key named *policy_test*:

```
jindo key -create -keyIdName policy_test
```

After the *policy_test* key is created, it is displayed on the Secrets page of the KMS console.



- Query keys: You can run the following command to query the names of all existing keys for JindoFS:

```
jindo key -list
```

The following information is returned:

```
Listing Keys:
        policy_test
        policy_test2
```

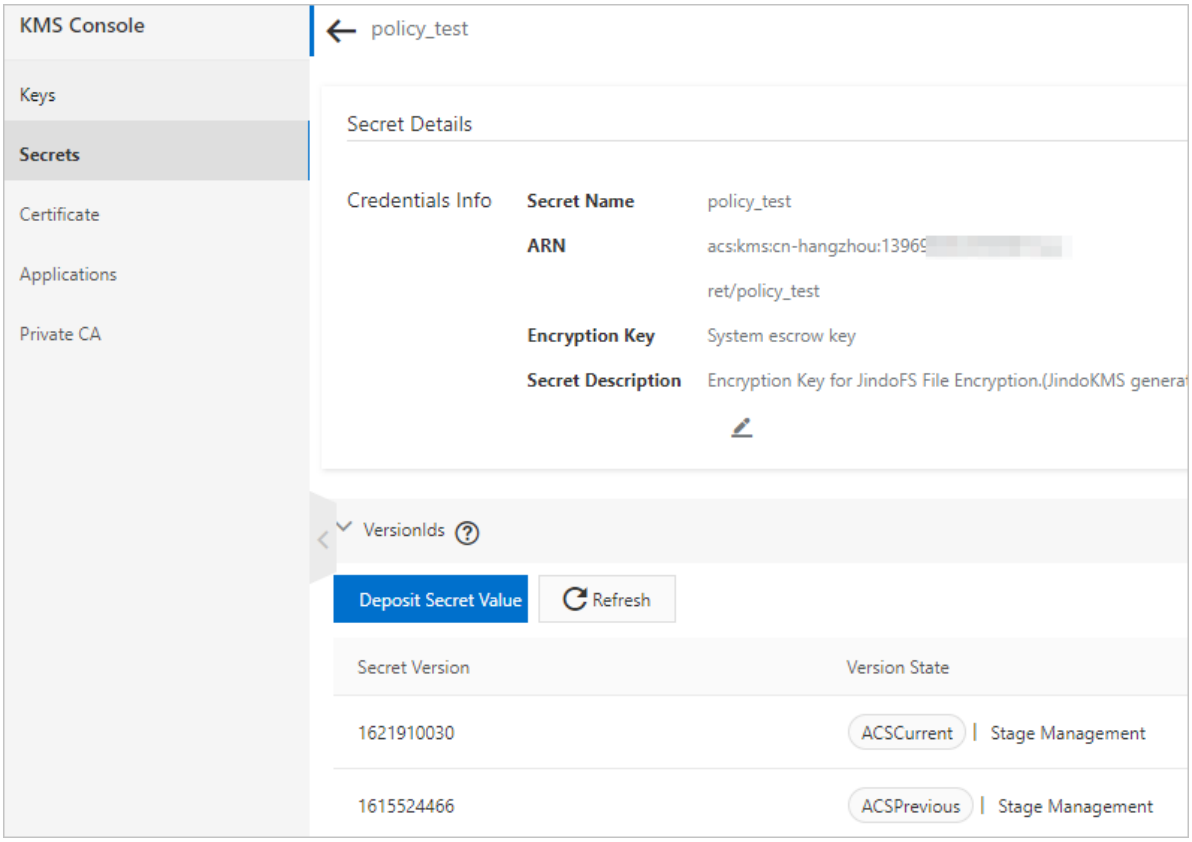
- Rotate a key: You can run the following command to periodically rotate a key based on its ID. After the key is rotated, the key version changes. If a file is encrypted after the rotation, the latest key version is used. If a file is decrypted after the rotation, the key version that corresponds to the file is used.

```
jindo key -roll -keyIdName <keyIdName>
```

For example, run the following command to rotate the *policy_test* key:

```
jindo key -roll -keyIdName policy_test
```

After the *policy_test* key is rotated, the version of the key is updated. The original version is in the **ACSPrevious** state and the new version is in the **ACSCurrent** state.



Manage encryption policies for JindoFS

You can run the following commands to configure and query encryption policies.

- Configure an encryption policy:

```
jindo jfs -setCryptoPolicy -keyIdName <keyIdName> <path>
```

> **Note** `<path>` in the preceding example indicates the path of a file on JindoFS, for example, *jfs://test/*.

- Query an encryption policy:

```
jindo jfs -getCryptoPolicy <path>
```

Example:

1. Query the encryption policy of the *jfs://test/* path:

```
jindo jfs -getCryptoPolicy jfs://test/
```

`{NONE}` is returned.

2. Configure an encryption policy for the *jfs://test/* path:

```
jindo jfs -setCryptoPolicy -keyIdName policy_test jfs://test/
```

3. Go to the *bigboot* directory and query the encryption policy of the *jfs://test/* path again:

```
jindo jfs -getCryptoPolicy jfs://test/
```

The following information is returned:

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/b2jindosdk/3.4.0-hadoop3.1/pack
age/b2jindosdk-3.4.0-hadoop3.1/lib/jindo-distcp-3.4.0.jar!/org/slf4j/impl/StaticLoggerB
inder.class]
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/hadoop/3.2.1-1.0.1/package/hado
op-3.2.1-1.0.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticL
oggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
21/03/12 13:52:34 WARN: Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable
21/03/12 13:52:35 INFO: Jboot log name is /var/log/bigboot/jboot-20210312-135234-12953.
LOG
21/03/12 13:52:35 INFO: Write buffer size 1048576, logic block size 134217728
21/03/12 13:52:35 INFO: cmd=getFileStatus, src=jfs://test/, dst=null, size=0, parameter
=null, time-in-ms=7, version=3.4.0
21/03/12 13:52:35 INFO: cmd=getCryptoPolicy, src=jfs://test/, dst=null, size=0, paramet
er=, time-in-ms=2, version=3.4.0
The crypto policy of path: jfs://test/ is {cipherSuite: AES_CTR_NOPADDING_256, keyIdNam
e: policy_test2, keyIdVersion: null, edek: , iv: }
21/03/12 13:52:35 INFO: Read total statistics: oss read average <none>, cache read aver
age <none>, read oss percent <none>
```

After you configure the encryption policy, you can read data from or write data to the files under this path.

- Copy a local file to HDFS:

```
hadoop fs -put test.log jfs://test/
```

- Display the content of a file:

```
hadoop fs -cat jfs://test/test.log
```

5.3. JindoFS in cache mode

5.3.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in Object Storage Service (OSS) based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

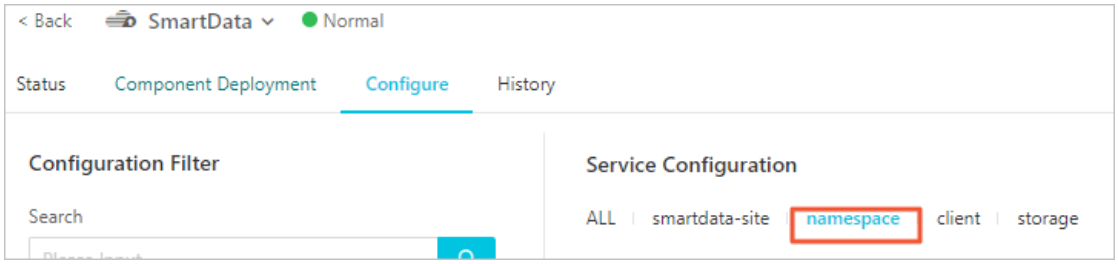
  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. Click **OK**.
5. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
6. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **client** tab.
2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   You do not need to restart the SmartData service to apply the new configurations.
3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

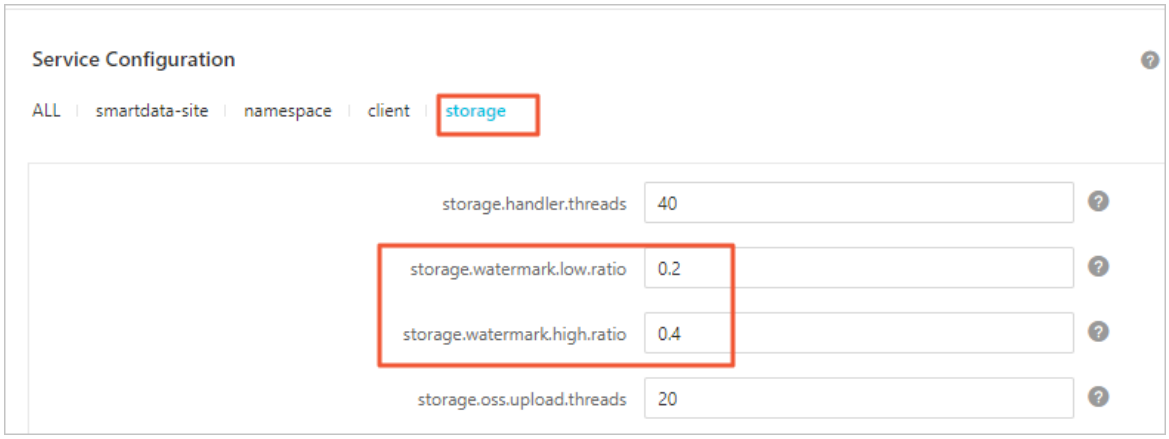
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you specify. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to specify an AccessKey pair. In other cases, you must specify an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you choose for accessing files in OSS:

- OSS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.
  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss.accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.endpoint	The endpoint of the OSS bucket that serves as the storage backend.

> **Note** You can set these parameters to the values of the parameters in an EMR version earlier than EMR V3.30.0.

- JFS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **namespace** tab.
  ii. Set **jfs.namespaces** to **test**.
  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize the cache performance. After you configure the parameters, you do not need to restart the SmartData service to apply the new configurations.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. You can use this parameter to prevent upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If an excessive amount of memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

5.3.2. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

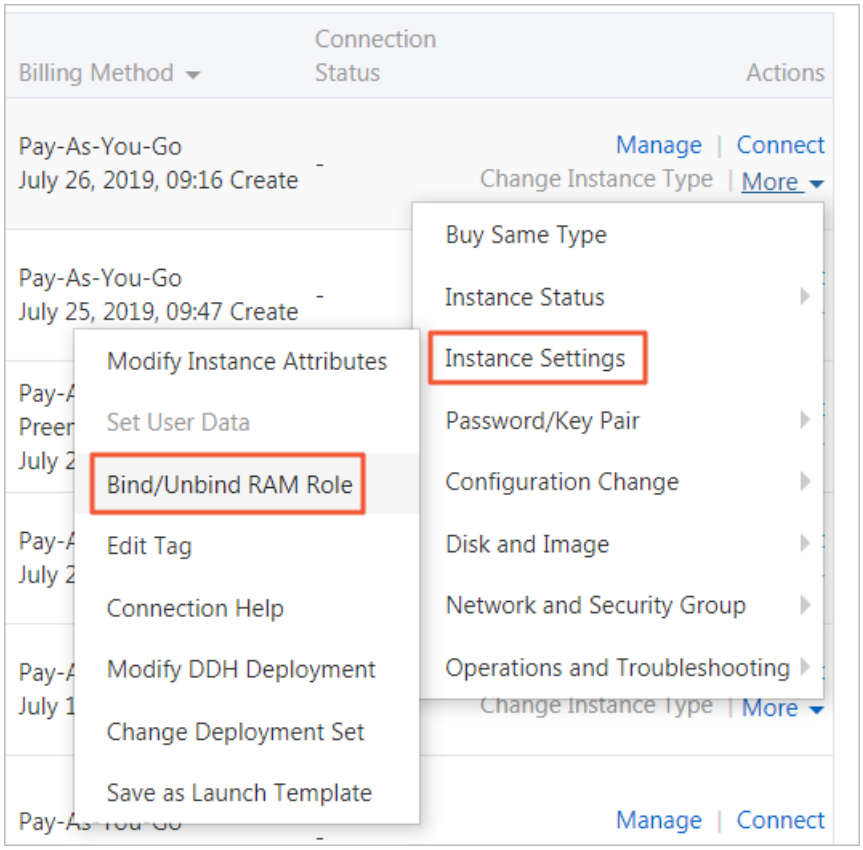
1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.
4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

   ```
   hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
   ```

2. Access OSS by using the FileSystem interface of Hadoop.

   Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

5.3.3. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false

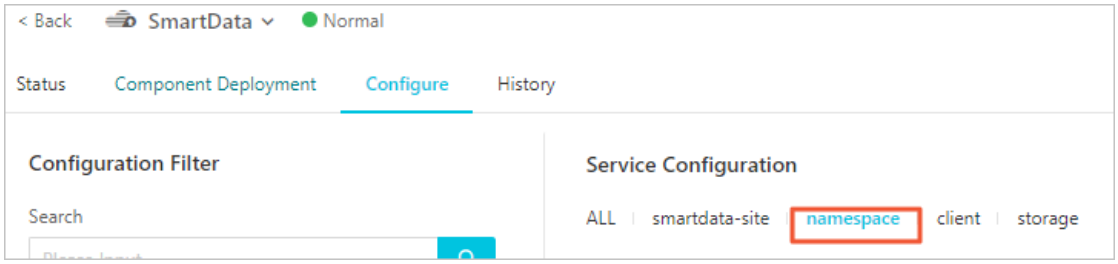
Parameter	Description
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-

right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

5.3.4. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

> **Notice**
>
> - The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
> - After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
> - Hive and Presto jobs do not use Hadoop Job Committer.
> - In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.

i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
ii. Click the **Configure** tab.
iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.
6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.

3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

   The default value is 8.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5.3.5. Use JindoFS OSS credential providers

You can use a credential provider to save an encrypted AccessKey pair into a file. This prevents the leak of the AccessKey pair.

Context

You can use a Hadoop credential provider to save an encrypted AccessKey pair into a file. This prevents the issue that the AccessKey pair is transmitted in plaintext. You can select an appropriate JindoOSS credential provider based on your business requirements.

Configure a JindoOSS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **smartdata-site** tab.

   i. Click the **Configure** tab.

   ii. In the **Service Configuration** section, click the **smartdata-site** tab.

3. Add configuration information.

   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

   - Global configuration (for all buckets)

Parameter	Description
fs.jfs.cache.oss.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

- Bucket-level configuration

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types. **Note** XXX indicates the name of an Object Storage Service (OSS) bucket.

Credential provider types

You can select different credential providers based on your business requirements. The following providers are supported:

- Global configuration
  - TemporaryAliyunCredentialsProvider

    This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.cache.oss.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

- Bucket-level configuration

- TemporaryAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.
fs.jfs.cache.oss.bucket.XXX.securityToken	The temporary security token used to access the OSS bucket.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access an OSS bucket.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access the OSS bucket.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access the OSS bucket. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

Use a Hadoop credential provider to store AccessKey pair information

> **Note** For more information about Hadoop credential providers, see CredentialProvider API Guide.

Use a command that is provided by Hadoop to store AccessKey pair and security token information into a credential file. Syntax:

```
hadoop credential <subcommand> [options]
```

For example, in global configuration mode, store AccessKey pair and token information into a JCEKS file. You can protect the file by using file permissions or you can specify a password to encrypt the information you want to store. If you do not specify a password, the default string is used for encryption.

```
hadoop credential create fs.jfs.cache.oss.accessKeyId -value AAA -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.accessKeySecret -value BBB -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.securityToken -value  CCC -provider jceks://file/root/oss.jceks
```

After a credential file is generated, you must configure the parameter described in the following table to specify the provider type and location.

Parameter	Description
fs.jfs.cache.oss.security.credential.provider.path	The path used to store the credential file that stores AccessKey pair information. For example, you can set this parameter to *jceks://file/${user.home}/oss.jceks*, which indicates that the *oss.jceks* file is stored in the home directory.

5.3.6. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

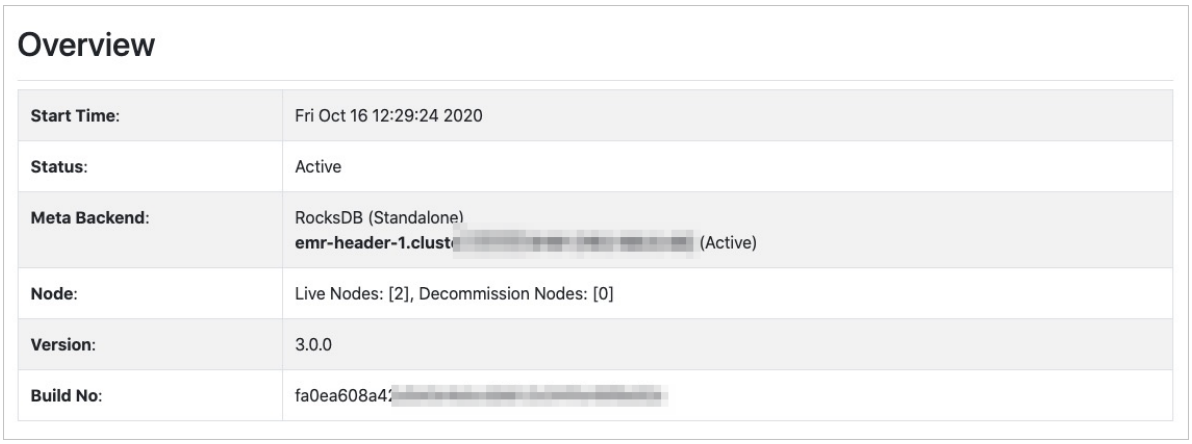
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

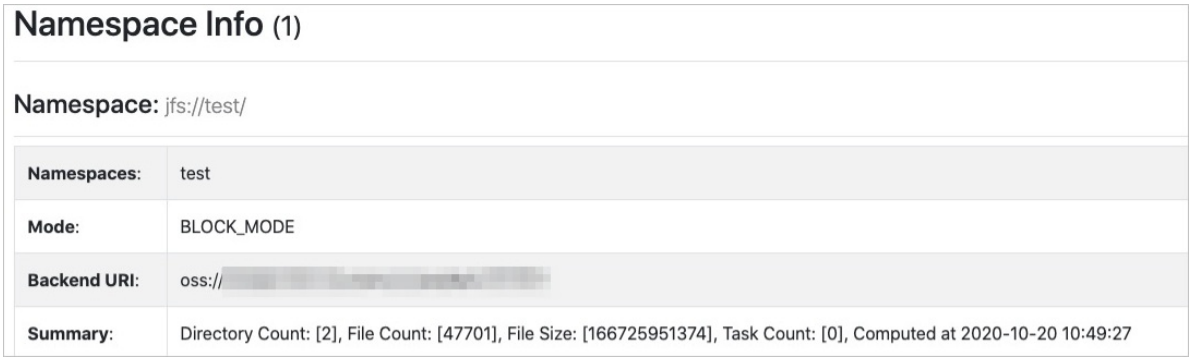
  

  Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.

  Namespace Info (1)

  Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.

StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.

Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

5.3.7. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

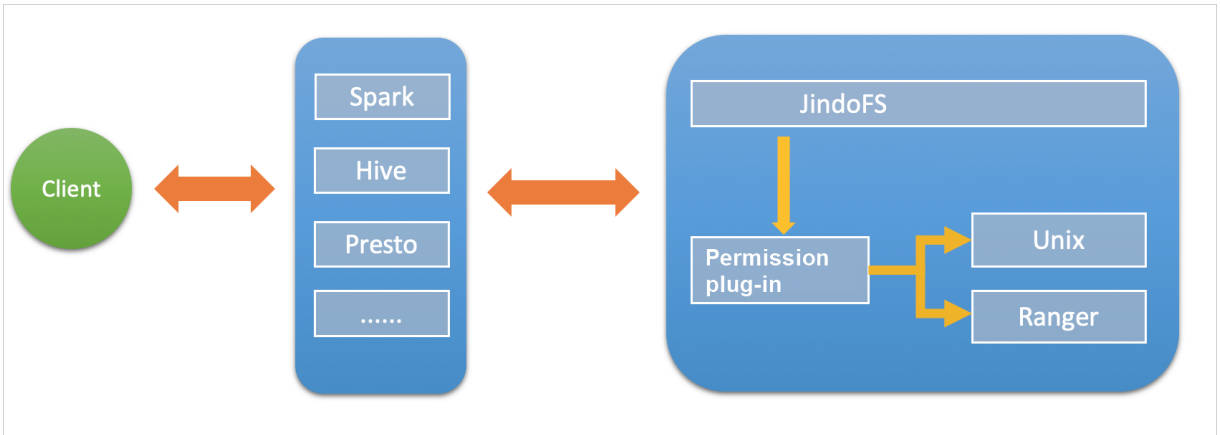
Permission management based on the storage mode:

- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.

- Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.

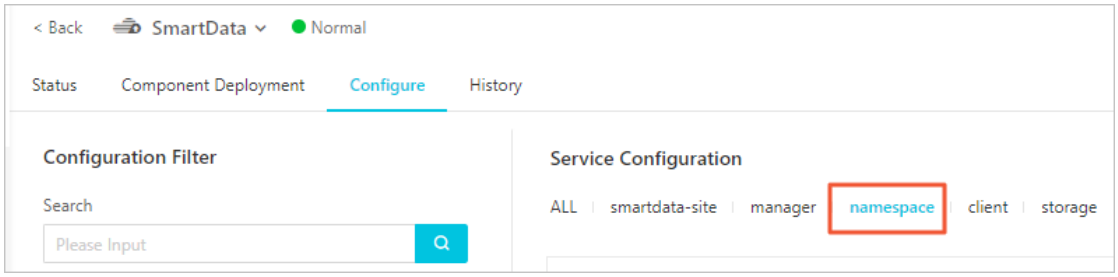
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

   After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

   If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

5.4. JindoTable

5.4.1. Enable query acceleration based on a native engine

JindoTable provides a native engine for you to accelerate the speed of using Spark, Hive, or Presto to query files in the ORC or Parquet format. This topic describes how to enable query acceleration based on the native engine to improve the performance of Spark, Hive, and Presto.

Prerequisites

An E-MapReduce (EMR) cluster is created, and a file in the ORC or Parquet format is stored in JindoFS or Object Storage Service (OSS). For more information about how to create a cluster, see Create a cluster.

Limits

- Binary files are not supported.

- Partitioned tables whose values of partition key columns are stored in files are not supported.
- EMR clusters of V5.X.X or later are not supported.
- spark.read.schema (userDefinedSchema) is not supported.
- Data of the DATE type must be in the YYYY-MM-DD format and range from 1400-01-01 to 9999-12-31.
- If a table contains two columns that have the same column name with different letter cases, such as ID and id, queries on the columns cannot be accelerated.
- The following table lists the supported Spark, Hive, and Presto engines and the file formats supported by each engine.

Engine	ORC	Parquet
Spark2	Supported	Supported
Spark3	Supported	Supported
Presto	Supported	Supported
Hive2	Not supported	Supported
Hive3	Not supported	Supported

- The following table lists the supported Spark, Hive, and Presto engines and the file systems supported by each engine.

Engine	OSS	JindoFS	HDFS
Spark2	Supported	Supported	Supported
Presto	Supported	Supported	Supported
Hive2	Supported	Supported	Not supported
Hive3	Supported	Supported	Not supported

Improve the performance of Spark

1. Enable query acceleration for ORC or Parquet files in JindoTable.

> **Note**
>
> - Query acceleration consumes off-heap memory. We recommend that you add `--conf spark.executor.memoryOverhead=4g` to a Spark task to apply for additional resources for query acceleration.
> - If you use Spark to read data from an ORC or Parquet file, the DataFrame API or Spark SQL is required.

   - Global configuration

a. Go to the Cluster Overview page of your cluster.
   a. Log on to the Alibaba Cloud EMR console.
   b. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   c. Click the **Cluster Management** tab.
   d. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
b. Modify the related parameter.
   a. In the left-side navigation pane, choose **Cluster Service > Spark**.
   b. On the Spark service page, click the **Configure** tab.
   c. Search for the spark.sql.extensions parameter and change its value to io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sql.JindoTableExtension.
c. Save the configuration.
   a. Click **Save** in the upper-right corner.
   b. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
d. Restart ThriftServer.
   a. Choose **Actions > Restart ThriftServer** in the upper-right corner.
   b. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   c. In the **Confirm** message, click **OK**.

- Job-level configuration

  When you configure a Spark Shell or Spark SQL job, add the following configuration to the code:

  ```
  --conf spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sq
  l.JindoTableExtension
  ```

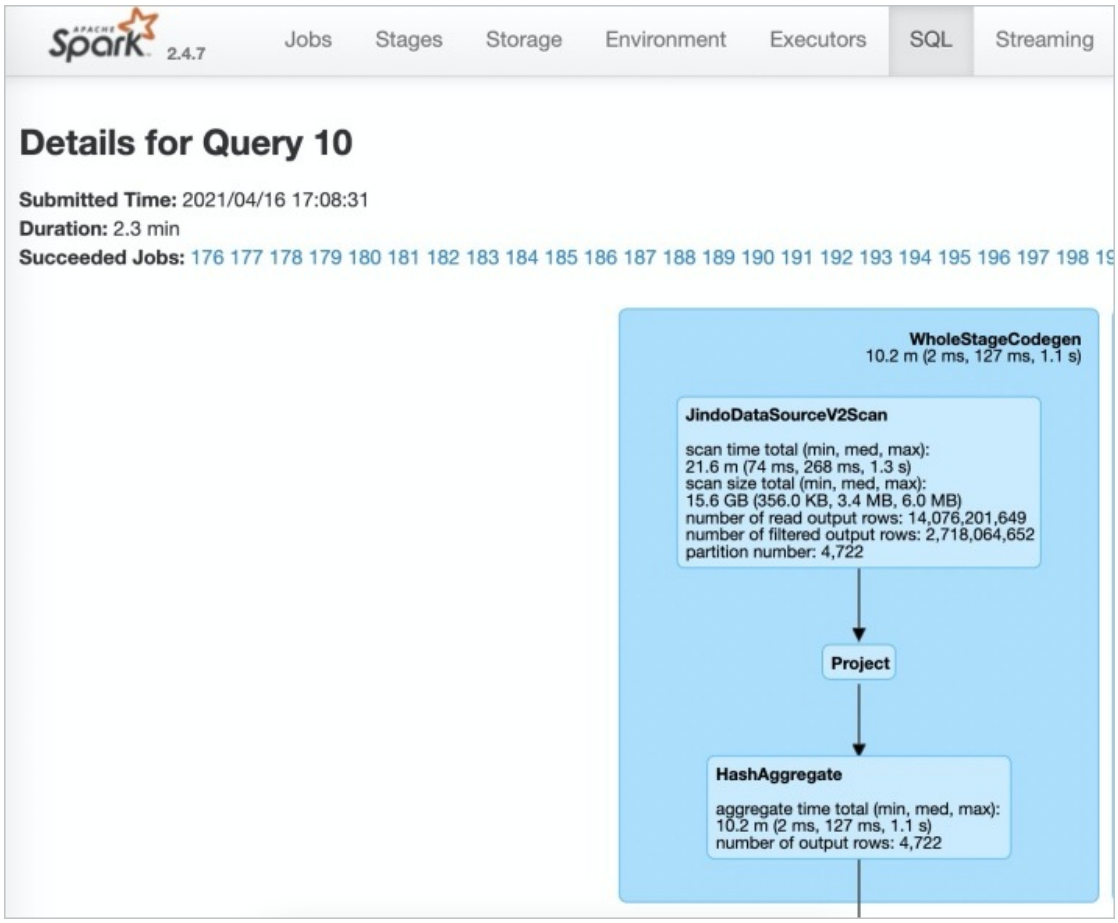
  For more information about how to configure a Spark job, see Configure a Spark Shell job or Configure a Spark SQL job.

2. Check whether query acceleration is enabled.
   i. Access the web UI of Spark History Server.

      For more information, see Access the web UIs of open source components.

ii. On the SQL tab of Spark, view information about the related task.

If JindoDataSourceV2Scan appears, query acceleration is enabled. Otherwise, check the configurations in Step 1.



Improve the performance of Presto

> **Notice** Presto has high query concurrency and uses off-heap memory for query acceleration. To use the query acceleration feature, make sure that the memory is greater than 10 GB.

By default, `catalog: hive-acc` of the native engine is built in the Presto service. You can use `catalog: hive-acc` to enable query acceleration.

Example:

```
presto --server https://emr-header-1.cluster-xxx:7778/ --catalog hive-acc --schema default
```

`emr-header-1.cluster-xxx` indicates the hostname of the emr-header-1 node.

> **Note** When you use this feature in Presto, complex data types such as MAP, STRUCT, and ARRAY are not supported.

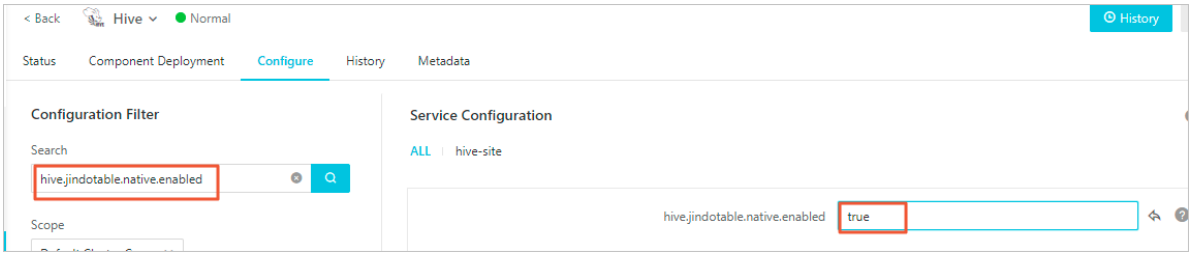
Improve the performance of Hive

> **Notice** If you have high requirements for job stability, we recommend that you do not enable query acceleration.

You can use one of the following methods to improve the performance of Hive:

- Use the EMR console

  On the **Configure** tab of the Hive service page, search for the **hive.jindotable.native.enabled** parameter and change its value to **true**. Then, save the configuration and restart the Hive service. This method is suitable for Hive on MapReduce and Hive on Tez jobs.



- Use the CLI

  Set `hive.jindotable.native.enabled` to `true` in the CLI to enable query acceleration. By default, the query acceleration plug-in for Parquet files is deployed in JindoTable in EMR V3.35.0 and later. You can directly set this parameter in EMR V3.35.0 and later.

```
set hive.jindotable.native.enabled=true;
```

> **Note** When you use this feature in Hive, complex data types such as MAP, STRUCT, and ARRAY are not supported.

5.4.2. Use JindoTable

JindoTable is used to implement tiered storage, optimize table files, and collect data statistics based on the popularity of tables or partitions. This topic describes how to use JindoTable.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your on-premises machine.
- An EMR cluster of the 3.30.0 version or later is created. For more information about how to create a cluster, see Create a cluster.

Use JindoTable

Common commands:

- -accessStat
- -cache
- -archive
- -unarchive
- -uncache
- -status
- -optimize

- -showTable
- -showPartition
- -listTables
- -dumpmc

> **Notice** Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

-accessStat

- Syntax

  jindo table -accessStat {-d} *<days>* **{-n}** <topNums>

- Description

  This command is used to query the access records in which tables or partitions are visited most in a specified time range.

  <days> and <topNums> must be positive integers. If <days> is 1, all the access records generated from 00:00 (local time) on the current day to the current time are queried.

- Example: Query the 20 access records of tables or partitions that are most frequently visited within the last seven days.

```
jindo table -accessStat -d 7 -n 20
```

-cache

- Syntax

  jindo table -cache {-t} <dbName.tableName> **[-p]** <partitionSpec> **[-pin]**

- Description

  This command is used to cache data of specified tables or partitions to local disks.

  The data of the tables or partitions must be stored in Object Storage Service (OSS) or JindoFileSystem (JindoFS). Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`. When you specify `-pin`, if the cache space is insufficient, do not delete related data if possible.

- Example: Cache data of the db1.t1 table that is generated on March 16, 2020 to local disks.

```
jindo table -cache -t db1.t1 -p date=2020-03-16
```

-uncache

- Syntax

  jindo table -uncache {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to delete cached data of specified tables or partitions from local disks.

  The data of the tables or partitions must be stored in OSS or JindoFS. Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

- Examples:
  - Delete cached data of the db1.t2 table from local disks.

    ```
    jindo table -uncache -t db1.t2
    ```

  - Delete cached data of the db1.t1 table from local disks.

    ```
    jindo table -uncache -t db1.t1 -p date=2020-03-16,category=1
    ```

-archive

- Syntax

  jindo table -archive {-a|i} {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to lower the level of the policy that is used to store data of specified tables or partitions. By default, the Archive storage class is used.

  To use the Infrequent Access (IA) storage class, add -i to the command. Specify tables in the format of database.table. Specify partitions in the format of 'partitionCol1=1,partitionCol2=2,...'.

- Example: Cache data of the db1.t1 table to local disks.

  ```
  jindo table -archive -t db1.t1 -p date=2020-10-12
  ```

-unarchive

- Syntax

  jindo table -unarchive [-o|-i] {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to change the storage class from Archive to Standard.

  If `-o` is added to the command, an Archived object is temporarily restored. If `-i` is added to the command, an Archived object is changed to an IA object.

- Example:

  ```
  jindo table -unarchive -o -t db1.t1 -p date=2020-03-16,category=1
  ```

  ```
  jindo table -unarchive -i -t db1.t2
  ```

-status

- Syntax

  jindo table -status {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to view the data storage status of specified tables or partitions.

- Examples:
  - View the data storage status of the db1.t2 table.

    ```
    jindo table -status -t db1.t2
    ```

- View the data storage status of the db1.t1 table on March 16, 2020.

```
jindo table -status -t db1.t1 -p date=2020-03-16
```

-optimize

- Syntax

  jindo table -optimize {-t} <dbName.tableName>

- Description

  This command is used to optimize the data organization of tables at the storage layer.

- Example: Optimize the data organization of the db1.t1 table at the storage layer.

```
jindo table -optimize -t db1.t1
```

-showTable

- Syntax

  jindo table -showTable {-t} <dbName.tableName>

- Description

  This command is used to display all partitions in a partitioned table or display the data storage of a non-partitioned table.

- Example: Display all partitions in the db1.t1 partitioned table.

```
jindo table -showTable -t db1.t1
```

-showPartition

- Syntax

  jindo table -showPartition {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to display the data storage of partitions.

- Example: Display the data storage of all partitions in the db1.t1 partitioned table on October 12, 2020.

```
jindo table -showPartition -t db1.t1 -p date=2020-10-12
```

-listTables

- Syntax

  jindo table -listTables [-db] <dbName>

- Description

  This command is used to display all tables in a specified database. If you do not specify `[-db]`, tables in the default database are displayed.

- Examples:

- Display tables in the default database.

```
jindo table -listTables
```

- Display tables in the db1 database.

```
jindo table -listTables -db db1
```

-dumpmc

- Syntax

jindo table -dumpmc {-i} <accessId> **{-k}** <accessKey> **{-m}** <numMaps> **{-t}** <tunnelUrl> **{-project}** <projectName> **{-table}** <tablename> **{-p}** <partitionSpec> **{-f}** <csv|tfrecord> **{-o}** <outputPath>

Parameter	Description	Required
-i	The AccessKey ID of your Alibaba Cloud account.	Yes
-k	The AccessKey secret of your Alibaba Cloud account.	Yes
-m	The number of map tasks.	Yes
-t	The VPC Tunnel endpoint of MaxCompute.	Yes
-project	The name of the MaxCompute project.	Yes
-table	The name of the MaxCompute table.	Yes
-p	The partition information. Example: `pt=xxx`. Separate multiple partitions with commas (,), such as `pt=xxx,dt=xxx`.	No
-f	The file format. Valid values: ○ tfrecord ○ csv	Yes
-o	The destination path.	Yes

- Description

  This command is used to dump MaxCompute tables to an EMR cluster or OSS. The formats CSV and TFRECORD are supported.

- Examples:

- Dump a MaxCompute table in the TFRECORD format to an EMR cluster.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o /tmp/outputtf1 -f tfrecord
```

- Dump a MaxCompute table in the CSV format to OSS.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o oss://bucket1/tmp/outputcsv -f csv
```

5.4.3. Archive and unarchive data in SDK mode

JindoTable allows you to run the archiveTable and unarchiveTable commands in SDK mode to archive and unarchive data in Object Storage Service (OSS). The commands do not rely on the Jindo Namespace Service component of SmartData. This topic describes how to use the archiveTable and unarchiveTable commands.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.
- The partitioned table or non-partitioned table that you want to archive is stored in OSS. Only table data can be archived.

Context

You can use the original archive and unarchive commands of JindoTable to archive or unarchive tables or partitions in OSS. However, these commands rely on the Jindo Namespace Service component of SmartData. The new commands archiveTable and unarchiveTable do not rely on the Jindo Namespace Service component.

The archiveTable and unarchiveTable commands have the following advantages over the archive and unarchive commands:

- You can run the archiveTable and unarchiveTable commands even if the SmartData service is not deployed in your cluster. For example, you can run the commands on a self-managed cluster.
- You can configure filter parameters in the archiveTable or unarchiveTable command to archive or unarchive a large number of partitions on multiple threads at the same time. If local multithreading cannot meet your business requirements, you can run MapReduce tasks on the entire cluster to archive or unarchive data.

For more information about the original archive and unarchive commands, see Use JindoTable.

Limits

The archiveTable and unarchiveTable commands are supported only in EMR V3.36.0 and later minor versions, and EMR V5.2.0 and later minor versions.

archiveTable

You can use the archiveTable command to archive tables or partitions in OSS.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to obtain help information:

```
jindo table -help archiveTable
```

The following information is returned:

```
  <dbName.tableName>      The table to archive.
  -a/-i                   storage policy, -a for Archive and -i for IA
                          (Infrequent Access).
  <condition>/-fullTable  A filter condition to determine which partitions should
                          be archived, supporting common operators (like '>'),
                          while -fullTable means that all partitions (or a whole
                          un-partitioned table) should be archived. One but only
                          one option must be specified among -c "<condition>" and
                          -fullTable.
  <before days>           Optional, saying that table/partitions should be
                          archived only when they are created (not updated or
                          modified) more than some days before from now.
  <parallelism>           The maximum concurrency when archiving partitions, 1 by
                          default.
  -mr/-mapReduce          Archive table/partitions using cluster-level MapReduce
                          job instead of local-level multi-thread.
  -e/-explain             If present, the command would not really archive data,
                          but only prints the table/partitions that would be
                          archived for given conditions.
        <working directory>: A directory to locate map-reduce temp files. Must not be a
  local file system directory. 'hdfs:///tmp/<current user>/jindotable-policy/' by
  default.
  <log directory>  A directory to locate log files, '/tmp/<current user>/' by
                   default.
```

archiveTable syntax:

```
-archiveTable -t <dbName.tableName> \
-a/-i \
[-c "<condition>" | -fullTable] \
[-b/-before <before days>] \
[-p/-parallel <parallelism>] \
[-mr/-mapReduce] \
[-e/-explain] \
[-w/-workingDir <working directory>] \
[-l/-logDir <log directory>]
```

Parameter	Description	Required
-t <dbName.tableName>	The name of the table that you want to archive. You must specify this parameter in the `Database name.Table name` format. Separate the database name and table name with a period (.). The table can be a partitioned table or a non-partitioned table.	Yes

Parameter	Description	Required
-a/-i	The storage class in which you want to archive data. You can use one of the following options to specify a storage class: ○ `-a` : Archive ○ `-i` : Infrequent Access (IA) If you use the -i option in the command, the files whose storage class is Archive are skipped.	Yes
-c "<condition>" | -fullTable	You must specify either `-fullTable` or `-c "<condition>"` . ○ If you specify `-fullTable` , the entire partitioned or non-partitioned table is archived. ○ If you specify `-c "<condition>"` , only the partitions that meet the filter condition are archived. Common operators, such as greater-than signs (>), are supported. For example, if the partition key column is the ds column whose data type is String and you want to archive partitions whose partition names are greater than 'd', use `-c " ds > ' d' "` .	No
-b/before <before days>	Only the tables or partitions that were created at least the specified days ago can be archived.	No
-p/-parallel <parallelism>	The parallelism of archiving operations.	No
-mr/-mapReduce	Hadoop MapReduce instead of local multithreading is used to archive data.	No
-e/-explain	The explain mode is used. In explain mode, the list of partitions to be archived is displayed, but no data is archived.	No
-w/-workingDir	The working directory of a MapReduce job. This option is used only when you use a MapReduce job to archive data. You must have read and write permissions on the directory. The directory can be empty or not. Temporary files are created when you run the MapReduce job and are automatically deleted after the job is completed.	No
-l/-logDir <log directory>	The directory in which log files are stored.	No

unarchiveTable

The syntax of the unarchiveTable command is similar to the syntax of the archiveTable command. You can use the unarchiveTable command to unarchive tables or partitions in OSS.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to obtain help information:

```
jindo table -help unarchiveTable
```

The following information is returned:

```
  <dbName.tableName>      The table to unarchive.
  -i                      unarchive to IA (Infrequent Access).
  -o                      restore to make archived data accessible temporarily.
  <condition>/-fullTable  A filter condition to determine which partitions should
                          be unarchived, supporting common operators (like '>'),
                          while -fullTable means that all partitions (or a whole
                          un-partitioned table) should be unarchived. One but
                          only one option must be specified among -c
                          "<condition>" and -fullTable.
  <before days>           Optional, saying that table/partitions should be
                          unarchived only when they are created (not updated or
                          modified) more than some days before from now.
  <parallelism>           The maximum concurrency when unarchiving partitions, 1
                          by default.
  -mr/-mapReduce          Unarchive table/partitions using cluster-level
                          MapReduce job instead of local-level multi-thread.
  -e/-explain             If present, the command would not really unarchive
                          data, but only prints the table/partitions that would
                          be unarchived for given conditions.
        <working directory>: A directory to locate map-reduce temp files. Must not be a
  local file system directory. 'hdfs:///tmp/<current user>/jindotable-policy/' by
  default.
  <log directory>  A directory to locate log files, '/tmp/<current user>/' by
                   default.
```

unarchiveTable syntax:

```
-unarchiveTable -t <dbName.tableName> \
[-i/-o] \
[-c "<condition>" | -fullTable] \
[-b/-before <before days>] \
[-p/-parallel <parallelism>] \
[-mr/-mapReduce] \
[-e/-explain] \
[-w/-workingDir <working directory>] \
[-l/-logDir <log directory>]
```

In the unarchiveTable command, the optional parameter -i/-o is used instead of the required parameter -a/-i. This is the only difference between the parameters of the unarchiveTable command and the archiveTable command.

Description of -i/-o:

- If you do not specify -i/-o, the storage class of the data that you want to unarchive is changed to Standard.

- If you specify the -i option, the storage class of the data that you want to unarchive is changed to IA. Files whose storage class is Standard are skipped.
- If you specify the -o option, data is only temporarily unarchived and its storage class is retained. Files whose storage class is Standard and files whose storage class is IA are skipped. Files that are previously unarchived are also skipped. This way, these files are not repeatedly unarchived.

5.4.4. Use the MoveTo command of JindoTable

JindoTable allows you to run the MoveTo command to migrate data in tables or partitions. This topic describes how to use the MoveTo command.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

The MoveTo command can automatically update metadata after the command copies the underlying data. This way, data in a table or partitions can be fully migrated to the destination path. You can configure filter conditions for the MoveTo command to migrate a large number of partitions at the same time. JindoTable also provides some protective measures to ensure data integrity and security when the MoveTo command is used to migrate data.

Limits

The MoveTo command is supported only in EMR V3.36.0 and later minor versions, and in EMR V5.2.0 and later minor versions.

Use the MoveTo command

> **Notice** Only one MoveTo process can be run at a time in an EMR cluster. If you attempt to start a MoveTo process on an EMR cluster on which another MoveTo process is running, your request is rejected because the configuration lock is unavailable. A message that contains information about the running MoveTo process is also displayed. In this case, you can terminate the running MoveTo process and start a new MoveTo process, or wait for the running MoveTo process to end.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to obtain help information:

   ```
   jindo table -help moveTo
   ```

   Information similar to the following output is returned:

```
<dbName.tableName>      The table to move.
<destination path>      The destination base directory which is always at the
                          same level of a 'table location', where the moved
                          partitions or un-partitioned data would located in.
<condition>/-fullTable  A filter condition to determine which partitions should
                          be moved, supporting common operators (like '>') and
                          built-in UDFs (like to_date) (UDFs not supported
                          yet...), while -fullTable means that all partitions (or
                          a whole un-partitioned table) should be moved. One but
                          only one option must be specified among -c
                          "<condition>" and -fullTable.
<before days>           Optional, saying that table/partitions should be moved
                          only when they are created (not updated or modified)
                          more than some days before from now.
<parallelism>           The maximum concurrency when copying partitions, 1 by
                          default.
        <OSS storage policy>: Storage policy for OSS destination, which can be Standard
  (by default), IA, Archive, or ColdArchive. Not applicable for destinations other
  than OSS. NOTE: if you are willing to use ColdArchive storage policy, please
  make sure that Cold Archive has been enabled for your OSS bucket.
-o/-overWrite     Overwriting the final paths where the data would be moved.
                    For partitioned tables this overwrites partitions' locations
                    which are subdirectories of <destination path>; for
                    un-partitioned table this overwrites the <destination path>
                    itself.
-r/-removeSource  Let the source data be removed when the corresponding
                    table/partition is successfully moved to the new destination.
                    Otherwise (by default), the source data would be left as it
                    was.
-skipTrash        Applicable only when [-r/-removeSource] is enabled. If
                    present, source data would be immediately deleted from the
                    file system, bypassing the trash.
-e/-explain       If present, the command would not really move data, but only
                    prints the table/partitions that would be moved for given
                    conditions.
<log directory>   A directory to locate log files, '/tmp/<current user>/' by
                    default.
```

MoveTo syntax:

```
jindo table -moveTo \
  -t <dbName.tableName> \
  -d <destination path> \
  [-c "<condition>" | -fullTable] \
  [-b/-before <before days>] \
  [-p/-parallel <parallelism>] \
  [-s/-storagePolicy <OSS storage policy>] \
  [-o/-overWrite] \
  [-r/-removeSource] \
  [-skipTrash] \
  [-e/-explain] \
  [-l/-logDir <log directory>]
```

Parameter	Description	Required
-t <dbName.tableName>	The name of the table that you want to migrate. You must specify this parameter in the `Database name.Table name` format. Separate the database name and table name with a period (.). The table can be a partitioned table or a non-partitioned table.	Yes
-d <destination path>	The destination path. No matter whether you want to migrate a specific partition or an entire non-partitioned table, this parameter specifies a table-level path. If you want to migrate a partition, the complete path of the partition is composed of the value of this parameter and the name of the partition, such as `<destination path>/p1=v1/p2=v2/` .	Yes
-c "<condition>" | -fullTable	You must specify either `-c "<condition>"` or `-fullTable` . ○ If you specify `-fullTable` , the entire partitioned or non-partitioned table is archived. ○ If you specify `-c "<condition>"` , only the partitions that meet the filter condition are archived. Common operators, such as greater-than signs (>), are supported. For example, if the partition key column is the ds column whose data type is String and you want to archive partitions whose partition names are greater than 'd', use `-c " ds > 'd' "` .	No
-b/before <before days>	Only the tables or partitions that were created at least the specified days ago can be migrated.	No
-p/-parallel <parallelism>	The parallelism among migration operations.	No

Parameter	Description	Required
-s/-storagePolicy <OSS storage policy>	The storage class that you want to use after data is migrated to Object Storage Service (OSS). Valid values: ○ Standard ○ IA ○ Archive ○ ColdArchive **Note** Make sure that the storage class you want to use is enabled on the destination OSS bucket.	No
-o/-overWrite	The destination path is forcibly cleared. For a partitioned table, only the destination path of the partition that you want to migrate is cleared.	No
-r/-removeSource	After data is migrated and metadata is updated, the source path is cleared. For a partitioned table, only the source path of the partition that is migrated is cleared.	No
-skipTrash	The trash is skipped when the source path is cleared. **Note** You can specify this option only if -r/-removeSource is specified.	No
-e/-explain	The explain mode is used. In explain mode, the list of partitions to be migrated is displayed, but no data is migrated.	No
-l/-logDir <log directory>	The directory in which log files are stored.	No

Configure a lock directory

The MoveTo command supports process locks. You must use an HDFS path to store lock files. The default path is *hdfs:///tmp/jindotable-lock/*.

Notice The path must be an HDFS path. If you have no permission on the default path, you can perform the following steps to customize a path.

1. Go to the HDFS service page.
   i.

ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.

iii. Click the **Cluster Management** tab.

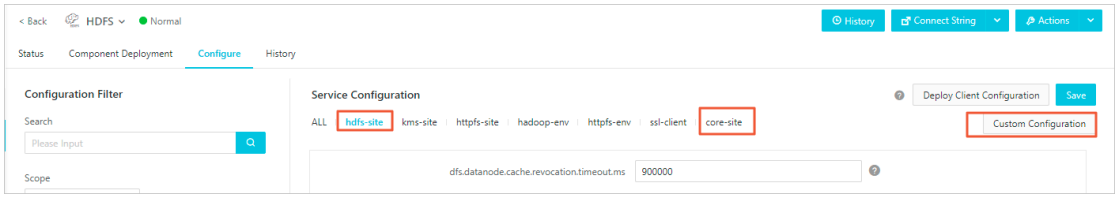
iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.

v. In the left-side navigation pane of the Cluster Overview page, choose **Cluster Service > HDFS**.

2. Add a custom item.

i. Click the **Configure** tab. Then, click the **hdfs-site** or **core-site** tab in the Service Configuration section.

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**.



iii. In the **Add Configuration Item** dialog box, add the **jindotable.moveto.tablelock.base.dir** parameter and set it to an existing HDFS path.

> **Notice** When you customize a lock directory, make sure that no MoveTo process is running on the nodes of the cluster. Otherwise, the MoveTo process may fail, which may even cause data pollution.

3. Save the configuration.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and click **OK**.

5.4.5. Use JindoTable to collect access frequency statistics on tables and partitions

You can use JindoTable to collect access frequency statistics on tables and partitions, and separate cold and hot data based on these statistics. This helps you reduce storage costs and improve cache usage efficiency.

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

JindoTable can be used to collect access frequency statistics on Hive tables. The collected data is saved in the namespaces of the SmartData service of a cluster.

In SmartData 3.2.X and later, JindoTable can be used to collect access frequency statistics on Spark, Hive, and Presto tables. This feature is enabled for Spark and Presto by default. If you want to disable the feature for Spark or Presto, perform the operations described in Disable the collection of access frequency statistics. The feature is disabled for Hive by default. If you want to enable the feature for Hive, perform the operations described in Enable the collection of access frequency statistics for Hive.

Query access frequency statistics

You can run a command to query access frequency statistics.

- Syntax

```
jindo table -accessStat <-d [days]> <-n [topNums]>
```

Set `days` and `topNums` to positive integers. For example, if days is set to 1 and topNums is not specified, the access frequency statistics of all the tables or partitions that are visited on the current day (from 00:00 to the current time) are queried.

- Description

This command is used to query the first N tables or partitions that are visited the most frequently within a specific period of time, and the number of times each of them is visited.

- For example, you can run the following command to query the first 20 tables or partitions that have been visited the most frequently within the last seven days, and the number of times each of them has been visited:

```
jindo table -accessStat -d 7 -n 20
```

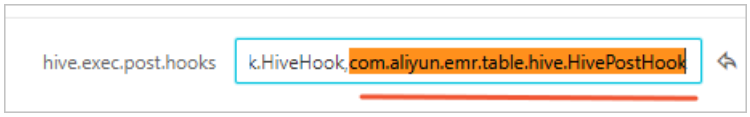
For more information about how to use JindoTable, see Use JindoTable.

Enable the collection of access frequency statistics for Hive

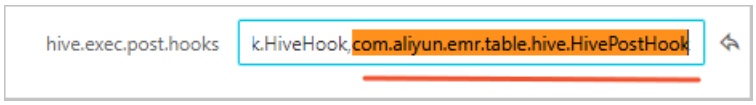
1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of a Hive parameter.
   i. In the left-side navigation pane, choose **Cluster Service > Hive**.
   ii. On the Hive service page, click the **Configure** tab.
   iii. Search for the hive.exec.post.hooks parameter and add **com.aliyun.emr.table.hive.HivePostHook** to the end of the existing parameter value. Separate the hooks with a comma (,).
6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   i. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
   ii. In the **Cluster Activities** dialog box, specify **Description**.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Disable the collection of access frequency statistics

1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of the related parameter.
   - Hive:
     a. In the left-side navigation pane, choose **Cluster Service > Hive**.
     b. On the Hive service page, click the **Configure** tab.
     c. Search for the hive.exec.post.hooks parameter and delete **com.aliyun.emr.table.hive.HivePostHook** from the parameter value.



   - Spark:
     a. In the left-side navigation pane, choose **Cluster Service > Spark**.
     b. On the Spark service page, click the **Configure** tab.
     c. Search for the spark.sql.queryExecutionListeners parameter and delete **com.aliyun.emr.table.spark.SparkSQLQueryListener** from the parameter value.



   - Presto:
     a. In the left-side navigation pane, choose **Cluster Service > Presto**.
     b. On the Presto service page, click the **Configure** tab.
     c. Search for the event-listener.name parameter and delete the parameter value.
6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   - Hive:
     a. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
     b. In the **Cluster Activities** dialog box, specify **Description**.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.
   - Spark:

a. In the upper-right corner of the Spark service page, choose **Actions > Restart ThriftServer**.
b. In the **Cluster Activities** dialog box, specify **Description**.
c. Click **OK**.
d. In the **Confirm** message, click **OK**.

- Presto:
  a. In the upper-right corner of the Presto service page, choose **Actions > Restart All Components**.
  b. In the **Cluster Activities** dialog box, specify **Description**.
  c. Click **OK**.
  d. In the **Confirm** message, click **OK**.

5.4.6. Use JindoTable to collect infrequent-access statistics of tables and partitions

JindoTable can be used to collect infrequent-access statistics of tables and partitions. It enables you to find the data that is not accessed recently based on the last access time of tables and partitions. This helps optimize data storage and reduce costs. For example, in data analytics scenarios, you can use this feature to move infrequently used partitions to lower-cost storage media.

Prerequisites

A cluster is created in E-MapReduce (EMR) V3.35.0 or later V3.X.X, or in EMR V4.9.0 or later V4.X.X. For more information, see Create a cluster.

Context

In SmartData 3.5.X and later, JindoTable can be used to collect infrequent-access statistics of Hive, Spark, and Presto tables. This feature is disabled by default. For more information about how to enable this feature, see Enable the collection of infrequent-access statistics for Spark, Enable the collection of infrequent-access statistics for Hive, and Enable the collection of infrequent-access statistics for Presto.

> **Note** The same hooks and listeners are used to collect frequent-access and infrequent-access statistics. Therefore, if you enable the collection of infrequent-access statistics, the collection of frequent-access statistics is also enabled. For more information about the collection of frequent-access statistics, see Use JindoTable to collect access frequency statistics on tables and partitions.

Limits

- Data Lake Formation (DLF) metadata is not supported.
- You must have the permissions to access the underlying metadata storage, such as MySQL or ApsaraDB RDS, from the IP addresses of the nodes on which the Hive CLI, HiveServer2, Spark SQL CLI, Spark Thrift Server, and Presto services reside.
- You can use JindoTable to collect infrequent-access statistics only of Hive, Spark, and Presto tables.

Query infrequent-access statistics

You can run a command to query infrequent-access statistics.

- Syntax

```
jindo table -leastUseStat -n <num> [-i/-ignoreNever]
```

num indicates the number of tables or partitions that you want to query. Set this parameter to a positive integer. -i/-ignoreNever is an optional parameter. If you specify this parameter, the tables or partitions that have never been accessed are filtered out.

- Description

This command is used to query the tables or partitions that have not been accessed for the longest time.

- Example: Query the first 20 tables or partitions that have not been accessed for the longest time.

```
jindo table -leastUseStat -n 20
```

The following figure shows the output.

```
tdb.t1      pid=20/qid=ten      2021-03-26 13:53:52
tdb.t2                          2021-03-26 13:53:58
```

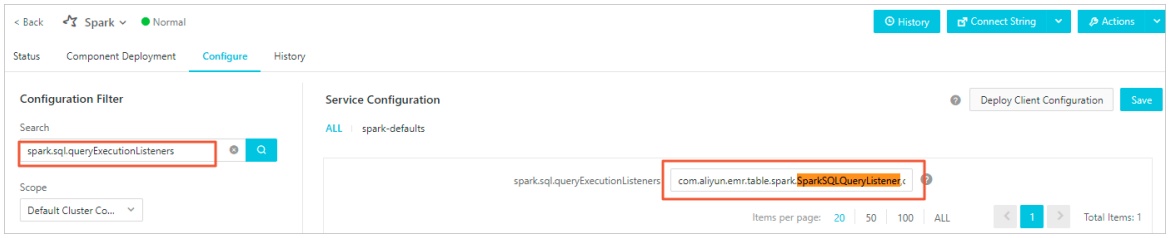
  - The first column lists tables in the Database name.Table name format.
  - The second column lists partitions in the First partition key column=Column value/Second partition key column=Column value/... format. For a non-partitioned table, this column is empty.
  - The third column displays the last access time of the tables or partitions in the yyyy-MM-dd HH:mm:ss format.

> **Note** For a partitioned table, only partition-level access statistics are displayed. Table-level access statistics are not displayed.

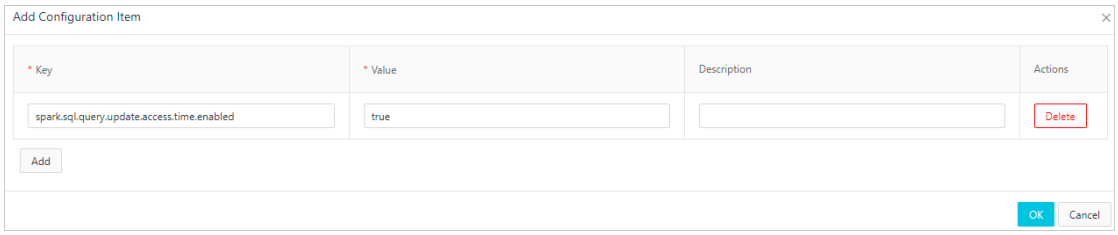
For more information about how to use JindoTable, see Use JindoTable.

Enable the collection of infrequent-access statistics for Spark

1. Go to the Spark service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Spark**.
2. On the Spark service page, click the **Configure** tab.
3. Search for the spark.sql.queryExecutionListeners parameter and check whether the value contains com.aliyun.emr.table.spark.SparkSQLQueryListener. If the value does not contain this listener, add the listener to the value. Separate multiple listeners with comma(,).



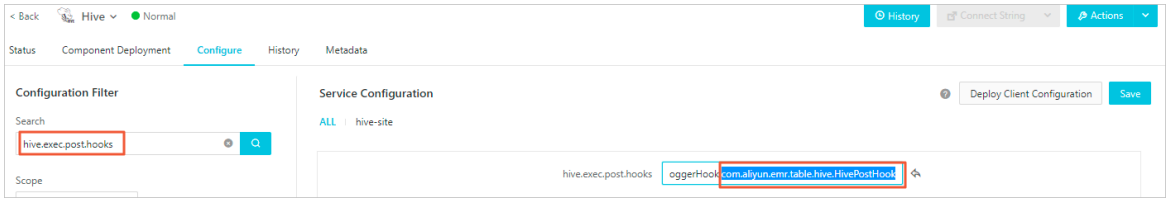
4. Add custom configurations.
   i. In the **Service Configuration** section, click the **spark-defaults** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **spark.sql.query.update.access.time.enabled** parameter and set it to **true**.



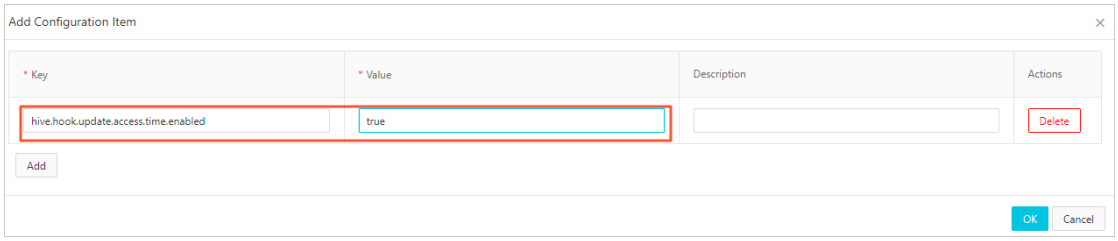
   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

Enable the collection of infrequent-access statistics for Hive

1. Go to the Hive service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Hive**.
2. On the Hive service page, click the **Configure** tab.
3. Search for the hive.exec.post.hooks parameter and check whether the value contains com.aliyun.emr.table.hive.HivePostHook. If the value does not contain this hook, add the hook to the value. Separate multiple hooks with comma(,).



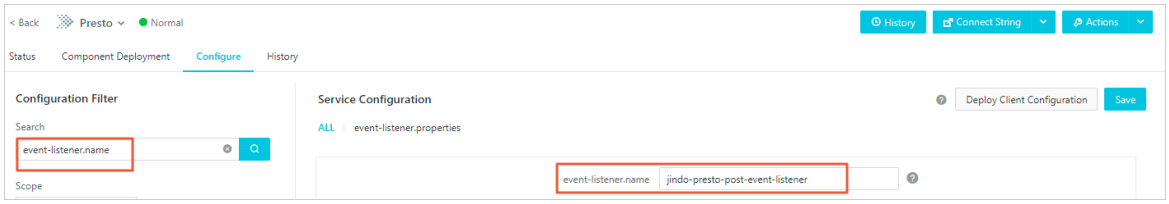
4. Add custom configurations.
   i. In the **Service Configuration** section, click the **hive-site** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **hive.hook.update.access.time.enabled** parameter and set it to **true**.



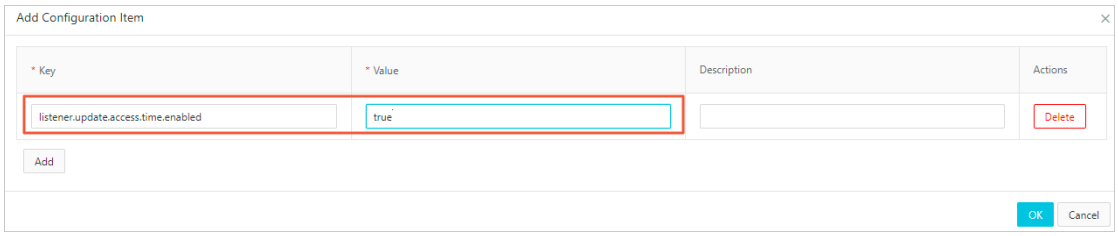
   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

Enable the collection of infrequent-access statistics for Presto

1. Go to the Presto service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Presto**.
2. On the Presto service page, click the **Configure** tab.
3. Search for the event-listener.name parameter and check whether the value contains jindo-presto-post-event-listener. If the value does not contain this listener, add the listener to the value.



4. Add custom configurations.
   i. In the **Service Configuration** section, click the **event-listener.properties** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **listener.update.access.time.enabled** parameter and set it to **true**.



   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

5.5. Toolset

5.5.1. Use Jindo SQL

Jindo SQL is a tool provided by JindoFS. You can use Jindo SQL to analyze metadata, JindoFS access logs, and Object Storage Service (OSS) access logs. This topic describes how to use Jindo SQL to analyze metadata, JindoFS access logs, and OSS access logs.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

You can use Jindo SQL to perform the following operations:

- Analyze JindoFS access logs
- Analyze metadata
- Analyze OSS access logs

Limits

Jindo SQL is supported only in EMR V3.36.0 and later minor versions, and EMR V5.2.0 and later minor versions.

Start Jindo SQL

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to start Jindo SQL:

```
jindo sql
```

The following table describes the common parameters that can be used in the jindo sql command.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically initializes the SQL script after the jindo sql command is run.
-d	Specifies parameters in key-value pairs, such as `-d A=B`.

Structures of built-in tables in Jindo SQL

- audit_log_source (partitioned table)

  The audit_log_source table is used to store raw access logs of JindoFS.

Parameter	Description
datetime	The time format is yyyy-MM-dd HH:mm:ss.
allowed	Indicates whether the current operation is allowed. Valid values: ○ true: The operation is allowed. ○ false: The operation is not allowed.
ugi	The user who performs the operation. Information about the authentication method is provided.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dst	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Parameter	Description
date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

- audit_log

  The audit_log table is used to store JindoFS access logs. You can use the partition key column to filter data in the table.

Parameter	Description
datetime	The time format is yyyy-MM-dd HH:mm:ss.
allowed	Indicates whether the current operation is allowed. Valid values: ○ true: The operation is allowed. ○ false: The operation is not allowed.
ugi	The user who performs the operation. Information about the authentication method is provided.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dst	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.
date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

- fs_image (partitioned table)

  The fs_image table is used to dump image information.

Parameter	Description
atime	The time when the current inode is last accessed.
attr	File-related attributes.
etag	The ETag value of OSS.

Parameter	Description
id	The ID of the inode.
mtime	The time when the inode is modified.
name	The name of the inode.
owner	The name of the owner.
ownerGroup	The name of the group to which the owner belongs.
parentId	The ID of the parent node.
permission	The operation permissions on the file.
size	The size of the inode.
state	The status of the inode.
type	The type of the inode.
storagePolicy	The storage policy.
namespace (partition key column)	The name of the namespace.
datetime (partition key column)	The time when the information is dumped.

- oss_access_log_source

  If the table partitioning mode is enabled, the oss_access_log_source table is a partitioned table. This table is used to store raw access logs of OSS.

Parameter	Description
line	A raw log entry.
bucket (partition key column)	The name of the bucket.
partition_date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

- oss_access_log

  If the table partitioning mode is enabled, you can use the partition key columns to filter data in the oss_access_log table. This table is used to store OSS access logs.

Parameter	Description
Remote_IP	The IP address of the requester.
Reserved	A reserved field. Fixed value: -.
Reserved1	A reserved field. Fixed value: -.

Parameter	Description
Time	The time when OSS receives the request.
Request_URI	The request URL. It contains a query string. OSS ignores the query string parameter that starts with x-. However, this parameter is recorded in access logs. Therefore, you can use the query string parameter that starts with x- as the tag of a request. Then, you can use this tag to query the log that corresponds to the request.
HTTP_Status	The HTTP status code that is returned by OSS.
SentBytes	The downstream traffic generated by the request. Unit: bytes.
RequestTime	The time required to process the request. Unit: milliseconds.
Referer	The HTTP referer of the request.
User_Agent	The user-agent header in the HTTP request.
HostName	The destination domain name in the request.
Request_ID	The ID of the request.
LoggingFlag	Indicates whether log dump is enabled.
Requester	The user ID of the requester. A value of - indicates anonymous access.
Operation	The request type.
Bucket	The name of the destination bucket.
Key	The name of the destination object.
ObjectSize	The size of the destination object. Unit: bytes.
Server_Cost_Time	The time required by OSS to process the request. Unit: milliseconds.
ErrorCode	The error code that is returned by OSS. A value of - indicates that no error code is returned.
RequestLength	The size of the request. Unit: bytes.
UserID	The ID of the bucket owner.
Delta_DataSize	The variation of the bucket size. A value of - indicates that this request does not involve write operations on objects.

Parameter	Description
SyncRequest	Indicates whether the request is an Alibaba Cloud CDN back-to-origin request. Valid values: ○ cdn: The request is a CDN back-to-origin request. ○ -: The request is not a CDN back-to-origin request.
StorageClass	The storage class of the destination object. Valid values: ○ Standard ○ IA ○ Archive ○ Cold Archive ○ -: no storage class obtained
TargetStorageClass	The storage class to which the object is converted after a lifecycle rule is triggered or the CopyObject operation is called. Valid values: ○ Standard ○ IA ○ Archive ○ Cold Archive ○ -: no storage class conversion involved
TransmissionAccelerationAccessPoint	The accelerate endpoint that is used to access the destination bucket based on the transfer acceleration feature. A value of - indicates that no accelerate endpoint is used or the accelerate endpoint is in the same region as the destination bucket. For example, if the requester accesses the destination bucket by using the accelerate endpoint that corresponds to the China (Hangzhou) region, cn-hangzhou is used as the value of this parameter.
AccessKeyID	The AccessKey ID that is used to access the bucket.
bucket (partition key column)	The name of the bucket.
partition_date (partition key column)	The date when the log entry is generated. The format is YYYY-mm-DD.

Analyze JindoFS access logs

JindoFS allows you to use SQL queries to analyze the JindoFS access logs that are stored in OSS. You can use SQL queries to analyze the most active commands or IP addresses in the related tables. You can use the `jindo sql` command to analyze the data.

> **Note** The AuditLog feature is enabled. For more information, see Use Jindo AuditLog.

You can use the following commands related to Jindo SQL:

- Run the following command to query tables:

```
show tables;
```

> **Note** For more information about the table structures, see Structures of built-in tables in Jindo SQL.

The information shown in the following figure is returned.

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to query partitions:

```
show partitions audit_log_source;
```

Information similar to the output in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

Information similar to the output in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

Information similar to the output in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics about the use frequencies of different commands that were run on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

Analyze metadata

JindoFS allows you to use SQL queries to analyze the metadata files that are stored in JindoFS. You can use SQL queries to analyze the related tables. You can use the `jindo sql` command to analyze the metadata files.

> **Note** The AuditLog feature is enabled. For more information, see Use Jindo AuditLog.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to start Jindo SQL:

   ```
   jindo sql
   ```

3. Query the tables that can be analyzed by using Jindo SQL.
   - Run the `show tables` command to query the tables that can be analyzed. Jindo SQL provides two built-in modules: audit_log and fs_image, which are used for auditing and metadata analysis.

     The following figure shows an example.

     ```
     [root@emr-header-1 ~]# jindo sql
     Spark master: yarn, Application Id: application_1603081647416_0050
     jindo-sql> show tables;
     database        tableName       isTemporary
     default audit_log       false
     default audit_log_source        false
     default fs_image        false
     Time taken: 0.33 seconds, Fetched 3 row(s)
     ```

   - Run the `show partitions fs_image` command to view the partition information of the

fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

4. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287249    1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       15464684820176650002    1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The following figure shows an example on how to obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

   Note The namespace and datetime columns are added. The namespace column contains the namespace name, and the datetime column contains the timestamp when metadata is uploaded.

Analyze OSS access logs

> **Notice** To analyze OSS access logs, you must specify a log directory and whether to enable the table partitioning mode. If you enable the table partitioning mode, OSS archives logs in the partitioned table by bucket or partition_date, and you can execute statements to query logs in a specific partition. This improves query efficiency. If you enable the partitioning table mode, you must use this mode each time you query data in the table. Otherwise, some files may be archived to the specified log directory, which causes the failure to query some data.

JindoFS allows you to use SQL queries to analyze the OSS access log files that are stored in OSS. You can use SQL queries to analyze the related tables. You can run the `jindo sql` command to analyze the log files.

> **Note** Make sure that log dump is enabled. For more information, see Logging.

1. Log on to your cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to start Jindo SQL:

   ```
   jindo sql
   ```

3. Run the following command to specify the storage path of the OSS access logs and whether the table partitioning mode is enabled.

   ```
   jindo sql -d access_log_path=oss://test-sh/oss-accesslog -d partition.table.enabled=tru
   e
   ```

   `access_log_path` specifies the storage path, and `partition.table.enabled` specifies whether the table partitioning mode is enabled.

FAQ

- Q: How do I modify the startup parameters of Jindo SQL?

  `Jindo SQL` is developed based on Spark. Therefore, the initial resources of Jindo SQL may be insufficient. If you want to increase the resources of Jindo SQL, you can modify the startup parameters of Jindo SQL by using the environment variable `JINDO_SPARK_OPTS`.

  ```
  export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
  --conf spark.executor.cores=5 --conf spark.executor.memory=20G"
  ```

- Q: How do I use Hive to analyze tables?

  Tables in the default database are invisible to Hive jobs. This prevents Hive metadata from being polluted. If you want to use Hive to analyze the tables, you can execute the `show create table {t able_name}` statement to query the statement that is executed to create a specific table or execute an SQL statement to create a table. In addition, you must use Hive to load external tables.

5.5.2. Use JindoFS FUSE

This topic describes how to access JindoFileSystem (JindoFS) from a Filesystem in Userspace (FUSE) client. FUSE allows you to use JindoFS in block storage mode and cache mode (only JFS Scheme supported).

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

FUSE is a method provided by the Linux kernel to mount a file system. You can use a FUSE client to map files in a JindoFS cluster to a local disk. Then, you can directly access data in the JindoFS cluster without the need to use the `hadoop fs -ls jfs://<namespace>/` command.

Mount a directory

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs
   ```

3. Run the following command to mount the directory:

   ```
   jindofs-fuse /mnt/jfs
   ```

   In this example, the */mnt/jfs/* directory is mounted.

Read and write files

1. Run the following command to query all the sub-directories under the /mnt/jfs/ directory:

   ```
   ls /mnt/jfs/
   ```

   All the namespaces that you have configured on the server are returned.

   ```
   test  testcache
   ```

2. Run the following command to query all the files in the test namespace:

   ```
   ls /mnt/jfs/test/
   ```

3. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs/test/dir1
   ls /mnt/jfs/test/
   ```

4. Run the following command to write data into a file:

   ```
   echo "hello world" > /tmp/hello.txt
   cp /tmp/hello.txt /mnt/jfs/test/dir1/
   ```

5. Run the following command to read data from a file:

   ```
   cat /mnt/jfs/test/dir1/hello.txt
   ```

   The following information is returned:

   ```
   hello world
   ```

The following example shows how to read data from and write data into files in Python:

1. Write data into a file in Python. Create the *write.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'w',encoding = 'utf-8') as f:
   f.write("my first file\n")
   f.write("This file\n\n")
   f.write("contains three lines\n")
```

2. Read data from a file in Python. Create the *read.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'r',encoding = 'utf-8') as f:
    lines = f.readlines()
    [print(x, end = '') for x in lines]
```

Run the following command to read the data that is written into the *test.txt* file:

```
[hadoop@emr-header-1 ~]$ ./read.py
```

The following information is returned:

```
my first file
This file
```

Unmount FUSE

Note Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to unmount FUSE:

```
umount jindofs-fuse
```

If the `target is busy` error is returned, switch to another directory, stop all the programs that are reading data from and writing data into FUSE files, and then unmount FUSE.

5.5.3. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain help information:

```
jindo distcp --help
```

The following information is returned:

```
     --help           - Print help text
     --src=VALUE          - Directory to copy files from
     --dest=VALUE              - Directory to copy files to
     --parallelism=VALUE         - Copy task parallelism
     --outputManifest=VALUE       - The name of the manifest file
     --previousManifest=VALUE   -   The path to an existing manifest file
     --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
if specified
     --copyFromManifest   -   Copy from a manifest instead of listing a directory
     --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
     --srcPattern=VALUE   -   Include only source files matching this pattern
     --deleteOnSuccess   -   Delete input files after a successful copy
     --outputCodec=VALUE   -   Compression codec for output files
     --groupBy=VALUE   -   Pattern to group input files by
     --targetSize=VALUE   -   Target size for output files
     --enableBalancePlan   -   Enable plan copy task to make balance
     --enableDynamicPlan   -   Enable plan copy task dynamically
     --enableTransaction   -   Enable transation on Job explicitly
     --diff   -   show the difference between src and dest filelist
     --ossKey=VALUE   -   Specify your oss key if needed
     --ossSecret=VALUE   -   Specify your oss secret if needed
     --ossEndPoint=VALUE   -   Specify your oss endPoint if needed
     --policy=VALUE   -   Specify your oss storage policy
     --cleanUpPending   -   clean up the incomplete upload when distcp job finish
     --queue=VALUE   -   Specify yarn queuename if needed
     --bandwidth=VALUE   -   Specify bandwidth per map/reduce in MB if needed
     --s3Key=VALUE   -   Specify your s3 key
     --s3Secret=VALUE   -   Specify your s3 Sercet
     --s3EndPoint=VALUE   -   Specify your s3 EndPoint
```

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-
01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03
/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03
/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017
-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-0
1/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-0
1/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_tab
le/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/
03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/
03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromMani
fest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

> **Note** If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

--queue

--queue specifies the name of the YARN queue where the current DistCp task resides.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<your_bucket>/hourly_table --qu
eue yarnqueue
```

--bandwidth

--bandwidth specifies the bandwidth allocated to the current DistCp task. This prevents the task from using excessive bandwidth. Unit: MB/s.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.c
om --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy archive --parallelism 20
  ```

- Example of the IA storage class (`--ia`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy ia --parallelism 20
  ```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

Use Amazon S3 as a data source

You can use the --s3Key, --s3Secret, and --s3EndPoint parameters in a command to specify information related to Amazon S3.

Example:

```
jindo distcp jindo-distcp-2.7.3.jar --src s3a://yourbucket/ --dest oss://<your_bucket>/hour
ly_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com
```

You can configure the s3Key, s3Secret, and s3EndPoint parameters in the *core-site.xml* file of Hadoop. This way, you do not need to specify an AccessKey pair each time.

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

Example:

```
jindo distcp /tmp/jindo-distcp-2.7.3.jar --src s3://smartdata1/ --dest s3://smartdata1/tmp
--s3EndPoint  s3-us-west-1.amazonaws.com
```

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

> **Note** If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

5.5.4. Use Jindo DistCp in specific scenarios

This topic describes how to use Jindo DistCp in specific scenarios.

Prerequisites

- An E-MapReduce (EMR) cluster of a required version is created. For more information, see Create a cluster.
- Java Development Kit (JDK) 1.8 is installed.
- *jindo-distcp-<version>.jar* is downloaded based on your Hadoop version.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 2.7 and later 2.X versions.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 3.X.

Use scenarios

Common use scenarios of Jindo DistCp:

- Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?
- Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?
- Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?
- Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?
- Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file size?
- Scenario 8: What parameters are required if Amazon S3 is used as a data source?
- Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?
- Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?
- Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?
- Scenario 12: What parameters are required if I want to delete original files after a copy operation?
- Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?

If you are not using EMR, the following conditions must be met before you can import HDFS data to Object Storage Service (OSS):

- You can read data from HDFS.
- The AccessKey ID, AccessKey secret, and endpoint of OSS are obtained. You can write data to a destination OSS bucket.
- The storage class of the OSS bucket is not Archive.
- You can submit MapReduce tasks.
- The JAR package of Jindo DistCp is downloaded.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 10
```

> **Note** For more information about the parameters, see Use Jindo DistCp.

To import a large amount of data or a large number of files, such as millions or tens of millions of files, to OSS, you can set parallelism to a large value to increase the concurrency. You can also set `enableBatch` to true for optimization. Optimization command:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 500 --enableBatch
```

Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?

You can use one of the following methods to verify data integrity:

- Jindo DistCp Counters

  Check DistCp Counters in the counter information of a MapReduce task.

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

where:

- Bytes Destination Copied specifies the size of the file that you copy to a destination directory, in bytes.
- Bytes Source Read specifies the size of the file that you read from the source directory, in bytes.
- Files Copied specifies the number of copied files.

- Jindo DistCp --diff

  Use `--diff` to compare the information of files in the source and destination directories. The information includes file name and file size. Missing files and the files that fail to be copied are recorded in a manifest file. The manifest file is generated in the directory where you perform the comparison.

  Add `--diff` to the command described in Scenario 1. Example:

  ```
  hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
  -hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
  aliyuncs.com --diff
  ```

  If all files are copied, the following information is returned:

  ```
  INFO distcp.JindoDistCp: distcp has been done completely
  ```

Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?

If your DistCp task fails, and you want to perform resumable upload to copy only the files that fail to be copied, perform the following operations on the basis of the command described in Scenario 1:

1. Add `--diff` to check whether all files are copied.

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
   hou.aliyuncs.com --diff
   ```

   If all the files are copied, the following information is returned. Otherwise, a manifest file is generated. In this case, go to the next step.

   ```
   INFO distcp.JindoDistCp: distcp has been done completely.
   ```

2. Use `--copyFromManifest` and `--previousManifest` to copy the remaining files. Example:

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --dest oss://yang-hhht/hourly_table --previousManifest=file:///opt
   /manifest-2020-04-17.gz --copyFromManifest --parallelism 20
   ```

   `file:///opt/manifest-2020-04-17.gz` is the path where the manifest file is stored.

Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?

1. If no information is generated for the files that were copied last time, specify parameters to

generate a manifest file to record the information about the files that are copied.

Add `--outputManifest=manifest-2020-04-17.gz` and `--requirePreviousManifest=false` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

where:

- `--outputManifest` generates a manifest file whose name can be customized and must have a file name extension of gz, such as *manifest-2020-04-17.gz*. This file is stored in the directory specified by `--dest`.
- `--requirePreviousManifest` specifies whether to require a previous manifest file.

2. After a DistCp task is complete, copy the new files that may be generated in the source directory.

   Add `--outputManifest=manifest-2020-04-17.gz` and `--previousManifest=oss://yang-hhht/hourly_table/manifest-2020-04-17.gz` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

```
hadoop jar jindo-distcp-2.7.3.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --outputManifest=manifest-2020-04-18.gz --previousManifest=oss://yang-hhht
/hourly_table/manifest-2020-04-17.gz --parallelism 10
```

3. Repeat Step 2 to continuously copy incremental files.

Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?

Add the following two parameters to the command described in Scenario 1. These two parameters can be used together or separately.

- `--queue`: the name of the YARN queue.
- `--bandwidth`: the size of the specified bandwidth , in MB.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --queue yarnqueue --bandwidth 6 --parallelism 10
```

Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?

- To write data to OSS Archive storage, add `--archive` to the command described in Scenario 1.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --archive --parallelism 20
```

- To write data to OSS Infrequent Access (IA) storage, add `--ia` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --ia --parallelism 20
```

Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file sizes?

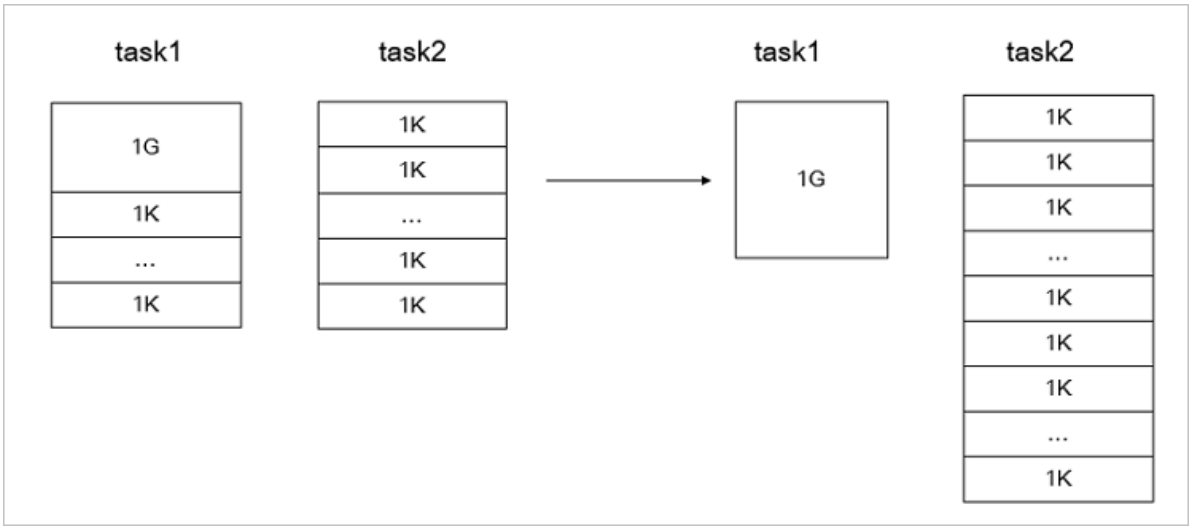
- A large number of small files and large data volume for a single large file

  If most of the files that you want to copy are small files, and the data volume of a single large file is large, the general solution is to randomly allocate the files to be copied. In this case, if you do not optimize the job allocation plan, few large files and many small files may be allocated to the same copy process. This cannot achieve excellent copy performance.

  Add `--enableDynamicPlan` to the command described in Scenario 1 to enable the optimization. --enableDynamicPlan cannot be used with `--enableBalancePlan`. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableDynamicPlan --parallelism 10
```

  The following figure shows the data copy before and after the plan is optimized.

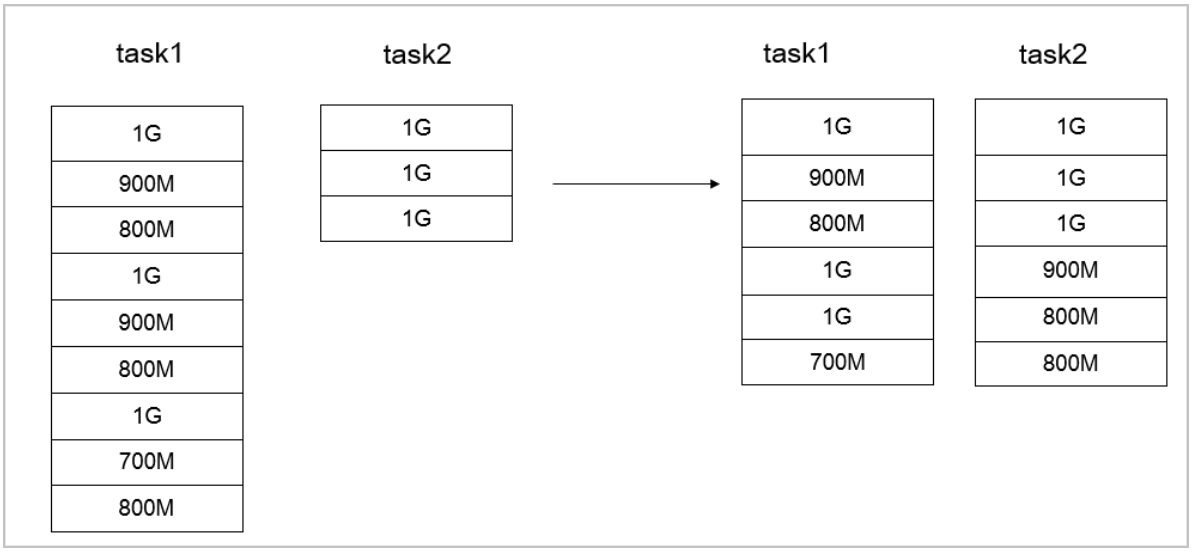


- No significant differences in sizes of files

  If the sizes are not significantly different among the files that you want to copy, use `--enableBalancePlan` to optimize the job allocation plan. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableBalancePlan --parallelism 10
```

The following figure shows the data copy before and after the plan is optimized.



Scenario 8: What parameters are required if Amazon S3 is used as a data source?

Replace the parameters that specify the AccessKey ID, AccessKey secret, and endpoint of OSS in the command described in Scenario 1 with the following parameters of Amazon S3:

- `--s3Key` : the AccessKey ID for Amazon S3
- `--s3Secret` : the AccessKey secret for Amazon S3
- `--s3EndPoint` : the endpoint for Amazon S3

Example:

```
hadoop jar jindo-distcp-<version>.jar --src s3a://yourbucket/ --dest oss://yang-hhht/hourly
_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com --para
llelism 10
```

Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?

You can use `--outputCodec` to compress copied files in a format such as LZO or GZ to reduce the space that is used to store the files.

Add `--outputCodec` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --outputCodec=gz --parallelism 10
```

You can set --outputCodec to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files are compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change to the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?

- If you want to copy the files that meet specific rules, add `--srcPattern` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPattern .*\.log --parallelism 10
```

  `--srcPattern`: specifies a regular expression that filters files for the copy operation.

- If you want to copy the files stored in some sub-directories of the same parent directory, add `--srcPrefixesFile` to the command described in Scenario 1.

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

  `--srcPrefixesFile`: enables Jindo DistCp to copy files in multiple folders under the same parent directory at a time.

  Content of the *folders.txt* file:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?

Add the following parameters to the command described in Scenario 1:

- `--targetSize`: the maximum size of the merged file, in MB
- `--groupBy`: the merging rule, which is a regular expression

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Scenario 12: What parameters are required if I want to delete original files after a copy operation?

Add `--deleteOnSuccess` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --deleteOnSuccess --parallelism 10
```

Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Jindo DistCp allows you to save the AccessKey ID, AccessKey secret, and endpoint of OSS in the *core-site.xml* file so that you do not need to repeatedly specify the information.

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of OSS, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.jfs.cache.oss-accessKeyId</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-accessKeySecret</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-endpoint</name>
        <value>oss-cn-xxx.aliyuncs.com</value>
    </property>
</configuration>
```

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of Amazon S3, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

5.5.5. Use tiered storage commands of JindoFS

In E-MapReduce (EMR) V3.30, JindoFS provides a tiered storage feature. This feature allows you to store cold and hot data in different storage media. This helps reduce the data storage costs or accelerate data access.

Use jindo jfs

Run the following command to obtain the help information:

```
jindo jfs -help archive
-archive -i/a <path> ... :
  Archive commands.
```

Tiered storage commands in JindoFS are asynchronous and start only related tasks.

Common commands:

- Cache
- Uncache
- Archive
- Unarchive
- Status
- ls2 command

Cache

You can use this command to back up data stored in a specific path to local disks. Then, you can read the data from local disks, without the need to read data from Object Storage Service (OSS).

```
jindo jfs -cache -p <path>
```

`-p` can be used to ensure that local data is not cleared based on disk usage.

Uncache

You can use this command to delete backup data from local disks and store data only to OSS Standard storage.

```
jindo jfs -uncache  <path>
```

Archive

You can use this command to delete backup data from local disks and store data to OSS in Infrequent Access (IA) or Archive mode. For information about the storage classes, see Overview.

```
jindo jfs -archive -i|-a <path>
```

`-i` is used to store data to OSS in IA mode. `-a` is used to store data to OSS in Archive mode.

Unarchive

You can use this command to restore data from the Archive mode to the IA or Standard mode. You can temporarily restore data stored in Archive mode to allow the data to be readable within one day.

```
jindo jfs -unarchive -i/-o <path>
```

`-i` is used to store data to OSS in IA mode. `-o` is used to temporarily restore data stored in Archive mode to allow the data to be readable.

Status

You can use this command to view task details. By default, the number of files to be subject to tiered storage in a specific directory and the data that has been subject to tiered storage are measured.

```
jindo jfs -status -detail/-sync <path>
```

`-detail` is used to view the storage progress of file data. `-sync` indicates that the command exits after a tiered storage task is completed.

ls2 command

JindoFS provides the ls2 command that allows you to view the file storage status on the basis of Hadoop ls commands:

```
hadoop fs -ls2 <path>
```

Example of command output, which includes the file storage class:

```
drwxrwxrwx  - -         0    2020-06-05 04:27 oss://xxxx/warehouse
-rw-rw-rw-  1 Archive   1484 2020-09-23 16:40 oss://xxxx/wikipedia_data.csv
-rw-rw-rw-  1 Standard  1676 2020-06-07 20:04 oss://xxxx/wikipedia_data.json
```

6.SmartData 3.5.x

6.1. SmartData 3.5.X

SmartData is a storage service for the E-MapReduce (EMR) Jindo engine. SmartData provides centralized storage and optimized caching and computing for EMR computing engines and extends storage features. SmartData consists of JindoFS, JindoTable, and related tools. This topic describes the updates in SmartData 3.5.X.

OSS storage scalability on JindoFS

The performance of deleting Object Storage Service (OSS) directories is optimized.

JindoSDK

- If JindoSDK for Java is used, JindoSDK logs are generated as Java logs to improve diagnostic performance.
- Statistical logs of the memory used by JindoSDK are generated. You can view the logs to obtain information about the memory used by JindoSDK.

JindoTable-based computing optimization

- The native query acceleration feature is added. It accelerates queries when you use Spark, Hive, or Presto to read data from ORC or Parquet files that are stored in OSS or JindoFS. For more information, see Enable query acceleration based on a native engine.
- The infrequent-access statistics on Hive tables can be collected in JindoTable. For more information, see Use JindoTable to collect infrequent-access statistics of tables and partitions.

Other JindoFS tools

Jindo DistCp is enhanced. The CloudMonitor service can be used to monitor and alert failed tasks. The dependency on Advanced Vector Extensions (AVX) is removed. Cold Archive is supported when data is written to OSS. For more information, see Use Jindo DistCp.

6.2. JindoFS in block storage mode

6.2.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read and write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

Context

The block storage mode of JindoFS has the following features:

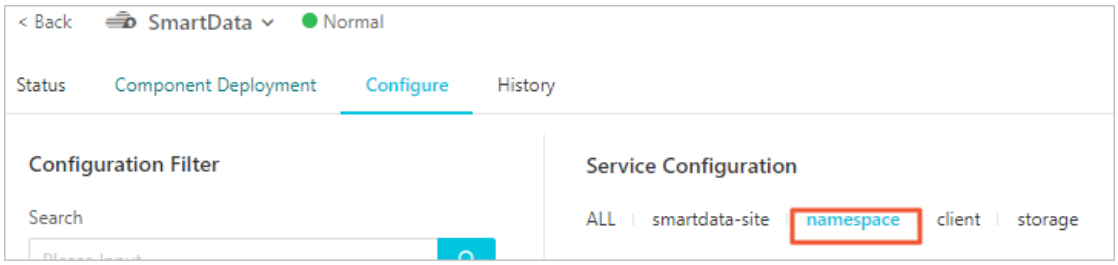
- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM)

solutions.

- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

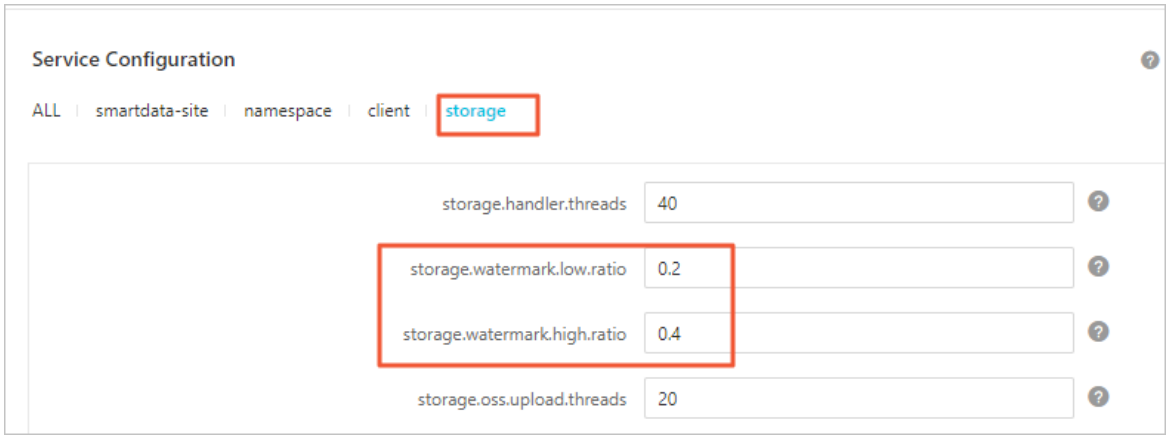
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Choose **Actions > Restart Jindo Storage Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

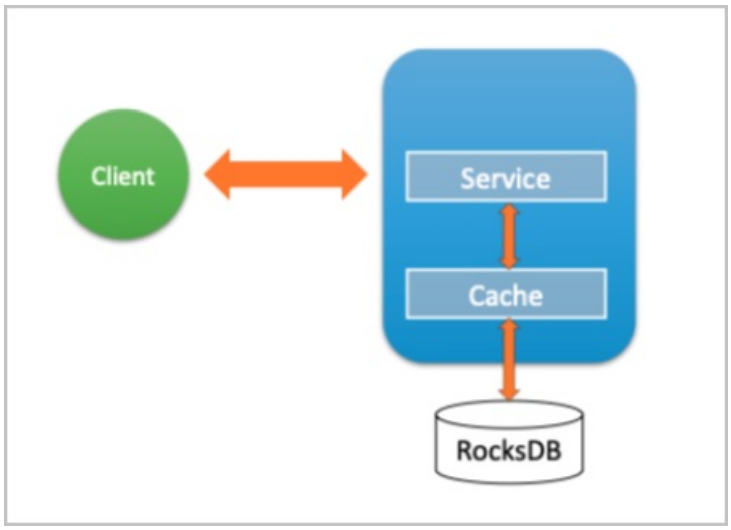
6.2.2. Use RocksDB to store metadata

Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

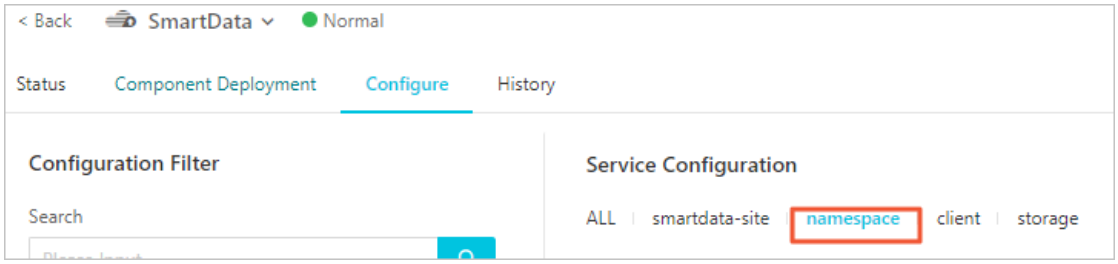
RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage backend in high availability mode, we recommend that you use Raft instances. For more information, see Use Raft-RocksDB-Tablestore to store metadata.

The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
6. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

7. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

8. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. Prepare for the recovery.

   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

   ```
   hadoop fs -count jfs://test/
   ```

   Information similar to the following example is returned:

   ```
    1596      1482809                 25 jfs://test/
   ```

   The number of folders is 1596, and the number of files is 1,482,809.

   ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1`, the latest metadata is synchronized to the Tablestore instance.

   ```
   jindo jfs -metaStatus
   ```

   Information similar to that shown in the following figure is returned.

   ```
   [hadoop@emr-header-1 ~]$ jindo jfs -metaStatus
   ==== emr-header-1:8101 ====
   OtsUploader: _synced=1
   [RocksDB Row Counts of each CF(Table)]
   ```

   iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus
```

   If the state is FINISH, the recovery is completed, as shown in the following figure.

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

6.2.3. Use Raft-RocksDB-Tablestore to store metadata

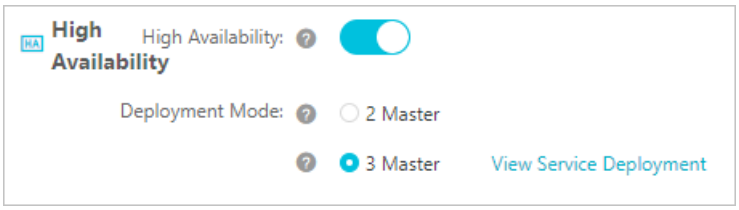
In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata.

Prerequisites

- A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  Note You must enable the transaction feature.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.

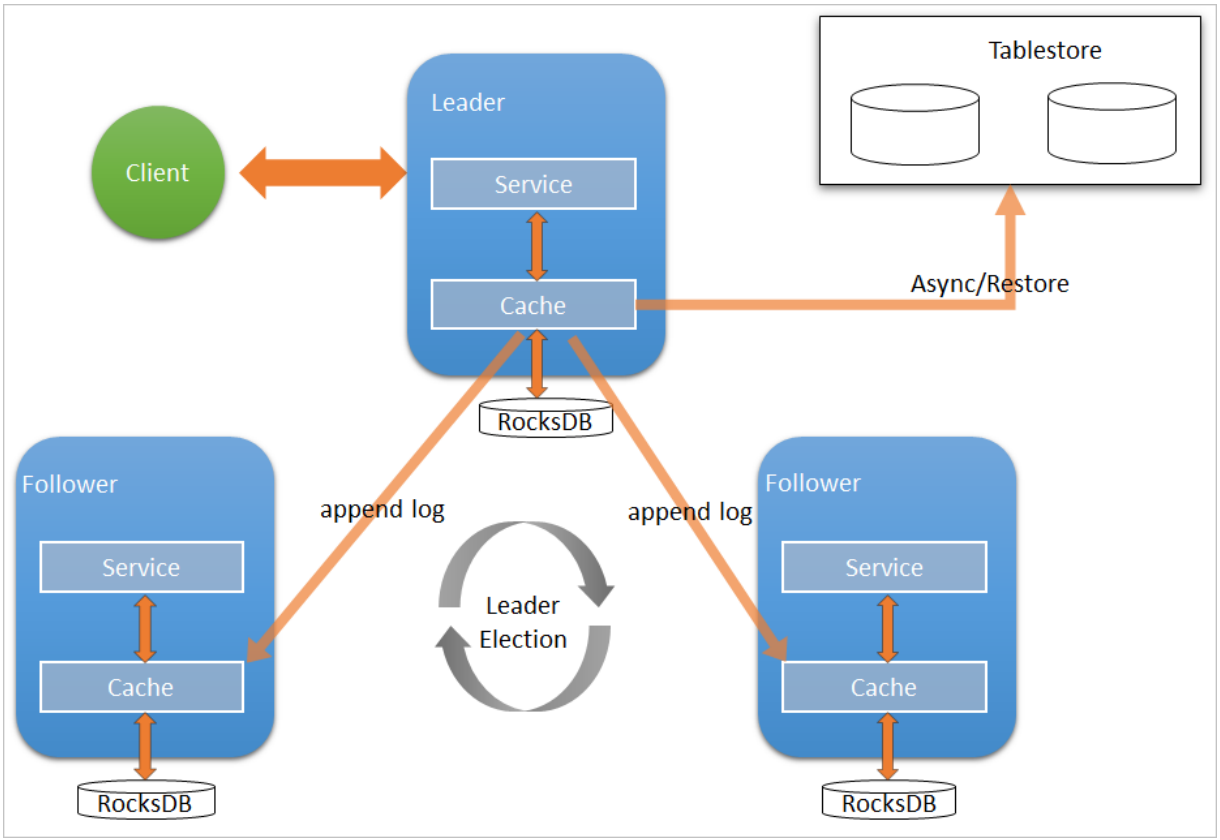


  Note If no deployment modes are available, submit a ticket.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.



Configure a Raft instance as the local storage backend

1. Suspend all the SmartData components of an EMR cluster.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   vi. Select **Stop All Components** from the **Actions** drop-down list in the upper-right corner.
2. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
3. Go to the **namespace** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **namespace** tab.
4. Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. Valid values: ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

Note If you do not need to use Tablestore for remote storage, skip Step 5.

5. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

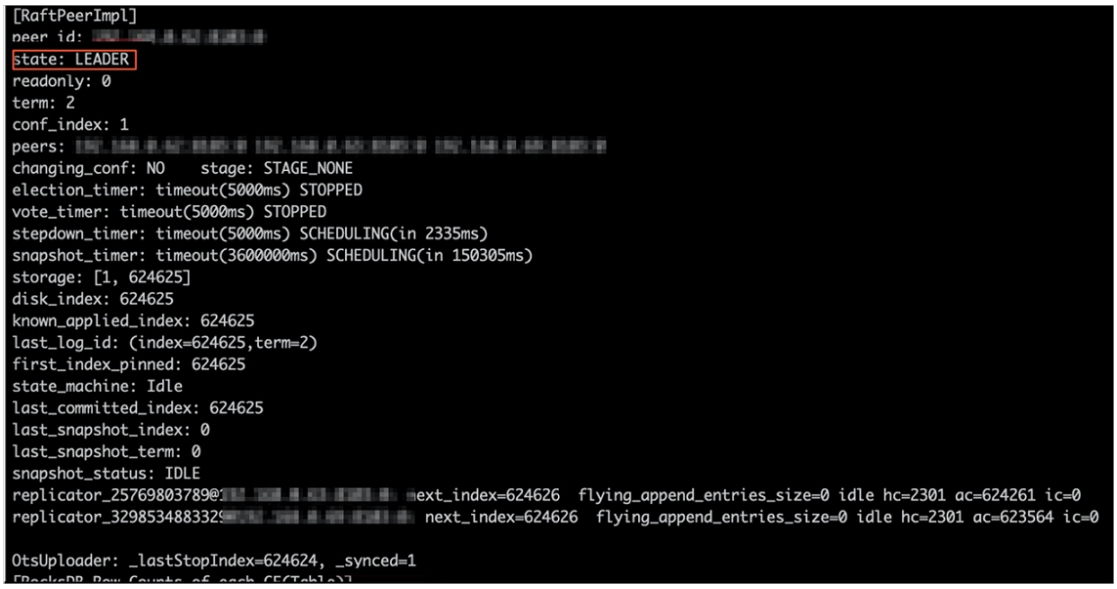
If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

```
hadoop fs -count jfs://test/
      1596      1482809                 25 jfs://test/
  (Number of folders) (Number of files)
```

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```



iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.

```
[RaftPeerImpl]
peer_id: [redacted]:8103:0
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 3382ms)
snapshot_timer: timeout(600000ms) SCHEDULING(in 474855ms)
storage: [1, 153]
disk_index: 153
known_applied_index: 153
last_log_id: (index=153,term=2)
first_index_pinned: 1
state_machine: Idle
last_committed_index: 153
last_snapshot_index: 1
last_snapshot_term: 2
snapshot_status: IDLE
replicator_1116691496965@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0
replicator_3311419785217@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0

[Recovery From OTS Status]
state: FINISH
total rows: 1484409
table `jfs_block_test` 2 rows.
table `jfs_namespace_cache_ns` 1 rows.
table `jfs_namespace_test` 1484406 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

6.2.4. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see

Create a cluster.

- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.

ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

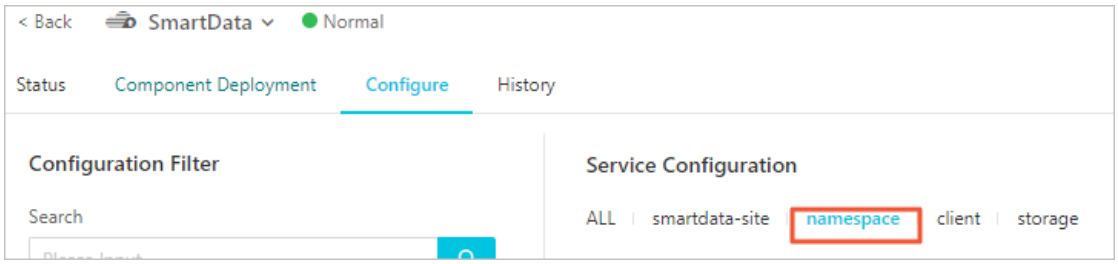
iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **namespace** tab for the SmartData service.

   i. Click the **Configure** tab.

   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:

   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

6.2.5. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

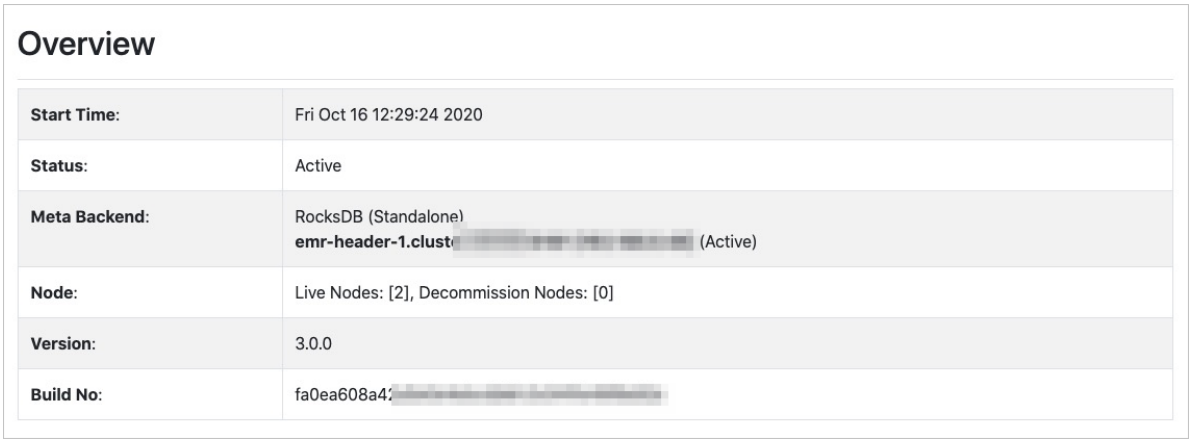
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

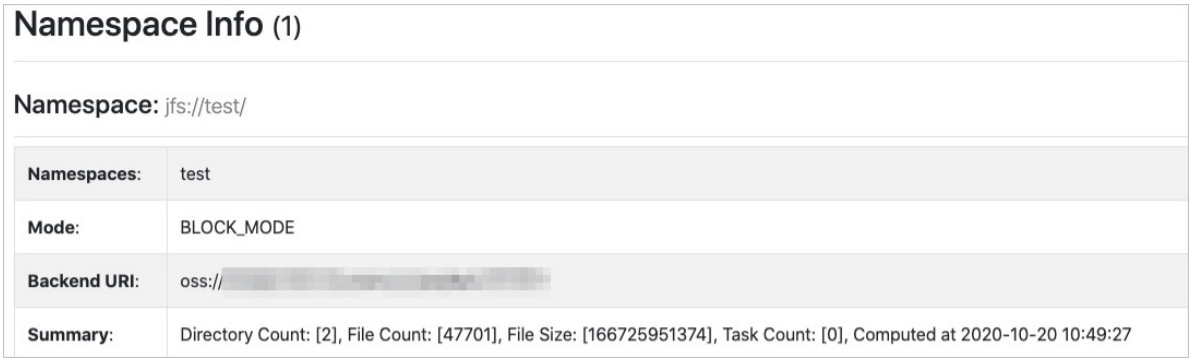


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



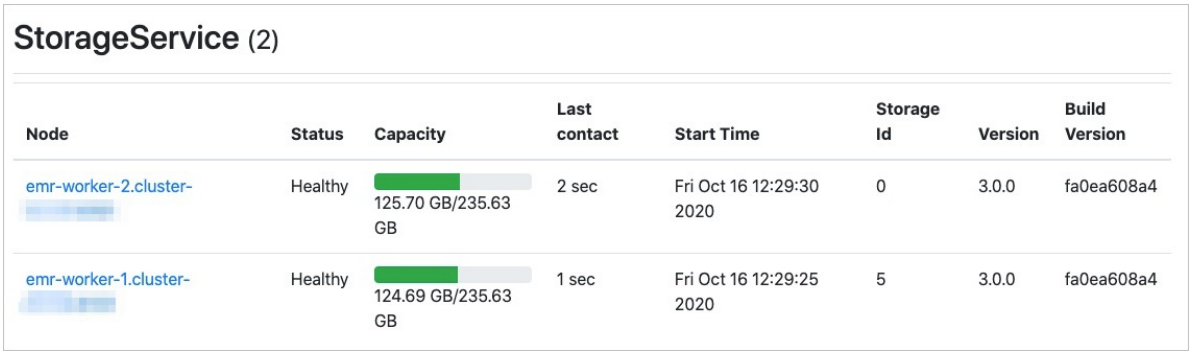
Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

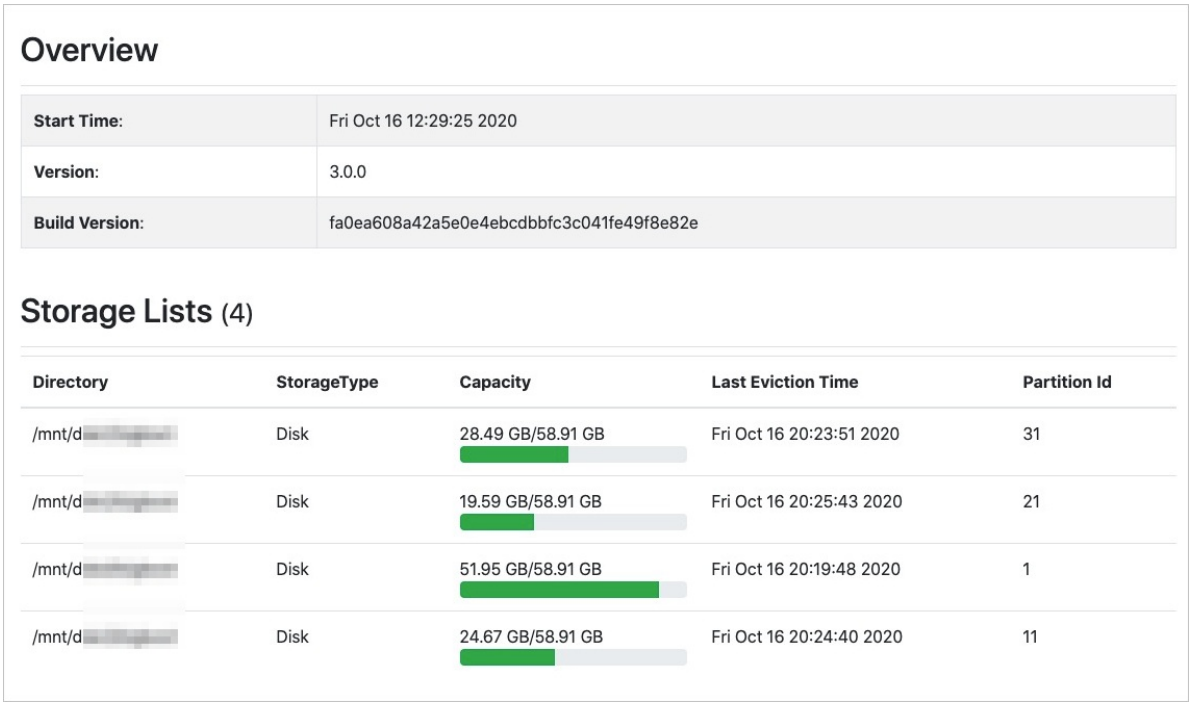
  This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.



StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.



Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

6.2.6. Manage JindoFS permissions

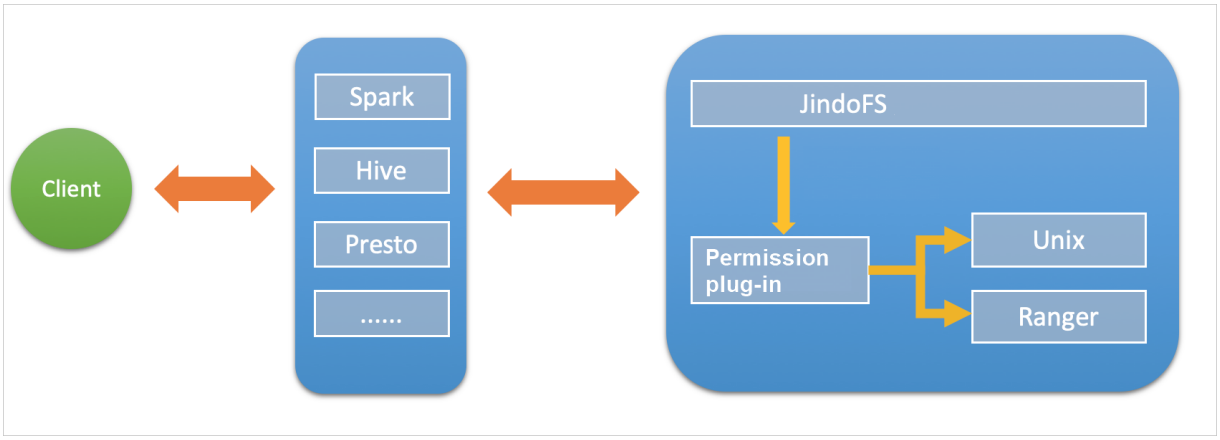
This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

Permission management based on the storage mode:

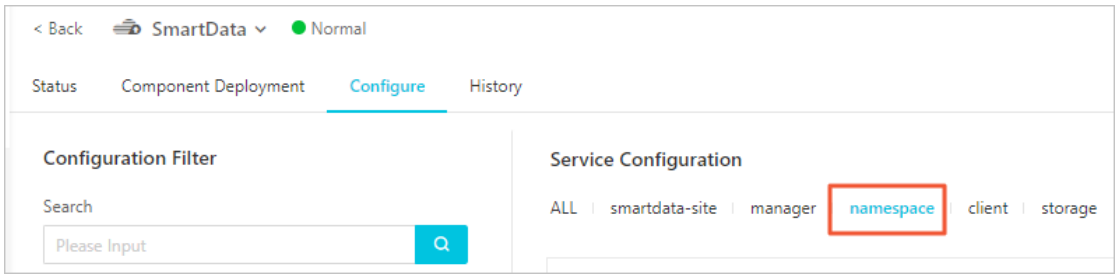
- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.
  - Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

6.2.7. Data management policies

JindoFS in block storage mode provides advanced data management policies to meet storage requirements in different scenarios. The advanced policies include storage policies and compression policies. This topic describes the policies in detail and provides examples to show how to use the policies.

Storage policies

JindoFS provides the five storage policies described in the following table to support flexible storage.

Policy	Description
AR	Data has only one backup. The backup is stored as an Archive object in OSS.
IA	Data has only one backup. The backup is stored as an Infrequent Access (IA) object in OSS.

Policy	Description
COLD	Data has only one backup. The backup is stored as a Standard object in OSS.
WARM	Data has one backup in OSS and one local backup. The local backup can effectively accelerate data read operations. By default, this storage policy is used.
HOT	Data has one backup in OSS and one local backup. The local backup is forcibly locked. This way, it is not deleted when the cache is automatically cleared. This policy can achieve a better acceleration effect than the WARM policy and is suitable for the hottest data.

For more information about the storage classes of OSS, see Overview.

A newly added file is stored based on the storage policy specified for its parent directory. Example:

- Run the following command to configure a storage policy:

```
jindo jfs -setStoragePolicy [-R] <StoragePolicy>(AR/IA/COLD/WARM/HOT) <path> ...
```

  where:

  - `[-R]`: specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.
  - `<path>`: the name of the directory for which the storage policy is configured.

- Run the following command to obtain the storage policy configured for a directory:

```
jindo jfs -getStoragePolicy <path>
```

Compression policies

JindoFS allows you to compress data blocks before you store them. This feature effectively reduces the storage space occupied by data blocks and improves data read and write efficiency. This feature is suitable for files that have a high compression ratio. The following table describes the supported compression policies.

Policy	Description
NONE	Data blocks are not compressed. By default, this compression policy is used.
ZSTD	The Zstandard compression algorithm is used to compress data blocks.

A newly added file is compressed based on the compression policy specified for its parent directory before it is stored. Example:

- Run the following command to configure a compression policy:

```
jindo jfs -setCompressionPolicy [-R] <CompressionPolicy>(NONE/ZSTD) <path> ...
```

where:

- `[-R]`: specifies that a recursive operation is performed to configure the same compression policy for all sub-directories of the directory.
- `<path>`: the name of the directory for which the compression policy is configured.

- Run the following command to obtain the compression policy configured for a directory:

```
jindo jfs -getCompressionPolicy <path> ...
```

6.2.8. Analyze metadata offline

In E-MapReduce (EMR) V3.30.0 or later, JindoFS in block storage mode allows you to dump the metadata of an entire namespace to Object Storage Service (OSS) and use Jindo SQL to analyze the metadata.

Context

In Hadoop Distributed File System (HDFS), the metadata is stored in a snapshot file named fsimage. The snapshot file contains the following information about metadata: namespace, file that stores the metadata, blocks, and file system quota. In HDFS, you can run a command to download the fsimage file in the XML format to your on-premises machine and view this file to analyze metadata offline. JindoFS does not require you to download metadata to your on-premises machine.

Upload HDFS metadata to OSS

Run the following command to upload metadata of a specific namespace to OSS:

```
jindo jfs -dumpMetadata <nsName>
```

`<nsName>` indicates the name of the namespace in block storage mode.

For example, upload the metadata of the test-block namespace to OSS and analyze the metadata offline.

```
jindo jfs -dumpMetadata test-block
```

If the following information appears, data is uploaded to OSS and stored as a file in the JSON format.

```
Sucessfully upload namespace metadata to OSS.
```

Metadata upload path

The metadata upload path is the metadataDump sub-directory under *sysinfo* in JindoFS.

For example, if `namespace.sysinfo.oss.uri` is set to `oss://abc/`, metadata is uploaded to the `oss://abc/metadataDump` directory.

Parameter	Description
namespace.sysinfo.oss.uri	The path of the OSS bucket.
namespace.sysinfo.oss.endpoint	The endpoint. An endpoint specified in a different region can also be used.
namespace.sysinfo.oss.access.key	The AccessKey ID of your Alibaba Cloud account.
namespace.sysinfo.oss.access.secret	The AccessKey secret of your Alibaba Cloud account.

Batch information: Metadata from HDFS changes based on the usage. Each time you want to analyze metadata, you must run commands to view the snapshot file of the metadata. Each time you run a Jindo command to upload metadata, a batch number that is generated based on the upload time is used as the root directory to store the metadata. This ensures that the data you uploaded each time is not overwritten. You can delete historical data based on your requirements.

- 1 indicates the configuration path of system information in OSS.
- 2 indicates a namespace.
- 3 indicates a batch number.

Metadata schema

HDFS metadata that is uploaded to OSS is stored as a file in the JSON format. Schema information:

```
{
  "type":"string",          /*Inode type, FILE DIRECTORY*/
  "id": "string",            /*INode id*/
  "parentId" :"string",         /*Parent node ID*/
  "name":"string",         /*Inode name*/
  "size": "int",         /*Inode size, BIGINT*/
  "permission":"int",         /*Permission, INT*/
  "owner":"string",          /*Owner name*/
  "ownerGroup":"string",     /*Owner group name*/
  "mtime":"int",              /*Inode modification time, BIGINT*/
  "atime":"int",              /*The time when the inode was last accessed, BIGINT*/
  "attributes":"string",       /*File-related attributes*/
  "state":"string",            /*Inode status*/
  "storagePolicy":"string",    /*Storage policy*/
  "etag":"string"           /*etag*/
}
```

Use Jindo SQL to analyze metadata

1. Run the command shown in the following figure to start Jindo SQL.

```
[root@emr-header-1 ~]# jindo sql
Spark master: yarn, Application Id: application_1603081647416_0050
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.33 seconds, Fetched 3 row(s)
```

2. Query the tables whose data can be analyzed by using Jindo SQL.
   - Run the `show tables` command to view the tables whose data can be analyzed. Jindo SQL has two built-in modules audit_log and fs_image for audit and metadata analysis.
   - Run the `show partitions fs_image` command to view the partition information of the fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

     The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[illegible]/datetime=2020_10_20_10_47_14
namespace=k[illegible]/datetime=2020_10_20_10_50_36
namespace=k[illegible]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

3. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[illegible]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287249    1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       15464684820176650002    1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The namespace and datetime columns are added and indicate the namespace name and the timestamp when metadata is uploaded.

Example: Obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

Use Hive to analyze metadata

1. Create a table schema in Hive.

   You can use the following command to create a table schema of the metadata that is used for queries in Hive:

```
CREATE EXTERNAL TABLE `table_name`
(`type` string,
 `id` string,
 `parentId` string,
 `name` string,
 `size` bigint,
 `permission` int,
 `owner` string,
 `ownerGroup` string,
 `mtime` bigint,
 `atime` bigint,
 `attr` string,
 `state` string,
 `storagePolicy` string,
 `etag` string)
 ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
 STORED AS TEXTFILE
 LOCATION 'The OSS path to which file data is uploaded';
```

2. Use Hive to analyze data offline.

   After you create a Hive table, you can use Hive SQL to analyze metadata.

```
select * from table_name limit 200;
```

   The following figure shows an example.

```
hive> select * from inode_metadata_test0 limit 100;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_202009[redacted]
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1[redacted], Tracking URL = http://emr-heade[redacted]:20888/proxy/applic[redacted]
Kill Command = /usr/lib/hadoop-current/bin/hadoop job  -kill job_1599[redacted]
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-09-08 14:57:26,112 Stage-1 map = 0%,  reduce = 0%
2020-09-08 14:57:31,263 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.22 sec
MapReduce Total cumulative CPU time: 1 seconds 220 msec
Ended Job = job_[redacted]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.22 sec   HDFS Read: 6867 HDFS Write: 1524 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 220 msec
OK
Directory       1127433438684721971 4    1127433438684721971 3    /uttest/oss     0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Directory       1127433438684721971 9    1127433438684721971 3    /uttest/oss2    0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
Directory       1127433438684721971 6    1127433438684721971 4    /uttest/oss/dir 0       511     caojie  staff   1599545017636   1599545017636           Finalized       WARM
File    1127433438684721971 5    1127433438684721971 4    /uttest/oss/file1       0       420     caojie  staff   1599545017632   1599545017632           Finalized       WARM
File    1127433438684721971 7    1127433438684721971 6    /uttest/oss/dir/file2   0       420     caojie  staff   1599545017642   1599545017642           Finalized       WARM
File    1127433438684721971 8    1127433438684721971 6    /uttest/oss/dir/file3   0       420     caojie  staff   1599545017651   1599545017651           Finalized       WARM
Directory       1127433438684721972 0    1127433438684721971 9    /uttest/oss2/dir        0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
File    1127433438684721972 1    1127433438684721972 0    /uttest/oss2/dir/file2  0       420     caojie  staff   1599545017658   1599545017658           Finalized       WARM
File    1127433438684721972 2    1127433438684721972 0    /uttest/oss2/dir/file3  0       420     caojie  staff   1599545017666   1599545017666           Finalized       WARM
Directory       1127433438684721971 3    1767282355743385190 5    /uttest 0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Time taken: 10.734 seconds, Fetched: 10 row(s)
hive>
```

6.2.9. Use a JindoFS credential provider

The data of JindoFS is stored in Object Storage Service (OSS). If you want to access the data of JindoFS, you must provide an AccessKey pair that can be used to access OSS. SmartData V3.4.0 and later support JindoFS credential providers. You can use a JindoFS credential provider to save an encrypted AccessKey pair into a file. This prevents the AccessKey pair from being leaked.

Configure a JindoFS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **smartdata-site** tab.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **smartdata-site** tab.
3. Add configuration information.
   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

Parameter	Description
fs.jfs.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

   iii. Click **OK**.
4. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Credential provider types

- TemporaryAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a validity period and a security token with a validity period are used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this credential provider, you must configure the parameters described in the following table in environment variables.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**.
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- JindoCommonCredentialsProvider

This is a common credential provider. After you configure the parameters described in the following table, you can use this credential provider in both JindoOSS and JindoFS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.JindoCommonCredentialsProvider**.
jindo.common.accessKeyId	The AccessKey ID used to access OSS.
jindo.common.accessKeySecret	The AccessKey secret used to access OSS.
jindo.common.securityToken	The temporary security token used to access OSS.

- EcsStsCredentialsProvider

  This credential provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EcsStsCredentialsProvider**.

6.2.10. Use JindoFS in block storage mode to store encrypted files

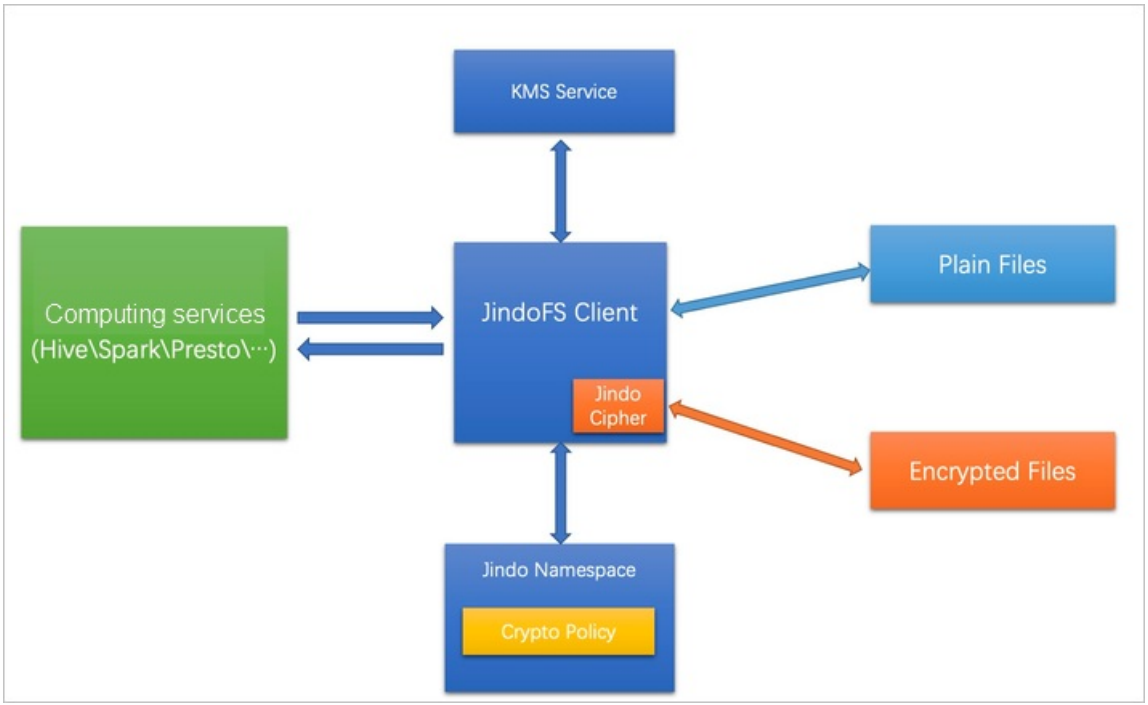
JindoFS in block storage mode supports file encryption. The encryption mechanism and usage method are similar to those of encryption zones for Apache HDFS. Keys are managed by Key Management Service (KMS). You can configure encryption policies for a directory that contains sensitive data. This way, you can transparently encrypt the data that you want to write to the directory and transparently decrypt the data that you want to read from the directory. In this process, you do not need to change your code.

Prerequisites

- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.
- KMS is activated. For more information, see Activate KMS.

Context

The following figure shows the architecture of JindoFS in block storage mode with file encryption supported.



Configure KMS-related parameters for JindoFS

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **namespace** tab.
3. Add configuration information.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description
crypto.provider.type	The type of the provider. Set this parameter to **ALIYUN**.
crypto.provider.endpoint	The public endpoint of KMS. For more information, see Make API requests.
crypto.provider.kms.accessKeyId	The AccessKey ID used to access KMS.
crypto.provider.kms.accessKeySecret	The AccessKey secret used to access KMS.

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Use JindoFS KeyProvider

JindoFS KeyProvider is used to connect JindoFS to KMS. The keys for JindoFS are stored in KMS. You can use JindoFS KeyProvider to create, query, and rotate keys based on KMS.

- Create a key: You can run the following command to create a key:

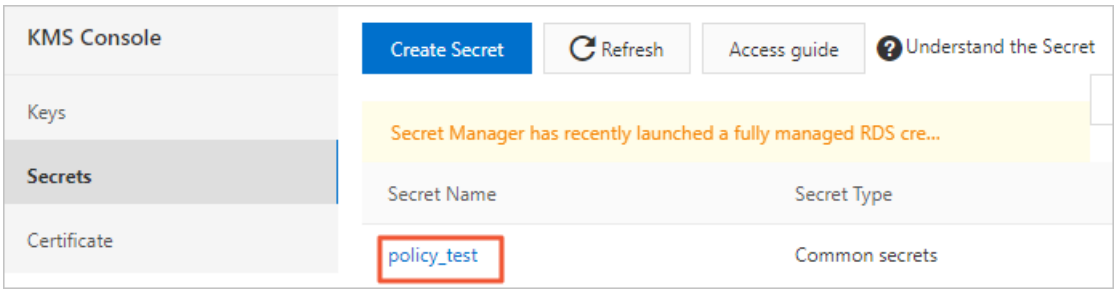
```
jindo key -create -keyIdName <keyIdName>
```

> **Note** In this topic, <keyIdName> in commands indicates the name of the key that you created.

For example, run the following command to create a key named *policy_test*:

```
jindo key -create -keyIdName policy_test
```

After the *policy_test* key is created, it is displayed on the Secrets page of the KMS console.



- Query keys: You can run the following command to query the names of all existing keys for JindoFS:

```
jindo key -list
```

The following information is returned:

```
Listing Keys:
        policy_test
        policy_test2
```

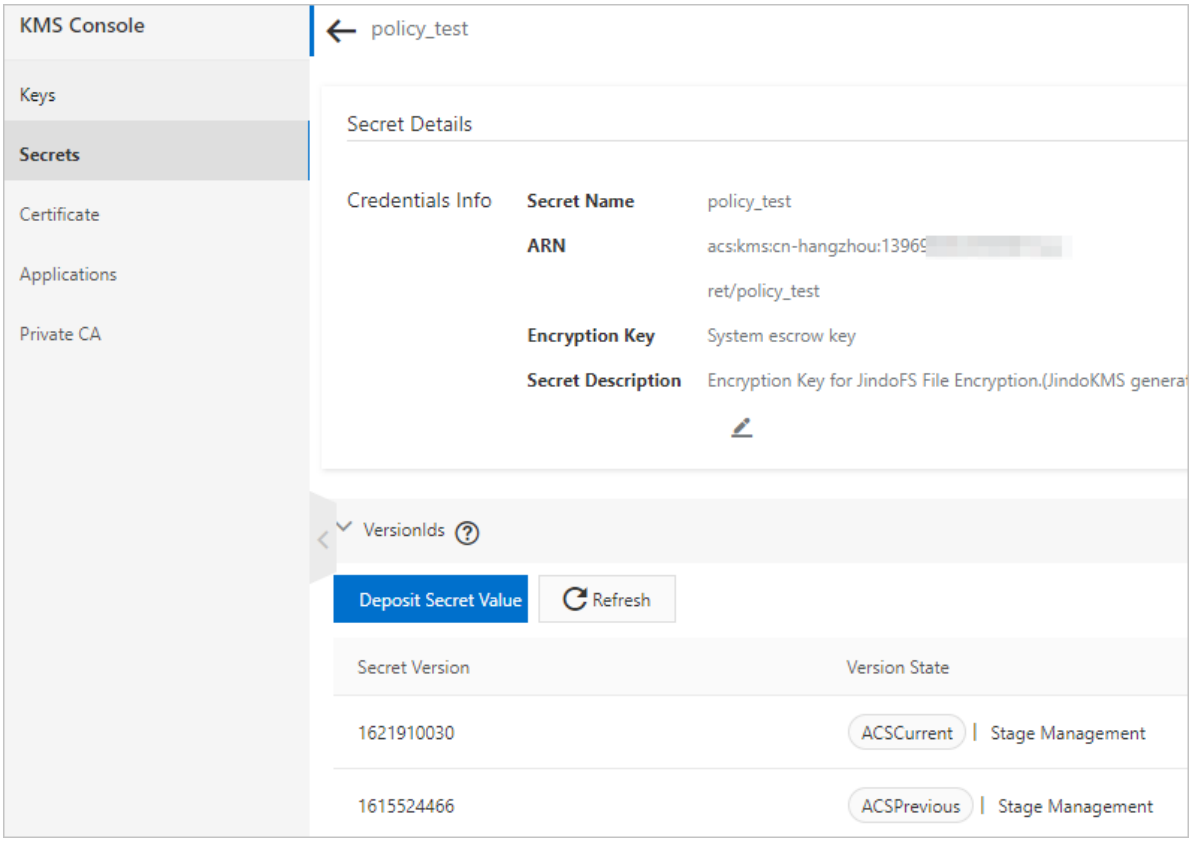
- Rotate a key: You can run the following command to periodically rotate a key based on its ID. After the key is rotated, the key version changes. If a file is encrypted after the rotation, the latest key version is used. If a file is decrypted after the rotation, the key version that corresponds to the file is used.

```
jindo key -roll -keyIdName <keyIdName>
```

For example, run the following command to rotate the *policy_test* key:

```
jindo key -roll -keyIdName policy_test
```

After the *policy_test* key is rotated, the version of the key is updated. The original version is in the **ACSPrevious** state and the new version is in the **ACSCurrent** state.



Manage encryption policies for JindoFS

You can run the following commands to configure and query encryption policies.

- Configure an encryption policy:

```
jindo jfs -setCryptoPolicy -keyIdName <keyIdName> <path>
```

> **Note** `<path>` in the preceding example indicates the path of a file on JindoFS, for example, *jfs://test/*.

- Query an encryption policy:

```
jindo jfs -getCryptoPolicy <path>
```

Example:

1. Query the encryption policy of the *jfs://test/* path:

```
jindo jfs -getCryptoPolicy jfs://test/
```

`{NONE}` is returned.

2. Configure an encryption policy for the *jfs://test/* path:

```
jindo jfs -setCryptoPolicy -keyIdName policy_test jfs://test/
```

3. Go to the *bigboot* directory and query the encryption policy of the *jfs://test/* path again:

```
jindo jfs -getCryptoPolicy jfs://test/
```

The following information is returned:

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/b2jindosdk/3.4.0-hadoop3.1/pack
age/b2jindosdk-3.4.0-hadoop3.1/lib/jindo-distcp-3.4.0.jar!/org/slf4j/impl/StaticLoggerB
inder.class]
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/hadoop/3.2.1-1.0.1/package/hado
op-3.2.1-1.0.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticL
oggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
21/03/12 13:52:34 WARN: Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable
21/03/12 13:52:35 INFO: Jboot log name is /var/log/bigboot/jboot-20210312-135234-12953.
LOG
21/03/12 13:52:35 INFO: Write buffer size 1048576, logic block size 134217728
21/03/12 13:52:35 INFO: cmd=getFileStatus, src=jfs://test/, dst=null, size=0, parameter
=null, time-in-ms=7, version=3.4.0
21/03/12 13:52:35 INFO: cmd=getCryptoPolicy, src=jfs://test/, dst=null, size=0, paramet
er=, time-in-ms=2, version=3.4.0
The crypto policy of path: jfs://test/ is {cipherSuite: AES_CTR_NOPADDING_256, keyIdNam
e: policy_test2, keyIdVersion: null, edek: , iv: }
21/03/12 13:52:35 INFO: Read total statistics: oss read average <none>, cache read aver
age <none>, read oss percent <none>
```

After you configure the encryption policy, you can read data from or write data to the files under this path.

- Copy a local file to HDFS:

```
hadoop fs -put test.log jfs://test/
```

- Display the content of a file:

```
hadoop fs -cat jfs://test/test.log
```

6.3. JindoFS in cache mode

6.3.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in Object Storage Service (OSS) based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

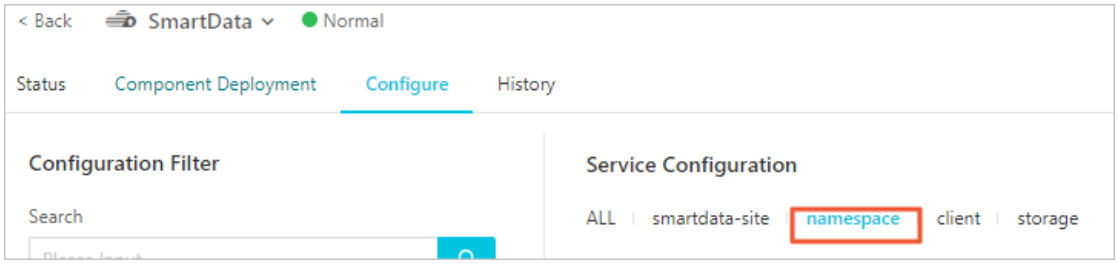
  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. Click **OK**.
5. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
6. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **client** tab.
2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   You do not need to restart the SmartData service to apply the new configurations.
3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

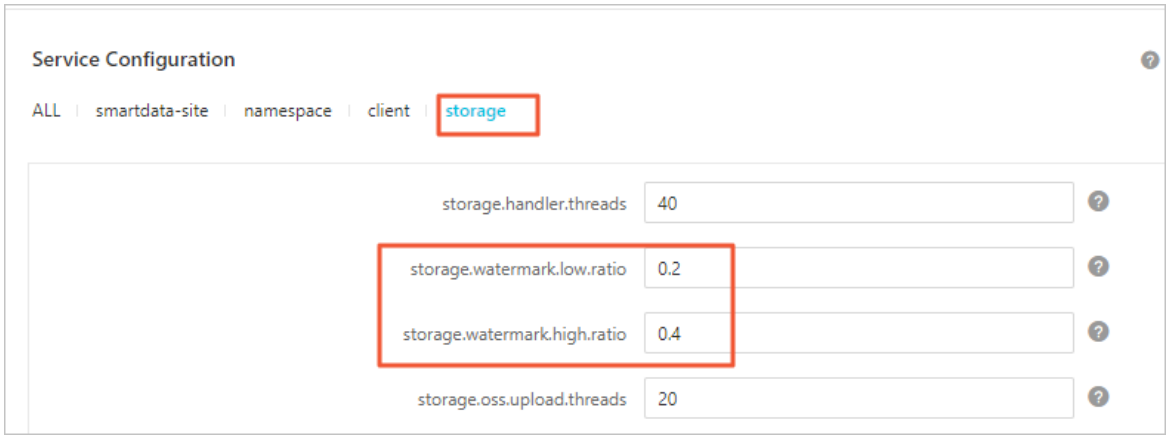
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you specify. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to specify an AccessKey pair. In other cases, you must specify an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you choose for accessing files in OSS:

- OSS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.
  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss.accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.endpoint	The endpoint of the OSS bucket that serves as the storage backend.

> **Note** You can set these parameters to the values of the parameters in an EMR version earlier than EMR V3.30.0.

- JFS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **namespace** tab.
  ii. Set **jfs.namespaces** to **test**.
  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize the cache performance. After you configure the parameters, you do not need to restart the SmartData service to apply the new configurations.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. You can use this parameter to prevent upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If an excessive amount of memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

6.3.2. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

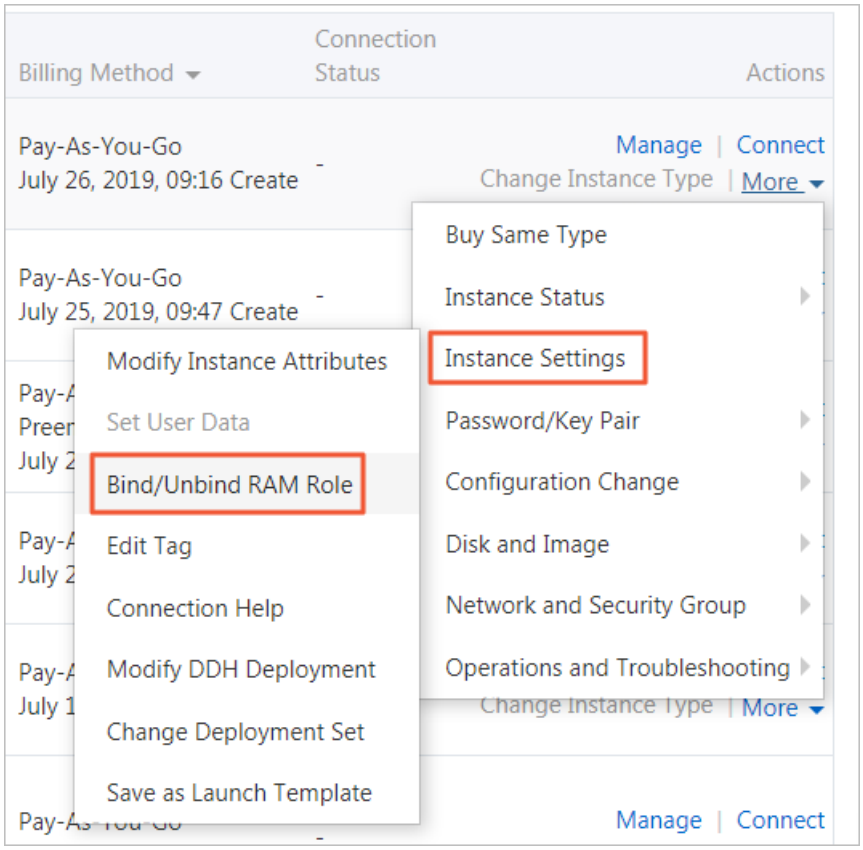
1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.
4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

   ```
   hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
   ```

2. Access OSS by using the FileSystem interface of Hadoop.

   Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

6.3.3. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false

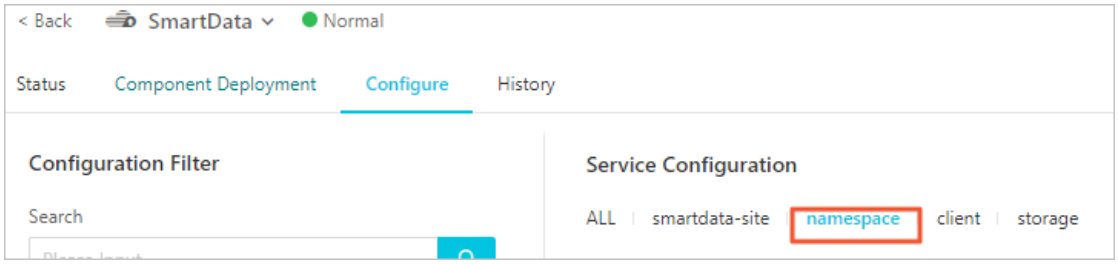
Parameter	Description
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-

right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

6.3.4. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

> **Notice**
>
> - The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
> - After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
> - Hive and Presto jobs do not use Hadoop Job Committer.
> - In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.

i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
ii. Click the **Configure** tab.
iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.
6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.

3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

   The default value is 8.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

6.3.5. Use JindoFS OSS credential providers

You can use a credential provider to save an encrypted AccessKey pair into a file. This prevents the leak of the AccessKey pair.

Context

You can use a Hadoop credential provider to save an encrypted AccessKey pair into a file. This prevents the issue that the AccessKey pair is transmitted in plaintext. You can select an appropriate JindoOSS credential provider based on your business requirements.

Configure a JindoOSS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **smartdata-site** tab.

   i. Click the **Configure** tab.

   ii. In the **Service Configuration** section, click the **smartdata-site** tab.

3. Add configuration information.

   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

   - Global configuration (for all buckets)

Parameter	Description
fs.jfs.cache.oss.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

- Bucket-level configuration

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types. **Note** XXX indicates the name of an Object Storage Service (OSS) bucket.

Credential provider types

You can select different credential providers based on your business requirements. The following providers are supported:

- Global configuration
  - TemporaryAliyunCredentialsProvider

    This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.cache.oss.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

- Bucket-level configuration

- TemporaryAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.
fs.jfs.cache.oss.bucket.XXX.securityToken	The temporary security token used to access the OSS bucket.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access an OSS bucket.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access the OSS bucket.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access the OSS bucket. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

Use a Hadoop credential provider to store AccessKey pair information

> **Note** For more information about Hadoop credential providers, see CredentialProvider API Guide.

Use a command that is provided by Hadoop to store AccessKey pair and security token information into a credential file. Syntax:

```
hadoop credential <subcommand> [options]
```

For example, in global configuration mode, store AccessKey pair and token information into a JCEKS file. You can protect the file by using file permissions or you can specify a password to encrypt the information you want to store. If you do not specify a password, the default string is used for encryption.

```
hadoop credential create fs.jfs.cache.oss.accessKeyId -value AAA -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.accessKeySecret -value BBB -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.securityToken -value  CCC -provider jceks://file/root/oss.jceks
```

After a credential file is generated, you must configure the parameter described in the following table to specify the provider type and location.

Parameter	Description
fs.jfs.cache.oss.security.credential.provider.path	The path used to store the credential file that stores AccessKey pair information. For example, you can set this parameter to *jceks://file/${user.home}/oss.jceks*, which indicates that the *oss.jceks* file is stored in the home directory.

6.3.6. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

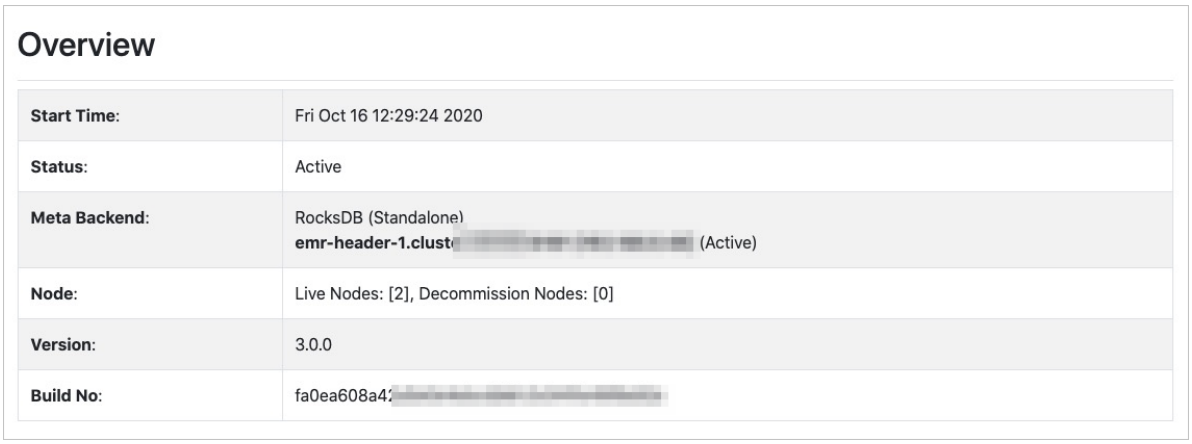
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

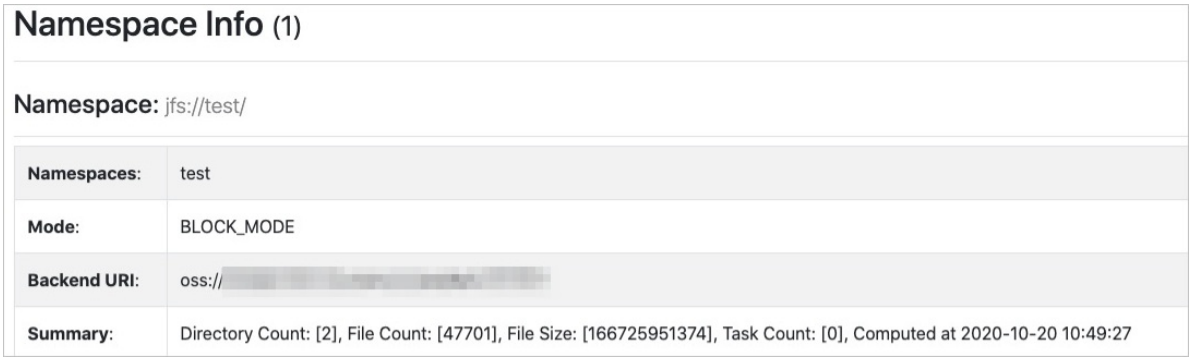


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.

StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.

Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

6.3.7. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

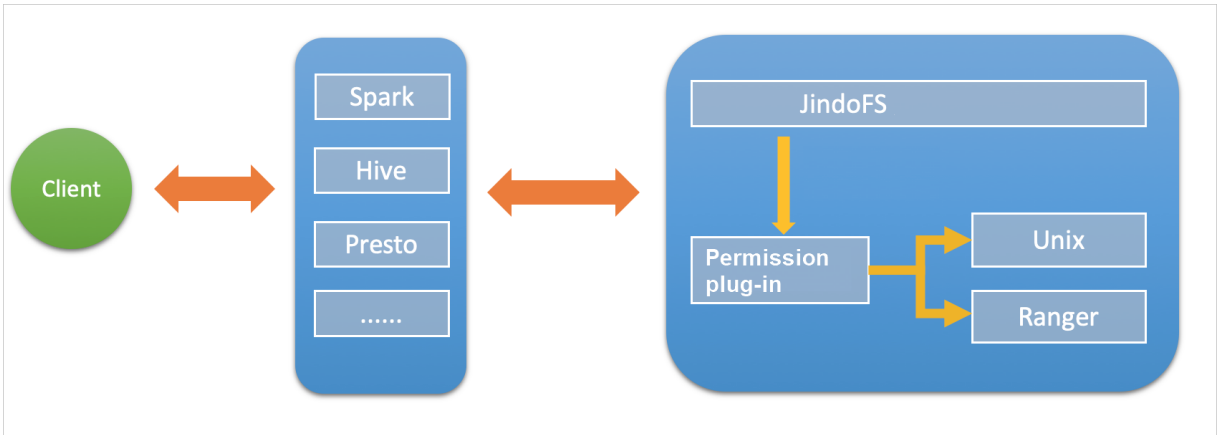
Permission management based on the storage mode:

- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.

- Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.

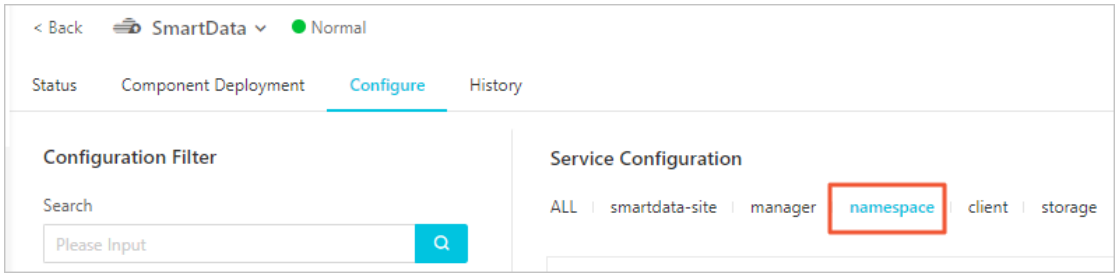
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

   After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

   If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

6.4. JindoTable

6.4.1. Enable query acceleration based on a native engine

JindoTable provides a native engine that can be used to accelerate the queries of files. This feature accelerates the speed of Spark, Hive, or Presto to query ORC or Parquet files. It is disabled by default.

Prerequisites

An E-MapReduce (EMR) cluster of V3.35.0 or later V3.X.X, or V4.9.0 or later V4.X.X is created. ORC or Parquet files are stored in JindoFS or Object Storage Service (OSS). For more information about how to create an EMR cluster, see Create a cluster.

Context

The following table lists the supported Spark, Hive, and Presto engines and the file formats supported by each engine.

Engine	ORC	Parquet
Spark2	Supported	Supported
Presto	Supported	Unsupported
Hive2	Unsupported	Supported

Limits

- Data of the binary type is not supported.
- Partitioned tables whose values of partition key columns are stored in files are not supported.
- You are not allowed to use spark.read.schema (userDefinedSchema) to define a schema because the schema may be inconsistent with the existing file schema.
- Data of the date type must be in the YYYY-MM-DD format and range from 1400-01-01 to 9999-12-31.
- Queries of columns that are case-sensitive in the same table cannot be accelerated. For example, queries of the ID and id columns of the same table cannot be accelerated.

Improve performance of Spark to read data

1. Enable query acceleration for ORC or Parquet files in JindoTable.

   > **Note**
   > - Query acceleration consumes off-heap memory. We recommend that you add `--conf spark.executor.memoryOverhead=4g` to a Spark task to apply for additional resources for query acceleration.
   > - You can use the native engine when Spark reads data from ORC or Parquet files.

   - Configure global parameters.

     For more information, see Configure the global parameters of Spark.

   - Configure job-level parameters.

     You can add Spark startup parameters when you run Spark Shell or Spark SQL jobs.

     ```
     spark.sql.extensions==io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sql.JindoTableExtension
     ```

     For more information about the configurations of jobs, see Configure a Spark Shell job or Configure a Spark SQL job.

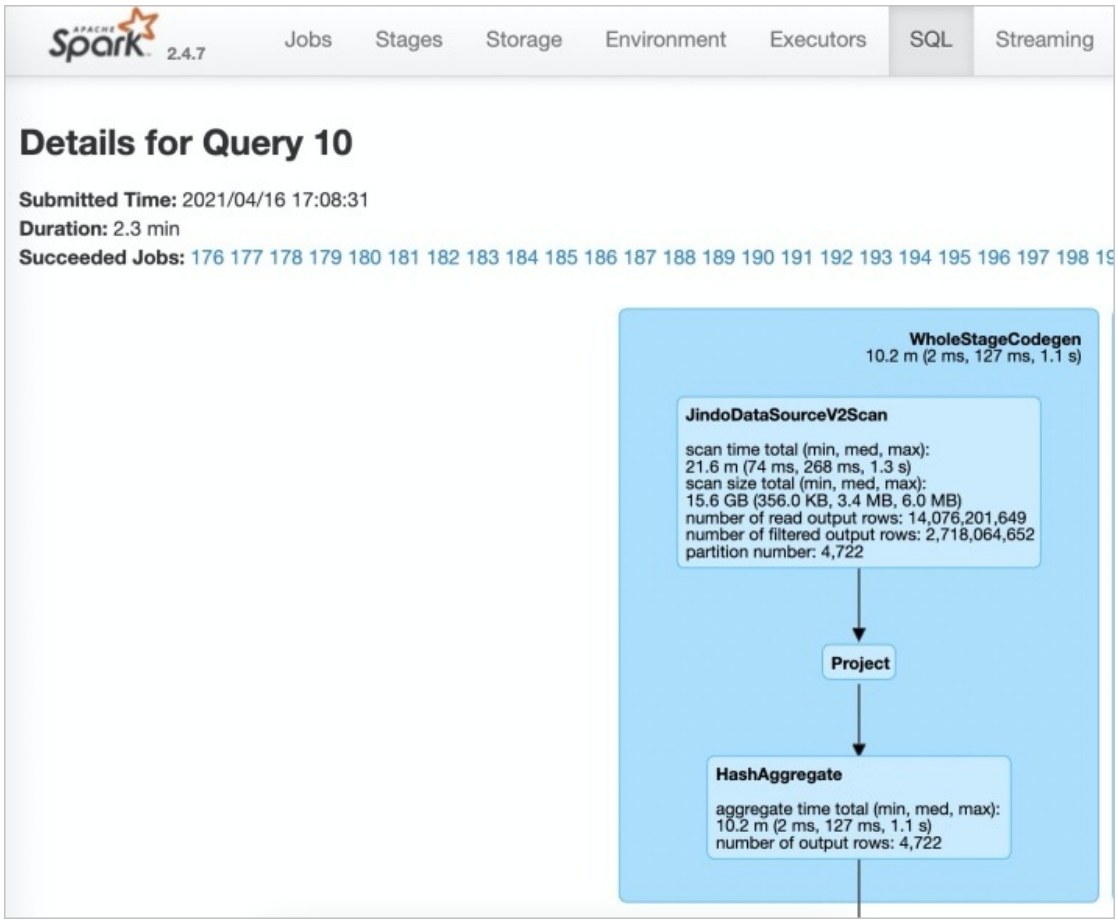
2. Check whether query acceleration is enabled.

   i. Access the web UI of Spark History Server.

      For more information, see Access the web UIs of open source components.

ii. On the SQL tab of Spark, view the execution task.

If JindoDataSourceV2Scan appears, query acceleration is enabled. Otherwise, check the configurations in Step 1.



Improve performance of Presto to read data

Presto has built-in `catalog: hive-acc`. You can use `catalog: hive-acc` to enable query acceleration.

Example:

```
presto --server https://emr-header-1.cluster-xxx:7778/ --catalog hive-acc --schema default
```

`emr-header-1.cluster-xxx` indicates the hostname of the emr-header-1 node.

> **Note** Complex data types, such as MAP, STRUCT, and ARRAY, are not supported when you use this feature in Presto.

Improve performance of Hive to read data

> **Notice** If you want to schedule jobs in a more stable environment, we recommend that you disable this feature in Hive.

EMR Hive V2.3.7 (EMR V3.35.0) has integrated a plug-in provided by JindoTable. The plug-in accelerates queries of Parquet files. You can set hive.jindotable.native.enabled in your Hive job to enable query acceleration. Alternatively, you can add the **hive.jindotable.native.enabled** custom parameter and set it to **true** on the *hive-site.xml* tab of the Hive configuration page to enable query acceleration, and restart Hive. The latter is suitable to Hive on MapReduce and Hive on Tez.

Example:

```
set hive.jindotable.native.enabled=true;
```

> **Note** Complex data types, such as MAP, STRUCT, and ARRAY, are not supported when you use this feature in Hive.

Configure the global parameter of Spark

1. Go to the Spark service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Spark**.
2. On the Spark service page, click the **Configure** tab.
3. Find the spark.sql.extensions parameter and change its value to io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sql.JindoTableExtension.
4. Save the configurations.
   i. Click **Save** in the upper-right corner of the Service Configuration section.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
5. Restart ThriftServer.
   i. Choose **Actions > Restart ThriftServer** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

6.4.2. Use JindoCube

JindoCube is supported in E-MapReduce (EMR) V3.24.0 and later. This topic describes how to install, deploy, and use JindoCube in EMR.

Prerequisites

A table or view is created.

Overview

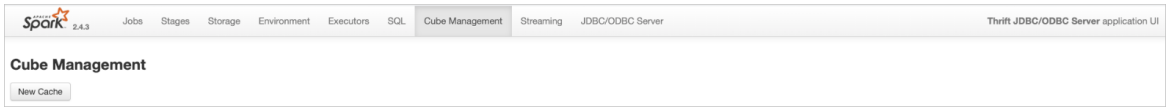
JindoCube is an advanced feature of EMR Spark. It implements pre-calculation to accelerate data processing. It improves query performance by dozens or even hundreds of times. You can persist data of a view to a data source supported by EMR Spark, such as HDFS or Object Storage Service (OSS). EMR Spark automatically discovers available persistent data and optimizes the execution plans for the data. This process is completely visible to users. JindoCube applies to business scenarios where a fixed query mode is used. JindoCube pre-calculates and pre-organizes data to accelerate data queries. For example, you can use JindoCube to implement multidimensional online analytical processing (MOLAP), generate data reports, create data dashboards, and synchronize data across clusters.

Install and deploy JindoCube

JindoCube is an advanced feature of EMR Spark. You can use JindoCube to accelerate all Dataset, DataFrame API, and SQL tasks that are submitted by using EMR Spark. No additional component deployment or maintenance is required.

1. Manage JindoCube on the Spark web UI.

   You can manage JindoCube on the Spark web UI. For example, you can create, delete, or update JindoCube caches. After you create a JindoCube cache, it automatically accelerates all Spark queries in the cluster. You can use the **spark.sql.cache.tab.display** parameter to specify whether to display the Cube Management tab on the Spark web UI. You can specify this parameter on the configuration page of the Spark service in the EMR console or in a Spark task submission command. The default value of this parameter is **false**.

   You can also use the **spark.sql.cache.useDatabase** parameter to create a database based on your business requirements and store the view for which you want to create a cache in this database. If partitioned tables are used, you can use the **spark.sql.cache.cacheByPartition** parameter to specify whether to cache data by partition.

Parameter	Description	Example
spark.sql.cache.tab.display	Specifies whether to display the Cube Management tab.	true
spark.sql.cache.useDatabase	The databases where cubes are stored.	db1,db2,dbn
spark.sql.cache.cacheByPartition	Specifies whether to store cubes by partition.	true

2. Optimize Spark queries.

   You can use the **spark.sql.cache.queryRewrite** parameter to specify whether to use a JindoCube cache to accelerate Spark queries. You can specify this parameter at the cluster, session, or SQL statement level. The default value is **true**.

Use JindoCube

1. Create a JindoCube cache.
   i. Log on to the Alibaba Cloud EMR console by using your Alibaba Cloud account.
   ii. Click the **Cluster Management** tab.

iii. Find the cluster where you want to create a JindoCube cache and click the cluster ID.

iv. In the left-side navigation pane, click **Connect Strings**.

v. On the **Public Connect Strings** page, click the link of YARN UI to go to the YARN web UI as a Knox user.

For more information about Knox, see Knox.

vi. Click **ApplicationMaster** in the row where **Application Type** is **SPARK** and **Name** is **Thrift JDBC/ODBC Server**.

vii. On the Spark web UI, click the **Cube Management** tab.

viii. Click **New Cache**.

Select a table or view, and click raw cache or cube cache in the action column. You can create two types of caches:

- Raw cache: persists the data of a table or view based on the cache configurations.

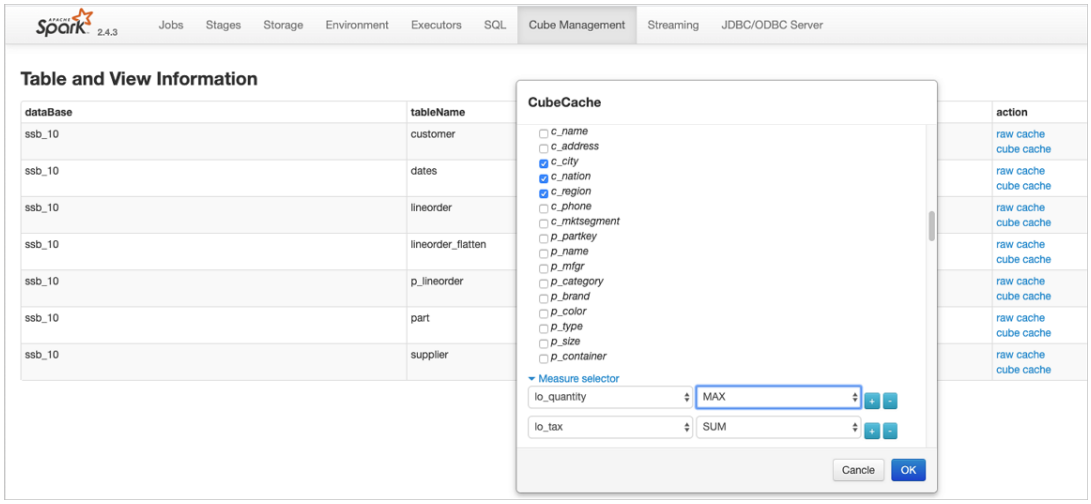
  When you create a raw cache, specify the parameters described in the following table.

Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Column Selector	The columns from which you want to cache data.	Yes
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No

- Cube cache: builds a cube based on the raw data of a table or view and persists the cube data.

When you create a cube cache, specify the parameters described in the following table.

Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Dimension Selector	The dimension fields that are used to build a cube.	Yes
Measure Selector	The measure fields and pre-calculation functions that are used to build a cube.	Yes
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No



JindoCube builds a cube based on specified dimensions and measures. The following SQL statement shows the configurations of the cube cache in the preceding figure:

```
SELECT c_city, c_nation, c_region, MAX(lo_quantity), SUM(lo_tax)
FROM lineorder_flatten
GROUP BY c_city, c_nation, c_region;
```

JindoCube calculates the finest-grained combination of dimensions. Powered by the onsite computing capability of Spark, JindoCube implements re-aggregation to optimize the queries where a coarse-grained combination of dimensions is used. You must use the pre-calculation functions supported by JindoCube to define a cube cache. The following table lists the supported pre-calculation functions and the aggregate functions that correspond to them.

Aggregate function	Pre-calculation function
COUNT	COUNT
SUM	SUM
MAX	MAX
MIN	MIN
AVG	COUNT and SUM
COUNT (DISTINCT)	PRE_COUNT_DISTINCT
APPROX_COUNT_DISTINCT	PRE_APPROX_COUNT_DISTINCT

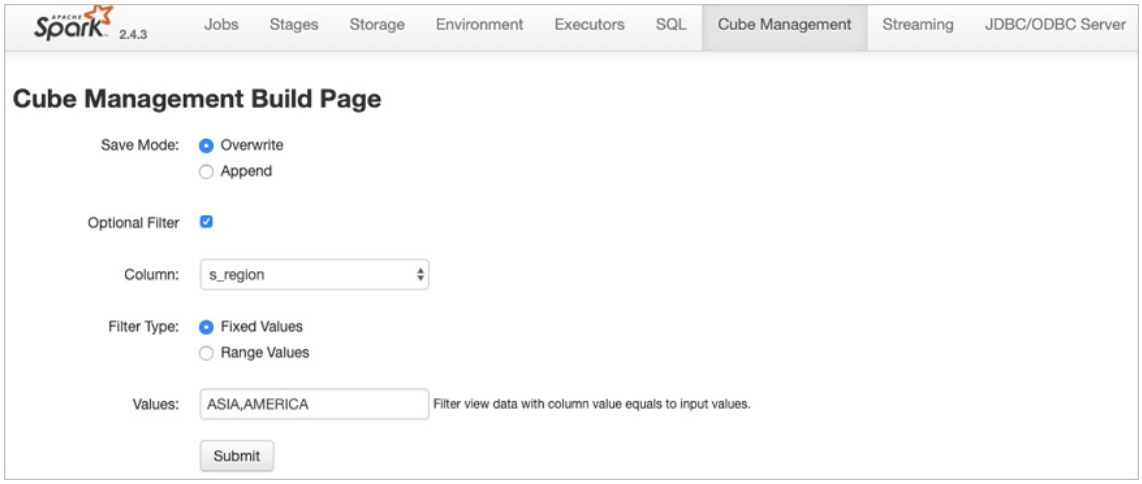
All created caches are displayed on the **Cube Management** tab. You can click **Detail** for a cache to view its details, including its basic information, partition information, building information, and building history.

2. Build a JindoCube cache.

   During the creation of a cache, metadata is prepared, but no data is persisted. Therefore, after you create a cache, you must continue to build the cache. This way, the related data can be persisted to permanent storage, such as HDFS or OSS. In addition, the source table of a cache may be appended with new data or updated. In this case, you must update the data in the cache to ensure data consistency. You can use one of the following methods to build a cache:

   - Manually build a cache

     Click Build Cache to manually build a cache. On the page that appears, configure the parameters, as shown in the following figure.



The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.

Optional Filter	You can select this parameter to add filter conditions. This way, JindoCube filters data based on the filter conditions before it persists the data.
	■ Column: the field based on which data is filtered.
	■ Filter Type: specifies whether to filter data by fixed values or by value range.
	  ■ Fixed Values: If this option is selected, specify one or more values. Separate values with commas (,).
	  ■ Range Values: If this option is selected, specify a minimum value and a maximum value. You can leave the maximum value empty. In this case, the value range is the minimum value and all values greater than it.

The following SQL statement shows the configurations of the raw cache for the lineorder_flatten view in the preceding figure:

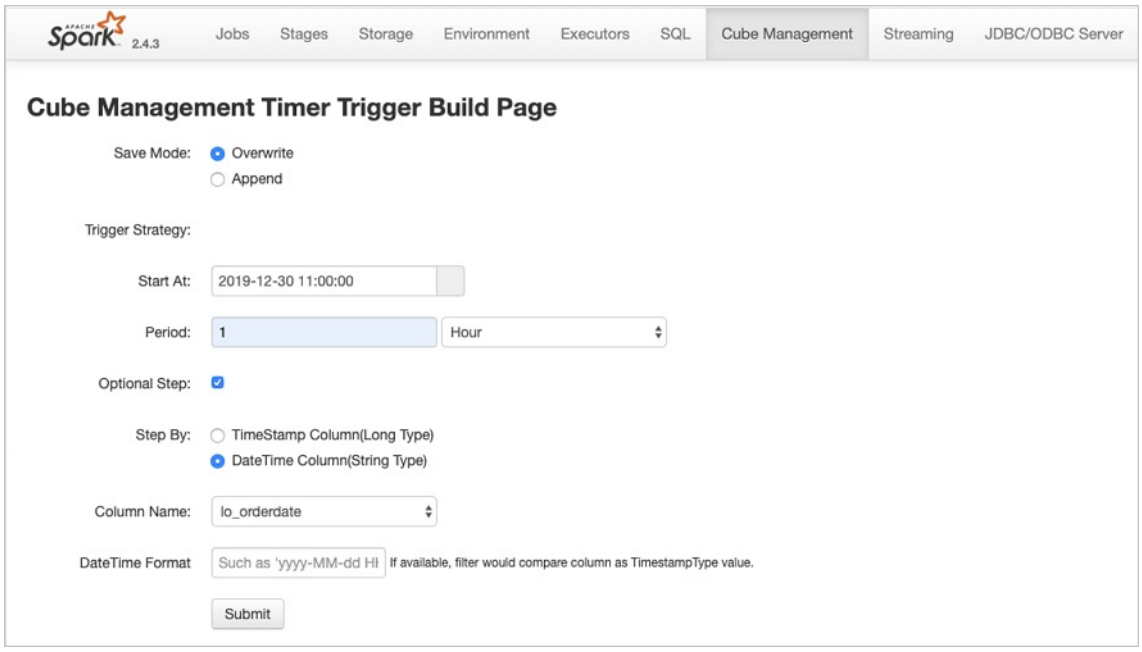
```
SELECT * FROM lineorder_flatten
WHERE s_region == 'ASIA' OR s_region == 'AMERICA';
```

Click **Submit** to submit the cache building task to a Spark cluster. You can view the task status in the Build Information section of the cache details page. After the task is complete, you can view the cache information in the Build History section.

> **Note** Spark writes the cached data to a specified directory in the same way common Spark tasks write data to a table or directory. If multiple users concurrently build a cache whose data format is Parquet, JSON, or ORC, the cached data may be inaccurate and invalid. If concurrent cache building operations or updates are inevitable, we recommend that you use a data format that supports concurrent writes, for example, Delta.

- Configure periodic cache building

  Click Trigger Period Build to configure a periodic cache update policy to ensure data consistency between the cache and source table. On the page that appears, configure the parameters, as shown in the following figure.



The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.

Parameter	Description
Trigger Strategy	Specifies the start time of the first building task, and the building interval. ■ Start At: Select or enter the time you want to trigger the first building task. Format: yyyy-MM-dd hh:mm:ss. ■ Period: Specify the interval at which cache building tasks are triggered.
Optional Step	Specifies the filter conditions for cache building. JindoCube updates the cache in incremental mode based on the filter conditions and interval. If you do not select Optional Step, JindoCube updates the entire cache each time. ■ Step By: Specify the type of the field that you want to update in incremental mode. TimeStamp Column(Long Type) indicates a timestamp field of the LONG type. DateTime Column(String Type) indicates a datetime field of the STRING type. ■ Column Name: Specify the name of the field that you want to update in incremental mode.

On the cache details page, you can view the configured cache update policy. You can click Cancel Period Build to cancel periodic updates. You can view the information of a complete building task in the Build History section.

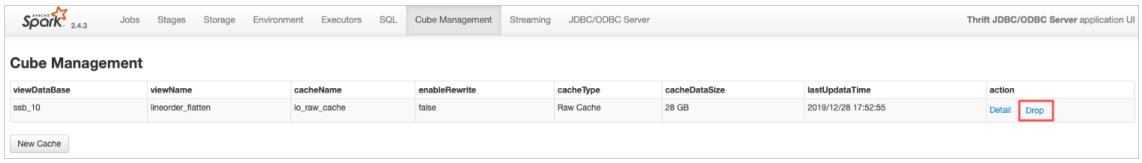
> **Note**
>
> - Periodic building tasks run on a Spark cluster. The related configurations are all saved in SparkContext, and the tasks are periodically triggered by Spark Driver. If you release the Spark cluster, periodic building is disabled.
> - The Build History section displays the following information of each complete task in the current Spark cluster: start time, end time, save mode, building conditions, and final status. After you release the Spark cluster, the information in the Build History section is cleared.

3. Manage a JindoCube cache.

   After you create and build a JindoCube cache, you can manage it on the Cube Management page.

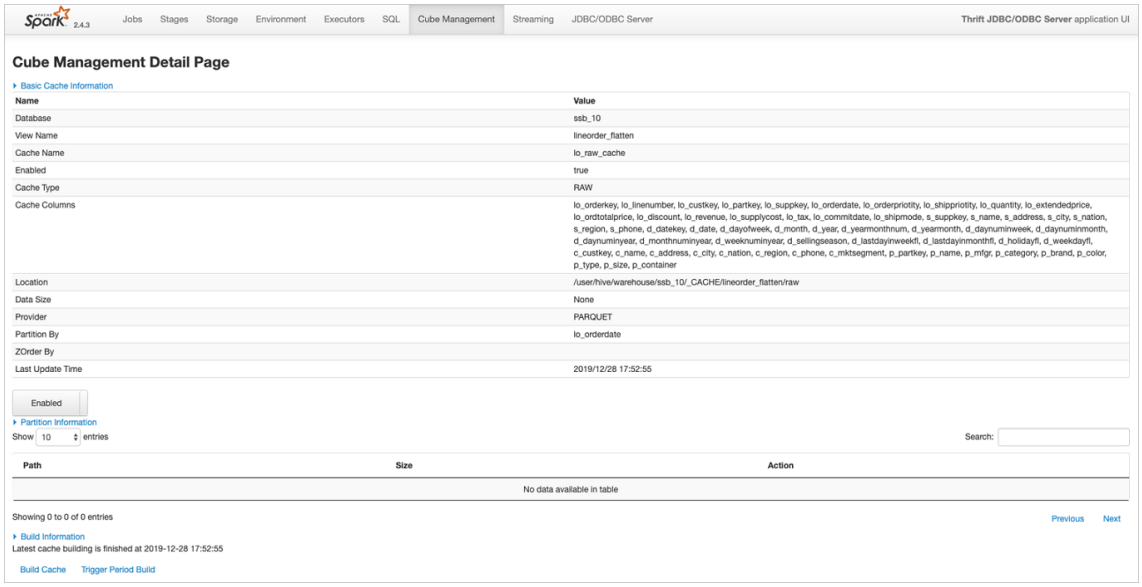
   - Delete a cache

     On the Cube Management page, find the cache you want to delete, and click **Drop** in the action column. All the metadata and data of the cache are deleted.



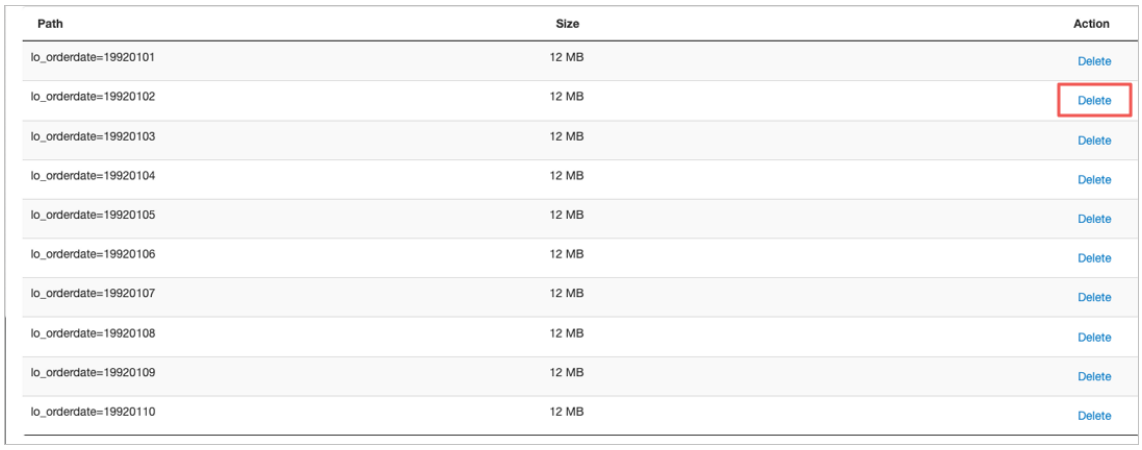
- Enable and disable optimization of Spark queries

  JindoCube allows you to specify whether to optimize Spark queries at the cache level. On the details page of a cache, click **Enabled** or **Disabled** in the lower part of the Basic Cache Information section to enable or disable the optimization.



- Delete partition data from a cache

  If data in a cache is stored by partition, you can delete the unnecessary partition data from the cache to release more space. On the details page of the cache, find the partition whose data you want to delete, and click **Delete** in the Action column.



Path	Size	Action
lo_orderdate=19920101	12 MB	Delete
lo_orderdate=19920102	12 MB	Delete
lo_orderdate=19920103	12 MB	Delete
lo_orderdate=19920104	12 MB	Delete
lo_orderdate=19920105	12 MB	Delete
lo_orderdate=19920106	12 MB	Delete
lo_orderdate=19920107	12 MB	Delete
lo_orderdate=19920108	12 MB	Delete
lo_orderdate=19920109	12 MB	Delete
lo_orderdate=19920110	12 MB	Delete

> **Note** Before you delete the data of a partition, make sure that the data is no longer needed. If you delete data that is required in subsequent queries, the queries return invalid results.

4. Optimize Spark queries.

JindoCube optimizes view-based queries. After you have created a raw cache or cube cache for a view, EMR Spark overwrites the execution plan of queries on this view. The new execution plan still uses the original logical semantics but can directly access the data in the cache. This accelerates Spark queries. Use the lineorder_flatten view as an example. It is a wide table view generated by associating the lineorder table with other dimension tables. The following figure shows the information about this view.

```
0: jdbc:hive2://localhost:10001/ssb_10> desc extended lineorder_flatten;
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+-------------------------------+----------------------------------------------------+----------+
|           col_name            |                     data_type                      | comment  |
+-------------------------------+----------------------------------------------------+----------+
| lo_orderkey                   | bigint                                             | NULL     |
| lo_linenumber                 | bigint                                             | NULL     |
| lo_custkey                    | int                                                | NULL     |
| lo_partkey                    | int                                                | NULL     |
| lo_suppkey                    | int                                                | NULL     |
| lo_orderdate                  | int                                                | NULL     |
| lo_orderpriotity              | string                                             | NULL     |
| lo_shippriotity               | int                                                | NULL     |
| lo_quantity                   | bigint                                             | NULL     |
| lo_extendedprice              | bigint                                             | NULL     |
| lo_ordtotalprice              | bigint                                             | NULL     |
| lo_discount                   | bigint                                             | NULL     |
| lo_revenue                    | bigint                                             | NULL     |
| lo_supplycost                 | bigint                                             | NULL     |
| lo_tax                        | bigint                                             | NULL     |
| lo_commitdate                 | int                                                | NULL     |
| lo_shipmode                   | string                                             | NULL     |
| s_suppkey                     | int                                                | NULL     |
| s_name                        | string                                             | NULL     |
| s_address                     | string                                             | NULL     |
| s_city                        | string                                             | NULL     |
| s_nation                      | string                                             | NULL     |
| s_region                      | string                                             | NULL     |
| s_phone                       | string                                             | NULL     |
| d_datekey                     | int                                                | NULL     |
| d_date                        | string                                             | NULL     |
| d_dayofweek                   | string                                             | NULL     |
| d_month                       | string                                             | NULL     |
| d_year                        | int                                                | NULL     |
| d_yearmonthnum                | int                                                | NULL     |
| d_yearmonth                   | string                                             | NULL     |
| d_daynuminweek                | int                                                | NULL     |
| d_daynuminmonth               | int                                                | NULL     |
| d_daynuminyear                | int                                                | NULL     |
| d_monthnuminyear              | int                                                | NULL     |
| d_weeknuminyear               | int                                                | NULL     |
| d_sellingseason               | string                                             | NULL     |
| d_lastdayinweekfl             | int                                                | NULL     |
| d_lastdayinmonthfl            | int                                                | NULL     |
| d_holidayfl                   | int                                                | NULL     |
| d_weekdayfl                   | int                                                | NULL     |
| c_custkey                     | int                                                | NULL     |
| c_name                        | string                                             | NULL     |
| c_address                     | string                                             | NULL     |
| c_city                        | string                                             | NULL     |
| c_nation                      | string                                             | NULL     |
| c_region                      | string                                             | NULL     |
| c_phone                       | string                                             | NULL     |
| c_mktsegment                  | string                                             | NULL     |
| p_partkey                     | int                                                | NULL     |
| p_name                        | string                                             | NULL     |
| p_mfgr                        | string                                             | NULL     |
| p_category                    | string                                             | NULL     |
| p_brand                       | string                                             | NULL     |
| p_color                       | string                                             | NULL     |
| p_type                        | string                                             | NULL     |
| p_size                        | int                                                | NULL     |
| p_container                   | string                                             | NULL     |
|                               |                                                    |          |
| # Detailed Table Information  |                                                    |          |
| Database                      | ssb_10                                             |          |
| Table                         | lineorder_flatten                                  |          |
| Owner                         | hadoop                                             |          |
| Created Time                  | Sat Dec 28 17:30:02 CST 2019                       |          |
| Last Access                   | Thu Jan 01 08:00:00 CST 1970                       |          |
| Created By                    | Spark 2.2 or prior                                 |          |
| Type                          | VIEW                                               |          |
| View Text                     | SELECT `lineorder`.`lo_orderkey`, `lineorder`.`lo_linenumber`, `lineorder`.`lo_custkey`, `lineorder`.`lo_partkey`, `
lineorder`.`lo_suppkey`, `lineorder`.`lo_orderdate`, `lineorder`.`lo_orderpriotity`, `lineorder`.`lo_shippriotity`, `lineorder`.`lo_quantity`, `lineor
der`.`lo_extendedprice`, `lineorder`.`lo_ordtotalprice`, `lineorder`.`lo_discount`, `lineorder`.`lo_revenue`, `lineorder`.`lo_supplycost`, `lineorder`
.`lo_tax`, `lineorder`.`lo_commitdate`, `lineorder`.`lo_shipmode`, `supplier`.`s_suppkey`, `supplier`.`s_name`, `supplier`.`s_address`, `supplier`.`s_
city`, `supplier`.`s_nation`, `supplier`.`s_region`, `supplier`.`s_phone`, `dates`.`d_datekey`, `dates`.`d_date`, `dates`.`d_dayofweek`, `dates`.`d_mo
nth`, `dates`.`d_year`, `dates`.`d_yearmonthnum`, `dates`.`d_yearmonth`, `dates`.`d_daynuminweek`, `dates`.`d_daynuminmonth`, `dates`.`d_daynuminyear`
, `dates`.`d_monthnuminyear`, `dates`.`d_weeknuminyear`, `dates`.`d_sellingseason`, `dates`.`d_lastdayinweekfl`, `dates`.`d_lastdayinmonthfl`, `dates`
.`d_holidayfl`, `dates`.`d_weekdayfl`, `customer`.`c_custkey`, `customer`.`c_name`, `customer`.`c_address`, `customer`.`c_city`, `customer`.`c_nation`
, `customer`.`c_region`, `customer`.`c_phone`, `customer`.`c_mktsegment`, `part`.`p_partkey`, `part`.`p_name`, `part`.`p_mfgr`, `part`.`p_category`, `
part`.`p_brand`, `part`.`p_color`, `part`.`p_type`, `part`.`p_size`, `part`.`p_container` FROM `ssb_10`.`lineorder`, `ssb_10`.`supplier`, `ssb_10`.`da
tes`, `ssb_10`.`customer`, `ssb_10`.`part`
WHERE `lineorder`.`lo_orderdate` = `dates`.`d_datekey` AND `lineorder`.`lo_custkey` = `customer`.`c_custkey` AND `lineorder`.`lo_suppkey` = `supplier`
.`s_suppkey` AND `lineorder`.`lo_partkey` = `part`.`p_partkey` |          |
| Table Properties              | [transient_lastDdlTime=1577677599]                 |          |
| Serde Library                 | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe |          |
| InputFormat                   | org.apache.hadoop.mapred.TextInputFormat           |          |
| OutputFormat                  | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat |          |
| Storage Properties            | [serialization.format=1]                           |          |
+-------------------------------+----------------------------------------------------+----------+
73 rows selected (0.057 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

The following figure shows the execution plan of simple queries on the lineorder_flatten view.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(11) SortMergeJoin [lo_partkey#313], [p_partkey#359], Inner
   :- *(8) Sort [lo_partkey#313 ASC NULLS FIRST], false, 0
   :  +- Exchange hashpartitioning(lo_partkey#313, 200)
   :     +- *(7) SortMergeJoin [lo_custkey#312], [c_custkey#351], Inner
   :        :- *(4) Sort [lo_custkey#312 ASC NULLS FIRST], false, 0
   :        :  +- Exchange hashpartitioning(lo_custkey#312, 200)
   :        :     +- *(3) BroadcastHashJoin [lo_orderdate#315], [d_datekey#334], Inner, BuildRight
   :        :        :- *(3) BroadcastHashJoin [lo_suppkey#314], [s_suppkey#327], Inner, BuildRight
   :        :        :  :- *(3) Filter (((isnotnull(lo_suppkey#314) && isnotnull(lo_orderdate#315)) && isnotnull(lo_custkey#312)) && isnotnull(lo_part
key#313))
   :        :        :  :  +- Scan hive ssb_10.lineorder [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_ord
erdate#315, lo_orderpriotity#316, lo_shippriotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#32
2L, lo_supplycost#323L, lo_tax#324L, lo_commitdate#325, lo_shipmode#326], HiveTableRelation `ssb_10`.`lineorder`, org.apache.hadoop.hive.serde2.lazy.L
azySimpleSerDe, [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_orderdate#315, lo_orderpriotity#316, lo_ship
priotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#322L, lo_supplycost#323L, lo_tax#324L, lo_c
ommitdate#325, lo_shipmode#326]
   :        :        :  +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :        :     +- *(1) Filter isnotnull(s_suppkey#327)
   :        :        :        +- Scan hive ssb_10.supplier [s_suppkey#327, s_name#328, s_address#329, s_city#330, s_nation#331, s_region#332, s_phone#
333], HiveTableRelation `ssb_10`.`supplier`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [s_suppkey#327, s_name#328, s_address#329, s_city#330
, s_nation#331, s_region#332, s_phone#333]
   :        :        +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :           +- *(2) Filter isnotnull(d_datekey#334)
   :        :              +- Scan hive ssb_10.dates [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearm
onth#340, d_daynuminweek#341, d_daynuminmonth#342, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweek
fl#347, d_lastdayinmonthfl#348, d_holidayfl#349, d_weekdayfl#350], HiveTableRelation `ssb_10`.`dates`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSe
rDe, [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearmonth#340, d_daynuminweek#341, d_daynuminmonth#34
2, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweekfl#347, d_lastdayinmonthfl#348, d_holidayfl#349,
 d_weekdayfl#350]
   :        +- *(6) Sort [c_custkey#351 ASC NULLS FIRST], false, 0
   :           +- Exchange hashpartitioning(c_custkey#351, 200)
   :              +- *(5) Filter ((isnotnull(c_region#356) && (c_region#356 = ASIA)) && isnotnull(c_custkey#351))
   :                 +- Scan hive ssb_10.customer [c_custkey#351, c_name#352, c_address#353, c_city#354, c_nation#355, c_region#356, c_phone#357, c_mk
tsegment#358], HiveTableRelation `ssb_10`.`customer`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [c_custkey#351, c_name#352, c_address#353, c
_city#354, c_nation#355, c_region#356, c_phone#357, c_mktsegment#358]
   +- *(10) Sort [p_partkey#359 ASC NULLS FIRST], false, 0
      +- Exchange hashpartitioning(p_partkey#359, 200)
         +- *(9) Filter isnotnull(p_partkey#359)
            +- Scan hive ssb_10.part [p_partkey#359, p_name#360, p_mfgr#361, p_category#362, p_brand#363, p_color#364, p_type#365, p_size#366, p_conta
iner#367], HiveTableRelation `ssb_10`.`part`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [p_partkey#359, p_name#360, p_mfgr#361, p_category#3
62, p_brand#363, p_color#364, p_type#365, p_size#366, p_container#367] |
+----------------------------------------------------+
1 row selected (0.435 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

After a raw cache is created and built for the lineorder_flatten view, the same queries are performed on the view. EMR Spark automatically uses the cache to optimize the original execution plan. The following figure shows the new execution plan.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:17:47 INFO [main] HiveConf: Found configuration file file:/etc/ecm/spark-conf-2.4.3-hadoop2.8-1.4.2/hive-site.xml
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(1) Project [lo_orderkey#128L, lo_linenumber#129L, lo_custkey#130, lo_partkey#131, lo_suppkey#132, lo_orderdate#133, lo_orderpriotity#134, lo_ship
priotity#135, lo_quantity#136L, lo_extendedprice#137L, lo_ordtotalprice#138L, lo_discount#139L, lo_revenue#140L, lo_supplycost#141L, lo_tax#142L, lo_c
ommitdate#143, lo_shipmode#144, s_suppkey#145, s_name#146, s_address#147, s_city#148, s_nation#149, s_region#150, s_phone#151, ... 34 more fields]
   +- *(1) Filter (isnotnull(c_region#174) && (c_region#174 = ASIA))
      +- *(1) FileScan parquet [lo_orderkey#128L,lo_linenumber#129L,lo_custkey#130,lo_partkey#131,lo_suppkey#132,lo_orderpriotity#134,lo_shippriotity#
135,lo_quantity#136L,lo_extendedprice#137L,lo_ordtotalprice#138L,lo_discount#139L,lo_revenue#140L,lo_supplycost#141L,lo_tax#142L,lo_commitdate#143,lo_
shipmode#144,s_suppkey#145,s_name#146,s_address#147,s_city#148,s_nation#149,s_region#150,s_phone#151,d_datekey#152,... 34 more fields] Batched: true,
Format: Parquet, Location: InMemoryFileIndex[/user/hive/warehouse/ssb_10/_CACHE/lineorder_flatten/raw], PartitionCount: 2406, PartitionFilters: [], Pu
shedFilters: [IsNotNull(c_region), EqualTo(c_region,ASIA)], ReadSchema: struct<lo_orderkey:bigint,lo_linenumber:bigint,lo_custkey:int,lo_partkey:int,l
o_suppkey:int,lo_or... |
+----------------------------------------------------+
```

The new execution plan no longer uses the calculation logic of the lineorder_flatten view. Instead, it directly accesses the cached data in HDFS.

Usage notes

1. JindoCube does not ensure data consistency between the cache and source table. You must manually trigger data updates or configure a periodic update policy based on your requirements for data consistency.
2. JindoCube determines whether to use a cache to optimize execution plans based on the metadata of views. If cached data is incomplete when JindoCube optimizes an execution plan, the query results may be invalid or incomplete. The following operations or configurations may cause incomplete cached data: 1. Delete partition data that is required in subsequent queries on the cache details page. 2. Data required in subsequent queries is filtered out based on the filter conditions specified when you build or update the cache. 3. Required data has not been updated to the cache when the query is performed.

6.4.3. Use JindoTable

JindoTable is used to implement tiered storage, optimize table files, and collect data statistics based on the popularity of tables or partitions. This topic describes how to use JindoTable.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your on-premises machine.
- An EMR cluster of the 3.30.0 version or later is created. For more information about how to create a cluster, see Create a cluster.

Use JindoTable

Common commands:

- -accessStat
- -cache
- -archive
- -unarchive
- -uncache
- -status
- -optimize
- -showTable
- -showPartition
- -listTables
- -dumpmc

> **Notice** Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

-accessStat

- Syntax

  jindo table -accessStat {-d} *<days>* **{-n}** <topNums>

- Description

  This command is used to query the access records in which tables or partitions are visited most in a specified time range.

  <days> and <topNums> must be positive integers. If <days> is 1, all the access records generated from 00:00 (local time) on the current day to the current time are queried.

- Example: Query the 20 access records of tables or partitions that are most frequently visited within the last seven days.

```
jindo table -accessStat -d 7 -n 20
```

-cache

- Syntax

jindo table -cache {-t} <dbName.tableName> **[-p]** <partitionSpec> **[-pin]**

- Description

  This command is used to cache data of specified tables or partitions to local disks.

  The data of the tables or partitions must be stored in Object Storage Service (OSS) or JindoFileSystem (JindoFS). Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`. When you specify `-pin`, if the cache space is insufficient, do not delete related data if possible.

- Example: Cache data of the db1.t1 table that is generated on March 16, 2020 to local disks.

```
jindo table -cache -t db1.t1 -p date=2020-03-16
```

-uncache

- Syntax

  jindo table -uncache {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to delete cached data of specified tables or partitions from local disks.

  The data of the tables or partitions must be stored in OSS or JindoFS. Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

- Examples:
  - Delete cached data of the db1.t2 table from local disks.

```
jindo table -uncache -t db1.t2
```

  - Delete cached data of the db1.t1 table from local disks.

```
jindo table -uncache -t db1.t1 -p date=2020-03-16,category=1
```

-archive

- Syntax

  jindo table -archive {-a|i} {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to lower the level of the policy that is used to store data of specified tables or partitions. By default, the Archive storage class is used.

  To use the Infrequent Access (IA) storage class, add -i to the command. Specify tables in the format of database.table. Specify partitions in the format of 'partitionCol1=1,partitionCol2=2,...'.

- Example: Cache data of the db1.t1 table to local disks.

```
jindo table -archive -t db1.t1 -p date=2020-10-12
```

-unarchive

- Syntax

  jindo table -unarchive [-o|-i] {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

This command is used to change the storage class from Archive to Standard.

If `-o` is added to the command, an Archived object is temporarily restored. If `-i` is added to the command, an Archived object is changed to an IA object.

- Example:

```
jindo table -unarchive -o -t db1.t1 -p date=2020-03-16,category=1
```

```
jindo table -unarchive -i -t db1.t2
```

-status

- Syntax

  jindo table -status {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to view the data storage status of specified tables or partitions.

- Examples:
  - View the data storage status of the db1.t2 table.

    ```
    jindo table -status -t db1.t2
    ```

  - View the data storage status of the db1.t1 table on March 16, 2020.

    ```
    jindo table -status -t db1.t1 -p date=2020-03-16
    ```

-optimize

- Syntax

  jindo table -optimize {-t} <dbName.tableName>

- Description

  This command is used to optimize the data organization of tables at the storage layer.

- Example: Optimize the data organization of the db1.t1 table at the storage layer.

```
jindo table -optimize -t db1.t1
```

-showTable

- Syntax

  jindo table -showTable {-t} <dbName.tableName>

- Description

  This command is used to display all partitions in a partitioned table or display the data storage of a non-partitioned table.

- Example: Display all partitions in the db1.t1 partitioned table.

```
jindo table -showTable -t db1.t1
```

-showPartition

- Syntax

  jindo table -showPartition {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to display the data storage of partitions.

- Example: Display the data storage of all partitions in the db1.t1 partitioned table on October 12, 2020.

```
jindo table -showPartition -t db1.t1 -p date=2020-10-12
```

-listTables

- Syntax

  jindo table -listTables [-db] <dbName>

- Description

  This command is used to display all tables in a specified database. If you do not specify `[-db]`, tables in the default database are displayed.

- Examples:
  - Display tables in the default database.

```
jindo table -listTables
```

  - Display tables in the db1 database.

```
jindo table -listTables -db db1
```

-dumpmc

- Syntax

  jindo table -dumpmc {-i} <accessId> **{-k}** <accessKey> **{-m}** <numMaps> **{-t}** <tunnelUrl> **{-project}** <projectName> **{-table}** <tablename> **{-p}** <partitionSpec> **{-f}** <csv|tfrecord> **{-o}** <outputPath>

Parameter	Description	Required
-i	The AccessKey ID of your Alibaba Cloud account.	Yes
-k	The AccessKey secret of your Alibaba Cloud account.	Yes
-m	The number of map tasks.	Yes
-t	The VPC Tunnel endpoint of MaxCompute.	Yes
-project	The name of the MaxCompute project.	Yes

Parameter	Description	Required
-table	The name of the MaxCompute table.	Yes
-p	The partition information. Example: `pt=xxx`. Separate multiple partitions with commas (,), such as `pt=xxx,dt=xxx`.	No
-f	The file format. Valid values: ○ tfrecord ○ csv	Yes
-o	The destination path.	Yes

- Description

  This command is used to dump MaxCompute tables to an EMR cluster or OSS. The formats CSV and TFRECORD are supported.

- Examples:
  - Dump a MaxCompute table in the TFRECORD format to an EMR cluster.

    ```
    jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
    .aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o /tmp/outputtf1 -f tfrecord
    ```

  - Dump a MaxCompute table in the CSV format to OSS.

    ```
    jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
    .aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o oss://bucket1/tmp/outputcsv -f csv
    ```

6.4.4. Use JindoTable to collect access frequency statistics on tables and partitions

You can use JindoTable to collect access frequency statistics on tables and partitions, and separate cold and hot data based on these statistics. This helps you reduce storage costs and improve cache usage efficiency.

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

JindoTable can be used to collect access frequency statistics on Hive tables. The collected data is saved in the namespaces of the SmartData service of a cluster.

In SmartData 3.2.X and later, JindoTable can be used to collect access frequency statistics on Spark, Hive, and Presto tables. This feature is enabled for Spark and Presto by default. If you want to disable the feature for Spark or Presto, perform the operations described in Disable the collection of access frequency statistics. The feature is disabled for Hive by default. If you want to enable the feature for Hive, perform the operations described in Enable the collection of access frequency statistics for Hive.

Query access frequency statistics

You can run a command to query access frequency statistics.

- Syntax

```
jindo table -accessStat <-d [days]> <-n [topNums]>
```

  Set `days` and `topNums` to positive integers. For example, if days is set to 1 and topNums is not specified, the access frequency statistics of all the tables or partitions that are visited on the current day (from 00:00 to the current time) are queried.

- Description

  This command is used to query the first N tables or partitions that are visited the most frequently within a specific period of time, and the number of times each of them is visited.

- For example, you can run the following command to query the first 20 tables or partitions that have been visited the most frequently within the last seven days, and the number of times each of them has been visited:

```
jindo table -accessStat -d 7 -n 20
```

For more information about how to use JindoTable, see Use JindoTable.

Enable the collection of access frequency statistics for Hive

1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of a Hive parameter.
   i. In the left-side navigation pane, choose **Cluster Service > Hive**.
   ii. On the Hive service page, click the **Configure** tab.
   iii. Search for the hive.exec.post.hooks parameter and add **com.aliyun.emr.table.hive.HivePostHook** to the end of the existing parameter value. Separate the hooks with a comma (,).
6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   i. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.

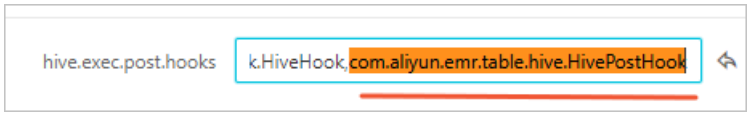
ii. In the **Cluster Activities** dialog box, specify **Description**.

iii. Click **OK**.

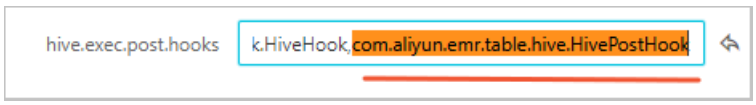
iv. In the **Confirm** message, click **OK**.

Disable the collection of access frequency statistics

1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of the related parameter.
   - Hive:
     a. In the left-side navigation pane, choose **Cluster Service > Hive**.
     b. On the Hive service page, click the **Configure** tab.
     c. Search for the hive.exec.post.hooks parameter and delete **com.aliyun.emr.table.hive.HivePostHook** from the parameter value.



   - Spark:
     a. In the left-side navigation pane, choose **Cluster Service > Spark**.
     b. On the Spark service page, click the **Configure** tab.
     c. Search for the spark.sql.queryExecutionListeners parameter and delete **com.aliyun.emr.table.spark.SparkSQLQueryListener** from the parameter value.



   - Presto:
     a. In the left-side navigation pane, choose **Cluster Service > Presto**.
     b. On the Presto service page, click the **Configure** tab.
     c. Search for the event-listener.name parameter and delete the parameter value.
6. Save the configurations.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.

   iii. Click **OK**.
7. Restart the related service.
   - Hive:
     a. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
     b. In the **Cluster Activities** dialog box, specify **Description**.

c. Click **OK**.

d. In the **Confirm** message, click **OK**.

- Spark:

  a. In the upper-right corner of the Spark service page, choose **Actions > Restart ThriftServer**.

  b. In the **Cluster Activities** dialog box, specify **Description**.

  c. Click **OK**.

  d. In the **Confirm** message, click **OK**.

- Presto:

  a. In the upper-right corner of the Presto service page, choose **Actions > Restart All Components**.

  b. In the **Cluster Activities** dialog box, specify **Description**.

  c. Click **OK**.

  d. In the **Confirm** message, click **OK**.

6.4.5. Use JindoTable to collect infrequent-access statistics of tables and partitions

JindoTable can be used to collect infrequent-access statistics of tables and partitions. It enables you to find the data that is not accessed recently based on the last access time of tables and partitions. This helps optimize data storage and reduce costs. For example, in data analytics scenarios, you can use this feature to move infrequently used partitions to lower-cost storage media.

Prerequisites

A cluster is created in E-MapReduce (EMR) V3.35.0 or later V3.X.X, or in EMR V4.9.0 or later V4.X.X. For more information, see Create a cluster.

Context

In SmartData 3.5.X and later, JindoTable can be used to collect infrequent-access statistics of Hive, Spark, and Presto tables. This feature is disabled by default. For more information about how to enable this feature, see Enable the collection of infrequent-access statistics for Spark, Enable the collection of infrequent-access statistics for Hive, and Enable the collection of infrequent-access statistics for Presto.

> **Note** The same hooks and listeners are used to collect frequent-access and infrequent-access statistics. Therefore, if you enable the collection of infrequent-access statistics, the collection of frequent-access statistics is also enabled. For more information about the collection of frequent-access statistics, see Use JindoTable to collect access frequency statistics on tables and partitions.

Limits

- Data Lake Formation (DLF) metadata is not supported.
- You must have the permissions to access the underlying metadata storage, such as MySQL or

ApsaraDB RDS, from the IP addresses of the nodes on which the Hive CLI, HiveServer2, Spark SQL CLI, Spark Thrift Server, and Presto services reside.

- You can use JindoTable to collect infrequent-access statistics only of Hive, Spark, and Presto tables.

Query infrequent-access statistics

You can run a command to query infrequent-access statistics.

- Syntax

```
jindo table -leastUseStat -n <num> [-i/-ignoreNever]
```

num indicates the number of tables or partitions that you want to query. Set this parameter to a positive integer. -i/-ignoreNever is an optional parameter. If you specify this parameter, the tables or partitions that have never been accessed are filtered out.

- Description

This command is used to query the tables or partitions that have not been accessed for the longest time.

- Example: Query the first 20 tables or partitions that have not been accessed for the longest time.

```
jindo table -leastUseStat -n 20
```

The following figure shows the output.

```
tdb.t1      pid=20/qid=ten      2021-03-26 13:53:52
tdb.t2                          2021-03-26 13:53:58
```

  - The first column lists tables in the Database name.Table name format.
  - The second column lists partitions in the First partition key column=Column value/Second partition key column=Column value/... format. For a non-partitioned table, this column is empty.
  - The third column displays the last access time of the tables or partitions in the yyyy-MM-dd HH:mm:ss format.

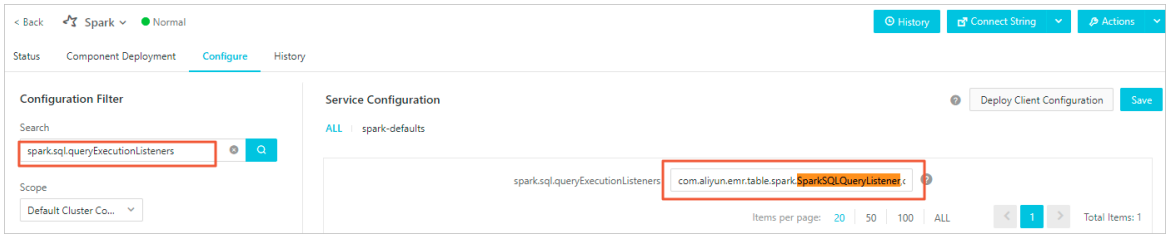
> **Note** For a partitioned table, only partition-level access statistics are displayed. Table-level access statistics are not displayed.

For more information about how to use JindoTable, see Use JindoTable.

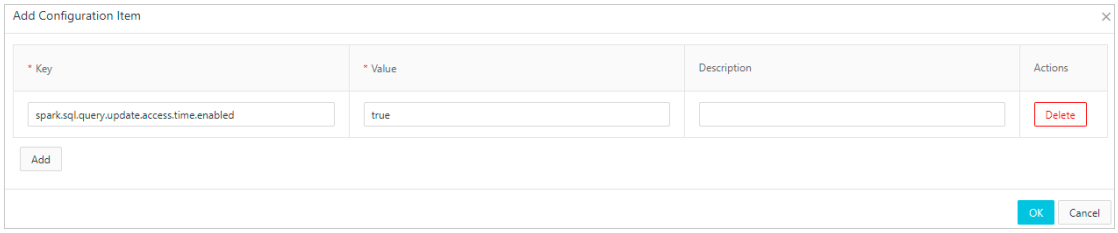
Enable the collection of infrequent-access statistics for Spark

1. Go to the Spark service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Spark**.
2. On the Spark service page, click the **Configure** tab.
3. Search for the spark.sql.queryExecutionListeners parameter and check whether the value contains

com.aliyun.emr.table.spark.SparkSQLQueryListener. If the value does not contain this listener, add the listener to the value. Separate multiple listeners with comma(,).



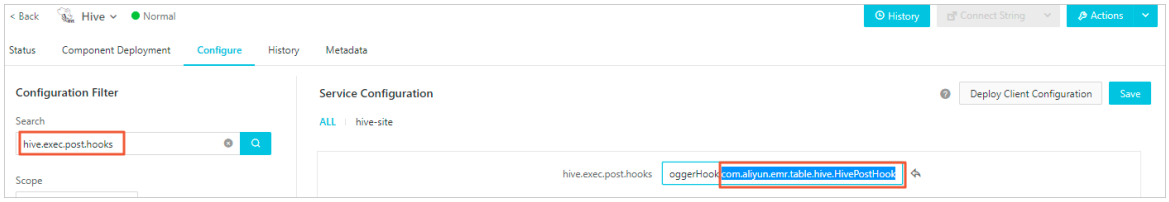
4. Add custom configurations.
   i. In the **Service Configuration** section, click the **spark-defaults** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **spark.sql.query.update.access.time.enabled** parameter and set it to **true**.



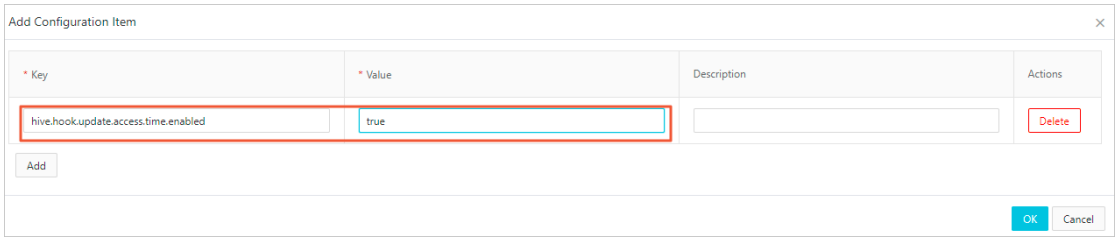
   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

Enable the collection of infrequent-access statistics for Hive

1. Go to the Hive service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Hive**.
2. On the Hive service page, click the **Configure** tab.
3. Search for the hive.exec.post.hooks parameter and check whether the value contains com.aliyun.emr.table.hive.HivePostHook. If the value does not contain this hook, add the hook to the value. Separate multiple hooks with comma(,).



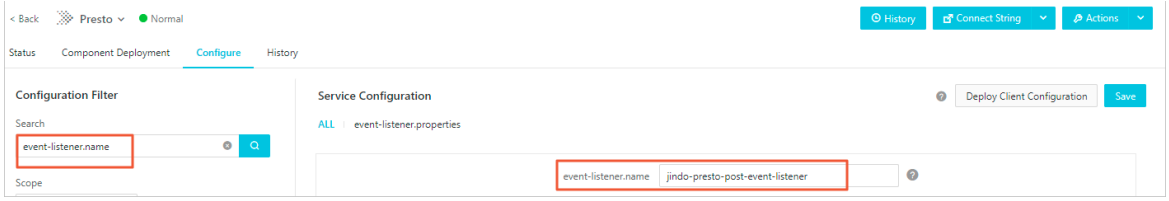
4. Add custom configurations.
   i. In the **Service Configuration** section, click the **hive-site** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **hive.hook.update.access.time.enabled** parameter and set it to **true**.



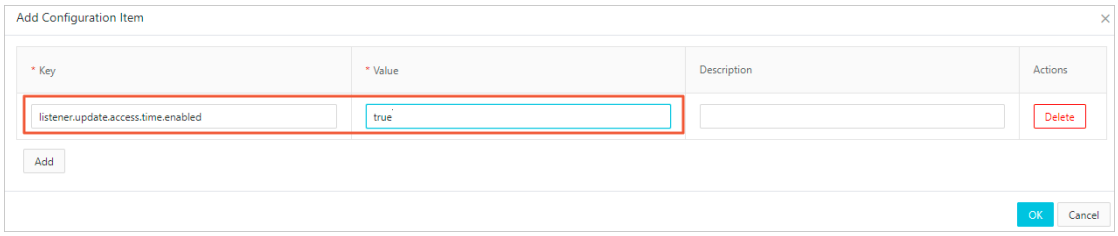
   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

Enable the collection of infrequent-access statistics for Presto

1. Go to the Presto service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Presto**.
2. On the Presto service page, click the **Configure** tab.
3. Search for the event-listener.name parameter and check whether the value contains jindo-presto-post-event-listener. If the value does not contain this listener, add the listener to the value.



4. Add custom configurations.
   i. In the **Service Configuration** section, click the **event-listener.properties** tab.
   ii. Click **Custom Configuration** in the upper-right corner.
   iii. In the **Add Configuration Item** dialog box, add the **listener.update.access.time.enabled** parameter and set it to **true**.



   iv. Click **OK**.
5. Save the configurations.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
6. Restart all components.
   i. In the upper-right corner of the Spark service page, choose **Actions > Restart All Components**.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

6.5. Toolset

6.5.1. Use JindoFS FUSE

This topic describes how to access JindoFileSystem (JindoFS) from a Filesystem in Userspace (FUSE) client. FUSE allows you to use JindoFS in block storage mode and cache mode (only JFS Scheme supported).

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

FUSE is a method provided by the Linux kernel to mount a file system. You can use a FUSE client to map files in a JindoFS cluster to a local disk. Then, you can directly access data in the JindoFS cluster without the need to use the `hadoop fs -ls jfs://<namespace>/` command.

Mount a directory

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to create a directory:

```
mkdir /mnt/jfs
```

3. Run the following command to mount the directory:

```
jindofs-fuse /mnt/jfs
```

In this example, the */mnt/jfs/* directory is mounted.

Read and write files

1. Run the following command to query all the sub-directories under the /mnt/jfs/ directory:

```
ls /mnt/jfs/
```

All the namespaces that you have configured on the server are returned.

```
test  testcache
```

2. Run the following command to query all the files in the test namespace:

```
ls /mnt/jfs/test/
```

3. Run the following command to create a directory:

```
mkdir /mnt/jfs/test/dir1
ls /mnt/jfs/test/
```

4. Run the following command to write data into a file:

```
echo "hello world" > /tmp/hello.txt
cp /tmp/hello.txt /mnt/jfs/test/dir1/
```

5. Run the following command to read data from a file:

```
cat /mnt/jfs/test/dir1/hello.txt
```

The following information is returned:

```
hello world
```

The following example shows how to read data from and write data into files in Python:

1. Write data into a file in Python. Create the *write.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'w',encoding = 'utf-8') as f:
   f.write("my first file\n")
   f.write("This file\n\n")
   f.write("contains three lines\n")
```

2. Read data from a file in Python. Create the *read.py* script file, which contains the following

content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'r',encoding = 'utf-8') as f:
    lines = f.readlines()
    [print(x, end = '') for x in lines]
```

Run the following command to read the data that is written into the *test.txt* file:

```
[hadoop@emr-header-1 ~]$ ./read.py
```

The following information is returned:

```
my first file
This file
```

Unmount FUSE

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to unmount FUSE:

   ```
   umount jindofs-fuse
   ```

   If the `target is busy` error is returned, switch to another directory, stop all the programs that are reading data from and writing data into FUSE files, and then unmount FUSE.

6.5.2. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain help information:

   ```
   jindo distcp --help
   ```

   The following information is returned:

```
     --help           - Print help text
     --src=VALUE          - Directory to copy files from
     --dest=VALUE              - Directory to copy files to
     --parallelism=VALUE         - Copy task parallelism
     --outputManifest=VALUE       - The name of the manifest file
     --previousManifest=VALUE   -   The path to an existing manifest file
     --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
if specified
     --copyFromManifest   -   Copy from a manifest instead of listing a directory
     --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
     --srcPattern=VALUE   -   Include only source files matching this pattern
     --deleteOnSuccess   -   Delete input files after a successful copy
     --outputCodec=VALUE   -   Compression codec for output files
     --groupBy=VALUE   -   Pattern to group input files by
     --targetSize=VALUE   -   Target size for output files
     --enableBalancePlan   -   Enable plan copy task to make balance
     --enableDynamicPlan   -   Enable plan copy task dynamically
     --enableTransaction   -   Enable transation on Job explicitly
     --diff   -   show the difference between src and dest filelist
     --ossKey=VALUE   -   Specify your oss key if needed
     --ossSecret=VALUE   -   Specify your oss secret if needed
     --ossEndPoint=VALUE   -   Specify your oss endPoint if needed
     --policy=VALUE   -   Specify your oss storage policy
     --cleanUpPending   -   clean up the incomplete upload when distcp job finish
     --queue=VALUE   -   Specify yarn queuename if needed
     --bandwidth=VALUE   -   Specify bandwidth per map/reduce in MB if needed
     --s3Key=VALUE   -   Specify your s3 key
     --s3Secret=VALUE   -   Specify your s3 Sercet
     --s3EndPoint=VALUE   -   Specify your s3 EndPoint
```

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-
01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03
/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03
/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017
-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-0
1/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-0
1/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_tab
le/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/
03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/
03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromMani
fest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

Note You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

> **Note** If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

--queue

--queue specifies the name of the YARN queue where the current DistCp task resides.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<your_bucket>/hourly_table --qu
eue yarnqueue
```

--bandwidth

--bandwidth specifies the bandwidth allocated to the current DistCp task. This prevents the task from using excessive bandwidth. Unit: MB/s.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.c
om --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy archive --parallelism 20
  ```

- Example of the IA storage class (`--ia`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy ia --parallelism 20
  ```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

Use Amazon S3 as a data source

You can use the --s3Key, --s3Secret, and --s3EndPoint parameters in a command to specify information related to Amazon S3.

Example:

```
jindo distcp jindo-distcp-2.7.3.jar --src s3a://yourbucket/ --dest oss://<your_bucket>/hour
ly_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com
```

You can configure the s3Key, s3Secret, and s3EndPoint parameters in the *core-site.xml* file of Hadoop. This way, you do not need to specify an AccessKey pair each time.

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

Example:

```
jindo distcp /tmp/jindo-distcp-2.7.3.jar --src s3://smartdata1/ --dest s3://smartdata1/tmp
--s3EndPoint  s3-us-west-1.amazonaws.com
```

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

> **Note** If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

6.5.3. Use Jindo DistCp in specific scenarios

This topic describes how to use Jindo DistCp in specific scenarios.

Prerequisites

- An E-MapReduce (EMR) cluster of a required version is created. For more information, see Create a cluster.
- Java Development Kit (JDK) 1.8 is installed.
- *jindo-distcp-<version>.jar* is downloaded based on your Hadoop version.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 2.7 and later 2.X versions.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 3.X.

Use scenarios

Common use scenarios of Jindo DistCp:

- Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?
- Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?
- Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?
- Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?
- Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file size?
- Scenario 8: What parameters are required if Amazon S3 is used as a data source?
- Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?
- Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?
- Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?
- Scenario 12: What parameters are required if I want to delete original files after a copy operation?
- Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?

If you are not using EMR, the following conditions must be met before you can import HDFS data to Object Storage Service (OSS):

- You can read data from HDFS.
- The AccessKey ID, AccessKey secret, and endpoint of OSS are obtained. You can write data to a destination OSS bucket.
- The storage class of the OSS bucket is not Archive.
- You can submit MapReduce tasks.
- The JAR package of Jindo DistCp is downloaded.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 10
```

> **Note** For more information about the parameters, see Use Jindo DistCp.

To import a large amount of data or a large number of files, such as millions or tens of millions of files, to OSS, you can set parallelism to a large value to increase the concurrency. You can also set `enableBatch` to true for optimization. Optimization command:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 500 --enableBatch
```

Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?

You can use one of the following methods to verify data integrity:

- Jindo DistCp Counters

  Check DistCp Counters in the counter information of a MapReduce task.

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

where:

- Bytes Destination Copied specifies the size of the file that you copy to a destination directory, in bytes.
- Bytes Source Read specifies the size of the file that you read from the source directory, in bytes.
- Files Copied specifies the number of copied files.

- Jindo DistCp --diff

  Use `--diff` to compare the information of files in the source and destination directories. The information includes file name and file size. Missing files and the files that fail to be copied are recorded in a manifest file. The manifest file is generated in the directory where you perform the comparison.

  Add `--diff` to the command described in Scenario 1. Example:

  ```
  hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
  -hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
  aliyuncs.com --diff
  ```

  If all files are copied, the following information is returned:

  ```
  INFO distcp.JindoDistCp: distcp has been done completely
  ```

Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?

If your DistCp task fails, and you want to perform resumable upload to copy only the files that fail to be copied, perform the following operations on the basis of the command described in Scenario 1:

1. Add `--diff` to check whether all files are copied.

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
   hou.aliyuncs.com --diff
   ```

   If all the files are copied, the following information is returned. Otherwise, a manifest file is generated. In this case, go to the next step.

   ```
   INFO distcp.JindoDistCp: distcp has been done completely.
   ```

2. Use `--copyFromManifest` and `--previousManifest` to copy the remaining files. Example:

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --dest oss://yang-hhht/hourly_table --previousManifest=file:///opt
   /manifest-2020-04-17.gz --copyFromManifest --parallelism 20
   ```

   `file:///opt/manifest-2020-04-17.gz` is the path where the manifest file is stored.

Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?

1. If no information is generated for the files that were copied last time, specify parameters to

generate a manifest file to record the information about the files that are copied.

Add `--outputManifest=manifest-2020-04-17.gz` and `--requirePreviousManifest=false` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

where:

- `--outputManifest` generates a manifest file whose name can be customized and must have a file name extension of gz, such as *manifest-2020-04-17.gz*. This file is stored in the directory specified by `--dest`.
- `--requirePreviousManifest` specifies whether to require a previous manifest file.

2. After a DistCp task is complete, copy the new files that may be generated in the source directory.

   Add `--outputManifest=manifest-2020-04-17.gz` and `--previousManifest=oss://yang-hhht/hourly_table/manifest-2020-04-17.gz` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

```
hadoop jar jindo-distcp-2.7.3.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --outputManifest=manifest-2020-04-18.gz --previousManifest=oss://yang-hhht
/hourly_table/manifest-2020-04-17.gz --parallelism 10
```

3. Repeat Step 2 to continuously copy incremental files.

Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?

Add the following two parameters to the command described in Scenario 1. These two parameters can be used together or separately.

- `--queue`: the name of the YARN queue.
- `--bandwidth`: the size of the specified bandwidth , in MB.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --queue yarnqueue --bandwidth 6 --parallelism 10
```

Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?

- To write data to OSS Archive storage, add `--archive` to the command described in Scenario 1.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --archive --parallelism 20
```

- To write data to OSS Infrequent Access (IA) storage, add `--ia` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --ia --parallelism 20
```

Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file sizes?

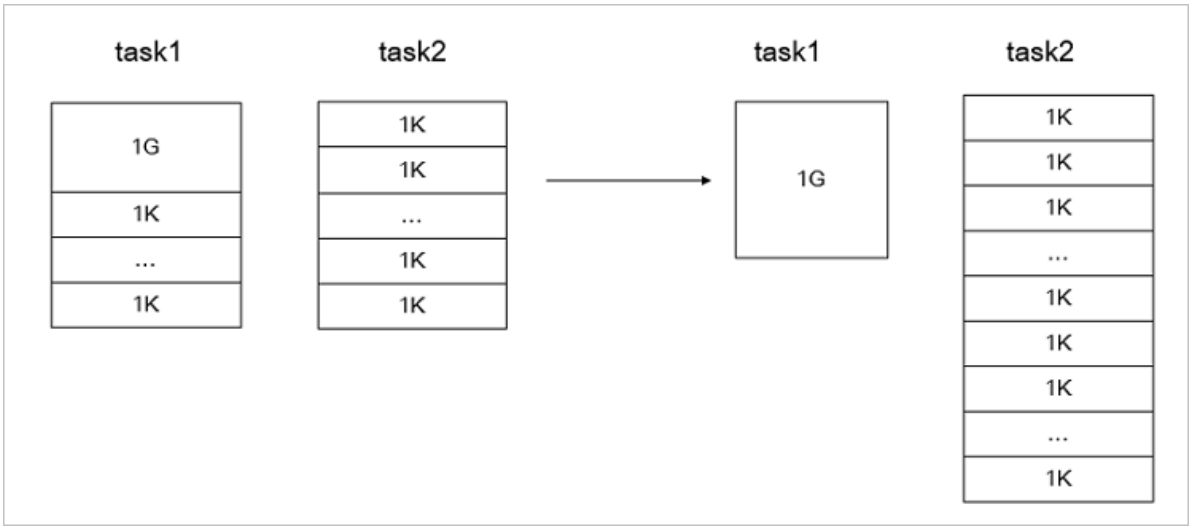
- A large number of small files and large data volume for a single large file

  If most of the files that you want to copy are small files, and the data volume of a single large file is large, the general solution is to randomly allocate the files to be copied. In this case, if you do not optimize the job allocation plan, few large files and many small files may be allocated to the same copy process. This cannot achieve excellent copy performance.

  Add `--enableDynamicPlan` to the command described in Scenario 1 to enable the optimization. --enableDynamicPlan cannot be used with `--enableBalancePlan`. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableDynamicPlan --parallelism 10
```

  The following figure shows the data copy before and after the plan is optimized.

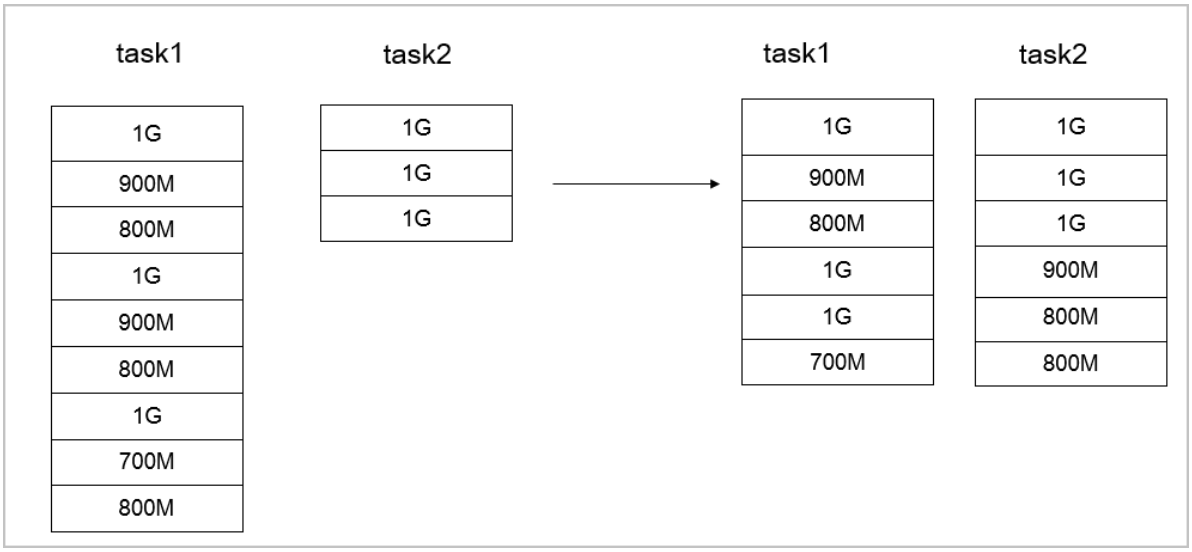


- No significant differences in sizes of files

  If the sizes are not significantly different among the files that you want to copy, use `--enableBalancePlan` to optimize the job allocation plan. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableBalancePlan --parallelism 10
```

The following figure shows the data copy before and after the plan is optimized.



Scenario 8: What parameters are required if Amazon S3 is used as a data source?

Replace the parameters that specify the AccessKey ID, AccessKey secret, and endpoint of OSS in the command described in Scenario 1 with the following parameters of Amazon S3:

- `--s3Key` : the AccessKey ID for Amazon S3
- `--s3Secret` : the AccessKey secret for Amazon S3
- `--s3EndPoint` : the endpoint for Amazon S3

Example:

```
hadoop jar jindo-distcp-<version>.jar --src s3a://yourbucket/ --dest oss://yang-hhht/hourly
_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com --para
llelism 10
```

Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?

You can use `--outputCodec` to compress copied files in a format such as LZO or GZ to reduce the space that is used to store the files.

Add `--outputCodec` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --outputCodec=gz --parallelism 10
```

You can set --outputCodec to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files are compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change to the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?

- If you want to copy the files that meet specific rules, add `--srcPattern` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPattern .*\.log --parallelism 10
```

  `--srcPattern`: specifies a regular expression that filters files for the copy operation.

- If you want to copy the files stored in some sub-directories of the same parent directory, add `--srcPrefixesFile` to the command described in Scenario 1.

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

  `--srcPrefixesFile`: enables Jindo DistCp to copy files in multiple folders under the same parent directory at a time.

  Content of the *folders.txt* file:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?

Add the following parameters to the command described in Scenario 1:

- `--targetSize`: the maximum size of the merged file, in MB
- `--groupBy`: the merging rule, which is a regular expression

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Scenario 12: What parameters are required if I want to delete original files after a copy operation?

Add `--deleteOnSuccess` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --deleteOnSuccess --parallelism 10
```

Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Jindo DistCp allows you to save the AccessKey ID, AccessKey secret, and endpoint of OSS in the *core-site.xml* file so that you do not need to repeatedly specify the information.

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of OSS, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.jfs.cache.oss-accessKeyId</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-accessKeySecret</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-endpoint</name>
        <value>oss-cn-xxx.aliyuncs.com</value>
    </property>
</configuration>
```

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of Amazon S3, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

6.5.4. Use tiered storage commands of JindoFS

In E-MapReduce (EMR) V3.30, JindoFS provides a tiered storage feature. This feature allows you to store cold and hot data in different storage media. This helps reduce the data storage costs or accelerate data access.

Use jindo jfs

Run the following command to obtain the help information:

```
jindo jfs -help archive
-archive -i/a <path> ... :
  Archive commands.
```

Tiered storage commands in JindoFS are asynchronous and start only related tasks.

Common commands:

- Cache
- Uncache
- Archive
- Unarchive
- Status
- ls2 command

Cache

You can use this command to back up data stored in a specific path to local disks. Then, you can read the data from local disks, without the need to read data from Object Storage Service (OSS).

```
jindo jfs -cache -p <path>
```

`-p` can be used to ensure that local data is not cleared based on disk usage.

Uncache

You can use this command to delete backup data from local disks and store data only to OSS Standard storage.

```
jindo jfs -uncache  <path>
```

Archive

You can use this command to delete backup data from local disks and store data to OSS in Infrequent Access (IA) or Archive mode. For information about the storage classes, see Overview.

```
jindo jfs -archive -i|-a <path>
```

`-i` is used to store data to OSS in IA mode. `-a` is used to store data to OSS in Archive mode.

Unarchive

You can use this command to restore data from the Archive mode to the IA or Standard mode. You can temporarily restore data stored in Archive mode to allow the data to be readable within one day.

```
jindo jfs -unarchive -i/-o <path>
```

`-i` is used to store data to OSS in IA mode. `-o` is used to temporarily restore data stored in Archive mode to allow the data to be readable.

Status

You can use this command to view task details. By default, the number of files to be subject to tiered storage in a specific directory and the data that has been subject to tiered storage are measured.

```
jindo jfs -status -detail/-sync <path>
```

`-detail` is used to view the storage progress of file data. `-sync` indicates that the command exits after a tiered storage task is completed.

ls2 command

JindoFS provides the ls2 command that allows you to view the file storage status on the basis of Hadoop ls commands:

```
hadoop fs -ls2 <path>
```

Example of command output, which includes the file storage class:

```
drwxrwxrwx  - -         0    2020-06-05 04:27 oss://xxxx/warehouse
-rw-rw-rw-  1 Archive   1484 2020-09-23 16:40 oss://xxxx/wikipedia_data.csv
-rw-rw-rw-  1 Standard  1676 2020-06-07 20:04 oss://xxxx/wikipedia_data.json
```

7.SmartData 3.4.x

7.1. SmartData 3.4.X

SmartData is a storage service for the E-MapReduce (EMR) Jindo engine. SmartData provides centralized storage and optimized caching and computing for EMR computing engines and extends storage features. SmartData consists of JindoFS, JindoTable, and related tools. This topic describes the updates in SmartData 3.4.X.

OSS storage scalability on JindoFS

- The Object Storage Service (OSS) Recoverable OutputStream feature is added. It is implemented over the Flush and Recover APIs. This feature is suitable for highly reliable write scenarios, such as Flume-based write.
- The efficiency of rename operations on the OSS server is improved.
- The performance of list operations on directories in OSS with bucket versioning enabled is optimized. This prevents list operations on directories from being affected by a large number of temporary files.
- The performance of Jindo OSS Magic Committer for OSS with bucket versioning enabled is optimized. Jindo OSS Direct Committer is added.
- The credential provider framework is enhanced. JindoCommonCredentialsProvider is added.
- The file creation performance is optimized. A redundancy check is no longer performed when you write data to OSS.

JindoFS-based storage optimization

Data encryption is supported in JindoFS in block storage mode. You can use Alibaba Cloud Key Management Service (KMS) and the Advanced Encryption Standard (AES) algorithm to encrypt data.

JindoTable-based computing optimization

The native Optimized Row Columnar (ORC) reader of JindoTable is enhanced.

Other JindoFS tools

Jindo DistCp is enhanced. Incremental migration is optimized. For example, when you migrate HDFS data to OSS, checksums can be implemented on the source and destination paths.

Ecosystem support for JindoFS

Jindo OSS SDK for Python is added. The SDK supports basic OSS operations and is compatible with OSS2 Python libraries.

7.2. JindoFS in block storage mode

7.2.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read and write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

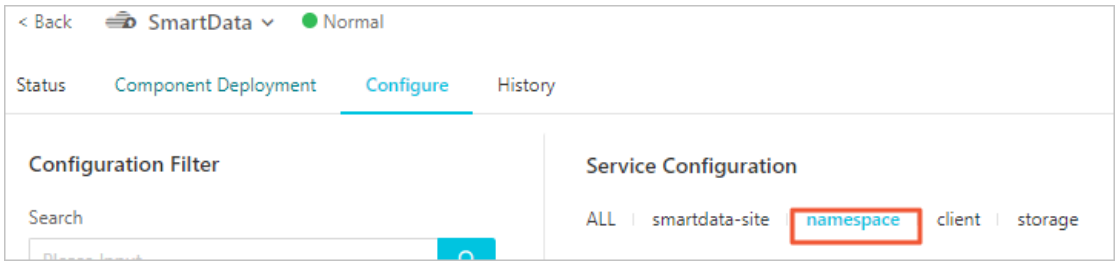
Context

The block storage mode of JindoFS has the following features:

- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.
- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

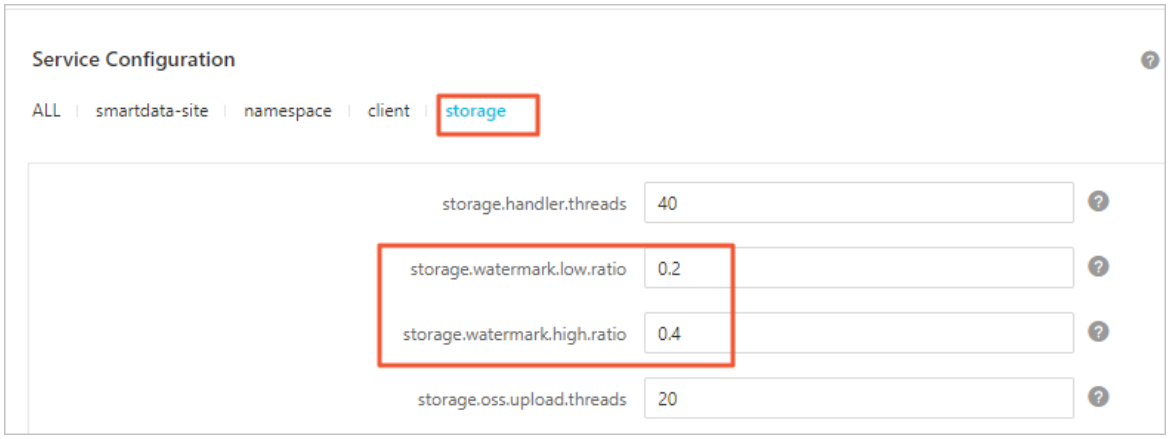
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Choose **Actions > Restart Jindo Storage Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

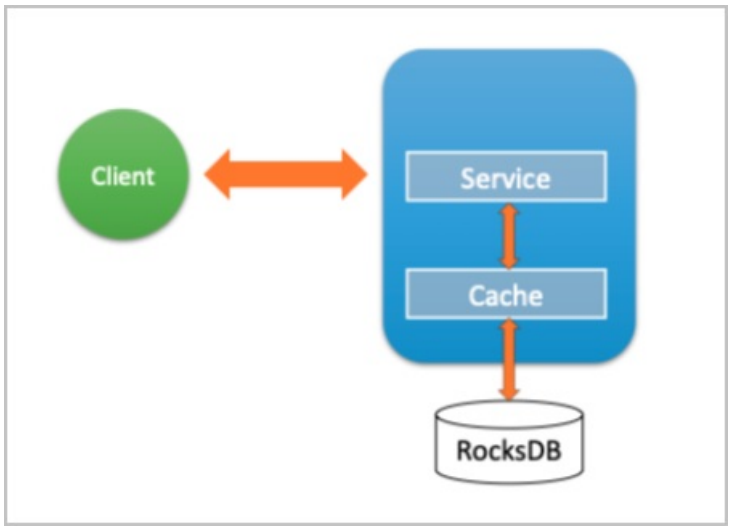
7.2.2. Use RocksDB to store metadata

Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

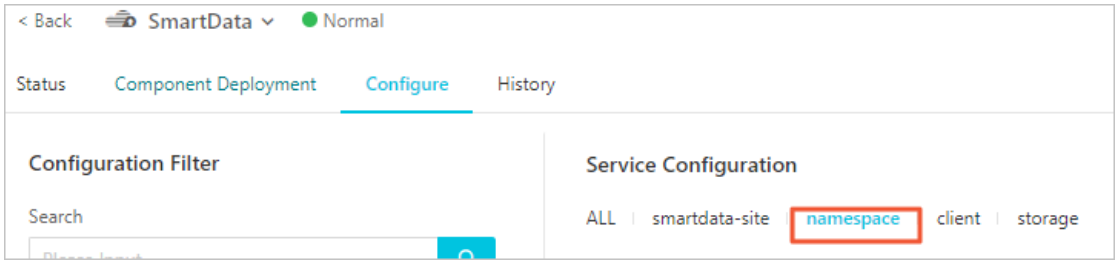
RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage backend in high availability mode, we recommend that you use Raft instances. For more information, see Use Raft-RocksDB-Tablestore to store metadata.

The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
6. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

7. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

8. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. Prepare for the recovery.

   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

   ```
   hadoop fs -count jfs://test/
   ```

   Information similar to the following example is returned:

   ```
    1596      1482809                 25 jfs://test/
   ```

   The number of folders is 1596, and the number of files is 1,482,809.

   ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1`, the latest metadata is synchronized to the Tablestore instance.

   ```
   jindo jfs -metaStatus
   ```

   Information similar to that shown in the following figure is returned.

   ```
   [hadoop@emr-header-1 ~]$ jindo jfs -metaStatus
   ==== emr-header-1:8101 ====
   OtsUploader: _synced=1
   [RocksDB Row Counts of each CF(Table)]
   ```

   iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

   ```
   jindo jfs -metaStatus
   ```

   If the state is FINISH, the recovery is completed, as shown in the following figure.

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

7.2.3. Use Raft-RocksDB-Tablestore to store metadata

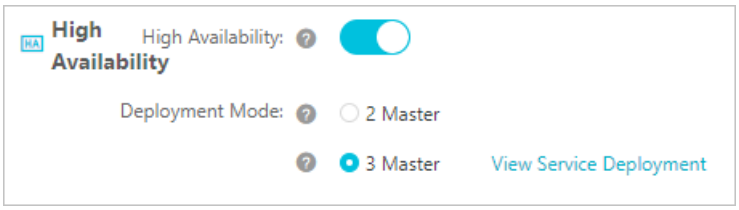
In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata.

Prerequisites

- A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  > **Note** You must enable the transaction feature.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.

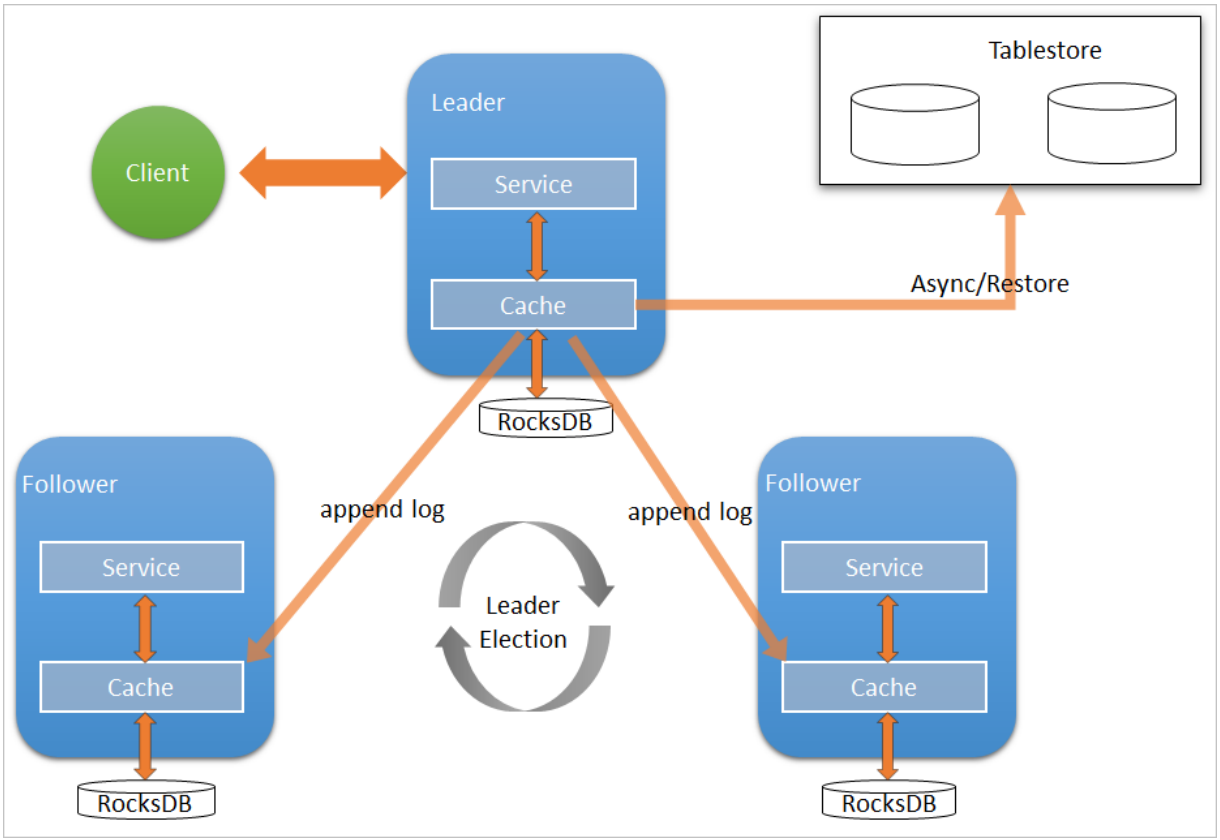


> **Note** If no deployment modes are available, submit a ticket.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.



Configure a Raft instance as the local storage backend

1. Suspend all the SmartData components of an EMR cluster.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   vi. Select **Stop All Components** from the **Actions** drop-down list in the upper-right corner.
2. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
3. Go to the **namespace** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **namespace** tab.
4. Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. Valid values: ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

Note If you do not need to use Tablestore for remote storage, skip Step 5.

5. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

```
hadoop fs -count jfs://test/
      1596      1482809                 25 jfs://test/
  (Number of folders) (Number of files)
```

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```

```
[RaftPeerImpl]
peer id: [redacted]
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 2335ms)
snapshot_timer: timeout(3600000ms) SCHEDULING(in 150305ms)
storage: [1, 624625]
disk_index: 624625
known_applied_index: 624625
last_log_id: (index=624625,term=2)
first_index_pinned: 624625
state_machine: Idle
last_committed_index: 624625
last_snapshot_index: 0
last_snapshot_term: 0
snapshot_status: IDLE
replicator_25769803789@[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=624261 ic=0
replicator_329853488332[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=623564 ic=0

OtsUploader: _lastStopIndex=624624, _synced=1
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

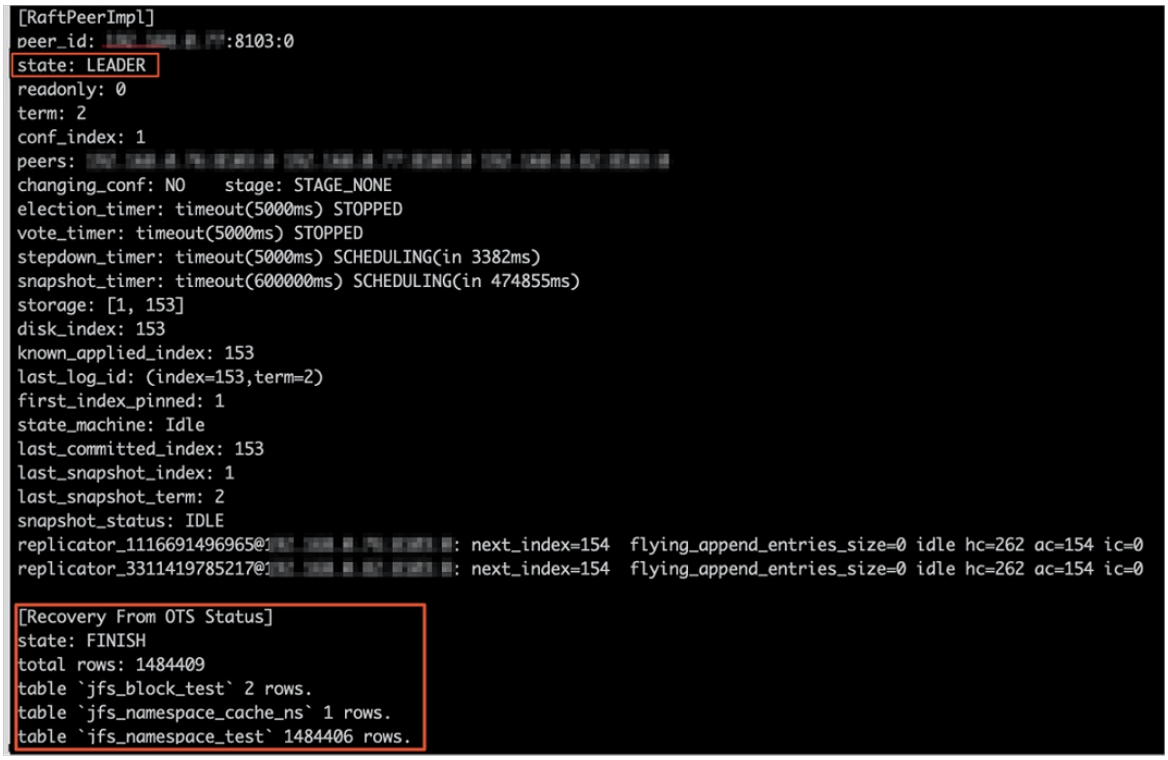
ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.



7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

7.2.4. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see

Create a cluster.

- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.

ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

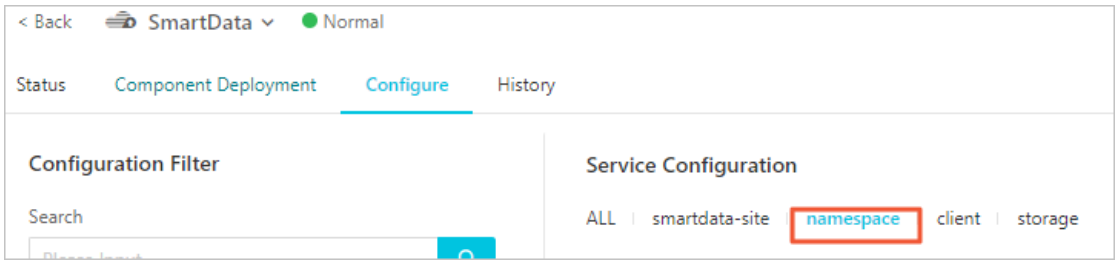
iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **namespace** tab for the SmartData service.

   i. Click the **Configure** tab.

   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:

   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

7.2.5. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

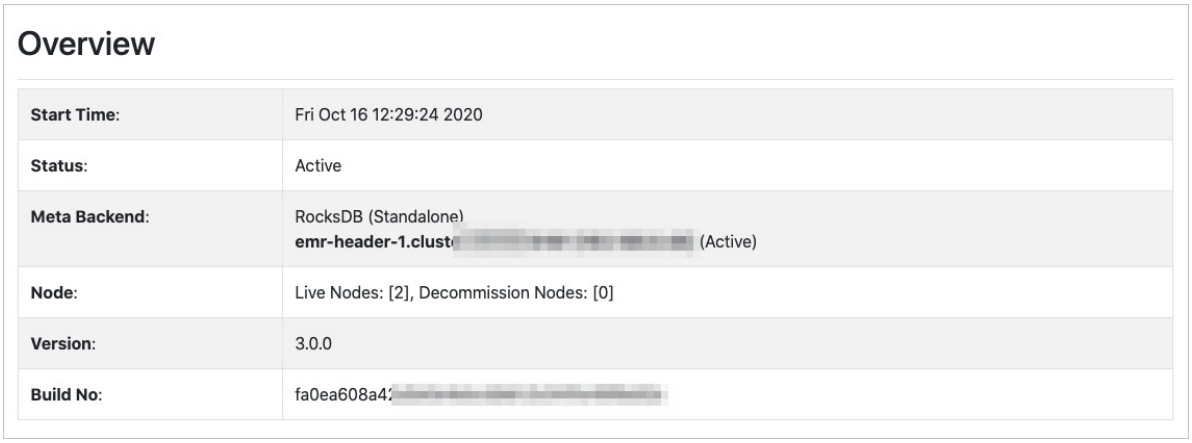
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

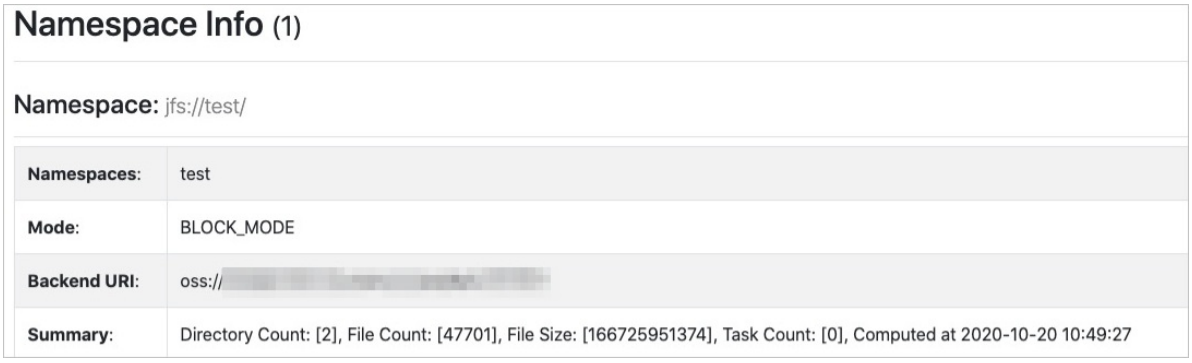


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



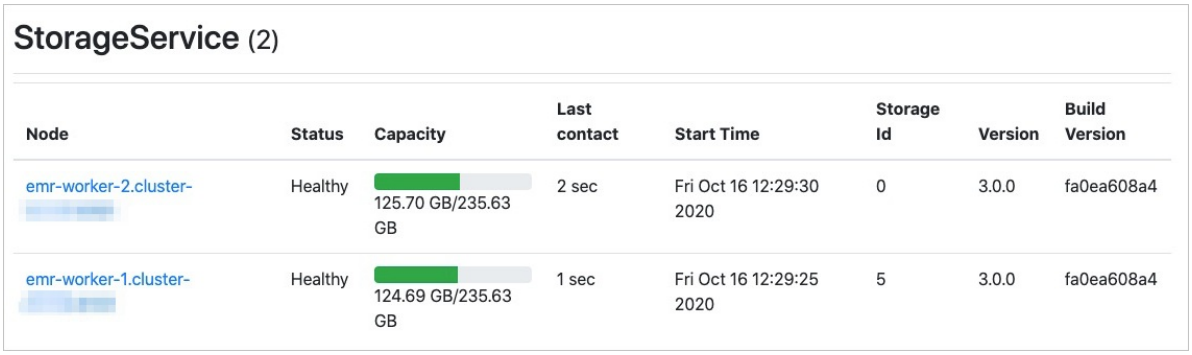
Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

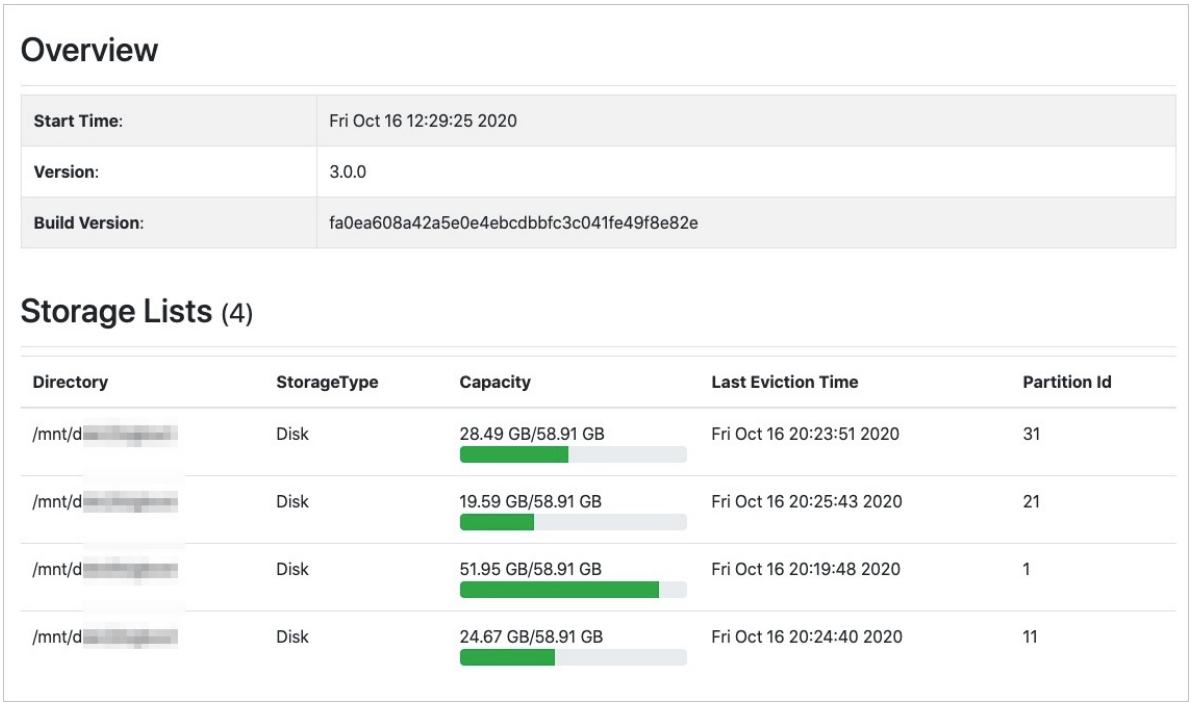
  This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.



StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.



Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

7.2.6. Manage JindoFS permissions

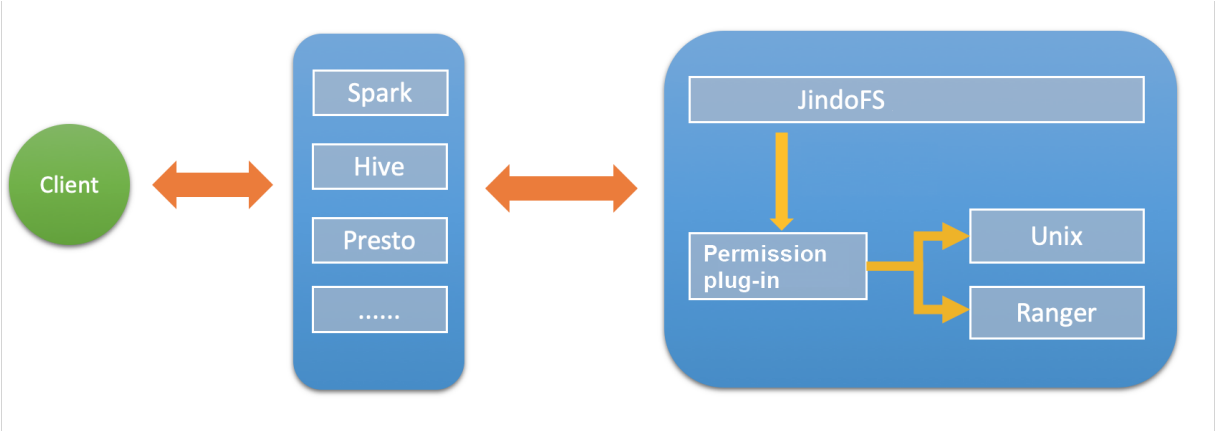
This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

Permission management based on the storage mode:

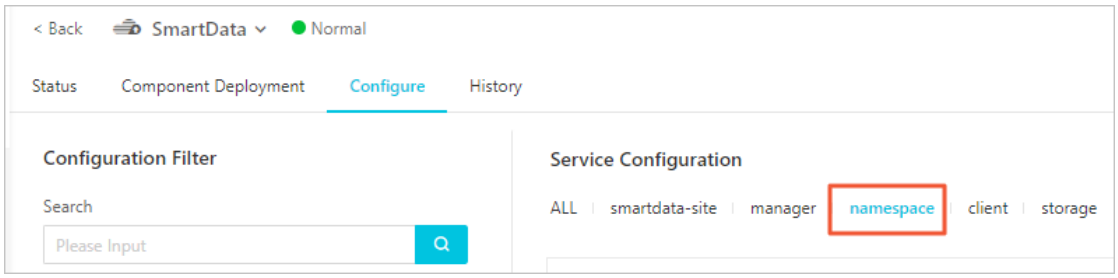
- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.
  - Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

7.2.7. Data management policies

JindoFS in block storage mode provides advanced data management policies to meet storage requirements in different scenarios. The advanced policies include storage policies and compression policies. This topic describes the policies in detail and provides examples to show how to use the policies.

Storage policies

JindoFS provides the five storage policies described in the following table to support flexible storage.

Policy	Description
AR	Data has only one backup. The backup is stored as an Archive object in OSS.
IA	Data has only one backup. The backup is stored as an Infrequent Access (IA) object in OSS.

Policy	Description
COLD	Data has only one backup. The backup is stored as a Standard object in OSS.
WARM	Data has one backup in OSS and one local backup. The local backup can effectively accelerate data read operations. By default, this storage policy is used.
HOT	Data has one backup in OSS and one local backup. The local backup is forcibly locked. This way, it is not deleted when the cache is automatically cleared. This policy can achieve a better acceleration effect than the WARM policy and is suitable for the hottest data.

For more information about the storage classes of OSS, see Overview.

A newly added file is stored based on the storage policy specified for its parent directory. Example:

- Run the following command to configure a storage policy:

```
jindo jfs -setStoragePolicy [-R] <StoragePolicy>(AR/IA/COLD/WARM/HOT) <path> ...
```

  where:

  - `[-R]`: specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.
  - `<path>`: the name of the directory for which the storage policy is configured.

- Run the following command to obtain the storage policy configured for a directory:

```
jindo jfs -getStoragePolicy <path>
```

Compression policies

JindoFS allows you to compress data blocks before you store them. This feature effectively reduces the storage space occupied by data blocks and improves data read and write efficiency. This feature is suitable for files that have a high compression ratio. The following table describes the supported compression policies.

Policy	Description
NONE	Data blocks are not compressed. By default, this compression policy is used.
ZSTD	The Zstandard compression algorithm is used to compress data blocks.

A newly added file is compressed based on the compression policy specified for its parent directory before it is stored. Example:

- Run the following command to configure a compression policy:

```
jindo jfs -setCompressionPolicy [-R] <CompressionPolicy>(NONE/ZSTD) <path> ...
```

where:

- `[-R]`: specifies that a recursive operation is performed to configure the same compression policy for all sub-directories of the directory.
- `<path>`: the name of the directory for which the compression policy is configured.

- Run the following command to obtain the compression policy configured for a directory:

```
jindo jfs -getCompressionPolicy <path> ...
```

7.2.8. Analyze metadata offline

In E-MapReduce (EMR) V3.30.0 or later, JindoFS in block storage mode allows you to dump the metadata of an entire namespace to Object Storage Service (OSS) and use Jindo SQL to analyze the metadata.

Context

In Hadoop Distributed File System (HDFS), the metadata is stored in a snapshot file named fsimage. The snapshot file contains the following information about metadata: namespace, file that stores the metadata, blocks, and file system quota. In HDFS, you can run a command to download the fsimage file in the XML format to your on-premises machine and view this file to analyze metadata offline. JindoFS does not require you to download metadata to your on-premises machine.

Upload HDFS metadata to OSS

Run the following command to upload metadata of a specific namespace to OSS:

```
jindo jfs -dumpMetadata <nsName>
```

`<nsName>` indicates the name of the namespace in block storage mode.

For example, upload the metadata of the test-block namespace to OSS and analyze the metadata offline.

```
jindo jfs -dumpMetadata test-block
```

If the following information appears, data is uploaded to OSS and stored as a file in the JSON format.

```
Sucessfully upload namespace metadata to OSS.
```

Metadata upload path

The metadata upload path is the metadataDump sub-directory under *sysinfo* in JindoFS.

For example, if `namespace.sysinfo.oss.uri` is set to `oss://abc/`, metadata is uploaded to the `oss://abc/metadataDump` directory.

Parameter	Description
namespace.sysinfo.oss.uri	The path of the OSS bucket.
namespace.sysinfo.oss.endpoint	The endpoint. An endpoint specified in a different region can also be used.
namespace.sysinfo.oss.access.key	The AccessKey ID of your Alibaba Cloud account.
namespace.sysinfo.oss.access.secret	The AccessKey secret of your Alibaba Cloud account.

Batch information: Metadata from HDFS changes based on the usage. Each time you want to analyze metadata, you must run commands to view the snapshot file of the metadata. Each time you run a Jindo command to upload metadata, a batch number that is generated based on the upload time is used as the root directory to store the metadata. This ensures that the data you uploaded each time is not overwritten. You can delete historical data based on your requirements.

- 1 indicates the configuration path of system information in OSS.
- 2 indicates a namespace.
- 3 indicates a batch number.

Metadata schema

HDFS metadata that is uploaded to OSS is stored as a file in the JSON format. Schema information:

```
{
  "type":"string",          /*Inode type, FILE DIRECTORY*/
  "id": "string",            /*INode id*/
  "parentId" :"string",         /*Parent node ID*/
  "name":"string",         /*Inode name*/
  "size": "int",         /*Inode size, BIGINT*/
  "permission":"int",         /*Permission, INT*/
  "owner":"string",          /*Owner name*/
  "ownerGroup":"string",     /*Owner group name*/
  "mtime":"int",              /*Inode modification time, BIGINT*/
  "atime":"int",              /*The time when the inode was last accessed, BIGINT*/
  "attributes":"string",       /*File-related attributes*/
  "state":"string",            /*Inode status*/
  "storagePolicy":"string",    /*Storage policy*/
  "etag":"string"           /*etag*/
}
```

Use Jindo SQL to analyze metadata

1. Run the command shown in the following figure to start Jindo SQL.

```
[root@emr-header-1 ~]# jindo sql
Spark master: yarn, Application Id: application_1603081647416_0050
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.33 seconds, Fetched 3 row(s)
```

2. Query the tables whose data can be analyzed by using Jindo SQL.
   - Run the `show tables` command to view the tables whose data can be analyzed. Jindo SQL has two built-in modules audit_log and fs_image for audit and metadata analysis.
   - Run the `show partitions fs_image` command to view the partition information of the fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

     The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

3. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287249    1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       15464684820176650002    1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The namespace and datetime columns are added and indicate the namespace name and the timestamp when metadata is uploaded.

Example: Obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

Use Hive to analyze metadata

1. Create a table schema in Hive.

   You can use the following command to create a table schema of the metadata that is used for queries in Hive:

```
CREATE EXTERNAL TABLE `table_name`
(`type` string,
 `id` string,
 `parentId` string,
 `name` string,
 `size` bigint,
 `permission` int,
 `owner` string,
 `ownerGroup` string,
 `mtime` bigint,
 `atime` bigint,
 `attr` string,
 `state` string,
 `storagePolicy` string,
 `etag` string)
 ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
 STORED AS TEXTFILE
 LOCATION 'The OSS path to which file data is uploaded';
```

2. Use Hive to analyze data offline.

   After you create a Hive table, you can use Hive SQL to analyze metadata.

```
select * from table_name limit 200;
```

   The following figure shows an example.

```
hive> select * from inode_metadata_test0 limit 100;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_202009[redacted]
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1[redacted], Tracking URL = http://emr-heade[redacted]:20888/proxy/applic[redacted]
Kill Command = /usr/lib/hadoop-current/bin/hadoop job  -kill job_1599[redacted]
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-09-08 14:57:26,112 Stage-1 map = 0%,  reduce = 0%
2020-09-08 14:57:31,263 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.22 sec
MapReduce Total cumulative CPU time: 1 seconds 220 msec
Ended Job = job_[redacted]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.22 sec   HDFS Read: 6867 HDFS Write: 1524 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 220 msec
OK
Directory       1127433438684721971 4    1127433438684721971 3    /uttest/oss     0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Directory       1127433438684721971 9    1127433438684721971 3    /uttest/oss2    0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
Directory       1127433438684721971 6    1127433438684721971 4    /uttest/oss/dir 0       511     caojie  staff   1599545017636   1599545017636           Finalized       WARM
File    1127433438684721971 5    1127433438684721971 4    /uttest/oss/file1       0       420     caojie  staff   1599545017632   1599545017632           Finalized       WARM
File    1127433438684721971 7    1127433438684721971 6    /uttest/oss/dir/file2   0       420     caojie  staff   1599545017642   1599545017642           Finalized       WARM
File    1127433438684721971 8    1127433438684721971 6    /uttest/oss/dir/file3   0       420     caojie  staff   1599545017651   1599545017651           Finalized       WARM
Directory       1127433438684721972 0    1127433438684721971 9    /uttest/oss2/dir        0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
File    1127433438684721972 1    1127433438684721972 0    /uttest/oss2/dir/file2  0       420     caojie  staff   1599545017658   1599545017658           Finalized       WARM
File    1127433438684721972 2    1127433438684721972 0    /uttest/oss2/dir/file3  0       420     caojie  staff   1599545017666   1599545017666           Finalized       WARM
Directory       1127433438684721971 3    1767282355743385190 5    /uttest 0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Time taken: 10.734 seconds, Fetched: 10 row(s)
hive>
```

7.2.9. Use a JindoFS credential provider

The data of JindoFS is stored in Object Storage Service (OSS). If you want to access the data of JindoFS, you must provide an AccessKey pair that can be used to access OSS. SmartData V3.4.0 and later support JindoFS credential providers. You can use a JindoFS credential provider to save an encrypted AccessKey pair into a file. This prevents the AccessKey pair from being leaked.

Configure a JindoFS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **smartdata-site** tab.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **smartdata-site** tab.
3. Add configuration information.
   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

Parameter	Description
fs.jfs.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

   iii. Click **OK**.
4. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Credential provider types

- TemporaryAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a validity period and a security token with a validity period are used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This credential provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**.
fs.jfs.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this credential provider, you must configure the parameters described in the following table in environment variables.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**.
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- JindoCommonCredentialsProvider

This is a common credential provider. After you configure the parameters described in the following table, you can use this credential provider in both JindoOSS and JindoFS.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.JindoCommonCredentialsProvider**.
jindo.common.accessKeyId	The AccessKey ID used to access OSS.
jindo.common.accessKeySecret	The AccessKey secret used to access OSS.
jindo.common.securityToken	The temporary security token used to access OSS.

- EcsStsCredentialsProvider

  This credential provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.credentials.provider	Set this parameter to **com.aliyun.emr.fs.auth.EcsStsCredentialsProvider**.

7.2.10. Use JindoFS in block storage mode to store encrypted files

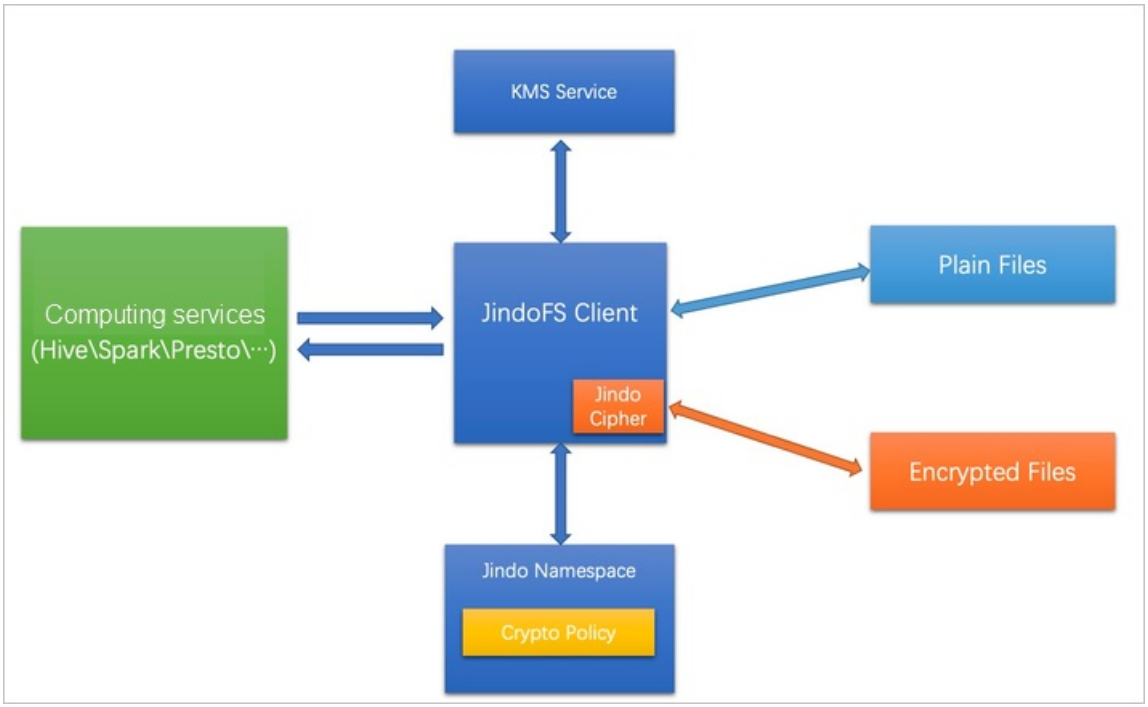
JindoFS in block storage mode supports file encryption. The encryption mechanism and usage method are similar to those of encryption zones for Apache HDFS. Keys are managed by Key Management Service (KMS). You can configure encryption policies for a directory that contains sensitive data. This way, you can transparently encrypt the data that you want to write to the directory and transparently decrypt the data that you want to read from the directory. In this process, you do not need to change your code.

Prerequisites

- An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.
- KMS is activated. For more information, see Activate KMS.

Context

The following figure shows the architecture of JindoFS in block storage mode with file encryption supported.



Configure KMS-related parameters for JindoFS

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **namespace** tab.
3. Add configuration information.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description
crypto.provider.type	The type of the provider. Set this parameter to **ALIYUN**.
crypto.provider.endpoint	The public endpoint of KMS. For more information, see Make API requests.
crypto.provider.kms.accessKeyId	The AccessKey ID used to access KMS.
crypto.provider.kms.accessKeySecret	The AccessKey secret used to access KMS.

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Use JindoFS KeyProvider

JindoFS KeyProvider is used to connect JindoFS to KMS. The keys for JindoFS are stored in KMS. You can use JindoFS KeyProvider to create, query, and rotate keys based on KMS.

- Create a key: You can run the following command to create a key:

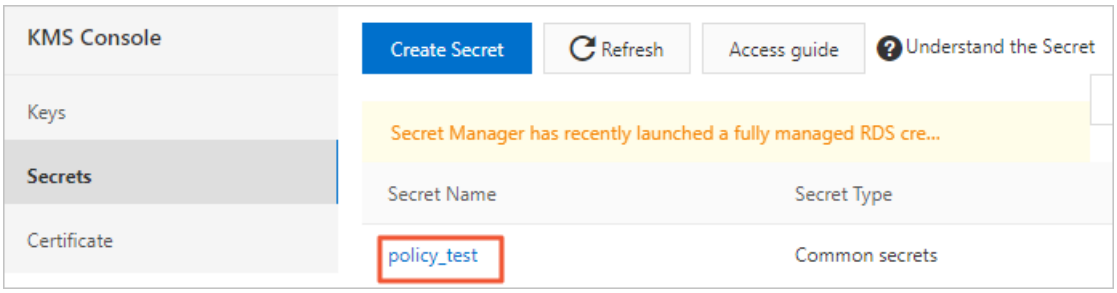
```
jindo key -create -keyIdName <keyIdName>
```

> **Note** In this topic, <keyIdName> in commands indicates the name of the key that you created.

For example, run the following command to create a key named *policy_test*:

```
jindo key -create -keyIdName policy_test
```

After the *policy_test* key is created, it is displayed on the Secrets page of the KMS console.



- Query keys: You can run the following command to query the names of all existing keys for JindoFS:

```
jindo key -list
```

The following information is returned:

```
Listing Keys:
        policy_test
        policy_test2
```

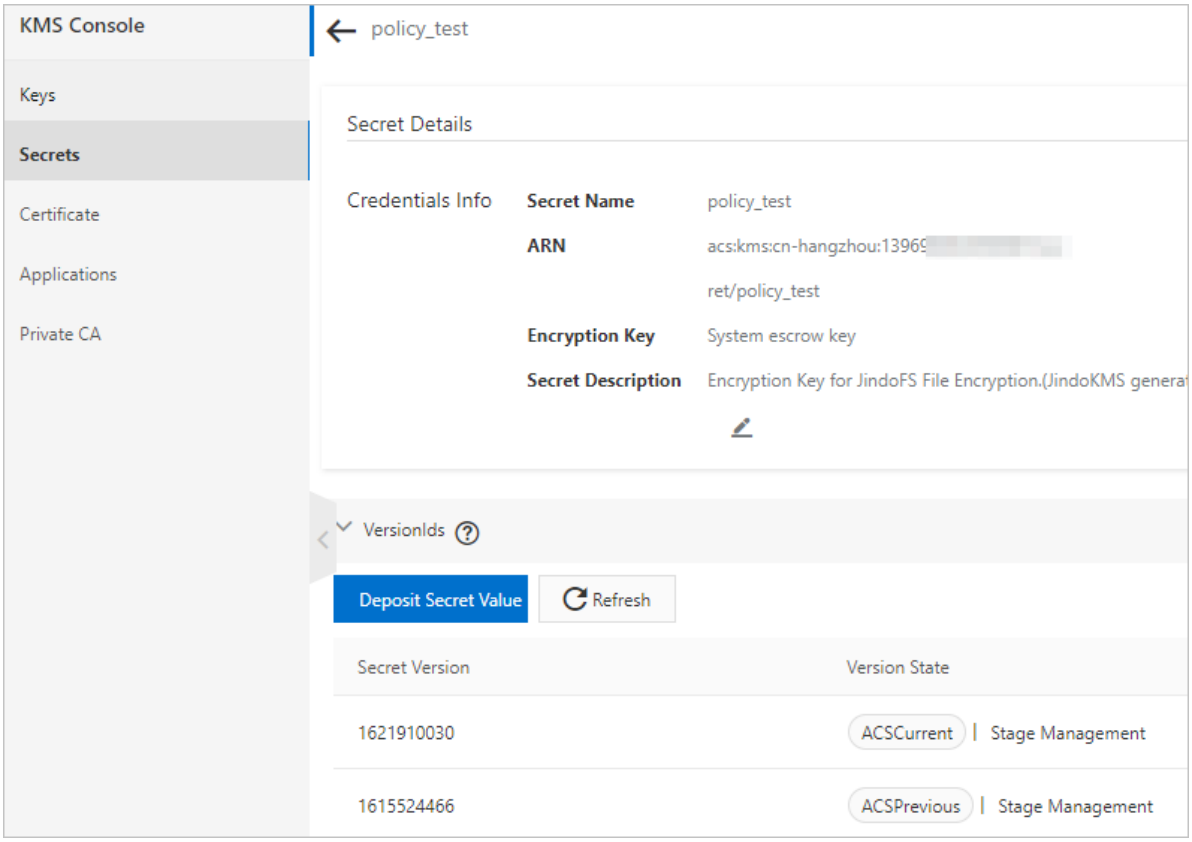
- Rotate a key: You can run the following command to periodically rotate a key based on its ID. After the key is rotated, the key version changes. If a file is encrypted after the rotation, the latest key version is used. If a file is decrypted after the rotation, the key version that corresponds to the file is used.

```
jindo key -roll -keyIdName <keyIdName>
```

For example, run the following command to rotate the *policy_test* key:

```
jindo key -roll -keyIdName policy_test
```

After the *policy_test* key is rotated, the version of the key is updated. The original version is in the **ACSPrevious** state and the new version is in the **ACSCurrent** state.



Manage encryption policies for JindoFS

You can run the following commands to configure and query encryption policies.

- Configure an encryption policy:

```
jindo jfs -setCryptoPolicy -keyIdName <keyIdName> <path>
```

> **Note** `<path>` in the preceding example indicates the path of a file on JindoFS, for example, *jfs://test/*.

- Query an encryption policy:

```
jindo jfs -getCryptoPolicy <path>
```

Example:

1. Query the encryption policy of the *jfs://test/* path:

```
jindo jfs -getCryptoPolicy jfs://test/
```

`{NONE}` is returned.

2. Configure an encryption policy for the *jfs://test/* path:

```
jindo jfs -setCryptoPolicy -keyIdName policy_test jfs://test/
```

3. Go to the *bigboot* directory and query the encryption policy of the *jfs://test/* path again:

```
jindo jfs -getCryptoPolicy jfs://test/
```

The following information is returned:

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/b2jindosdk/3.4.0-hadoop3.1/pack
age/b2jindosdk-3.4.0-hadoop3.1/lib/jindo-distcp-3.4.0.jar!/org/slf4j/impl/StaticLoggerB
inder.class]
SLF4J: Found binding in [jar:file:/opt/apps/ecm/service/hadoop/3.2.1-1.0.1/package/hado
op-3.2.1-1.0.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticL
oggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
21/03/12 13:52:34 WARN: Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable
21/03/12 13:52:35 INFO: Jboot log name is /var/log/bigboot/jboot-20210312-135234-12953.
LOG
21/03/12 13:52:35 INFO: Write buffer size 1048576, logic block size 134217728
21/03/12 13:52:35 INFO: cmd=getFileStatus, src=jfs://test/, dst=null, size=0, parameter
=null, time-in-ms=7, version=3.4.0
21/03/12 13:52:35 INFO: cmd=getCryptoPolicy, src=jfs://test/, dst=null, size=0, paramet
er=, time-in-ms=2, version=3.4.0
The crypto policy of path: jfs://test/ is {cipherSuite: AES_CTR_NOPADDING_256, keyIdNam
e: policy_test2, keyIdVersion: null, edek: , iv: }
21/03/12 13:52:35 INFO: Read total statistics: oss read average <none>, cache read aver
age <none>, read oss percent <none>
```

After you configure the encryption policy, you can read data from or write data to the files under this path.

- Copy a local file to HDFS:

```
hadoop fs -put test.log jfs://test/
```

- Display the content of a file:

```
hadoop fs -cat jfs://test/test.log
```

7.3. JindoFS in cache mode

7.3.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in Object Storage Service (OSS) based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

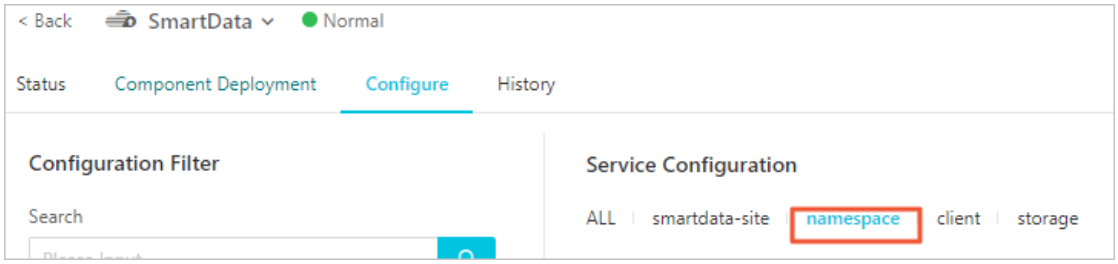
  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. Click **OK**.
5. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
6. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **client** tab.
2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   You do not need to restart the SmartData service to apply the new configurations.
3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

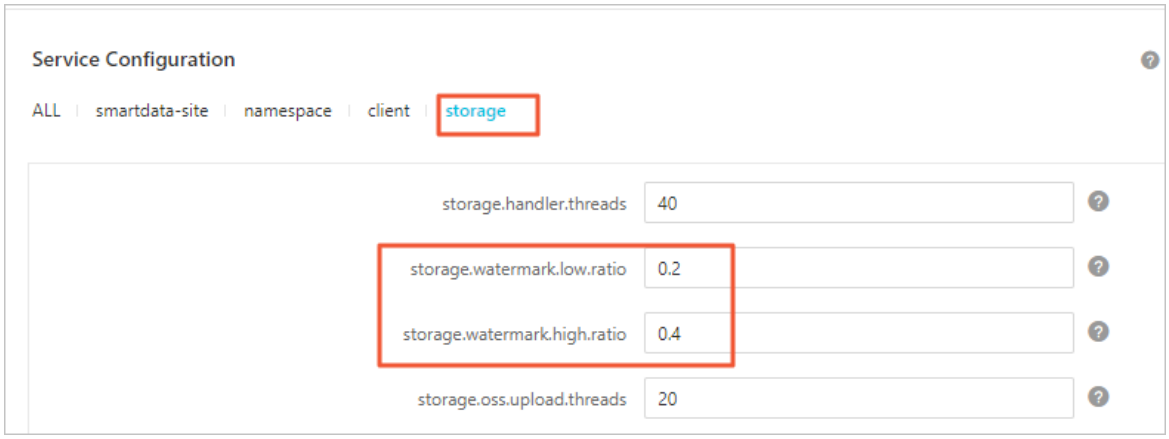
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you specify. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to specify an AccessKey pair. In other cases, you must specify an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you choose for accessing files in OSS:

- OSS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.
  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss.accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.endpoint	The endpoint of the OSS bucket that serves as the storage backend.

> **Note** You can set these parameters to the values of the parameters in an EMR version earlier than EMR V3.30.0.

- JFS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **namespace** tab.
  ii. Set **jfs.namespaces** to **test**.
  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize the cache performance. After you configure the parameters, you do not need to restart the SmartData service to apply the new configurations.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. You can use this parameter to prevent upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If an excessive amount of memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

7.3.2. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

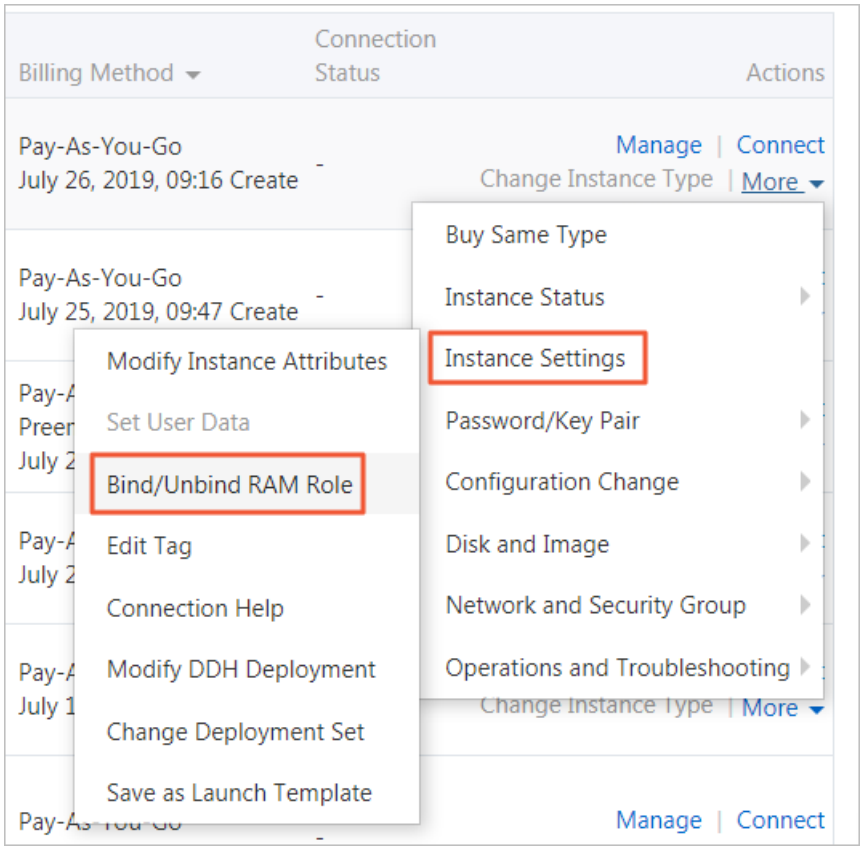
1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.
4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

   ```
   hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
   ```

2. Access OSS by using the FileSystem interface of Hadoop.

   Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

7.3.3. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false

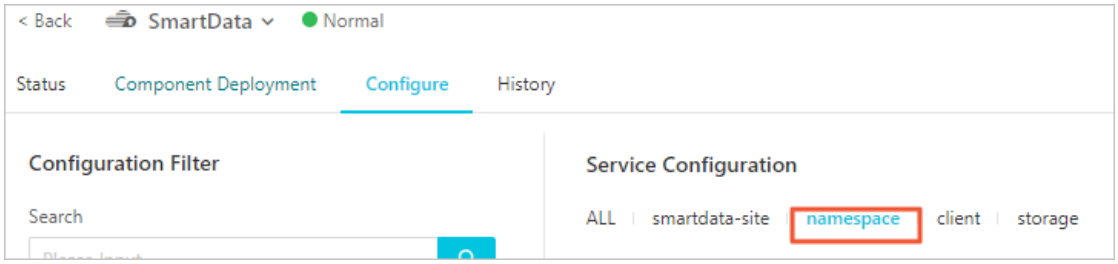
Parameter	Description
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-

right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

7.3.4. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

> **Notice**
>
> - The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
> - After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
> - Hive and Presto jobs do not use Hadoop Job Committer.
> - In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.

i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
ii. Click the **Configure** tab.
iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.
6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.

3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

   The default value is 8.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

7.3.5. Use JindoFS OSS credential providers

You can use a credential provider to save an encrypted AccessKey pair into a file. This prevents the leak of the AccessKey pair.

Context

You can use a Hadoop credential provider to save an encrypted AccessKey pair into a file. This prevents the issue that the AccessKey pair is transmitted in plaintext. You can select an appropriate JindoOSS credential provider based on your business requirements.

Configure a JindoOSS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **smartdata-site** tab.

   i. Click the **Configure** tab.

   ii. In the **Service Configuration** section, click the **smartdata-site** tab.

3. Add configuration information.

   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

   - Global configuration (for all buckets)

Parameter	Description
fs.jfs.cache.oss.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

- Bucket-level configuration

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types. **Note** XXX indicates the name of an Object Storage Service (OSS) bucket.

Credential provider types

You can select different credential providers based on your business requirements. The following providers are supported:

- Global configuration
  - TemporaryAliyunCredentialsProvider

    This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.cache.oss.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

- Bucket-level configuration

- TemporaryAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.
fs.jfs.cache.oss.bucket.XXX.securityToken	The temporary security token used to access the OSS bucket.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access an OSS bucket.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access the OSS bucket.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access the OSS bucket. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

Use a Hadoop credential provider to store AccessKey pair information

> **Note** For more information about Hadoop credential providers, see CredentialProvider API Guide.

Use a command that is provided by Hadoop to store AccessKey pair and security token information into a credential file. Syntax:

```
hadoop credential <subcommand> [options]
```

For example, in global configuration mode, store AccessKey pair and token information into a JCEKS file. You can protect the file by using file permissions or you can specify a password to encrypt the information you want to store. If you do not specify a password, the default string is used for encryption.

```
hadoop credential create fs.jfs.cache.oss.accessKeyId -value AAA -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.accessKeySecret -value BBB -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.securityToken -value  CCC -provider jceks://file/root/oss.jceks
```

After a credential file is generated, you must configure the parameter described in the following table to specify the provider type and location.

Parameter	Description
fs.jfs.cache.oss.security.credential.provider.path	The path used to store the credential file that stores AccessKey pair information. For example, you can set this parameter to *jceks://file/${user.home}/oss.jceks*, which indicates that the *oss.jceks* file is stored in the home directory.

7.3.6. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

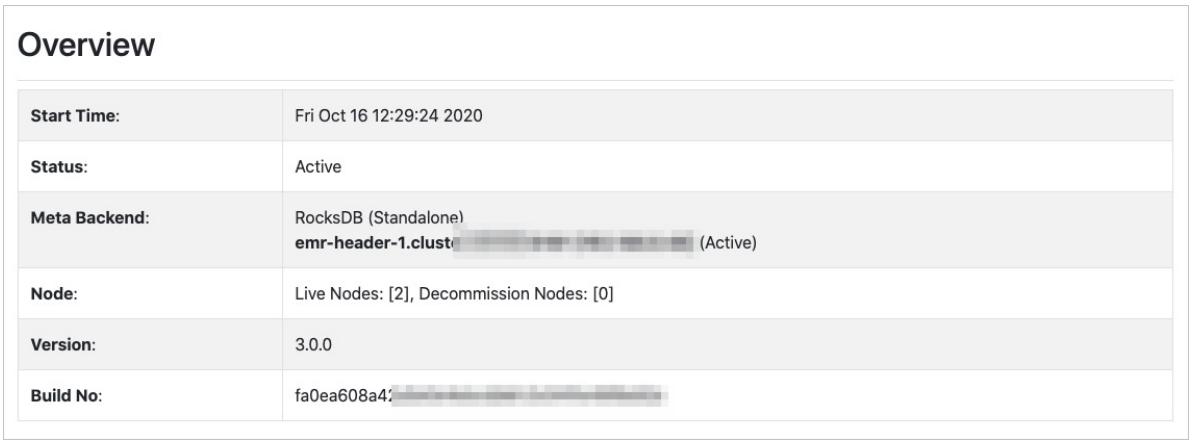
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

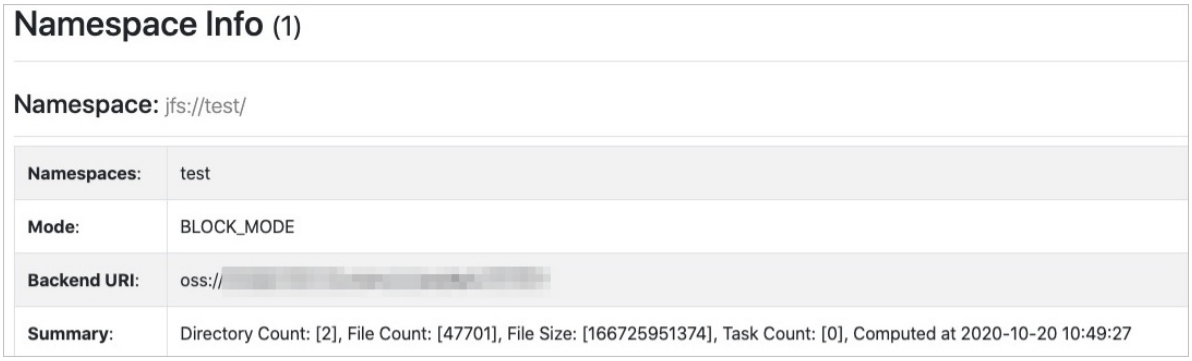
  

  Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.

  Namespace Info (1)

  Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.

StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.

Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

7.3.7. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

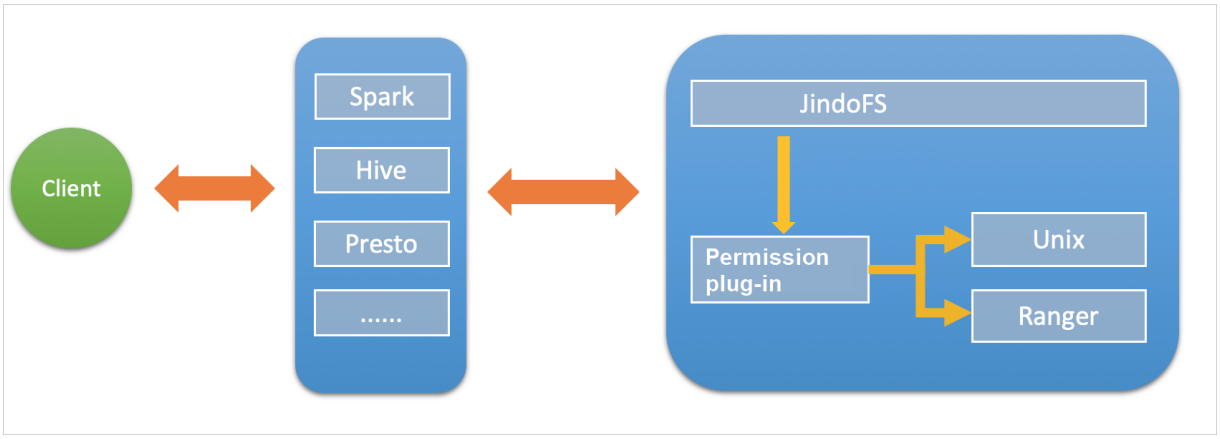
Permission management based on the storage mode:

- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.

- Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.

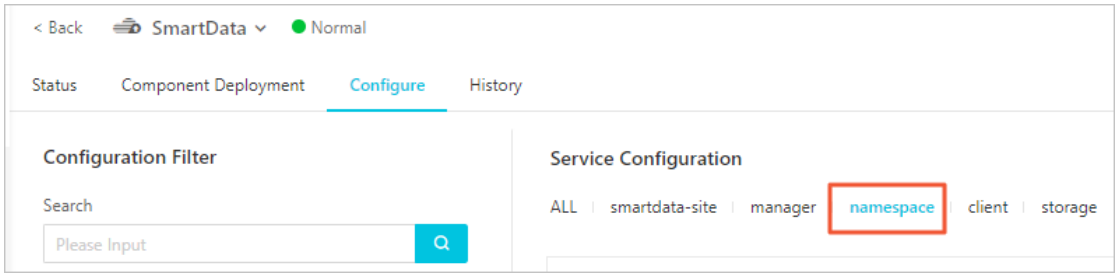
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

   After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

   If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

7.4. JindoTable

7.4.1. Enable query acceleration for ORC files

JindoTable provides a native Optimized Row Columnar (ORC) reader that can be used to accelerate queries of ORC files. By default, query acceleration is disabled. Enabling query acceleration can improve the performance of Spark or Presto to read ORC files.

Prerequisites

ORC files are stored to JindoFS or Object Storage Service (OSS).

> **Note** You cannot accelerate queries for HDFS.

Improve performance of Spark to read data

1. Enable query acceleration.

> **Note** When Spark is used to read ORC files, the DataFrame or Spark-SQL API is required to enable acceleration.

- Configure the global parameter.

  For more information, see Configure the global parameter of Spark.

- Configure job-level parameters.

  You can add Spark startup parameters when you run Spark Shell or Spark SQL jobs.

  ```
  --conf spark.sql.extensions==io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sql.JindoTableExtension
  ```

  For more information about the configuration of jobs, see Configure a Spark Shell job or Configure a Spark SQL job.

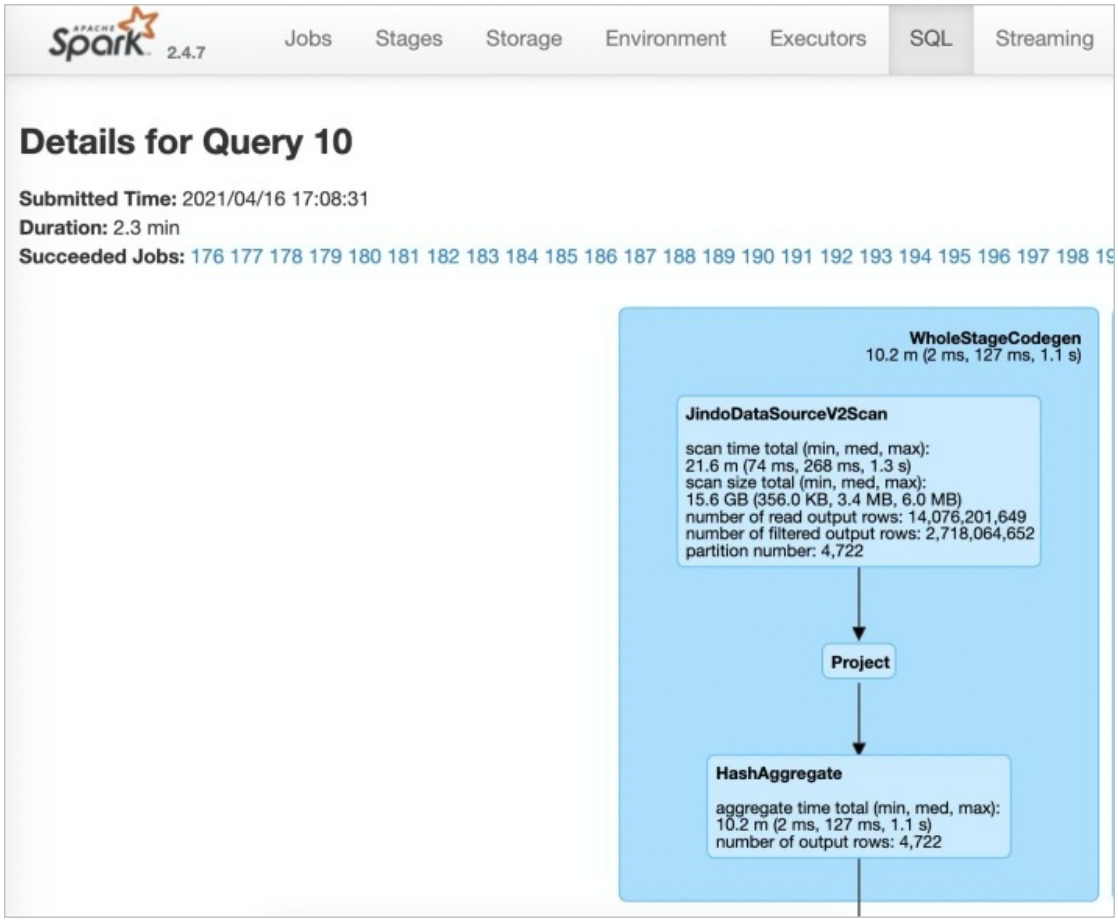
2. Check whether query acceleration is enabled.

   i. Access the web UI of Spark History Server UI.

   For more information, see Access the web UIs of open source components.

   ii. On the SQL tab of Spark, view the execution task.

   If JindoDataSourceV2Scan appears, query acceleration is enabled. Otherwise, check the configurations in Step 1.



Improve performance of Presto to read data

Presto has built-in `catalog: hive-acc`. You can use `catalog: hive-acc` to enable query acceleration.

Example:

```
presto --server https://emr-header-1.cluster-xxx:7778/ --catalog hive-acc --schema default
```

> **Note** `emr-header-1.cluster-xxx` indicates the hostname of the emr-header-1 node.

Configure the global parameter of Spark

1. Go to the Spark service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Spark**.
2. On the Spark service page, click the **Configure** tab.
3. Find the spark.sql.extensions parameter and change its value to io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sql.JindoTableExtension.
4. Save the configurations.
   i. Click **Save** in the upper-right corner of the Service Configuration section.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
5. Restart ThriftServer.
   i. Choose **Actions > Restart ThriftServer** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

7.4.2. Use JindoTable

JindoTable is used to implement tiered storage, optimize table files, and collect data statistics based on the popularity of tables or partitions. This topic describes how to use JindoTable.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your on-premises machine.
- An EMR cluster of the 3.30.0 version or later is created. For more information about how to create a cluster, see Create a cluster.

Use JindoTable

Common commands:

- -accessStat
- -cache

- -archive
- -unarchive
- -uncache
- -status
- -optimize
- -showTable
- -showPartition
- -listTables
- -dumpmc

> **Notice** Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

-accessStat

- Syntax

  jindo table -accessStat {-d} *<days>* **{-n}** <topNums>

- Description

  This command is used to query the access records in which tables or partitions are visited most in a specified time range.

  <days> and <topNums> must be positive integers. If <days> is 1, all the access records generated from 00:00 (local time) on the current day to the current time are queried.

- Example: Query the 20 access records of tables or partitions that are most frequently visited within the last seven days.

```
jindo table -accessStat -d 7 -n 20
```

-cache

- Syntax

  jindo table -cache {-t} <dbName.tableName> **[-p]** <partitionSpec> **[-pin]**

- Description

  This command is used to cache data of specified tables or partitions to local disks.

  The data of the tables or partitions must be stored in Object Storage Service (OSS) or JindoFileSystem (JindoFS). Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`. When you specify `-pin`, if the cache space is insufficient, do not delete related data if possible.

- Example: Cache data of the db1.t1 table that is generated on March 16, 2020 to local disks.

```
jindo table -cache -t db1.t1 -p date=2020-03-16
```

-uncache

- Syntax

jindo table -uncache {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to delete cached data of specified tables or partitions from local disks.

  The data of the tables or partitions must be stored in OSS or JindoFS. Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

- Examples:
  - Delete cached data of the db1.t2 table from local disks.

    ```
    jindo table -uncache -t db1.t2
    ```

  - Delete cached data of the db1.t1 table from local disks.

    ```
    jindo table -uncache -t db1.t1 -p date=2020-03-16,category=1
    ```

-archive

- Syntax

  jindo table -archive {-a|i} {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to lower the level of the policy that is used to store data of specified tables or partitions. By default, the Archive storage class is used.

  To use the Infrequent Access (IA) storage class, add -i to the command. Specify tables in the format of database.table. Specify partitions in the format of 'partitionCol1=1,partitionCol2=2,...'.

- Example: Cache data of the db1.t1 table to local disks.

  ```
  jindo table -archive -t db1.t1 -p date=2020-10-12
  ```

-unarchive

- Syntax

  jindo table -unarchive [-o|-i] {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to change the storage class from Archive to Standard.

  If `-o` is added to the command, an Archived object is temporarily restored. If `-i` is added to the command, an Archived object is changed to an IA object.

- Example:

  ```
  jindo table -unarchive -o -t db1.t1 -p date=2020-03-16,category=1
  ```

  ```
  jindo table -unarchive -i -t db1.t2
  ```

-status

- Syntax

  jindo table -status {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to view the data storage status of specified tables or partitions.

- Examples:
  - View the data storage status of the db1.t2 table.

    ```
    jindo table -status -t db1.t2
    ```

  - View the data storage status of the db1.t1 table on March 16, 2020.

    ```
    jindo table -status -t db1.t1 -p date=2020-03-16
    ```

-optimize

- Syntax

  jindo table -optimize {-t} <dbName.tableName>

- Description

  This command is used to optimize the data organization of tables at the storage layer.

- Example: Optimize the data organization of the db1.t1 table at the storage layer.

  ```
  jindo table -optimize -t db1.t1
  ```

-showTable

- Syntax

  jindo table -showTable {-t} <dbName.tableName>

- Description

  This command is used to display all partitions in a partitioned table or display the data storage of a non-partitioned table.

- Example: Display all partitions in the db1.t1 partitioned table.

  ```
  jindo table -showTable -t db1.t1
  ```

-showPartition

- Syntax

  jindo table -showPartition {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to display the data storage of partitions.

- Example: Display the data storage of all partitions in the db1.t1 partitioned table on October 12, 2020.

  ```
  jindo table -showPartition -t db1.t1 -p date=2020-10-12
  ```

-listTables

- Syntax

  jindo table -listTables [-db] <dbName>

- Description

  This command is used to display all tables in a specified database. If you do not specify `[-db]`, tables in the default database are displayed.

- Examples:
  - Display tables in the default database.

    ```
    jindo table -listTables
    ```

  - Display tables in the db1 database.

    ```
    jindo table -listTables -db db1
    ```

-dumpmc

- Syntax

  jindo table -dumpmc {-i} <accessId> **{-k}** <accessKey> **{-m}** <numMaps> **{-t}** <tunnelUrl> **{-project}** <projectName> **{-table}** <tablename> **{-p}** <partitionSpec> **{-f}** <csv|tfrecord> **{-o}** <outputPath>

Parameter	Description	Required
-i	The AccessKey ID of your Alibaba Cloud account.	Yes
-k	The AccessKey secret of your Alibaba Cloud account.	Yes
-m	The number of map tasks.	Yes
-t	The VPC Tunnel endpoint of MaxCompute.	Yes
-project	The name of the MaxCompute project.	Yes
-table	The name of the MaxCompute table.	Yes
-p	The partition information. Example: `pt=xxx`. Separate multiple partitions with commas (,), such as `pt=xxx,dt=xxx`.	No
-f	The file format. Valid values: ○ tfrecord ○ csv	Yes
-o	The destination path.	Yes

- Description

This command is used to dump MaxCompute tables to an EMR cluster or OSS. The formats CSV and TFRECORD are supported.

- Examples:
  - Dump a MaxCompute table in the TFRECORD format to an EMR cluster.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o /tmp/outputtf1 -f tfrecord
```

  - Dump a MaxCompute table in the CSV format to OSS.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o oss://bucket1/tmp/outputcsv -f csv
```

7.4.3. Use JindoCube

JindoCube is supported in E-MapReduce (EMR) V3.24.0 and later. This topic describes how to install, deploy, and use JindoCube in EMR.

Prerequisites

A table or view is created.

Overview

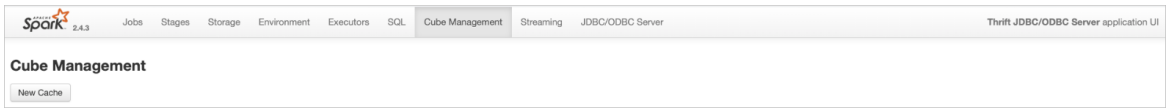
JindoCube is an advanced feature of EMR Spark. It implements pre-calculation to accelerate data processing. It improves query performance by dozens or even hundreds of times. You can persist data of a view to a data source supported by EMR Spark, such as HDFS or Object Storage Service (OSS). EMR Spark automatically discovers available persistent data and optimizes the execution plans for the data. This process is completely visible to users. JindoCube applies to business scenarios where a fixed query mode is used. JindoCube pre-calculates and pre-organizes data to accelerate data queries. For example, you can use JindoCube to implement multidimensional online analytical processing (MOLAP), generate data reports, create data dashboards, and synchronize data across clusters.

Install and deploy JindoCube

JindoCube is an advanced feature of EMR Spark. You can use JindoCube to accelerate all Dataset, DataFrame API, and SQL tasks that are submitted by using EMR Spark. No additional component deployment or maintenance is required.

1. Manage JindoCube on the Spark web UI.

   You can manage JindoCube on the Spark web UI. For example, you can create, delete, or update JindoCube caches. After you create a JindoCube cache, it automatically accelerates all Spark queries in the cluster. You can use the **spark.sql.cache.tab.display** parameter to specify whether to display the Cube Management tab on the Spark web UI. You can specify this parameter on the configuration page of the Spark service in the EMR console or in a Spark task submission command. The default value of this parameter is **false**.



You can also use the **spark.sql.cache.useDatabase** parameter to create a database based on your business requirements and store the view for which you want to create a cache in this database. If partitioned tables are used, you can use the **spark.sql.cache.cacheByPartition** parameter to specify whether to cache data by partition.

Parameter	Description	Example
spark.sql.cache.tab.display	Specifies whether to display the Cube Management tab.	true
spark.sql.cache.useDatabase	The databases where cubes are stored.	db1,db2,dbn
spark.sql.cache.cacheByPartition	Specifies whether to store cubes by partition.	true

2. Optimize Spark queries.

   You can use the **spark.sql.cache.queryRewrite** parameter to specify whether to use a JindoCube cache to accelerate Spark queries. You can specify this parameter at the cluster, session, or SQL statement level. The default value is **true**.

Use JindoCube

1. Create a JindoCube cache.
   i. Log on to the Alibaba Cloud EMR console by using your Alibaba Cloud account.
   ii. Click the **Cluster Management** tab.
   iii. Find the cluster where you want to create a JindoCube cache and click the cluster ID.
   iv. In the left-side navigation pane, click **Connect Strings**.
   v. On the **Public Connect Strings** page, click the link of YARN UI to go to the YARN web UI as a Knox user.

      For more information about Knox, see Knox.
   vi. Click **ApplicationMaster** in the row where **Application Type** is **SPARK** and **Name** is **Thrift JDBC/ODBC Server**.
   vii. On the Spark web UI, click the **Cube Management** tab.
   viii. Click **New Cache**.

      Select a table or view, and click raw cache or cube cache in the action column. You can create two types of caches:

- Raw cache: persists the data of a table or view based on the cache configurations.

  When you create a raw cache, specify the parameters described in the following table.

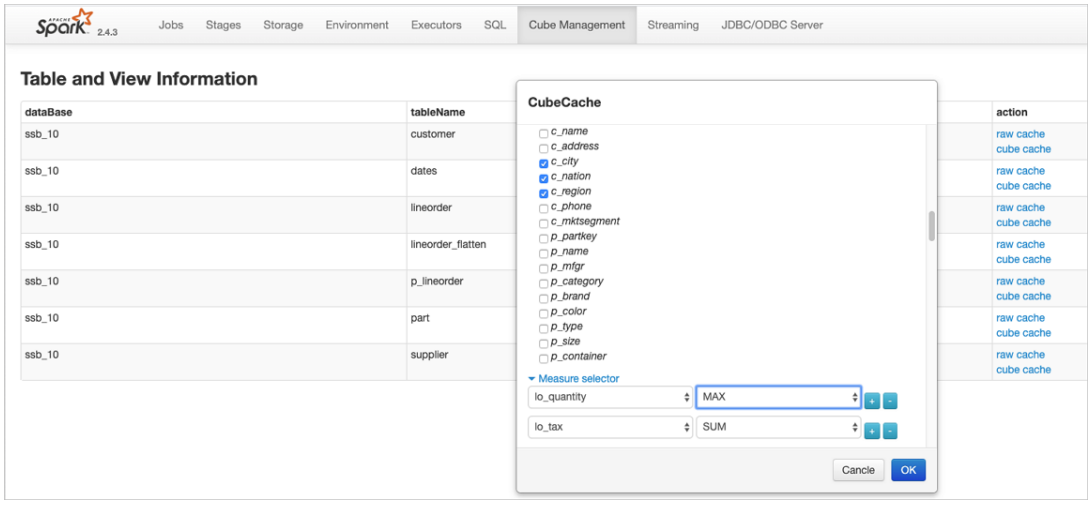
Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Column Selector	The columns from which you want to cache data.	Yes
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No

- Cube cache: builds a cube based on the raw data of a table or view and persists the cube data.

  When you create a cube cache, specify the parameters described in the following table.

Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Dimension Selector	The dimension fields that are used to build a cube.	Yes
Measure Selector	The measure fields and pre-calculation functions that are used to build a cube.	Yes

Parameter	Description	Required
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No



JindoCube builds a cube based on specified dimensions and measures. The following SQL statement shows the configurations of the cube cache in the preceding figure:

```
SELECT c_city, c_nation, c_region, MAX(lo_quantity), SUM(lo_tax)
FROM lineorder_flatten
GROUP BY c_city, c_nation, c_region;
```

JindoCube calculates the finest-grained combination of dimensions. Powered by the onsite computing capability of Spark, JindoCube implements re-aggregation to optimize the queries where a coarse-grained combination of dimensions is used. You must use the pre-calculation functions supported by JindoCube to define a cube cache. The following table lists the supported pre-calculation functions and the aggregate functions that correspond to them.

Aggregate function	Pre-calculation function
COUNT	COUNT
SUM	SUM
MAX	MAX
MIN	MIN
AVG	COUNT and SUM
COUNT (DISTINCT)	PRE_COUNT_DISTINCT
APPROX_COUNT_DISTINCT	PRE_APPROX_COUNT_DISTINCT

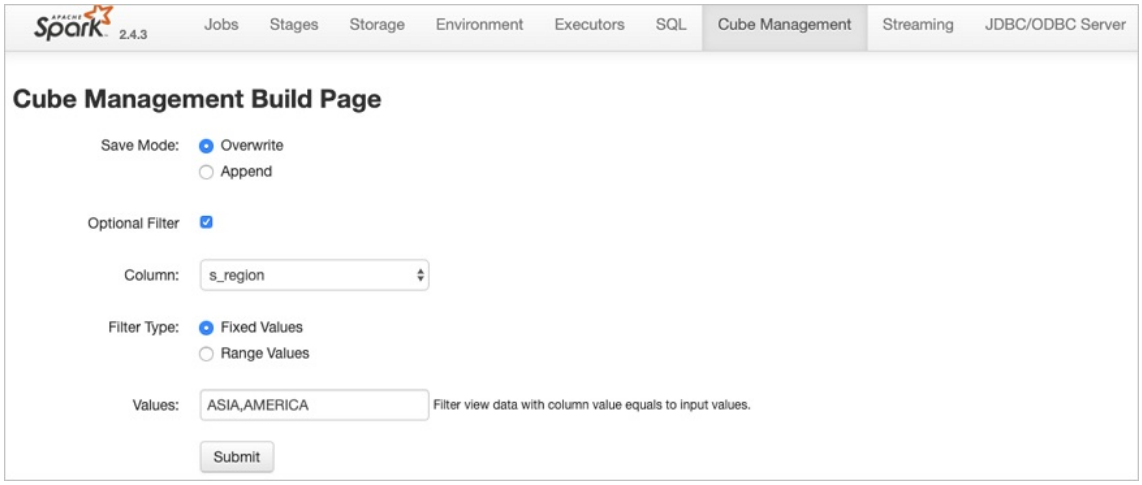
All created caches are displayed on the **Cube Management** tab. You can click **Detail** for a cache to view its details, including its basic information, partition information, building information, and building history.

2. Build a JindoCube cache.

   During the creation of a cache, metadata is prepared, but no data is persisted. Therefore, after you create a cache, you must continue to build the cache. This way, the related data can be persisted to permanent storage, such as HDFS or OSS. In addition, the source table of a cache may be appended with new data or updated. In this case, you must update the data in the cache to ensure data consistency. You can use one of the following methods to build a cache:

   - Manually build a cache

     Click Build Cache to manually build a cache. On the page that appears, configure the parameters, as shown in the following figure.



     The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.

Optional Filter	You can select this parameter to add filter conditions. This way, JindoCube filters data based on the filter conditions before it persists the data.
	■ Column: the field based on which data is filtered.
	■ Filter Type: specifies whether to filter data by fixed values or by value range.
	  ■ Fixed Values: If this option is selected, specify one or more values. Separate values with commas (,).
	  ■ Range Values: If this option is selected, specify a minimum value and a maximum value. You can leave the maximum value empty. In this case, the value range is the minimum value and all values greater than it.

The following SQL statement shows the configurations of the raw cache for the lineorder_flatten view in the preceding figure:

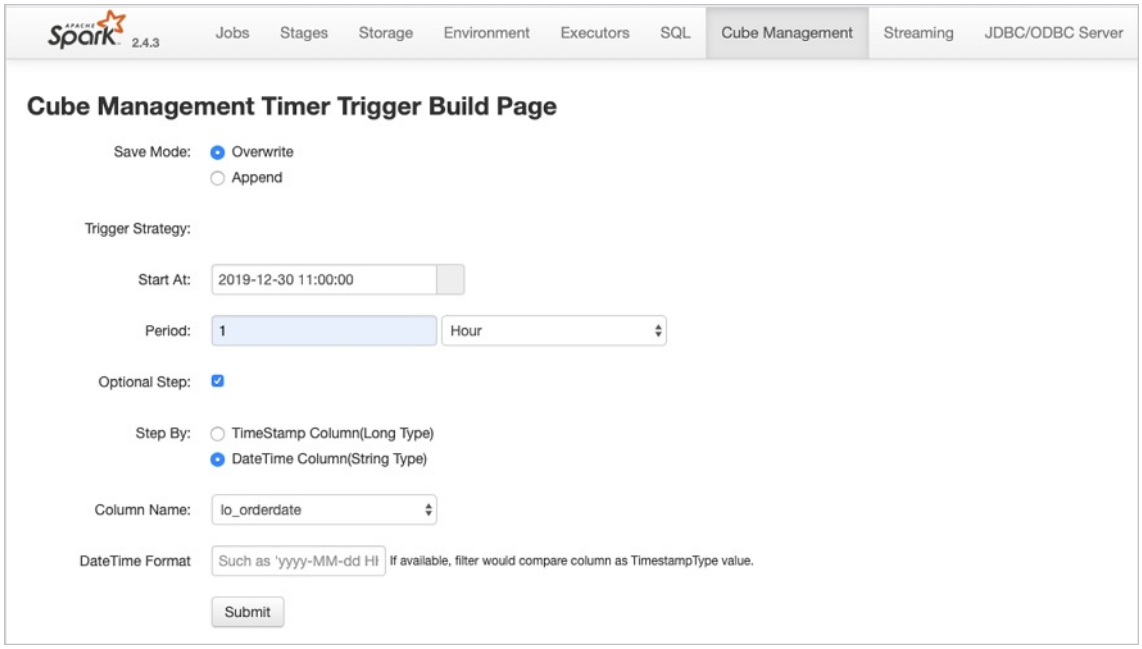
```
SELECT * FROM lineorder_flatten
WHERE s_region == 'ASIA' OR s_region == 'AMERICA';
```

Click **Submit** to submit the cache building task to a Spark cluster. You can view the task status in the Build Information section of the cache details page. After the task is complete, you can view the cache information in the Build History section.

> **Note** Spark writes the cached data to a specified directory in the same way common Spark tasks write data to a table or directory. If multiple users concurrently build a cache whose data format is Parquet, JSON, or ORC, the cached data may be inaccurate and invalid. If concurrent cache building operations or updates are inevitable, we recommend that you use a data format that supports concurrent writes, for example, Delta.

- Configure periodic cache building

  Click Trigger Period Build to configure a periodic cache update policy to ensure data consistency between the cache and source table. On the page that appears, configure the parameters, as shown in the following figure.



The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.
Trigger Strategy	Specifies the start time of the first building task, and the building interval. ■ Start At: Select or enter the time you want to trigger the first building task. Format: yyyy-MM-dd hh:mm:ss. ■ Period: Specify the interval at which cache building tasks are triggered.

Parameter	Description
Optional Step	Specifies the filter conditions for cache building. JindoCube updates the cache in incremental mode based on the filter conditions and interval. If you do not select Optional Step, JindoCube updates the entire cache each time. ■ Step By: Specify the type of the field that you want to update in incremental mode. TimeStamp Column(Long Type) indicates a timestamp field of the LONG type. DateTime Column(String Type) indicates a datetime field of the STRING type. ■ Column Name: Specify the name of the field that you want to update in incremental mode.

On the cache details page, you can view the configured cache update policy. You can click Cancel Period Build to cancel periodic updates. You can view the information of a complete building task in the Build History section.

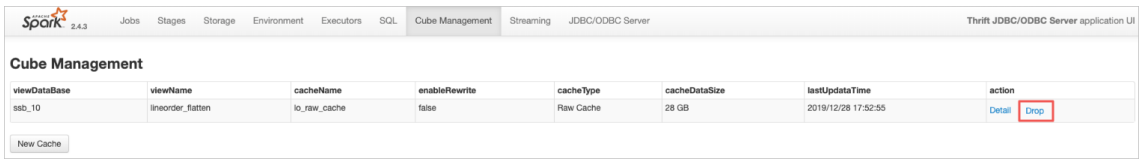
> **Note**
>
> - Periodic building tasks run on a Spark cluster. The related configurations are all saved in SparkContext, and the tasks are periodically triggered by Spark Driver. If you release the Spark cluster, periodic building is disabled.
> - The Build History section displays the following information of each complete task in the current Spark cluster: start time, end time, save mode, building conditions, and final status. After you release the Spark cluster, the information in the Build History section is cleared.

3. Manage a JindoCube cache.

   After you create and build a JindoCube cache, you can manage it on the Cube Management page.

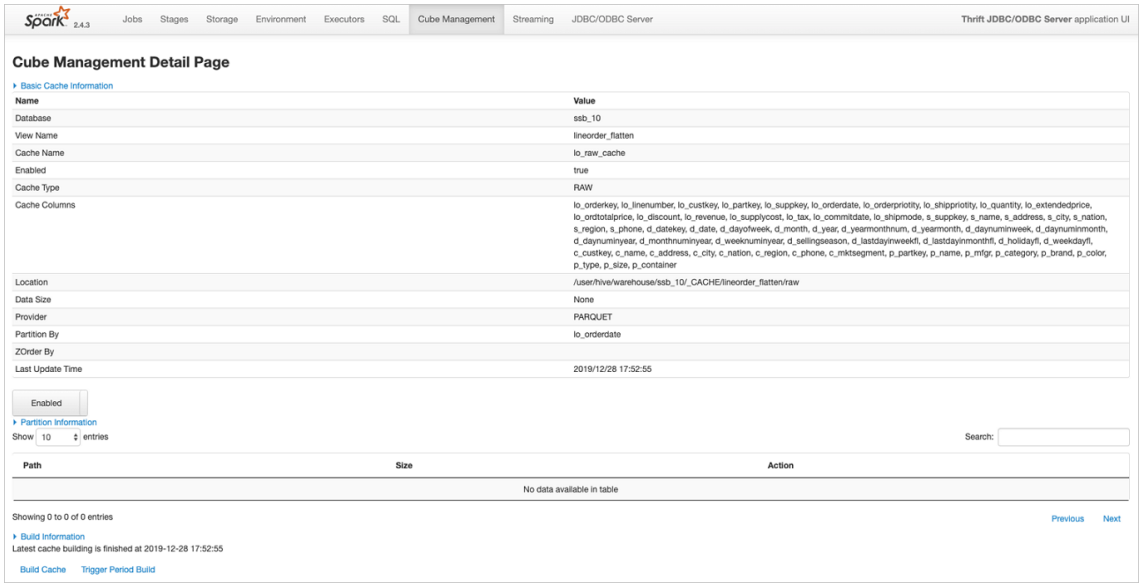
   - Delete a cache

     On the Cube Management page, find the cache you want to delete, and click **Drop** in the action column. All the metadata and data of the cache are deleted.



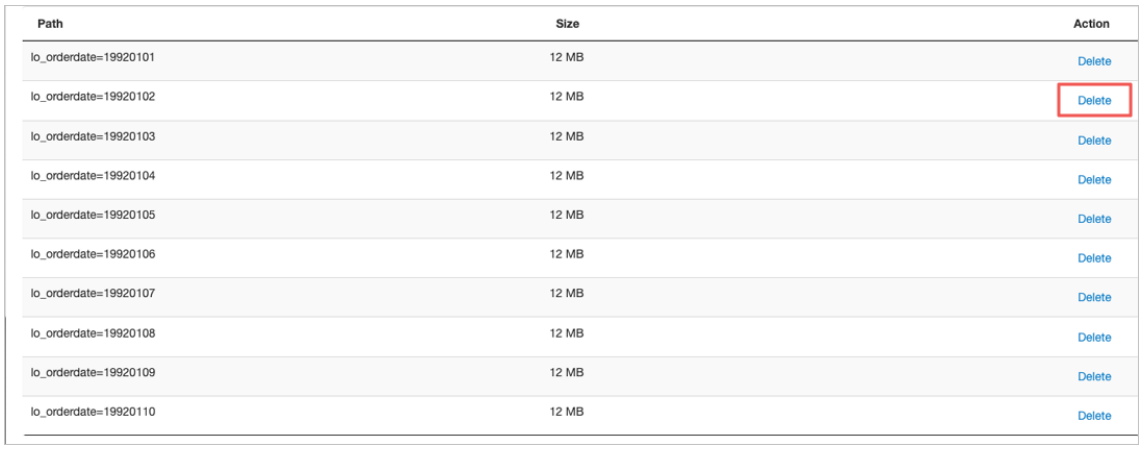
   - Enable and disable optimization of Spark queries

     JindoCube allows you to specify whether to optimize Spark queries at the cache level. On the details page of a cache, click **Enabled** or **Disabled** in the lower part of the Basic Cache Information section to enable or disable the optimization.



- Delete partition data from a cache

  If data in a cache is stored by partition, you can delete the unnecessary partition data from the cache to release more space. On the details page of the cache, find the partition whose data you want to delete, and click **Delete** in the Action column.



Path	Size	Action
lo_orderdate=19920101	12 MB	Delete
lo_orderdate=19920102	12 MB	Delete
lo_orderdate=19920103	12 MB	Delete
lo_orderdate=19920104	12 MB	Delete
lo_orderdate=19920105	12 MB	Delete
lo_orderdate=19920106	12 MB	Delete
lo_orderdate=19920107	12 MB	Delete
lo_orderdate=19920108	12 MB	Delete
lo_orderdate=19920109	12 MB	Delete
lo_orderdate=19920110	12 MB	Delete

> **Note** Before you delete the data of a partition, make sure that the data is no longer needed. If you delete data that is required in subsequent queries, the queries return invalid results.

4. Optimize Spark queries.

   JindoCube optimizes view-based queries. After you have created a raw cache or cube cache for a view, EMR Spark overwrites the execution plan of queries on this view. The new execution plan still uses the original logical semantics but can directly access the data in the cache. This accelerates Spark queries. Use the lineorder_flatten view as an example. It is a wide table view generated by associating the lineorder table with other dimension tables. The following figure shows the information about this view.

```
0: jdbc:hive2://localhost:10001/ssb_10> desc extended lineorder_flatten;
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+-----------------------------+-----------------------------------------------------+----------+
|           col_name          |                      data_type                      | comment  |
+-----------------------------+-----------------------------------------------------+----------+
| lo_orderkey                 | bigint                                              | NULL     |
| lo_linenumber               | bigint                                              | NULL     |
| lo_custkey                  | int                                                 | NULL     |
| lo_partkey                  | int                                                 | NULL     |
| lo_suppkey                  | int                                                 | NULL     |
| lo_orderdate                | int                                                 | NULL     |
| lo_orderpriotity            | string                                              | NULL     |
| lo_shippriotity             | int                                                 | NULL     |
| lo_quantity                 | bigint                                              | NULL     |
| lo_extendedprice            | bigint                                              | NULL     |
| lo_ordtotalprice            | bigint                                              | NULL     |
| lo_discount                 | bigint                                              | NULL     |
| lo_revenue                  | bigint                                              | NULL     |
| lo_supplycost               | bigint                                              | NULL     |
| lo_tax                      | bigint                                              | NULL     |
| lo_commitdate               | int                                                 | NULL     |
| lo_shipmode                 | string                                              | NULL     |
| s_suppkey                   | int                                                 | NULL     |
| s_name                      | string                                              | NULL     |
| s_address                   | string                                              | NULL     |
| s_city                      | string                                              | NULL     |
| s_nation                    | string                                              | NULL     |
| s_region                    | string                                              | NULL     |
| s_phone                     | string                                              | NULL     |
| d_datekey                   | int                                                 | NULL     |
| d_date                      | string                                              | NULL     |
| d_dayofweek                 | string                                              | NULL     |
| d_month                     | string                                              | NULL     |
| d_year                      | int                                                 | NULL     |
| d_yearmonthnum              | int                                                 | NULL     |
| d_yearmonth                 | string                                              | NULL     |
| d_daynuminweek              | int                                                 | NULL     |
| d_daynuminmonth             | int                                                 | NULL     |
| d_daynuminyear              | int                                                 | NULL     |
| d_monthnuminyear            | int                                                 | NULL     |
| d_weeknuminyear             | int                                                 | NULL     |
| d_sellingseason             | string                                              | NULL     |
| d_lastdayinweekfl           | int                                                 | NULL     |
| d_lastdayinmonthfl          | int                                                 | NULL     |
| d_holidayfl                 | int                                                 | NULL     |
| d_weekdayfl                 | int                                                 | NULL     |
| c_custkey                   | int                                                 | NULL     |
| c_name                      | string                                              | NULL     |
| c_address                   | string                                              | NULL     |
| c_city                      | string                                              | NULL     |
| c_nation                    | string                                              | NULL     |
| c_region                    | string                                              | NULL     |
| c_phone                     | string                                              | NULL     |
| c_mktsegment                | string                                              | NULL     |
| p_partkey                   | int                                                 | NULL     |
| p_name                      | string                                              | NULL     |
| p_mfgr                      | string                                              | NULL     |
| p_category                  | string                                              | NULL     |
| p_brand                     | string                                              | NULL     |
| p_color                     | string                                              | NULL     |
| p_type                      | string                                              | NULL     |
| p_size                      | int                                                 | NULL     |
| p_container                 | string                                              | NULL     |
|                             |                                                     |          |
| # Detailed Table Information |                                                     |          |
| Database                    | ssb_10                                              |          |
| Table                       | lineorder_flatten                                   |          |
| Owner                       | hadoop                                              |          |
| Created Time                | Sat Dec 28 17:30:02 CST 2019                        |          |
| Last Access                 | Thu Jan 01 08:00:00 CST 1970                        |          |
| Created By                  | Spark 2.2 or prior                                  |          |
| Type                        | VIEW                                                |          |
| View Text                   | SELECT `lineorder`.`lo_orderkey`, `lineorder`.`lo_linenumber`, `lineorder`.`lo_custkey`, `lineorder`.`lo_partkey`, `
lineorder`.`lo_suppkey`, `lineorder`.`lo_orderdate`, `lineorder`.`lo_orderpriotity`, `lineorder`.`lo_shippriotity`, `lineorder`.`lo_quantity`, `lineor
der`.`lo_extendedprice`, `lineorder`.`lo_ordtotalprice`, `lineorder`.`lo_discount`, `lineorder`.`lo_revenue`, `lineorder`.`lo_supplycost`, `lineorder`
.`lo_tax`, `lineorder`.`lo_commitdate`, `lineorder`.`lo_shipmode`, `supplier`.`s_suppkey`, `supplier`.`s_name`, `supplier`.`s_address`, `supplier`.`s_
city`, `supplier`.`s_nation`, `supplier`.`s_region`, `supplier`.`s_phone`, `dates`.`d_datekey`, `dates`.`d_date`, `dates`.`d_dayofweek`, `dates`.`d_mo
nth`, `dates`.`d_year`, `dates`.`d_yearmonthnum`, `dates`.`d_yearmonth`, `dates`.`d_daynuminweek`, `dates`.`d_daynuminmonth`, `dates`.`d_daynuminyear`
, `dates`.`d_monthnuminyear`, `dates`.`d_weeknuminyear`, `dates`.`d_sellingseason`, `dates`.`d_lastdayinweekfl`, `dates`.`d_lastdayinmonthfl`, `dates`
.`d_holidayfl`, `dates`.`d_weekdayfl`, `customer`.`c_custkey`, `customer`.`c_name`, `customer`.`c_address`, `customer`.`c_city`, `customer`.`c_nation`
, `customer`.`c_region`, `customer`.`c_phone`, `customer`.`c_mktsegment`, `part`.`p_partkey`, `part`.`p_name`, `part`.`p_mfgr`, `part`.`p_category`, `
part`.`p_brand`, `part`.`p_color`, `part`.`p_type`, `part`.`p_size`, `part`.`p_container` FROM `ssb_10`.`lineorder`, `ssb_10`.`supplier`, `ssb_10`.`da
tes`, `ssb_10`.`customer`, `ssb_10`.`part`
WHERE `lineorder`.`lo_orderdate` = `dates`.`d_datekey` AND `lineorder`.`lo_custkey` = `customer`.`c_custkey` AND `lineorder`.`lo_suppkey` = `supplier`
.`s_suppkey` AND `lineorder`.`lo_partkey` = `part`.`p_partkey` |                                                     |          |
| Table Properties            | [transient_lastDdlTime=1577677599]                  |          |
| Serde Library               | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe  |          |
| InputFormat                 | org.apache.hadoop.mapred.TextInputFormat            |          |
| OutputFormat                | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat |          |
| Storage Properties          | [serialization.format=1]                            |          |
+-----------------------------+-----------------------------------------------------+----------+
73 rows selected (0.057 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

The following figure shows the execution plan of simple queries on the lineorder_flatten view.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(11) SortMergeJoin [lo_partkey#313], [p_partkey#359], Inner
   :- *(8) Sort [lo_partkey#313 ASC NULLS FIRST], false, 0
   :  +- Exchange hashpartitioning(lo_partkey#313, 200)
   :     +- *(7) SortMergeJoin [lo_custkey#312], [c_custkey#351], Inner
   :        :- *(4) Sort [lo_custkey#312 ASC NULLS FIRST], false, 0
   :        :  +- Exchange hashpartitioning(lo_custkey#312, 200)
   :        :     +- *(3) BroadcastHashJoin [lo_orderdate#315], [d_datekey#334], Inner, BuildRight
   :        :        :- *(3) BroadcastHashJoin [lo_suppkey#314], [s_suppkey#327], Inner, BuildRight
   :        :        :  :- *(3) Filter (((isnotnull(lo_suppkey#314) && isnotnull(lo_orderdate#315)) && isnotnull(lo_custkey#312)) && isnotnull(lo_part
key#313))
   :        :        :  :  +- Scan hive ssb_10.lineorder [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_ord
erdate#315, lo_orderpriotity#316, lo_shippriotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#32
2L, lo_supplycost#323L, lo_tax#324L, lo_commitdate#325, lo_shipmode#326], HiveTableRelation `ssb_10`.`lineorder`, org.apache.hadoop.hive.serde2.lazy.L
azySimpleSerDe, [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_orderdate#315, lo_orderpriotity#316, lo_ship
priotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#322L, lo_supplycost#323L, lo_tax#324L, lo_c
ommitdate#325, lo_shipmode#326]
   :        :        :  +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :        :     +- *(1) Filter isnotnull(s_suppkey#327)
   :        :        :        +- Scan hive ssb_10.supplier [s_suppkey#327, s_name#328, s_address#329, s_city#330, s_nation#331, s_region#332, s_phone#
333], HiveTableRelation `ssb_10`.`supplier`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [s_suppkey#327, s_name#328, s_address#329, s_city#330
, s_nation#331, s_region#332, s_phone#333]
   :        :        +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :           +- *(2) Filter isnotnull(d_datekey#334)
   :        :              +- Scan hive ssb_10.dates [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearm
onth#340, d_daynuminweek#341, d_daynuminmonth#342, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweek
fl#347, d_lastdayinmonthfl#348, d_holidayfl#349, d_weekdayfl#350], HiveTableRelation `ssb_10`.`dates`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSe
rDe, [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearmonth#340, d_daynuminweek#341, d_daynuminmonth#34
2, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweekfl#347, d_lastdayinmonthfl#348, d_holidayfl#349,
 d_weekdayfl#350]
   :        +- *(6) Sort [c_custkey#351 ASC NULLS FIRST], false, 0
   :           +- Exchange hashpartitioning(c_custkey#351, 200)
   :              +- *(5) Filter ((isnotnull(c_region#356) && (c_region#356 = ASIA)) && isnotnull(c_custkey#351))
   :                 +- Scan hive ssb_10.customer [c_custkey#351, c_name#352, c_address#353, c_city#354, c_nation#355, c_region#356, c_phone#357, c_mk
tsegment#358], HiveTableRelation `ssb_10`.`customer`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [c_custkey#351, c_name#352, c_address#353, c
_city#354, c_nation#355, c_region#356, c_phone#357, c_mktsegment#358]
   +- *(10) Sort [p_partkey#359 ASC NULLS FIRST], false, 0
      +- Exchange hashpartitioning(p_partkey#359, 200)
         +- *(9) Filter isnotnull(p_partkey#359)
            +- Scan hive ssb_10.part [p_partkey#359, p_name#360, p_mfgr#361, p_category#362, p_brand#363, p_color#364, p_type#365, p_size#366, p_conta
iner#367], HiveTableRelation `ssb_10`.`part`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [p_partkey#359, p_name#360, p_mfgr#361, p_category#3
62, p_brand#363, p_color#364, p_type#365, p_size#366, p_container#367] |
+----------------------------------------------------+
1 row selected (0.435 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

After a raw cache is created and built for the lineorder_flatten view, the same queries are performed on the view. EMR Spark automatically uses the cache to optimize the original execution plan. The following figure shows the new execution plan.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:17:47 INFO [main] HiveConf: Found configuration file file:/etc/ecm/spark-conf-2.4.3-hadoop2.8-1.4.2/hive-site.xml
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(1) Project [lo_orderkey#128L, lo_linenumber#129L, lo_custkey#130, lo_partkey#131, lo_suppkey#132, lo_orderdate#133, lo_orderpriotity#134, lo_ship
priotity#135, lo_quantity#136L, lo_extendedprice#137L, lo_ordtotalprice#138L, lo_discount#139L, lo_revenue#140L, lo_supplycost#141L, lo_tax#142L, lo_c
ommitdate#143, lo_shipmode#144, s_suppkey#145, s_name#146, s_address#147, s_city#148, s_nation#149, s_region#150, s_phone#151, ... 34 more fields]
   +- *(1) Filter (isnotnull(c_region#174) && (c_region#174 = ASIA))
      +- *(1) FileScan parquet [lo_orderkey#128L,lo_linenumber#129L,lo_custkey#130,lo_partkey#131,lo_suppkey#132,lo_orderpriotity#134,lo_shippriotity#
135,lo_quantity#136L,lo_extendedprice#137L,lo_ordtotalprice#138L,lo_discount#139L,lo_revenue#140L,lo_supplycost#141L,lo_tax#142L,lo_commitdate#143,lo_
shipmode#144,s_suppkey#145,s_name#146,s_address#147,s_city#148,s_nation#149,s_region#150,s_phone#151,d_datekey#152,... 34 more fields] Batched: true,
Format: Parquet, Location: InMemoryFileIndex[/user/hive/warehouse/ssb_10/_CACHE/lineorder_flatten/raw], PartitionCount: 2406, PartitionFilters: [], Pu
shedFilters: [IsNotNull(c_region), EqualTo(c_region,ASIA)], ReadSchema: struct<lo_orderkey:bigint,lo_linenumber:bigint,lo_custkey:int,lo_partkey:int,l
o_suppkey:int,lo_or... |
+----------------------------------------------------+
```

The new execution plan no longer uses the calculation logic of the lineorder_flatten view. Instead, it directly accesses the cached data in HDFS.

Usage notes

1. JindoCube does not ensure data consistency between the cache and source table. You must manually trigger data updates or configure a periodic update policy based on your requirements for data consistency.
2. JindoCube determines whether to use a cache to optimize execution plans based on the metadata of views. If cached data is incomplete when JindoCube optimizes an execution plan, the query results may be invalid or incomplete. The following operations or configurations may cause incomplete cached data: 1. Delete partition data that is required in subsequent queries on the cache details page. 2. Data required in subsequent queries is filtered out based on the filter conditions specified when you build or update the cache. 3. Required data has not been updated to the cache when the query is performed.

7.4.4. Use JindoTable to collect access frequency statistics on tables and partitions

You can use JindoTable to collect access frequency statistics on tables and partitions, and separate cold and hot data based on these statistics. This helps you reduce storage costs and improve cache usage efficiency.

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

JindoTable can be used to collect access frequency statistics on Hive tables. The collected data is saved in the namespaces of the SmartData service of a cluster.

In SmartData 3.2.X and later, JindoTable can be used to collect access frequency statistics on Spark, Hive, and Presto tables. This feature is enabled for Spark and Presto by default. If you want to disable the feature for Spark or Presto, perform the operations described in Disable the collection of access frequency statistics. The feature is disabled for Hive by default. If you want to enable the feature for Hive, perform the operations described in Enable the collection of access frequency statistics for Hive.

Query access frequency statistics

You can run a command to query access frequency statistics.

- Syntax

  ```
  jindo table -accessStat <-d [days]> <-n [topNums]>
  ```

  Set `days` and `topNums` to positive integers. For example, if days is set to 1 and topNums is not specified, the access frequency statistics of all the tables or partitions that are visited on the current day (from 00:00 to the current time) are queried.

- Description

  This command is used to query the first N tables or partitions that are visited the most frequently within a specific period of time, and the number of times each of them is visited.

- For example, you can run the following command to query the first 20 tables or partitions that have been visited the most frequently within the last seven days, and the number of times each of them has been visited:

  ```
  jindo table -accessStat -d 7 -n 20
  ```

For more information about how to use JindoTable, see Use JindoTable.

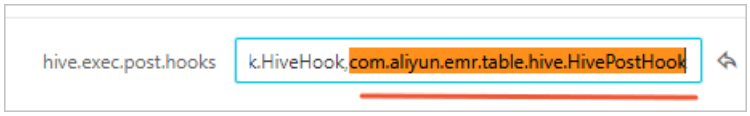
Enable the collection of access frequency statistics for Hive

1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.

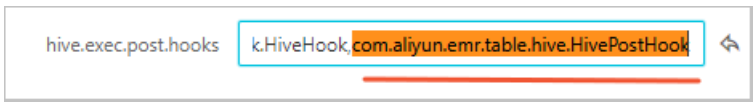
5. Modify the setting of a Hive parameter.
   i. In the left-side navigation pane, choose **Cluster Service > Hive**.
   ii. On the Hive service page, click the **Configure** tab.
   iii. Search for the hive.exec.post.hooks parameter and add **com.aliyun.emr.table.hive.HivePostHook** to the end of the existing parameter value. Separate the hooks with a comma (,).
6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   i. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
   ii. In the **Cluster Activities** dialog box, specify **Description**.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Disable the collection of access frequency statistics

1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of the related parameter.
   - Hive:
     a. In the left-side navigation pane, choose **Cluster Service > Hive**.
     b. On the Hive service page, click the **Configure** tab.
     c. Search for the hive.exec.post.hooks parameter and delete **com.aliyun.emr.table.hive.HivePostHook** from the parameter value.



   - Spark:
     a. In the left-side navigation pane, choose **Cluster Service > Spark**.
     b. On the Spark service page, click the **Configure** tab.
     c. Search for the spark.sql.queryExecutionListeners parameter and delete **com.aliyun.emr.table.spark.SparkSQLQueryListener** from the parameter value.



   - Presto:

a. In the left-side navigation pane, choose **Cluster Service > Presto**.
b. On the Presto service page, click the **Configure** tab.
c. Search for the event-listener.name parameter and delete the parameter value.

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   - Hive:
     a. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
     b. In the **Cluster Activities** dialog box, specify **Description**.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.
   - Spark:
     a. In the upper-right corner of the Spark service page, choose **Actions > Restart ThriftServer**.
     b. In the **Cluster Activities** dialog box, specify **Description**.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.
   - Presto:
     a. In the upper-right corner of the Presto service page, choose **Actions > Restart All Components**.
     b. In the **Cluster Activities** dialog box, specify **Description**.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.

7.5. Toolset

7.5.1. Use JindoFS FUSE

This topic describes how to access JindoFileSystem (JindoFS) from a Filesystem in Userspace (FUSE) client. FUSE allows you to use JindoFS in block storage mode and cache mode (only JFS Scheme supported).

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

FUSE is a method provided by the Linux kernel to mount a file system. You can use a FUSE client to map files in a JindoFS cluster to a local disk. Then, you can directly access data in the JindoFS cluster without the need to use the `hadoop fs -ls jfs://<namespace>/` command.

Mount a directory

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs
   ```

3. Run the following command to mount the directory:

   ```
   jindofs-fuse /mnt/jfs
   ```

   In this example, the */mnt/jfs/* directory is mounted.

Read and write files

1. Run the following command to query all the sub-directories under the /mnt/jfs/ directory:

   ```
   ls /mnt/jfs/
   ```

   All the namespaces that you have configured on the server are returned.

   ```
   test  testcache
   ```

2. Run the following command to query all the files in the test namespace:

   ```
   ls /mnt/jfs/test/
   ```

3. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs/test/dir1
   ls /mnt/jfs/test/
   ```

4. Run the following command to write data into a file:

   ```
   echo "hello world" > /tmp/hello.txt
   cp /tmp/hello.txt /mnt/jfs/test/dir1/
   ```

5. Run the following command to read data from a file:

   ```
   cat /mnt/jfs/test/dir1/hello.txt
   ```

   The following information is returned:

   ```
   hello world
   ```

The following example shows how to read data from and write data into files in Python:

1. Write data into a file in Python. Create the *write.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'w',encoding = 'utf-8') as f:
   f.write("my first file\n")
   f.write("This file\n\n")
   f.write("contains three lines\n")
```

2. Read data from a file in Python. Create the *read.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'r',encoding = 'utf-8') as f:
    lines = f.readlines()
    [print(x, end = '') for x in lines]
```

Run the following command to read the data that is written into the *test.txt* file:

```
[hadoop@emr-header-1 ~]$ ./read.py
```

The following information is returned:

```
my first file
This file
```

Unmount FUSE

Note Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to unmount FUSE:

```
umount jindofs-fuse
```

If the `target is busy` error is returned, switch to another directory, stop all the programs that are reading data from and writing data into FUSE files, and then unmount FUSE.

7.5.2. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain help information:

```
jindo distcp --help
```

The following information is returned:

```
     --help           - Print help text
     --src=VALUE          - Directory to copy files from
     --dest=VALUE              - Directory to copy files to
     --parallelism=VALUE         - Copy task parallelism
     --outputManifest=VALUE       - The name of the manifest file
     --previousManifest=VALUE   -   The path to an existing manifest file
     --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
if specified
     --copyFromManifest   -   Copy from a manifest instead of listing a directory
     --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
     --srcPattern=VALUE   -   Include only source files matching this pattern
     --deleteOnSuccess   -   Delete input files after a successful copy
     --outputCodec=VALUE   -   Compression codec for output files
     --groupBy=VALUE   -   Pattern to group input files by
     --targetSize=VALUE   -   Target size for output files
     --enableBalancePlan   -   Enable plan copy task to make balance
     --enableDynamicPlan   -   Enable plan copy task dynamically
     --enableTransaction   -   Enable transation on Job explicitly
     --diff   -   show the difference between src and dest filelist
     --ossKey=VALUE   -   Specify your oss key if needed
     --ossSecret=VALUE   -   Specify your oss secret if needed
     --ossEndPoint=VALUE   -   Specify your oss endPoint if needed
     --policy=VALUE   -   Specify your oss storage policy
     --cleanUpPending   -   clean up the incomplete upload when distcp job finish
     --queue=VALUE   -   Specify yarn queuename if needed
     --bandwidth=VALUE   -   Specify bandwidth per map/reduce in MB if needed
     --s3Key=VALUE   -   Specify your s3 key
     --s3Secret=VALUE   -   Specify your s3 Sercet
     --s3EndPoint=VALUE   -   Specify your s3 EndPoint
```

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-
01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03
/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03
/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017
-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-0
1/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-0
1/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_tab
le/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/
03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/
03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromMani
fest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

Note You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

> **Note** If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

--queue

--queue specifies the name of the YARN queue where the current DistCp task resides.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<your_bucket>/hourly_table --qu
eue yarnqueue
```

--bandwidth

--bandwidth specifies the bandwidth allocated to the current DistCp task. This prevents the task from using excessive bandwidth. Unit: MB/s.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.c
om --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
--policy archive --parallelism 20
```

- Example of the IA storage class (`--ia`):

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
--policy ia --parallelism 20
```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

Use Amazon S3 as a data source

You can use the --s3Key, --s3Secret, and --s3EndPoint parameters in a command to specify information related to Amazon S3.

Example:

```
jindo distcp jindo-distcp-2.7.3.jar --src s3a://yourbucket/ --dest oss://<your_bucket>/hour
ly_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com
```

You can configure the s3Key, s3Secret, and s3EndPoint parameters in the *core-site.xml* file of Hadoop. This way, you do not need to specify an AccessKey pair each time.

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

Example:

```
jindo distcp /tmp/jindo-distcp-2.7.3.jar --src s3://smartdata1/ --dest s3://smartdata1/tmp
--s3EndPoint  s3-us-west-1.amazonaws.com
```

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

> **Note** If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

7.5.3. Use Jindo DistCp in specific scenarios

This topic describes how to use Jindo DistCp in specific scenarios.

Prerequisites

- An E-MapReduce (EMR) cluster of a required version is created. For more information, see Create a cluster.
- Java Development Kit (JDK) 1.8 is installed.
- *jindo-distcp-<version>.jar* is downloaded based on your Hadoop version.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 2.7 and later 2.X versions.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 3.X.

Use scenarios

Common use scenarios of Jindo DistCp:

- Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?
- Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?
- Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?
- Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?
- Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file size?
- Scenario 8: What parameters are required if Amazon S3 is used as a data source?
- Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?
- Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?
- Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?
- Scenario 12: What parameters are required if I want to delete original files after a copy operation?
- Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?

If you are not using EMR, the following conditions must be met before you can import HDFS data to Object Storage Service (OSS):

- You can read data from HDFS.
- The AccessKey ID, AccessKey secret, and endpoint of OSS are obtained. You can write data to a destination OSS bucket.
- The storage class of the OSS bucket is not Archive.
- You can submit MapReduce tasks.
- The JAR package of Jindo DistCp is downloaded.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 10
```

> **Note** For more information about the parameters, see Use Jindo DistCp.

To import a large amount of data or a large number of files, such as millions or tens of millions of files, to OSS, you can set parallelism to a large value to increase the concurrency. You can also set `enableBatch` to true for optimization. Optimization command:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 500 --enableBatch
```

Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?

You can use one of the following methods to verify data integrity:

- Jindo DistCp Counters

  Check DistCp Counters in the counter information of a MapReduce task.

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

where:

- Bytes Destination Copied specifies the size of the file that you copy to a destination directory, in bytes.
- Bytes Source Read specifies the size of the file that you read from the source directory, in bytes.
- Files Copied specifies the number of copied files.

- Jindo DistCp --diff

  Use `--diff` to compare the information of files in the source and destination directories. The information includes file name and file size. Missing files and the files that fail to be copied are recorded in a manifest file. The manifest file is generated in the directory where you perform the comparison.

  Add `--diff` to the command described in Scenario 1. Example:

  ```
  hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
  -hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
  aliyuncs.com --diff
  ```

  If all files are copied, the following information is returned:

  ```
  INFO distcp.JindoDistCp: distcp has been done completely
  ```

Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?

If your DistCp task fails, and you want to perform resumable upload to copy only the files that fail to be copied, perform the following operations on the basis of the command described in Scenario 1:

1. Add `--diff` to check whether all files are copied.

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
   hou.aliyuncs.com --diff
   ```

   If all the files are copied, the following information is returned. Otherwise, a manifest file is generated. In this case, go to the next step.

   ```
   INFO distcp.JindoDistCp: distcp has been done completely.
   ```

2. Use `--copyFromManifest` and `--previousManifest` to copy the remaining files. Example:

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --dest oss://yang-hhht/hourly_table --previousManifest=file:///opt
   /manifest-2020-04-17.gz --copyFromManifest --parallelism 20
   ```

   `file:///opt/manifest-2020-04-17.gz` is the path where the manifest file is stored.

Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?

1. If no information is generated for the files that were copied last time, specify parameters to

generate a manifest file to record the information about the files that are copied.

Add `--outputManifest=manifest-2020-04-17.gz` and `--requirePreviousManifest=false` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

where:

- `--outputManifest` generates a manifest file whose name can be customized and must have a file name extension of gz, such as *manifest-2020-04-17.gz*. This file is stored in the directory specified by `--dest`.
- `--requirePreviousManifest` specifies whether to require a previous manifest file.

2. After a DistCp task is complete, copy the new files that may be generated in the source directory.

   Add `--outputManifest=manifest-2020-04-17.gz` and `--previousManifest=oss://yang-hhht/hourly_table/manifest-2020-04-17.gz` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

```
hadoop jar jindo-distcp-2.7.3.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --outputManifest=manifest-2020-04-18.gz --previousManifest=oss://yang-hhht
/hourly_table/manifest-2020-04-17.gz --parallelism 10
```

3. Repeat Step 2 to continuously copy incremental files.

Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?

Add the following two parameters to the command described in Scenario 1. These two parameters can be used together or separately.

- `--queue`: the name of the YARN queue.
- `--bandwidth`: the size of the specified bandwidth , in MB.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --queue yarnqueue --bandwidth 6 --parallelism 10
```

Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?

- To write data to OSS Archive storage, add `--archive` to the command described in Scenario 1.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --archive --parallelism 20
```

- To write data to OSS Infrequent Access (IA) storage, add `--ia` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --ia --parallelism 20
```

Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file sizes?

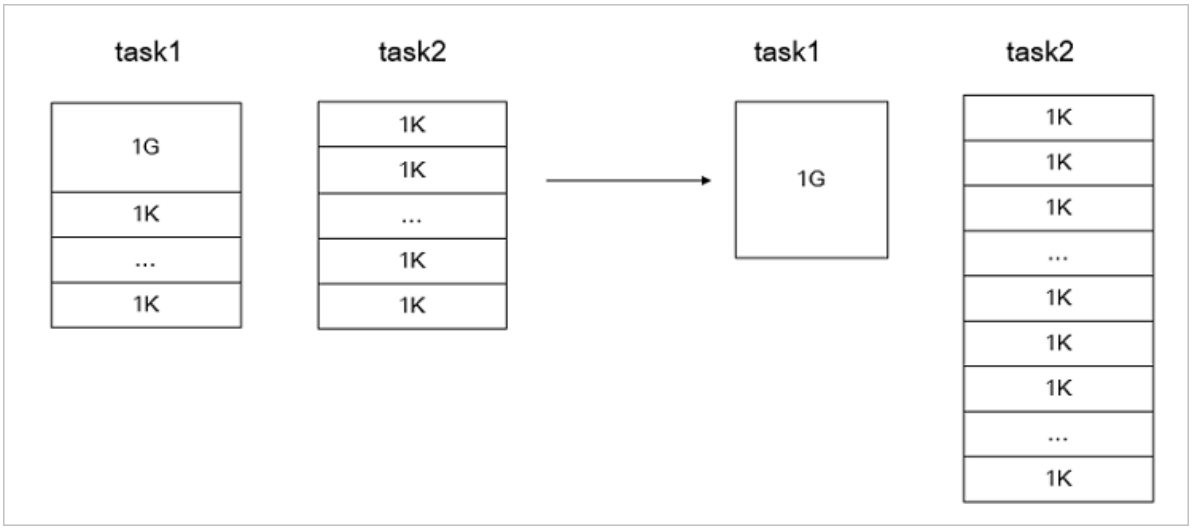
- A large number of small files and large data volume for a single large file

  If most of the files that you want to copy are small files, and the data volume of a single large file is large, the general solution is to randomly allocate the files to be copied. In this case, if you do not optimize the job allocation plan, few large files and many small files may be allocated to the same copy process. This cannot achieve excellent copy performance.

  Add `--enableDynamicPlan` to the command described in Scenario 1 to enable the optimization. --enableDynamicPlan cannot be used with `--enableBalancePlan`. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableDynamicPlan --parallelism 10
```

  The following figure shows the data copy before and after the plan is optimized.

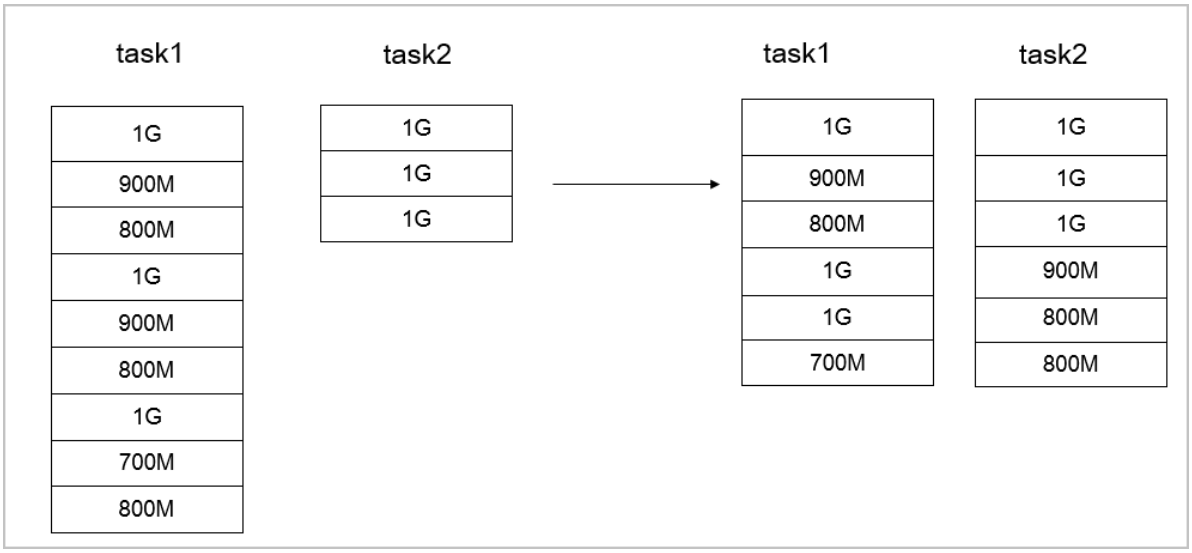


- No significant differences in sizes of files

  If the sizes are not significantly different among the files that you want to copy, use `--enableBalancePlan` to optimize the job allocation plan. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableBalancePlan --parallelism 10
```

The following figure shows the data copy before and after the plan is optimized.



Scenario 8: What parameters are required if Amazon S3 is used as a data source?

Replace the parameters that specify the AccessKey ID, AccessKey secret, and endpoint of OSS in the command described in Scenario 1 with the following parameters of Amazon S3:

- `--s3Key` : the AccessKey ID for Amazon S3
- `--s3Secret` : the AccessKey secret for Amazon S3
- `--s3EndPoint` : the endpoint for Amazon S3

Example:

```
hadoop jar jindo-distcp-<version>.jar --src s3a://yourbucket/ --dest oss://yang-hhht/hourly
_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com --para
llelism 10
```

Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?

You can use `--outputCodec` to compress copied files in a format such as LZO or GZ to reduce the space that is used to store the files.

Add `--outputCodec` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --outputCodec=gz --parallelism 10
```

You can set --outputCodec to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files are compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change to the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?

- If you want to copy the files that meet specific rules, add `--srcPattern` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPattern .*\.log --parallelism 10
```

  `--srcPattern` : specifies a regular expression that filters files for the copy operation.

- If you want to copy the files stored in some sub-directories of the same parent directory, add `--srcPrefixesFile` to the command described in Scenario 1.

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

  `--srcPrefixesFile` : enables Jindo DistCp to copy files in multiple folders under the same parent directory at a time.

  Content of the *folders.txt* file:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?

Add the following parameters to the command described in Scenario 1:

- `--targetSize` : the maximum size of the merged file, in MB
- `--groupBy` : the merging rule, which is a regular expression

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Scenario 12: What parameters are required if I want to delete original files after a copy operation?

Add `--deleteOnSuccess` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --deleteOnSuccess --parallelism 10
```

Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Jindo DistCp allows you to save the AccessKey ID, AccessKey secret, and endpoint of OSS in the *core-site.xml* file so that you do not need to repeatedly specify the information.

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of OSS, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.jfs.cache.oss-accessKeyId</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-accessKeySecret</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-endpoint</name>
        <value>oss-cn-xxx.aliyuncs.com</value>
    </property>
</configuration>
```

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of Amazon S3, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

7.5.4. Use tiered storage commands of JindoFS

In E-MapReduce (EMR) V3.30, JindoFS provides a tiered storage feature. This feature allows you to store cold and hot data in different storage media. This helps reduce the data storage costs or accelerate data access.

Use jindo jfs

Run the following command to obtain the help information:

```
jindo jfs -help archive
-archive -i/a <path> ... :
  Archive commands.
```

Tiered storage commands in JindoFS are asynchronous and start only related tasks.

Common commands:

- Cache
- Uncache
- Archive
- Unarchive
- Status
- ls2 command

Cache

You can use this command to back up data stored in a specific path to local disks. Then, you can read the data from local disks, without the need to read data from Object Storage Service (OSS).

```
jindo jfs -cache -p <path>
```

`-p` can be used to ensure that local data is not cleared based on disk usage.

Uncache

You can use this command to delete backup data from local disks and store data only to OSS Standard storage.

```
jindo jfs -uncache  <path>
```

Archive

You can use this command to delete backup data from local disks and store data to OSS in Infrequent Access (IA) or Archive mode. For information about the storage classes, see Overview.

```
jindo jfs -archive -i|-a <path>
```

`-i` is used to store data to OSS in IA mode. `-a` is used to store data to OSS in Archive mode.

Unarchive

You can use this command to restore data from the Archive mode to the IA or Standard mode. You can temporarily restore data stored in Archive mode to allow the data to be readable within one day.

```
jindo jfs -unarchive -i/-o <path>
```

`-i` is used to store data to OSS in IA mode. `-o` is used to temporarily restore data stored in Archive mode to allow the data to be readable.

Status

You can use this command to view task details. By default, the number of files to be subject to tiered storage in a specific directory and the data that has been subject to tiered storage are measured.

```
jindo jfs -status -detail/-sync <path>
```

`-detail` is used to view the storage progress of file data. `-sync` indicates that the command exits after a tiered storage task is completed.

ls2 command

JindoFS provides the ls2 command that allows you to view the file storage status on the basis of Hadoop ls commands:

```
hadoop fs -ls2 <path>
```

Example of command output, which includes the file storage class:

```
drwxrwxrwx  - -         0    2020-06-05 04:27 oss://xxxx/warehouse
-rw-rw-rw-  1 Archive   1484 2020-09-23 16:40 oss://xxxx/wikipedia_data.csv
-rw-rw-rw-  1 Standard  1676 2020-06-07 20:04 oss://xxxx/wikipedia_data.json
```

8.SmartData 3.2.X

8.1. SmartData 3.2.X

SmartData is a storage service for the E-MapReduce (EMR) Jindo engine. SmartData provides centralized storage and optimized caching and computing for EMR computing engines and extends storage features. SmartData is composed of JindoFS, JindoTable, and related tools. This topic describes the updates in SmartData 3.2.X.

OSS storage scalability on JindoFS

- JindoFS uses multiple password-free methods to obtain a token that is used to access OSS. The token can be customized or extended.
- Alibaba Cloud Tablestore is used to implement mutual exclusion on the operations that are performed with the rename operation at the same time.
- Data can be written to OSS by using Delta or Hudi.

JindoFS-based caching optimization

JindoFS optimizes the caching of metadata in a large number of small files in AI training scenarios and improves the performance of metadata preloading operations and list operations.

JindoTable-based computing optimization

- JindoTable is integrated with AliORC to provide a native Optimized Row Columnar (ORC) reader. JindoTable allows Spark and Presto to use a native ORC reader to read ORC files. This helps accelerate data reading and improve the computing performance.
- JindoTable can be used to collect access frequency statistics of Hive tables for Presto.

Ecosystem support for JindoFS

When you use Spark to write data to OSS, you can set `spark.hadoop.mapreduce.fileoutputcommitter.marksuccessfuljobs` to false to avoid generating a *_SUCCESS* file.

8.2. JindoFS in block storage mode

8.2.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read and write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

Context

The block storage mode of JindoFS has the following features:

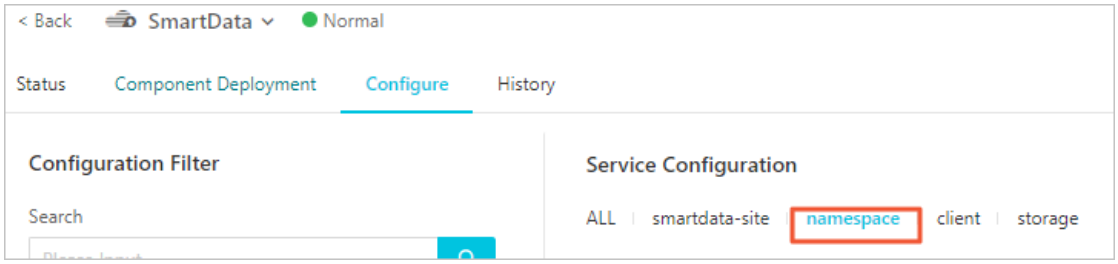
- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the

throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.

- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

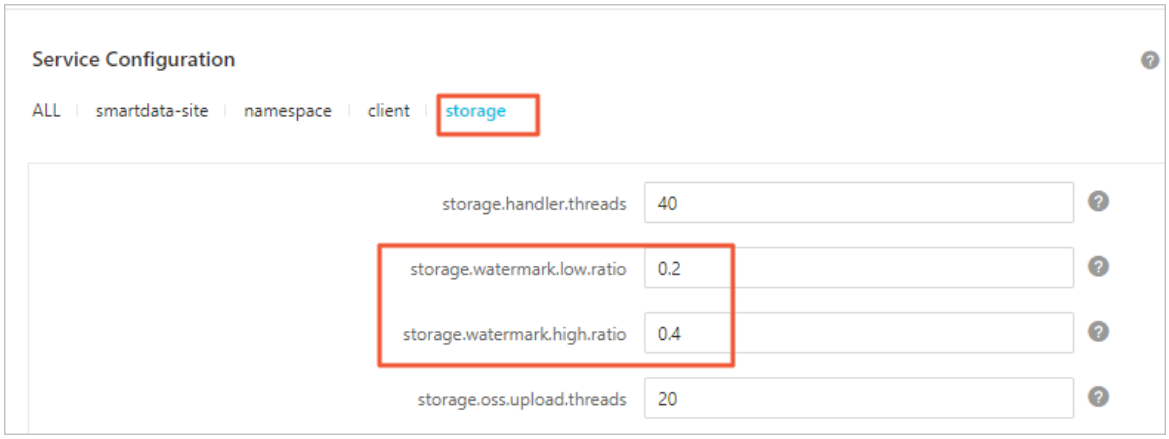
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Choose **Actions > Restart Jindo Storage Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

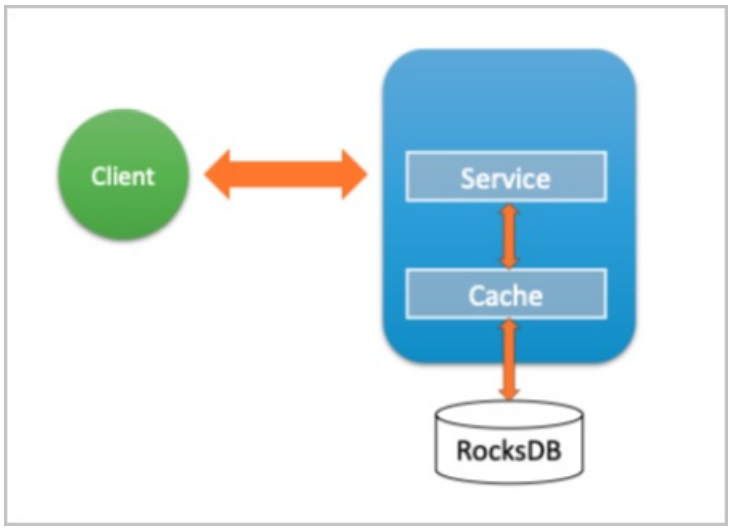
8.2.2. Use RocksDB to store metadata

Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

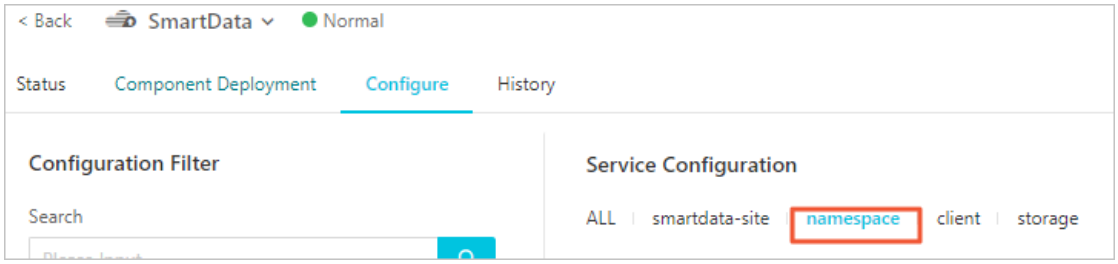
RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage backend in high availability mode, we recommend that you use Raft instances. For more information, see Use Raft-RocksDB-Tablestore to store metadata.

The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
6. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

7. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

8. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. Prepare for the recovery.

   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

   ```
   hadoop fs -count jfs://test/
   ```

   Information similar to the following example is returned:

   ```
    1596      1482809                 25 jfs://test/
   ```

   The number of folders is 1596, and the number of files is 1,482,809.

   ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1`, the latest metadata is synchronized to the Tablestore instance.

   ```
   jindo jfs -metaStatus
   ```

   Information similar to that shown in the following figure is returned.

   ```
   [hadoop@emr-header-1 ~]$ jindo jfs -metaStatus
   ==== emr-header-1:8101 ====
   OtsUploader: _synced=1
   [RocksDB Row Counts of each CF(Table)]
   ```

   iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

   ```
   jindo jfs -metaStatus
   ```

   If the state is FINISH, the recovery is completed, as shown in the following figure.

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

8.2.3. Use Raft-RocksDB-Tablestore to store metadata

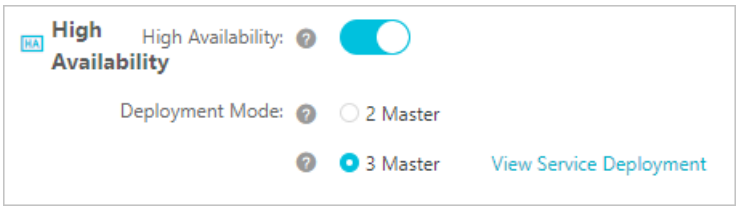
In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata.

Prerequisites

- A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  Note You must enable the transaction feature.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.

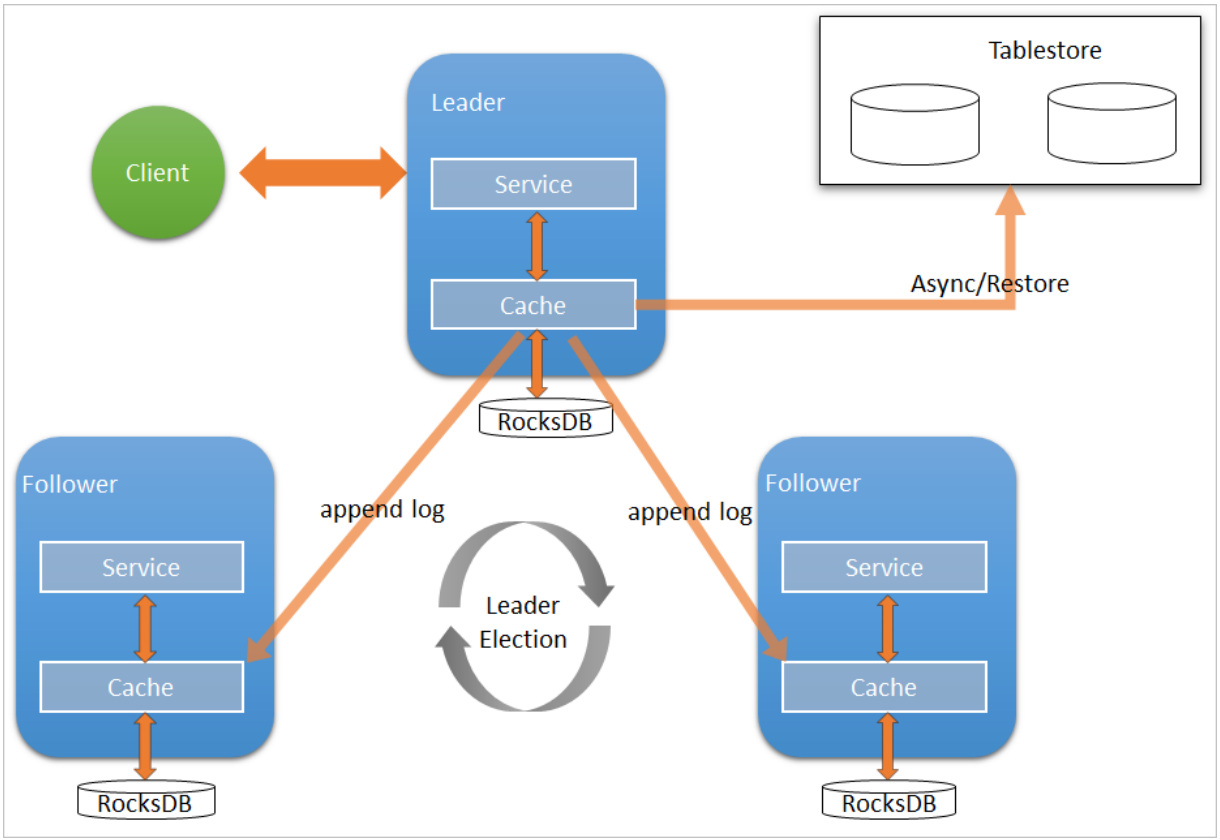


  Note If no deployment modes are available, submit a ticket.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.



Configure a Raft instance as the local storage backend

1. Suspend all the SmartData components of an EMR cluster.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   vi. Select **Stop All Components** from the **Actions** drop-down list in the upper-right corner.
2. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
3. Go to the **namespace** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **namespace** tab.
4. Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. Valid values: ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

Note If you do not need to use Tablestore for remote storage, skip Step 5.

5. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

```
hadoop fs -count jfs://test/
      1596      1482809                 25 jfs://test/
  (Number of folders) (Number of files)
```

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```

```
[RaftPeerImpl]
peer id: [illegible]
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [illegible]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 2335ms)
snapshot_timer: timeout(3600000ms) SCHEDULING(in 150305ms)
storage: [1, 624625]
disk_index: 624625
known_applied_index: 624625
last_log_id: (index=624625,term=2)
first_index_pinned: 624625
state_machine: Idle
last_committed_index: 624625
last_snapshot_index: 0
last_snapshot_term: 0
snapshot_status: IDLE
replicator_25769803789@[illegible] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=624261 ic=0
replicator_329853488332[illegible] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=623564 ic=0

OtsUploader: _lastStopIndex=624624, _synced=1
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

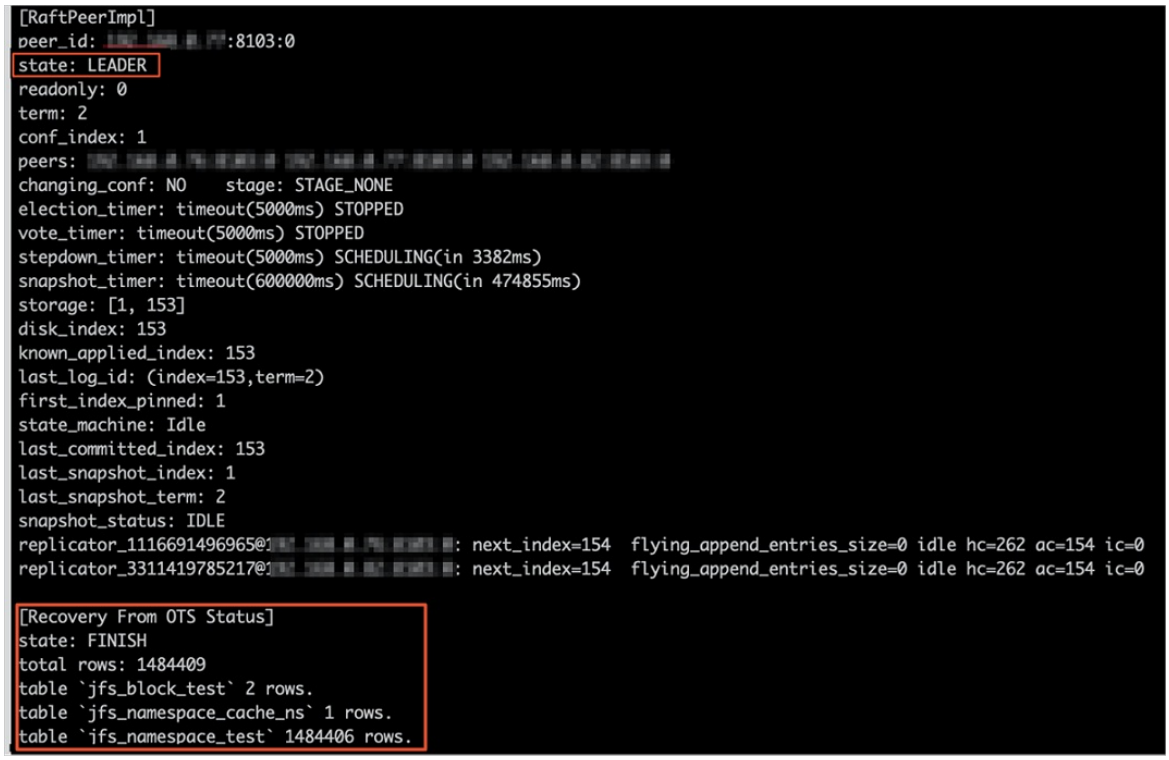
ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.



7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

8.2.4. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see

Create a cluster.

- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.

ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

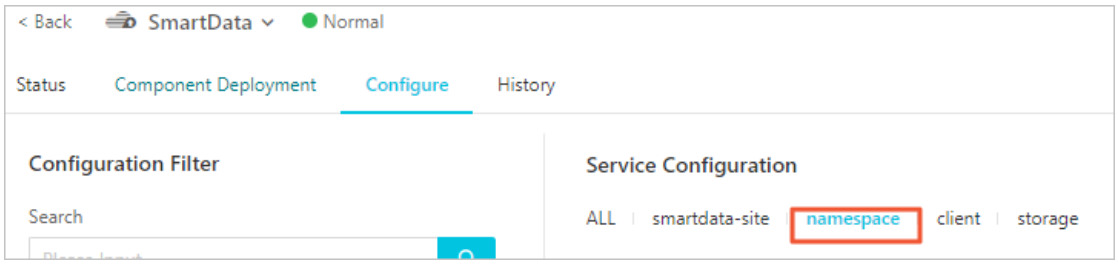
iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **namespace** tab for the SmartData service.

   i. Click the **Configure** tab.

   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:

   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

8.2.5. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

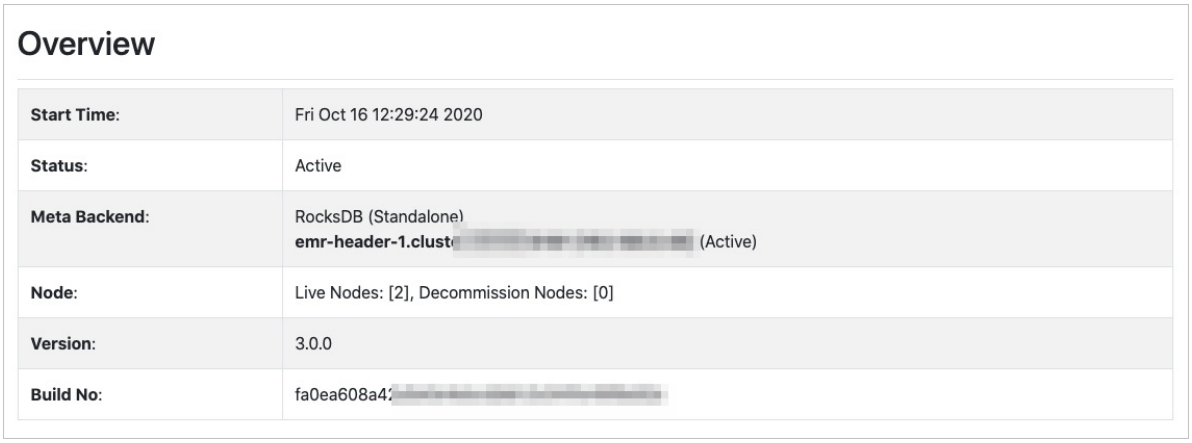
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

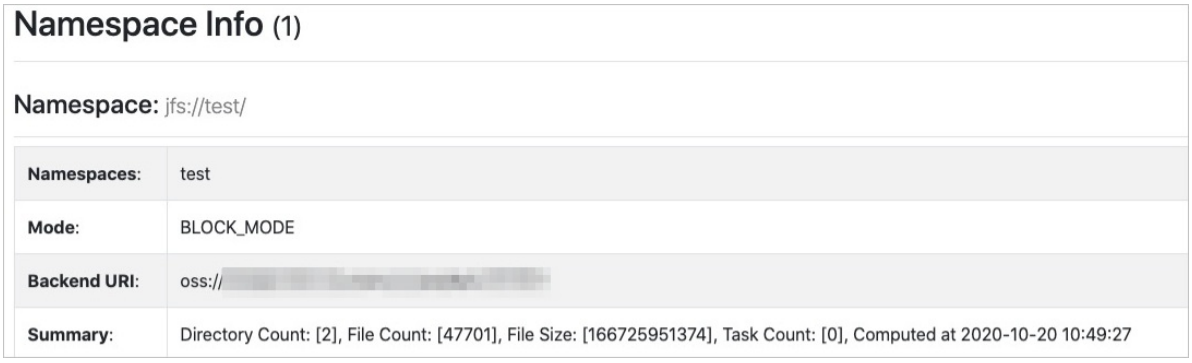


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



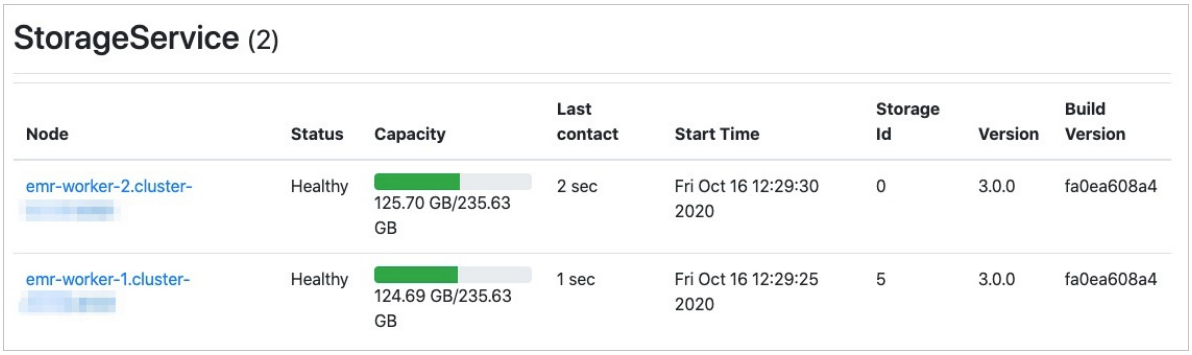
Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

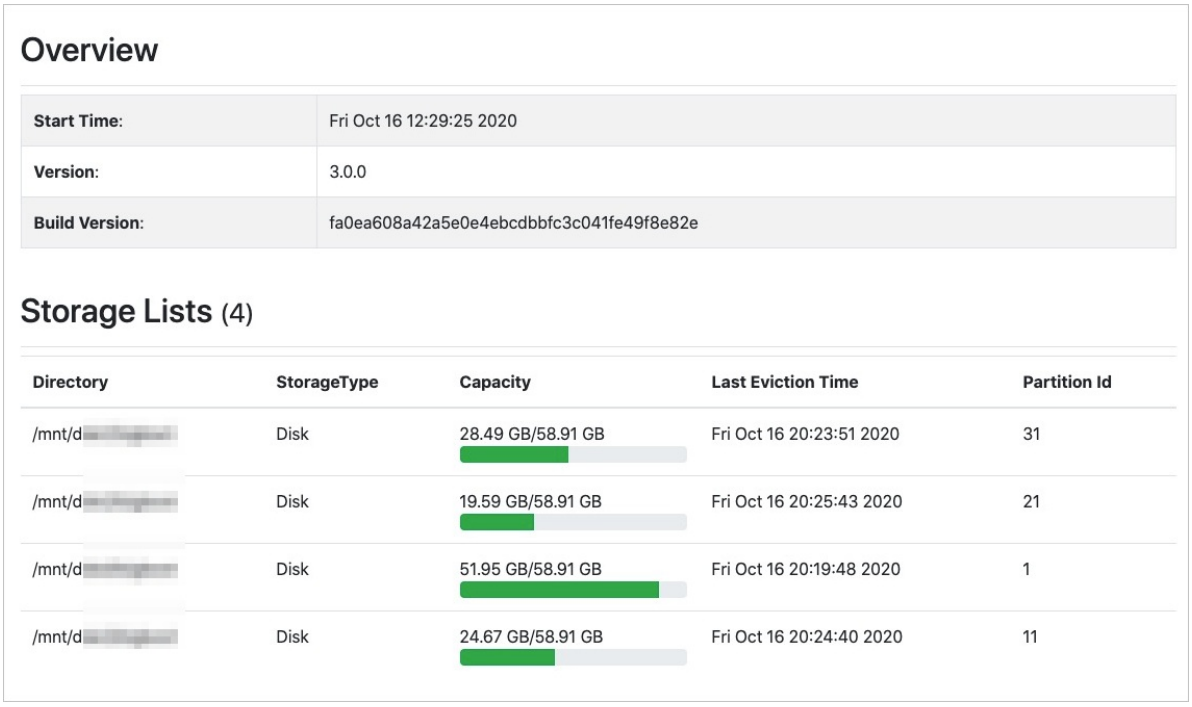
  This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.



StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.



Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

8.2.6. Manage JindoFS permissions

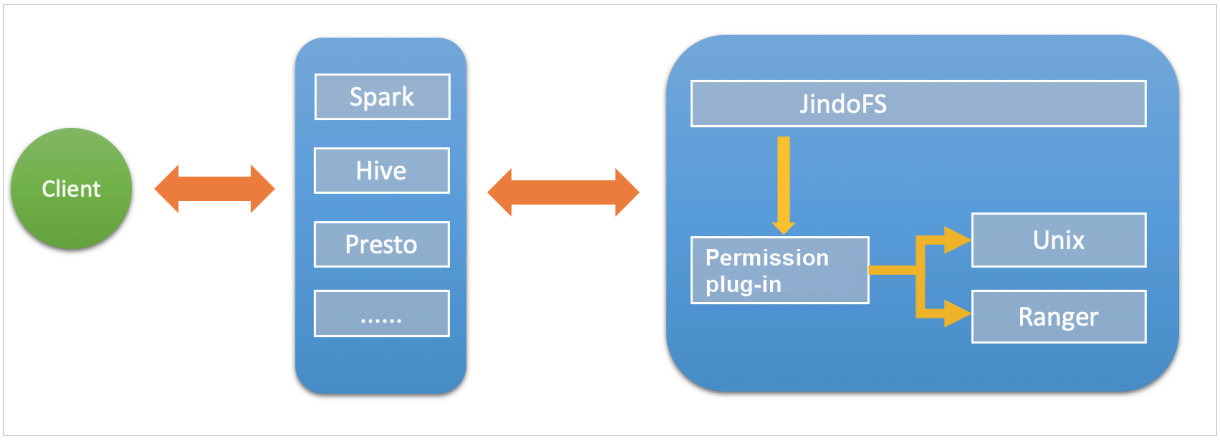
This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

Permission management based on the storage mode:

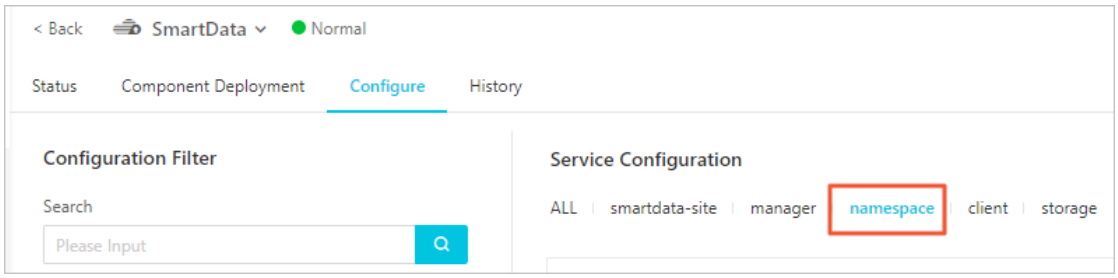
- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.
  - Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

8.2.7. Data management policies

JindoFS in block storage mode provides advanced data management policies to meet storage requirements in different scenarios. The advanced policies include storage policies and compression policies. This topic describes the policies in detail and provides examples to show how to use the policies.

Storage policies

JindoFS provides the five storage policies described in the following table to support flexible storage.

Policy	Description
AR	Data has only one backup. The backup is stored as an Archive object in OSS.
IA	Data has only one backup. The backup is stored as an Infrequent Access (IA) object in OSS.

Policy	Description
COLD	Data has only one backup. The backup is stored as a Standard object in OSS.
WARM	Data has one backup in OSS and one local backup. The local backup can effectively accelerate data read operations. By default, this storage policy is used.
HOT	Data has one backup in OSS and one local backup. The local backup is forcibly locked. This way, it is not deleted when the cache is automatically cleared. This policy can achieve a better acceleration effect than the WARM policy and is suitable for the hottest data.

For more information about the storage classes of OSS, see Overview.

A newly added file is stored based on the storage policy specified for its parent directory. Example:

- Run the following command to configure a storage policy:

```
jindo jfs -setStoragePolicy [-R] <StoragePolicy>(AR/IA/COLD/WARM/HOT) <path> ...
```

  where:

  - `[-R]`: specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.
  - `<path>`: the name of the directory for which the storage policy is configured.

- Run the following command to obtain the storage policy configured for a directory:

```
jindo jfs -getStoragePolicy <path>
```

Compression policies

JindoFS allows you to compress data blocks before you store them. This feature effectively reduces the storage space occupied by data blocks and improves data read and write efficiency. This feature is suitable for files that have a high compression ratio. The following table describes the supported compression policies.

Policy	Description
NONE	Data blocks are not compressed. By default, this compression policy is used.
ZSTD	The Zstandard compression algorithm is used to compress data blocks.

A newly added file is compressed based on the compression policy specified for its parent directory before it is stored. Example:

- Run the following command to configure a compression policy:

```
jindo jfs -setCompressionPolicy [-R] <CompressionPolicy>(NONE/ZSTD) <path> ...
```

where:

- `[-R]`: specifies that a recursive operation is performed to configure the same compression policy for all sub-directories of the directory.
- `<path>`: the name of the directory for which the compression policy is configured.

- Run the following command to obtain the compression policy configured for a directory:

```
jindo jfs -getCompressionPolicy <path> ...
```

8.2.8. Analyze metadata offline

In E-MapReduce (EMR) V3.30.0 or later, JindoFS in block storage mode allows you to dump the metadata of an entire namespace to Object Storage Service (OSS) and use Jindo SQL to analyze the metadata.

Context

In Hadoop Distributed File System (HDFS), the metadata is stored in a snapshot file named fsimage. The snapshot file contains the following information about metadata: namespace, file that stores the metadata, blocks, and file system quota. In HDFS, you can run a command to download the fsimage file in the XML format to your on-premises machine and view this file to analyze metadata offline. JindoFS does not require you to download metadata to your on-premises machine.

Upload HDFS metadata to OSS

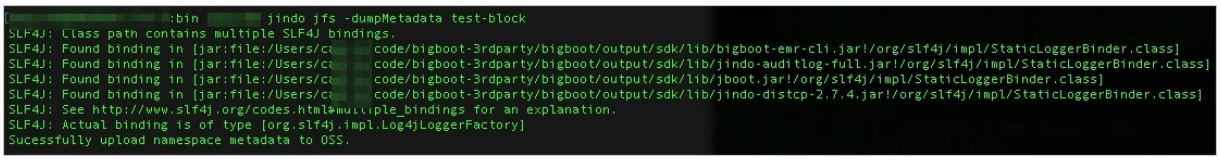
Run the following command to upload metadata of a specific namespace to OSS:

```
jindo jfs -dumpMetadata <nsName>
```

`<nsName>` indicates the name of the namespace in block storage mode.

For example, upload the metadata of the test-block namespace to OSS and analyze the metadata offline.

```
jindo jfs -dumpMetadata test-block
```



If the following information appears, data is uploaded to OSS and stored as a file in the JSON format.

```
Sucessfully upload namespace metadata to OSS.
```

Metadata upload path

The metadata upload path is the metadataDump sub-directory under *sysinfo* in JindoFS.

For example, if `namespace.sysinfo.oss.uri` is set to `oss://abc/`, metadata is uploaded to the `oss://abc/metadataDump` directory.

Parameter	Description
namespace.sysinfo.oss.uri	The path of the OSS bucket.
namespace.sysinfo.oss.endpoint	The endpoint. An endpoint specified in a different region can also be used.
namespace.sysinfo.oss.access.key	The AccessKey ID of your Alibaba Cloud account.
namespace.sysinfo.oss.access.secret	The AccessKey secret of your Alibaba Cloud account.

Batch information: Metadata from HDFS changes based on the usage. Each time you want to analyze metadata, you must run commands to view the snapshot file of the metadata. Each time you run a Jindo command to upload metadata, a batch number that is generated based on the upload time is used as the root directory to store the metadata. This ensures that the data you uploaded each time is not overwritten. You can delete historical data based on your requirements.

- 1 indicates the configuration path of system information in OSS.
- 2 indicates a namespace.
- 3 indicates a batch number.

Metadata schema

HDFS metadata that is uploaded to OSS is stored as a file in the JSON format. Schema information:

```
{
  "type":"string",          /*Inode type, FILE DIRECTORY*/
  "id": "string",            /*INode id*/
  "parentId" :"string",         /*Parent node ID*/
  "name":"string",         /*Inode name*/
  "size": "int",         /*Inode size, BIGINT*/
  "permission":"int",         /*Permission, INT*/
  "owner":"string",          /*Owner name*/
  "ownerGroup":"string",     /*Owner group name*/
  "mtime":"int",              /*Inode modification time, BIGINT*/
  "atime":"int",              /*The time when the inode was last accessed, BIGINT*/
  "attributes":"string",       /*File-related attributes*/
  "state":"string",            /*Inode status*/
  "storagePolicy":"string",    /*Storage policy*/
  "etag":"string"           /*etag*/
}
```

Use Jindo SQL to analyze metadata

1. Run the command shown in the following figure to start Jindo SQL.

```
[root@emr-header-1 ~]# jindo sql
Spark master: yarn, Application Id: application_1603081647416_0050
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.33 seconds, Fetched 3 row(s)
```

2. Query the tables whose data can be analyzed by using Jindo SQL.
   - Run the `show tables` command to view the tables whose data can be analyzed. Jindo SQL has two built-in modules audit_log and fs_image for audit and metadata analysis.
   - Run the `show partitions fs_image` command to view the partition information of the fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

     The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

3. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       1192276202347928724 9   1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       1546468482017665002     1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The namespace and datetime columns are added and indicate the namespace name and the timestamp when metadata is uploaded.

Example: Obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

Use Hive to analyze metadata

1. Create a table schema in Hive.

   You can use the following command to create a table schema of the metadata that is used for queries in Hive:

```
CREATE EXTERNAL TABLE `table_name`
(`type` string,
 `id` string,
 `parentId` string,
 `name` string,
 `size` bigint,
 `permission` int,
 `owner` string,
 `ownerGroup` string,
 `mtime` bigint,
 `atime` bigint,
 `attr` string,
 `state` string,
 `storagePolicy` string,
 `etag` string)
 ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
 STORED AS TEXTFILE
 LOCATION 'The OSS path to which file data is uploaded';
```

2. Use Hive to analyze data offline.

   After you create a Hive table, you can use Hive SQL to analyze metadata.

```
select * from table_name limit 200;
```

   The following figure shows an example.

```
hive> select * from inode_metadata_test0 limit 100;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_202009[redacted]
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1[redacted], Tracking URL = http://emr-heade[redacted]:20888/proxy/applica[redacted]
Kill Command = /usr/lib/hadoop-current/bin/hadoop job  -kill job_1599[redacted]
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-09-08 14:57:26,112 Stage-1 map = 0%,  reduce = 0%
2020-09-08 14:57:31,263 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.22 sec
MapReduce Total cumulative CPU time: 1 seconds 220 msec
Ended Job = job_[redacted]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.22 sec   HDFS Read: 6867 HDFS Write: 1524 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 220 msec
OK
Directory       1127433438684721971 4    1127433438684721971 3    /uttest/oss     0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Directory       1127433438684721971 9    1127433438684721971 3    /uttest/oss2    0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
Directory       1127433438684721971 6    1127433438684721971 4    /uttest/oss/dir 0       511     caojie  staff   1599545017636   1599545017636           Finalized       WARM
File    1127433438684721971 5    1127433438684721971 4    /uttest/oss/file1       0       420     caojie  staff   1599545017632   1599545017632           Finalized       WARM
File    1127433438684721971 7    1127433438684721971 6    /uttest/oss/dir/file2   0       420     caojie  staff   1599545017642   1599545017642           Finalized       WARM
File    1127433438684721971 8    1127433438684721971 6    /uttest/oss/dir/file3   0       420     caojie  staff   1599545017651   1599545017651           Finalized       WARM
Directory       1127433438684721972 0    1127433438684721971 9    /uttest/oss2/dir        0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
File    1127433438684721972 1    1127433438684721972 0    /uttest/oss2/dir/file2  0       420     caojie  staff   1599545017658   1599545017658           Finalized       WARM
File    1127433438684721972 2    1127433438684721972 0    /uttest/oss2/dir/file3  0       420     caojie  staff   1599545017666   1599545017666           Finalized       WARM
Directory       1127433438684721971 3    1767282355743385190 5    /uttest 0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Time taken: 10.734 seconds, Fetched: 10 row(s)
hive>
```

8.3. JindoFS in cache mode

8.3.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in Object Storage Service (OSS) based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

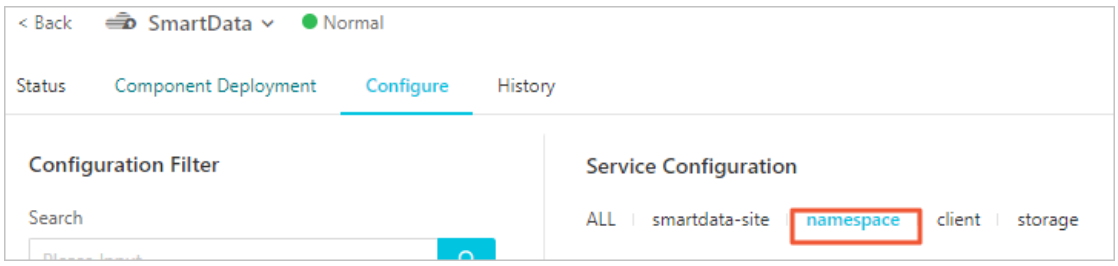
  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

   ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. Click **OK**.
5. Save the configurations.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

   iii. Click **OK**.

6. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **client** tab.
2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   You do not need to restart the SmartData service to apply the new configurations.

3. Save the configurations.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

   iii. Click **OK**.

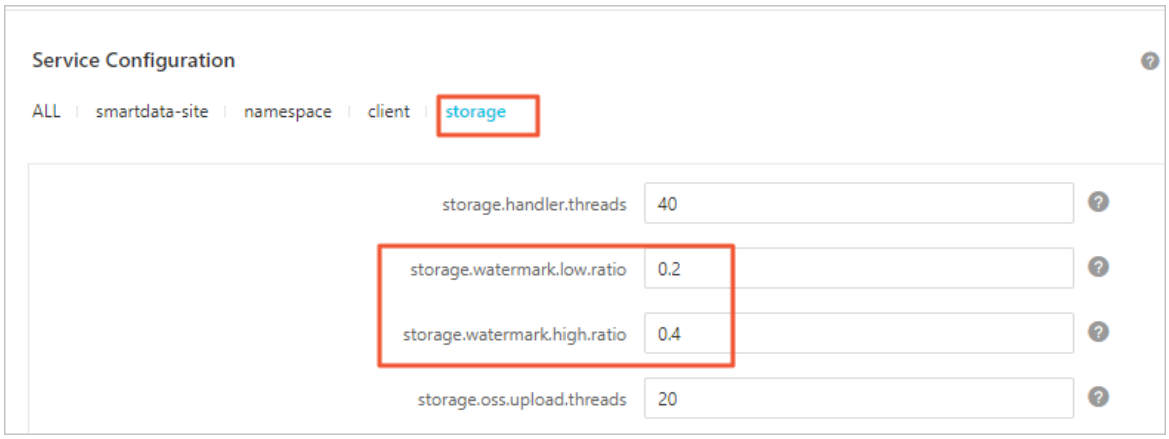
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you specify. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right

corner.

ii. In the **Cluster Activities** dialog box, specify the related parameters.

iii. Click **OK**.

iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to specify an AccessKey pair. In other cases, you must specify an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you choose for accessing files in OSS:

- OSS Scheme

  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.

  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss.accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.endpoint	The endpoint of the OSS bucket that serves as the storage backend.

> **Note** You can set these parameters to the values of the parameters in an EMR version earlier than EMR V3.30.0.

- JFS Scheme

  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **namespace** tab.

  ii. Set **jfs.namespaces** to **test**.

  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.

Parameter	Description
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize the cache performance. After you configure the parameters, you do not need to restart the SmartData service to apply the new configurations.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. You can use this parameter to prevent upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If an excessive amount of memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

8.3.2. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

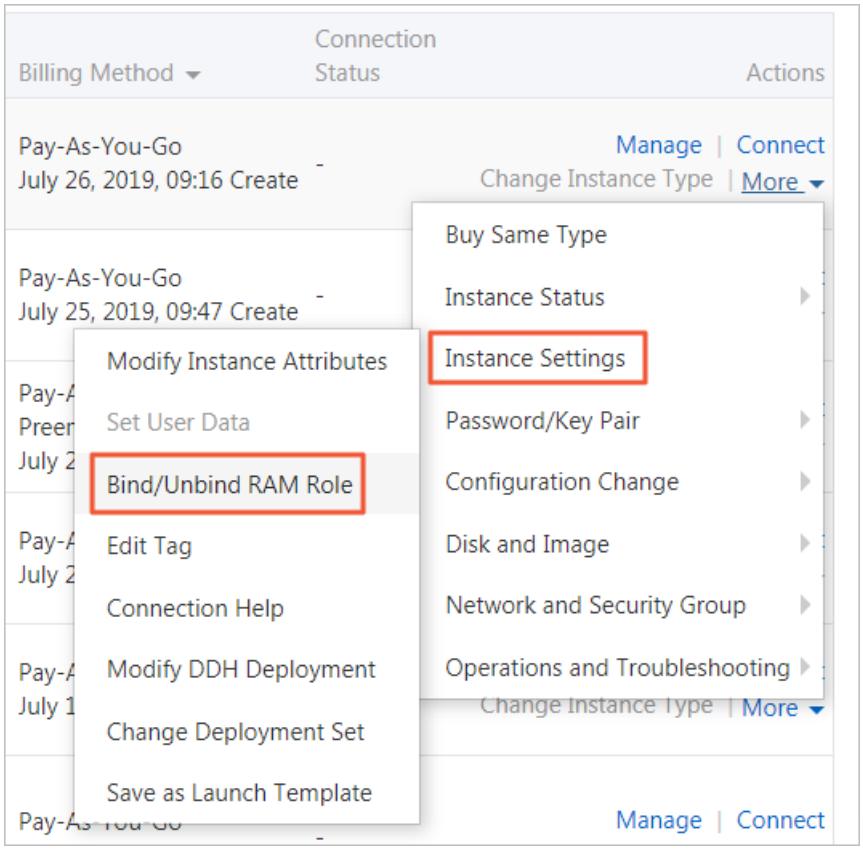
1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.
4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

   ```
   hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
   ```

2. Access OSS by using the FileSystem interface of Hadoop.

   Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

8.3.3. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false

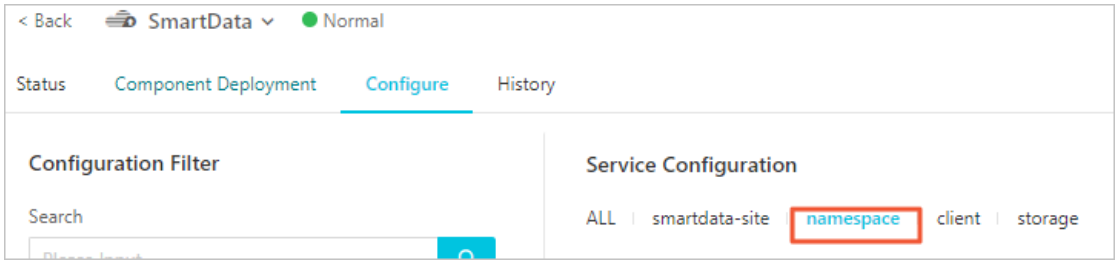
Parameter	Description
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-

right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

8.3.4. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

> **Notice**
>
> - The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
> - After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
> - Hive and Presto jobs do not use Hadoop Job Committer.
> - In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.

i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
ii. Click the **Configure** tab.
iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.
6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.

3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

   The default value is 8.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

8.3.5. Use a credential provider

You can use a credential provider to save an encrypted AccessKey pair into a file. This prevents the leak of the AccessKey pair.

Context

You can use a Hadoop credential provider to save an encrypted AccessKey pair into a file. This prevents the issue that the AccessKey pair is transmitted in plaintext. You can select an appropriate JindoOSS credential provider based on your business requirements.

Configure a JindoOSS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **smartdata-site** tab.

   i. Click the **Configure** tab.

   ii. In the **Service Configuration** section, click the **smartdata-site** tab.

3. Add configuration information.

   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

   - Global configuration (for all buckets)

Parameter	Description
fs.jfs.cache.oss.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

- Bucket-level configuration

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types. **Note** XXX indicates the name of an Object Storage Service (OSS) bucket.

Credential provider types

You can select different credential providers based on your business requirements. The following providers are supported:

- Global configuration
  - TemporaryAliyunCredentialsProvider

    This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.cache.oss.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

- Bucket-level configuration

- TemporaryAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.
fs.jfs.cache.oss.bucket.XXX.securityToken	The temporary security token used to access the OSS bucket.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access an OSS bucket.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access the OSS bucket.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access the OSS bucket. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

Use a Hadoop credential provider to store AccessKey pair information

> **Note** For more information about Hadoop credential providers, see CredentialProvider API Guide.

Use a command that is provided by Hadoop to store AccessKey pair and security token information into a credential file. Syntax:

```
hadoop credential <subcommand> [options]
```

For example, in global configuration mode, store AccessKey pair and token information into a JCEKS file. You can protect the file by using file permissions or you can specify a password to encrypt the information you want to store. If you do not specify a password, the default string is used for encryption.

```
hadoop credential create fs.jfs.cache.oss.accessKeyId -value AAA -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.accessKeySecret -value BBB -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.securityToken -value  CCC -provider jceks://file/root/oss.jceks
```

After a credential file is generated, you must configure the parameter described in the following table to specify the provider type and location.

Parameter	Description
fs.jfs.cache.oss.security.credential.provider.path	The path used to store the credential file that stores AccessKey pair information. For example, you can set this parameter to *jceks://file/${user.home}/oss.jceks*, which indicates that the *oss.jceks* file is stored in the home directory.

8.3.6. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

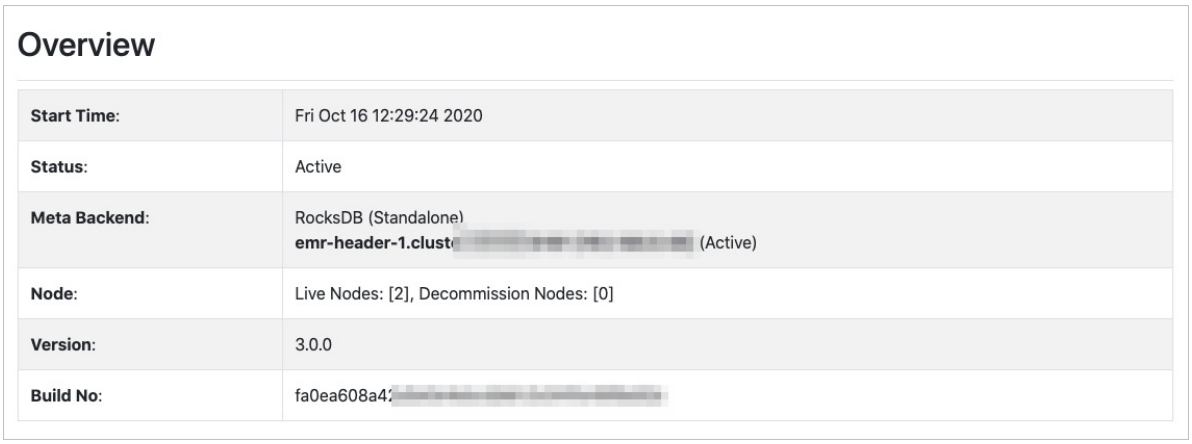
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

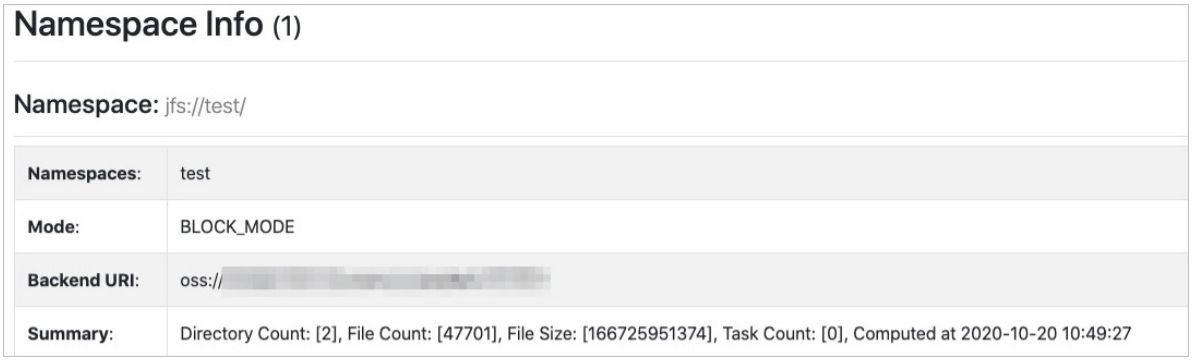


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.

StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.

Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

8.3.7. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

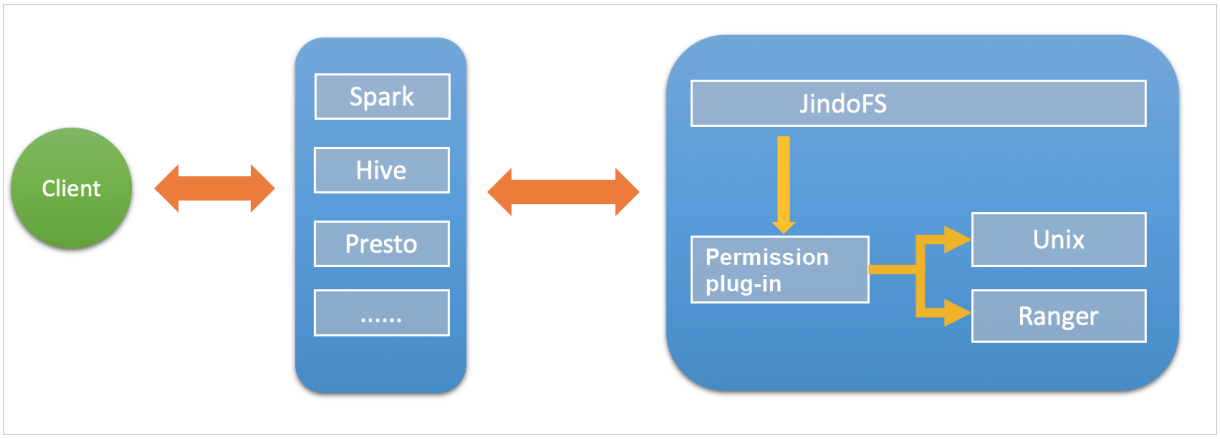
Permission management based on the storage mode:

- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.

- Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.

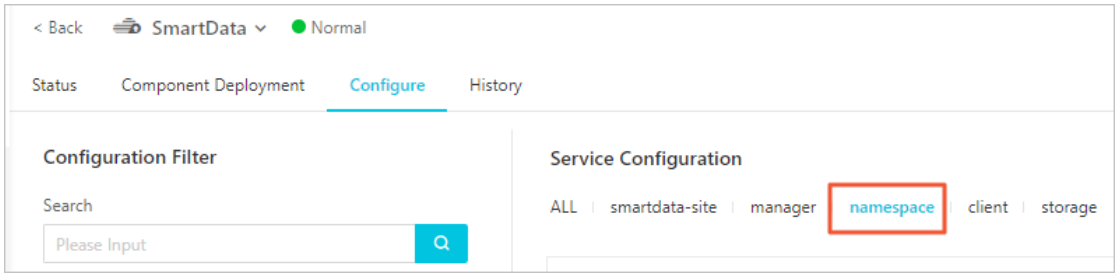
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

   After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

   If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

8.4. JindoTable

8.4.1. Enable query acceleration for ORC files

JindoTable provides a native Optimized Row Columnar (ORC) reader that can be used to accelerate queries of ORC files. By default, query acceleration is disabled. Enabling query acceleration can improve the performance of Spark or Presto to read ORC files.

Prerequisites

ORC files are stored to JindoFS or Object Storage Service (OSS).

> **Note** You cannot accelerate queries for HDFS.

Improve performance of Spark to read data

1. Enable query acceleration.

> **Note** When Spark is used to read ORC files, the DataFrame or Spark-SQL API is required to enable acceleration.

- Configure the global parameter.

  For more information, see Configure the global parameter of Spark.

- Configure job-level parameters.

  You can add Spark startup parameters when you run Spark Shell or Spark SQL jobs.

  ```
  --conf spark.sql.extensions==io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sql.JindoTableExtension
  ```

  For more information about the configuration of jobs, see Configure a Spark Shell job or Configure a Spark SQL job.

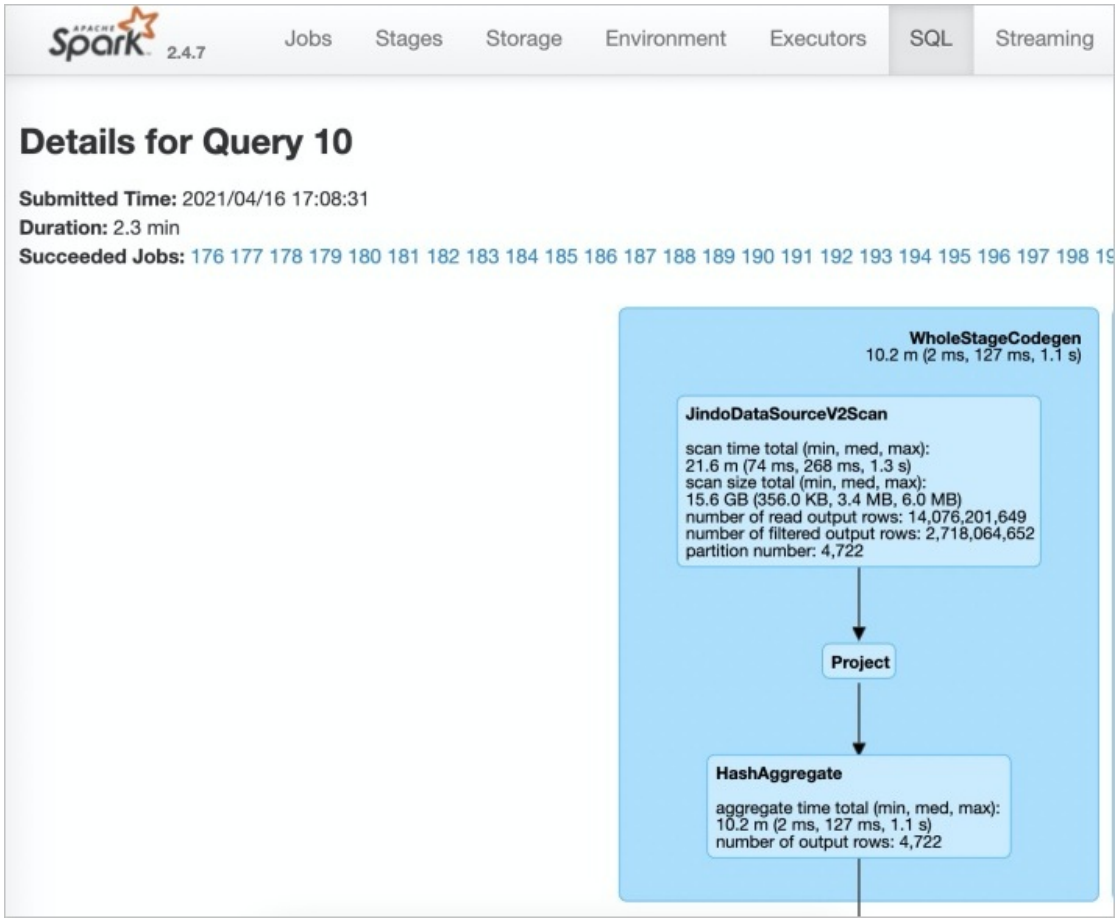
2. Check whether query acceleration is enabled.

   i. Access the web UI of Spark History Server UI.

   For more information, see Access the web UIs of open source components.

   ii. On the SQL tab of Spark, view the execution task.

   If JindoDataSourceV2Scan appears, query acceleration is enabled. Otherwise, check the configurations in Step 1.



Improve performance of Presto to read data

Presto has built-in `catalog: hive-acc`. You can use `catalog: hive-acc` to enable query acceleration.

Example:

```
presto --server https://emr-header-1.cluster-xxx:7778/ --catalog hive-acc --schema default
```

> **Note** `emr-header-1.cluster-xxx` indicates the hostname of the emr-header-1 node.

Configure the global parameter of Spark

1. Go to the Spark service page.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > Spark**.
2. On the Spark service page, click the **Configure** tab.
3. Find the spark.sql.extensions parameter and change its value to io.delta.sql.DeltaSparkSessionExtension,com.aliyun.emr.sql.JindoTableExtension.
4. Save the configurations.
   i. Click **Save** in the upper-right corner of the Service Configuration section.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
5. Restart ThriftServer.
   i. Choose **Actions > Restart ThriftServer** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

8.4.2. Use JindoTable

JindoTable is used to implement tiered storage, optimize table files, and collect data statistics based on the popularity of tables or partitions. This topic describes how to use JindoTable.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your on-premises machine.
- An EMR cluster of the 3.30.0 version or later is created. For more information about how to create a cluster, see Create a cluster.

Use JindoTable

Common commands:

- -accessStat
- -cache

- -archive
- -unarchive
- -uncache
- -status
- -optimize
- -showTable
- -showPartition
- -listTables
- -dumpmc

> **Notice** Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

-accessStat

- Syntax

  jindo table -accessStat {-d} *<days>* **{-n}** <topNums>

- Description

  This command is used to query the access records in which tables or partitions are visited most in a specified time range.

  <days> and <topNums> must be positive integers. If <days> is 1, all the access records generated from 00:00 (local time) on the current day to the current time are queried.

- Example: Query the 20 access records of tables or partitions that are most frequently visited within the last seven days.

```
jindo table -accessStat -d 7 -n 20
```

-cache

- Syntax

  jindo table -cache {-t} <dbName.tableName> **[-p]** <partitionSpec> **[-pin]**

- Description

  This command is used to cache data of specified tables or partitions to local disks.

  The data of the tables or partitions must be stored in Object Storage Service (OSS) or JindoFileSystem (JindoFS). Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`. When you specify `-pin`, if the cache space is insufficient, do not delete related data if possible.

- Example: Cache data of the db1.t1 table that is generated on March 16, 2020 to local disks.

```
jindo table -cache -t db1.t1 -p date=2020-03-16
```

-uncache

- Syntax

jindo table -uncache {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to delete cached data of specified tables or partitions from local disks.

  The data of the tables or partitions must be stored in OSS or JindoFS. Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

- Examples:
  - Delete cached data of the db1.t2 table from local disks.

    ```
    jindo table -uncache -t db1.t2
    ```

  - Delete cached data of the db1.t1 table from local disks.

    ```
    jindo table -uncache -t db1.t1 -p date=2020-03-16,category=1
    ```

-archive

- Syntax

  jindo table -archive {-a|i} {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to lower the level of the policy that is used to store data of specified tables or partitions. By default, the Archive storage class is used.

  To use the Infrequent Access (IA) storage class, add -i to the command. Specify tables in the format of database.table. Specify partitions in the format of 'partitionCol1=1,partitionCol2=2,...'.

- Example: Cache data of the db1.t1 table to local disks.

  ```
  jindo table -archive -t db1.t1 -p date=2020-10-12
  ```

-unarchive

- Syntax

  jindo table -unarchive [-o|-i] {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to change the storage class from Archive to Standard.

  If `-o` is added to the command, an Archived object is temporarily restored. If `-i` is added to the command, an Archived object is changed to an IA object.

- Example:

  ```
  jindo table -unarchive -o -t db1.t1 -p date=2020-03-16,category=1
  ```

  ```
  jindo table -unarchive -i -t db1.t2
  ```

-status

- Syntax

  jindo table -status {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to view the data storage status of specified tables or partitions.

- Examples:
  - View the data storage status of the db1.t2 table.

    ```
    jindo table -status -t db1.t2
    ```

  - View the data storage status of the db1.t1 table on March 16, 2020.

    ```
    jindo table -status -t db1.t1 -p date=2020-03-16
    ```

-optimize

- Syntax

  jindo table -optimize {-t} <dbName.tableName>

- Description

  This command is used to optimize the data organization of tables at the storage layer.

- Example: Optimize the data organization of the db1.t1 table at the storage layer.

  ```
  jindo table -optimize -t db1.t1
  ```

-showTable

- Syntax

  jindo table -showTable {-t} <dbName.tableName>

- Description

  This command is used to display all partitions in a partitioned table or display the data storage of a non-partitioned table.

- Example: Display all partitions in the db1.t1 partitioned table.

  ```
  jindo table -showTable -t db1.t1
  ```

-showPartition

- Syntax

  jindo table -showPartition {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to display the data storage of partitions.

- Example: Display the data storage of all partitions in the db1.t1 partitioned table on October 12, 2020.

  ```
  jindo table -showPartition -t db1.t1 -p date=2020-10-12
  ```

-listTables

- Syntax

  jindo table -listTables [-db] <dbName>

- Description

  This command is used to display all tables in a specified database. If you do not specify `[-db]`, tables in the default database are displayed.

- Examples:
  - Display tables in the default database.

    ```
    jindo table -listTables
    ```

  - Display tables in the db1 database.

    ```
    jindo table -listTables -db db1
    ```

-dumpmc

- Syntax

  jindo table -dumpmc {-i} <accessId> **{-k}** <accessKey> **{-m}** <numMaps> **{-t}** <tunnelUrl> **{-project}** <projectName> **{-table}** <tablename> **{-p}** <partitionSpec> **{-f}** <csv|tfrecord> **{-o}** <outputPath>

Parameter	Description	Required
-i	The AccessKey ID of your Alibaba Cloud account.	Yes
-k	The AccessKey secret of your Alibaba Cloud account.	Yes
-m	The number of map tasks.	Yes
-t	The VPC Tunnel endpoint of MaxCompute.	Yes
-project	The name of the MaxCompute project.	Yes
-table	The name of the MaxCompute table.	Yes
-p	The partition information. Example: `pt=xxx`. Separate multiple partitions with commas (,), such as `pt=xxx,dt=xxx`.	No
-f	The file format. Valid values: ○ tfrecord ○ csv	Yes
-o	The destination path.	Yes

- Description

This command is used to dump MaxCompute tables to an EMR cluster or OSS. The formats CSV and TFRECORD are supported.

- Examples:
  - Dump a MaxCompute table in the TFRECORD format to an EMR cluster.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o /tmp/outputtf1 -f tfrecord
```

  - Dump a MaxCompute table in the CSV format to OSS.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o oss://bucket1/tmp/outputcsv -f csv
```

8.4.3. Use JindoCube

JindoCube is supported in E-MapReduce (EMR) V3.24.0 and later. This topic describes how to install, deploy, and use JindoCube in EMR.

Prerequisites

A table or view is created.

Overview

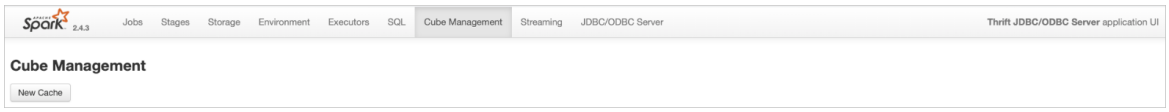
JindoCube is an advanced feature of EMR Spark. It implements pre-calculation to accelerate data processing. It improves query performance by dozens or even hundreds of times. You can persist data of a view to a data source supported by EMR Spark, such as HDFS or Object Storage Service (OSS). EMR Spark automatically discovers available persistent data and optimizes the execution plans for the data. This process is completely visible to users. JindoCube applies to business scenarios where a fixed query mode is used. JindoCube pre-calculates and pre-organizes data to accelerate data queries. For example, you can use JindoCube to implement multidimensional online analytical processing (MOLAP), generate data reports, create data dashboards, and synchronize data across clusters.

Install and deploy JindoCube

JindoCube is an advanced feature of EMR Spark. You can use JindoCube to accelerate all Dataset, DataFrame API, and SQL tasks that are submitted by using EMR Spark. No additional component deployment or maintenance is required.

1. Manage JindoCube on the Spark web UI.

   You can manage JindoCube on the Spark web UI. For example, you can create, delete, or update JindoCube caches. After you create a JindoCube cache, it automatically accelerates all Spark queries in the cluster. You can use the **spark.sql.cache.tab.display** parameter to specify whether to display the Cube Management tab on the Spark web UI. You can specify this parameter on the configuration page of the Spark service in the EMR console or in a Spark task submission command. The default value of this parameter is **false**.



You can also use the **spark.sql.cache.useDatabase** parameter to create a database based on your business requirements and store the view for which you want to create a cache in this database. If partitioned tables are used, you can use the **spark.sql.cache.cacheByPartition** parameter to specify whether to cache data by partition.

Parameter	Description	Example
spark.sql.cache.tab.display	Specifies whether to display the Cube Management tab.	true
spark.sql.cache.useDatabase	The databases where cubes are stored.	db1,db2,dbn
spark.sql.cache.cacheByPartition	Specifies whether to store cubes by partition.	true

2. Optimize Spark queries.

   You can use the **spark.sql.cache.queryRewrite** parameter to specify whether to use a JindoCube cache to accelerate Spark queries. You can specify this parameter at the cluster, session, or SQL statement level. The default value is **true**.

Use JindoCube

1. Create a JindoCube cache.
   i. Log on to the Alibaba Cloud EMR console by using your Alibaba Cloud account.
   ii. Click the **Cluster Management** tab.
   iii. Find the cluster where you want to create a JindoCube cache and click the cluster ID.
   iv. In the left-side navigation pane, click **Connect Strings**.
   v. On the **Public Connect Strings** page, click the link of YARN UI to go to the YARN web UI as a Knox user.

      For more information about Knox, see Knox.
   vi. Click **ApplicationMaster** in the row where **Application Type** is **SPARK** and **Name** is **Thrift JDBC/ODBC Server**.
   vii. On the Spark web UI, click the **Cube Management** tab.
   viii. Click **New Cache**.

      Select a table or view, and click raw cache or cube cache in the action column. You can create two types of caches:

- Raw cache: persists the data of a table or view based on the cache configurations.

  When you create a raw cache, specify the parameters described in the following table.

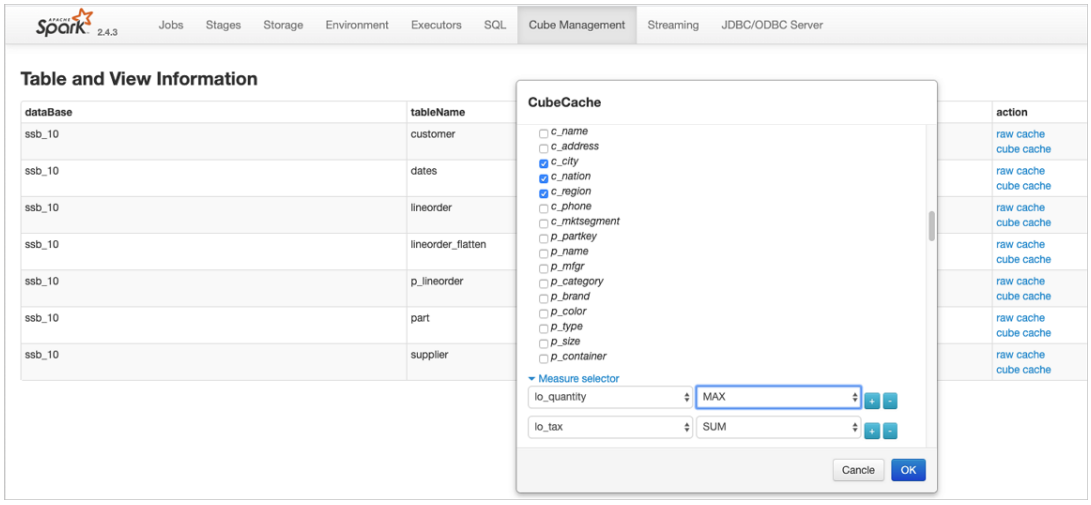
Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Column Selector	The columns from which you want to cache data.	Yes
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No

- Cube cache: builds a cube based on the raw data of a table or view and persists the cube data.

  When you create a cube cache, specify the parameters described in the following table.

Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Dimension Selector	The dimension fields that are used to build a cube.	Yes
Measure Selector	The measure fields and pre-calculation functions that are used to build a cube.	Yes

Parameter	Description	Required
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No



JindoCube builds a cube based on specified dimensions and measures. The following SQL statement shows the configurations of the cube cache in the preceding figure:

```
SELECT c_city, c_nation, c_region, MAX(lo_quantity), SUM(lo_tax)
FROM lineorder_flatten
GROUP BY c_city, c_nation, c_region;
```

JindoCube calculates the finest-grained combination of dimensions. Powered by the onsite computing capability of Spark, JindoCube implements re-aggregation to optimize the queries where a coarse-grained combination of dimensions is used. You must use the pre-calculation functions supported by JindoCube to define a cube cache. The following table lists the supported pre-calculation functions and the aggregate functions that correspond to them.

Aggregate function	Pre-calculation function
COUNT	COUNT
SUM	SUM
MAX	MAX
MIN	MIN
AVG	COUNT and SUM
COUNT (DISTINCT)	PRE_COUNT_DISTINCT
APPROX_COUNT_DISTINCT	PRE_APPROX_COUNT_DISTINCT

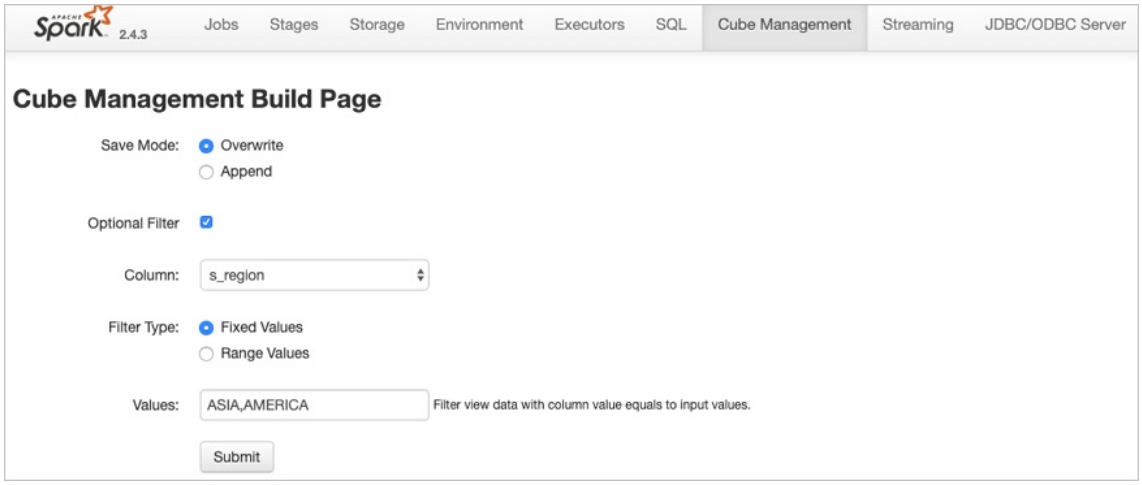
All created caches are displayed on the **Cube Management** tab. You can click **Detail** for a cache to view its details, including its basic information, partition information, building information, and building history.

2. Build a JindoCube cache.

   During the creation of a cache, metadata is prepared, but no data is persisted. Therefore, after you create a cache, you must continue to build the cache. This way, the related data can be persisted to permanent storage, such as HDFS or OSS. In addition, the source table of a cache may be appended with new data or updated. In this case, you must update the data in the cache to ensure data consistency. You can use one of the following methods to build a cache:

   - Manually build a cache

     Click Build Cache to manually build a cache. On the page that appears, configure the parameters, as shown in the following figure.



     The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.

Optional Filter	You can select this parameter to add filter conditions. This way, JindoCube filters data based on the filter conditions before it persists the data.
	■ Column: the field based on which data is filtered.
	■ Filter Type: specifies whether to filter data by fixed values or by value range.
	  ■ Fixed Values: If this option is selected, specify one or more values. Separate values with commas (,).
	  ■ Range Values: If this option is selected, specify a minimum value and a maximum value. You can leave the maximum value empty. In this case, the value range is the minimum value and all values greater than it.

The following SQL statement shows the configurations of the raw cache for the lineorder_flatten view in the preceding figure:

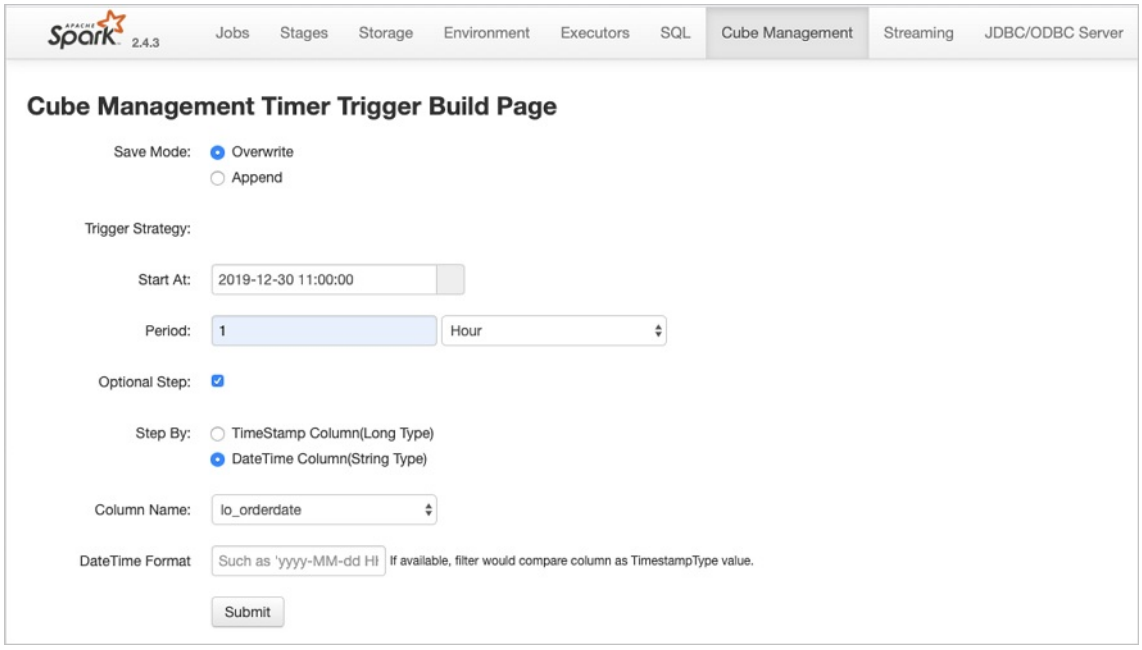
```
SELECT * FROM lineorder_flatten
WHERE s_region == 'ASIA' OR s_region == 'AMERICA';
```

Click **Submit** to submit the cache building task to a Spark cluster. You can view the task status in the Build Information section of the cache details page. After the task is complete, you can view the cache information in the Build History section.

> **Note** Spark writes the cached data to a specified directory in the same way common Spark tasks write data to a table or directory. If multiple users concurrently build a cache whose data format is Parquet, JSON, or ORC, the cached data may be inaccurate and invalid. If concurrent cache building operations or updates are inevitable, we recommend that you use a data format that supports concurrent writes, for example, Delta.

- Configure periodic cache building

  Click Trigger Period Build to configure a periodic cache update policy to ensure data consistency between the cache and source table. On the page that appears, configure the parameters, as shown in the following figure.



The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.
Trigger Strategy	Specifies the start time of the first building task, and the building interval. ■ Start At: Select or enter the time you want to trigger the first building task. Format: yyyy-MM-dd hh:mm:ss. ■ Period: Specify the interval at which cache building tasks are triggered.

Parameter	Description
Optional Step	Specifies the filter conditions for cache building. JindoCube updates the cache in incremental mode based on the filter conditions and interval. If you do not select Optional Step, JindoCube updates the entire cache each time. ■ Step By: Specify the type of the field that you want to update in incremental mode. TimeStamp Column(Long Type) indicates a timestamp field of the LONG type. DateTime Column(String Type) indicates a datetime field of the STRING type. ■ Column Name: Specify the name of the field that you want to update in incremental mode.

On the cache details page, you can view the configured cache update policy. You can click Cancel Period Build to cancel periodic updates. You can view the information of a complete building task in the Build History section.

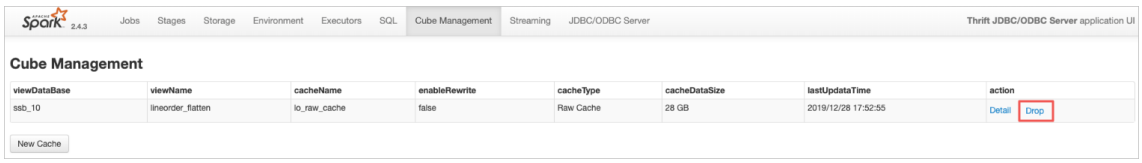
> **Note**
>
> - Periodic building tasks run on a Spark cluster. The related configurations are all saved in SparkContext, and the tasks are periodically triggered by Spark Driver. If you release the Spark cluster, periodic building is disabled.
> - The Build History section displays the following information of each complete task in the current Spark cluster: start time, end time, save mode, building conditions, and final status. After you release the Spark cluster, the information in the Build History section is cleared.

3. Manage a JindoCube cache.

   After you create and build a JindoCube cache, you can manage it on the Cube Management page.

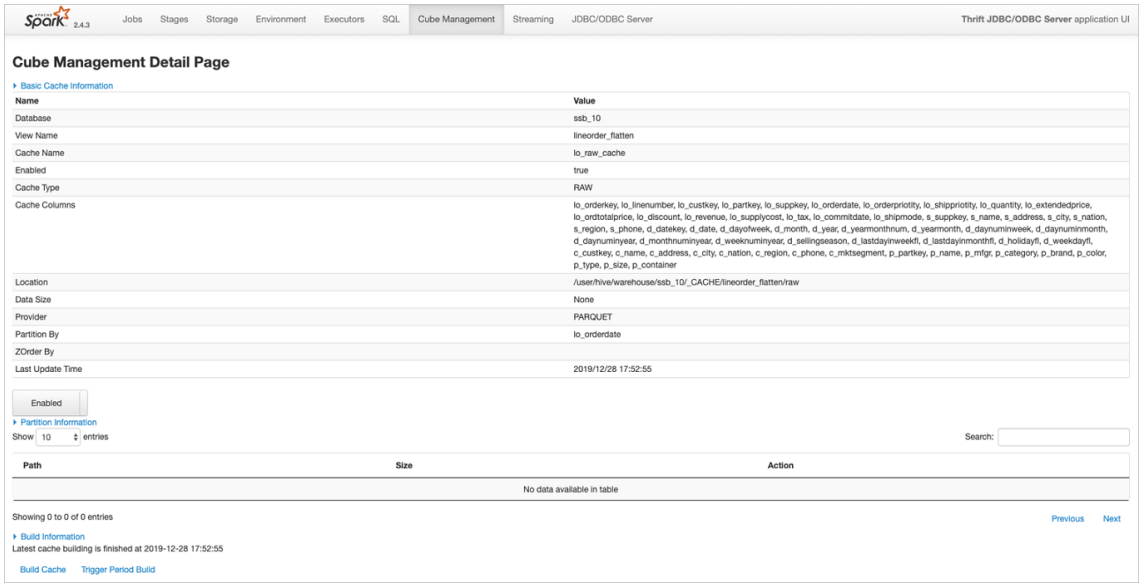
   - Delete a cache

     On the Cube Management page, find the cache you want to delete, and click **Drop** in the action column. All the metadata and data of the cache are deleted.



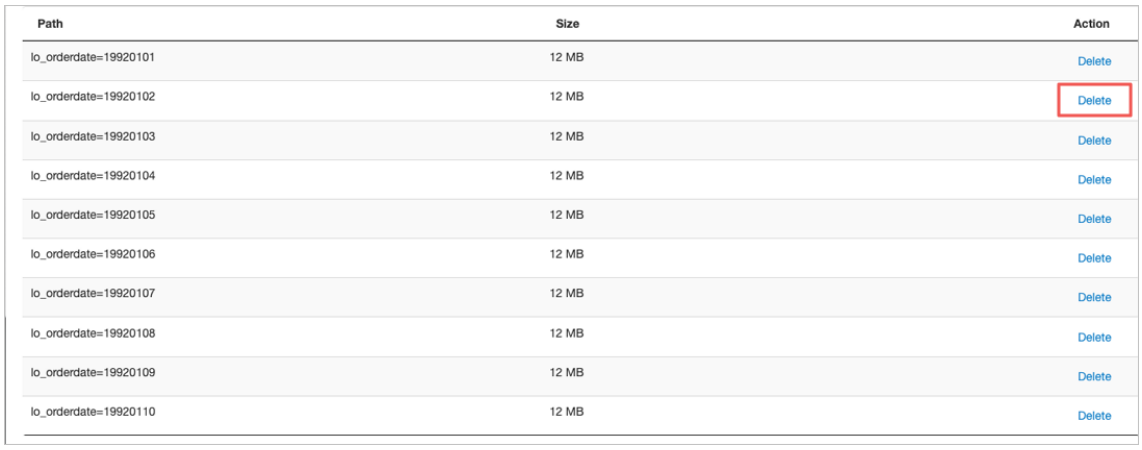
   - Enable and disable optimization of Spark queries

     JindoCube allows you to specify whether to optimize Spark queries at the cache level. On the details page of a cache, click **Enabled** or **Disabled** in the lower part of the Basic Cache Information section to enable or disable the optimization.



- Delete partition data from a cache

  If data in a cache is stored by partition, you can delete the unnecessary partition data from the cache to release more space. On the details page of the cache, find the partition whose data you want to delete, and click **Delete** in the Action column.



Path	Size	Action
lo_orderdate=19920101	12 MB	Delete
lo_orderdate=19920102	12 MB	Delete
lo_orderdate=19920103	12 MB	Delete
lo_orderdate=19920104	12 MB	Delete
lo_orderdate=19920105	12 MB	Delete
lo_orderdate=19920106	12 MB	Delete
lo_orderdate=19920107	12 MB	Delete
lo_orderdate=19920108	12 MB	Delete
lo_orderdate=19920109	12 MB	Delete
lo_orderdate=19920110	12 MB	Delete

> **Note** Before you delete the data of a partition, make sure that the data is no longer needed. If you delete data that is required in subsequent queries, the queries return invalid results.

4. Optimize Spark queries.

   JindoCube optimizes view-based queries. After you have created a raw cache or cube cache for a view, EMR Spark overwrites the execution plan of queries on this view. The new execution plan still uses the original logical semantics but can directly access the data in the cache. This accelerates Spark queries. Use the lineorder_flatten view as an example. It is a wide table view generated by associating the lineorder table with other dimension tables. The following figure shows the information about this view.

```
0: jdbc:hive2://localhost:10001/ssb_10> desc extended lineorder_flatten;
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+-----------------------------+-----------------------------------------------------+----------+
|           col_name          |                     data_type                       | comment  |
+-----------------------------+-----------------------------------------------------+----------+
| lo_orderkey                 | bigint                                              | NULL     |
| lo_linenumber               | bigint                                              | NULL     |
| lo_custkey                  | int                                                 | NULL     |
| lo_partkey                  | int                                                 | NULL     |
| lo_suppkey                  | int                                                 | NULL     |
| lo_orderdate                | int                                                 | NULL     |
| lo_orderpriotity            | string                                              | NULL     |
| lo_shippriotity             | int                                                 | NULL     |
| lo_quantity                 | bigint                                              | NULL     |
| lo_extendedprice            | bigint                                              | NULL     |
| lo_ordtotalprice            | bigint                                              | NULL     |
| lo_discount                 | bigint                                              | NULL     |
| lo_revenue                  | bigint                                              | NULL     |
| lo_supplycost               | bigint                                              | NULL     |
| lo_tax                      | bigint                                              | NULL     |
| lo_commitdate               | int                                                 | NULL     |
| lo_shipmode                 | string                                              | NULL     |
| s_suppkey                   | int                                                 | NULL     |
| s_name                      | string                                              | NULL     |
| s_address                   | string                                              | NULL     |
| s_city                      | string                                              | NULL     |
| s_nation                    | string                                              | NULL     |
| s_region                    | string                                              | NULL     |
| s_phone                     | string                                              | NULL     |
| d_datekey                   | int                                                 | NULL     |
| d_date                      | string                                              | NULL     |
| d_dayofweek                 | string                                              | NULL     |
| d_month                     | string                                              | NULL     |
| d_year                      | int                                                 | NULL     |
| d_yearmonthnum              | int                                                 | NULL     |
| d_yearmonth                 | string                                              | NULL     |
| d_daynuminweek              | int                                                 | NULL     |
| d_daynuminmonth             | int                                                 | NULL     |
| d_daynuminyear              | int                                                 | NULL     |
| d_monthnuminyear            | int                                                 | NULL     |
| d_weeknuminyear             | int                                                 | NULL     |
| d_sellingseason             | string                                              | NULL     |
| d_lastdayinweekfl           | int                                                 | NULL     |
| d_lastdayinmonthfl          | int                                                 | NULL     |
| d_holidayfl                 | int                                                 | NULL     |
| d_weekdayfl                 | int                                                 | NULL     |
| c_custkey                   | int                                                 | NULL     |
| c_name                      | string                                              | NULL     |
| c_address                   | string                                              | NULL     |
| c_city                      | string                                              | NULL     |
| c_nation                    | string                                              | NULL     |
| c_region                    | string                                              | NULL     |
| c_phone                     | string                                              | NULL     |
| c_mktsegment                | string                                              | NULL     |
| p_partkey                   | int                                                 | NULL     |
| p_name                      | string                                              | NULL     |
| p_mfgr                      | string                                              | NULL     |
| p_category                  | string                                              | NULL     |
| p_brand                     | string                                              | NULL     |
| p_color                     | string                                              | NULL     |
| p_type                      | string                                              | NULL     |
| p_size                      | int                                                 | NULL     |
| p_container                 | string                                              | NULL     |
|                             |                                                     |          |
| # Detailed Table Information |                                                     |          |
| Database                    | ssb_10                                              |          |
| Table                       | lineorder_flatten                                   |          |
| Owner                       | hadoop                                              |          |
| Created Time                | Sat Dec 28 17:30:02 CST 2019                        |          |
| Last Access                 | Thu Jan 01 08:00:00 CST 1970                        |          |
| Created By                  | Spark 2.2 or prior                                  |          |
| Type                        | VIEW                                                |          |
| View Text                   | SELECT `lineorder`.`lo_orderkey`, `lineorder`.`lo_linenumber`, `lineorder`.`lo_custkey`, `lineorder`.`lo_partkey`, `
lineorder`.`lo_suppkey`, `lineorder`.`lo_orderdate`, `lineorder`.`lo_orderpriotity`, `lineorder`.`lo_shippriotity`, `lineorder`.`lo_quantity`, `lineor
der`.`lo_extendedprice`, `lineorder`.`lo_ordtotalprice`, `lineorder`.`lo_discount`, `lineorder`.`lo_revenue`, `lineorder`.`lo_supplycost`, `lineorder`
.`lo_tax`, `lineorder`.`lo_commitdate`, `lineorder`.`lo_shipmode`, `supplier`.`s_suppkey`, `supplier`.`s_name`, `supplier`.`s_address`, `supplier`.`s_
city`, `supplier`.`s_nation`, `supplier`.`s_region`, `supplier`.`s_phone`, `dates`.`d_datekey`, `dates`.`d_date`, `dates`.`d_dayofweek`, `dates`.`d_mo
nth`, `dates`.`d_year`, `dates`.`d_yearmonthnum`, `dates`.`d_yearmonth`, `dates`.`d_daynuminweek`, `dates`.`d_daynuminmonth`, `dates`.`d_daynuminyear`
, `dates`.`d_monthnuminyear`, `dates`.`d_weeknuminyear`, `dates`.`d_sellingseason`, `dates`.`d_lastdayinweekfl`, `dates`.`d_lastdayinmonthfl`, `dates`
.`d_holidayfl`, `dates`.`d_weekdayfl`, `customer`.`c_custkey`, `customer`.`c_name`, `customer`.`c_address`, `customer`.`c_city`, `customer`.`c_nation`
, `customer`.`c_region`, `customer`.`c_phone`, `customer`.`c_mktsegment`, `part`.`p_partkey`, `part`.`p_name`, `part`.`p_mfgr`, `part`.`p_category`, `
part`.`p_brand`, `part`.`p_color`, `part`.`p_type`, `part`.`p_size`, `part`.`p_container` FROM `ssb_10`.`lineorder`, `ssb_10`.`supplier`, `ssb_10`.`da
tes`, `ssb_10`.`customer`, `ssb_10`.`part`
WHERE `lineorder`.`lo_orderdate` = `dates`.`d_datekey` AND `lineorder`.`lo_custkey` = `customer`.`c_custkey` AND `lineorder`.`lo_suppkey` = `supplier`
.`s_suppkey` AND `lineorder`.`lo_partkey` = `part`.`p_partkey` |                                                     |          |
| Table Properties            | [transient_lastDdlTime=1577677599]                  |          |
| Serde Library               | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe  |          |
| InputFormat                 | org.apache.hadoop.mapred.TextInputFormat            |          |
| OutputFormat                | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat |          |
| Storage Properties          | [serialization.format=1]                            |          |
+-----------------------------+-----------------------------------------------------+----------+
73 rows selected (0.057 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

The following figure shows the execution plan of simple queries on the lineorder_flatten view.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(11) SortMergeJoin [lo_partkey#313], [p_partkey#359], Inner
   :- *(8) Sort [lo_partkey#313 ASC NULLS FIRST], false, 0
   :  +- Exchange hashpartitioning(lo_partkey#313, 200)
   :     +- *(7) SortMergeJoin [lo_custkey#312], [c_custkey#351], Inner
   :        :- *(4) Sort [lo_custkey#312 ASC NULLS FIRST], false, 0
   :        :  +- Exchange hashpartitioning(lo_custkey#312, 200)
   :        :     +- *(3) BroadcastHashJoin [lo_orderdate#315], [d_datekey#334], Inner, BuildRight
   :        :        :- *(3) BroadcastHashJoin [lo_suppkey#314], [s_suppkey#327], Inner, BuildRight
   :        :        :  :- *(3) Filter (((isnotnull(lo_suppkey#314) && isnotnull(lo_orderdate#315)) && isnotnull(lo_custkey#312)) && isnotnull(lo_part
key#313))
   :        :        :  :  +- Scan hive ssb_10.lineorder [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_ord
erdate#315, lo_orderpriotity#316, lo_shippriotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#32
2L, lo_supplycost#323L, lo_tax#324L, lo_commitdate#325, lo_shipmode#326], HiveTableRelation `ssb_10`.`lineorder`, org.apache.hadoop.hive.serde2.lazy.L
azySimpleSerDe, [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_orderdate#315, lo_orderpriotity#316, lo_ship
priotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#322L, lo_supplycost#323L, lo_tax#324L, lo_c
ommitdate#325, lo_shipmode#326]
   :        :        :  +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :        :     +- *(1) Filter isnotnull(s_suppkey#327)
   :        :        :        +- Scan hive ssb_10.supplier [s_suppkey#327, s_name#328, s_address#329, s_city#330, s_nation#331, s_region#332, s_phone#
333], HiveTableRelation `ssb_10`.`supplier`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [s_suppkey#327, s_name#328, s_address#329, s_city#330
, s_nation#331, s_region#332, s_phone#333]
   :        :        +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :           +- *(2) Filter isnotnull(d_datekey#334)
   :        :              +- Scan hive ssb_10.dates [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearm
onth#340, d_daynuminweek#341, d_daynuminmonth#342, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweek
fl#347, d_lastdayinmonthfl#348, d_holidayfl#349, d_weekdayfl#350], HiveTableRelation `ssb_10`.`dates`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSe
rDe, [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearmonth#340, d_daynuminweek#341, d_daynuminmonth#34
2, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweekfl#347, d_lastdayinmonthfl#348, d_holidayfl#349,
 d_weekdayfl#350]
   :        +- *(6) Sort [c_custkey#351 ASC NULLS FIRST], false, 0
   :           +- Exchange hashpartitioning(c_custkey#351, 200)
   :              +- *(5) Filter ((isnotnull(c_region#356) && (c_region#356 = ASIA)) && isnotnull(c_custkey#351))
   :                 +- Scan hive ssb_10.customer [c_custkey#351, c_name#352, c_address#353, c_city#354, c_nation#355, c_region#356, c_phone#357, c_mk
tsegment#358], HiveTableRelation `ssb_10`.`customer`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [c_custkey#351, c_name#352, c_address#353, c
_city#354, c_nation#355, c_region#356, c_phone#357, c_mktsegment#358]
   +- *(10) Sort [p_partkey#359 ASC NULLS FIRST], false, 0
      +- Exchange hashpartitioning(p_partkey#359, 200)
         +- *(9) Filter isnotnull(p_partkey#359)
            +- Scan hive ssb_10.part [p_partkey#359, p_name#360, p_mfgr#361, p_category#362, p_brand#363, p_color#364, p_type#365, p_size#366, p_conta
iner#367], HiveTableRelation `ssb_10`.`part`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [p_partkey#359, p_name#360, p_mfgr#361, p_category#3
62, p_brand#363, p_color#364, p_type#365, p_size#366, p_container#367] |
+----------------------------------------------------+
1 row selected (0.435 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

After a raw cache is created and built for the lineorder_flatten view, the same queries are performed on the view. EMR Spark automatically uses the cache to optimize the original execution plan. The following figure shows the new execution plan.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:17:47 INFO [main] HiveConf: Found configuration file file:/etc/ecm/spark-conf-2.4.3-hadoop2.8-1.4.2/hive-site.xml
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(1) Project [lo_orderkey#128L, lo_linenumber#129L, lo_custkey#130, lo_partkey#131, lo_suppkey#132, lo_orderdate#133, lo_orderpriotity#134, lo_ship
priotity#135, lo_quantity#136L, lo_extendedprice#137L, lo_ordtotalprice#138L, lo_discount#139L, lo_revenue#140L, lo_supplycost#141L, lo_tax#142L, lo_c
ommitdate#143, lo_shipmode#144, s_suppkey#145, s_name#146, s_address#147, s_city#148, s_nation#149, s_region#150, s_phone#151, ... 34 more fields]
   +- *(1) Filter (isnotnull(c_region#174) && (c_region#174 = ASIA))
      +- *(1) FileScan parquet [lo_orderkey#128L,lo_linenumber#129L,lo_custkey#130,lo_partkey#131,lo_suppkey#132,lo_orderpriotity#134,lo_shippriotity#
135,lo_quantity#136L,lo_extendedprice#137L,lo_ordtotalprice#138L,lo_discount#139L,lo_revenue#140L,lo_supplycost#141L,lo_tax#142L,lo_commitdate#143,lo_
shipmode#144,s_suppkey#145,s_name#146,s_address#147,s_city#148,s_nation#149,s_region#150,s_phone#151,d_datekey#152,... 34 more fields] Batched: true,
Format: Parquet, Location: InMemoryFileIndex[/user/hive/warehouse/ssb_10/_CACHE/lineorder_flatten/raw], PartitionCount: 2406, PartitionFilters: [], Pu
shedFilters: [IsNotNull(c_region), EqualTo(c_region,ASIA)], ReadSchema: struct<lo_orderkey:bigint,lo_linenumber:bigint,lo_custkey:int,lo_partkey:int,l
o_suppkey:int,lo_or... |
+----------------------------------------------------+
```

The new execution plan no longer uses the calculation logic of the lineorder_flatten view. Instead, it directly accesses the cached data in HDFS.

Usage notes

1. JindoCube does not ensure data consistency between the cache and source table. You must manually trigger data updates or configure a periodic update policy based on your requirements for data consistency.
2. JindoCube determines whether to use a cache to optimize execution plans based on the metadata of views. If cached data is incomplete when JindoCube optimizes an execution plan, the query results may be invalid or incomplete. The following operations or configurations may cause incomplete cached data: 1. Delete partition data that is required in subsequent queries on the cache details page. 2. Data required in subsequent queries is filtered out based on the filter conditions specified when you build or update the cache. 3. Required data has not been updated to the cache when the query is performed.

8.4.4. Use JindoTable to collect access frequency statistics on tables and partitions

You can use JindoTable to collect access frequency statistics on tables and partitions, and separate cold and hot data based on these statistics. This helps you reduce storage costs and improve cache usage efficiency.

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

JindoTable can be used to collect access frequency statistics on Hive tables. The collected data is saved in the namespaces of the SmartData service of a cluster.

In SmartData 3.2.X and later, JindoTable can be used to collect access frequency statistics on Spark, Hive, and Presto tables. This feature is enabled for Spark and Presto by default. If you want to disable the feature for Spark or Presto, perform the operations described in Disable the collection of access frequency statistics. The feature is disabled for Hive by default. If you want to enable the feature for Hive, perform the operations described in Enable the collection of access frequency statistics for Hive.

Query access frequency statistics

You can run a command to query access frequency statistics.

- Syntax

  ```
  jindo table -accessStat <-d [days]> <-n [topNums]>
  ```

  Set `days` and `topNums` to positive integers. For example, if days is set to 1 and topNums is not specified, the access frequency statistics of all the tables or partitions that are visited on the current day (from 00:00 to the current time) are queried.

- Description

  This command is used to query the first N tables or partitions that are visited the most frequently within a specific period of time, and the number of times each of them is visited.

- For example, you can run the following command to query the first 20 tables or partitions that have been visited the most frequently within the last seven days, and the number of times each of them has been visited:

  ```
  jindo table -accessStat -d 7 -n 20
  ```

For more information about how to use JindoTable, see Use JindoTable.

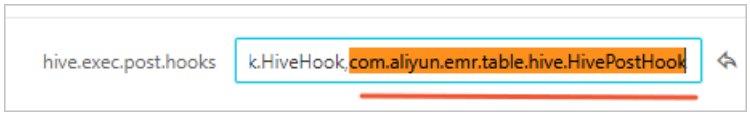
Enable the collection of access frequency statistics for Hive

1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.

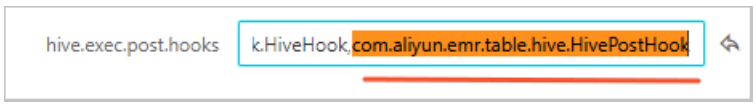
5. Modify the setting of a Hive parameter.
   i. In the left-side navigation pane, choose **Cluster Service > Hive**.
   ii. On the Hive service page, click the **Configure** tab.
   iii. Search for the hive.exec.post.hooks parameter and add **com.aliyun.emr.table.hive.HivePostHook** to the end of the existing parameter value. Separate the hooks with a comma (,).
6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   i. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
   ii. In the **Cluster Activities** dialog box, specify **Description**.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Disable the collection of access frequency statistics

1.
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify the setting of the related parameter.
   - Hive:
     a. In the left-side navigation pane, choose **Cluster Service > Hive**.
     b. On the Hive service page, click the **Configure** tab.
     c. Search for the hive.exec.post.hooks parameter and delete **com.aliyun.emr.table.hive.HivePostHook** from the parameter value.



   - Spark:
     a. In the left-side navigation pane, choose **Cluster Service > Spark**.
     b. On the Spark service page, click the **Configure** tab.
     c. Search for the spark.sql.queryExecutionListeners parameter and delete **com.aliyun.emr.table.spark.SparkSQLQueryListener** from the parameter value.



   - Presto:

a. In the left-side navigation pane, choose **Cluster Service > Presto**.
b. On the Presto service page, click the **Configure** tab.
c. Search for the event-listener.name parameter and delete the parameter value.

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify **Description** and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   - Hive:
     a. In the upper-right corner of the Hive service page, choose **Actions > Restart HiveServer2**.
     b. In the **Cluster Activities** dialog box, specify **Description**.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.
   - Spark:
     a. In the upper-right corner of the Spark service page, choose **Actions > Restart ThriftServer**.
     b. In the **Cluster Activities** dialog box, specify **Description**.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.
   - Presto:
     a. In the upper-right corner of the Presto service page, choose **Actions > Restart All Components**.
     b. In the **Cluster Activities** dialog box, specify **Description**.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.

8.5. Toolset

8.5.1. Use JindoFS FUSE

This topic describes how to access JindoFileSystem (JindoFS) from a Filesystem in Userspace (FUSE) client. FUSE allows you to use JindoFS in block storage mode and cache mode (only JFS Scheme supported).

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

FUSE is a method provided by the Linux kernel to mount a file system. You can use a FUSE client to map files in a JindoFS cluster to a local disk. Then, you can directly access data in the JindoFS cluster without the need to use the `hadoop fs -ls jfs://<namespace>/` command.

Mount a directory

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs
   ```

3. Run the following command to mount the directory:

   ```
   jindofs-fuse /mnt/jfs
   ```

   In this example, the */mnt/jfs/* directory is mounted.

Read and write files

1. Run the following command to query all the sub-directories under the /mnt/jfs/ directory:

   ```
   ls /mnt/jfs/
   ```

   All the namespaces that you have configured on the server are returned.

   ```
   test  testcache
   ```

2. Run the following command to query all the files in the test namespace:

   ```
   ls /mnt/jfs/test/
   ```

3. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs/test/dir1
   ls /mnt/jfs/test/
   ```

4. Run the following command to write data into a file:

   ```
   echo "hello world" > /tmp/hello.txt
   cp /tmp/hello.txt /mnt/jfs/test/dir1/
   ```

5. Run the following command to read data from a file:

   ```
   cat /mnt/jfs/test/dir1/hello.txt
   ```

   The following information is returned:

   ```
   hello world
   ```

The following example shows how to read data from and write data into files in Python:

1. Write data into a file in Python. Create the *write.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'w',encoding = 'utf-8') as f:
   f.write("my first file\n")
   f.write("This file\n\n")
   f.write("contains three lines\n")
```

2. Read data from a file in Python. Create the *read.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'r',encoding = 'utf-8') as f:
    lines = f.readlines()
    [print(x, end = '') for x in lines]
```

Run the following command to read the data that is written into the *test.txt* file:

```
[hadoop@emr-header-1 ~]$ ./read.py
```

The following information is returned:

```
my first file
This file
```

Unmount FUSE

Note Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to unmount FUSE:

```
umount jindofs-fuse
```

If the `target is busy` error is returned, switch to another directory, stop all the programs that are reading data from and writing data into FUSE files, and then unmount FUSE.

8.5.2. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain help information:

```
jindo distcp --help
```

The following information is returned:

```
     --help           - Print help text
     --src=VALUE          - Directory to copy files from
     --dest=VALUE              - Directory to copy files to
     --parallelism=VALUE         - Copy task parallelism
     --outputManifest=VALUE       - The name of the manifest file
     --previousManifest=VALUE   -   The path to an existing manifest file
     --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
if specified
     --copyFromManifest   -   Copy from a manifest instead of listing a directory
     --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
     --srcPattern=VALUE   -   Include only source files matching this pattern
     --deleteOnSuccess   -   Delete input files after a successful copy
     --outputCodec=VALUE   -   Compression codec for output files
     --groupBy=VALUE   -   Pattern to group input files by
     --targetSize=VALUE   -   Target size for output files
     --enableBalancePlan   -   Enable plan copy task to make balance
     --enableDynamicPlan   -   Enable plan copy task dynamically
     --enableTransaction   -   Enable transation on Job explicitly
     --diff   -   show the difference between src and dest filelist
     --ossKey=VALUE   -   Specify your oss key if needed
     --ossSecret=VALUE   -   Specify your oss secret if needed
     --ossEndPoint=VALUE   -   Specify your oss endPoint if needed
     --policy=VALUE   -   Specify your oss storage policy
     --cleanUpPending   -   clean up the incomplete upload when distcp job finish
     --queue=VALUE   -   Specify yarn queuename if needed
     --bandwidth=VALUE   -   Specify bandwidth per map/reduce in MB if needed
     --s3Key=VALUE   -   Specify your s3 key
     --s3Secret=VALUE   -   Specify your s3 Sercet
     --s3EndPoint=VALUE   -   Specify your s3 EndPoint
```

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-
01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03
/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03
/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017
-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-0
1/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-0
1/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_tab
le/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/
03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/
03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromMani
fest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

Note You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

> **Note** If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

--queue

--queue specifies the name of the YARN queue where the current DistCp task resides.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<your_bucket>/hourly_table --qu
eue yarnqueue
```

--bandwidth

--bandwidth specifies the bandwidth allocated to the current DistCp task. This prevents the task from using excessive bandwidth. Unit: MB/s.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.c
om --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy archive --parallelism 20
  ```

- Example of the IA storage class (`--ia`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy ia --parallelism 20
  ```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

Use Amazon S3 as a data source

You can use the --s3Key, --s3Secret, and --s3EndPoint parameters in a command to specify information related to Amazon S3.

Example:

```
jindo distcp jindo-distcp-2.7.3.jar --src s3a://yourbucket/ --dest oss://<your_bucket>/hour
ly_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com
```

You can configure the s3Key, s3Secret, and s3EndPoint parameters in the *core-site.xml* file of Hadoop. This way, you do not need to specify an AccessKey pair each time.

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

Example:

```
jindo distcp /tmp/jindo-distcp-2.7.3.jar --src s3://smartdata1/ --dest s3://smartdata1/tmp
--s3EndPoint  s3-us-west-1.amazonaws.com
```

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

> **Note** If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

8.5.3. Use Jindo DistCp in specific scenarios

This topic describes how to use Jindo DistCp in specific scenarios.

Prerequisites

- An E-MapReduce (EMR) cluster of a required version is created. For more information, see Create a cluster.
- Java Development Kit (JDK) 1.8 is installed.
- *jindo-distcp-<version>.jar* is downloaded based on your Hadoop version.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 2.7 and later 2.X versions.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 3.X.

Use scenarios

Common use scenarios of Jindo DistCp:

- Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?
- Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?
- Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?
- Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?
- Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file size?
- Scenario 8: What parameters are required if Amazon S3 is used as a data source?
- Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?
- Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?
- Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?
- Scenario 12: What parameters are required if I want to delete original files after a copy operation?
- Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?

If you are not using EMR, the following conditions must be met before you can import HDFS data to Object Storage Service (OSS):

- You can read data from HDFS.
- The AccessKey ID, AccessKey secret, and endpoint of OSS are obtained. You can write data to a destination OSS bucket.
- The storage class of the OSS bucket is not Archive.
- You can submit MapReduce tasks.
- The JAR package of Jindo DistCp is downloaded.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 10
```

> **Note** For more information about the parameters, see Use Jindo DistCp.

To import a large amount of data or a large number of files, such as millions or tens of millions of files, to OSS, you can set parallelism to a large value to increase the concurrency. You can also set `enableBatch` to true for optimization. Optimization command:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 500 --enableBatch
```

Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?

You can use one of the following methods to verify data integrity:

- Jindo DistCp Counters

  Check DistCp Counters in the counter information of a MapReduce task.

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

where:

- Bytes Destination Copied specifies the size of the file that you copy to a destination directory, in bytes.
- Bytes Source Read specifies the size of the file that you read from the source directory, in bytes.
- Files Copied specifies the number of copied files.

- Jindo DistCp --diff

  Use `--diff` to compare the information of files in the source and destination directories. The information includes file name and file size. Missing files and the files that fail to be copied are recorded in a manifest file. The manifest file is generated in the directory where you perform the comparison.

  Add `--diff` to the command described in Scenario 1. Example:

  ```
  hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
  -hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
  aliyuncs.com --diff
  ```

  If all files are copied, the following information is returned:

  ```
  INFO distcp.JindoDistCp: distcp has been done completely
  ```

Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?

If your DistCp task fails, and you want to perform resumable upload to copy only the files that fail to be copied, perform the following operations on the basis of the command described in Scenario 1:

1. Add `--diff` to check whether all files are copied.

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
   hou.aliyuncs.com --diff
   ```

   If all the files are copied, the following information is returned. Otherwise, a manifest file is generated. In this case, go to the next step.

   ```
   INFO distcp.JindoDistCp: distcp has been done completely.
   ```

2. Use `--copyFromManifest` and `--previousManifest` to copy the remaining files. Example:

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --dest oss://yang-hhht/hourly_table --previousManifest=file:///opt
   /manifest-2020-04-17.gz --copyFromManifest --parallelism 20
   ```

   `file:///opt/manifest-2020-04-17.gz` is the path where the manifest file is stored.

Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?

1. If no information is generated for the files that were copied last time, specify parameters to

generate a manifest file to record the information about the files that are copied.

Add `--outputManifest=manifest-2020-04-17.gz` and `--requirePreviousManifest=false` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

where:

- `--outputManifest` generates a manifest file whose name can be customized and must have a file name extension of gz, such as *manifest-2020-04-17.gz*. This file is stored in the directory specified by `--dest`.
- `--requirePreviousManifest` specifies whether to require a previous manifest file.

2. After a DistCp task is complete, copy the new files that may be generated in the source directory.

   Add `--outputManifest=manifest-2020-04-17.gz` and `--previousManifest=oss://yang-hhht/hourly_table/manifest-2020-04-17.gz` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

```
hadoop jar jindo-distcp-2.7.3.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --outputManifest=manifest-2020-04-18.gz --previousManifest=oss://yang-hhht
/hourly_table/manifest-2020-04-17.gz --parallelism 10
```

3. Repeat Step 2 to continuously copy incremental files.

Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?

Add the following two parameters to the command described in Scenario 1. These two parameters can be used together or separately.

- `--queue`: the name of the YARN queue.
- `--bandwidth`: the size of the specified bandwidth , in MB.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --queue yarnqueue --bandwidth 6 --parallelism 10
```

Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?

- To write data to OSS Archive storage, add `--archive` to the command described in Scenario 1.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --archive --parallelism 20
```

- To write data to OSS Infrequent Access (IA) storage, add `--ia` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --ia --parallelism 20
```

Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file sizes?

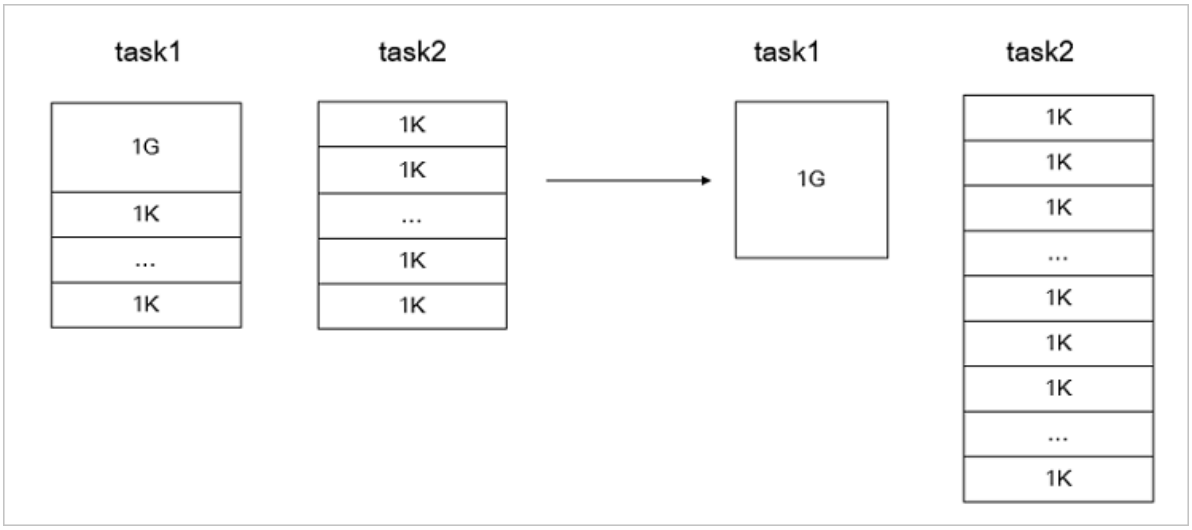
- A large number of small files and large data volume for a single large file

  If most of the files that you want to copy are small files, and the data volume of a single large file is large, the general solution is to randomly allocate the files to be copied. In this case, if you do not optimize the job allocation plan, few large files and many small files may be allocated to the same copy process. This cannot achieve excellent copy performance.

  Add `--enableDynamicPlan` to the command described in Scenario 1 to enable the optimization. --enableDynamicPlan cannot be used with `--enableBalancePlan`. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableDynamicPlan --parallelism 10
```

  The following figure shows the data copy before and after the plan is optimized.

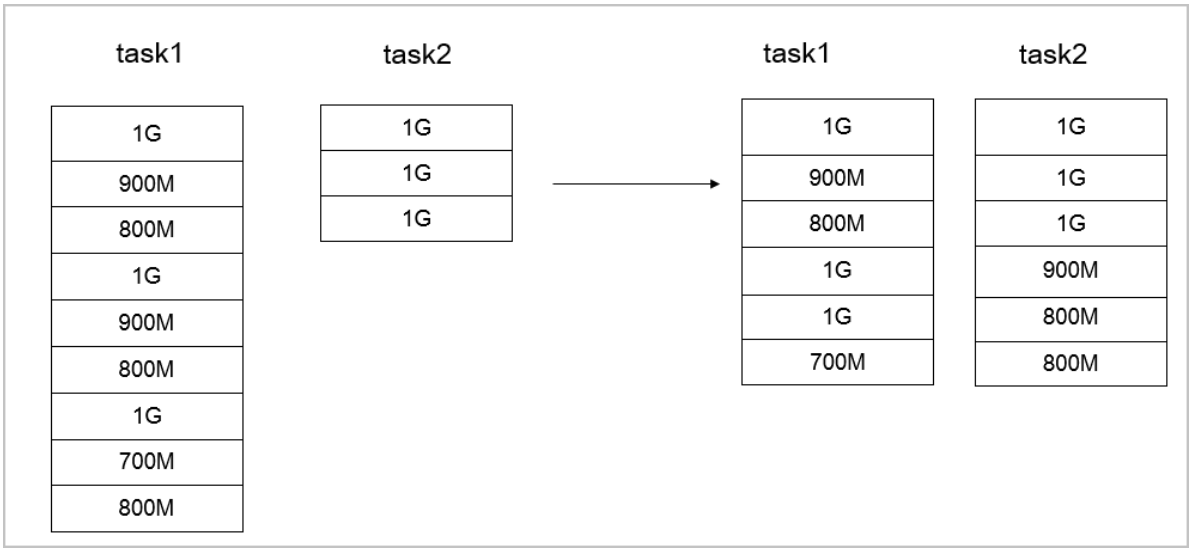


- No significant differences in sizes of files

  If the sizes are not significantly different among the files that you want to copy, use `--enableBalancePlan` to optimize the job allocation plan. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableBalancePlan --parallelism 10
```

The following figure shows the data copy before and after the plan is optimized.



Scenario 8: What parameters are required if Amazon S3 is used as a data source?

Replace the parameters that specify the AccessKey ID, AccessKey secret, and endpoint of OSS in the command described in Scenario 1 with the following parameters of Amazon S3:

- `--s3Key` : the AccessKey ID for Amazon S3
- `--s3Secret` : the AccessKey secret for Amazon S3
- `--s3EndPoint` : the endpoint for Amazon S3

Example:

```
hadoop jar jindo-distcp-<version>.jar --src s3a://yourbucket/ --dest oss://yang-hhht/hourly
_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com --para
llelism 10
```

Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?

You can use `--outputCodec` to compress copied files in a format such as LZO or GZ to reduce the space that is used to store the files.

Add `--outputCodec` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --outputCodec=gz --parallelism 10
```

You can set --outputCodec to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files are compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change to the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?

- If you want to copy the files that meet specific rules, add `--srcPattern` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPattern .*\.log --parallelism 10
```

  `--srcPattern`: specifies a regular expression that filters files for the copy operation.

- If you want to copy the files stored in some sub-directories of the same parent directory, add `--srcPrefixesFile` to the command described in Scenario 1.

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

  `--srcPrefixesFile`: enables Jindo DistCp to copy files in multiple folders under the same parent directory at a time.

  Content of the *folders.txt* file:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?

Add the following parameters to the command described in Scenario 1:

- `--targetSize`: the maximum size of the merged file, in MB
- `--groupBy`: the merging rule, which is a regular expression

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Scenario 12: What parameters are required if I want to delete original files after a copy operation?

Add `--deleteOnSuccess` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --deleteOnSuccess --parallelism 10
```

Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Jindo DistCp allows you to save the AccessKey ID, AccessKey secret, and endpoint of OSS in the *core-site.xml* file so that you do not need to repeatedly specify the information.

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of OSS, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.jfs.cache.oss-accessKeyId</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-accessKeySecret</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-endpoint</name>
        <value>oss-cn-xxx.aliyuncs.com</value>
    </property>
</configuration>
```

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of Amazon S3, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

8.5.4. Use tiered storage commands of JindoFS

In E-MapReduce (EMR) V3.30, JindoFS provides a tiered storage feature. This feature allows you to store cold and hot data in different storage media. This helps reduce the data storage costs or accelerate data access.

Use jindo jfs

Run the following command to obtain the help information:

```
jindo jfs -help archive
-archive -i/a <path> ... :
  Archive commands.
```

Tiered storage commands in JindoFS are asynchronous and start only related tasks.

Common commands:

- Cache
- Uncache
- Archive
- Unarchive
- Status
- ls2 command

Cache

You can use this command to back up data stored in a specific path to local disks. Then, you can read the data from local disks, without the need to read data from Object Storage Service (OSS).

```
jindo jfs -cache -p <path>
```

`-p` can be used to ensure that local data is not cleared based on disk usage.

Uncache

You can use this command to delete backup data from local disks and store data only to OSS Standard storage.

```
jindo jfs -uncache  <path>
```

Archive

You can use this command to delete backup data from local disks and store data to OSS in Infrequent Access (IA) or Archive mode. For information about the storage classes, see Overview.

```
jindo jfs -archive -i|-a <path>
```

`-i` is used to store data to OSS in IA mode. `-a` is used to store data to OSS in Archive mode.

Unarchive

You can use this command to restore data from the Archive mode to the IA or Standard mode. You can temporarily restore data stored in Archive mode to allow the data to be readable within one day.

```
jindo jfs -unarchive -i/-o <path>
```

`-i` is used to store data to OSS in IA mode. `-o` is used to temporarily restore data stored in Archive mode to allow the data to be readable.

Status

You can use this command to view task details. By default, the number of files to be subject to tiered storage in a specific directory and the data that has been subject to tiered storage are measured.

```
jindo jfs -status -detail/-sync <path>
```

`-detail` is used to view the storage progress of file data. `-sync` indicates that the command exits after a tiered storage task is completed.

ls2 command

JindoFS provides the ls2 command that allows you to view the file storage status on the basis of Hadoop ls commands:

```
hadoop fs -ls2 <path>
```

Example of command output, which includes the file storage class:

```
drwxrwxrwx  - -         0    2020-06-05 04:27 oss://xxxx/warehouse
-rw-rw-rw-  1 Archive   1484 2020-09-23 16:40 oss://xxxx/wikipedia_data.csv
-rw-rw-rw-  1 Standard  1676 2020-06-07 20:04 oss://xxxx/wikipedia_data.json
```

9.SmartData 3.1.x

9.1. SmartData 3.1.x

SmartData is a storage service for the E-MapReduce (EMR) Jindo engine. SmartData provides centralized storage, caching, and computing optimization for EMR computing engines and extends storage features. SmartData is composed of JindoFS, JindoTable, and related tools. This topic describes the updates in SmartData 3.1.x.

Background information

Limits on SmartData 3.1.0:

- You can turn on the meta-cache switch to enable the cache mode of JindoFS to cache metadata. However, we recommend that you enable the cache mode only in training scenarios. If you use the cache mode in analysis scenarios, inappropriate parameter configurations may cause data synchronization failures between the current path and other OSS paths.
- The name of a namespace in JindoFS can contain only letters, digits, and hyphens (-).
- The size of a large file that you want to copy by using Jindo DistCp cannot exceed 78 GB.
- Although JindoFS in block storage mode supports the checksum feature, Jindo DistCp does not support this feature.

Documentation updates

Compared with SmartData 3.0.0, the following updates are made to the documentation of SmartData 3.1.0:

- Use a credential provider
- Data management policies
- Use JindoTable to collect access frequency statistics of tables and partitions

Feature updates

- JindoFS-based storage optimization
- JindoFS-based caching optimization
- JindoTable-based computing optimization
- JindoManager-based system management
- Optimization based on Jindo tools
- Ecosystem support for JindoFS

JindoFS-based storage optimization

- JindoFS can perform file checksums by using the MD5MD5CRC or COMPOSITE_CRC algorithm. The checksum interfaces of open source HDFS are inherited. If the MD5MD5CRC algorithm is used, JindoFS provides an extended interface that supports the input of block size and also provides related Shell commands. This facilitates file comparison between JindoFS and HDFS.
- JindoFS supports transparent file compression. It allows you to specify compression policies for directories, compress data blocks of files that are newly written to the directories, and then store them in OSS. This feature significantly reduces the storage space and read/write workloads for data that has a high compression ratio.

- JindoFS supports flush semantics for data writes. After you call the flush API to flush data in a file, the data is persisted to the current location and is readable.
- The following issue is resolved: If a file is stored in a deep directory that involves multiple levels of directories, the `hadoop fs -ls -R` command cannot be executed because the thread is in the waiting state.
- The `hadoop fs -stat` command is enhanced. More information, such as atime and privilege, is returned.
- The HDFS client paths for the Jindo system can be modified. This reduces the workload needed to modify paths when you migrate cluster data.

JindoFS-based caching optimization

- JindoFS optimizes the caching of a large number of small files in machine learning training scenarios and improves the small-file caching efficiency and read performance.
- A `cache` command can be executed to preload the directories of small files. This improves the preloading efficiency.
- Data caching can be automatically triggered. You can specify the directory whose status you want to track and configure a time interval for directory check. This way, the system checks the directory at the specified interval and triggers data caching if any new file is found.

JindoTable-based computing optimization

- JindoTable Dump TF supports two-dimensional arrays.
- The Jindo mc dump command supports the GZIP compression format. The `-c` parameter can be used in the command.

JindoManager-based system management

JindoManager is added to manage the Jindo system, such as to perform O&M and monitor service status. JindoManager provides a web UI where you can view the status of each Jindo service.

Optimization based on Jindo tools

- Jindo DistCp optimizes the Job Committer logic for processing small files. This reduces the number of requests sent to OSS and improves the performance of DistCp when a large number of small files exist.
- Jindo DistCp optimizes the policy of dividing files into batches and improves the overall copy performance.

Ecosystem support for JindoFS

- You can run Flink streaming jobs to write data into JindoFS and store the data in block storage or cache mode. Flink streaming jobs can automatically recover from failures. The exactly_once semantics can be implemented in the jobs based on the combination with data sources that support data retransmission, such as Kafka.
- Flink supports the entropy injection feature. If you run Flink streaming jobs to write data into OSS or JindoFS in block storage or cache mode, you can use the entropy injection feature to replace a specific part of the destination path with a random string. This improves data write efficiency.
- The JindoFS TensorFlow connector is provided to support TensorFlow Filesystem. The native I/O interfaces are used. The following versions are supported: TensorFlow 1.15 and later 1.x versions, and 2.x versions later than 2.3.

9.2. JindoFS in block storage mode

9.2.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read and write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

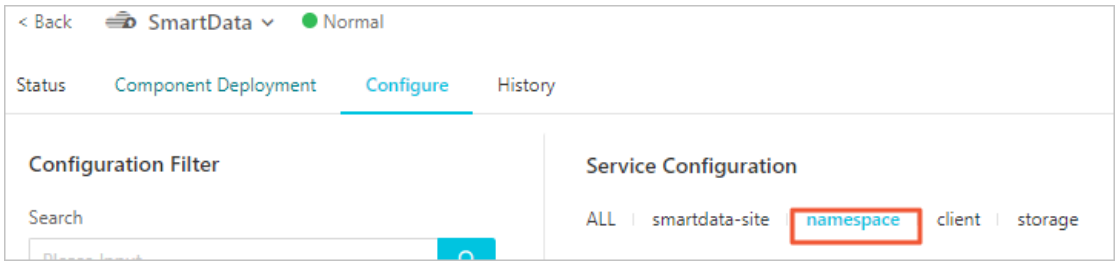
Context

The block storage mode of JindoFS has the following features:

- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.
- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

i. Set **jfs.namespaces** to **test**.

If you configure multiple namespaces, separate them with commas (,).

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.

5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

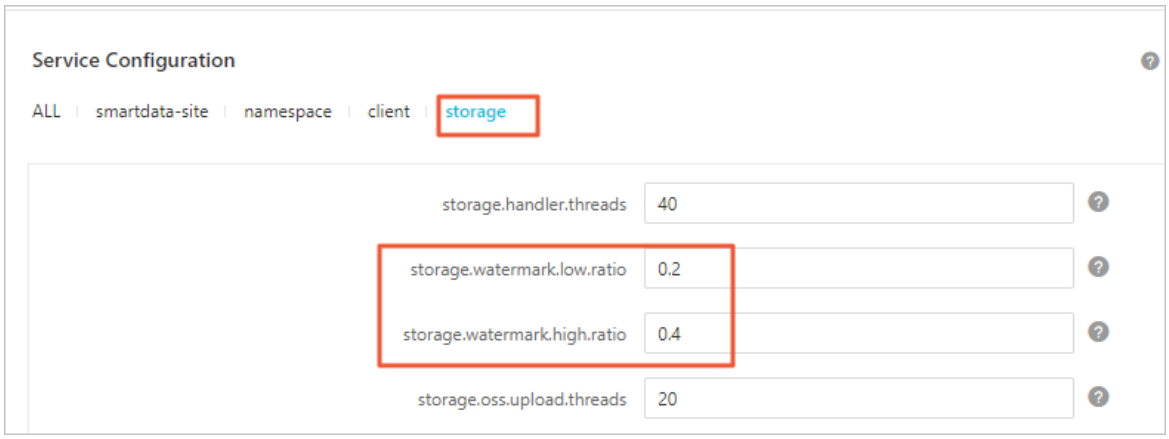
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.

i. Choose **Actions > Restart Jindo Storage Service** in the upper-right corner.
ii. In the **Cluster Activities** dialog box, specify the related parameters.
iii. Click **OK**.
iv. In the **Confirm** message, click **OK**.

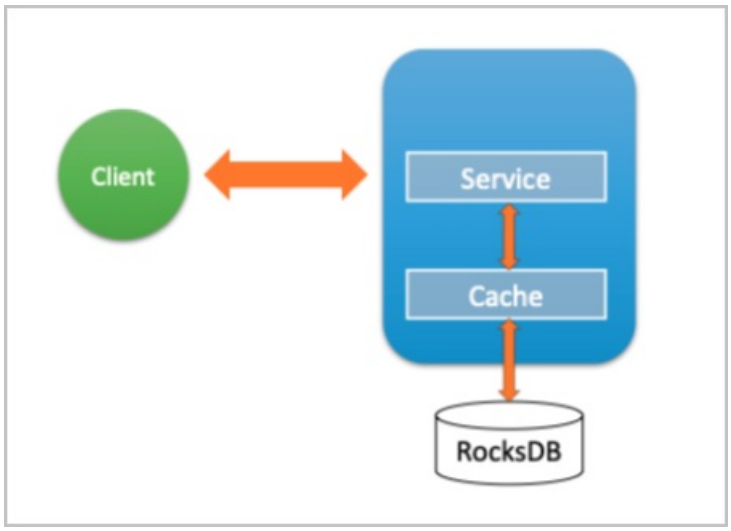
9.2.2. Use RocksDB to store metadata

Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage backend in high availability mode, we recommend that you use Raft instances. For more information, see Use Raft-RocksDB-Tablestore to store metadata.

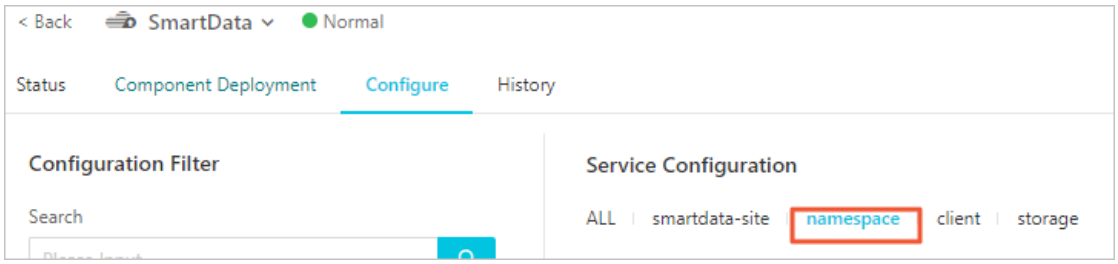
The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.

ii. Click the **namespace** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
6. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

7. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
8. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

   ```
   hadoop fs -count jfs://test/
   ```

   Information similar to the following example is returned:

   ```
    1596      1482809                 25 jfs://test/
   ```

   The number of folders is 1596, and the number of files is 1,482,809.

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1`, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus
```

Information similar to that shown in the following figure is returned.

```
[hadoop@emr-header-1 ~]$ jindo jfs -metaStatus
==== emr-header-1:8101 ====
OtsUploader: _synced=1
[RocksDB Row Counts of each CF(Table)]
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

   iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus
```

If the state is FINISH, the recovery is completed, as shown in the following figure.

```
==== emr-header-1:8101 ====
[Recovery From OTS Status]
  state: FINISH
  total 22855 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

9.2.3. Use Raft-RocksDB-Tablestore to store metadata

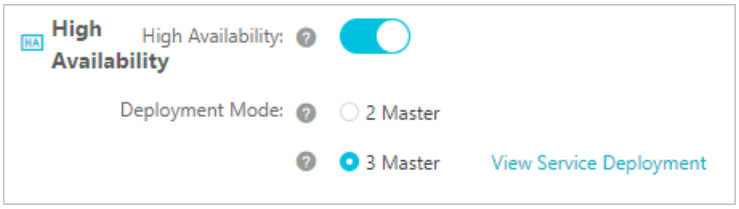
In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata.

Prerequisites

- A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  > **Note** You must enable the transaction feature.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.

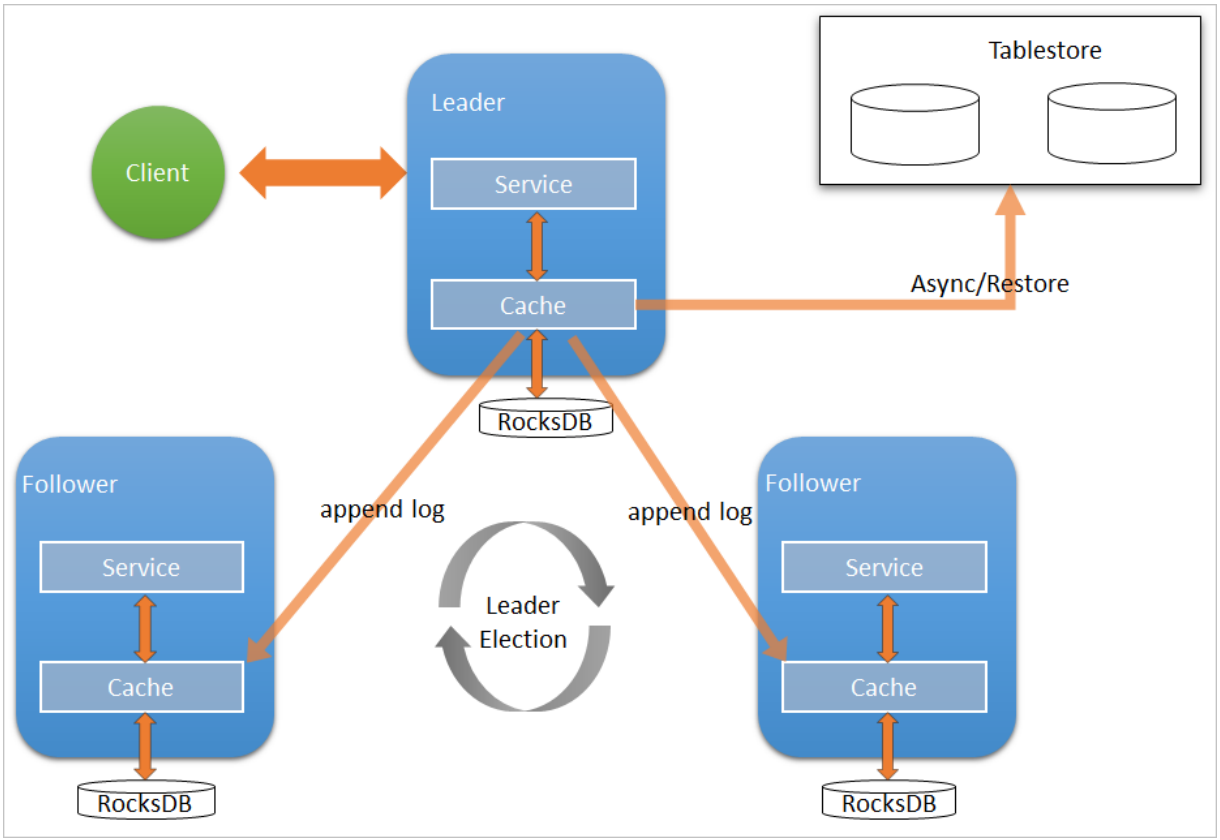


  > **Note** If no deployment modes are available, submit a ticket.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.



Configure a Raft instance as the local storage backend

1. Suspend all the SmartData components of an EMR cluster.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   vi. Select **Stop All Components** from the **Actions** drop-down list in the upper-right corner.
2. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
3. Go to the **namespace** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **namespace** tab.
4. Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. Valid values: ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

Note If you do not need to use Tablestore for remote storage, skip Step 5.

5. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

```
hadoop fs -count jfs://test/
      1596      1482809                 25 jfs://test/
  (Number of folders) (Number of files)
```

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```

```
[RaftPeerImpl]
peer id: [redacted]
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 2335ms)
snapshot_timer: timeout(3600000ms) SCHEDULING(in 150305ms)
storage: [1, 624625]
disk_index: 624625
known_applied_index: 624625
last_log_id: (index=624625,term=2)
first_index_pinned: 624625
state_machine: Idle
last_committed_index: 624625
last_snapshot_index: 0
last_snapshot_term: 0
snapshot_status: IDLE
replicator_25769803789@[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=624261 ic=0
replicator_329853488332[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=623564 ic=0

OtsUploader: _lastStopIndex=624624, _synced=1
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.

```
[RaftPeerImpl]
peer_id: [redacted]:8103:0
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 3382ms)
snapshot_timer: timeout(600000ms) SCHEDULING(in 474855ms)
storage: [1, 153]
disk_index: 153
known_applied_index: 153
last_log_id: (index=153,term=2)
first_index_pinned: 1
state_machine: Idle
last_committed_index: 153
last_snapshot_index: 1
last_snapshot_term: 2
snapshot_status: IDLE
replicator_1116691496965@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0
replicator_3311419785217@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0

[Recovery From OTS Status]
state: FINISH
total rows: 1484409
table `jfs_block_test` 2 rows.
table `jfs_namespace_cache_ns` 1 rows.
table `jfs_namespace_test` 1484406 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

9.2.4. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see

Create a cluster.

- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.

ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

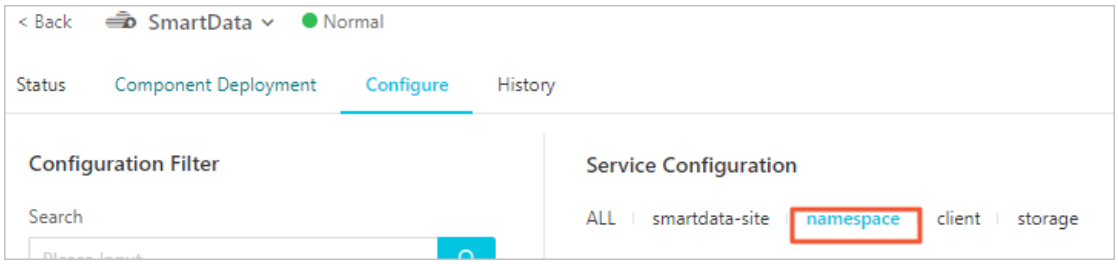
iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **namespace** tab for the SmartData service.

   i. Click the **Configure** tab.

   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:

   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

9.2.5. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

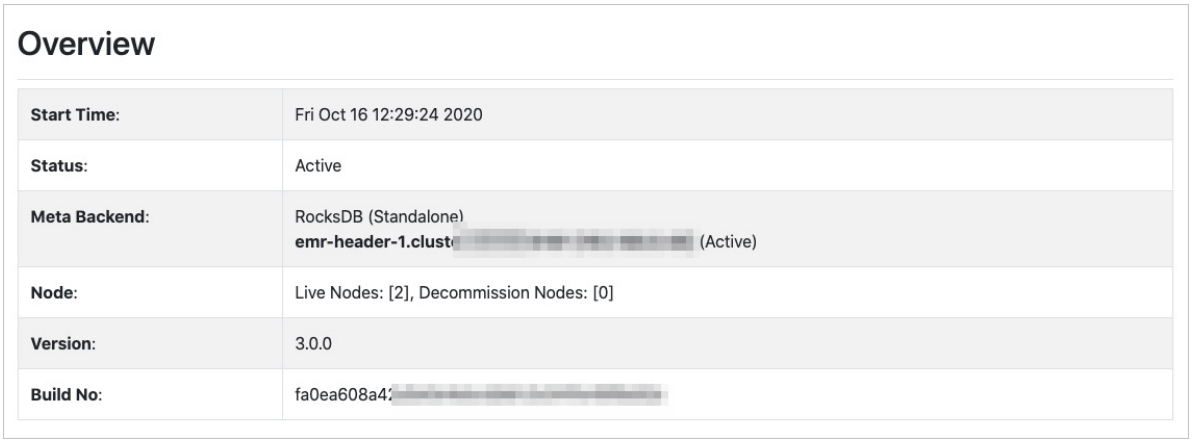
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

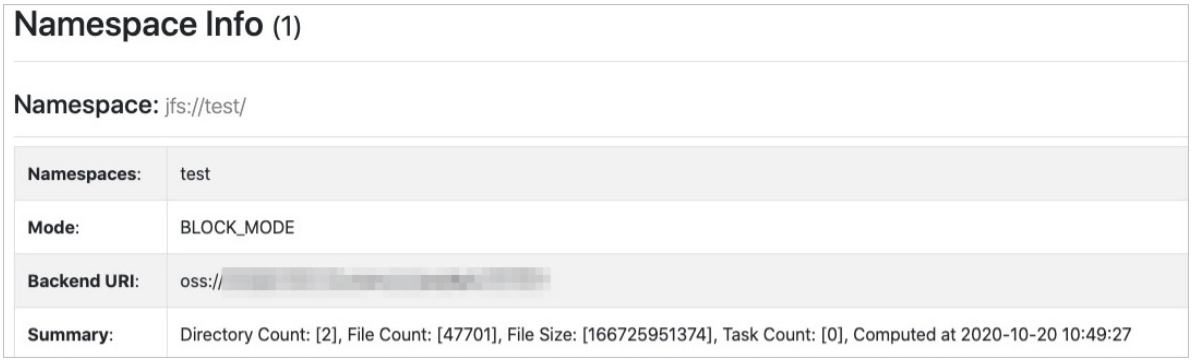


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



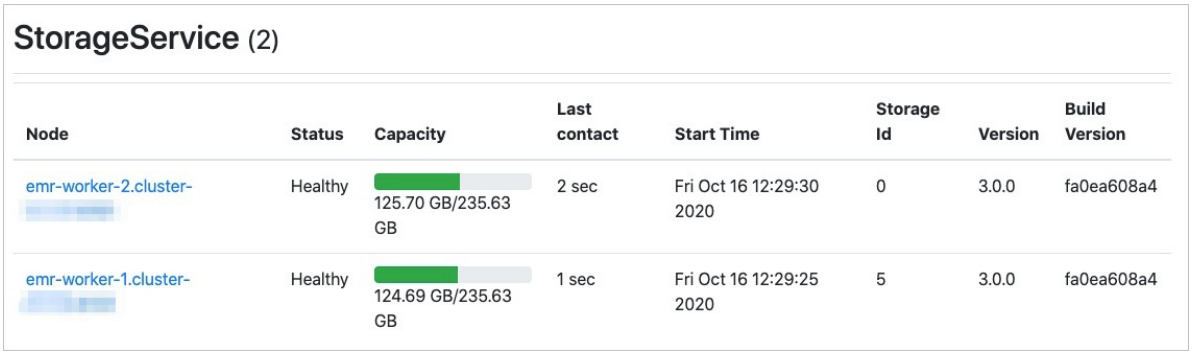
Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

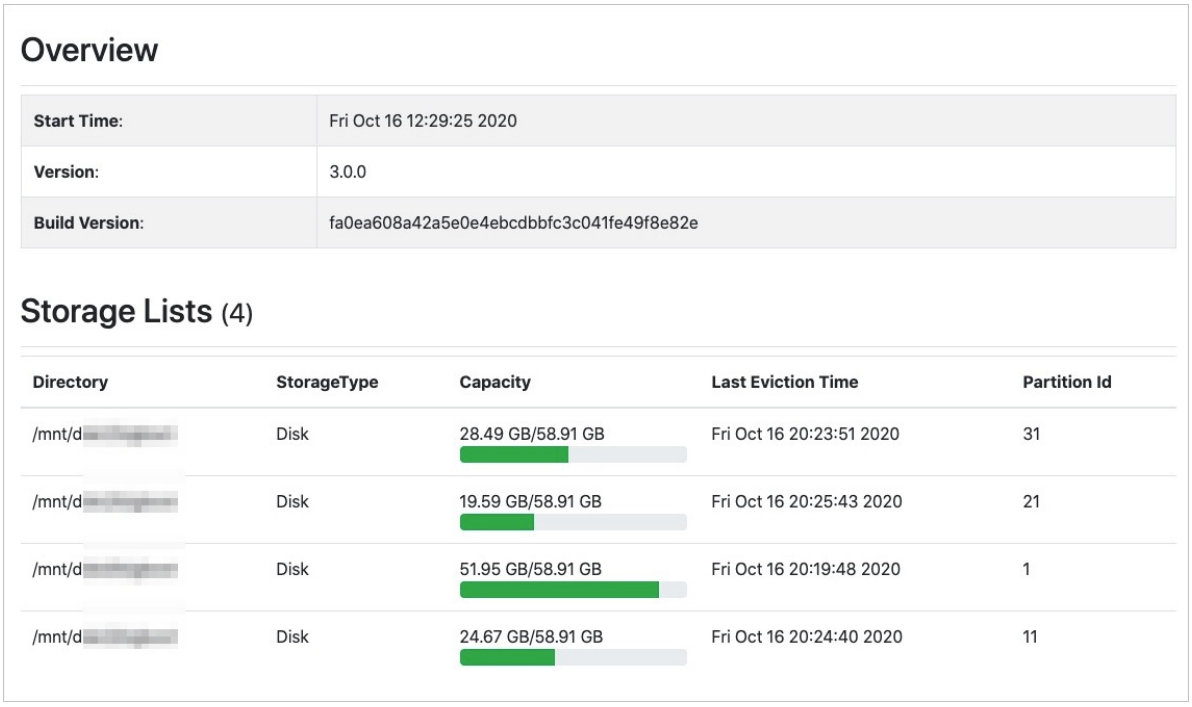
  This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.



StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.



Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

9.2.6. Manage JindoFS permissions

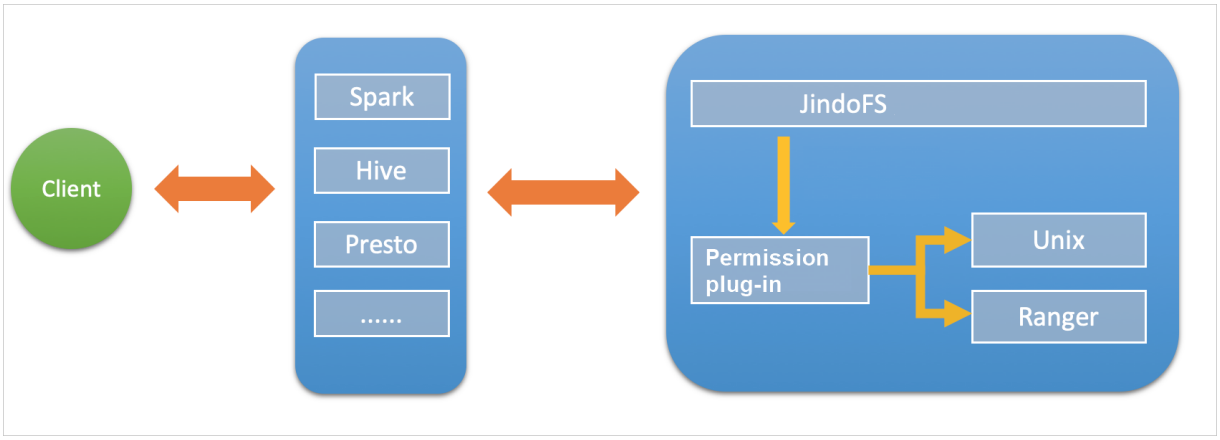
This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

Permission management based on the storage mode:

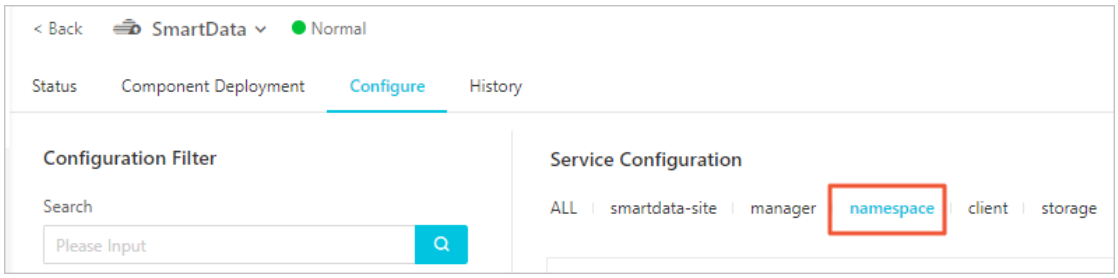
- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.
  - Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

9.2.7. Data management policies

JindoFS in block storage mode provides advanced data management policies to meet storage requirements in different scenarios. The advanced policies include storage policies and compression policies. This topic describes the policies in detail and provides examples to show how to use the policies.

Storage policies

JindoFS provides the five storage policies described in the following table to support flexible storage.

Policy	Description
AR	Data has only one backup. The backup is stored as an Archive object in OSS.
IA	Data has only one backup. The backup is stored as an Infrequent Access (IA) object in OSS.

Policy	Description
COLD	Data has only one backup. The backup is stored as a Standard object in OSS.
WARM	Data has one backup in OSS and one local backup. The local backup can effectively accelerate data read operations. By default, this storage policy is used.
HOT	Data has one backup in OSS and one local backup. The local backup is forcibly locked. This way, it is not deleted when the cache is automatically cleared. This policy can achieve a better acceleration effect than the WARM policy and is suitable for the hottest data.

For more information about the storage classes of OSS, see Overview.

A newly added file is stored based on the storage policy specified for its parent directory. Example:

- Run the following command to configure a storage policy:

```
jindo jfs -setStoragePolicy [-R] <StoragePolicy>(AR/IA/COLD/WARM/HOT) <path> ...
```

  where:

  - `[-R]`: specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.
  - `<path>`: the name of the directory for which the storage policy is configured.

- Run the following command to obtain the storage policy configured for a directory:

```
jindo jfs -getStoragePolicy <path>
```

Compression policies

JindoFS allows you to compress data blocks before you store them. This feature effectively reduces the storage space occupied by data blocks and improves data read and write efficiency. This feature is suitable for files that have a high compression ratio. The following table describes the supported compression policies.

Policy	Description
NONE	Data blocks are not compressed. By default, this compression policy is used.
ZSTD	The Zstandard compression algorithm is used to compress data blocks.

A newly added file is compressed based on the compression policy specified for its parent directory before it is stored. Example:

- Run the following command to configure a compression policy:

```
jindo jfs -setCompressionPolicy [-R] <CompressionPolicy>(NONE/ZSTD) <path> ...
```

where:

- `[-R]`: specifies that a recursive operation is performed to configure the same compression policy for all sub-directories of the directory.
- `<path>`: the name of the directory for which the compression policy is configured.

- Run the following command to obtain the compression policy configured for a directory:

```
jindo jfs -getCompressionPolicy <path> ...
```

9.2.8. Analyze metadata offline

In E-MapReduce (EMR) V3.30.0 or later, JindoFS in block storage mode allows you to dump the metadata of an entire namespace to Object Storage Service (OSS) and use Jindo SQL to analyze the metadata.

Context

In Hadoop Distributed File System (HDFS), the metadata is stored in a snapshot file named fsimage. The snapshot file contains the following information about metadata: namespace, file that stores the metadata, blocks, and file system quota. In HDFS, you can run a command to download the fsimage file in the XML format to your on-premises machine and view this file to analyze metadata offline. JindoFS does not require you to download metadata to your on-premises machine.

Upload HDFS metadata to OSS

Run the following command to upload metadata of a specific namespace to OSS:

```
jindo jfs -dumpMetadata <nsName>
```

`<nsName>` indicates the name of the namespace in block storage mode.

For example, upload the metadata of the test-block namespace to OSS and analyze the metadata offline.

```
jindo jfs -dumpMetadata test-block
```

If the following information appears, data is uploaded to OSS and stored as a file in the JSON format.

```
Sucessfully upload namespace metadata to OSS.
```

Metadata upload path

The metadata upload path is the metadataDump sub-directory under *sysinfo* in JindoFS.

For example, if `namespace.sysinfo.oss.uri` is set to `oss://abc/`, metadata is uploaded to the `oss://abc/metadataDump` directory.

Parameter	Description
namespace.sysinfo.oss.uri	The path of the OSS bucket.
namespace.sysinfo.oss.endpoint	The endpoint. An endpoint specified in a different region can also be used.
namespace.sysinfo.oss.access.key	The AccessKey ID of your Alibaba Cloud account.
namespace.sysinfo.oss.access.secret	The AccessKey secret of your Alibaba Cloud account.

Batch information: Metadata from HDFS changes based on the usage. Each time you want to analyze metadata, you must run commands to view the snapshot file of the metadata. Each time you run a Jindo command to upload metadata, a batch number that is generated based on the upload time is used as the root directory to store the metadata. This ensures that the data you uploaded each time is not overwritten. You can delete historical data based on your requirements.

- 1 indicates the configuration path of system information in OSS.
- 2 indicates a namespace.
- 3 indicates a batch number.

Metadata schema

HDFS metadata that is uploaded to OSS is stored as a file in the JSON format. Schema information:

```
{
  "type":"string",          /*Inode type, FILE DIRECTORY*/
  "id": "string",            /*INode id*/
  "parentId" :"string",         /*Parent node ID*/
  "name":"string",         /*Inode name*/
  "size": "int",         /*Inode size, BIGINT*/
  "permission":"int",         /*Permission, INT*/
  "owner":"string",          /*Owner name*/
  "ownerGroup":"string",     /*Owner group name*/
  "mtime":"int",              /*Inode modification time, BIGINT*/
  "atime":"int",              /*The time when the inode was last accessed, BIGINT*/
  "attributes":"string",       /*File-related attributes*/
  "state":"string",            /*Inode status*/
  "storagePolicy":"string",    /*Storage policy*/
  "etag":"string"           /*etag*/
}
```

Use Jindo SQL to analyze metadata

1. Run the command shown in the following figure to start Jindo SQL.

```
[root@emr-header-1 ~]# jindo sql
Spark master: yarn, Application Id: application_1603081647416_0050
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.33 seconds, Fetched 3 row(s)
```

2. Query the tables whose data can be analyzed by using Jindo SQL.
   - Run the `show tables` command to view the tables whose data can be analyzed. Jindo SQL has two built-in modules audit_log and fs_image for audit and metadata analysis.
   - Run the `show partitions fs_image` command to view the partition information of the fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

     The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

3. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287249    1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       11922762023479287307    1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       15464684820176650002    1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The namespace and datetime columns are added and indicate the namespace name and the timestamp when metadata is uploaded.

Example: Obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

Use Hive to analyze metadata

1. Create a table schema in Hive.

   You can use the following command to create a table schema of the metadata that is used for queries in Hive:

```
CREATE EXTERNAL TABLE `table_name`
(`type` string,
 `id` string,
 `parentId` string,
 `name` string,
 `size` bigint,
 `permission` int,
 `owner` string,
 `ownerGroup` string,
 `mtime` bigint,
 `atime` bigint,
 `attr` string,
 `state` string,
 `storagePolicy` string,
 `etag` string)
 ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
 STORED AS TEXTFILE
 LOCATION 'The OSS path to which file data is uploaded';
```

2. Use Hive to analyze data offline.

   After you create a Hive table, you can use Hive SQL to analyze metadata.

```
select * from table_name limit 200;
```

   The following figure shows an example.

```
hive> select * from inode_metadata_test0 limit 100;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_202009[redacted]
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1[redacted], Tracking URL = http://emr-heade[redacted]:20888/proxy/applica[redacted]
Kill Command = /usr/lib/hadoop-current/bin/hadoop job  -kill job_1599[redacted]
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-09-08 14:57:26,112 Stage-1 map = 0%,  reduce = 0%
2020-09-08 14:57:31,263 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.22 sec
MapReduce Total cumulative CPU time: 1 seconds 220 msec
Ended Job = job_[redacted]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.22 sec   HDFS Read: 6867 HDFS Write: 1524 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 220 msec
OK
Directory       1127433438684721971 4    1127433438684721971 3    /uttest/oss     0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Directory       1127433438684721971 9    1127433438684721971 3    /uttest/oss2    0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
Directory       1127433438684721971 6    1127433438684721971 4    /uttest/oss/dir 0       511     caojie  staff   1599545017636   1599545017636           Finalized       WARM
File    1127433438684721971 5    1127433438684721971 4    /uttest/oss/file1       0       420     caojie  staff   1599545017632   1599545017632           Finalized       WARM
File    1127433438684721971 7    1127433438684721971 6    /uttest/oss/dir/file2   0       420     caojie  staff   1599545017642   1599545017642           Finalized       WARM
File    1127433438684721971 8    1127433438684721971 6    /uttest/oss/dir/file3   0       420     caojie  staff   1599545017651   1599545017651           Finalized       WARM
Directory       1127433438684721972 0    1127433438684721971 9    /uttest/oss2/dir        0       511     caojie  staff   1599545017654   1599545017654           Finalized       WARM
File    1127433438684721972 1    1127433438684721972 0    /uttest/oss2/dir/file2  0       420     caojie  staff   1599545017658   1599545017658           Finalized       WARM
File    1127433438684721972 2    1127433438684721972 0    /uttest/oss2/dir/file3  0       420     caojie  staff   1599545017666   1599545017666           Finalized       WARM
Directory       1127433438684721971 3    1767282355743385190 5    /uttest 0       511     caojie  staff   1599545017615   1599545017615           Finalized       WARM
Time taken: 10.734 seconds, Fetched: 10 row(s)
hive>
```

9.3. JindoFS in cache mode

9.3.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in Object Storage Service (OSS) based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

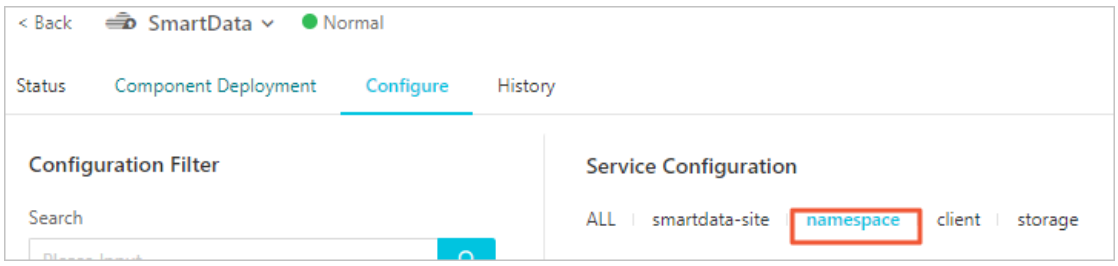
  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

   ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. Click **OK**.
5. Save the configurations.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

   iii. Click **OK**.

6. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **client** tab.
2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   You do not need to restart the SmartData service to apply the new configurations.

3. Save the configurations.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

   iii. Click **OK**.

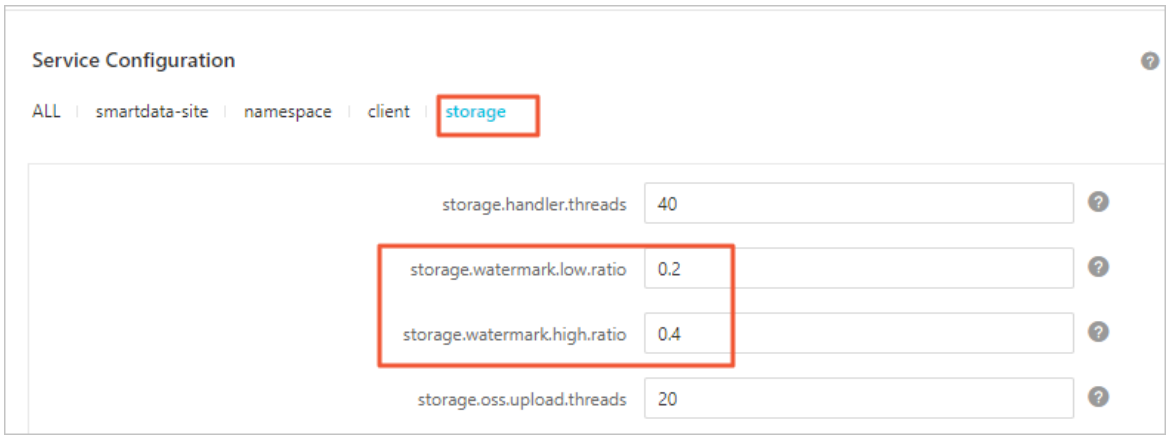
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you specify. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right

corner.

ii. In the **Cluster Activities** dialog box, specify the related parameters.

iii. Click **OK**.

iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to specify an AccessKey pair. In other cases, you must specify an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you choose for accessing files in OSS:

- OSS Scheme

  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.

  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss.accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.endpoint	The endpoint of the OSS bucket that serves as the storage backend.

> **Note** You can set these parameters to the values of the parameters in an EMR version earlier than EMR V3.30.0.

- JFS Scheme

  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **namespace** tab.

  ii. Set **jfs.namespaces** to **test**.

  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.

Parameter	Description
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize the cache performance. After you configure the parameters, you do not need to restart the SmartData service to apply the new configurations.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. You can use this parameter to prevent upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If an excessive amount of memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

9.3.2. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

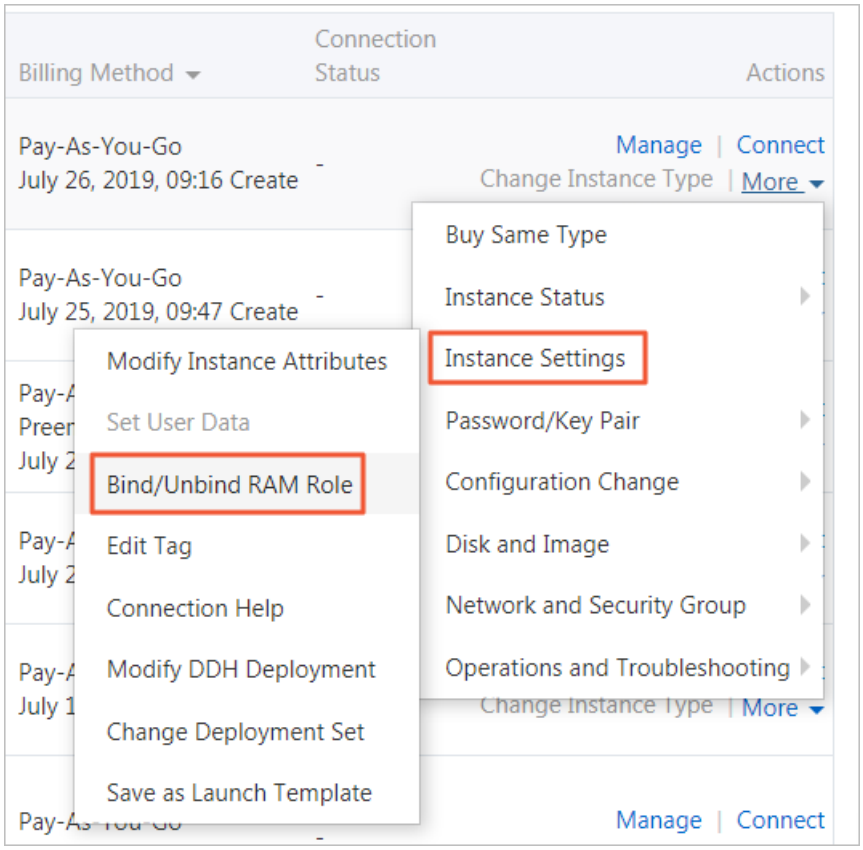
1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.
4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

   ```
   hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
   ```

2. Access OSS by using the FileSystem interface of Hadoop.

   Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

9.3.3. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false

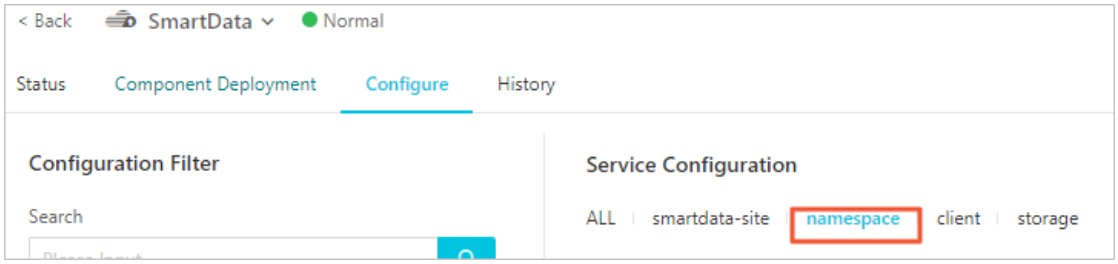
Parameter	Description
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-

right corner.

ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

For more information, see Configure lifecycle rules.

v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

9.3.4. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

> **Notice**
>
> - The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
> - After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
> - Hive and Presto jobs do not use Hadoop Job Committer.
> - In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.

i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
ii. Click the **Configure** tab.
iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.
6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.

3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

   The default value is 8.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

9.3.5. Use a credential provider

You can use a credential provider to save an encrypted AccessKey pair into a file. This prevents the leak of the AccessKey pair.

Context

You can use a Hadoop credential provider to save an encrypted AccessKey pair into a file. This prevents the issue that the AccessKey pair is transmitted in plaintext. You can select an appropriate JindoOSS credential provider based on your business requirements.

Configure a JindoOSS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **smartdata-site** tab.

   i. Click the **Configure** tab.

   ii. In the **Service Configuration** section, click the **smartdata-site** tab.

3. Add configuration information.

   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

   - Global configuration (for all buckets)

Parameter	Description
fs.jfs.cache.oss.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types.

- Bucket-level configuration

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	Specifies the implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). The system reads credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider,com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider`. For more information about credential providers, see Credential provider types. **Note** XXX indicates the name of an Object Storage Service (OSS) bucket.

Credential provider types

You can select different credential providers based on your business requirements. The following providers are supported:

- Global configuration
  - TemporaryAliyunCredentialsProvider

    This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.cache.oss.securityToken	The temporary security token used to access OSS.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access OSS. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

- Bucket-level configuration

- TemporaryAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair and a security token that have a validity period are used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.
fs.jfs.cache.oss.bucket.XXX.securityToken	The temporary security token used to access the OSS bucket.

- SimpleAliyunCredentialsProvider

  This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.bucket.XXX.accessKeyId	The AccessKey ID used to access an OSS bucket.
fs.jfs.cache.oss.bucket.XXX.accessKeySecret	The AccessKey secret used to access the OSS bucket.

- EnvironmentVariableCredentialsProvider

  To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access an OSS bucket.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access the OSS bucket.
ALIYUN_SECURITY_TOKEN	The temporary security token used to access the OSS bucket. **Note** This parameter is required only when you configure a token that has a validity period.

- InstanceProfileCredentialsProvider

  This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.oss.bucket.XXX.credentials.provider	**com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider**

Use a Hadoop credential provider to store AccessKey pair information

> **Note** For more information about Hadoop credential providers, see CredentialProvider API Guide.

Use a command that is provided by Hadoop to store AccessKey pair and security token information into a credential file. Syntax:

```
hadoop credential <subcommand> [options]
```

For example, in global configuration mode, store AccessKey pair and token information into a JCEKS file. You can protect the file by using file permissions or you can specify a password to encrypt the information you want to store. If you do not specify a password, the default string is used for encryption.

```
hadoop credential create fs.jfs.cache.oss.accessKeyId -value AAA -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.accessKeySecret -value BBB -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.securityToken -value  CCC -provider jceks://file/root/oss.jceks
```

After a credential file is generated, you must configure the parameter described in the following table to specify the provider type and location.

Parameter	Description
fs.jfs.cache.oss.security.credential.provider.path	The path used to store the credential file that stores AccessKey pair information. For example, you can set this parameter to *jceks://file/${user.home}/oss.jceks*, which indicates that the *oss.jceks* file is stored in the home directory.

9.3.6. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

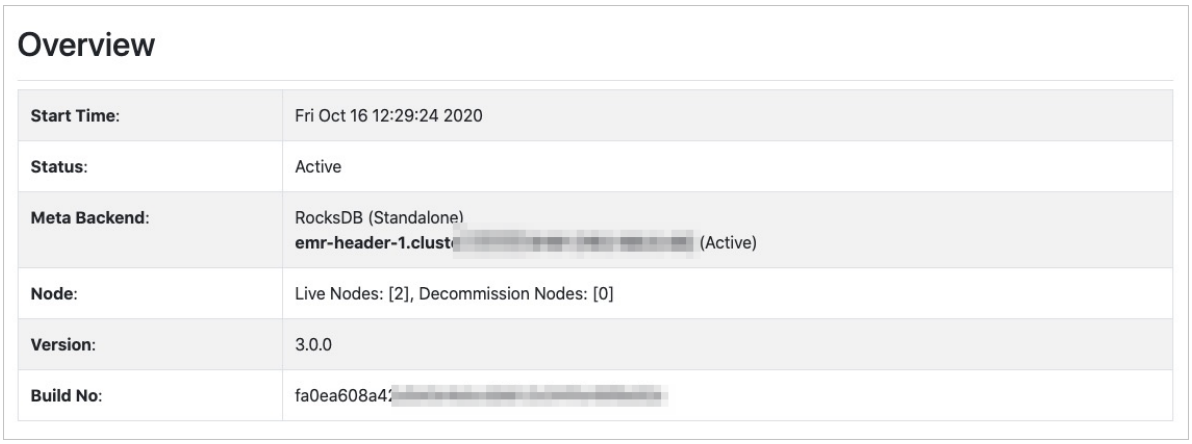
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8104/* to access the web UI of JindoFS. The web UI of JindoFS 3.1.X consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

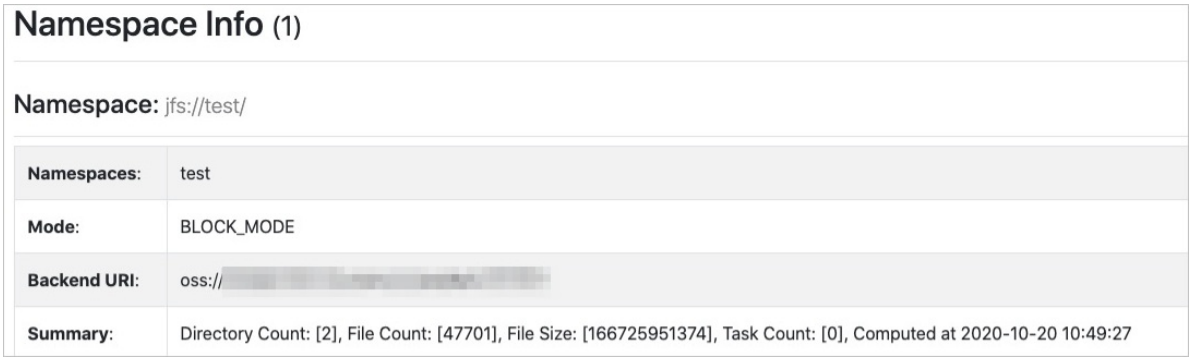


Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.



Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.

StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.

Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

9.3.7. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

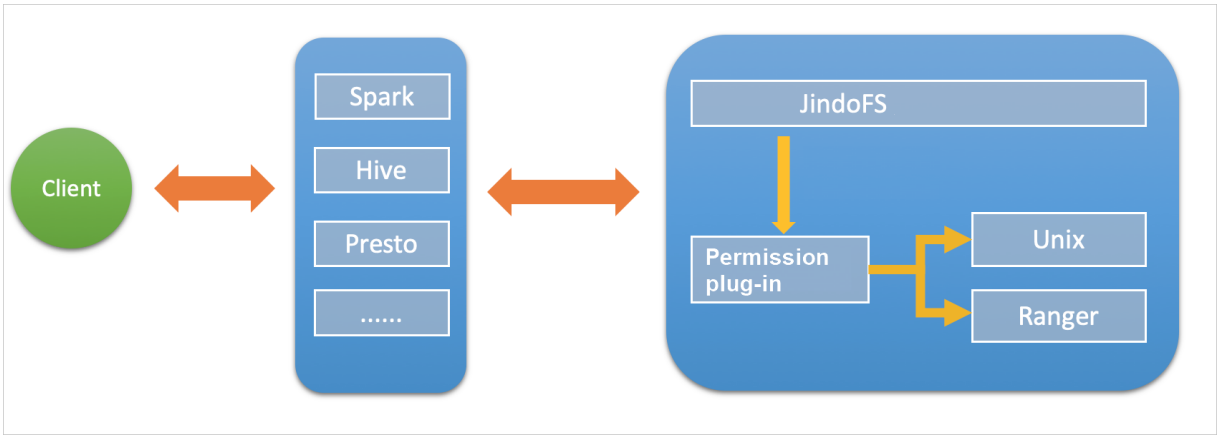
Permission management based on the storage mode:

- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.

- Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.

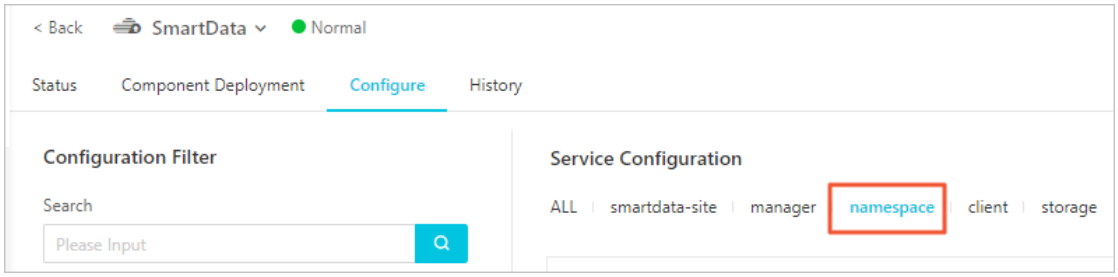
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

   After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

   If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

9.4. JindoTable

9.4.1. Use JindoTable

JindoTable is used to implement tiered storage, optimize table files, and collect data statistics based on the popularity of tables or partitions. This topic describes how to use JindoTable.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your on-premises machine.
- An EMR cluster of the 3.30.0 version or later is created. For more information about how to create a cluster, see Create a cluster.

Use JindoTable

Common commands:

- -accessStat
- -cache

- -archive
- -unarchive
- -uncache
- -status
- -optimize
- -showTable
- -showPartition
- -listTables
- -dumpmc

> **Notice** Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

-accessStat

- Syntax

  jindo table -accessStat {-d} *<days>* **{-n}** <topNums>

- Description

  This command is used to query the access records in which tables or partitions are visited most in a specified time range.

  <days> and <topNums> must be positive integers. If <days> is 1, all the access records generated from 00:00 (local time) on the current day to the current time are queried.

- Example: Query the 20 access records of tables or partitions that are most frequently visited within the last seven days.

```
jindo table -accessStat -d 7 -n 20
```

-cache

- Syntax

  jindo table -cache {-t} <dbName.tableName> **[-p]** <partitionSpec> **[-pin]**

- Description

  This command is used to cache data of specified tables or partitions to local disks.

  The data of the tables or partitions must be stored in Object Storage Service (OSS) or JindoFileSystem (JindoFS). Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`. When you specify `-pin`, if the cache space is insufficient, do not delete related data if possible.

- Example: Cache data of the db1.t1 table that is generated on March 16, 2020 to local disks.

```
jindo table -cache -t db1.t1 -p date=2020-03-16
```

-uncache

- Syntax

jindo table -uncache {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to delete cached data of specified tables or partitions from local disks.

  The data of the tables or partitions must be stored in OSS or JindoFS. Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

- Examples:
  - Delete cached data of the db1.t2 table from local disks.

    ```
    jindo table -uncache -t db1.t2
    ```

  - Delete cached data of the db1.t1 table from local disks.

    ```
    jindo table -uncache -t db1.t1 -p date=2020-03-16,category=1
    ```

-archive

- Syntax

  jindo table -archive {-a|i} {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to lower the level of the policy that is used to store data of specified tables or partitions. By default, the Archive storage class is used.

  To use the Infrequent Access (IA) storage class, add -i to the command. Specify tables in the format of database.table. Specify partitions in the format of 'partitionCol1=1,partitionCol2=2,...'.

- Example: Cache data of the db1.t1 table to local disks.

  ```
  jindo table -archive -t db1.t1 -p date=2020-10-12
  ```

-unarchive

- Syntax

  jindo table -unarchive [-o|-i] {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to change the storage class from Archive to Standard.

  If `-o` is added to the command, an Archived object is temporarily restored. If `-i` is added to the command, an Archived object is changed to an IA object.

- Example:

  ```
  jindo table -unarchive -o -t db1.t1 -p date=2020-03-16,category=1
  ```

  ```
  jindo table -unarchive -i -t db1.t2
  ```

-status

- Syntax

  jindo table -status {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to view the data storage status of specified tables or partitions.

- Examples:
  - View the data storage status of the db1.t2 table.

    ```
    jindo table -status -t db1.t2
    ```

  - View the data storage status of the db1.t1 table on March 16, 2020.

    ```
    jindo table -status -t db1.t1 -p date=2020-03-16
    ```

-optimize

- Syntax

  jindo table -optimize {-t} <dbName.tableName>

- Description

  This command is used to optimize the data organization of tables at the storage layer.

- Example: Optimize the data organization of the db1.t1 table at the storage layer.

  ```
  jindo table -optimize -t db1.t1
  ```

-showTable

- Syntax

  jindo table -showTable {-t} <dbName.tableName>

- Description

  This command is used to display all partitions in a partitioned table or display the data storage of a non-partitioned table.

- Example: Display all partitions in the db1.t1 partitioned table.

  ```
  jindo table -showTable -t db1.t1
  ```

-showPartition

- Syntax

  jindo table -showPartition {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to display the data storage of partitions.

- Example: Display the data storage of all partitions in the db1.t1 partitioned table on October 12, 2020.

  ```
  jindo table -showPartition -t db1.t1 -p date=2020-10-12
  ```

-listTables

- Syntax

  jindo table -listTables [-db] <dbName>

- Description

  This command is used to display all tables in a specified database. If you do not specify `[-db]`, tables in the default database are displayed.

- Examples:
  - Display tables in the default database.

    ```
    jindo table -listTables
    ```

  - Display tables in the db1 database.

    ```
    jindo table -listTables -db db1
    ```

-dumpmc

- Syntax

  jindo table -dumpmc {-i} <accessId> **{-k}** <accessKey> **{-m}** <numMaps> **{-t}** <tunnelUrl> **{-project}** <projectName> **{-table}** <tablename> **{-p}** <partitionSpec> **{-f}** <csv|tfrecord> **{-o}** <outputPath>

Parameter	Description	Required
-i	The AccessKey ID of your Alibaba Cloud account.	Yes
-k	The AccessKey secret of your Alibaba Cloud account.	Yes
-m	The number of map tasks.	Yes
-t	The VPC Tunnel endpoint of MaxCompute.	Yes
-project	The name of the MaxCompute project.	Yes
-table	The name of the MaxCompute table.	Yes
-p	The partition information. Example: `pt=xxx`. Separate multiple partitions with commas (,), such as `pt=xxx,dt=xxx`.	No
-f	The file format. Valid values: ○ tfrecord ○ csv	Yes
-o	The destination path.	Yes

- Description

This command is used to dump MaxCompute tables to an EMR cluster or OSS. The formats CSV and TFRECORD are supported.

- Examples:
  - Dump a MaxCompute table in the TFRECORD format to an EMR cluster.

    ```
    jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
    .aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o /tmp/outputtf1 -f tfrecord
    ```

  - Dump a MaxCompute table in the CSV format to OSS.

    ```
    jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
    .aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o oss://bucket1/tmp/outputcsv -f csv
    ```

9.4.2. Use JindoTable to collect access frequency statistics of tables and partitions

You can use JindoTable to collect access frequency statistics of tables and partitions, and separate cold and hot data based on the statistics. This helps you reduce storage costs and improves cache usage efficiency.

Collect access records

JindoTable allows you to collect access records of Hive tables. The Spark and Hive engines support this feature. The collected data is saved in the namespaces of the SmartData service of a cluster.

By default, the collection of access records is enabled. If you want to disable this feature, perform the operations described in Disable the collection of access records.

Query access frequency statistics

JindoTable allows you to run a command to query access frequency statistics.

- Syntax

  ```
  jindo table -accessStat <-d [days]> <-n [topNums]>
  ```

  Set `days` and `topNums` to positive integers. If days is set to 1, all the access records generated from 0:00 (local time) on the current day to the current time are queried.

- Description

  This command is used to query a specific number of access records of tables or partitions that are most frequently visited within a specified time range.

- Example: Query 20 access records of tables or partitions that are most frequently visited within the last seven days.

  ```
  jindo table -accessStat -d 7 -n 20
  ```

For more information about how to use JindoTable, see Use JindoTable.

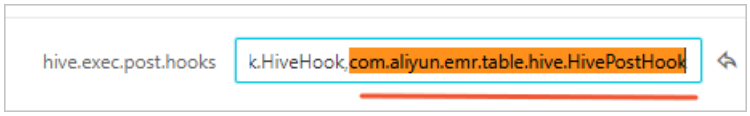
Disable the collection of access records

1.

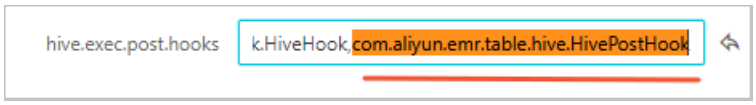
2. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
3. Click the **Cluster Management** tab.
4. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
5. Modify parameters.

> **Notice** You can perform the following operations to delete part of the value of the related parameter.

- Hive:
  a. In the left-side navigation pane, choose **Cluster Service > Hive**.
  b. Click the **Configure** tab.
  c. Click the **hive-site** tab in the Service Configuration section.
  d. Search for the hive.exec.post.hooks parameter in the Configuration Filter section and delete **com.aliyun.emr.table.hive.HivePostHook** from the parameter value.



- Spark:
  a. In the left-side navigation pane, choose **Cluster Service > Spark**.
  b. Click the **Configure** tab.
  c. Click the **spark-defaults** tab in the Service Configuration section.
  d. Search for the spark.sql.queryExecutionListeners parameter in the Configuration Filter section and delete **com.aliyun.emr.table.spark.SparkSQLQueryListener** from the parameter value.



6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Restart the related service.
   - Hive:
     a. In the upper-right corner of the page, choose **Actions > Restart HiveServer2**.
     b. In the **Cluster Activities** dialog box, specify the related parameters.
     c. Click **OK**.
     d. In the **Confirm** message, click **OK**.
   - Spark:
     a. In the upper-right corner of the page, choose **Actions > Restart ThriftServer**.

b. In the **Cluster Activities** dialog box, specify the related parameters.

c. Click **OK**.

d. In the **Confirm** message, click **OK**.

9.4.3. Use EMR JindoCube

JindoCube is supported in E-MapReduce (EMR) V3.24.0 and later. This topic describes how to install, deploy, and use JindoCube in EMR.

Prerequisites

A table or view is created.

Overview

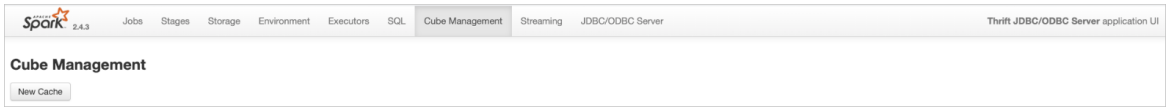
JindoCube is an advanced feature of EMR Spark. It implements pre-calculation to accelerate data processing. It improves query performance by dozens or even hundreds of times. You can persist data of a view to a data source supported by EMR Spark, such as HDFS or Object Storage Service (OSS). EMR Spark automatically discovers available persistent data and optimizes the execution plans for the data. This process is completely visible to users. JindoCube applies to business scenarios where a fixed query mode is used. JindoCube pre-calculates and pre-organizes data to accelerate data queries. For example, you can use JindoCube to implement multidimensional online analytical processing (MOLAP), generate data reports, create data dashboards, and synchronize data across clusters.

Install and deploy JindoCube

JindoCube is an advanced feature of EMR Spark. You can use JindoCube to accelerate all Dataset, DataFrame API, and SQL tasks that are submitted by using EMR Spark. No additional component deployment or maintenance is required.

1. Manage JindoCube on the Spark web UI.

   You can manage JindoCube on the Spark web UI. For example, you can create, delete, or update JindoCube caches. After you create a JindoCube cache, it automatically accelerates all Spark queries in the cluster. You can use the **spark.sql.cache.tab.display** parameter to specify whether to display the Cube Management tab on the Spark web UI. You can specify this parameter on the configuration page of the Spark service in the EMR console or in a Spark task submission command. The default value of this parameter is **false**.

   You can also use the **spark.sql.cache.useDatabase** parameter to create a database based on your business requirements and store the view for which you want to create a cache in this database. If partitioned tables are used, you can use the **spark.sql.cache.cacheByPartition** parameter to specify whether to cache data by partition.

Parameter	Description	Example
spark.sql.cache.tab.display	Specifies whether to display the Cube Management tab.	true

Parameter	Description	Example
spark.sql.cache.useDatabase	The databases where cubes are stored.	db1,db2,dbn
spark.sql.cache.cacheByPartition	Specifies whether to store cubes by partition.	true

2. Optimize Spark queries.

   You can use the **spark.sql.cache.queryRewrite** parameter to specify whether to use a JindoCube cache to accelerate Spark queries. You can specify this parameter at the cluster, session, or SQL statement level. The default value is **true**.

Use JindoCube

1. Create a JindoCube cache.
   i. Log on to the Alibaba Cloud EMR console by using your Alibaba Cloud account.
   ii. Click the **Cluster Management** tab.
   iii. Find the cluster where you want to create a JindoCube cache and click the cluster ID.
   iv. In the left-side navigation pane, click **Connect Strings**.
   v. On the **Public Connect Strings** page, click the link of YARN UI to go to the YARN web UI as a Knox user.

      For more information about Knox, see Knox.
   vi. Click **ApplicationMaster** in the row where **Application Type** is **SPARK** and **Name** is **Thrift JDBC/ODBC Server**.
   vii. On the Spark web UI, click the **Cube Management** tab.
   viii. Click **New Cache**.

      Select a table or view, and click raw cache or cube cache in the action column. You can create two types of caches:

- Raw cache: persists the data of a table or view based on the cache configurations.

  When you create a raw cache, specify the parameters described in the following table.

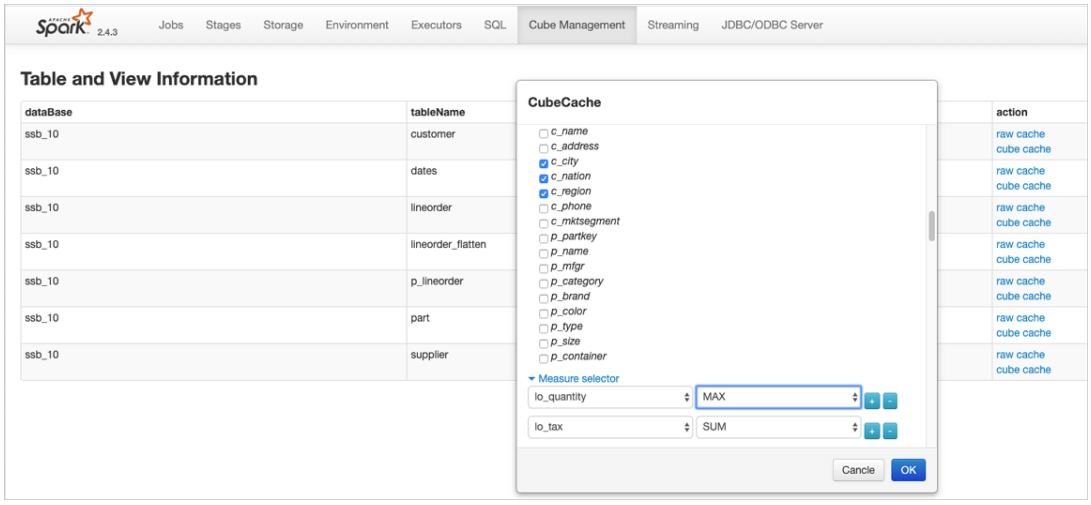
Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Column Selector	The columns from which you want to cache data.	Yes
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No

- Cube cache: builds a cube based on the raw data of a table or view and persists the cube data.

  When you create a cube cache, specify the parameters described in the following table.

Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Dimension Selector	The dimension fields that are used to build a cube.	Yes
Measure Selector	The measure fields and pre-calculation functions that are used to build a cube.	Yes

Parameter	Description	Required
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No



JindoCube builds a cube based on specified dimensions and measures. The following SQL statement shows the configurations of the cube cache in the preceding figure:

```
SELECT c_city, c_nation, c_region, MAX(lo_quantity), SUM(lo_tax)
FROM lineorder_flatten
GROUP BY c_city, c_nation, c_region;
```

JindoCube calculates the finest-grained combination of dimensions. Powered by the onsite computing capability of Spark, JindoCube implements re-aggregation to optimize the queries where a coarse-grained combination of dimensions is used. You must use the pre-calculation functions supported by JindoCube to define a cube cache. The following table lists the supported pre-calculation functions and the aggregate functions that correspond to them.

Aggregate function	Pre-calculation function
COUNT	COUNT
SUM	SUM
MAX	MAX
MIN	MIN
AVG	COUNT and SUM
COUNT (DISTINCT)	PRE_COUNT_DISTINCT
APPROX_COUNT_DISTINCT	PRE_APPROX_COUNT_DISTINCT

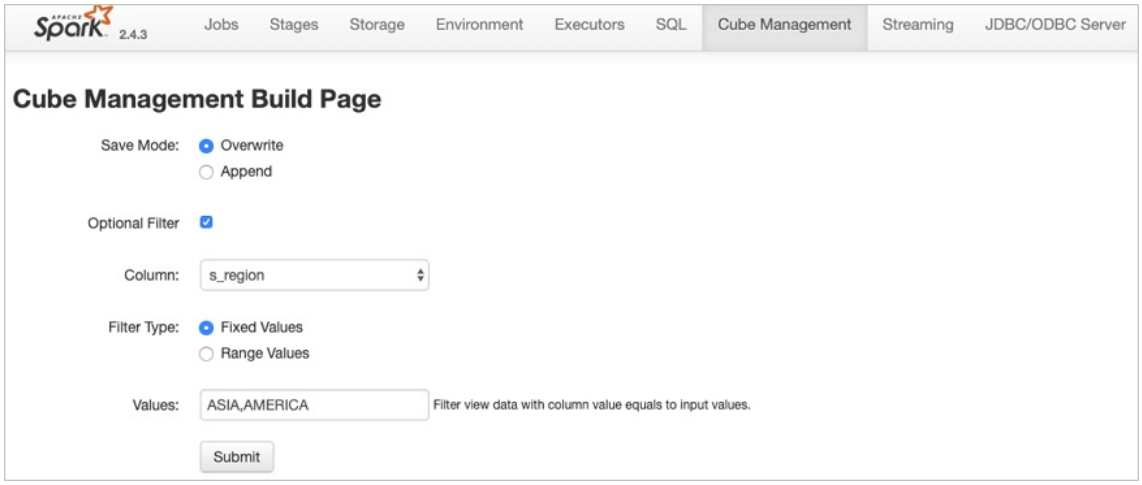
All created caches are displayed on the **Cube Management** tab. You can click **Detail** for a cache to view its details, including its basic information, partition information, building information, and building history.

2. Build a JindoCube cache.

   During the creation of a cache, metadata is prepared, but no data is persisted. Therefore, after you create a cache, you must continue to build the cache. This way, the related data can be persisted to permanent storage, such as HDFS or OSS. In addition, the source table of a cache may be appended with new data or updated. In this case, you must update the data in the cache to ensure data consistency. You can use one of the following methods to build a cache:

   - Manually build a cache

     Click Build Cache to manually build a cache. On the page that appears, configure the parameters, as shown in the following figure.



     The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.

Optional Filter	You can select this parameter to add filter conditions. This way, JindoCube filters data based on the filter conditions before it persists the data.
	■ Column: the field based on which data is filtered.
	■ Filter Type: specifies whether to filter data by fixed values or by value range.
	  ■ Fixed Values: If this option is selected, specify one or more values. Separate values with commas (,).
	  ■ Range Values: If this option is selected, specify a minimum value and a maximum value. You can leave the maximum value empty. In this case, the value range is the minimum value and all values greater than it.

The following SQL statement shows the configurations of the raw cache for the lineorder_flatten view in the preceding figure:

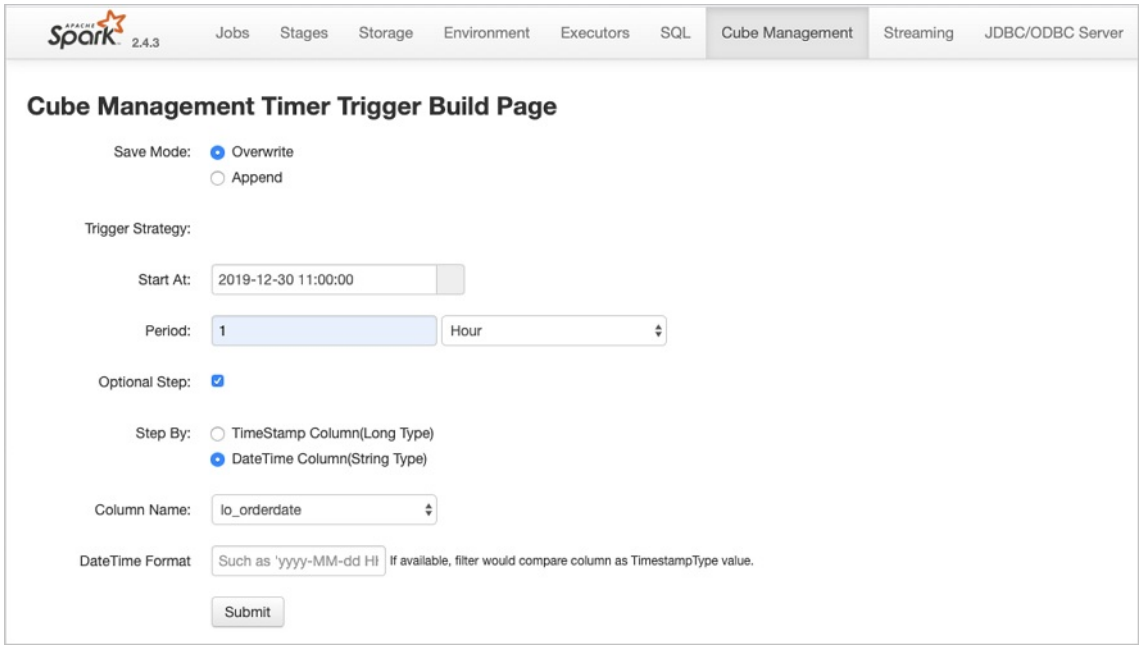
```
SELECT * FROM lineorder_flatten
WHERE s_region == 'ASIA' OR s_region == 'AMERICA';
```

Click **Submit** to submit the cache building task to a Spark cluster. You can view the task status in the Build Information section of the cache details page. After the task is complete, you can view the cache information in the Build History section.

> **Note** Spark writes the cached data to a specified directory in the same way common Spark tasks write data to a table or directory. If multiple users concurrently build a cache whose data format is Parquet, JSON, or ORC, the cached data may be inaccurate and invalid. If concurrent cache building operations or updates are inevitable, we recommend that you use a data format that supports concurrent writes, for example, Delta.

- Configure periodic cache building

  Click Trigger Period Build to configure a periodic cache update policy to ensure data consistency between the cache and source table. On the page that appears, configure the parameters, as shown in the following figure.



The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.
Trigger Strategy	Specifies the start time of the first building task, and the building interval. ■ Start At: Select or enter the time you want to trigger the first building task. Format: yyyy-MM-dd hh:mm:ss. ■ Period: Specify the interval at which cache building tasks are triggered.

Parameter	Description
Optional Step	Specifies the filter conditions for cache building. JindoCube updates the cache in incremental mode based on the filter conditions and interval. If you do not select Optional Step, JindoCube updates the entire cache each time. ■ Step By: Specify the type of the field that you want to update in incremental mode. TimeStamp Column(Long Type) indicates a timestamp field of the LONG type. DateTime Column(String Type) indicates a datetime field of the STRING type. ■ Column Name: Specify the name of the field that you want to update in incremental mode.

On the cache details page, you can view the configured cache update policy. You can click Cancel Period Build to cancel periodic updates. You can view the information of a complete building task in the Build History section.

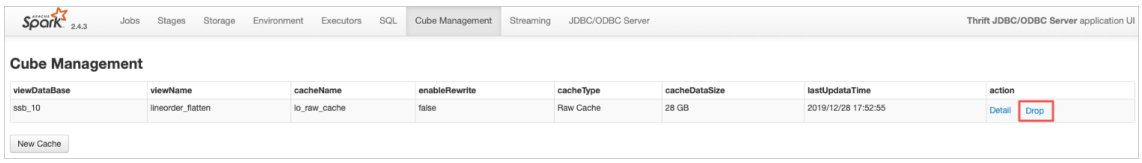
> **Note**
>
> - Periodic building tasks run on a Spark cluster. The related configurations are all saved in SparkContext, and the tasks are periodically triggered by Spark Driver. If you release the Spark cluster, periodic building is disabled.
> - The Build History section displays the following information of each complete task in the current Spark cluster: start time, end time, save mode, building conditions, and final status. After you release the Spark cluster, the information in the Build History section is cleared.

3. Manage a JindoCube cache.

   After you create and build a JindoCube cache, you can manage it on the Cube Management page.

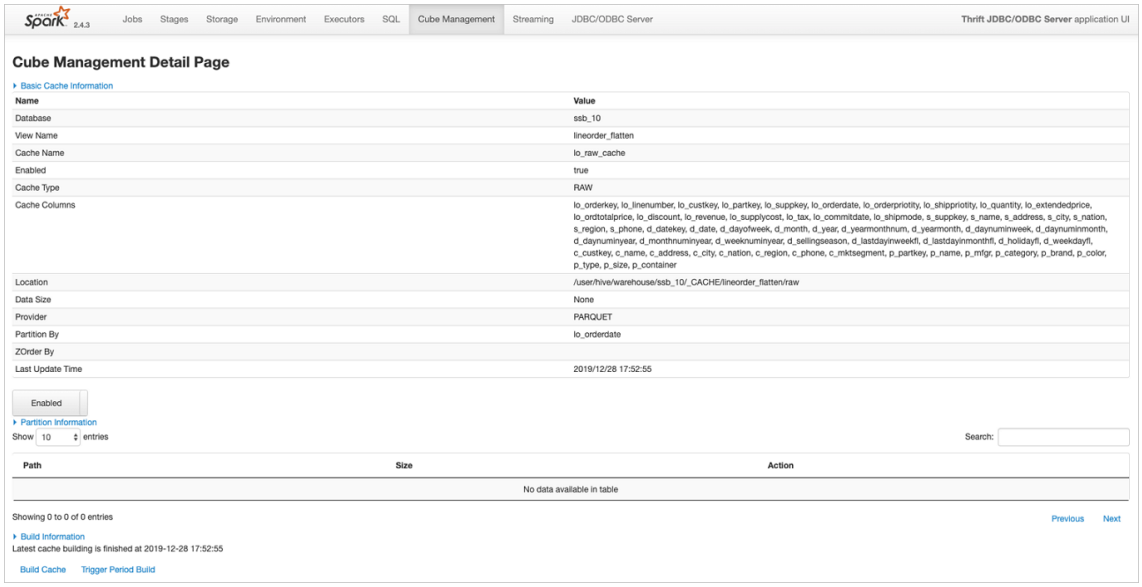
   - Delete a cache

     On the Cube Management page, find the cache you want to delete, and click **Drop** in the action column. All the metadata and data of the cache are deleted.



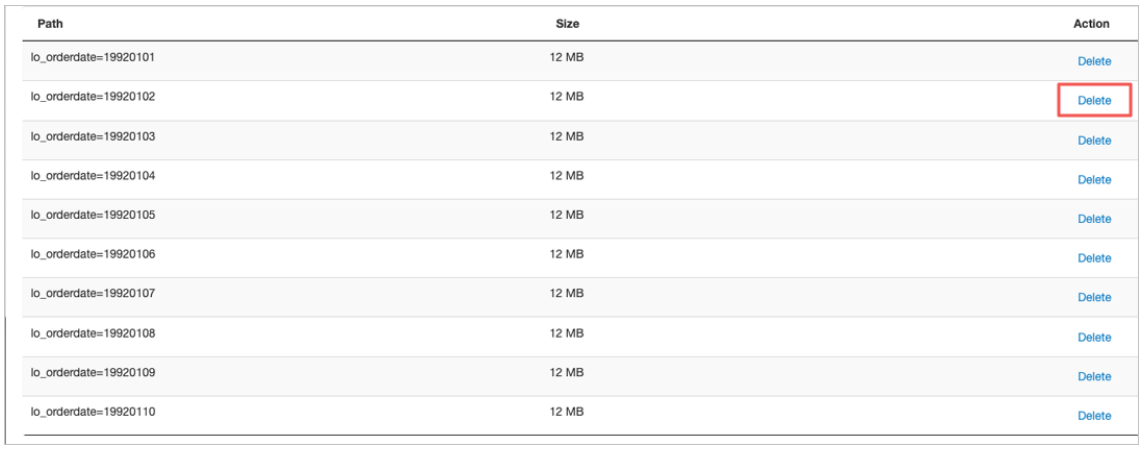
   - Enable and disable optimization of Spark queries

     JindoCube allows you to specify whether to optimize Spark queries at the cache level. On the details page of a cache, click **Enabled** or **Disabled** in the lower part of the Basic Cache Information section to enable or disable the optimization.



- Delete partition data from a cache

  If data in a cache is stored by partition, you can delete the unnecessary partition data from the cache to release more space. On the details page of the cache, find the partition whose data you want to delete, and click **Delete** in the Action column.



Path	Size	Action
lo_orderdate=19920101	12 MB	Delete
lo_orderdate=19920102	12 MB	Delete
lo_orderdate=19920103	12 MB	Delete
lo_orderdate=19920104	12 MB	Delete
lo_orderdate=19920105	12 MB	Delete
lo_orderdate=19920106	12 MB	Delete
lo_orderdate=19920107	12 MB	Delete
lo_orderdate=19920108	12 MB	Delete
lo_orderdate=19920109	12 MB	Delete
lo_orderdate=19920110	12 MB	Delete

> **Note** Before you delete the data of a partition, make sure that the data is no longer needed. If you delete data that is required in subsequent queries, the queries return invalid results.

4. Optimize Spark queries.

   JindoCube optimizes view-based queries. After you have created a raw cache or cube cache for a view, EMR Spark overwrites the execution plan of queries on this view. The new execution plan still uses the original logical semantics but can directly access the data in the cache. This accelerates Spark queries. Use the lineorder_flatten view as an example. It is a wide table view generated by associating the lineorder table with other dimension tables. The following figure shows the information about this view.

```
0: jdbc:hive2://localhost:10001/ssb_10> desc extended lineorder_flatten;
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+-----------------------------+-----------------------------------------------------+----------+
|           col_name          |                     data_type                       | comment  |
+-----------------------------+-----------------------------------------------------+----------+
| lo_orderkey                 | bigint                                              | NULL     |
| lo_linenumber               | bigint                                              | NULL     |
| lo_custkey                  | int                                                 | NULL     |
| lo_partkey                  | int                                                 | NULL     |
| lo_suppkey                  | int                                                 | NULL     |
| lo_orderdate                | int                                                 | NULL     |
| lo_orderpriotity            | string                                              | NULL     |
| lo_shippriotity             | int                                                 | NULL     |
| lo_quantity                 | bigint                                              | NULL     |
| lo_extendedprice            | bigint                                              | NULL     |
| lo_ordtotalprice            | bigint                                              | NULL     |
| lo_discount                 | bigint                                              | NULL     |
| lo_revenue                  | bigint                                              | NULL     |
| lo_supplycost               | bigint                                              | NULL     |
| lo_tax                      | bigint                                              | NULL     |
| lo_commitdate               | int                                                 | NULL     |
| lo_shipmode                 | string                                              | NULL     |
| s_suppkey                   | int                                                 | NULL     |
| s_name                      | string                                              | NULL     |
| s_address                   | string                                              | NULL     |
| s_city                      | string                                              | NULL     |
| s_nation                    | string                                              | NULL     |
| s_region                    | string                                              | NULL     |
| s_phone                     | string                                              | NULL     |
| d_datekey                   | int                                                 | NULL     |
| d_date                      | string                                              | NULL     |
| d_dayofweek                 | string                                              | NULL     |
| d_month                     | string                                              | NULL     |
| d_year                      | int                                                 | NULL     |
| d_yearmonthnum              | int                                                 | NULL     |
| d_yearmonth                 | string                                              | NULL     |
| d_daynuminweek              | int                                                 | NULL     |
| d_daynuminmonth             | int                                                 | NULL     |
| d_daynuminyear              | int                                                 | NULL     |
| d_monthnuminyear            | int                                                 | NULL     |
| d_weeknuminyear             | int                                                 | NULL     |
| d_sellingseason             | string                                              | NULL     |
| d_lastdayinweekfl           | int                                                 | NULL     |
| d_lastdayinmonthfl          | int                                                 | NULL     |
| d_holidayfl                 | int                                                 | NULL     |
| d_weekdayfl                 | int                                                 | NULL     |
| c_custkey                   | int                                                 | NULL     |
| c_name                      | string                                              | NULL     |
| c_address                   | string                                              | NULL     |
| c_city                      | string                                              | NULL     |
| c_nation                    | string                                              | NULL     |
| c_region                    | string                                              | NULL     |
| c_phone                     | string                                              | NULL     |
| c_mktsegment                | string                                              | NULL     |
| p_partkey                   | int                                                 | NULL     |
| p_name                      | string                                              | NULL     |
| p_mfgr                      | string                                              | NULL     |
| p_category                  | string                                              | NULL     |
| p_brand                     | string                                              | NULL     |
| p_color                     | string                                              | NULL     |
| p_type                      | string                                              | NULL     |
| p_size                      | int                                                 | NULL     |
| p_container                 | string                                              | NULL     |
|                             |                                                     |          |
| # Detailed Table Information |                                                     |          |
| Database                    | ssb_10                                              |          |
| Table                       | lineorder_flatten                                   |          |
| Owner                       | hadoop                                              |          |
| Created Time                | Sat Dec 28 17:30:02 CST 2019                        |          |
| Last Access                 | Thu Jan 01 08:00:00 CST 1970                        |          |
| Created By                  | Spark 2.2 or prior                                  |          |
| Type                        | VIEW                                                |          |
| View Text                   | SELECT `lineorder`.`lo_orderkey`, `lineorder`.`lo_linenumber`, `lineorder`.`lo_custkey`, `lineorder`.`lo_partkey`, `
lineorder`.`lo_suppkey`, `lineorder`.`lo_orderdate`, `lineorder`.`lo_orderpriotity`, `lineorder`.`lo_shippriotity`, `lineorder`.`lo_quantity`, `lineor
der`.`lo_extendedprice`, `lineorder`.`lo_ordtotalprice`, `lineorder`.`lo_discount`, `lineorder`.`lo_revenue`, `lineorder`.`lo_supplycost`, `lineorder`
.`lo_tax`, `lineorder`.`lo_commitdate`, `lineorder`.`lo_shipmode`, `supplier`.`s_suppkey`, `supplier`.`s_name`, `supplier`.`s_address`, `supplier`.`s_
city`, `supplier`.`s_nation`, `supplier`.`s_region`, `supplier`.`s_phone`, `dates`.`d_datekey`, `dates`.`d_date`, `dates`.`d_dayofweek`, `dates`.`d_mo
nth`, `dates`.`d_year`, `dates`.`d_yearmonthnum`, `dates`.`d_yearmonth`, `dates`.`d_daynuminweek`, `dates`.`d_daynuminmonth`, `dates`.`d_daynuminyear`
, `dates`.`d_monthnuminyear`, `dates`.`d_weeknuminyear`, `dates`.`d_sellingseason`, `dates`.`d_lastdayinweekfl`, `dates`.`d_lastdayinmonthfl`, `dates`
.`d_holidayfl`, `dates`.`d_weekdayfl`, `customer`.`c_custkey`, `customer`.`c_name`, `customer`.`c_address`, `customer`.`c_city`, `customer`.`c_nation`
, `customer`.`c_region`, `customer`.`c_phone`, `customer`.`c_mktsegment`, `part`.`p_partkey`, `part`.`p_name`, `part`.`p_mfgr`, `part`.`p_category`, `
part`.`p_brand`, `part`.`p_color`, `part`.`p_type`, `part`.`p_size`, `part`.`p_container` FROM `ssb_10`.`lineorder`, `ssb_10`.`supplier`, `ssb_10`.`da
tes`, `ssb_10`.`customer`, `ssb_10`.`part`
WHERE `lineorder`.`lo_orderdate` = `dates`.`d_datekey` AND `lineorder`.`lo_custkey` = `customer`.`c_custkey` AND `lineorder`.`lo_suppkey` = `supplier`
.`s_suppkey` AND `lineorder`.`lo_partkey` = `part`.`p_partkey` |          |
| Table Properties            | [transient_lastDdlTime=1577677599]                  |          |
| Serde Library               | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe  |          |
| InputFormat                 | org.apache.hadoop.mapred.TextInputFormat            |          |
| OutputFormat                | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat |          |
| Storage Properties          | [serialization.format=1]                            |          |
+-----------------------------+-----------------------------------------------------+----------+
73 rows selected (0.057 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

The following figure shows the execution plan of simple queries on the lineorder_flatten view.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(11) SortMergeJoin [lo_partkey#313], [p_partkey#359], Inner
   :- *(8) Sort [lo_partkey#313 ASC NULLS FIRST], false, 0
   :  +- Exchange hashpartitioning(lo_partkey#313, 200)
   :     +- *(7) SortMergeJoin [lo_custkey#312], [c_custkey#351], Inner
   :        :- *(4) Sort [lo_custkey#312 ASC NULLS FIRST], false, 0
   :        :  +- Exchange hashpartitioning(lo_custkey#312, 200)
   :        :     +- *(3) BroadcastHashJoin [lo_orderdate#315], [d_datekey#334], Inner, BuildRight
   :        :        :- *(3) BroadcastHashJoin [lo_suppkey#314], [s_suppkey#327], Inner, BuildRight
   :        :        :  :- *(3) Filter (((isnotnull(lo_suppkey#314) && isnotnull(lo_orderdate#315)) && isnotnull(lo_custkey#312)) && isnotnull(lo_part
key#313))
   :        :        :  :  +- Scan hive ssb_10.lineorder [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_ord
erdate#315, lo_orderpriotity#316, lo_shippriotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#32
2L, lo_supplycost#323L, lo_tax#324L, lo_commitdate#325, lo_shipmode#326], HiveTableRelation `ssb_10`.`lineorder`, org.apache.hadoop.hive.serde2.lazy.L
azySimpleSerDe, [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_orderdate#315, lo_orderpriotity#316, lo_ship
priotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#322L, lo_supplycost#323L, lo_tax#324L, lo_c
ommitdate#325, lo_shipmode#326]
   :        :        :  +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :        :     +- *(1) Filter isnotnull(s_suppkey#327)
   :        :        :        +- Scan hive ssb_10.supplier [s_suppkey#327, s_name#328, s_address#329, s_city#330, s_nation#331, s_region#332, s_phone#
333], HiveTableRelation `ssb_10`.`supplier`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [s_suppkey#327, s_name#328, s_address#329, s_city#330
, s_nation#331, s_region#332, s_phone#333]
   :        :        +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :           +- *(2) Filter isnotnull(d_datekey#334)
   :        :              +- Scan hive ssb_10.dates [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearm
onth#340, d_daynuminweek#341, d_daynuminmonth#342, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweek
fl#347, d_lastdayinmonthfl#348, d_holidayfl#349, d_weekdayfl#350], HiveTableRelation `ssb_10`.`dates`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSe
rDe, [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearmonth#340, d_daynuminweek#341, d_daynuminmonth#34
2, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweekfl#347, d_lastdayinmonthfl#348, d_holidayfl#349,
 d_weekdayfl#350]
   :        +- *(6) Sort [c_custkey#351 ASC NULLS FIRST], false, 0
   :           +- Exchange hashpartitioning(c_custkey#351, 200)
   :              +- *(5) Filter ((isnotnull(c_region#356) && (c_region#356 = ASIA)) && isnotnull(c_custkey#351))
   :                 +- Scan hive ssb_10.customer [c_custkey#351, c_name#352, c_address#353, c_city#354, c_nation#355, c_region#356, c_phone#357, c_mk
tsegment#358], HiveTableRelation `ssb_10`.`customer`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [c_custkey#351, c_name#352, c_address#353, c
_city#354, c_nation#355, c_region#356, c_phone#357, c_mktsegment#358]
   +- *(10) Sort [p_partkey#359 ASC NULLS FIRST], false, 0
      +- Exchange hashpartitioning(p_partkey#359, 200)
         +- *(9) Filter isnotnull(p_partkey#359)
            +- Scan hive ssb_10.part [p_partkey#359, p_name#360, p_mfgr#361, p_category#362, p_brand#363, p_color#364, p_type#365, p_size#366, p_conta
iner#367], HiveTableRelation `ssb_10`.`part`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [p_partkey#359, p_name#360, p_mfgr#361, p_category#3
62, p_brand#363, p_color#364, p_type#365, p_size#366, p_container#367] |
+----------------------------------------------------+
1 row selected (0.435 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

After a raw cache is created and built for the lineorder_flatten view, the same queries are performed on the view. EMR Spark automatically uses the cache to optimize the original execution plan. The following figure shows the new execution plan.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:17:47 INFO [main] HiveConf: Found configuration file file:/etc/ecm/spark-conf-2.4.3-hadoop2.8-1.4.2/hive-site.xml
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(1) Project [lo_orderkey#128L, lo_linenumber#129L, lo_custkey#130, lo_partkey#131, lo_suppkey#132, lo_orderdate#133, lo_orderpriotity#134, lo_ship
priotity#135, lo_quantity#136L, lo_extendedprice#137L, lo_ordtotalprice#138L, lo_discount#139L, lo_revenue#140L, lo_supplycost#141L, lo_tax#142L, lo_c
ommitdate#143, lo_shipmode#144, s_suppkey#145, s_name#146, s_address#147, s_city#148, s_nation#149, s_region#150, s_phone#151, ... 34 more fields]
   +- *(1) Filter (isnotnull(c_region#174) && (c_region#174 = ASIA))
      +- *(1) FileScan parquet [lo_orderkey#128L,lo_linenumber#129L,lo_custkey#130,lo_partkey#131,lo_suppkey#132,lo_orderpriotity#134,lo_shippriotity#
135,lo_quantity#136L,lo_extendedprice#137L,lo_ordtotalprice#138L,lo_discount#139L,lo_revenue#140L,lo_supplycost#141L,lo_tax#142L,lo_commitdate#143,lo_
shipmode#144,s_suppkey#145,s_name#146,s_address#147,s_city#148,s_nation#149,s_region#150,s_phone#151,d_datekey#152,... 34 more fields] Batched: true,
Format: Parquet, Location: InMemoryFileIndex[/user/hive/warehouse/ssb_10/_CACHE/lineorder_flatten/raw], PartitionCount: 2406, PartitionFilters: [], Pu
shedFilters: [IsNotNull(c_region), EqualTo(c_region,ASIA)], ReadSchema: struct<lo_orderkey:bigint,lo_linenumber:bigint,lo_custkey:int,lo_partkey:int,l
o_suppkey:int,lo_or... |
+----------------------------------------------------+
```

The new execution plan no longer uses the calculation logic of the lineorder_flatten view. Instead, it directly accesses the cached data in HDFS.

Usage notes

1. JindoCube does not ensure data consistency between the cache and source table. You must manually trigger data updates or configure a periodic update policy based on your requirements for data consistency.
2. JindoCube determines whether to use a cache to optimize execution plans based on the metadata of views. If cached data is incomplete when JindoCube optimizes an execution plan, the query results may be invalid or incomplete. The following operations or configurations may cause incomplete cached data: 1. Delete partition data that is required in subsequent queries on the cache details page. 2. Data required in subsequent queries is filtered out based on the filter conditions specified when you build or update the cache. 3. Required data has not been updated to the cache when the query is performed.

9.5. Toolset

9.5.1. Use JindoFS FUSE

This topic describes how to access JindoFileSystem (JindoFS) from a Filesystem in Userspace (FUSE) client. FUSE allows you to use JindoFS in block storage mode and cache mode (only JFS Scheme supported).

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

FUSE is a method provided by the Linux kernel to mount a file system. You can use a FUSE client to map files in a JindoFS cluster to a local disk. Then, you can directly access data in the JindoFS cluster without the need to use the `hadoop fs -ls jfs://<namespace>/` command.

Mount a directory

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs
   ```

3. Run the following command to mount the directory:

   ```
   jindofs-fuse /mnt/jfs
   ```

   In this example, the */mnt/jfs/* directory is mounted.

Read and write files

1. Run the following command to query all the sub-directories under the /mnt/jfs/ directory:

   ```
   ls /mnt/jfs/
   ```

   All the namespaces that you have configured on the server are returned.

   ```
   test  testcache
   ```

2. Run the following command to query all the files in the test namespace:

   ```
   ls /mnt/jfs/test/
   ```

3. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs/test/dir1
   ls /mnt/jfs/test/
   ```

4. Run the following command to write data into a file:

```
echo "hello world" > /tmp/hello.txt
cp /tmp/hello.txt /mnt/jfs/test/dir1/
```

5. Run the following command to read data from a file:

```
cat /mnt/jfs/test/dir1/hello.txt
```

The following information is returned:

```
hello world
```

The following example shows how to read data from and write data into files in Python:

1. Write data into a file in Python. Create the *write.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'w',encoding = 'utf-8') as f:
   f.write("my first file\n")
   f.write("This file\n\n")
   f.write("contains three lines\n")
```

2. Read data from a file in Python. Create the *read.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'r',encoding = 'utf-8') as f:
    lines = f.readlines()
    [print(x, end = '') for x in lines]
```

Run the following command to read the data that is written into the *test.txt* file:

```
[hadoop@emr-header-1 ~]$ ./read.py
```

The following information is returned:

```
my first file
This file
```

Unmount FUSE

Note Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to unmount FUSE:

```
umount jindofs-fuse
```

If the `target is busy` error is returned, switch to another directory, stop all the programs that are reading data from and writing data into FUSE files, and then unmount FUSE.

9.5.2. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain help information:

```
jindo distcp --help
```

   The following information is returned:

```
     --help           - Print help text
     --src=VALUE          - Directory to copy files from
     --dest=VALUE              - Directory to copy files to
     --parallelism=VALUE         - Copy task parallelism
     --outputManifest=VALUE       - The name of the manifest file
     --previousManifest=VALUE   -   The path to an existing manifest file
     --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
if specified
     --copyFromManifest   -   Copy from a manifest instead of listing a directory
     --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
     --srcPattern=VALUE   -   Include only source files matching this pattern
     --deleteOnSuccess   -   Delete input files after a successful copy
     --outputCodec=VALUE   -   Compression codec for output files
     --groupBy=VALUE   -   Pattern to group input files by
     --targetSize=VALUE   -   Target size for output files
     --enableBalancePlan   -   Enable plan copy task to make balance
     --enableDynamicPlan   -   Enable plan copy task dynamically
     --enableTransaction   -   Enable transation on Job explicitly
     --diff   -   show the difference between src and dest filelist
     --ossKey=VALUE   -   Specify your oss key if needed
     --ossSecret=VALUE   -   Specify your oss secret if needed
     --ossEndPoint=VALUE   -   Specify your oss endPoint if needed
     --policy=VALUE   -   Specify your oss storage policy
     --cleanUpPending   -   clean up the incomplete upload when distcp job finish
     --queue=VALUE   -   Specify yarn queuename if needed
     --bandwidth=VALUE   -   Specify bandwidth per map/reduce in MB if needed
     --s3Key=VALUE   -   Specify your s3 key
     --s3Secret=VALUE   -   Specify your s3 Sercet
     --s3EndPoint=VALUE   -   Specify your s3 EndPoint
```

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-
01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03
/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03
/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017
-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-0
1/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-0
1/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_tab
le/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/
03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/
03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromMani
fest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

Note If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

--queue

--queue specifies the name of the YARN queue where the current DistCp task resides.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<your_bucket>/hourly_table --qu
eue yarnqueue
```

--bandwidth

--bandwidth specifies the bandwidth allocated to the current DistCp task. This prevents the task from using excessive bandwidth. Unit: MB/s.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.c
om --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy archive --parallelism 20
  ```

- Example of the IA storage class (`--ia`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy ia --parallelism 20
  ```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

Use Amazon S3 as a data source

You can use the --s3Key, --s3Secret, and --s3EndPoint parameters in a command to specify information related to Amazon S3.

Example:

```
jindo distcp jindo-distcp-2.7.3.jar --src s3a://yourbucket/ --dest oss://<your_bucket>/hour
ly_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com
```

You can configure the s3Key, s3Secret, and s3EndPoint parameters in the *core-site.xml* file of Hadoop. This way, you do not need to specify an AccessKey pair each time.

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

Example:

```
jindo distcp /tmp/jindo-distcp-2.7.3.jar --src s3://smartdata1/ --dest s3://smartdata1/tmp
--s3EndPoint  s3-us-west-1.amazonaws.com
```

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

Note If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

9.5.3. Use Jindo DistCp in specific scenarios

This topic describes how to use Jindo DistCp in specific scenarios.

Prerequisites

- An E-MapReduce (EMR) cluster of a required version is created. For more information, see Create a cluster.
- Java Development Kit (JDK) 1.8 is installed.
- *jindo-distcp-<version>.jar* is downloaded based on your Hadoop version.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 2.7 and later 2.X versions.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 3.X.

Use scenarios

Common use scenarios of Jindo DistCp:

- Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?
- Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?
- Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?
- Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?
- Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file size?
- Scenario 8: What parameters are required if Amazon S3 is used as a data source?

- Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?
- Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?
- Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?
- Scenario 12: What parameters are required if I want to delete original files after a copy operation?
- Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?

If you are not using EMR, the following conditions must be met before you can import HDFS data to Object Storage Service (OSS):

- You can read data from HDFS.
- The AccessKey ID, AccessKey secret, and endpoint of OSS are obtained. You can write data to a destination OSS bucket.
- The storage class of the OSS bucket is not Archive.
- You can submit MapReduce tasks.
- The JAR package of Jindo DistCp is downloaded.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 10
```

> **Note** For more information about the parameters, see Use Jindo DistCp.

To import a large amount of data or a large number of files, such as millions or tens of millions of files, to OSS, you can set parallelism to a large value to increase the concurrency. You can also set `enableBatch` to true for optimization. Optimization command:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 500 --enableBatch
```

Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?

You can use one of the following methods to verify data integrity:

- Jindo DistCp Counters

  Check DistCp Counters in the counter information of a MapReduce task.

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

where:

- Bytes Destination Copied specifies the size of the file that you copy to a destination directory, in bytes.
- Bytes Source Read specifies the size of the file that you read from the source directory, in bytes.
- Files Copied specifies the number of copied files.

- Jindo DistCp --diff

  Use `--diff` to compare the information of files in the source and destination directories. The information includes file name and file size. Missing files and the files that fail to be copied are recorded in a manifest file. The manifest file is generated in the directory where you perform the comparison.

  Add `--diff` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --diff
```

  If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?

If your DistCp task fails, and you want to perform resumable upload to copy only the files that fail to be copied, perform the following operations on the basis of the command described in Scenario 1:

1. Add `--diff` to check whether all files are copied.

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --diff
```

   If all the files are copied, the following information is returned. Otherwise, a manifest file is generated. In this case, go to the next step.

```
INFO distcp.JindoDistCp: distcp has been done completely.
```

2. Use `--copyFromManifest` and `--previousManifest` to copy the remaining files. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --dest oss://yang-hhht/hourly_table --previousManifest=file:///opt
/manifest-2020-04-17.gz --copyFromManifest --parallelism 20
```

`file:///opt/manifest-2020-04-17.gz` is the path where the manifest file is stored.

Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?

1. If no information is generated for the files that were copied last time, specify parameters to generate a manifest file to record the information about the files that are copied.

   Add `--outputManifest=manifest-2020-04-17.gz` and `--requirePreviousManifest=false` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

   where:

   - `--outputManifest` generates a manifest file whose name can be customized and must have a file name extension of gz, such as *manifest-2020-04-17.gz*. This file is stored in the directory specified by `--dest`.
   - `--requirePreviousManifest` specifies whether to require a previous manifest file.

2. After a DistCp task is complete, copy the new files that may be generated in the source directory.

   Add `--outputManifest=manifest-2020-04-17.gz` and `--previousManifest=oss://yang-hhht/hourly_table/manifest-2020-04-17.gz` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

```
hadoop jar jindo-distcp-2.7.3.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --outputManifest=manifest-2020-04-18.gz --previousManifest=oss://yang-hhht
/hourly_table/manifest-2020-04-17.gz --parallelism 10
```

3. Repeat Step 2 to continuously copy incremental files.

Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?

Add the following two parameters to the command described in Scenario 1. These two parameters can be used together or separately.

- `--queue` : the name of the YARN queue.
- `--bandwidth` : the size of the specified bandwidth , in MB.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --queue yarnqueue --bandwidth 6 --parallelism 10
```

Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?

- To write data to OSS Archive storage, add `--archive` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --archive --parallelism 20
```

- To write data to OSS Infrequent Access (IA) storage, add `--ia` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --ia --parallelism 20
```

Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file sizes?

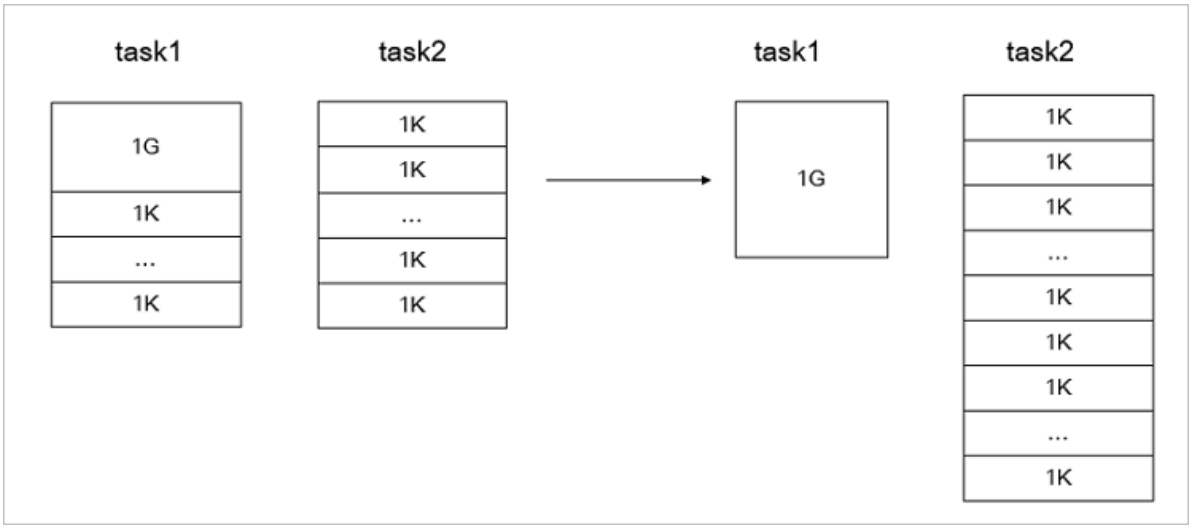
- A large number of small files and large data volume for a single large file

  If most of the files that you want to copy are small files, and the data volume of a single large file is large, the general solution is to randomly allocate the files to be copied. In this case, if you do not optimize the job allocation plan, few large files and many small files may be allocated to the same copy process. This cannot achieve excellent copy performance.

  Add `--enableDynamicPlan` to the command described in Scenario 1 to enable the optimization. --enableDynamicPlan cannot be used with `--enableBalancePlan`. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableDynamicPlan --parallelism 10
```

The following figure shows the data copy before and after the plan is optimized.

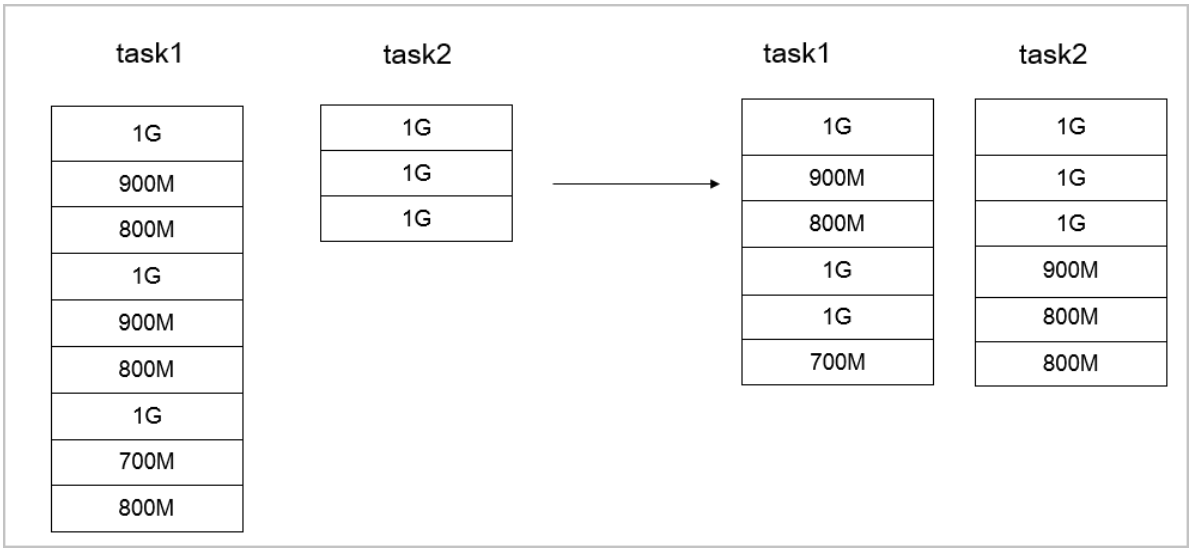


- No significant differences in sizes of files

  If the sizes are not significantly different among the files that you want to copy, use `--enableBalancePlan` to optimize the job allocation plan. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableBalancePlan --parallelism 10
```

  The following figure shows the data copy before and after the plan is optimized.



Scenario 8: What parameters are required if Amazon S3 is used as a data source?

Replace the parameters that specify the AccessKey ID, AccessKey secret, and endpoint of OSS in the command described in Scenario 1 with the following parameters of Amazon S3:

- `--s3Key`: the AccessKey ID for Amazon S3
- `--s3Secret`: the AccessKey secret for Amazon S3

- `--s3EndPoint` : the endpoint for Amazon S3

Example:

```
hadoop jar jindo-distcp-<version>.jar --src s3a://yourbucket/ --dest oss://yang-hhht/hourly
_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com --para
llelism 10
```

Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?

You can use `--outputCodec` to compress copied files in a format such as LZO or GZ to reduce the space that is used to store the files.

Add `--outputCodec` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --outputCodec=gz --parallelism 10
```

You can set --outputCodec to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files are compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change to the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?

- If you want to copy the files that meet specific rules, add `--srcPattern` to the command described in Scenario 1. Example:

  ```
  hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
  -hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
  aliyuncs.com --srcPattern .*\.log --parallelism 10
  ```

  `--srcPattern` : specifies a regular expression that filters files for the copy operation.

- If you want to copy the files stored in some sub-directories of the same parent directory, add `--srcPrefixesFile` to the command described in Scenario 1.

  ```
  hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
  -hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
  aliyuncs.com --srcPrefixesFile file:///opt/folders.txt --parallelism 20
  ```

  `--srcPrefixesFile` : enables Jindo DistCp to copy files in multiple folders under the same parent directory at a time.

Content of the *folders.txt* file:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?

Add the following parameters to the command described in Scenario 1:

- `--targetSize` : the maximum size of the merged file, in MB
- `--groupBy` : the merging rule, which is a regular expression

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Scenario 12: What parameters are required if I want to delete original files after a copy operation?

Add `--deleteOnSuccess` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --deleteOnSuccess --parallelism 10
```

Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Jindo DistCp allows you to save the AccessKey ID, AccessKey secret, and endpoint of OSS in the *core-site.xml* file so that you do not need to repeatedly specify the information.

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of OSS, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.jfs.cache.oss-accessKeyId</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-accessKeySecret</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-endpoint</name>
        <value>oss-cn-xxx.aliyuncs.com</value>
    </property>
</configuration>
```

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of Amazon S3, save the

following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

9.5.4. Use tiered storage commands of JindoFS

In E-MapReduce (EMR) V3.30, JindoFS provides a tiered storage feature. This feature allows you to store cold and hot data in different storage media. This helps reduce the data storage costs or accelerate data access.

Use jindo jfs

Run the following command to obtain the help information:

```
jindo jfs -help archive
-archive -i/a <path> ... :
  Archive commands.
```

Tiered storage commands in JindoFS are asynchronous and start only related tasks.

Common commands:

- Cache
- Uncache
- Archive
- Unarchive
- Status
- ls2 command

Cache

You can use this command to back up data stored in a specific path to local disks. Then, you can read the data from local disks, without the need to read data from Object Storage Service (OSS).

```
jindo jfs -cache -p <path>
```

`-p` can be used to ensure that local data is not cleared based on disk usage.

Uncache

You can use this command to delete backup data from local disks and store data only to OSS Standard storage.

```
jindo jfs -uncache  <path>
```

Archive

You can use this command to delete backup data from local disks and store data to OSS in Infrequent Access (IA) or Archive mode. For information about the storage classes, see Overview.

```
jindo jfs -archive -i|-a <path>
```

`-i` is used to store data to OSS in IA mode. `-a` is used to store data to OSS in Archive mode.

Unarchive

You can use this command to restore data from the Archive mode to the IA or Standard mode. You can temporarily restore data stored in Archive mode to allow the data to be readable within one day.

```
jindo jfs -unarchive -i/-o <path>
```

`-i` is used to store data to OSS in IA mode. `-o` is used to temporarily restore data stored in Archive mode to allow the data to be readable.

Status

You can use this command to view task details. By default, the number of files to be subject to tiered storage in a specific directory and the data that has been subject to tiered storage are measured.

```
jindo jfs -status -detail/-sync <path>
```

`-detail` is used to view the storage progress of file data. `-sync` indicates that the command exits after a tiered storage task is completed.

ls2 command

JindoFS provides the ls2 command that allows you to view the file storage status on the basis of Hadoop ls commands:

```
hadoop fs -ls2 <path>
```

Example of command output, which includes the file storage class:

```
drwxrwxrwx  - -         0    2020-06-05 04:27 oss://xxxx/warehouse
-rw-rw-rw-  1 Archive   1484 2020-09-23 16:40 oss://xxxx/wikipedia_data.csv
-rw-rw-rw-  1 Standard  1676 2020-06-07 20:04 oss://xxxx/wikipedia_data.json
```

10.SmartData 3.0.x

10.1. SmartData 3.0.x

SmartData is a storage service for the E-MapReduce (EMR) Jindo engine. SmartData provides centralized storage, caching, computing optimization, and feature extension for EMR computing engines. SmartData includes JindoFileSystem (JindoFS), JindoTable, and related tool sets. This topic describes the updates in SmartData 3.0.x.

Optimization of JindoFS storage

- The stand-alone configuration of Namespace Service is improved. In this case, metadata can be updated and asynchronously written into a Tablestore instance.
- The configuration of using a Tablestore instance as the metadata storage backend is removed. The Tablestore-based HA solution is no longer supported.
- File data can be stored in Archive mode in OSS to save costs.
- JindoFS tiered storage commands Archive, Unarchive, and Status are provided. You can use these commands to archive data to a specified directory and view the archive operation progress and related status.
- The ls2 command is provided to view the file information.
- Offline export, analysis, and query of fsimage are supported.
- Cross-cluster access to JindoFS is implemented.

For more information about the tiered storage commands of JindoFS, see Use tiered storage commands of JindoFS.

Optimization of JindoFS caching

- The structure of disks used to cache data is optimized. Dependencies on the system disk are removed. Data disks are independent of each other. The operation of taking disk resources offline is enhanced.
- The cache service is improved. Node fault tolerance and node disconnection operations are enhanced.
- The policies that are used to select disks into which data from cache blocks is written are optimized. By default, round-robin scheduling is supported.
- Read and write processes are improved. Fault tolerance is enhanced.
- The JindoFS tiered storage commands Cache, Uncache, and Status are provided. You can use these commands to cache data to a specified directory, preload data, and view the cache progress and related status.
- The problem that small files occupy much cache space is resolved. Related metrics are correctly measured.

Optimization of JindoTable computing

- JindoTable provides the -optimize command. You can use this command to optimize Hive table operations, such as merging small files in partitions.
- JindoTable provides the -archive, -unarchive, and -status commands. You can use these commands to archive data to specified tables or partitions, and view the archive operation progress and related status.

- JindoTable provides the -cache, -uncache, and -status commands. You can use these commands to cache data to specified tables or partitions, preload data, and view the cache progress and related status.
- You can export a MaxCompute table to JindoFS. Then, you can preload structured data before machine learning training.

For more information about JindoTable, see Use JindoTable.

OSS storage scalability on JindoFS

- Integration of Ranger permissions on the client is supported. This way, you can obtain the permissions to perform various operations in OSS. You can use service records in JindoFS to check Ranger permissions.
- Operation audit on the client is supported. This way, you can obtain the permissions to perform various operations in OSS. Operation records that are generated by using JindoFS are used for auditing.
- A Hadoop credential provider is supported. You can use common Hadoop methods to configure AccessKey pair information of OSS.
- A Flink connector is supported. OSS serves as a source, sink, or checkpoint in a Flink engine.
- JindoFS OSS SDKs of a Lite edition, such as a Hadoop connector, are provided and suitable for non-standard environments. A self-managed data center is a non-standard environment.

JindoManager system management

You can access a web UI to view the status and file statistics about the storage system and statistics about cache metrics on the cache system of JindoFS.

JindoTools

The distribution mechanism of Jindo DistCp is improved. Different distribution packages are offered for EMR clusters and non-EMR clusters.

Jindo DistCp provides a Lite edition, which is suitable for non-standard environments, such as a self-managed data center.

10.2. JindoFS in block storage mode

10.2.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read and write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

Context

The block storage mode of JindoFS has the following features:

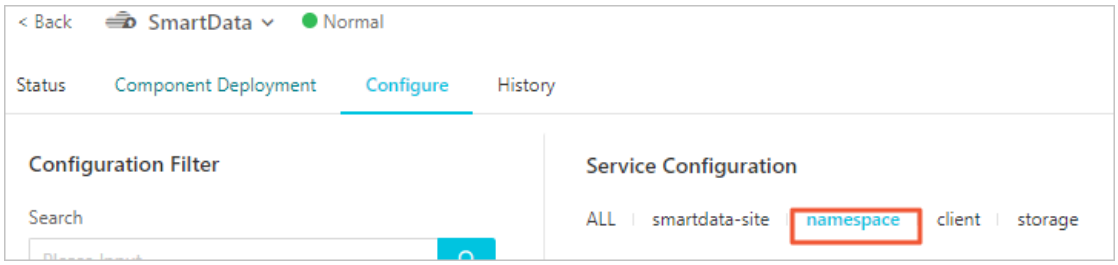
- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM)

solutions.

- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

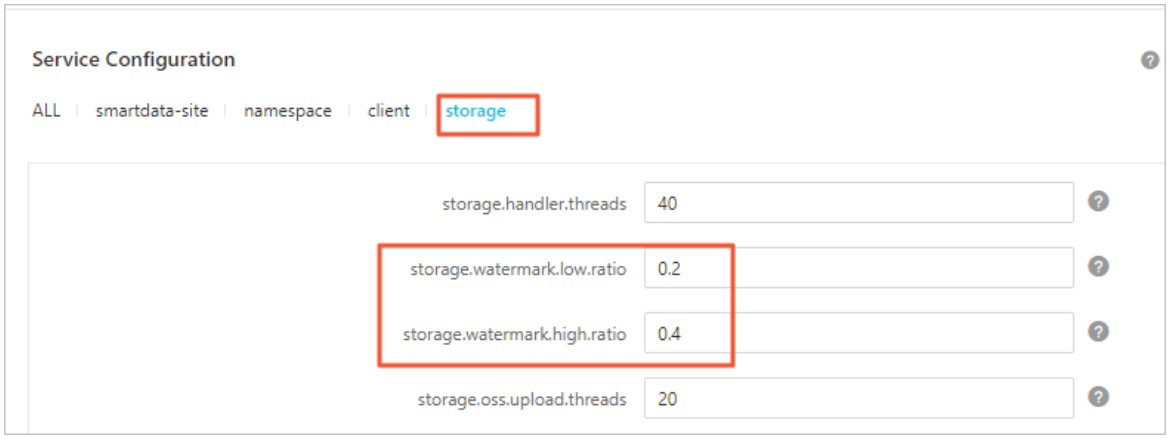
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Choose **Actions > Restart Jindo Storage Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

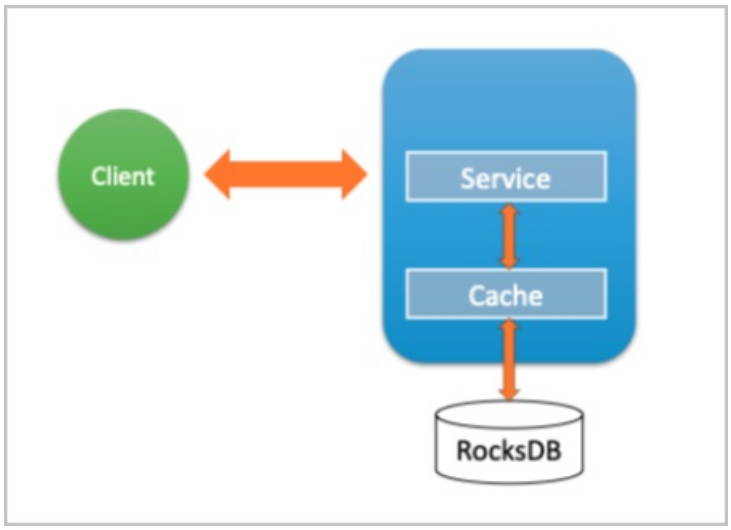
10.2.2. Use RocksDB to store metadata

Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

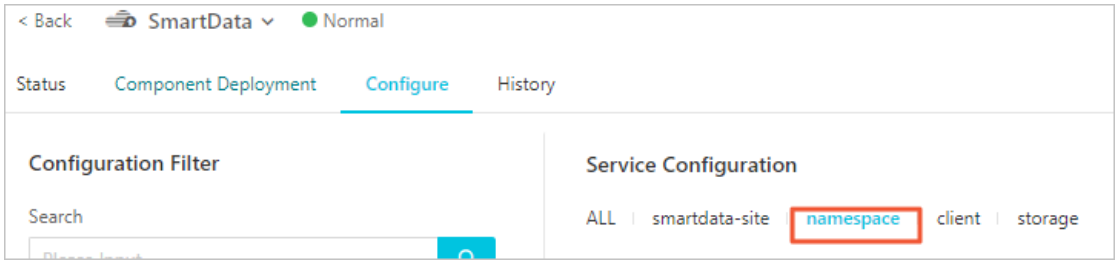
RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage backend in high availability mode, we recommend that you use Raft instances. For more information, see Use Raft-RocksDB-Tablestore to store metadata.

The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
6. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

7. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

8. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. Prepare for the recovery.

   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

   ```
   hadoop fs -count jfs://test/
   ```

   Information similar to the following example is returned:

   ```
    1596      1482809                 25 jfs://test/
   ```

   The number of folders is 1596, and the number of files is 1,482,809.

   ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1`, the latest metadata is synchronized to the Tablestore instance.

   ```
   jindo jfs -metaStatus
   ```

   Information similar to that shown in the following figure is returned.

   ```
   [hadoop@emr-header-1 ~]$ jindo jfs -metaStatus
   ==== emr-header-1:8101 ====
   OtsUploader: _synced=1
   [RocksDB Row Counts of each CF(Table)]
   ```

   iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

   ```
   jindo jfs -metaStatus
   ```

   If the state is FINISH, the recovery is completed, as shown in the following figure.

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.rocksdb.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.rocksdb.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

10.2.3. Use Raft-RocksDB-Tablestore to store metadata

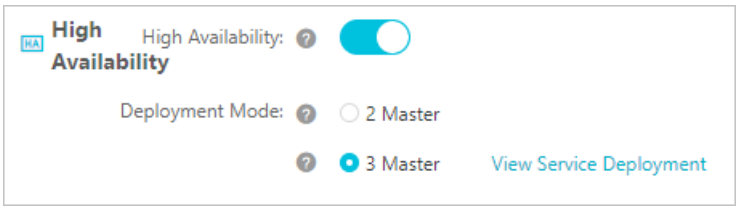
In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata.

Prerequisites

- A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  Note You must enable the transaction feature.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.

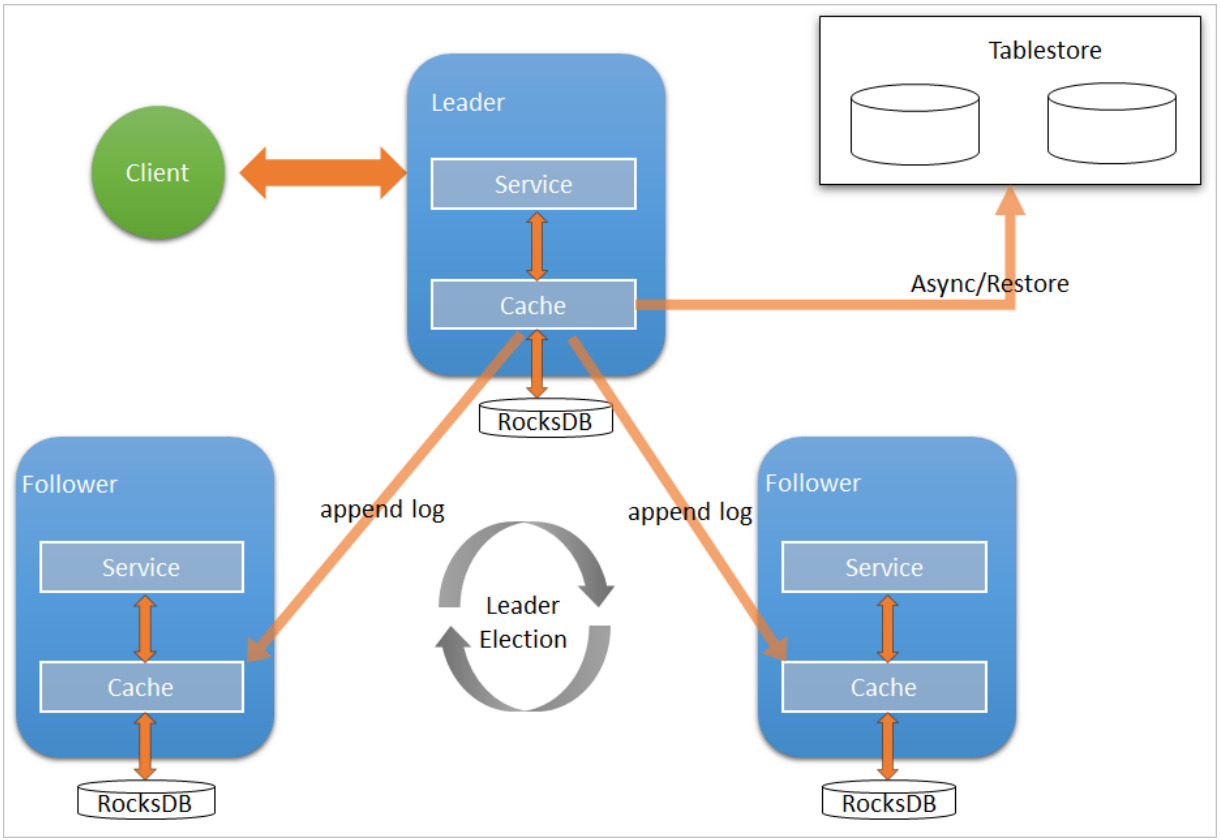


  Note If no deployment modes are available, submit a ticket.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.



Configure a Raft instance as the local storage backend

1. Suspend all the SmartData components of an EMR cluster.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   vi. Select **Stop All Components** from the **Actions** drop-down list in the upper-right corner.
2. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
3. Go to the **namespace** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **namespace** tab.
4. Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. Valid values: ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

Note If you do not need to use Tablestore for remote storage, skip Step 5.

5. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

```
hadoop fs -count jfs://test/
      1596      1482809                 25 jfs://test/
  (Number of folders) (Number of files)
```

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```

```
[RaftPeerImpl]
peer id: [redacted]
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 2335ms)
snapshot_timer: timeout(3600000ms) SCHEDULING(in 150305ms)
storage: [1, 624625]
disk_index: 624625
known_applied_index: 624625
last_log_id: (index=624625,term=2)
first_index_pinned: 624625
state_machine: Idle
last_committed_index: 624625
last_snapshot_index: 0
last_snapshot_term: 0
snapshot_status: IDLE
replicator_25769803789@[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=624261 ic=0
replicator_329853488332[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=623564 ic=0

OtsUploader: _lastStopIndex=624624, _synced=1
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.

```
[RaftPeerImpl]
peer_id: [redacted]:8103:0
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 3382ms)
snapshot_timer: timeout(600000ms) SCHEDULING(in 474855ms)
storage: [1, 153]
disk_index: 153
known_applied_index: 153
last_log_id: (index=153,term=2)
first_index_pinned: 1
state_machine: Idle
last_committed_index: 153
last_snapshot_index: 1
last_snapshot_term: 2
snapshot_status: IDLE
replicator_1116691496965@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0
replicator_3311419785217@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0

[Recovery From OTS Status]
state: FINISH
total rows: 1484409
table `jfs_block_test` 2 rows.
table `jfs_namespace_cache_ns` 1 rows.
table `jfs_namespace_test` 1484406 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

10.2.4. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to

access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8101/* to access the web UI of JindoFS. The web UI of JindoFS 3.0 consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.

  Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

  This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.

  Namespace Info (1)

  Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.

StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.

Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

10.2.5. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

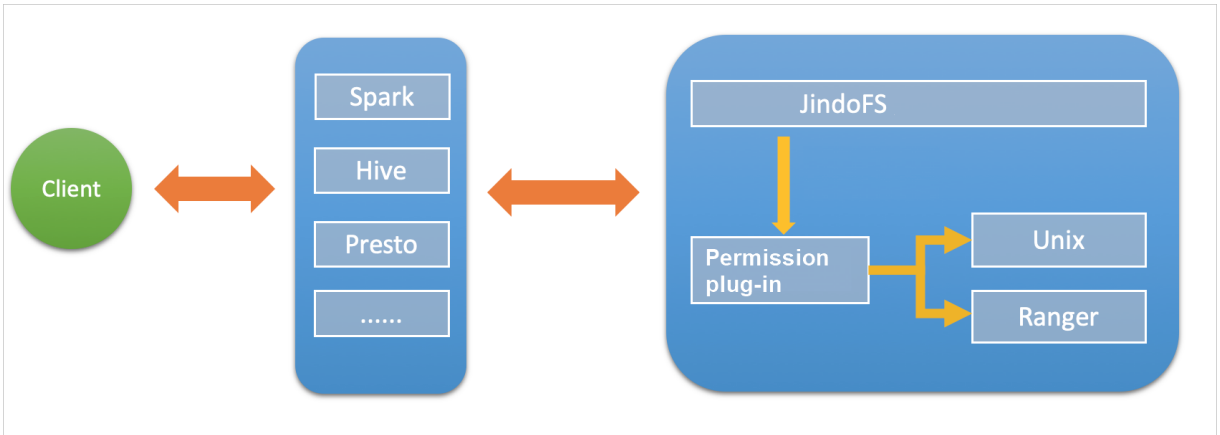
Permission management based on the storage mode:

- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.

- Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.

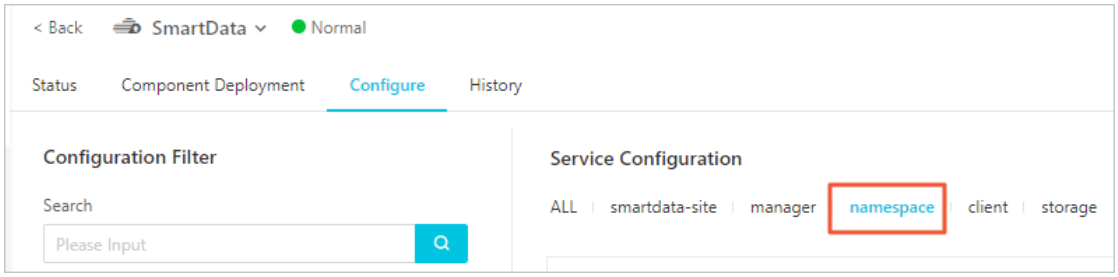
- In cache mode, you can use only Ranger to manage permissions.

  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

   After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

   If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
   iv. Restart Namespace Service.
      a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
      b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
2. Add the HDFS service on the web UI of Ranger and configure related parameters.
   i. Log on to the Ranger web UI.

      For more information, see Overview.
   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true

Parameter	Example
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

10.2.6. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

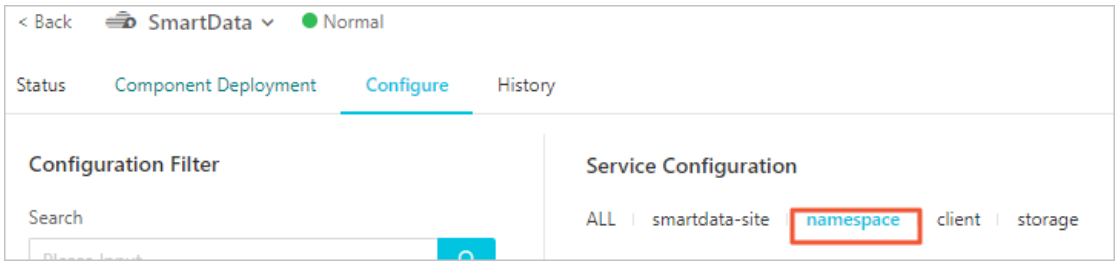
Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.

ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:

   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.

   ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

   iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.

   iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

   v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.

   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.

   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.

   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.

   iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

   For more information, see Configure lifecycle rules.

   v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted] ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted] d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

10.2.7. Analyze metadata offline

In E-MapReduce (EMR) V3.30.0 or later, JindoFS in block storage mode allows you to dump the metadata of an entire namespace to Object Storage Service (OSS) and use Jindo SQL to analyze the metadata.

Context

In Hadoop Distributed File System (HDFS), the metadata is stored in a snapshot file named fsimage. The snapshot file contains the following information about metadata: namespace, file that stores the metadata, blocks, and file system quota. In HDFS, you can run a command to download the fsimage file in the XML format to your on-premises machine and view this file to analyze metadata offline. JindoFS does not require you to download metadata to your on-premises machine.

Upload HDFS metadata to OSS

Run the following command to upload metadata of a specific namespace to OSS:

```
jindo jfs -dumpMetadata <nsName>
```

`<nsName>` indicates the name of the namespace in block storage mode.

For example, upload the metadata of the test-block namespace to OSS and analyze the metadata offline.

```
jindo jfs -dumpMetadata test-block
```

If the following information appears, data is uploaded to OSS and stored as a file in the JSON format.

```
Sucessfully upload namespace metadata to OSS.
```

Metadata upload path

The metadata upload path is the metadataDump sub-directory under *sysinfo* in JindoFS.

For example, if `namespace.sysinfo.oss.uri` is set to `oss://abc/`, metadata is uploaded to the `oss://abc/metadataDump` directory.

Parameter	Description
namespace.sysinfo.oss.uri	The path of the OSS bucket.
namespace.sysinfo.oss.endpoint	The endpoint. An endpoint specified in a different region can also be used.
namespace.sysinfo.oss.access.key	The AccessKey ID of your Alibaba Cloud account.
namespace.sysinfo.oss.access.secret	The AccessKey secret of your Alibaba Cloud account.

Batch information: Metadata from HDFS changes based on the usage. Each time you want to analyze metadata, you must run commands to view the snapshot file of the metadata. Each time you run a Jindo command to upload metadata, a batch number that is generated based on the upload time is used as the root directory to store the metadata. This ensures that the data you uploaded each time is not overwritten. You can delete historical data based on your requirements.

- 1 indicates the configuration path of system information in OSS.
- 2 indicates a namespace.
- 3 indicates a batch number.

Metadata schema

HDFS metadata that is uploaded to OSS is stored as a file in the JSON format. Schema information:

```
{
  "type":"string",          /*Inode type, FILE DIRECTORY*/
  "id": "string",            /*INode id*/
  "parentId" :"string",         /*Parent node ID*/
  "name":"string",         /*Inode name*/
  "size": "int",         /*Inode size, BIGINT*/
  "permission":"int",          /*Permission, INT*/
  "owner":"string",          /*Owner name*/
  "ownerGroup":"string",     /*Owner group name*/
  "mtime":"int",              /*Inode modification time, BIGINT*/
  "atime":"int",              /*The time when the inode was last accessed, BIGINT*/
  "attributes":"string",       /*File-related attributes*/
  "state":"string",            /*Inode status*/
  "storagePolicy":"string",    /*Storage policy*/
  "etag":"string"           /*etag*/
}
```

Use Jindo SQL to analyze metadata

1. Run the command shown in the following figure to start Jindo SQL.

```
[root@emr-header-1 ~]# jindo sql
Spark master: yarn, Application Id: application_1603081647416_0050
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.33 seconds, Fetched 3 row(s)
```

2. Query the tables whose data can be analyzed by using Jindo SQL.
   - Run the `show tables` command to view the tables whose data can be analyzed. Jindo SQL has

two built-in modules audit_log and fs_image for audit and metadata analysis.

- Run the `show partitions fs_image` command to view the partition information of the fs_image table. Each partition contains the data generated by using the `jindo jfs -dumpMetadata` command.

  The following figure shows an example.

```
jindo-sql> show partitions fs_image;
partition
namespace=k[redacted]/datetime=2020_10_20_10_47_14
namespace=k[redacted]/datetime=2020_10_20_10_50_36
namespace=k[redacted]/datetime=2020_10_20_10_52_06
Time taken: 0.045 seconds, Fetched 3 row(s)
```

3. Query and analyze metadata.

   Jindo SQL uses the Spark-SQL syntax. You can use Jindo SQL to query and analyze data in the fs_image table.

   The following figure shows an example.

```
[root@emr-worker-2 hadoop]# jindo sql
Spark master: yarn, Application Id: app[redacted]
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
Time taken: 0.345 seconds, Fetched 3 row(s)
jindo-sql> select * from fs_image limit 10;
atime   attr    etag    id      mtime   name    owner   ownerGroup      parentId        permission      size    state   storagePolicy   type    name
space   datetime
0                       7311076005051899448     1603084070081   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450819 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675495    1603084071350   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450820 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899470     1603084070185   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450821 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       1192276202347928724 9   1603084069581   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450822 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       10769840518872441036    1603084073592   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450823 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       2699389986624511354     1603084068996   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450824 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       1192276202347928730 7   1603084069875   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450825 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       1546468482017665002     1603084072440   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450826 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       16534448041906675460    1603084071170   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450827 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
0                       7311076005051899544     1603084070572   /tpcds/orc/5000/web_returns/wr_returned_date_sk=2450828 root    root    334790833296
5855433 489     0       Finalized       WARM    Directory       kugou   2020_10_20_10_50_36
Time taken: 6.764 seconds, Fetched 10 row(s)
```

   The namespace and datetime columns are added and indicate the namespace name and the timestamp when metadata is uploaded.

   Example: Obtain the number of directories in a specific namespace based on the dumped metadata.

```
jindo-sql>  select count(*) from fs_image where type = "Directory" and namespace="kugou" and datetime="2020_10_20_10_47_14";
count(1)
11837
Time taken: 6.852 seconds, Fetched 1 row(s)
```

Use Hive to analyze metadata

1. Create a table schema in Hive.

   You can use the following command to create a table schema of the metadata that is used for queries in Hive:

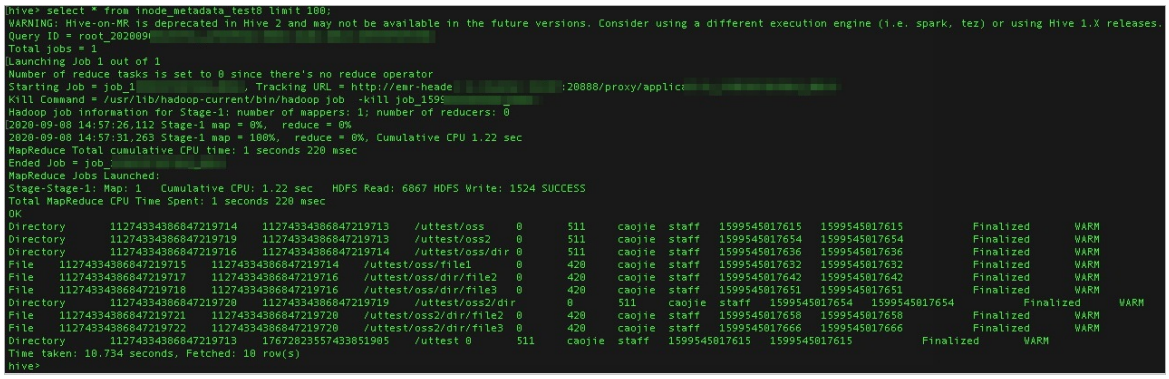
```
CREATE EXTERNAL TABLE `table_name`
(`type` string,
 `id` string,
 `parentId` string,
 `name` string,
 `size` bigint,
 `permission` int,
 `owner` string,
 `ownerGroup` string,
 `mtime` bigint,
 `atime` bigint,
 `attr` string,
 `state` string,
 `storagePolicy` string,
 `etag` string)
 ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
 STORED AS TEXTFILE
 LOCATION 'The OSS path to which file data is uploaded';
```

2. Use Hive to analyze data offline.

   After you create a Hive table, you can use Hive SQL to analyze metadata.

```
select * from table_name limit 200;
```

   The following figure shows an example.



10.3. JindoFS in cache mode

10.3.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in Object Storage Service (OSS) based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

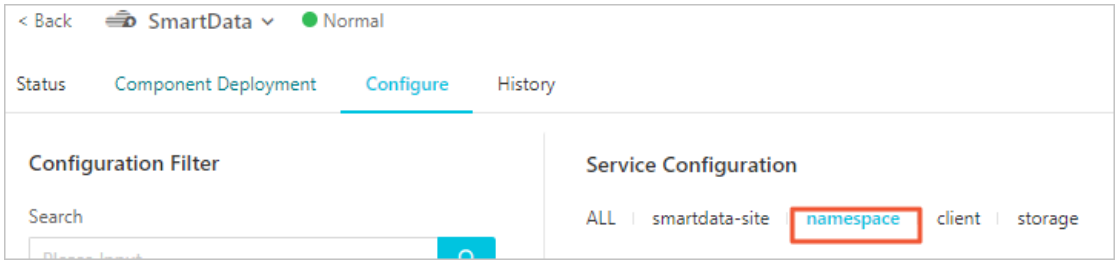
  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. Click **OK**.
5. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
6. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **client** tab.
2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   You do not need to restart the SmartData service to apply the new configurations.
3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

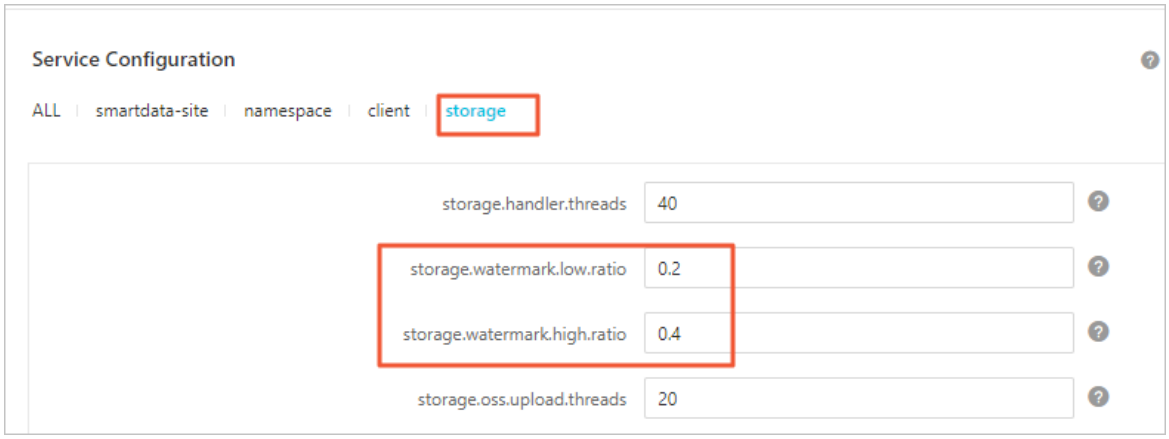
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you specify. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify the related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to specify an AccessKey pair. In other cases, you must specify an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you choose for accessing files in OSS:

- OSS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.
  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss.accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss.endpoint	The endpoint of the OSS bucket that serves as the storage backend.

> **Note** You can set these parameters to the values of the parameters in an EMR version earlier than EMR V3.30.0.

- JFS Scheme
  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the page that appears, click the **Configure** tab. In the Service Configuration section, click the **namespace** tab.
  ii. Set **jfs.namespaces** to **test**.
  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize the cache performance. After you configure the parameters, you do not need to restart the SmartData service to apply the new configurations.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. You can use this parameter to prevent upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If an excessive amount of memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

10.3.2. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

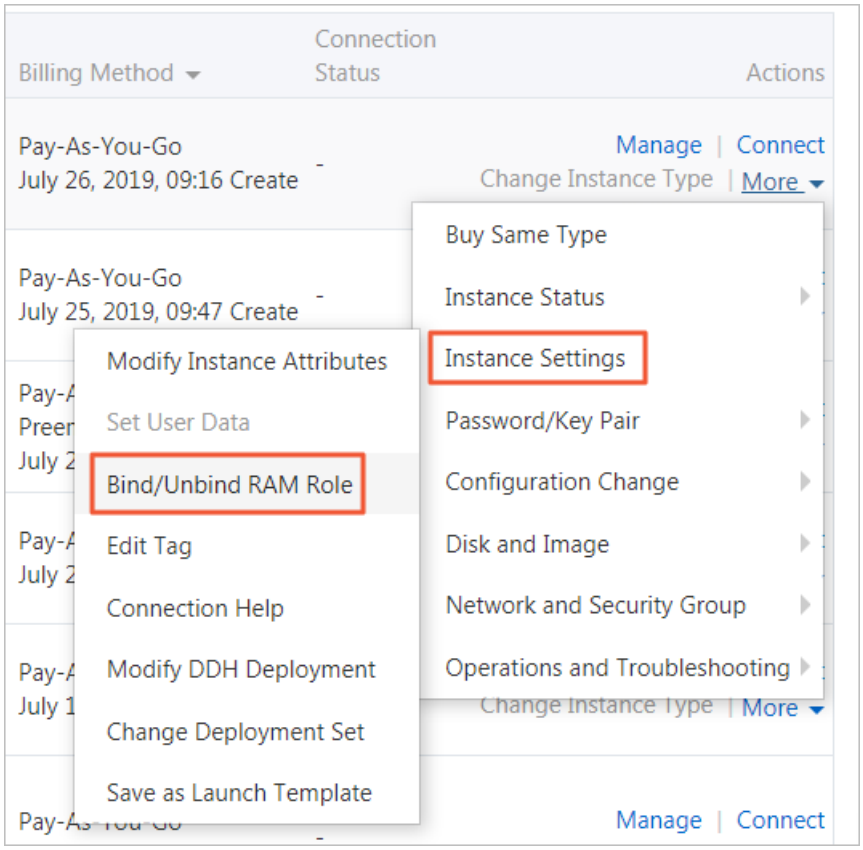
1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.
4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

   ```
   hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
   ```

2. Access OSS by using the FileSystem interface of Hadoop.

   Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

10.3.3. Access the web UI of JindoFS

You can view information about clusters on the web UI of JindoFileSystem (JindoFS). The information includes the running mode, namespace, storage service information, and startup status.

Prerequisites

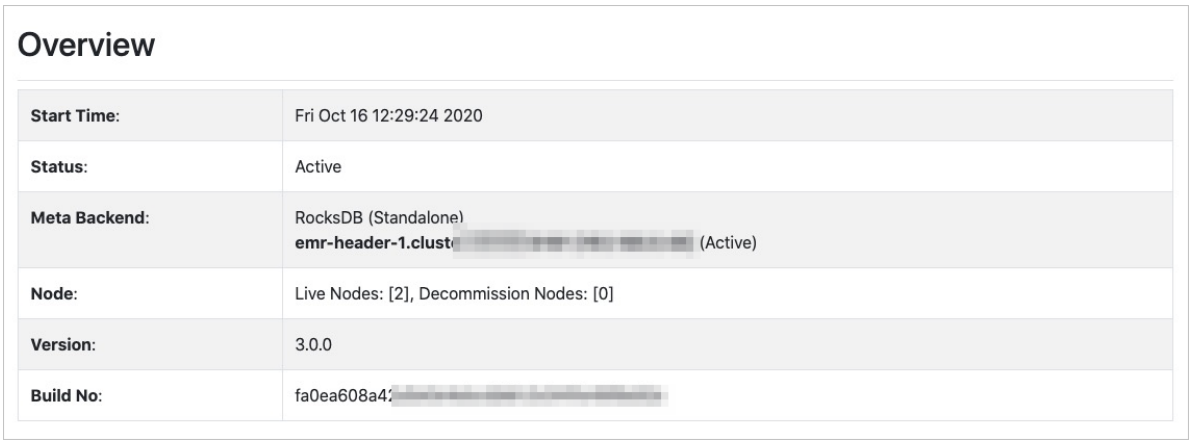
You can access the web UI by using an SSH tunnel. For more information, see Create an SSH tunnel to access web UIs of open source components.

Access the web UI of JindoFS

After an SSH tunnel is established, you can visit *http://emr-header-1:8101/* to access the web UI of JindoFS. The web UI of JindoFS 3.0 consists of the following sections: Overview, Namespace Info, StorageService, and Advanced.

- Overview

  This section provides information that includes Start Time, Status, Meta Backend, Node, and Version.



Overview

Start Time:	Fri Oct 16 12:29:24 2020
Status:	Active
Meta Backend:	RocksDB (Standalone) **emr-header-1.clust** (Active)
Node:	Live Nodes: [2], Decommission Nodes: [0]
Version:	3.0.0
Build No:	fa0ea608a4

- Namespace Info

This section provides information that includes Namespaces, Mode, Backend URI, and Summary. In block storage mode, you can view the following statistics of a namespace: the number of directories, the number of files, and total file size.

Namespace Info (1)

Namespace: jfs://test/

Namespaces:	test
Mode:	BLOCK_MODE
Backend URI:	oss://
Summary:	Directory Count: [2], File Count: [47701], File Size: [166725951374], Task Count: [0], Computed at 2020-10-20 10:49:27

- StorageService

  This section provides a list of Storage Services in a cluster and the information about each of the Storage Services. The information includes Node, Status, Capacity, Last contact, Start Time, Storage Id, Version, and Build Version.

StorageService (2)

Node	Status	Capacity	Last contact	Start Time	Storage Id	Version	Build Version
emr-worker-2.cluster-	Healthy	125.70 GB/235.63 GB	2 sec	Fri Oct 16 12:29:30 2020	0	3.0.0	fa0ea608a4
emr-worker-1.cluster-	Healthy	124.69 GB/235.63 GB	1 sec	Fri Oct 16 12:29:25 2020	5	3.0.0	fa0ea608a4

You can click a Node value to view the space usage of each disk.

Overview

Start Time:	Fri Oct 16 12:29:25 2020
Version:	3.0.0
Build Version:	fa0ea608a42a5e0e4ebcdbbfc3c041fe49f8e82e

Storage Lists (4)

Directory	StorageType	Capacity	Last Eviction Time	Partition Id
/mnt/d	Disk	28.49 GB/58.91 GB	Fri Oct 16 20:23:51 2020	31
/mnt/d	Disk	19.59 GB/58.91 GB	Fri Oct 16 20:25:43 2020	21
/mnt/d	Disk	51.95 GB/58.91 GB	Fri Oct 16 20:19:48 2020	1
/mnt/d	Disk	24.67 GB/58.91 GB	Fri Oct 16 20:24:40 2020	11

- Advanced

  This section is used for JindoFS developers to troubleshoot issues.

10.3.4. Manage JindoFS permissions

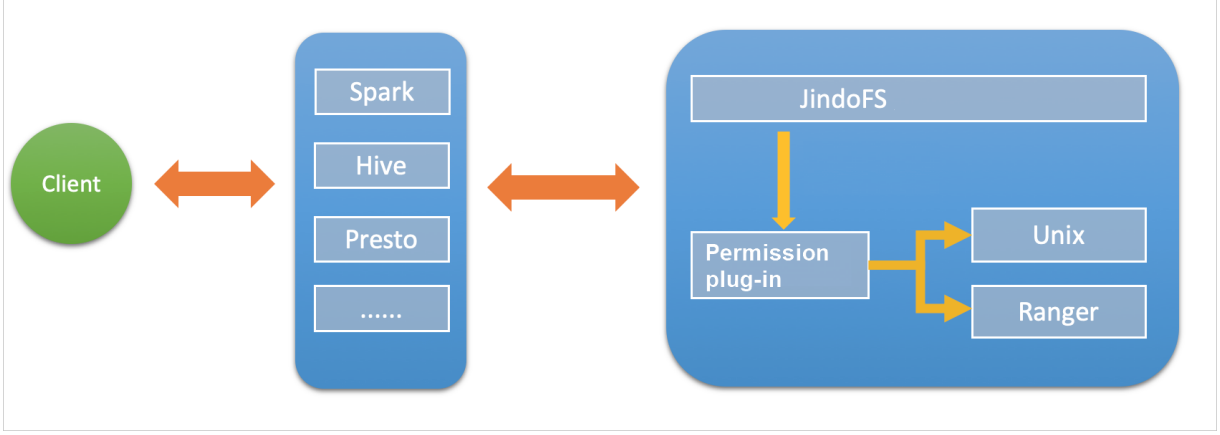
This topic describes how to manage the permissions of JindoFS in block storage mode or in cache mode. You cannot switch between the block storage mode and cache mode.

Context

Permission management based on the storage mode:

- In block storage mode, you can run UNIX commands or use Ranger to manage permissions:
  - UNIX allows you to grant the rwxrwxrwx permission on files and configure owners and groups of files.
  - Ranger allows you to perform complex or advanced operations. For example, use wildcards in paths.
- In cache mode, you can use only Ranger to manage permissions.

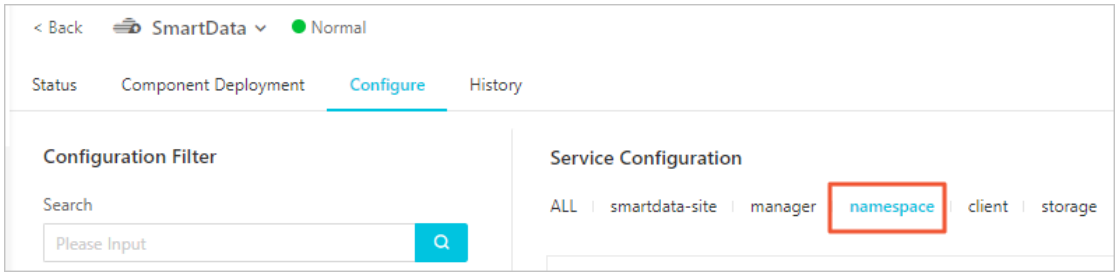
  You can perform complex or advanced operations. For example, use wildcards in paths.



Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the namespace tab for the SmartData service.
   i. Click the **Configure** tab.

ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration** in the upper-right corner. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix, and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the error shown in the following figure is returned.

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

Before you can use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of EMR and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

1. Configure Ranger as a permission management method in JindoFS.
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger, and click OK.
   iii. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.

iv. Restart Namespace Service.

a. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.

b. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.

2. Add the HDFS service on the web UI of Ranger and configure related parameters.

i. Log on to the Ranger web UI.

For more information, see Overview.

ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters that are described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}. Example: jfs-test.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}/*.
Authorization Enabled	Retain the default value No.
Authentication Type	Retain the default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you must also enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information about user groups that are synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.

2. In the **Add Configuration Item** dialog box, configure the LDAP parameters described in the following table and click **OK**.

   Configure the parameters based on the configurations in open source HDFS. For more information, see core-default.xml.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Choose **Actions > Restart All Components** in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**. In the Confirm message, click OK.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

10.3.5. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

Notice

- The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
- After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
- Hive and Presto jobs do not use Hadoop Job Committer.
- In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.
6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.

3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.

ii. Click the **Configure** tab.

iii. In the **Service Configuration** section, click the **smartdata-site** tab.

5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.

   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.

   ii. Click the **Configure** tab.

   iii. In the **Service Configuration** section, click the **smartdata-site** tab.

2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

   The default value is 8.

3. Save the configuration.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

   iii. Click **OK**.

10.3.6. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode or in cache mode. Jindo AuditLog records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.30.0 version is created. For information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For information about how to create an OSS bucket, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

The following table describes the parameters of an audit log that is recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Example:

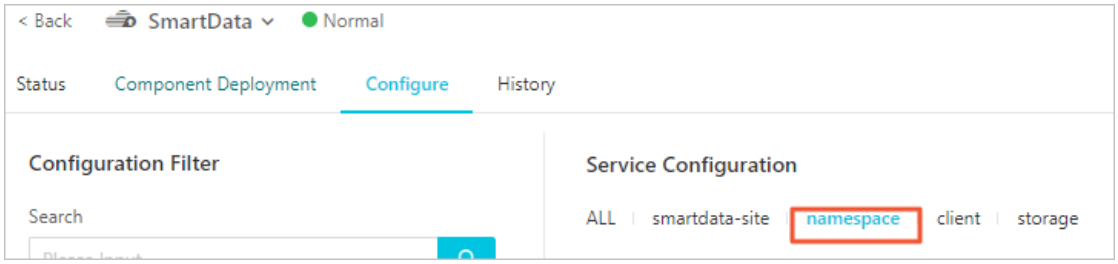
```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. On the **namespace** tab for the SmartData service, click **Custom Configuration** in the upper-right corner.
   ii. In the **Add Configuration Item** dialog box, configure the parameters that are described in the following table.

Parameter	Description	Required
jfs.namespaces.{ns}.auditlog.enable	Specifies whether to enable AuditLog for specific namespaces. Valid values: ■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.sysinfo.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.sysinfo.oss.access.key	The AccessKey ID used to access the OSS bucket.	No
namespace.sysinfo.oss.access.secret	The AccessKey secret used to access the OSS bucket.	No
namespace.sysinfo.oss.endpoint	The endpoint of the OSS bucket.	No

   iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.
   iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.

v. In the **Confirm** message, click **OK**.

4. Restart Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.
5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.
   i. Log on to the OSS console.
   ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the created bucket.
   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.
   iv. Click **Create Rule**. In the **Create Rule** panel, configure the parameters.

   For more information, see Configure lifecycle rules.
   v. Click **OK**.

Analyze log files

JindoFS allows you to use SQL queries to analyze the log files generated by AuditLog. You can use SQL queries to analyze the most active commands or IP addresses in the tables. The analysis command is `jindo sql`.

The `jindo sql` command uses the Spark SQL syntax, and is embedded with the audit_log_source, audit_log, and fs_image tables. The audit_log_source table stores the original data of AuditLog. The audit_log table stores the cleansed data of AuditLog. The fs_image table stores fsimage log data. The audit_log_source and fs_image tables are partitioned tables. Usage:

- Use `jindo sql --help` to query the settings of the parameters described in the following table.

Parameter	Description
-f	Specifies the SQL file to run.
-i	Automatically executes the initialization SQL script after the jindo sql command is run.

- Use `show partitions table_name` to obtain all partitions.
- Use `desc formatted table_name` to view the table structure.

The Jindo sql command is developed based on Spark. Therefore, this command may have small-sized initial resources. You can use the JINDO_SPARK_OPTS environment variable to modify startup parameters for this command. Sample modification:

```
 export JINDO_SPARK_OPTS="--conf spark.driver.memory=4G --conf spark.executor.instances=20
--conf spark.executor.cores=5 --conf spark.executor.memory=20G"
```

Examples:

- Run the following command to display tables:

```
show tables;
```

```
jindo-sql> show tables;
database        tableName       isTemporary
default audit_log       false
default audit_log_source        false
default fs_image        false
```

- Run the following command to display partitions in the audit_log_source table:

```
show partitions audit_log_source;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> show partitions audit_log_source;
partition
date=2020-10-20
Time taken: 0.031 seconds, Fetched 1 row(s)
```

- Run the following commands to query data:

```
select * from audit_log_source limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log_source limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 10:50:11.950 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:06.445 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
st=null perm=hadoop:hadoop:rwxrwxr-x    2020-10-20
2020-10-20 11:26:06.469 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:11.295 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
2020-10-20 11:26:11.320 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=listFileletRequest  src=jfs://kugou/[redacted]=
null    perm=null       2020-10-20
2020-10-20 11:26:14.368 allowed=true    ugi=root (auth:SIMPLE)  ip=192.[redacted]  ns=kugou        cmd=getFileStatusRequest        src=jfs:[redacted]  d
s       dst=null        perm=root:root:rwxr-x--x        2020-10-20
```

```
select * from audit_log limit 10;
```

The information similar to that shown in the following figure is returned.

```
jindo-sql> select * from audit_log limit 10;
datetime        allowed ugi     ip      ns      cmd     src     dst     perm    date
2020-10-20 10:50:11.924 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 10:50:11.950 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:06.445 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]ll    hadoop:hadoop:rwxrwx
r-x     2020-10-20
2020-10-20 11:26:06.469 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]ll    null    2020-10-20
2020-10-20 11:26:11.295 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:11.320 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:14.368 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:14.393 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
2020-10-20 11:26:16.230 true    root (auth:SIMPLE)      192.[redacted]  kugou   getFileStatusRequest    jfs://k[redacted]      null    root:root:rw
xr-x--x 2020-10-20
2020-10-20 11:26:16.255 true    root (auth:SIMPLE)      192.[redacted]  kugou   listFileletRequest      jfs://k[redacted]      null    null    2020
-10-20
Time taken: 1.918 seconds, Fetched 10 row(s)
```

- Run the command shown in the following figure to collect statistics on the use frequencies of different commands on October 20, 2020.

```
jindo-sql> select cmd ,count(*) from audit_log where date="2020-10-20" group by cmd order by cmd;
cmd     count(1)
getFileStatusRequest    387
listFileletRequest      387
Time taken: 5.767 seconds, Fetched 2 row(s)
```

10.3.7. Use a credential provider

You can use a credential provider to save an encrypted AccessKey pair into a file. This prevents the leak of the AccessKey pair.

Context

E-MapReduce (EMR) V3.30.0 provides a JindoOSS credential provider. You can use a Hadoop credential provider to save an encrypted AccessKey pair into a file. This prevents the issue that the AccessKey pair is transmitted in plaintext. You can select an appropriate JindoOSS credential provider based on your requirements.

Configure a JindoOSS credential provider

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **smartdata-site** tab.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **smartdata-site** tab.
3. Add configuration information.
   i. In the upper-right corner of the **smartdata-site** tab, click **Custom Configuration**.

ii. In the **Add Configuration Item** dialog box, add the parameter described in the following table.

Parameter	Description
fs.jfs.cache.credentials.provider	Specifies implementation classes of com.aliyun.emr.fs.auth.AliyunCredentialsProvider. Separate multiple classes with commas (,). Read credential values in sequence until a valid credential value is found. Example: `com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider, com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider,com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider` .

You can select different credential providers based on your requirements. The following providers are supported:

- TemporaryAliyunCredentialsProvider
- SimpleAliyunCredentialsProvider
- EnvironmentVariableCredentialsProvider
- InstanceProfileCredentialsProvider

Use a Hadoop credential provider to store AccessKey pair information

> **Note** For more information about Hadoop credential providers, see CredentialProvider API Guide.

fs.jfs.cache.oss.accessKeyId, **fs.jfs.cache.oss.accessKeySecret**, and **fs.jfs.cache.oss.securityToken** can be stored into Hadoop credential providers.

Use a command that is provided by Hadoop to store AccessKey pair and security token information into a credential file. Sample command:

```
hadoop credential <subcommand> [options]
```

For example, store AccessKey pair and token information into a jceks file. jceks indicates a Java keystore provider. You can protect the file by using file permissions or you can specify a key to encrypt the information you want to store. If you do not specify a key, the default string is used for encryption.

```
hadoop credential create fs.jfs.cache.oss.accessKeyId -value AAA -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.accessKeySecret -value BBB -provider jceks://file/root/oss.jceks
hadoop credential create fs.jfs.cache.oss.securityToken -value  CCC -provider jceks://file/root/oss.jceks
```

After a credential file is generated, you must configure the parameter described in the following table to specify the provider type and location.

Parameter	Description
fs.jfs.cache.oss.security.credential.provider.path	The path used to store the credential file that stores AccessKey pair information. For example, you can set this parameter to *jceks://file/${user.home}/oss.jceks*, which indicates that the *oss.jceks* file is stored in the home directory.

TemporaryAliyunCredentialsProvider

This provider is suitable for scenarios in which AccessKey pair and security token information that has a validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.credentials.provider	**com.aliyun.emr.fs.auth.TemporaryAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.
fs.jfs.cache.oss.securityToken	The temporary security token that is used to access OSS.

SimpleAliyunCredentialsProvider

This provider is suitable for scenarios in which an AccessKey pair with a long validity period is used to access OSS.

Parameter	Description
fs.jfs.cache.credentials.provider	**com.aliyun.emr.fs.auth.SimpleAliyunCredentialsProvider**
fs.jfs.cache.oss.accessKeyId	The AccessKey ID used to access OSS.
fs.jfs.cache.oss.accessKeySecret	The AccessKey secret used to access OSS.

EnvironmentVariableCredentialsProvider

To use this provider, you must configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.credentials.provider	**com.aliyun.emr.fs.auth.EnvironmentVariableCredentialsProvider**
ALIYUN_ACCESS_KEY_ID	The AccessKey ID used to access OSS.
ALIYUN_ACCESS_KEY_SECRET	The AccessKey secret used to access OSS.

Parameter	Description
ALIYUN_SECURITY_TOKEN	The temporary security token that is used to access OSS. Note This parameter is required only when you configure a token that has a validity period.

InstanceProfileCredentialsProvider

This provider does not require an AccessKey pair. You can access OSS in password-free mode.

Parameter	Description
fs.jfs.cache.credentials.provider	com.aliyun.emr.fs.auth.InstanceProfileCredentialsProvider

10.4. JindoTable

10.4.1. Use JindoTable

JindoTable is used to implement tiered storage, optimize table files, and collect data statistics based on the popularity of tables or partitions. This topic describes how to use JindoTable.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your on-premises machine.
- An EMR cluster of the 3.30.0 version or later is created. For more information about how to create a cluster, see Create a cluster.

Use JindoTable

Common commands:

- -accessStat
- -cache
- -archive
- -unarchive
- -uncache
- -status
- -optimize
- -showTable
- -showPartition
- -listTables
- -dumpmc

> **Notice** Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

-accessStat

- Syntax

  jindo table -accessStat {-d} *<days>* **{-n}** <topNums>

- Description

  This command is used to query the access records in which tables or partitions are visited most in a specified time range.

  <days> and <topNums> must be positive integers. If <days> is 1, all the access records generated from 00:00 (local time) on the current day to the current time are queried.

- Example: Query the 20 access records of tables or partitions that are most frequently visited within the last seven days.

```
jindo table -accessStat -d 7 -n 20
```

-cache

- Syntax

  jindo table -cache {-t} <dbName.tableName> **[-p]** <partitionSpec> **[-pin]**

- Description

  This command is used to cache data of specified tables or partitions to local disks.

  The data of the tables or partitions must be stored in Object Storage Service (OSS) or JindoFileSystem (JindoFS). Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`. When you specify `-pin`, if the cache space is insufficient, do not delete related data if possible.

- Example: Cache data of the db1.t1 table that is generated on March 16, 2020 to local disks.

```
jindo table -cache -t db1.t1 -p date=2020-03-16
```

-uncache

- Syntax

  jindo table -uncache {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to delete cached data of specified tables or partitions from local disks.

  The data of the tables or partitions must be stored in OSS or JindoFS. Specify tables in the format of `database.table`. Specify partitions in the format of `partitionCol1=1,partitionCol2=2,...`.

- Examples:
  - Delete cached data of the db1.t2 table from local disks.

```
jindo table -uncache -t db1.t2
```

- Delete cached data of the db1.t1 table from local disks.

```
jindo table -uncache -t db1.t1 -p date=2020-03-16,category=1
```

-archive

- Syntax

  jindo table -archive {-a|i} {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to lower the level of the policy that is used to store data of specified tables or partitions. By default, the Archive storage class is used.

  To use the Infrequent Access (IA) storage class, add -i to the command. Specify tables in the format of database.table. Specify partitions in the format of 'partitionCol1=1,partitionCol2=2,...'.

- Example: Cache data of the db1.t1 table to local disks.

```
jindo table -archive -t db1.t1 -p date=2020-10-12
```

-unarchive

- Syntax

  jindo table -unarchive [-o|-i] {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to change the storage class from Archive to Standard.

  If `-o` is added to the command, an Archived object is temporarily restored. If `-i` is added to the command, an Archived object is changed to an IA object.

- Example:

```
jindo table -unarchive -o -t db1.t1 -p date=2020-03-16,category=1
```

```
jindo table -unarchive -i -t db1.t2
```

-status

- Syntax

  jindo table -status {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to view the data storage status of specified tables or partitions.

- Examples:
  - View the data storage status of the db1.t2 table.

```
jindo table -status -t db1.t2
```

  - View the data storage status of the db1.t1 table on March 16, 2020.

```
jindo table -status -t db1.t1 -p date=2020-03-16
```

-optimize

- Syntax

  jindo table -optimize {-t} <dbName.tableName>

- Description

  This command is used to optimize the data organization of tables at the storage layer.

- Example: Optimize the data organization of the db1.t1 table at the storage layer.

```
jindo table -optimize -t db1.t1
```

-showTable

- Syntax

  jindo table -showTable {-t} <dbName.tableName>

- Description

  This command is used to display all partitions in a partitioned table or display the data storage of a non-partitioned table.

- Example: Display all partitions in the db1.t1 partitioned table.

```
jindo table -showTable -t db1.t1
```

-showPartition

- Syntax

  jindo table -showPartition {-t} <dbName.tableName> **[-p]** <partitionSpec>

- Description

  This command is used to display the data storage of partitions.

- Example: Display the data storage of all partitions in the db1.t1 partitioned table on October 12, 2020.

```
jindo table -showPartition -t db1.t1 -p date=2020-10-12
```

-listTables

- Syntax

  jindo table -listTables [-db] <dbName>

- Description

  This command is used to display all tables in a specified database. If you do not specify `[-db]`, tables in the default database are displayed.

- Examples:
  - Display tables in the default database.

```
jindo table -listTables
```

- Display tables in the db1 database.

```
jindo table -listTables -db db1
```

-dumpmc

- Syntax

jindo table -dumpmc {-i} <accessId> **{-k}** <accessKey> **{-m}** <numMaps> **{-t}** <tunnelUrl> **{-project}** <projectName> **{-table}** <tablename> **{-p}** <partitionSpec> **{-f}** <csv|tfrecord> **{-o}** <outputPath>

Parameter	Description	Required
-i	The AccessKey ID of your Alibaba Cloud account.	Yes
-k	The AccessKey secret of your Alibaba Cloud account.	Yes
-m	The number of map tasks.	Yes
-t	The VPC Tunnel endpoint of MaxCompute.	Yes
-project	The name of the MaxCompute project.	Yes
-table	The name of the MaxCompute table.	Yes
-p	The partition information. Example: `pt=xxx`. Separate multiple partitions with commas (,), such as `pt=xxx,dt=xxx`.	No
-f	The file format. Valid values: ○ tfrecord ○ csv	Yes
-o	The destination path.	Yes

- Description

This command is used to dump MaxCompute tables to an EMR cluster or OSS. The formats CSV and TFRECORD are supported.

- Examples:
  - Dump a MaxCompute table in the TFRECORD format to an EMR cluster.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o /tmp/outputtf1 -f tfrecord
```

- Dump a MaxCompute table in the CSV format to OSS.

```
jindo table -dumpmc -m 10 -project mctest_project -table t1 -t http://dt.xxx.maxcompute
.aliyun-inc.com -k xxxxxxxxx -i XXXXXX -o oss://bucket1/tmp/outputcsv -f csv
```

10.4.2. Use EMR JindoCube

JindoCube is supported in E-MapReduce (EMR) V3.24.0 and later. This topic describes how to install, deploy, and use JindoCube in EMR.

Prerequisites

A table or view is created.

Overview

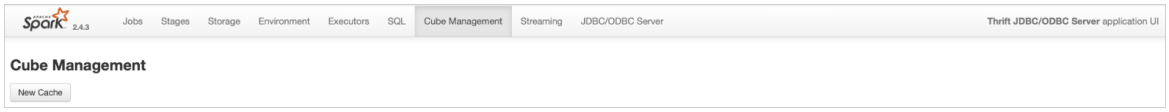
JindoCube is an advanced feature of EMR Spark. It implements pre-calculation to accelerate data processing. It improves query performance by dozens or even hundreds of times. You can persist data of a view to a data source supported by EMR Spark, such as HDFS or Object Storage Service (OSS). EMR Spark automatically discovers available persistent data and optimizes the execution plans for the data. This process is completely visible to users. JindoCube applies to business scenarios where a fixed query mode is used. JindoCube pre-calculates and pre-organizes data to accelerate data queries. For example, you can use JindoCube to implement multidimensional online analytical processing (MOLAP), generate data reports, create data dashboards, and synchronize data across clusters.

Install and deploy JindoCube

JindoCube is an advanced feature of EMR Spark. You can use JindoCube to accelerate all Dataset, DataFrame API, and SQL tasks that are submitted by using EMR Spark. No additional component deployment or maintenance is required.

1. Manage JindoCube on the Spark web UI.

   You can manage JindoCube on the Spark web UI. For example, you can create, delete, or update JindoCube caches. After you create a JindoCube cache, it automatically accelerates all Spark queries in the cluster. You can use the **spark.sql.cache.tab.display** parameter to specify whether to display the Cube Management tab on the Spark web UI. You can specify this parameter on the configuration page of the Spark service in the EMR console or in a Spark task submission command. The default value of this parameter is **false**.

   You can also use the **spark.sql.cache.useDatabase** parameter to create a database based on your business requirements and store the view for which you want to create a cache in this database. If partitioned tables are used, you can use the **spark.sql.cache.cacheByPartition** parameter to specify whether to cache data by partition.

Parameter	Description	Example
spark.sql.cache.tab.display	Specifies whether to display the Cube Management tab.	true

Parameter	Description	Example
spark.sql.cache.useDatabase	The databases where cubes are stored.	db1,db2,dbn
spark.sql.cache.cacheByPartition	Specifies whether to store cubes by partition.	true

2. Optimize Spark queries.

   You can use the **spark.sql.cache.queryRewrite** parameter to specify whether to use a JindoCube cache to accelerate Spark queries. You can specify this parameter at the cluster, session, or SQL statement level. The default value is **true**.

Use JindoCube

1. Create a JindoCube cache.
   i. Log on to the Alibaba Cloud EMR console by using your Alibaba Cloud account.
   ii. Click the **Cluster Management** tab.
   iii. Find the cluster where you want to create a JindoCube cache and click the cluster ID.
   iv. In the left-side navigation pane, click **Connect Strings**.
   v. On the **Public Connect Strings** page, click the link of YARN UI to go to the YARN web UI as a Knox user.

      For more information about Knox, see Knox.
   vi. Click **ApplicationMaster** in the row where **Application Type** is **SPARK** and **Name** is **Thrift JDBC/ODBC Server**.
   vii. On the Spark web UI, click the **Cube Management** tab.
   viii. Click **New Cache**.

      Select a table or view, and click raw cache or cube cache in the action column. You can create two types of caches:

- Raw cache: persists the data of a table or view based on the cache configurations.

  When you create a raw cache, specify the parameters described in the following table.

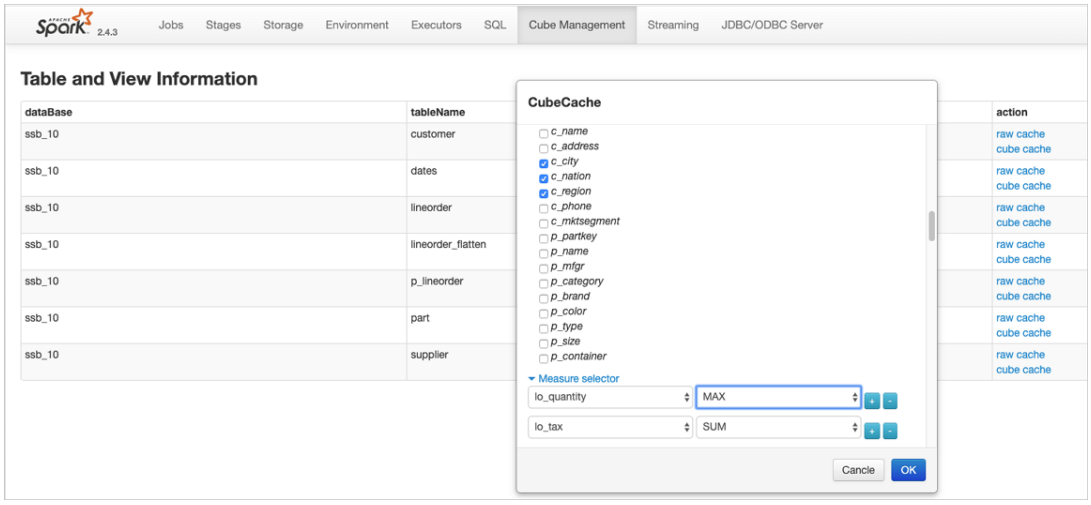
Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Column Selector	The columns from which you want to cache data.	Yes
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No

- Cube cache: builds a cube based on the raw data of a table or view and persists the cube data.

  When you create a cube cache, specify the parameters described in the following table.

Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Dimension Selector	The dimension fields that are used to build a cube.	Yes
Measure Selector	The measure fields and pre-calculation functions that are used to build a cube.	Yes

Parameter	Description	Required
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No



JindoCube builds a cube based on specified dimensions and measures. The following SQL statement shows the configurations of the cube cache in the preceding figure:

```
SELECT c_city, c_nation, c_region, MAX(lo_quantity), SUM(lo_tax)
FROM lineorder_flatten
GROUP BY c_city, c_nation, c_region;
```

JindoCube calculates the finest-grained combination of dimensions. Powered by the onsite computing capability of Spark, JindoCube implements re-aggregation to optimize the queries where a coarse-grained combination of dimensions is used. You must use the pre-calculation functions supported by JindoCube to define a cube cache. The following table lists the supported pre-calculation functions and the aggregate functions that correspond to them.

Aggregate function	Pre-calculation function
COUNT	COUNT
SUM	SUM
MAX	MAX
MIN	MIN
AVG	COUNT and SUM
COUNT (DISTINCT)	PRE_COUNT_DISTINCT
APPROX_COUNT_DISTINCT	PRE_APPROX_COUNT_DISTINCT

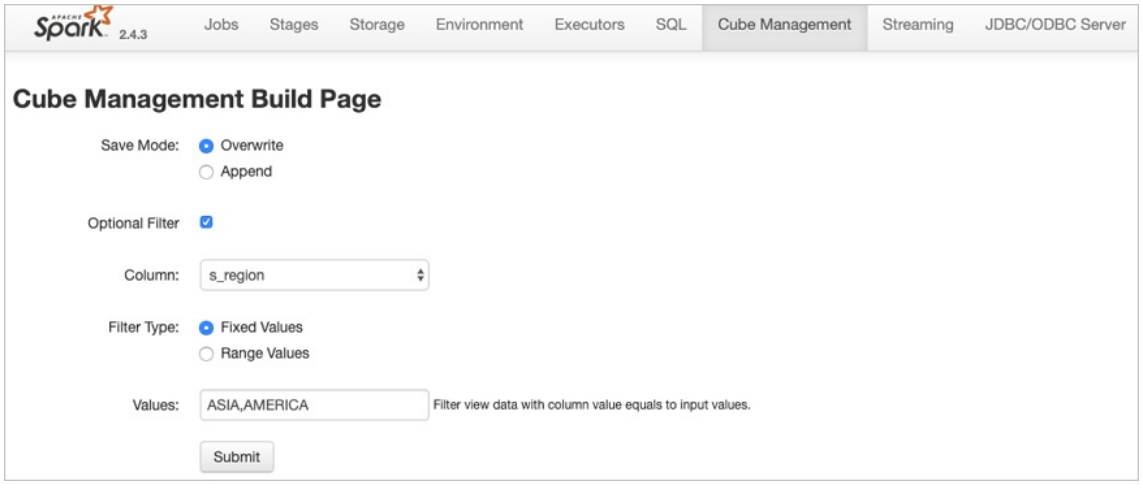
All created caches are displayed on the **Cube Management** tab. You can click **Detail** for a cache to view its details, including its basic information, partition information, building information, and building history.

2. Build a JindoCube cache.

   During the creation of a cache, metadata is prepared, but no data is persisted. Therefore, after you create a cache, you must continue to build the cache. This way, the related data can be persisted to permanent storage, such as HDFS or OSS. In addition, the source table of a cache may be appended with new data or updated. In this case, you must update the data in the cache to ensure data consistency. You can use one of the following methods to build a cache:

   - Manually build a cache

     Click Build Cache to manually build a cache. On the page that appears, configure the parameters, as shown in the following figure.



     The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.

Optional Filter	You can select this parameter to add filter conditions. This way, JindoCube filters data based on the filter conditions before it persists the data.
	■ Column: the field based on which data is filtered.
	■ Filter Type: specifies whether to filter data by fixed values or by value range.
	  ■ Fixed Values: If this option is selected, specify one or more values. Separate values with commas (,).
	  ■ Range Values: If this option is selected, specify a minimum value and a maximum value. You can leave the maximum value empty. In this case, the value range is the minimum value and all values greater than it.

The following SQL statement shows the configurations of the raw cache for the lineorder_flatten view in the preceding figure:

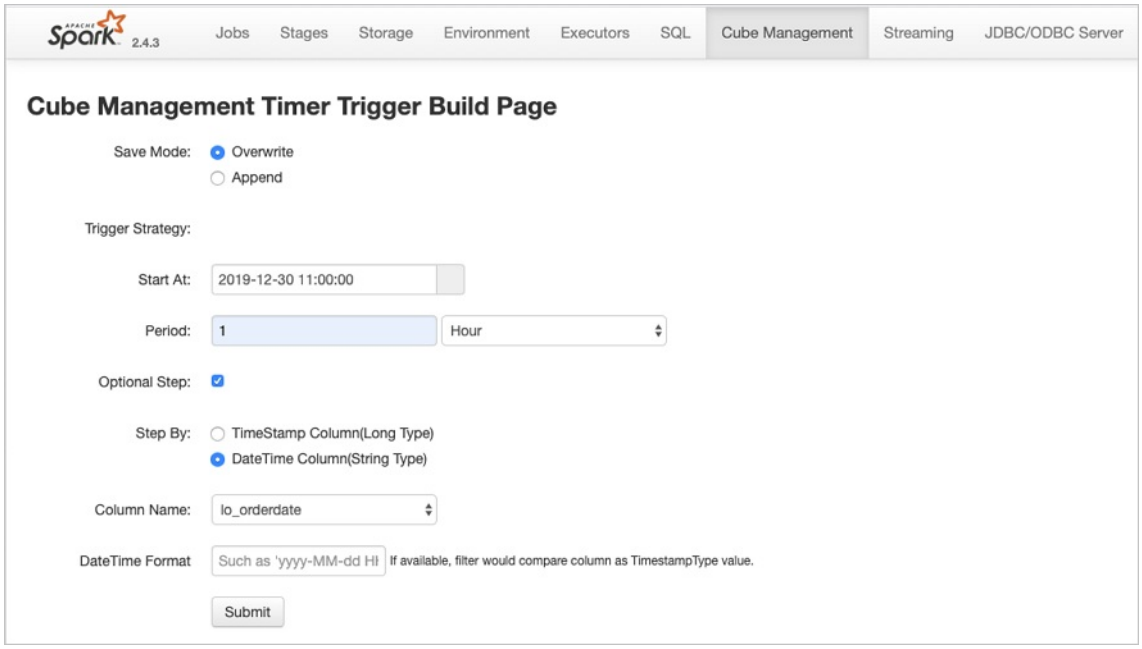
```
SELECT * FROM lineorder_flatten
WHERE s_region == 'ASIA' OR s_region == 'AMERICA';
```

Click **Submit** to submit the cache building task to a Spark cluster. You can view the task status in the Build Information section of the cache details page. After the task is complete, you can view the cache information in the Build History section.

> **Note** Spark writes the cached data to a specified directory in the same way common Spark tasks write data to a table or directory. If multiple users concurrently build a cache whose data format is Parquet, JSON, or ORC, the cached data may be inaccurate and invalid. If concurrent cache building operations or updates are inevitable, we recommend that you use a data format that supports concurrent writes, for example, Delta.

- Configure periodic cache building

  Click Trigger Period Build to configure a periodic cache update policy to ensure data consistency between the cache and source table. On the page that appears, configure the parameters, as shown in the following figure.



The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.
Trigger Strategy	Specifies the start time of the first building task, and the building interval. ■ Start At: Select or enter the time you want to trigger the first building task. Format: yyyy-MM-dd hh:mm:ss. ■ Period: Specify the interval at which cache building tasks are triggered.

Parameter	Description
Optional Step	Specifies the filter conditions for cache building. JindoCube updates the cache in incremental mode based on the filter conditions and interval. If you do not select Optional Step, JindoCube updates the entire cache each time. ■ Step By: Specify the type of the field that you want to update in incremental mode. TimeStamp Column(Long Type) indicates a timestamp field of the LONG type. DateTime Column(String Type) indicates a datetime field of the STRING type. ■ Column Name: Specify the name of the field that you want to update in incremental mode.

On the cache details page, you can view the configured cache update policy. You can click Cancel Period Build to cancel periodic updates. You can view the information of a complete building task in the Build History section.

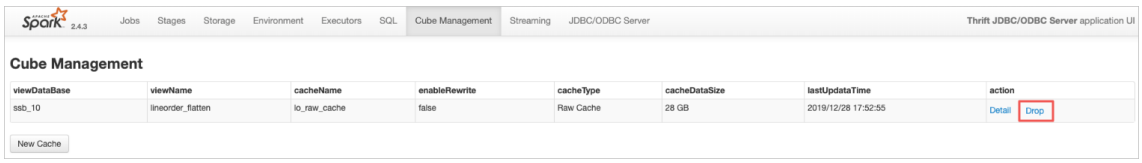
> **Note**
>
> - Periodic building tasks run on a Spark cluster. The related configurations are all saved in SparkContext, and the tasks are periodically triggered by Spark Driver. If you release the Spark cluster, periodic building is disabled.
> - The Build History section displays the following information of each complete task in the current Spark cluster: start time, end time, save mode, building conditions, and final status. After you release the Spark cluster, the information in the Build History section is cleared.

3. Manage a JindoCube cache.

   After you create and build a JindoCube cache, you can manage it on the Cube Management page.

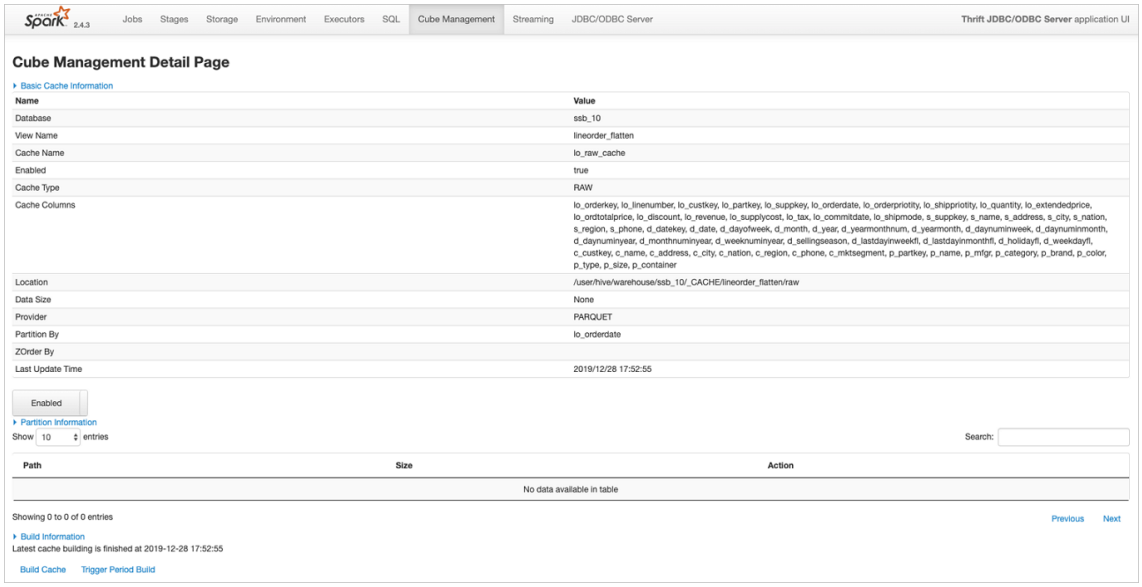
   - Delete a cache

     On the Cube Management page, find the cache you want to delete, and click **Drop** in the action column. All the metadata and data of the cache are deleted.



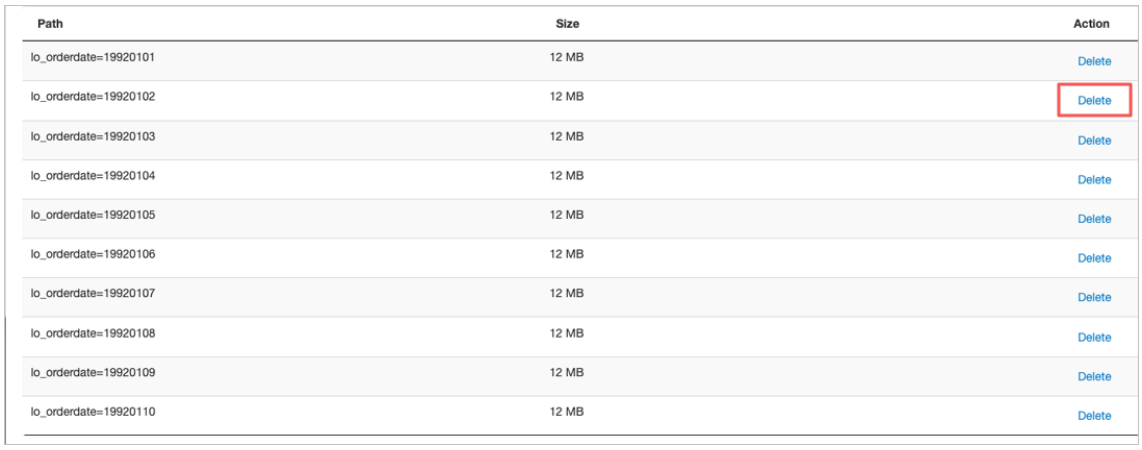
   - Enable and disable optimization of Spark queries

     JindoCube allows you to specify whether to optimize Spark queries at the cache level. On the details page of a cache, click **Enabled** or **Disabled** in the lower part of the Basic Cache Information section to enable or disable the optimization.



- Delete partition data from a cache

  If data in a cache is stored by partition, you can delete the unnecessary partition data from the cache to release more space. On the details page of the cache, find the partition whose data you want to delete, and click **Delete** in the Action column.



Path	Size	Action
lo_orderdate=19920101	12 MB	Delete
lo_orderdate=19920102	12 MB	Delete
lo_orderdate=19920103	12 MB	Delete
lo_orderdate=19920104	12 MB	Delete
lo_orderdate=19920105	12 MB	Delete
lo_orderdate=19920106	12 MB	Delete
lo_orderdate=19920107	12 MB	Delete
lo_orderdate=19920108	12 MB	Delete
lo_orderdate=19920109	12 MB	Delete
lo_orderdate=19920110	12 MB	Delete

> **Note** Before you delete the data of a partition, make sure that the data is no longer needed. If you delete data that is required in subsequent queries, the queries return invalid results.

4. Optimize Spark queries.

   JindoCube optimizes view-based queries. After you have created a raw cache or cube cache for a view, EMR Spark overwrites the execution plan of queries on this view. The new execution plan still uses the original logical semantics but can directly access the data in the cache. This accelerates Spark queries. Use the lineorder_flatten view as an example. It is a wide table view generated by associating the lineorder table with other dimension tables. The following figure shows the information about this view.

```
0: jdbc:hive2://localhost:10001/ssb_10> desc extended lineorder_flatten;
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+-----------------------------+-----------------------------------------------------+----------+
|           col_name          |                     data_type                       | comment  |
+-----------------------------+-----------------------------------------------------+----------+
| lo_orderkey                 | bigint                                              | NULL     |
| lo_linenumber               | bigint                                              | NULL     |
| lo_custkey                  | int                                                 | NULL     |
| lo_partkey                  | int                                                 | NULL     |
| lo_suppkey                  | int                                                 | NULL     |
| lo_orderdate                | int                                                 | NULL     |
| lo_orderpriotity            | string                                              | NULL     |
| lo_shippriotity             | int                                                 | NULL     |
| lo_quantity                 | bigint                                              | NULL     |
| lo_extendedprice            | bigint                                              | NULL     |
| lo_ordtotalprice            | bigint                                              | NULL     |
| lo_discount                 | bigint                                              | NULL     |
| lo_revenue                  | bigint                                              | NULL     |
| lo_supplycost               | bigint                                              | NULL     |
| lo_tax                      | bigint                                              | NULL     |
| lo_commitdate               | int                                                 | NULL     |
| lo_shipmode                 | string                                              | NULL     |
| s_suppkey                   | int                                                 | NULL     |
| s_name                      | string                                              | NULL     |
| s_address                   | string                                              | NULL     |
| s_city                      | string                                              | NULL     |
| s_nation                    | string                                              | NULL     |
| s_region                    | string                                              | NULL     |
| s_phone                     | string                                              | NULL     |
| d_datekey                   | int                                                 | NULL     |
| d_date                      | string                                              | NULL     |
| d_dayofweek                 | string                                              | NULL     |
| d_month                     | string                                              | NULL     |
| d_year                      | int                                                 | NULL     |
| d_yearmonthnum              | int                                                 | NULL     |
| d_yearmonth                 | string                                              | NULL     |
| d_daynuminweek              | int                                                 | NULL     |
| d_daynuminmonth             | int                                                 | NULL     |
| d_daynuminyear              | int                                                 | NULL     |
| d_monthnuminyear            | int                                                 | NULL     |
| d_weeknuminyear             | int                                                 | NULL     |
| d_sellingseason             | string                                              | NULL     |
| d_lastdayinweekfl           | int                                                 | NULL     |
| d_lastdayinmonthfl          | int                                                 | NULL     |
| d_holidayfl                 | int                                                 | NULL     |
| d_weekdayfl                 | int                                                 | NULL     |
| c_custkey                   | int                                                 | NULL     |
| c_name                      | string                                              | NULL     |
| c_address                   | string                                              | NULL     |
| c_city                      | string                                              | NULL     |
| c_nation                    | string                                              | NULL     |
| c_region                    | string                                              | NULL     |
| c_phone                     | string                                              | NULL     |
| c_mktsegment                | string                                              | NULL     |
| p_partkey                   | int                                                 | NULL     |
| p_name                      | string                                              | NULL     |
| p_mfgr                      | string                                              | NULL     |
| p_category                  | string                                              | NULL     |
| p_brand                     | string                                              | NULL     |
| p_color                     | string                                              | NULL     |
| p_type                      | string                                              | NULL     |
| p_size                      | int                                                 | NULL     |
| p_container                 | string                                              | NULL     |
|                             |                                                     |          |
| # Detailed Table Information |                                                     |          |
| Database                    | ssb_10                                              |          |
| Table                       | lineorder_flatten                                   |          |
| Owner                       | hadoop                                              |          |
| Created Time                | Sat Dec 28 17:30:02 CST 2019                        |          |
| Last Access                 | Thu Jan 01 08:00:00 CST 1970                        |          |
| Created By                  | Spark 2.2 or prior                                  |          |
| Type                        | VIEW                                                |          |
| View Text                   | SELECT `lineorder`.`lo_orderkey`, `lineorder`.`lo_linenumber`, `lineorder`.`lo_custkey`, `lineorder`.`lo_partkey`, `
lineorder`.`lo_suppkey`, `lineorder`.`lo_orderdate`, `lineorder`.`lo_orderpriotity`, `lineorder`.`lo_shippriotity`, `lineorder`.`lo_quantity`, `lineor
der`.`lo_extendedprice`, `lineorder`.`lo_ordtotalprice`, `lineorder`.`lo_discount`, `lineorder`.`lo_revenue`, `lineorder`.`lo_supplycost`, `lineorder`
.`lo_tax`, `lineorder`.`lo_commitdate`, `lineorder`.`lo_shipmode`, `supplier`.`s_suppkey`, `supplier`.`s_name`, `supplier`.`s_address`, `supplier`.`s_
city`, `supplier`.`s_nation`, `supplier`.`s_region`, `supplier`.`s_phone`, `dates`.`d_datekey`, `dates`.`d_date`, `dates`.`d_dayofweek`, `dates`.`d_mo
nth`, `dates`.`d_year`, `dates`.`d_yearmonthnum`, `dates`.`d_yearmonth`, `dates`.`d_daynuminweek`, `dates`.`d_daynuminmonth`, `dates`.`d_daynuminyear`
, `dates`.`d_monthnuminyear`, `dates`.`d_weeknuminyear`, `dates`.`d_sellingseason`, `dates`.`d_lastdayinweekfl`, `dates`.`d_lastdayinmonthfl`, `dates`
.`d_holidayfl`, `dates`.`d_weekdayfl`, `customer`.`c_custkey`, `customer`.`c_name`, `customer`.`c_address`, `customer`.`c_city`, `customer`.`c_nation`
, `customer`.`c_region`, `customer`.`c_phone`, `customer`.`c_mktsegment`, `part`.`p_partkey`, `part`.`p_name`, `part`.`p_mfgr`, `part`.`p_category`, `
part`.`p_brand`, `part`.`p_color`, `part`.`p_type`, `part`.`p_size`, `part`.`p_container` FROM `ssb_10`.`lineorder`, `ssb_10`.`supplier`, `ssb_10`.`da
tes`, `ssb_10`.`customer`, `ssb_10`.`part`
WHERE `lineorder`.`lo_orderdate` = `dates`.`d_datekey` AND `lineorder`.`lo_custkey` = `customer`.`c_custkey` AND `lineorder`.`lo_suppkey` = `supplier`
.`s_suppkey` AND `lineorder`.`lo_partkey` = `part`.`p_partkey` |                                                                  |
| Table Properties            | [transient_lastDdlTime=1577677599]                  |          |
| Serde Library               | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe  |          |
| InputFormat                 | org.apache.hadoop.mapred.TextInputFormat            |          |
| OutputFormat                | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat |          |
| Storage Properties          | [serialization.format=1]                            |          |
+-----------------------------+-----------------------------------------------------+----------+
73 rows selected (0.057 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

The following figure shows the execution plan of simple queries on the lineorder_flatten view.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(11) SortMergeJoin [lo_partkey#313], [p_partkey#359], Inner
   :- *(8) Sort [lo_partkey#313 ASC NULLS FIRST], false, 0
   :  +- Exchange hashpartitioning(lo_partkey#313, 200)
   :     +- *(7) SortMergeJoin [lo_custkey#312], [c_custkey#351], Inner
   :        :- *(4) Sort [lo_custkey#312 ASC NULLS FIRST], false, 0
   :        :  +- Exchange hashpartitioning(lo_custkey#312, 200)
   :        :     +- *(3) BroadcastHashJoin [lo_orderdate#315], [d_datekey#334], Inner, BuildRight
   :        :        :- *(3) BroadcastHashJoin [lo_suppkey#314], [s_suppkey#327], Inner, BuildRight
   :        :        :  :- *(3) Filter (((isnotnull(lo_suppkey#314) && isnotnull(lo_orderdate#315)) && isnotnull(lo_custkey#312)) && isnotnull(lo_part
key#313))
   :        :        :  :  +- Scan hive ssb_10.lineorder [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_ord
erdate#315, lo_orderpriotity#316, lo_shippriotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#32
2L, lo_supplycost#323L, lo_tax#324L, lo_commitdate#325, lo_shipmode#326], HiveTableRelation `ssb_10`.`lineorder`, org.apache.hadoop.hive.serde2.lazy.L
azySimpleSerDe, [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_orderdate#315, lo_orderpriotity#316, lo_ship
priotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#322L, lo_supplycost#323L, lo_tax#324L, lo_c
ommitdate#325, lo_shipmode#326]
   :        :        :  +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :        :     +- *(1) Filter isnotnull(s_suppkey#327)
   :        :        :        +- Scan hive ssb_10.supplier [s_suppkey#327, s_name#328, s_address#329, s_city#330, s_nation#331, s_region#332, s_phone#
333], HiveTableRelation `ssb_10`.`supplier`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [s_suppkey#327, s_name#328, s_address#329, s_city#330
, s_nation#331, s_region#332, s_phone#333]
   :        :        +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :           +- *(2) Filter isnotnull(d_datekey#334)
   :        :              +- Scan hive ssb_10.dates [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearm
onth#340, d_daynuminweek#341, d_daynuminmonth#342, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweek
fl#347, d_lastdayinmonthfl#348, d_holidayfl#349, d_weekdayfl#350], HiveTableRelation `ssb_10`.`dates`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSe
rDe, [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearmonth#340, d_daynuminweek#341, d_daynuminmonth#34
2, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweekfl#347, d_lastdayinmonthfl#348, d_holidayfl#349,
 d_weekdayfl#350]
   :        +- *(6) Sort [c_custkey#351 ASC NULLS FIRST], false, 0
   :           +- Exchange hashpartitioning(c_custkey#351, 200)
   :              +- *(5) Filter ((isnotnull(c_region#356) && (c_region#356 = ASIA)) && isnotnull(c_custkey#351))
   :                 +- Scan hive ssb_10.customer [c_custkey#351, c_name#352, c_address#353, c_city#354, c_nation#355, c_region#356, c_phone#357, c_mk
tsegment#358], HiveTableRelation `ssb_10`.`customer`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [c_custkey#351, c_name#352, c_address#353, c
_city#354, c_nation#355, c_region#356, c_phone#357, c_mktsegment#358]
   +- *(10) Sort [p_partkey#359 ASC NULLS FIRST], false, 0
      +- Exchange hashpartitioning(p_partkey#359, 200)
         +- *(9) Filter isnotnull(p_partkey#359)
            +- Scan hive ssb_10.part [p_partkey#359, p_name#360, p_mfgr#361, p_category#362, p_brand#363, p_color#364, p_type#365, p_size#366, p_conta
iner#367], HiveTableRelation `ssb_10`.`part`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [p_partkey#359, p_name#360, p_mfgr#361, p_category#3
62, p_brand#363, p_color#364, p_type#365, p_size#366, p_container#367] |
+----------------------------------------------------+
1 row selected (0.435 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

After a raw cache is created and built for the lineorder_flatten view, the same queries are performed on the view. EMR Spark automatically uses the cache to optimize the original execution plan. The following figure shows the new execution plan.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:17:47 INFO [main] HiveConf: Found configuration file file:/etc/ecm/spark-conf-2.4.3-hadoop2.8-1.4.2/hive-site.xml
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(1) Project [lo_orderkey#128L, lo_linenumber#129L, lo_custkey#130, lo_partkey#131, lo_suppkey#132, lo_orderdate#133, lo_orderpriotity#134, lo_ship
priotity#135, lo_quantity#136L, lo_extendedprice#137L, lo_ordtotalprice#138L, lo_discount#139L, lo_revenue#140L, lo_supplycost#141L, lo_tax#142L, lo_c
ommitdate#143, lo_shipmode#144, s_suppkey#145, s_name#146, s_address#147, s_city#148, s_nation#149, s_region#150, s_phone#151, ... 34 more fields]
   +- *(1) Filter (isnotnull(c_region#174) && (c_region#174 = ASIA))
      +- *(1) FileScan parquet [lo_orderkey#128L,lo_linenumber#129L,lo_custkey#130,lo_partkey#131,lo_suppkey#132,lo_orderpriotity#134,lo_shippriotity#
135,lo_quantity#136L,lo_extendedprice#137L,lo_ordtotalprice#138L,lo_discount#139L,lo_revenue#140L,lo_supplycost#141L,lo_tax#142L,lo_commitdate#143,lo_
shipmode#144,s_suppkey#145,s_name#146,s_address#147,s_city#148,s_nation#149,s_region#150,s_phone#151,d_datekey#152,... 34 more fields] Batched: true,
Format: Parquet, Location: InMemoryFileIndex[/user/hive/warehouse/ssb_10/_CACHE/lineorder_flatten/raw], PartitionCount: 2406, PartitionFilters: [], Pu
shedFilters: [IsNotNull(c_region), EqualTo(c_region,ASIA)], ReadSchema: struct<lo_orderkey:bigint,lo_linenumber:bigint,lo_custkey:int,lo_partkey:int,l
o_suppkey:int,lo_or... |
+----------------------------------------------------+
```

The new execution plan no longer uses the calculation logic of the lineorder_flatten view. Instead, it directly accesses the cached data in HDFS.

Usage notes

1. JindoCube does not ensure data consistency between the cache and source table. You must manually trigger data updates or configure a periodic update policy based on your requirements for data consistency.
2. JindoCube determines whether to use a cache to optimize execution plans based on the metadata of views. If cached data is incomplete when JindoCube optimizes an execution plan, the query results may be invalid or incomplete. The following operations or configurations may cause incomplete cached data: 1. Delete partition data that is required in subsequent queries on the cache details page. 2. Data required in subsequent queries is filtered out based on the filter conditions specified when you build or update the cache. 3. Required data has not been updated to the cache when the query is performed.

10.5. Toolset

10.5.1. Use JindoFS FUSE

This topic describes how to access JindoFileSystem (JindoFS) from a Filesystem in Userspace (FUSE) client. FUSE allows you to use JindoFS in block storage mode and cache mode (only JFS Scheme supported).

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

FUSE is a method provided by the Linux kernel to mount a file system. You can use a FUSE client to map files in a JindoFS cluster to a local disk. Then, you can directly access data in the JindoFS cluster without the need to use the `hadoop fs -ls jfs://<namespace>/` command.

Mount a directory

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs
   ```

3. Run the following command to mount the directory:

   ```
   jindofs-fuse /mnt/jfs
   ```

   In this example, the */mnt/jfs/* directory is mounted.

Read and write files

1. Run the following command to query all the sub-directories under the /mnt/jfs/ directory:

   ```
   ls /mnt/jfs/
   ```

   All the namespaces that you have configured on the server are returned.

   ```
   test  testcache
   ```

2. Run the following command to query all the files in the test namespace:

   ```
   ls /mnt/jfs/test/
   ```

3. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs/test/dir1
   ls /mnt/jfs/test/
   ```

4. Run the following command to write data into a file:

```
echo "hello world" > /tmp/hello.txt
cp /tmp/hello.txt /mnt/jfs/test/dir1/
```

5. Run the following command to read data from a file:

```
cat /mnt/jfs/test/dir1/hello.txt
```

The following information is returned:

```
hello world
```

The following example shows how to read data from and write data into files in Python:

1. Write data into a file in Python. Create the *write.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'w',encoding = 'utf-8') as f:
   f.write("my first file\n")
   f.write("This file\n\n")
   f.write("contains three lines\n")
```

2. Read data from a file in Python. Create the *read.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'r',encoding = 'utf-8') as f:
    lines = f.readlines()
    [print(x, end = '') for x in lines]
```

Run the following command to read the data that is written into the *test.txt* file:

```
[hadoop@emr-header-1 ~]$ ./read.py
```

The following information is returned:

```
my first file
This file
```

Unmount FUSE

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to unmount FUSE:

```
umount jindofs-fuse
```

If the `target is busy` error is returned, switch to another directory, stop all the programs that are reading data from and writing data into FUSE files, and then unmount FUSE.

10.5.2. Use tiered storage commands of JindoFS

In E-MapReduce (EMR) V3.30, JindoFS provides a tiered storage feature. This feature allows you to store cold and hot data in different storage media. This helps reduce the data storage costs or accelerate data access.

Use jindo jfs

Run the following command to obtain the help information:

```
jindo jfs -help archive
-archive -i/a <path> ... :
  Archive commands.
```

Tiered storage commands in JindoFS are asynchronous and start only related tasks.

Common commands:

- Cache
- Uncache
- Archive
- Unarchive
- Status
- ls2 command

Cache

You can use this command to back up data stored in a specific path to local disks. Then, you can read the data from local disks, without the need to read data from Object Storage Service (OSS).

```
jindo jfs -cache -p <path>
```

`-p` can be used to ensure that local data is not cleared based on disk usage.

Uncache

You can use this command to delete backup data from local disks and store data only to OSS Standard storage.

```
jindo jfs -uncache  <path>
```

Archive

You can use this command to delete backup data from local disks and store data to OSS in Infrequent Access (IA) or Archive mode. For information about the storage classes, see Overview.

```
jindo jfs -archive -i|-a <path>
```

`-i` is used to store data to OSS in IA mode. `-a` is used to store data to OSS in Archive mode.

Unarchive

You can use this command to restore data from the Archive mode to the IA or Standard mode. You can temporarily restore data stored in Archive mode to allow the data to be readable within one day.

```
jindo jfs -unarchive -i/-o <path>
```

`-i` is used to store data to OSS in IA mode. `-o` is used to temporarily restore data stored in Archive mode to allow the data to be readable.

Status

You can use this command to view task details. By default, the number of files to be subject to tiered storage in a specific directory and the data that has been subject to tiered storage are measured.

```
jindo jfs -status -detail/-sync <path>
```

`-detail` is used to view the storage progress of file data. `-sync` indicates that the command exits after a tiered storage task is completed.

ls2 command

JindoFS provides the ls2 command that allows you to view the file storage status on the basis of Hadoop ls commands:

```
hadoop fs -ls2 <path>
```

Example of command output, which includes the file storage class:

```
drwxrwxrwx  - -         0    2020-06-05 04:27 oss://xxxx/warehouse
-rw-rw-rw-  1 Archive   1484 2020-09-23 16:40 oss://xxxx/wikipedia_data.csv
-rw-rw-rw-  1 Standard  1676 2020-06-07 20:04 oss://xxxx/wikipedia_data.json
```

10.5.3. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain help information:

   ```
   jindo distcp --help
   ```

   The following information is returned:

```
     --help           - Print help text
     --src=VALUE          - Directory to copy files from
     --dest=VALUE              - Directory to copy files to
     --parallelism=VALUE         - Copy task parallelism
     --outputManifest=VALUE       - The name of the manifest file
     --previousManifest=VALUE   -   The path to an existing manifest file
     --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
if specified
     --copyFromManifest   -   Copy from a manifest instead of listing a directory
     --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
     --srcPattern=VALUE   -   Include only source files matching this pattern
     --deleteOnSuccess   -   Delete input files after a successful copy
     --outputCodec=VALUE   -   Compression codec for output files
     --groupBy=VALUE   -   Pattern to group input files by
     --targetSize=VALUE   -   Target size for output files
     --enableBalancePlan   -   Enable plan copy task to make balance
     --enableDynamicPlan   -   Enable plan copy task dynamically
     --enableTransaction   -   Enable transation on Job explicitly
     --diff   -   show the difference between src and dest filelist
     --ossKey=VALUE   -   Specify your oss key if needed
     --ossSecret=VALUE   -   Specify your oss secret if needed
     --ossEndPoint=VALUE   -   Specify your oss endPoint if needed
     --policy=VALUE   -   Specify your oss storage policy
     --cleanUpPending   -   clean up the incomplete upload when distcp job finish
     --queue=VALUE   -   Specify yarn queuename if needed
     --bandwidth=VALUE   -   Specify bandwidth per map/reduce in MB if needed
     --s3Key=VALUE   -   Specify your s3 key
     --s3Secret=VALUE   -   Specify your s3 Sercet
     --s3EndPoint=VALUE   -   Specify your s3 EndPoint
```

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-
01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03
/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03
/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017
-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-0
1/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-0
1/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_tab
le/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/
03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/
03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromMani
fest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

> **Note** If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

--queue

--queue specifies the name of the YARN queue where the current DistCp task resides.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<your_bucket>/hourly_table --qu
eue yarnqueue
```

--bandwidth

--bandwidth specifies the bandwidth allocated to the current DistCp task. This prevents the task from using excessive bandwidth. Unit: MB/s.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.c
om --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy archive --parallelism 20
  ```

- Example of the IA storage class (`--ia`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy ia --parallelism 20
  ```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

Use Amazon S3 as a data source

You can use the --s3Key, --s3Secret, and --s3EndPoint parameters in a command to specify information related to Amazon S3.

Example:

```
jindo distcp jindo-distcp-2.7.3.jar --src s3a://yourbucket/ --dest oss://<your_bucket>/hour
ly_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com
```

You can configure the s3Key, s3Secret, and s3EndPoint parameters in the *core-site.xml* file of Hadoop. This way, you do not need to specify an AccessKey pair each time.

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

Example:

```
jindo distcp /tmp/jindo-distcp-2.7.3.jar --src s3://smartdata1/ --dest s3://smartdata1/tmp
--s3EndPoint  s3-us-west-1.amazonaws.com
```

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

> **Note** If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

10.5.4. Use Jindo DistCp in specific scenarios

This topic describes how to use Jindo DistCp in specific scenarios.

Prerequisites

- An E-MapReduce (EMR) cluster of a required version is created. For more information, see Create a cluster.
- Java Development Kit (JDK) 1.8 is installed.
- *jindo-distcp-<version>.jar* is downloaded based on your Hadoop version.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 2.7 and later 2.X versions.
  - jindo-distcp-3.0.0.jar is downloaded for Hadoop 3.X.

Use scenarios

Common use scenarios of Jindo DistCp:

- Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?
- Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?
- Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?
- Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?
- Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?
- Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file size?
- Scenario 8: What parameters are required if Amazon S3 is used as a data source?
- Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?
- Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?
- Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?
- Scenario 12: What parameters are required if I want to delete original files after a copy operation?
- Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Scenario 1: What parameters are required for optimization if I import a large amount of HDFS data or a large number of files (millions or tens of millions) to OSS?

If you are not using EMR, the following conditions must be met before you can import HDFS data to Object Storage Service (OSS):

- You can read data from HDFS.
- The AccessKey ID, AccessKey secret, and endpoint of OSS are obtained. You can write data to a destination OSS bucket.
- The storage class of the OSS bucket is not Archive.
- You can submit MapReduce tasks.
- The JAR package of Jindo DistCp is downloaded.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 10
```

> **Note** For more information about the parameters, see Use Jindo DistCp.

To import a large amount of data or a large number of files, such as millions or tens of millions of files, to OSS, you can set parallelism to a large value to increase the concurrency. You can also set `enableBatch` to true for optimization. Optimization command:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --parallelism 500 --enableBatch
```

Scenario 2: How do I verify data integrity after I use Jindo DistCp to import data to OSS?

You can use one of the following methods to verify data integrity:

- Jindo DistCp Counters

  Check DistCp Counters in the counter information of a MapReduce task.

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

where:

- Bytes Destination Copied specifies the size of the file that you copy to a destination directory, in bytes.
- Bytes Source Read specifies the size of the file that you read from the source directory, in bytes.
- Files Copied specifies the number of copied files.

- Jindo DistCp --diff

  Use `--diff` to compare the information of files in the source and destination directories. The information includes file name and file size. Missing files and the files that fail to be copied are recorded in a manifest file. The manifest file is generated in the directory where you perform the comparison.

  Add `--diff` to the command described in Scenario 1. Example:

  ```
  hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
  -hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
  aliyuncs.com --diff
  ```

  If all files are copied, the following information is returned:

  ```
  INFO distcp.JindoDistCp: distcp has been done completely
  ```

Scenario 3: What parameters are required to support resumable upload in the case of a DistCp task failure when I import HDFS data to OSS?

If your DistCp task fails, and you want to perform resumable upload to copy only the files that fail to be copied, perform the following operations on the basis of the command described in Scenario 1:

1. Add `--diff` to check whether all files are copied.

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
   hou.aliyuncs.com --diff
   ```

   If all the files are copied, the following information is returned. Otherwise, a manifest file is generated. In this case, go to the next step.

   ```
   INFO distcp.JindoDistCp: distcp has been done completely.
   ```

2. Use `--copyFromManifest` and `--previousManifest` to copy the remaining files. Example:

   ```
   hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
   ng-hhht/hourly_table --dest oss://yang-hhht/hourly_table --previousManifest=file:///opt
   /manifest-2020-04-17.gz --copyFromManifest --parallelism 20
   ```

   `file:///opt/manifest-2020-04-17.gz` is the path where the manifest file is stored.

Scenario 4: What parameters are required to process the files that may be newly generated when I use Jindo DistCp to import HDFS data to OSS?

1. If no information is generated for the files that were copied last time, specify parameters to

generate a manifest file to record the information about the files that are copied.

Add `--outputManifest=manifest-2020-04-17.gz` and `--requirePreviousManifest=false` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

where:

- `--outputManifest` generates a manifest file whose name can be customized and must have a file name extension of gz, such as *manifest-2020-04-17.gz*. This file is stored in the directory specified by `--dest`.
- `--requirePreviousManifest` specifies whether to require a previous manifest file.

2. After a DistCp task is complete, copy the new files that may be generated in the source directory.

   Add `--outputManifest=manifest-2020-04-17.gz` and `--previousManifest=oss://yang-hhht/hourly_table/manifest-2020-04-17.gz` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://ya
ng-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangz
hou.aliyuncs.com --outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=fals
e --parallelism 20
```

```
hadoop jar jindo-distcp-2.7.3.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --outputManifest=manifest-2020-04-18.gz --previousManifest=oss://yang-hhht
/hourly_table/manifest-2020-04-17.gz --parallelism 10
```

3. Repeat Step 2 to continuously copy incremental files.

Scenario 5: What parameters are required to specify the YARN queue where a Jindo DistCp task resides and the available bandwidth allocated to the task?

Add the following two parameters to the command described in Scenario 1. These two parameters can be used together or separately.

- `--queue`: the name of the YARN queue.
- `--bandwidth`: the size of the specified bandwidth , in MB.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --queue yarnqueue --bandwidth 6 --parallelism 10
```

Scenario 6: What parameters are required when I write data to OSS IA or Archive storage?

- To write data to OSS Archive storage, add `--archive` to the command described in Scenario 1.

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --archive --parallelism 20
```

- To write data to OSS Infrequent Access (IA) storage, add `--ia` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --ia --parallelism 20
```

Scenario 7: What parameters are required to accelerate file transfer based on the proportion of small files and specific file sizes?

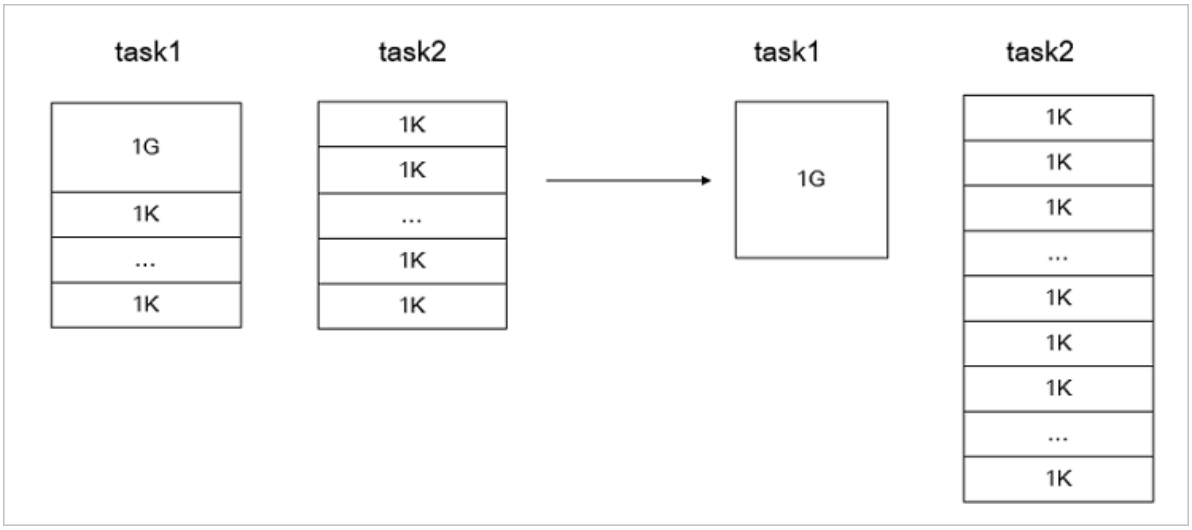
- A large number of small files and large data volume for a single large file

  If most of the files that you want to copy are small files, and the data volume of a single large file is large, the general solution is to randomly allocate the files to be copied. In this case, if you do not optimize the job allocation plan, few large files and many small files may be allocated to the same copy process. This cannot achieve excellent copy performance.

  Add `--enableDynamicPlan` to the command described in Scenario 1 to enable the optimization. --enableDynamicPlan cannot be used with `--enableBalancePlan`. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableDynamicPlan --parallelism 10
```

  The following figure shows the data copy before and after the plan is optimized.

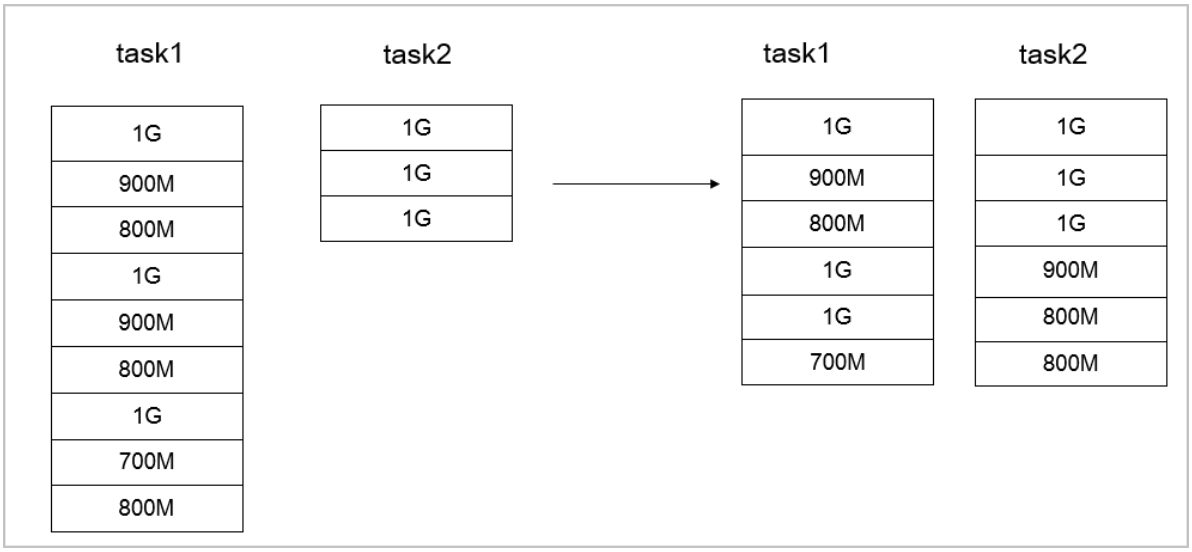


- No significant differences in sizes of files

  If the sizes are not significantly different among the files that you want to copy, use `--enableBalancePlan` to optimize the job allocation plan. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --enableBalancePlan --parallelism 10
```

The following figure shows the data copy before and after the plan is optimized.



Scenario 8: What parameters are required if Amazon S3 is used as a data source?

Replace the parameters that specify the AccessKey ID, AccessKey secret, and endpoint of OSS in the command described in Scenario 1 with the following parameters of Amazon S3:

- `--s3Key` : the AccessKey ID for Amazon S3
- `--s3Secret` : the AccessKey secret for Amazon S3
- `--s3EndPoint` : the endpoint for Amazon S3

Example:

```
hadoop jar jindo-distcp-<version>.jar --src s3a://yourbucket/ --dest oss://yang-hhht/hourly
_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com --para
llelism 10
```

Scenario 9: What parameters are required if I want to copy files to OSS and compress the copied files in the LZO or GZ format?

You can use `--outputCodec` to compress copied files in a format such as LZO or GZ to reduce the space that is used to store the files.

Add `--outputCodec` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --outputCodec=gz --parallelism 10
```

You can set --outputCodec to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files are compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change to the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

Scenario 10: What parameters are required if I want to copy the files that meet specific rules or the files in some sub-directories of the same parent directory?

- If you want to copy the files that meet specific rules, add `--srcPattern` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPattern .*\.log --parallelism 10
```

  `--srcPattern`: specifies a regular expression that filters files for the copy operation.

- If you want to copy the files stored in some sub-directories of the same parent directory, add `--srcPrefixesFile` to the command described in Scenario 1.

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang
-hhht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.
aliyuncs.com --srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

  `--srcPrefixesFile`: enables Jindo DistCp to copy files in multiple folders under the same parent directory at a time.

  Content of the *folders.txt* file:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

Scenario 11: What parameters are required if I want to merge the files that meet specific rules to reduce the number of files?

Add the following parameters to the command described in Scenario 1:

- `--targetSize`: the maximum size of the merged file, in MB
- `--groupBy`: the merging rule, which is a regular expression

Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Scenario 12: What parameters are required if I want to delete original files after a copy operation?

Add `--deleteOnSuccess` to the command described in Scenario 1. Example:

```
hadoop jar jindo-distcp-<version>.jar --src /data/incoming/hourly_table --dest oss://yang-h
hht/hourly_table --ossKey yourkey --ossSecret yoursecret --ossEndPoint oss-cn-hangzhou.aliy
uncs.com --deleteOnSuccess --parallelism 10
```

Scenario 13: What do I do if I do not want to specify AccessKey pair and endpoint information of OSS in a CLI?

Jindo DistCp allows you to save the AccessKey ID, AccessKey secret, and endpoint of OSS in the *core-site.xml* file so that you do not need to repeatedly specify the information.

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of OSS, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.jfs.cache.oss-accessKeyId</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-accessKeySecret</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.jfs.cache.oss-endpoint</name>
        <value>oss-cn-xxx.aliyuncs.com</value>
    </property>
</configuration>
```

- If you want to save the AccessKey ID, AccessKey secret, and endpoint of Amazon S3, save the following information into the *core-site.xml* file:

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

11.SmartData 2.7.301

11.1. JindoFS in block storage mode

11.1.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read and write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

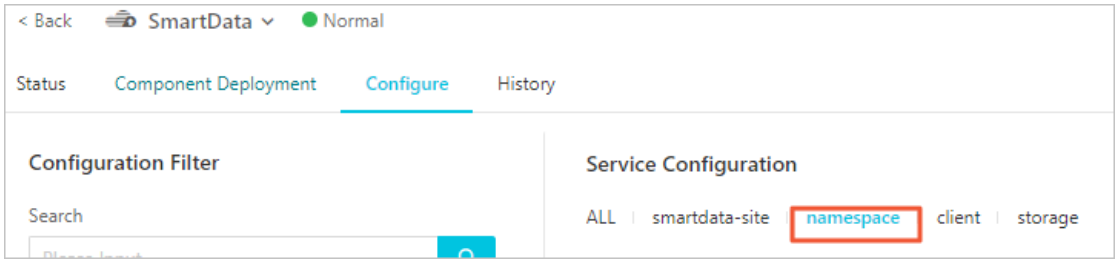
Context

The block storage mode of JindoFS has the following features:

- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.
- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

   ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

   iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.

5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

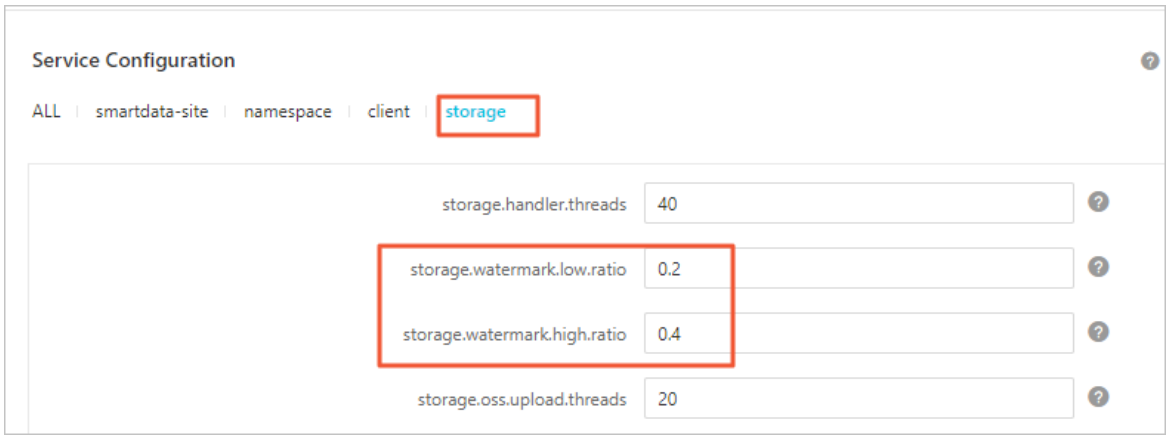
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.

i. Choose **Actions > Restart Jindo Storage Service** in the upper-right corner.
ii. In the **Cluster Activities** dialog box, specify the related parameters.
iii. Click **OK**.
iv. In the **Confirm** message, click **OK**.

11.1.2. Use Tablestore instances to store metadata

Namespace Service of JindoFS supports different metadata storage methods. This topic describes how to configure a Tablestore instance as the metadata storage backend.

Prerequisites

- An E-MapReduce (EMR) cluster is created.

  For more information, see Create a cluster.
- A Tablestore instance is created. We recommend that you use a high-performance instance.

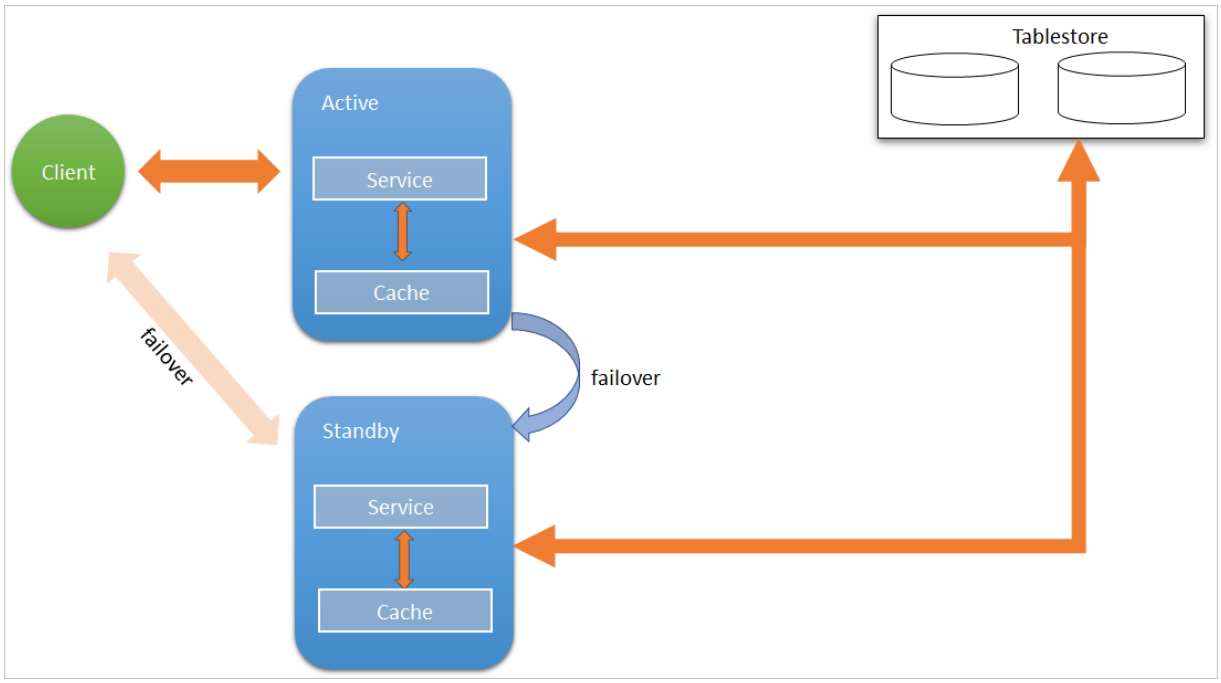
  For more information, see Create instances.

> **Note** You must enable the transaction feature.

Context

In SmartData 2.6.X and later, Namespace Service allows you to use Tablestore instances to store metadata. You can bind a Tablestore instance to each EMR JindoFS cluster. Namespace Service creates a Tablestore table for each namespace to manage and store metadata.

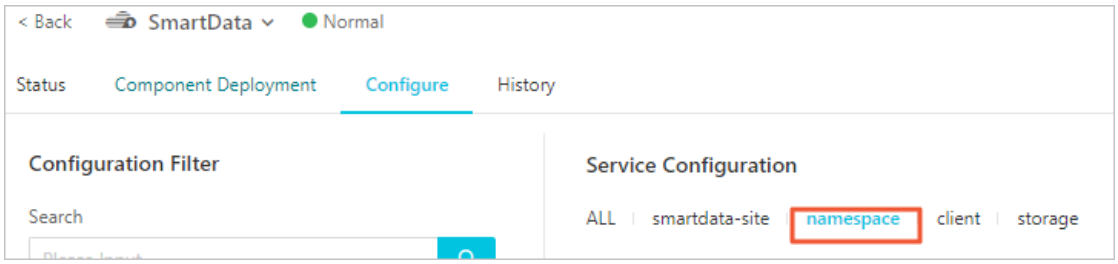
The following figure shows the structure of Namespace Service in high availability (HA) mode with a Tablestore instance as the metadata storage backend.



Configure a Tablestore instance

To use a Tablestore instance as the metadata storage backend, you must first bind the Tablestore instance to Namespace Service. Perform the following steps:

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the parameters described in the following table.

   For example, you have created a Tablestore instance named emr-jfs in the China (Hangzhou) region. Your EMR cluster is deployed in a VPC network. You have obtained the AccessKey ID and AccessKey secret that are used to access the Tablestore instance.

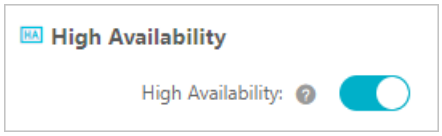
Parameter	Description	Required	Example
namespace.backend.type	The backend storage type of Namespace Service. ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to ots.	Yes	ots
namespace.ots.instance	The name of the Tablestore instance.	Yes	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	No	kkkkkk

Parameter	Description	Required	Example
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	No	XXXXXX
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	Yes	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

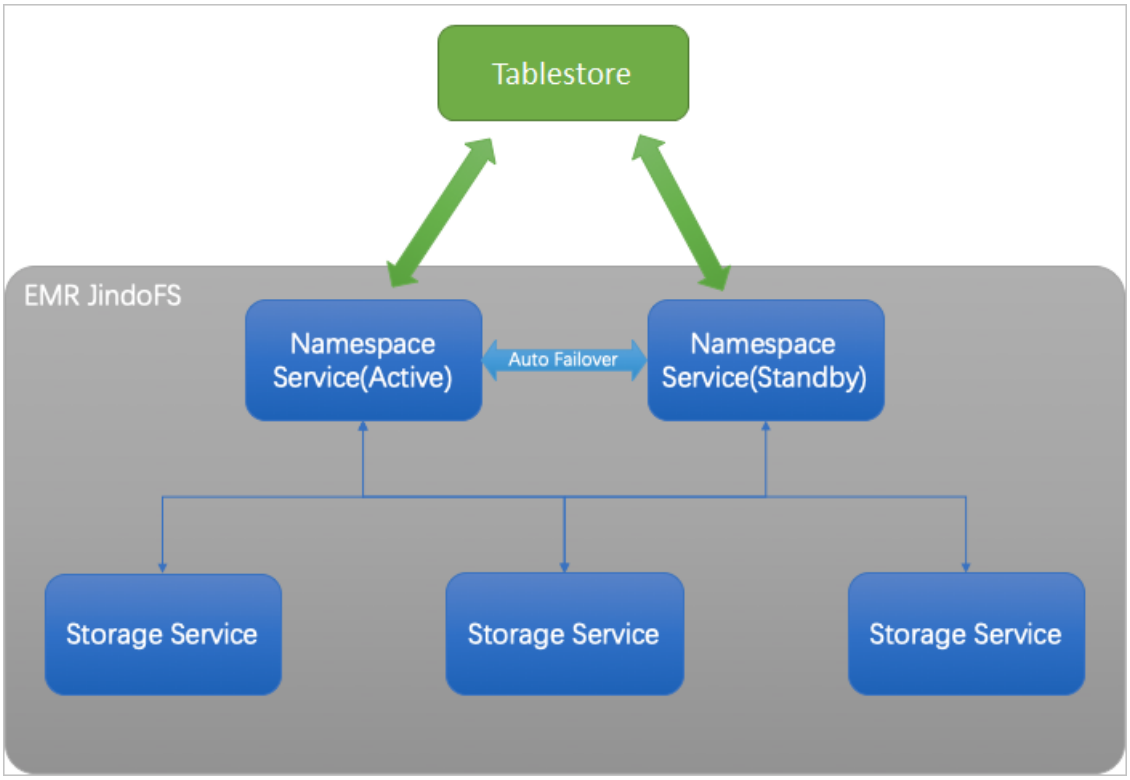
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

Configure a Tablestore instance (HA mode)

If your EMR cluster is deployed in HA mode, we recommend that you deploy Namespace Service in HA mode.



In HA mode, Namespace Service supports automatic failover. If the active namespace fails, the client automatically switches services to the standby namespace.



1. Go to the **namespace** tab for the SmartData service and perform the following operations.
   i. Change the value of the **jfs.namespace.server.rpc-address** parameter to **emr-header-1:8101,emr-header-2:8101**.
   ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the Add Configuration Item dialog box, add **namespace.backend.ots.ha** as a key and set this parameter to **true**.
   iii. Click **OK**.
   iv. Save the configurations.
      a. In the upper-right corner of the Service Configuration section, click **Save**.
      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
      c. Click **OK**.
2. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
3. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.

11.1.3. Use RocksDB to store metadata

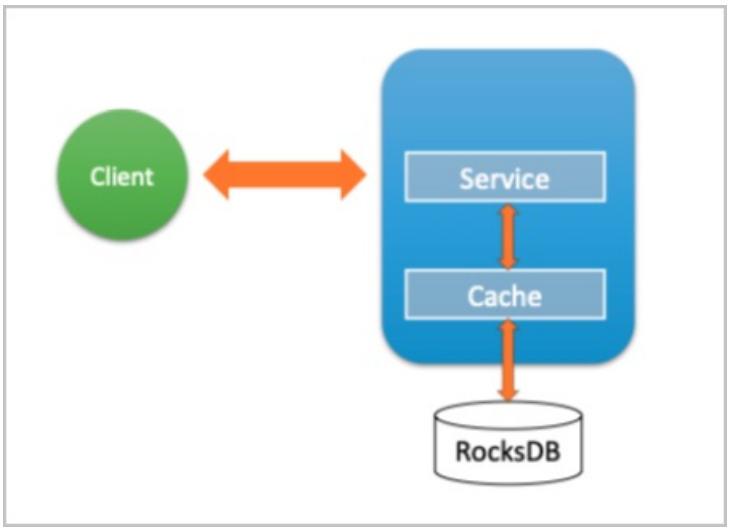
Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage

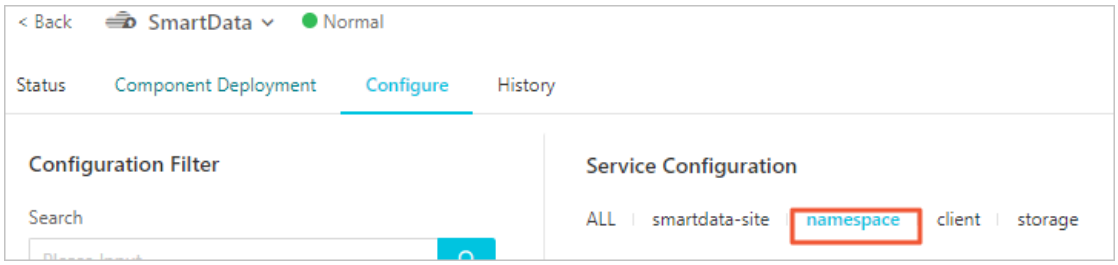
backend in high availability mode, we recommend that you use Tablestore or Raft instances. For more information, see Use Tablestore instances to store metadata and Use Raft-RocksDB-Tablestore to store metadata.

The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

11.1.4. Use Raft-RocksDB-Tablestore to store metadata

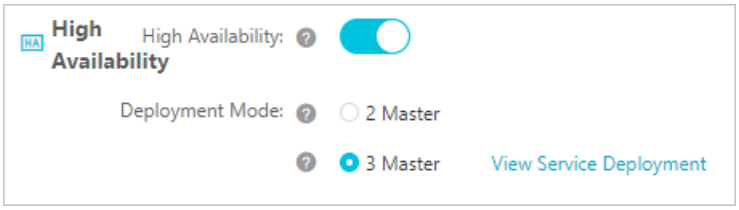
In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata.

Prerequisites

- A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  Note You must enable the transaction feature.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.

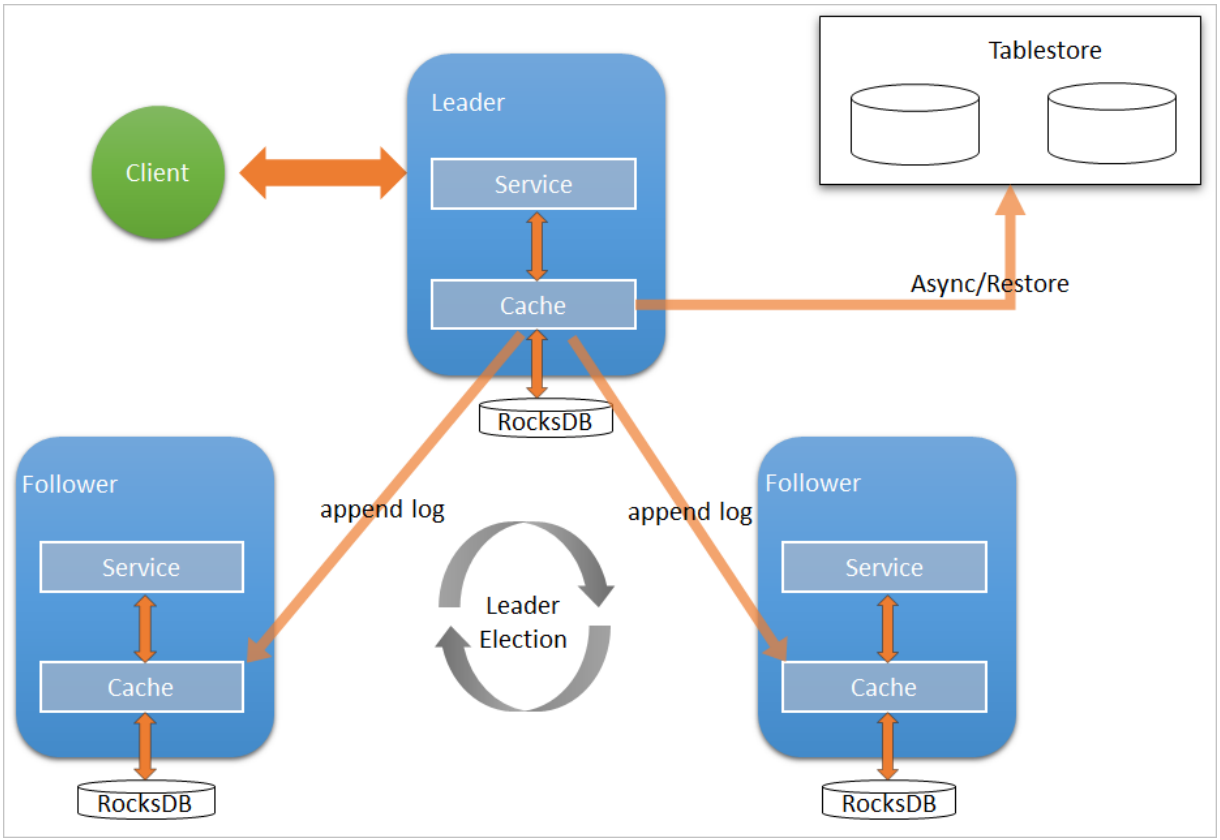


  Note If no deployment modes are available, submit a ticket.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.



Configure a Raft instance as the local storage backend

1. Suspend all the SmartData components of an EMR cluster.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   vi. Select **Stop All Components** from the **Actions** drop-down list in the upper-right corner.
2. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
3. Go to the **namespace** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **namespace** tab.
4. Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. Valid values: ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

Note If you do not need to use Tablestore for remote storage, skip Step 5.

5. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

```
hadoop fs -count jfs://test/
      1596      1482809                 25 jfs://test/
  (Number of folders) (Number of files)
```

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```

```
[RaftPeerImpl]
peer id: [redacted]
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 2335ms)
snapshot_timer: timeout(3600000ms) SCHEDULING(in 150305ms)
storage: [1, 624625]
disk_index: 624625
known_applied_index: 624625
last_log_id: (index=624625,term=2)
first_index_pinned: 624625
state_machine: Idle
last_committed_index: 624625
last_snapshot_index: 0
last_snapshot_term: 0
snapshot_status: IDLE
replicator_25769803789@[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=624261 ic=0
replicator_329853488332[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=623564 ic=0

OtsUploader: _lastStopIndex=624624, _synced=1
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.

```
[RaftPeerImpl]
peer_id: [redacted]:8103:0
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 3382ms)
snapshot_timer: timeout(600000ms) SCHEDULING(in 474855ms)
storage: [1, 153]
disk_index: 153
known_applied_index: 153
last_log_id: (index=153,term=2)
first_index_pinned: 1
state_machine: Idle
last_committed_index: 153
last_snapshot_index: 1
last_snapshot_term: 2
snapshot_status: IDLE
replicator_1116691496965@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0
replicator_3311419785217@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0

[Recovery From OTS Status]
state: FINISH
total rows: 1484409
table `jfs_block_test` 2 rows.
table `jfs_namespace_cache_ns` 1 rows.
table `jfs_namespace_test` 1484406 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   Configure the parameters described in the following table on the **namespace** tab.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

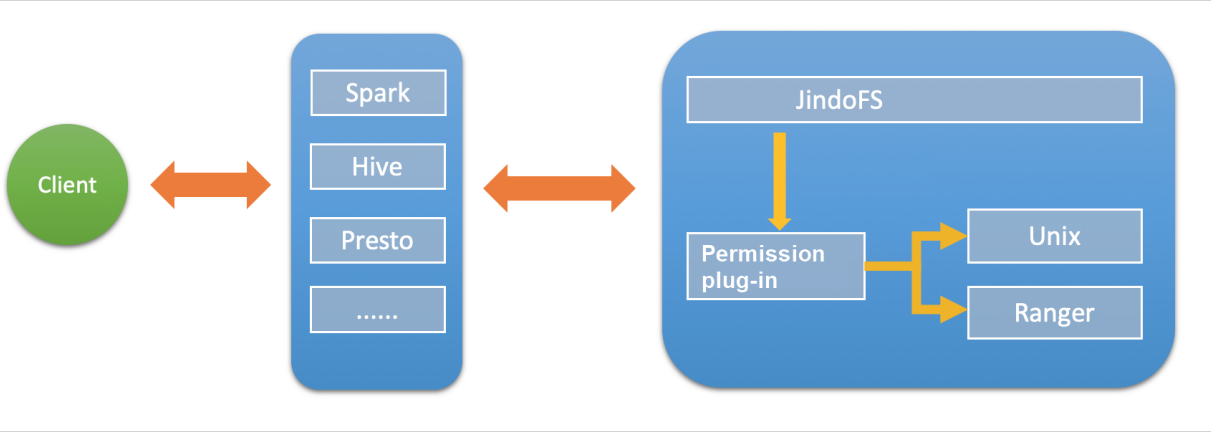
9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

11.1.5. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode. You can run UNIX commands or use Ranger to manage permissions.

Context

To use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of E-MapReduce (EMR) and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

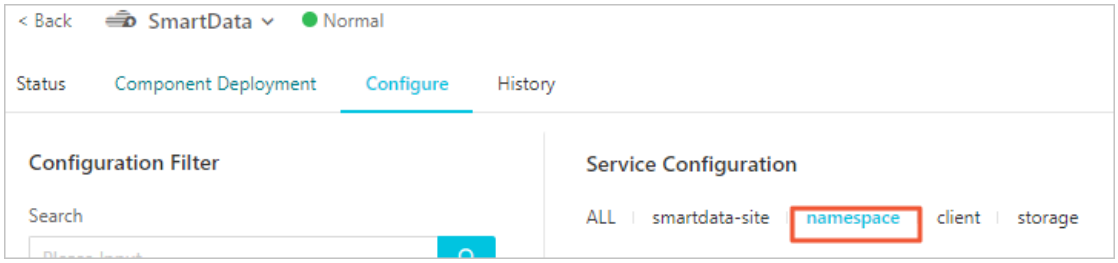


In block storage mode, you can run UNIX commands or use Ranger to manage permissions:

- UNIX allows you to grant the rwxrwxrwx permission on files.
- Ranger allows you to use wildcards in paths when you configure a permission.

Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Restart Namespace Service.

   i. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   ii. In the Cluster Activities dialog box, specify Description and click **OK**.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the following error is returned:

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

1. Configure Ranger as a permission management method in JindoFS.

   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger and click OK.

   iii. Save the configurations.

      a. In the upper-right corner of the Service Configuration section, click **Save**.

      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

      c. Click **OK**.

   iv. Restart Namespace Service.

      a. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

      b. In the Cluster Activities dialog box, specify Description and click **OK**.

2. Add the HDFS service on the web UI of Ranger and configure the required parameters.

   i. Log on to the Ranger web UI.

   For more information, see Overview.

   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}*.
Authorization Enabled	Retain default value No.
Authentication Type	Retain default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you also need to enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information of user groups synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters listed in the following table and click **OK**.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping

Parameter	Example
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

Note Configure the parameters based on the configurations in open source HDFS.

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Select **Restart All Components** from the **Actions** drop-down list in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

11.1.6. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode. It records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.29.X version is created. For more information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For more information, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

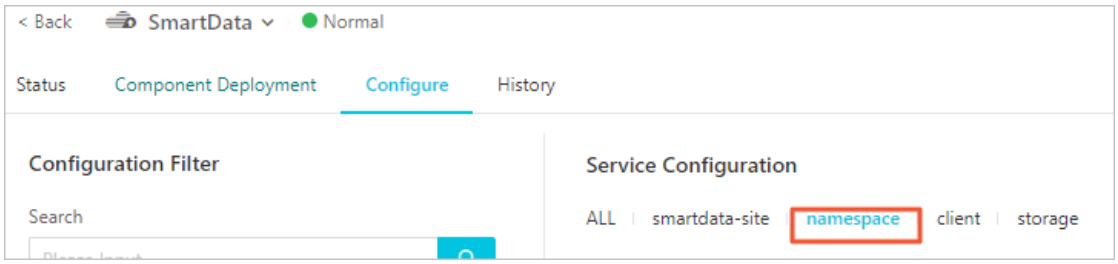
The following table describes the parameters of an audit log recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.

ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. In the upper-right corner of the **Service Configuration** section, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, configure the parameters described in the following table.

Parameter	Description	Required
namespace.auditlog.enable	■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.auditlog.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.auditlog.oss.accessKey	The AccessKey ID used to access the OSS bucket.	No
namespace.auditlog.oss.accessSecret	The AccessKey secret used to access the OSS bucket.	No
namespace.auditlog.oss.endpoint	The endpoint of the OSS bucket.	No

   iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.
   iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   v. In the **Confirm** message, click **OK**.
4. Restart Namespace Service.
   i. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.
   ii. In the left-side navigation pane, click Buckets. On the page that appears, click the name of the created bucket.
   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.
   iv. Click **Create Rule**. In the **Create Rule** dialog box, configure the parameters.

   For more information, see Configure lifecycle rules.

   v. Click **OK**.

Analyze log files

JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog. You can run a MapReduce job in the EMR console to analyze the most active commands or IP addresses in the log files. The analysis command is `jindo auditlog`.

The following table describes the parameters in the command.

Parameter	Description	Required
--src	The OSS bucket where the log files generated by AuditLog are stored. By default, this parameter is set to the value of the namespace.auditlog.oss.uri parameter specified in Step 3. You can customize a value.	No
--ns	The namespace you want to analyze. By default, all namespaces in block storage mode are analyzed.	No
--type	The analysis objects: • ip: the most active IP addresses • cmd: the most active commands	Yes
--min	The time range in minutes.	No
--day	The time range in days. day 1 indicates the current day.	**Note** You can specify only one of --min and --day.

In the EMR console, create a MapReduce job and run the jindo audit command. Example:

```
jindo auditlog --src oss://<yourbucket>/auditlog/ --ns test --type ip --day 1 --top 2
```

The following information is returned:

```
16 openFileStatusRequest
6  deleteFileletRequest
```

11.2. JindoFS in cache mode

11.2.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in OSS based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

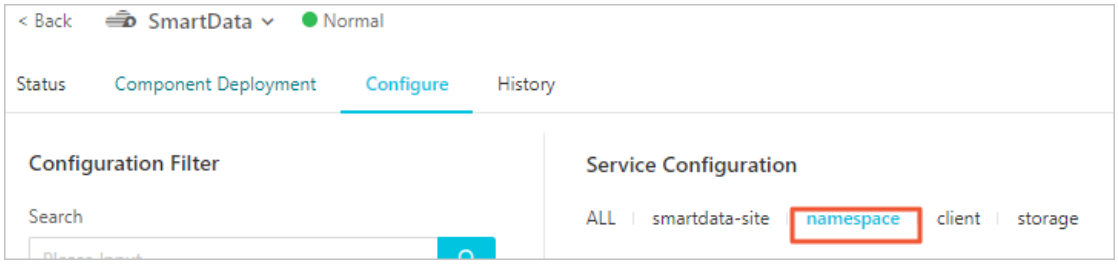
OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.

i. Click the **Configure** tab.

ii. Click the **namespace** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

   If you configure multiple namespaces, separate them with commas (,).

   ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. In the upper-right corner of the Service Configuration section, click **Save**.

5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, click **Cluster Service** and then **SmartData**. On the SMARTDATA page, click the **Configure** tab. In the Service Configuration section, click the **client** tab.

2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   The configuration immediately takes effect on the client, without the need to restart the SmartData service.

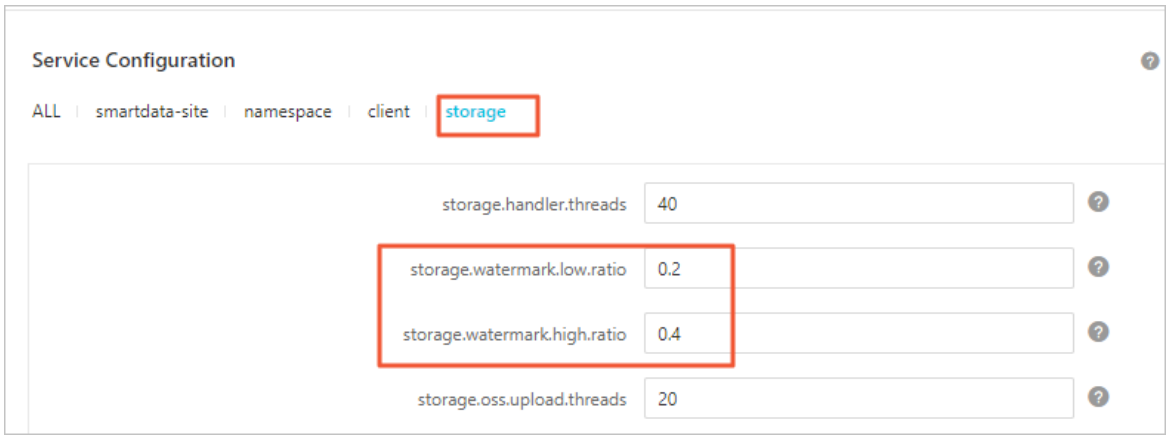
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you configured. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right

corner.

ii. In the **Cluster Activities** dialog box, specify the related parameters.

iii. Click **OK**.

iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to configure an AccessKey pair. In other cases, you must configure an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you use to access files in OSS:

- OSS Scheme

  i. In the left-side navigation pane, click **Cluster Service** and then **SmartData**. On the SMARTDATA page, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.

  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss-accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss-accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss-endpoint	The endpoint of the OSS bucket that serves as the storage backend.

- JFS Scheme

  i. In the left-side navigation pane, choose **Cluster Service > SmartData**. On the SMARTDATA page, click the **Configure** tab. In the Service Configuration section, click the **namespace** tab.

  ii. Set **jfs.namespaces** to **test**.

  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize cache performance. After you configure the parameters, you do not need to restart the SmartData service. The configurations take effect immediately on the client.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. This parameter prevents upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

Parameter	Description
fs.jfs.cache.copy.simple.max.byte	The threshold for the size of a file that is renamed over a common copy interface. If the size of a file is smaller than this threshold, a common copy interface is used. If the size is larger than this threshold, the Multipart Copy interface is used to improve copy efficiency. **Note** If you have enabled the fast copy feature of OSS, set this parameter to **-1**. This value indicates that all files are renamed over a common copy interface. This way, you can obtain the optimal rename performance.
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If too much memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.

Parameter	Description
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. This Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

11.2.2. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

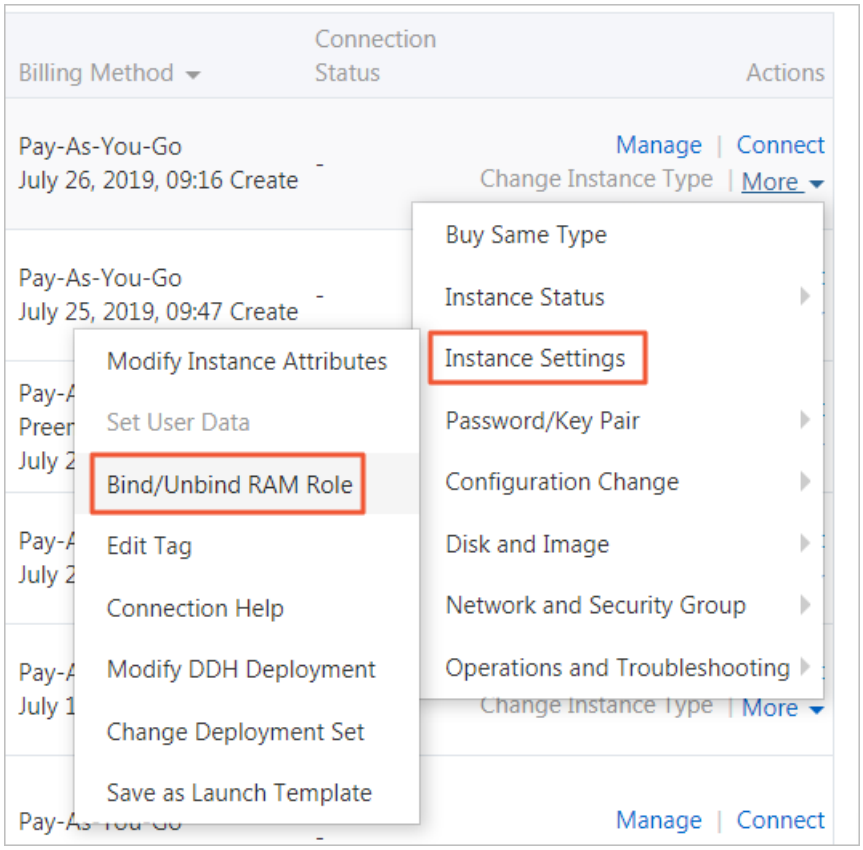
Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.

4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

```
hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
```

2. Access OSS by using the FileSystem interface of Hadoop.

   Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

11.2.3. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

Notice

- The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
- After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
- Hive and Presto jobs do not use Hadoop Job Committer.
- In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

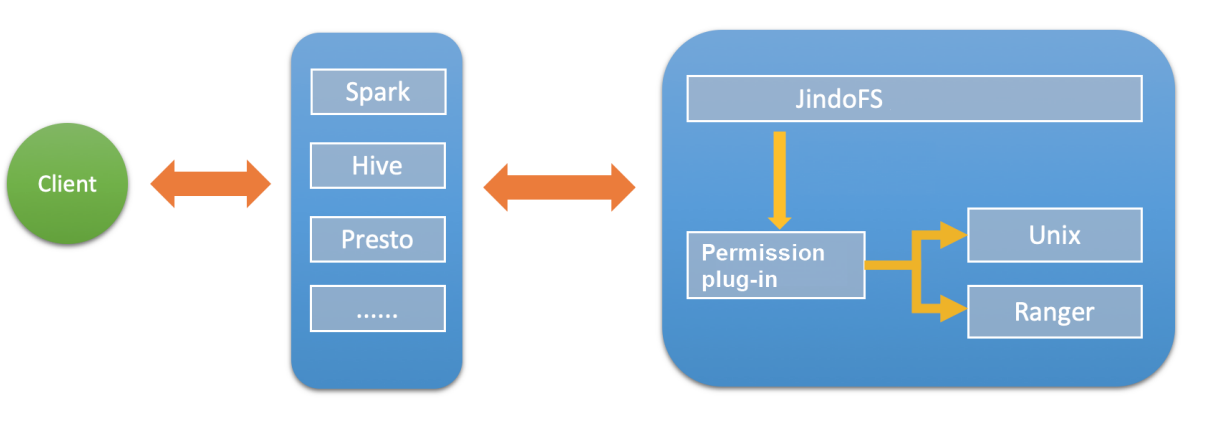
   The default value is 8.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

11.2.4. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode. You can run UNIX commands or use Ranger to manage permissions.

Context

To use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of E-MapReduce (EMR) and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

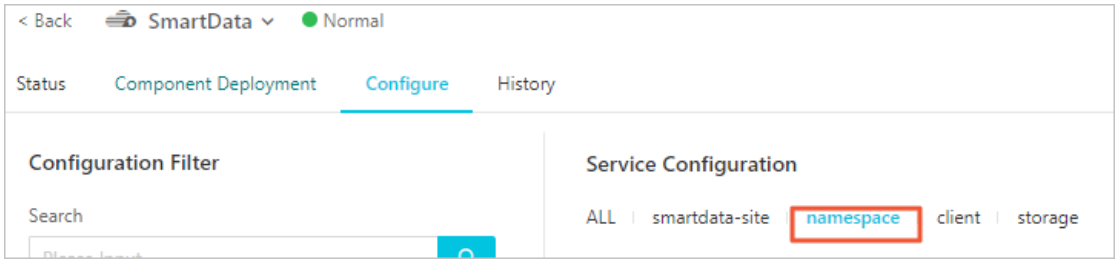


In block storage mode, you can run UNIX commands or use Ranger to manage permissions:

- UNIX allows you to grant the rwxrwxrwx permission on files.
- Ranger allows you to use wildcards in paths when you configure a permission.

Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.
   ii. Click the **namespace** tab in the Service Configuration section.



3. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Restart Namespace Service.

   i. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   ii. In the Cluster Activities dialog box, specify Description and click **OK**.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the following error is returned:

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

1. Configure Ranger as a permission management method in JindoFS.

   i. On the **namespace** tab for the SmartData service, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger and click OK.

   iii. Save the configurations.

      a. In the upper-right corner of the Service Configuration section, click **Save**.

      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

      c. Click **OK**.

   iv. Restart Namespace Service.

      a. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

      b. In the Cluster Activities dialog box, specify Description and click **OK**.

2. Add the HDFS service on the web UI of Ranger and configure the required parameters.

   i. Log on to the Ranger web UI.

   For more information, see Overview.

   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}*.
Authorization Enabled	Retain default value No.
Authentication Type	Retain default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you also need to enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information of user groups synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **namespace** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters listed in the following table and click **OK**.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping

Parameter	Example
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

Note Configure the parameters based on the configurations in open source HDFS.

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Select **Restart All Components** from the **Actions** drop-down list in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

11.2.5. Use Jindo AuditLog

Jindo AuditLog allows you to audit operations in the namespaces that are in block storage mode. It records addition, deletion, and renaming operations in the namespaces.

Prerequisites

- An EMR cluster of the 3.29.X version is created. For more information about how to create a cluster, see Create a cluster.
- An OSS bucket is created. For more information, see Create buckets.

Context

You can use AuditLog to analyze namespace access information, detect abnormal requests, and track errors. AuditLog stores log files in OSS. The size of a single log file cannot exceed 5 GB. You can use the lifecycle management feature of OSS to customize a retention period in days for the log files. JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog.

Audit log

Example:

```
2020-07-09 18:29:24.689 allowed=true ugi=hadoop (auth:SIMPLE) ip=127.0.0.1 ns=test-block cm
d=CreateFileletRequest src=jfs://test-block/test/test.snappy.parquet dst=null perm=::rwxrwx
r-x
```

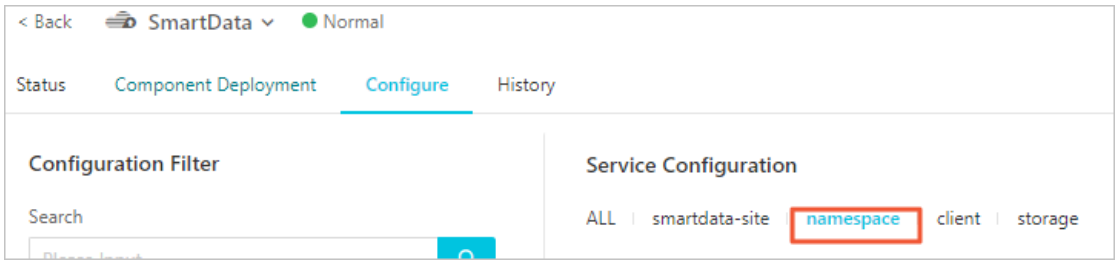
The following table describes the parameters of an audit log recorded by AuditLog for a namespace in block storage mode.

Parameter	Description
Time	The time format is yyyy-MM-dd hh:mm:ss.SSS.
allowed	Indicates whether the operation is allowed. Valid values: • true • false
ugi	The user who performed the operation. The information about the authentication method is also displayed.
ip	The client IP address.
ns	The name of the namespace in block storage mode.
cmd	The operation command.
src	The source path.
dest	The destination path. This parameter can be left empty.
perm	The operation permissions on the file.

Configure AuditLog

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **namespace** tab for the SmartData service.
   i. Click the **Configure** tab.

ii. Click the **namespace** tab in the Service Configuration section.



3. Perform the following operations:
   i. In the upper-right corner of the **Service Configuration** section, click **Custom Configuration**.
   ii. In the **Add Configuration Item** dialog box, configure the parameters described in the following table.

Parameter	Description	Required
namespace.auditlog.enable	■ true: Enable AuditLog. ■ false: Disable AuditLog.	Yes
namespace.auditlog.oss.uri	The OSS bucket where the log files generated by AuditLog are stored. Set this parameter in the format of oss://<yourbucket>/auditLog. Replace <yourbucket> with the OSS bucket name.	Yes
namespace.auditlog.oss.accessKey	The AccessKey ID used to access the OSS bucket.	No
namespace.auditlog.oss.accessSecret	The AccessKey secret used to access the OSS bucket.	No
namespace.auditlog.oss.endpoint	The endpoint of the OSS bucket.	No

   iii. In the upper-right corner of the Service Configuration section, click **Deploy Client Configuration**.
   iv. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   v. In the **Confirm** message, click **OK**.
4. Restart Namespace Service.
   i. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

5. Configure a retention period for log files.

   OSS provides the lifecycle management feature for you to manage the lifecycles of files in OSS. You can use this feature to customize a retention period for the log files generated by AuditLog.

   i. Log on to the OSS console.
   ii. In the left-side navigation pane, click Buckets. On the page that appears, click the name of the created bucket.
   iii. In the left-side navigation pane, choose **Basic Settings > Lifecycle**. In the **Lifecycle** section, click **Configure**.
   iv. Click **Create Rule**. In the **Create Rule** dialog box, configure the parameters.

   For more information, see Configure lifecycle rules.

   v. Click **OK**.

Analyze log files

JindoFS allows you to use Shell commands to analyze the log files generated by AuditLog. You can run a MapReduce job in the EMR console to analyze the most active commands or IP addresses in the log files. The analysis command is `jindo auditlog`.

The following table describes the parameters in the command.

Parameter	Description	Required
--src	The OSS bucket where the log files generated by AuditLog are stored. By default, this parameter is set to the value of the namespace.auditlog.oss.uri parameter specified in Step 3. You can customize a value.	No
--ns	The namespace you want to analyze. By default, all namespaces in block storage mode are analyzed.	No
--type	The analysis objects: • ip: the most active IP addresses • cmd: the most active commands	Yes
--min	The time range in minutes.	No
--day	The time range in days. day 1 indicates the current day.	**Note** You can specify only one of --min and --day.

In the EMR console, create a MapReduce job and run the jindo audit command. Example:

```
jindo auditlog --src oss://<yourbucket>/auditlog/ --ns test --type ip --day 1 --top 2
```

The following information is returned:

```
16 openFileStatusRequest
6  deleteFileletRequest
```

11.3. JindoTable

11.3.1. Use EMR JindoCube

JindoCube is supported in E-MapReduce (EMR) V3.24.0 and later. This topic describes how to install, deploy, and use JindoCube in EMR.

Prerequisites

A table or view is created.

Overview

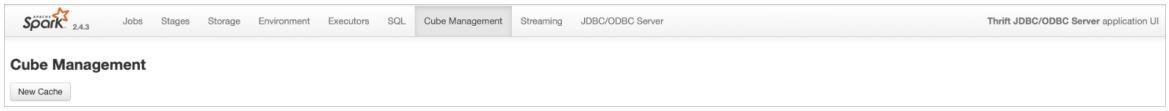
JindoCube is an advanced feature of EMR Spark. It implements pre-calculation to accelerate data processing. It improves query performance by dozens or even hundreds of times. You can persist data of a view to a data source supported by EMR Spark, such as HDFS or Object Storage Service (OSS). EMR Spark automatically discovers available persistent data and optimizes the execution plans for the data. This process is completely visible to users. JindoCube applies to business scenarios where a fixed query mode is used. JindoCube pre-calculates and pre-organizes data to accelerate data queries. For example, you can use JindoCube to implement multidimensional online analytical processing (MOLAP), generate data reports, create data dashboards, and synchronize data across clusters.

Install and deploy JindoCube

JindoCube is an advanced feature of EMR Spark. You can use JindoCube to accelerate all Dataset, DataFrame API, and SQL tasks that are submitted by using EMR Spark. No additional component deployment or maintenance is required.

1. Manage JindoCube on the Spark web UI.

   You can manage JindoCube on the Spark web UI. For example, you can create, delete, or update JindoCube caches. After you create a JindoCube cache, it automatically accelerates all Spark queries in the cluster. You can use the **spark.sql.cache.tab.display** parameter to specify whether to display the Cube Management tab on the Spark web UI. You can specify this parameter on the configuration page of the Spark service in the EMR console or in a Spark task submission command. The default value of this parameter is **false**.

   You can also use the **spark.sql.cache.useDatabase** parameter to create a database based on your business requirements and store the view for which you want to create a cache in this database. If partitioned tables are used, you can use the **spark.sql.cache.cacheByPartition** parameter to specify whether to cache data by partition.

Parameter	Description	Example

Parameter	Description	Example
spark.sql.cache.tab.display	Specifies whether to display the Cube Management tab.	true
spark.sql.cache.useDatabase	The databases where cubes are stored.	db1,db2,dbn
spark.sql.cache.cacheByPartition	Specifies whether to store cubes by partition.	true

2. Optimize Spark queries.

   You can use the **spark.sql.cache.queryRewrite** parameter to specify whether to use a JindoCube cache to accelerate Spark queries. You can specify this parameter at the cluster, session, or SQL statement level. The default value is **true**.

Use JindoCube

1. Create a JindoCube cache.
   i. Log on to the Alibaba Cloud EMR console by using your Alibaba Cloud account.
   ii. Click the **Cluster Management** tab.
   iii. Find the cluster where you want to create a JindoCube cache and click the cluster ID.
   iv. In the left-side navigation pane, click **Connect Strings**.
   v. On the **Public Connect Strings** page, click the link of YARN UI to go to the YARN web UI as a Knox user.

      For more information about Knox, see Knox.
   vi. Click **ApplicationMaster** in the row where **Application Type** is **SPARK** and **Name** is **Thrift JDBC/ODBC Server**.
   vii. On the Spark web UI, click the **Cube Management** tab.
   viii. Click **New Cache**.

      Select a table or view, and click raw cache or cube cache in the action column. You can create two types of caches:

- Raw cache: persists the data of a table or view based on the cache configurations.

  When you create a raw cache, specify the parameters described in the following table.

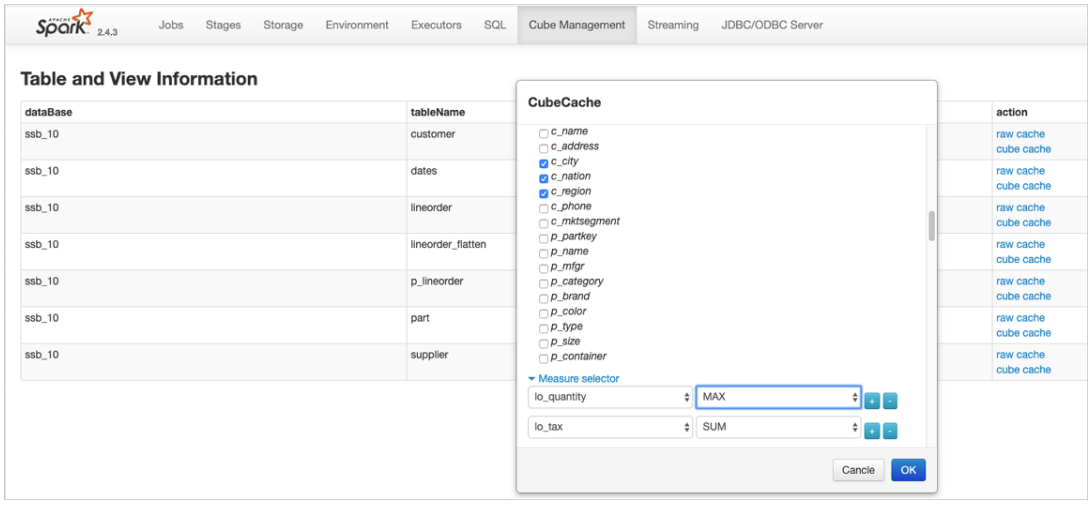
Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Column Selector	The columns from which you want to cache data.	Yes
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No

- Cube cache: builds a cube based on the raw data of a table or view and persists the cube data.

  When you create a cube cache, specify the parameters described in the following table.

Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Dimension Selector	The dimension fields that are used to build a cube.	Yes
Measure Selector	The measure fields and pre-calculation functions that are used to build a cube.	Yes

Parameter	Description	Required
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No



JindoCube builds a cube based on specified dimensions and measures. The following SQL statement shows the configurations of the cube cache in the preceding figure:

```
SELECT c_city, c_nation, c_region, MAX(lo_quantity), SUM(lo_tax)
FROM lineorder_flatten
GROUP BY c_city, c_nation, c_region;
```

JindoCube calculates the finest-grained combination of dimensions. Powered by the onsite computing capability of Spark, JindoCube implements re-aggregation to optimize the queries where a coarse-grained combination of dimensions is used. You must use the pre-calculation functions supported by JindoCube to define a cube cache. The following table lists the supported pre-calculation functions and the aggregate functions that correspond to them.

Aggregate function	Pre-calculation function
COUNT	COUNT
SUM	SUM
MAX	MAX
MIN	MIN
AVG	COUNT and SUM
COUNT (DISTINCT)	PRE_COUNT_DISTINCT
APPROX_COUNT_DISTINCT	PRE_APPROX_COUNT_DISTINCT

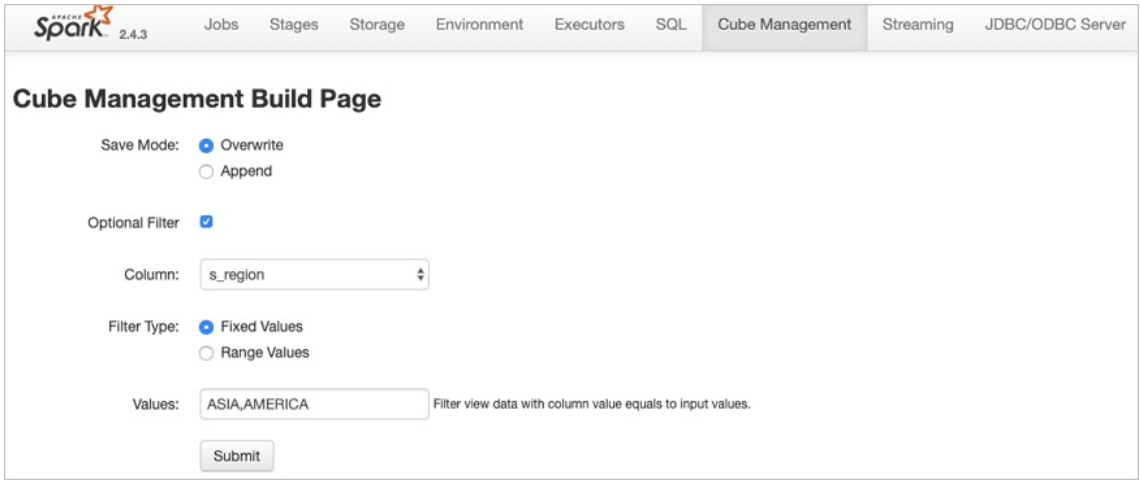
All created caches are displayed on the **Cube Management** tab. You can click **Detail** for a cache to view its details, including its basic information, partition information, building information, and building history.

2. Build a JindoCube cache.

   During the creation of a cache, metadata is prepared, but no data is persisted. Therefore, after you create a cache, you must continue to build the cache. This way, the related data can be persisted to permanent storage, such as HDFS or OSS. In addition, the source table of a cache may be appended with new data or updated. In this case, you must update the data in the cache to ensure data consistency. You can use one of the following methods to build a cache:

   - Manually build a cache

     Click Build Cache to manually build a cache. On the page that appears, configure the parameters, as shown in the following figure.



     The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.

Optional Filter	You can select this parameter to add filter conditions. This way, JindoCube filters data based on the filter conditions before it persists the data.
	■ Column: the field based on which data is filtered.
	■ Filter Type: specifies whether to filter data by fixed values or by value range.
	  ■ Fixed Values: If this option is selected, specify one or more values. Separate values with commas (,).
	  ■ Range Values: If this option is selected, specify a minimum value and a maximum value. You can leave the maximum value empty. In this case, the value range is the minimum value and all values greater than it.

The following SQL statement shows the configurations of the raw cache for the lineorder_flatten view in the preceding figure:

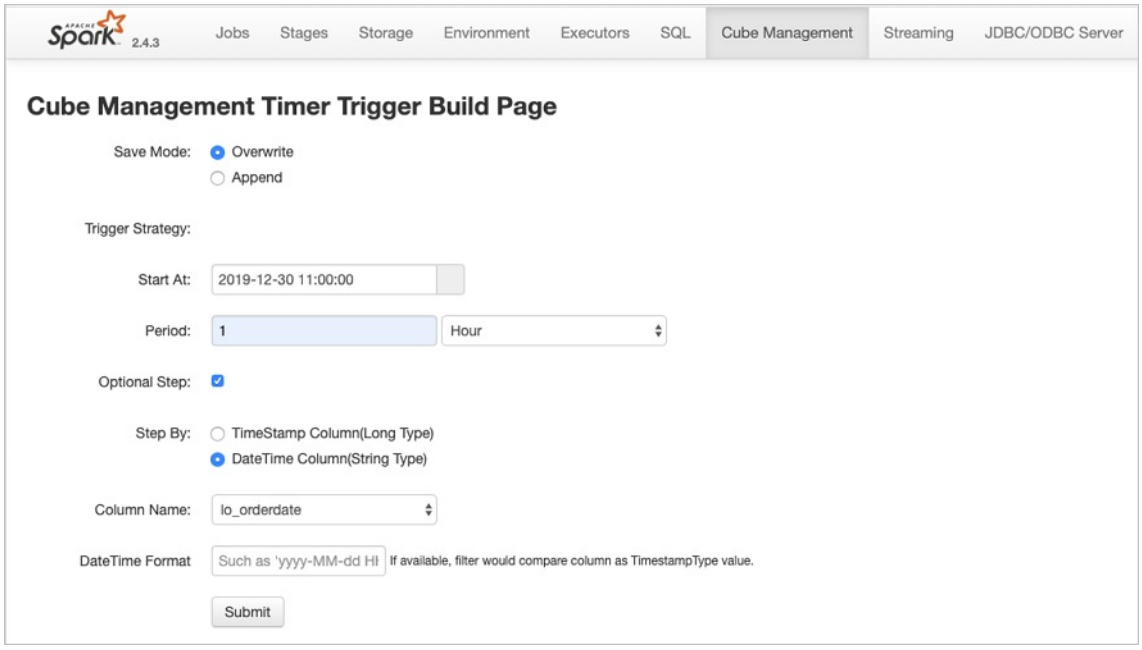
```
SELECT * FROM lineorder_flatten
WHERE s_region == 'ASIA' OR s_region == 'AMERICA';
```

Click **Submit** to submit the cache building task to a Spark cluster. You can view the task status in the Build Information section of the cache details page. After the task is complete, you can view the cache information in the Build History section.

> **Note** Spark writes the cached data to a specified directory in the same way common Spark tasks write data to a table or directory. If multiple users concurrently build a cache whose data format is Parquet, JSON, or ORC, the cached data may be inaccurate and invalid. If concurrent cache building operations or updates are inevitable, we recommend that you use a data format that supports concurrent writes, for example, Delta.

- Configure periodic cache building

  Click Trigger Period Build to configure a periodic cache update policy to ensure data consistency between the cache and source table. On the page that appears, configure the parameters, as shown in the following figure.



The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.
Trigger Strategy	Specifies the start time of the first building task, and the building interval. ■ Start At: Select or enter the time you want to trigger the first building task. Format: yyyy-MM-dd hh:mm:ss. ■ Period: Specify the interval at which cache building tasks are triggered.

Parameter	Description
Optional Step	Specifies the filter conditions for cache building. JindoCube updates the cache in incremental mode based on the filter conditions and interval. If you do not select Optional Step, JindoCube updates the entire cache each time. ■ Step By: Specify the type of the field that you want to update in incremental mode. TimeStamp Column(Long Type) indicates a timestamp field of the LONG type. DateTime Column(String Type) indicates a datetime field of the STRING type. ■ Column Name: Specify the name of the field that you want to update in incremental mode.

On the cache details page, you can view the configured cache update policy. You can click Cancel Period Build to cancel periodic updates. You can view the information of a complete building task in the Build History section.

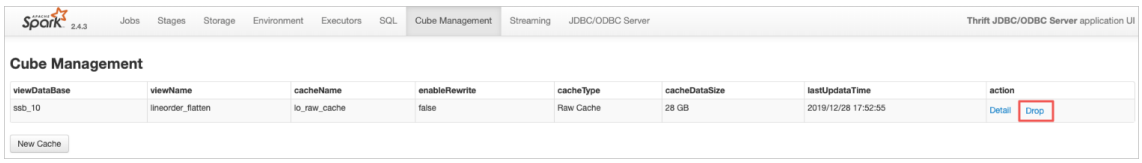
> **Note**
>
> - Periodic building tasks run on a Spark cluster. The related configurations are all saved in SparkContext, and the tasks are periodically triggered by Spark Driver. If you release the Spark cluster, periodic building is disabled.
> - The Build History section displays the following information of each complete task in the current Spark cluster: start time, end time, save mode, building conditions, and final status. After you release the Spark cluster, the information in the Build History section is cleared.

3. Manage a JindoCube cache.

   After you create and build a JindoCube cache, you can manage it on the Cube Management page.

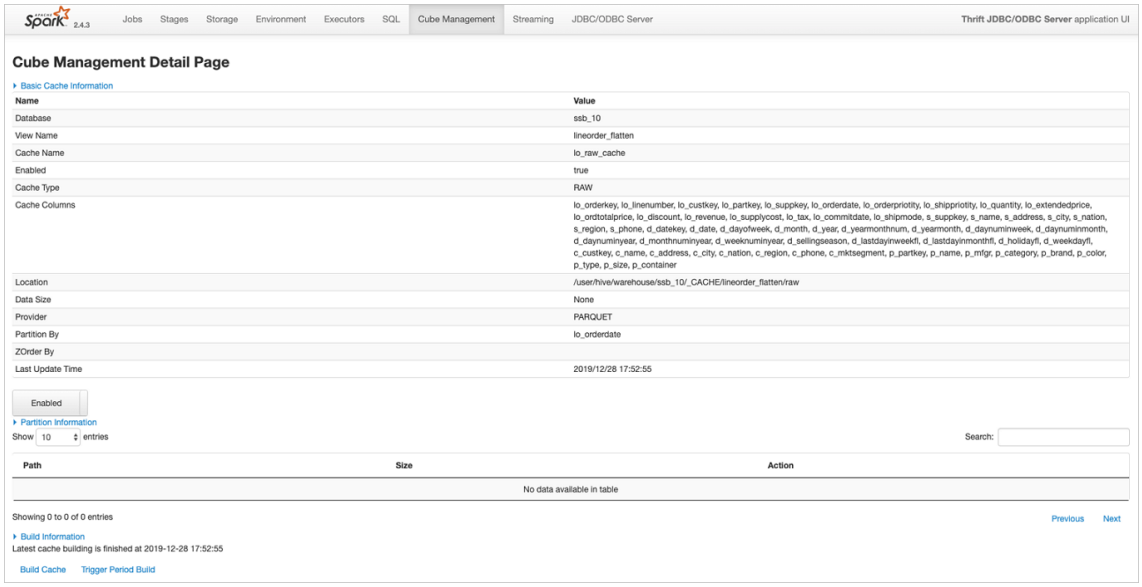
   - Delete a cache

     On the Cube Management page, find the cache you want to delete, and click **Drop** in the action column. All the metadata and data of the cache are deleted.



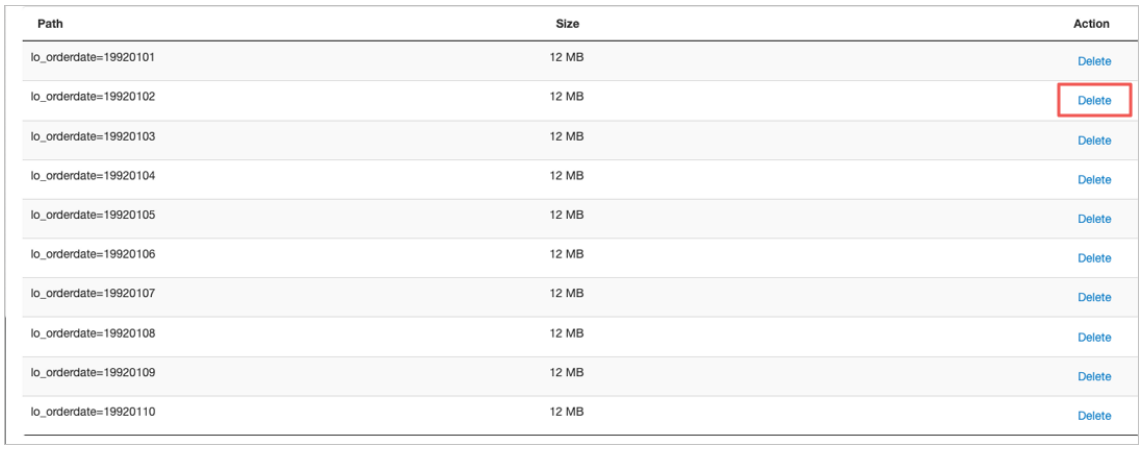
   - Enable and disable optimization of Spark queries

     JindoCube allows you to specify whether to optimize Spark queries at the cache level. On the details page of a cache, click **Enabled** or **Disabled** in the lower part of the Basic Cache Information section to enable or disable the optimization.



- Delete partition data from a cache

  If data in a cache is stored by partition, you can delete the unnecessary partition data from the cache to release more space. On the details page of the cache, find the partition whose data you want to delete, and click **Delete** in the Action column.



Path	Size	Action
lo_orderdate=19920101	12 MB	Delete
lo_orderdate=19920102	12 MB	Delete
lo_orderdate=19920103	12 MB	Delete
lo_orderdate=19920104	12 MB	Delete
lo_orderdate=19920105	12 MB	Delete
lo_orderdate=19920106	12 MB	Delete
lo_orderdate=19920107	12 MB	Delete
lo_orderdate=19920108	12 MB	Delete
lo_orderdate=19920109	12 MB	Delete
lo_orderdate=19920110	12 MB	Delete

> **Note** Before you delete the data of a partition, make sure that the data is no longer needed. If you delete data that is required in subsequent queries, the queries return invalid results.

4. Optimize Spark queries.

   JindoCube optimizes view-based queries. After you have created a raw cache or cube cache for a view, EMR Spark overwrites the execution plan of queries on this view. The new execution plan still uses the original logical semantics but can directly access the data in the cache. This accelerates Spark queries. Use the lineorder_flatten view as an example. It is a wide table view generated by associating the lineorder table with other dimension tables. The following figure shows the information about this view.

```
0: jdbc:hive2://localhost:10001/ssb_10> desc extended lineorder_flatten;
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+-------------------------------+----------------------------------------------------+----------+
|           col_name            |                     data_type                      | comment  |
+-------------------------------+----------------------------------------------------+----------+
| lo_orderkey                   | bigint                                             | NULL     |
| lo_linenumber                 | bigint                                             | NULL     |
| lo_custkey                    | int                                                | NULL     |
| lo_partkey                    | int                                                | NULL     |
| lo_suppkey                    | int                                                | NULL     |
| lo_orderdate                  | int                                                | NULL     |
| lo_orderpriotity              | string                                             | NULL     |
| lo_shippriotity               | int                                                | NULL     |
| lo_quantity                   | bigint                                             | NULL     |
| lo_extendedprice              | bigint                                             | NULL     |
| lo_ordtotalprice              | bigint                                             | NULL     |
| lo_discount                   | bigint                                             | NULL     |
| lo_revenue                    | bigint                                             | NULL     |
| lo_supplycost                 | bigint                                             | NULL     |
| lo_tax                        | bigint                                             | NULL     |
| lo_commitdate                 | int                                                | NULL     |
| lo_shipmode                   | string                                             | NULL     |
| s_suppkey                     | int                                                | NULL     |
| s_name                        | string                                             | NULL     |
| s_address                     | string                                             | NULL     |
| s_city                        | string                                             | NULL     |
| s_nation                      | string                                             | NULL     |
| s_region                      | string                                             | NULL     |
| s_phone                       | string                                             | NULL     |
| d_datekey                     | int                                                | NULL     |
| d_date                        | string                                             | NULL     |
| d_dayofweek                   | string                                             | NULL     |
| d_month                       | string                                             | NULL     |
| d_year                        | int                                                | NULL     |
| d_yearmonthnum                | int                                                | NULL     |
| d_yearmonth                   | string                                             | NULL     |
| d_daynuminweek                | int                                                | NULL     |
| d_daynuminmonth               | int                                                | NULL     |
| d_daynuminyear                | int                                                | NULL     |
| d_monthnuminyear              | int                                                | NULL     |
| d_weeknuminyear               | int                                                | NULL     |
| d_sellingseason               | string                                             | NULL     |
| d_lastdayinweekfl             | int                                                | NULL     |
| d_lastdayinmonthfl            | int                                                | NULL     |
| d_holidayfl                   | int                                                | NULL     |
| d_weekdayfl                   | int                                                | NULL     |
| c_custkey                     | int                                                | NULL     |
| c_name                        | string                                             | NULL     |
| c_address                     | string                                             | NULL     |
| c_city                        | string                                             | NULL     |
| c_nation                      | string                                             | NULL     |
| c_region                      | string                                             | NULL     |
| c_phone                       | string                                             | NULL     |
| c_mktsegment                  | string                                             | NULL     |
| p_partkey                     | int                                                | NULL     |
| p_name                        | string                                             | NULL     |
| p_mfgr                        | string                                             | NULL     |
| p_category                    | string                                             | NULL     |
| p_brand                       | string                                             | NULL     |
| p_color                       | string                                             | NULL     |
| p_type                        | string                                             | NULL     |
| p_size                        | int                                                | NULL     |
| p_container                   | string                                             | NULL     |
|                               |                                                    |          |
| # Detailed Table Information  |                                                    |          |
| Database                      | ssb_10                                             |          |
| Table                         | lineorder_flatten                                  |          |
| Owner                         | hadoop                                             |          |
| Created Time                  | Sat Dec 28 17:30:02 CST 2019                       |          |
| Last Access                   | Thu Jan 01 08:00:00 CST 1970                       |          |
| Created By                    | Spark 2.2 or prior                                 |          |
| Type                          | VIEW                                               |          |
| View Text                     | SELECT `lineorder`.`lo_orderkey`, `lineorder`.`lo_linenumber`, `lineorder`.`lo_custkey`, `lineorder`.`lo_partkey`, `
lineorder`.`lo_suppkey`, `lineorder`.`lo_orderdate`, `lineorder`.`lo_orderpriotity`, `lineorder`.`lo_shippriotity`, `lineorder`.`lo_quantity`, `lineor
der`.`lo_extendedprice`, `lineorder`.`lo_ordtotalprice`, `lineorder`.`lo_discount`, `lineorder`.`lo_revenue`, `lineorder`.`lo_supplycost`, `lineorder`
.`lo_tax`, `lineorder`.`lo_commitdate`, `lineorder`.`lo_shipmode`, `supplier`.`s_suppkey`, `supplier`.`s_name`, `supplier`.`s_address`, `supplier`.`s_
city`, `supplier`.`s_nation`, `supplier`.`s_region`, `supplier`.`s_phone`, `dates`.`d_datekey`, `dates`.`d_date`, `dates`.`d_dayofweek`, `dates`.`d_mo
nth`, `dates`.`d_year`, `dates`.`d_yearmonthnum`, `dates`.`d_yearmonth`, `dates`.`d_daynuminweek`, `dates`.`d_daynuminmonth`, `dates`.`d_daynuminyear`
, `dates`.`d_monthnuminyear`, `dates`.`d_weeknuminyear`, `dates`.`d_sellingseason`, `dates`.`d_lastdayinweekfl`, `dates`.`d_lastdayinmonthfl`, `dates`
.`d_holidayfl`, `dates`.`d_weekdayfl`, `customer`.`c_custkey`, `customer`.`c_name`, `customer`.`c_address`, `customer`.`c_city`, `customer`.`c_nation`
, `customer`.`c_region`, `customer`.`c_phone`, `customer`.`c_mktsegment`, `part`.`p_partkey`, `part`.`p_name`, `part`.`p_mfgr`, `part`.`p_category`, `
part`.`p_brand`, `part`.`p_color`, `part`.`p_type`, `part`.`p_size`, `part`.`p_container` FROM `ssb_10`.`lineorder`, `ssb_10`.`supplier`, `ssb_10`.`da
tes`, `ssb_10`.`customer`, `ssb_10`.`part`
WHERE `lineorder`.`lo_orderdate` = `dates`.`d_datekey` AND `lineorder`.`lo_custkey` = `customer`.`c_custkey` AND `lineorder`.`lo_suppkey` = `supplier`
.`s_suppkey` AND `lineorder`.`lo_partkey` = `part`.`p_partkey` |          |
| Table Properties              | [transient_lastDdlTime=1577677599]                 |          |
| Serde Library                 | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe |          |
| InputFormat                   | org.apache.hadoop.mapred.TextInputFormat           |          |
| OutputFormat                  | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat |          |
| Storage Properties            | [serialization.format=1]                           |          |
+-------------------------------+----------------------------------------------------+----------+
73 rows selected (0.057 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

The following figure shows the execution plan of simple queries on the lineorder_flatten view.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(11) SortMergeJoin [lo_partkey#313], [p_partkey#359], Inner
   :- *(8) Sort [lo_partkey#313 ASC NULLS FIRST], false, 0
   :  +- Exchange hashpartitioning(lo_partkey#313, 200)
   :     +- *(7) SortMergeJoin [lo_custkey#312], [c_custkey#351], Inner
   :        :- *(4) Sort [lo_custkey#312 ASC NULLS FIRST], false, 0
   :        :  +- Exchange hashpartitioning(lo_custkey#312, 200)
   :        :     +- *(3) BroadcastHashJoin [lo_orderdate#315], [d_datekey#334], Inner, BuildRight
   :        :        :- *(3) BroadcastHashJoin [lo_suppkey#314], [s_suppkey#327], Inner, BuildRight
   :        :        :  :- *(3) Filter (((isnotnull(lo_suppkey#314) && isnotnull(lo_orderdate#315)) && isnotnull(lo_custkey#312)) && isnotnull(lo_part
key#313))
   :        :        :  :  +- Scan hive ssb_10.lineorder [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_ord
erdate#315, lo_orderpriotity#316, lo_shippriotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#32
2L, lo_supplycost#323L, lo_tax#324L, lo_commitdate#325, lo_shipmode#326], HiveTableRelation `ssb_10`.`lineorder`, org.apache.hadoop.hive.serde2.lazy.L
azySimpleSerDe, [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_orderdate#315, lo_orderpriotity#316, lo_ship
priotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#322L, lo_supplycost#323L, lo_tax#324L, lo_c
ommitdate#325, lo_shipmode#326]
   :        :        :  +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :        :     +- *(1) Filter isnotnull(s_suppkey#327)
   :        :        :        +- Scan hive ssb_10.supplier [s_suppkey#327, s_name#328, s_address#329, s_city#330, s_nation#331, s_region#332, s_phone#
333], HiveTableRelation `ssb_10`.`supplier`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [s_suppkey#327, s_name#328, s_address#329, s_city#330
, s_nation#331, s_region#332, s_phone#333]
   :        :        +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :           +- *(2) Filter isnotnull(d_datekey#334)
   :        :              +- Scan hive ssb_10.dates [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearm
onth#340, d_daynuminweek#341, d_daynuminmonth#342, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweek
fl#347, d_lastdayinmonthfl#348, d_holidayfl#349, d_weekdayfl#350], HiveTableRelation `ssb_10`.`dates`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSe
rDe, [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearmonth#340, d_daynuminweek#341, d_daynuminmonth#34
2, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweekfl#347, d_lastdayinmonthfl#348, d_holidayfl#349,
 d_weekdayfl#350]
   :        +- *(6) Sort [c_custkey#351 ASC NULLS FIRST], false, 0
   :           +- Exchange hashpartitioning(c_custkey#351, 200)
   :              +- *(5) Filter ((isnotnull(c_region#356) && (c_region#356 = ASIA)) && isnotnull(c_custkey#351))
   :                 +- Scan hive ssb_10.customer [c_custkey#351, c_name#352, c_address#353, c_city#354, c_nation#355, c_region#356, c_phone#357, c_mk
tsegment#358], HiveTableRelation `ssb_10`.`customer`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [c_custkey#351, c_name#352, c_address#353, c
_city#354, c_nation#355, c_region#356, c_phone#357, c_mktsegment#358]
   +- *(10) Sort [p_partkey#359 ASC NULLS FIRST], false, 0
      +- Exchange hashpartitioning(p_partkey#359, 200)
         +- *(9) Filter isnotnull(p_partkey#359)
            +- Scan hive ssb_10.part [p_partkey#359, p_name#360, p_mfgr#361, p_category#362, p_brand#363, p_color#364, p_type#365, p_size#366, p_conta
iner#367], HiveTableRelation `ssb_10`.`part`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [p_partkey#359, p_name#360, p_mfgr#361, p_category#3
62, p_brand#363, p_color#364, p_type#365, p_size#366, p_container#367] |
+----------------------------------------------------+
1 row selected (0.435 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

After a raw cache is created and built for the lineorder_flatten view, the same queries are performed on the view. EMR Spark automatically uses the cache to optimize the original execution plan. The following figure shows the new execution plan.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:17:47 INFO [main] HiveConf: Found configuration file file:/etc/ecm/spark-conf-2.4.3-hadoop2.8-1.4.2/hive-site.xml
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(1) Project [lo_orderkey#128L, lo_linenumber#129L, lo_custkey#130, lo_partkey#131, lo_suppkey#132, lo_orderdate#133, lo_orderpriotity#134, lo_ship
priotity#135, lo_quantity#136L, lo_extendedprice#137L, lo_ordtotalprice#138L, lo_discount#139L, lo_revenue#140L, lo_supplycost#141L, lo_tax#142L, lo_c
ommitdate#143, lo_shipmode#144, s_suppkey#145, s_name#146, s_address#147, s_city#148, s_nation#149, s_region#150, s_phone#151, ... 34 more fields]
   +- *(1) Filter (isnotnull(c_region#174) && (c_region#174 = ASIA))
      +- *(1) FileScan parquet [lo_orderkey#128L,lo_linenumber#129L,lo_custkey#130,lo_partkey#131,lo_suppkey#132,lo_orderpriotity#134,lo_shippriotity#
135,lo_quantity#136L,lo_extendedprice#137L,lo_ordtotalprice#138L,lo_discount#139L,lo_revenue#140L,lo_supplycost#141L,lo_tax#142L,lo_commitdate#143,lo_
shipmode#144,s_suppkey#145,s_name#146,s_address#147,s_city#148,s_nation#149,s_region#150,s_phone#151,d_datekey#152,... 34 more fields] Batched: true,
Format: Parquet, Location: InMemoryFileIndex[/user/hive/warehouse/ssb_10/_CACHE/lineorder_flatten/raw], PartitionCount: 2406, PartitionFilters: [], Pu
shedFilters: [IsNotNull(c_region), EqualTo(c_region,ASIA)], ReadSchema: struct<lo_orderkey:bigint,lo_linenumber:bigint,lo_custkey:int,lo_partkey:int,l
o_suppkey:int,lo_or... |
+----------------------------------------------------+
```

The new execution plan no longer uses the calculation logic of the lineorder_flatten view. Instead, it directly accesses the cached data in HDFS.

Usage notes

1. JindoCube does not ensure data consistency between the cache and source table. You must manually trigger data updates or configure a periodic update policy based on your requirements for data consistency.
2. JindoCube determines whether to use a cache to optimize execution plans based on the metadata of views. If cached data is incomplete when JindoCube optimizes an execution plan, the query results may be invalid or incomplete. The following operations or configurations may cause incomplete cached data: 1. Delete partition data that is required in subsequent queries on the cache details page. 2. Data required in subsequent queries is filtered out based on the filter conditions specified when you build or update the cache. 3. Required data has not been updated to the cache when the query is performed.

11.4. Toolset

11.4.1. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

- Java Development Kit (JDK) 8 is installed on your computer.
- An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain help information:

```
jindo distcp --help
```

   The following information is returned:

```
     --help           - Print help text
     --src=VALUE          - Directory to copy files from
     --dest=VALUE              - Directory to copy files to
     --parallelism=VALUE         - Copy task parallelism
     --outputManifest=VALUE       - The name of the manifest file
     --previousManifest=VALUE   -   The path to an existing manifest file
     --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
if specified
     --copyFromManifest   -   Copy from a manifest instead of listing a directory
     --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
     --srcPattern=VALUE   -   Include only source files matching this pattern
     --deleteOnSuccess   -   Delete input files after a successful copy
     --outputCodec=VALUE   -   Compression codec for output files
     --groupBy=VALUE   -   Pattern to group input files by
     --targetSize=VALUE   -   Target size for output files
     --enableBalancePlan   -   Enable plan copy task to make balance
     --enableDynamicPlan   -   Enable plan copy task dynamically
     --enableTransaction   -   Enable transation on Job explicitly
     --diff   -   show the difference between src and dest filelist
     --ossKey=VALUE   -   Specify your oss key if needed
     --ossSecret=VALUE   -   Specify your oss secret if needed
     --ossEndPoint=VALUE   -   Specify your oss endPoint if needed
     --policy=VALUE   -   Specify your oss storage policy
     --cleanUpPending   -   clean up the incomplete upload when distcp job finish
     --queue=VALUE   -   Specify yarn queuename if needed
     --bandwidth=VALUE   -   Specify bandwidth per map/reduce in MB if needed
     --s3Key=VALUE   -   Specify your s3 key
     --s3Secret=VALUE   -   Specify your s3 Sercet
     --s3EndPoint=VALUE   -   Specify your s3 EndPoint
```

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-
01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03
/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03
/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017
-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-0
1/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-0
1/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_tab
le/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/
03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/
03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromMani
fest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

> **Note** If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

--queue

--queue specifies the name of the YARN queue where the current DistCp task resides.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<your_bucket>/hourly_table --qu
eue yarnqueue
```

--bandwidth

--bandwidth specifies the bandwidth allocated to the current DistCp task. This prevents the task from using excessive bandwidth. Unit: MB/s.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.c
om --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy archive --parallelism 20
  ```

- Example of the IA storage class (`--ia`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table
  --policy ia --parallelism 20
  ```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

Use Amazon S3 as a data source

You can use the --s3Key, --s3Secret, and --s3EndPoint parameters in a command to specify information related to Amazon S3.

Example:

```
jindo distcp jindo-distcp-2.7.3.jar --src s3a://yourbucket/ --dest oss://<your_bucket>/hour
ly_table --s3Key yourkey --s3Secret yoursecret --s3EndPoint s3-us-west-1.amazonaws.com
```

You can configure the s3Key, s3Secret, and s3EndPoint parameters in the *core-site.xml* file of Hadoop. This way, you do not need to specify an AccessKey pair each time.

```
<configuration>
    <property>
        <name>fs.s3a.access.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3a.secret.key</name>
        <value>xxx</value>
    </property>
    <property>
        <name>fs.s3.endpoint</name>
        <value>s3-us-west-1.amazonaws.com</value>
    </property>
</configuration>
```

Example:

```
jindo distcp /tmp/jindo-distcp-2.7.3.jar --src s3://smartdata1/ --dest s3://smartdata1/tmp
--s3EndPoint  s3-us-west-1.amazonaws.com
```

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

Note If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

11.4.2. Use JindoFS FUSE

This topic describes how to access JindoFileSystem (JindoFS) from a Filesystem in Userspace (FUSE) client. FUSE allows you to use JindoFS in block storage mode and cache mode (only JFS Scheme supported).

Prerequisites

An E-MapReduce (EMR) cluster is created. For more information, see Create a cluster.

Context

FUSE is a method provided by the Linux kernel to mount a file system. You can use a FUSE client to map files in a JindoFS cluster to a local disk. Then, you can directly access data in the JindoFS cluster without the need to use the `hadoop fs -ls jfs://<namespace>/` command.

Mount a directory

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs
   ```

3. Run the following command to mount the directory:

   ```
   jindofs-fuse /mnt/jfs
   ```

   In this example, the */mnt/jfs/* directory is mounted.

Read and write files

1. Run the following command to query all the sub-directories under the /mnt/jfs/ directory:

   ```
   ls /mnt/jfs/
   ```

   All the namespaces that you have configured on the server are returned.

   ```
   test  testcache
   ```

2. Run the following command to query all the files in the test namespace:

   ```
   ls /mnt/jfs/test/
   ```

3. Run the following command to create a directory:

   ```
   mkdir /mnt/jfs/test/dir1
   ls /mnt/jfs/test/
   ```

4. Run the following command to write data into a file:

   ```
   echo "hello world" > /tmp/hello.txt
   cp /tmp/hello.txt /mnt/jfs/test/dir1/
   ```

5. Run the following command to read data from a file:

```
cat /mnt/jfs/test/dir1/hello.txt
```

The following information is returned:

```
hello world
```

The following example shows how to read data from and write data into files in Python:

1. Write data into a file in Python. Create the *write.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'w',encoding = 'utf-8') as f:
   f.write("my first file\n")
   f.write("This file\n\n")
   f.write("contains three lines\n")
```

2. Read data from a file in Python. Create the *read.py* script file, which contains the following content:

```
#! /usr/bin/env python36
with open("/mnt/jfs/test/test.txt",'r',encoding = 'utf-8') as f:
    lines = f.readlines()
    [print(x, end = '') for x in lines]
```

Run the following command to read the data that is written into the *test.txt* file:

```
[hadoop@emr-header-1 ~]$ ./read.py
```

The following information is returned:

```
my first file
This file
```

Unmount FUSE

> **Note** Perform the following steps on each node of the cluster in sequence.

1. Log on to the master node of the cluster in SSH mode. For more information, see Log on to a cluster.
2. Run the following command to unmount FUSE:

```
umount jindofs-fuse
```

If the `target is busy` error is returned, switch to another directory, stop all the programs that are reading data from and writing data into FUSE files, and then unmount FUSE.

12.SmartData 2.6.0-2.7.2

12.1. SmartData 2.6.X

SmartData 2.6.X delivers multiple new features and optimizes performance. For example, Tablestore and Raft instances can be used as the metadata storage backends of Namespace Service. Namespace Service can be deployed in high availability (HA) mode. The read/write performance, block storage mode, and cache mode of JindoFS are optimized.

New metadata storage methods

In earlier versions, you can use only RocksDB as a metadata storage backend. In SmartData 2.6.X, you can also use Tablestore or Raft instances as metadata storage backends. This way, JindoFS achieves the storage of metadata in the cloud.

RocksDB is the default metadata storage method. It is simple and efficient when you use JindoFS in cache mode and do not require Namespace Service in HA mode or high performance of metadata storage. If Tablestore or Raft instances are used, you can deploy Namespace Service in HA mode.

For more information about each metadata storage method, see the following topics:

- Use Tablestore instances to store metadata
- Use Raft-RocksDB-Tablestore to store metadata
- Use RocksDB to store metadata

Optimization of JindoFS usage modes

JindoFS supports the block storage mode and cache mode:

- Block storage mode: You can use JindoFS in this mode in the same way as in versions earlier than E-MapReduce (EMR) V3.26.3. For more information, see Use JindoFS in block storage mode.
- Cache mode: Multiple file access methods are supported in this mode. For example, you can use the same method to access files as in block storage mode. You can also use the original file access method of OSS. For more information, see Use JindoFS in cache mode.

Supported permissions

In block storage mode, you can run UNIX commands or use Ranger to manage permissions:

- UNIX allows you to grant the rwxrwxrwx permission on files.
- Ranger allows you to use wildcards in paths when you configure a permission.

For more information about permissions, see Manage JindoFS permissions.

12.2. JindoFS in block storage mode

12.2.1. Use JindoFS in block storage mode

The block storage mode ensures efficient read/write operations and high metadata accessibility. JindoFS stores data as blocks in OSS and caches data in local disks of clusters to accelerate data access. JindoFS uses Namespace Service to manage metadata and ensure high metadata accessibility. This topic describes how to use JindoFS in block storage mode.

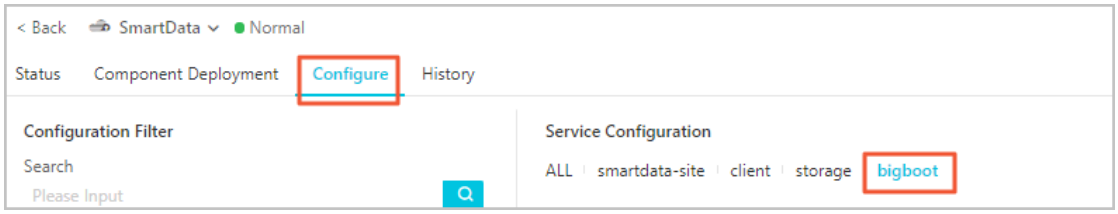
Context

The block storage mode of JindoFS has the following features:

- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.
- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Configure the block storage mode

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Configure bigboot parameters.
   i. Click the **Configure** tab.
   ii. Click the **bigboot** tab in the Service Configuration section.



3. Configure required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

   If you configure multiple namespaces, separate them with commas (,).

ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the **Add Configuration Item** dialog box, add the parameters described in the following table.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** We recommend that you set this parameter to a directory of an OSS bucket. The namespace stores data blocks in this directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to block.	block
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.	

iii. Click **OK**.

4. In the upper-right corner of the Service Configuration section, click **Save**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

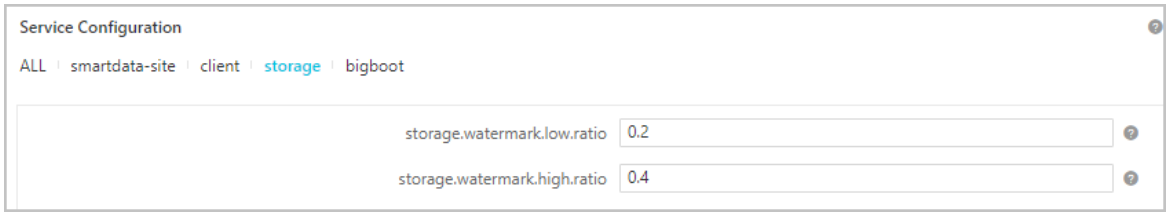
   After Namespace Service is restarted, you can use `jfs://test/<path_of_file>` to access files in JindoFS.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

   In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.

Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

   > **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

12.2.2. Use Tablestore instances to store metadata

Namespace Service of JindoFS supports different metadata storage methods. This topic describes how to configure a Tablestore instance as the metadata storage backend.

Prerequisites

- An E-MapReduce (EMR) cluster is created.

  For more information, see Create a cluster.

- A Tablestore instance is created. We recommend that you use a high-performance instance.

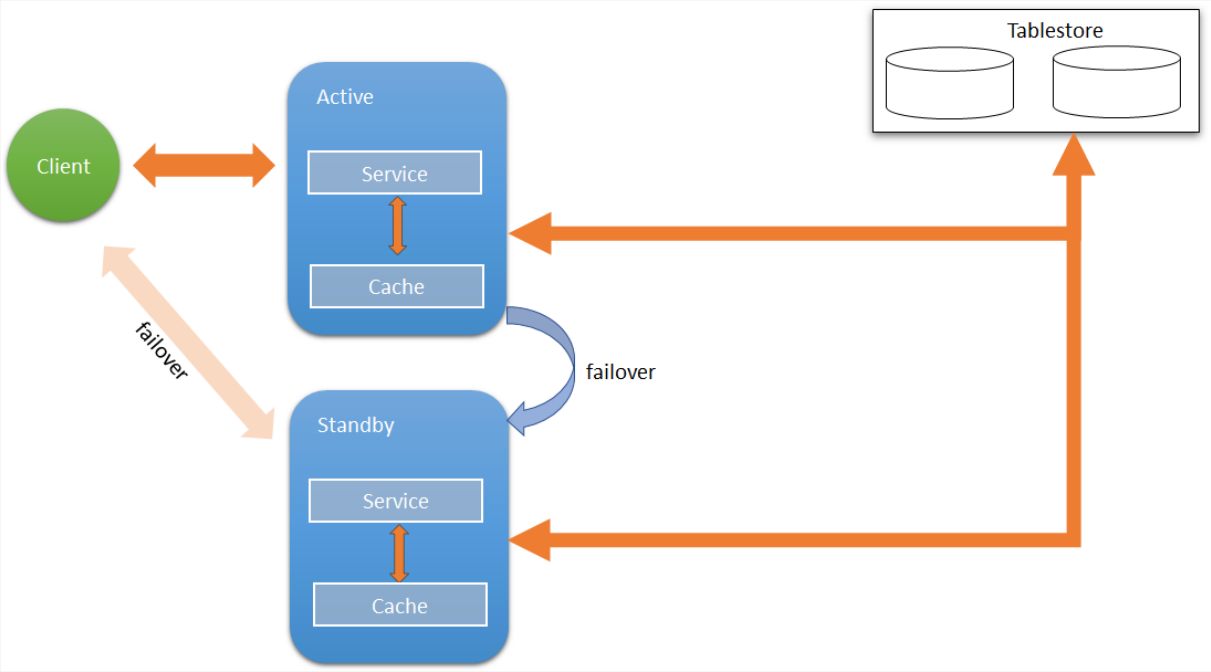
  For more information, see Create instances.

> **Note** You must enable the transaction feature.

Context

In SmartData 2.6.X and later, Namespace Service allows you to use Tablestore instances to store metadata. You can bind a Tablestore instance to each EMR JindoFS cluster. Namespace Service creates a Tablestore table for each namespace to manage and store metadata.

The following figure shows the structure of Namespace Service in high availability (HA) mode with a Tablestore instance as the metadata storage backend.



Configure a Tablestore instance

To use a Tablestore instance as the metadata storage backend, you must first bind the Tablestore instance to Namespace Service. Perform the following steps:

1. Go to the SmartData service.

   i.

   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

   iii. Click the **Cluster Management** tab.

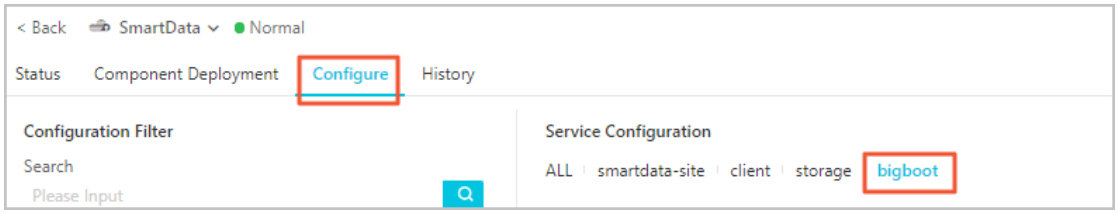
iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Configure bigboot parameters.

   i. Click the **Configure** tab.

   ii. Click the **bigboot** tab in the Service Configuration section.



3. Configure the parameters described in the following table.

   For example, you have created a Tablestore instance named emr-jfs in the China (Hangzhou) region. Your EMR cluster is deployed in a VPC network. You have obtained the AccessKey ID and AccessKey secret that are used to access the Tablestore instance.

Parameter	Description	Required	Example
namespace.backend.type	The backend storage type of Namespace Service. ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to ots.	Yes	ots
namespace.ots.instance	The name of the Tablestore instance.	Yes	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	No	kkkkkk
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	No	XXXXXX
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	Yes	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

4. Save the configurations.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

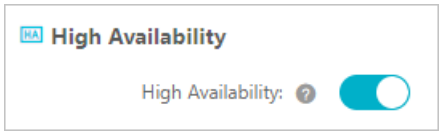
ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

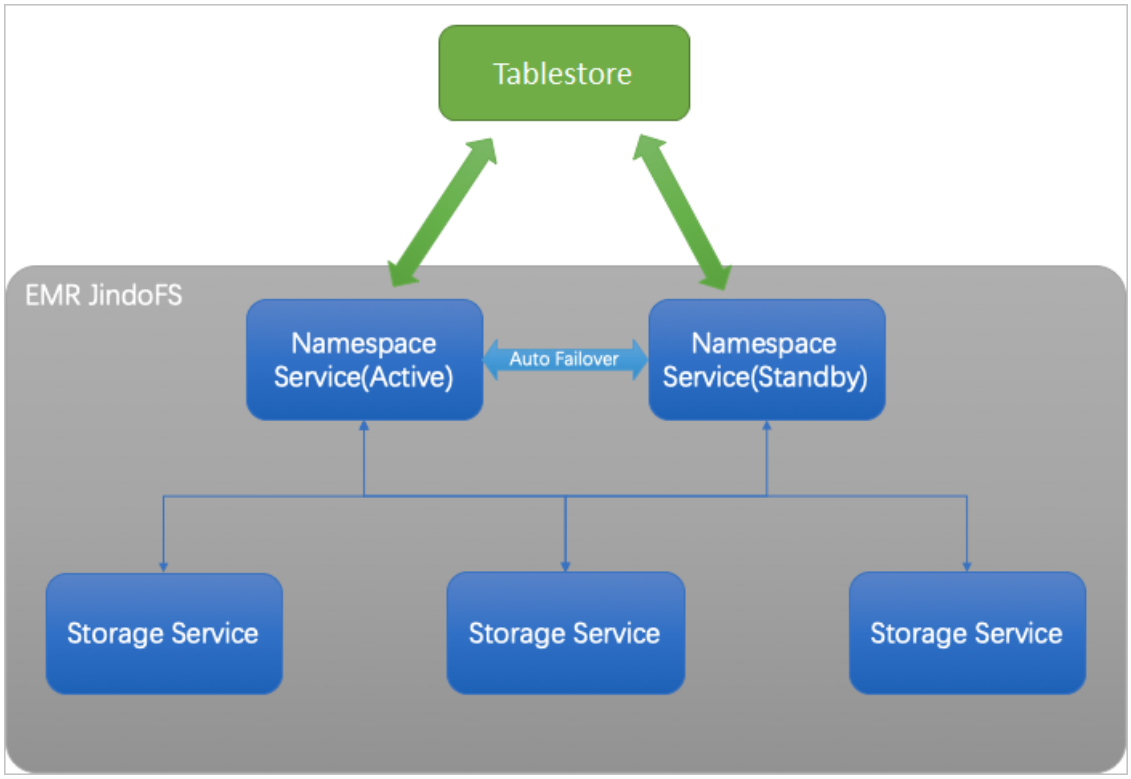
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

Configure a Tablestore instance (HA mode)

If your EMR cluster is deployed in HA mode, we recommend that you deploy Namespace Service in HA mode.



In HA mode, Namespace Service supports automatic failover. If the active namespace fails, the client automatically switches services to the standby namespace.



1. Go to the bigboot tab for the SmartData service and perform the following operations:

   i. Change the value of the **jfs.namespace.server.rpc-address** parameter to **emr-header-1:8101,emr-header-2:8101**.

   ii. In the upper-right corner of the Service Configuration section, click **Custom Configuration**. In the Add Configuration Item dialog box, add **namespace.backend.ots.ha** as a key and set this parameter to **true**.

   iii. Click **OK**.

iv. Save the configurations.

a. In the upper-right corner of the Service Configuration section, click **Save**.

b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

c. Click **OK**.

2. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.
3. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.

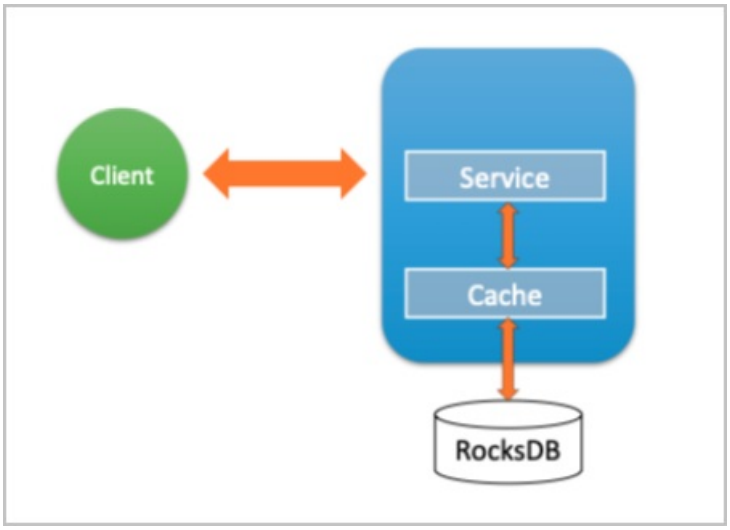
12.2.3. Use RocksDB to store metadata

Namespace Service of JindoFS supports different metadata storage backends. By default, RocksDB is used. This topic describes how to configure RocksDB as the metadata storage backend.

Context

RocksDB cannot be configured in high availability mode. If you need to configure a metadata storage backend in high availability mode, we recommend that you use Tablestore or Raft instances. For more information, see Use Tablestore instances to store metadata and Use Raft-RocksDB-Tablestore to store metadata.

The following figure shows the structure of a single RocksDB instance for Namespace Service.



Configure RocksDB as the metadata storage backend

1. Go to the SmartData service.

i.

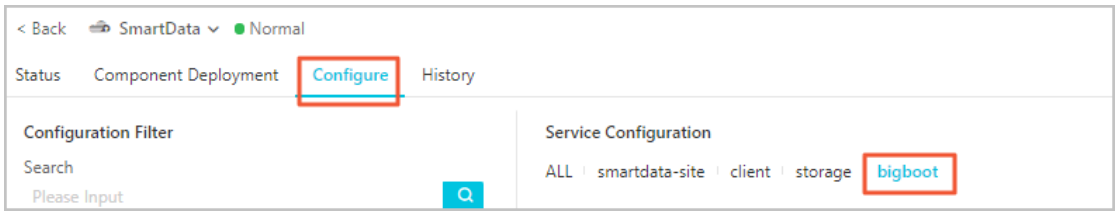
ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

iii. Click the **Cluster Management** tab.

iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Configure bigboot parameters.
   i. Click the **Configure** tab.
   ii. Click the **bigboot** tab in the Service Configuration section.



3. Set **namespace.backend.type** to **rocksdb**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

12.2.4. Use Raft-RocksDB-Tablestore to store metadata

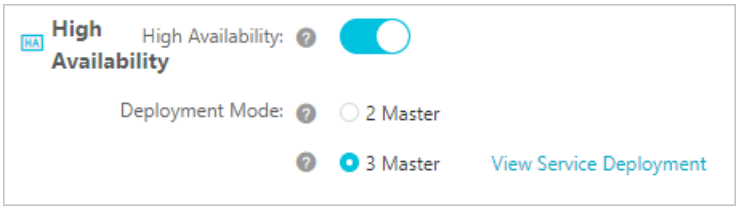
In E-MapReduce (EMR) V3.27.0 and later, you can use Raft-RocksDB-Tablestore to store the metadata managed by Namespace Service of JindoFS. When you create an EMR JindoFS cluster, create three master nodes and deploy a Raft instance on the master nodes. Each peer node of the Raft instance uses the local Raft-RocksDB to store metadata.

Prerequisites

- A Tablestore instance is created. We recommend that you use a high-performance instance. For more information, see Create instances.

  > **Note** You must enable the transaction feature.

- An EMR cluster that has three master nodes is created. For more information, see Create a cluster.

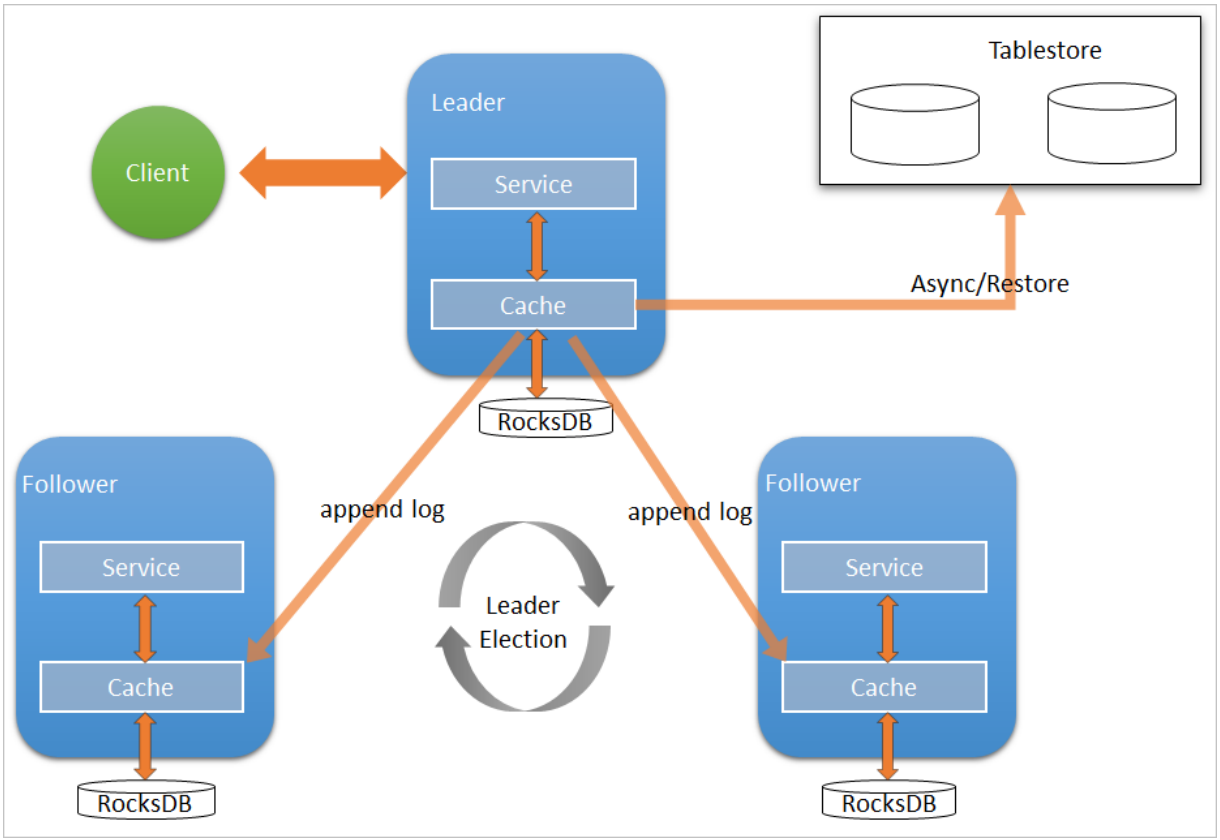


> **Note** If no deployment modes are available, submit a ticket.

Context

RocksDB supports data replication among three nodes based on the Raft protocol. You can bind an EMR cluster to a Tablestore instance and use the Tablestore instance as an additional storage medium for Namespace Service of JindoFS. EMR asynchronously uploads metadata from the local RocksDB to the Tablestore instance in real time.

The following figure shows the architecture of high-availability Raft-RocksDB-Tablestore for Namespace Service.



Configure a Raft instance as the local storage backend

1. Suspend all the SmartData components of an EMR cluster.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   vi. Select **Stop All Components** from the **Actions** drop-down list in the upper-right corner.
2. Configure Namespace Service parameters based on your business requirements.

   For more information, see Use JindoFS in block storage mode.
3. Go to the **bigboot** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **bigboot** tab.

4. Configure the parameters described in the following table on the **bigboot** tab.

Parameter	Description	Example
namespace.backend.type	The backend storage type of Namespace Service. ○ rocksdb ○ ots ○ raft Default value: rocksdb. Set this parameter to raft.	raft
namespace.backend.raft.initial-conf	The addresses of the three master nodes where the Raft instance is deployed. The value is fixed.	emr-header-1:8103:0,emr-header-2:8103:0,emr-header-3:8103:0
jfs.namespace.server.rpc-address	The addresses of the three master nodes that are used for the client to access the Raft instance. The value is fixed.	emr-header-1:8101,emr-header-2:8101,emr-header-3:8101

Note If you do not need to use Tablestore for remote storage, skip Step 5.

5. (Optional)Configure a Tablestore instance as the remote asynchronous storage backend.

   On the **bigboot** tab for the SmartData service, configure the parameters described in the following table.

Parameter	Description	Example
namespace.ots.instance	The name of the Tablestore instance.	emr-jfs
namespace.ots.accessKey	The AccessKey ID that is used to access the Tablestore instance.	*kkkkkk*
namespace.ots.accessSecret	The AccessKey secret that is used to access the Tablestore instance.	*XXXXXX*
namespace.ots.endpoint	The endpoint of the Tablestore instance. We recommend that you use a VPC endpoint.	*http://emr-jfs.cn-hangzhou.vpc.tablestore.aliyuncs.com*

Parameter	Description	Example
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false Set this parameter to true. Make sure that the initialization of the SmartData service is not completed. **Note** If the initialization is completed, the setting does not take effect because metadata has been generated in the local RocksDB.	true

6. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
7. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.

Recover metadata from a Tablestore instance

If you have configured a Tablestore instance as the remote asynchronous storage backend for your EMR cluster, a complete replica of JindoFS metadata is stored in the Tablestore instance. After you stop or release the EMR cluster, you can recover JindoFS metadata from the Tablestore instance to a new EMR cluster. This way, you can access the original files from the new EMR cluster.

1. (Optional)Prepare for the recovery.
   i. (Optional)Collect the metadata statistics of the original EMR cluster. The metadata statistics indicate the numbers of files and folders.

```
hadoop fs -count jfs://test/
      1596      1482809                 25 jfs://test/
  (Number of folders) (Number of files)
```

ii. Stop the jobs that are running on the original EMR cluster. You may need to wait for 30 to 120 seconds for EMR to synchronize all metadata of the cluster to the Tablestore instance. Run the following command to view the metadata status. If the command output contains `_synced=1` for the LEADER node, the latest metadata is synchronized to the Tablestore instance.

```
jindo jfs -metaStatus -detail
```

```
[RaftPeerImpl]
peer id: [redacted]
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 2335ms)
snapshot_timer: timeout(3600000ms) SCHEDULING(in 150305ms)
storage: [1, 624625]
disk_index: 624625
known_applied_index: 624625
last_log_id: (index=624625,term=2)
first_index_pinned: 624625
state_machine: Idle
last_committed_index: 624625
last_snapshot_index: 0
last_snapshot_term: 0
snapshot_status: IDLE
replicator_25769803789@[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=624261 ic=0
replicator_329853488332[redacted] next_index=624626  flying_append_entries_size=0 idle hc=2301 ac=623564 ic=0

OtsUploader: _lastStopIndex=624624, _synced=1
```

iii. Stop or release the original EMR cluster and make sure that no clusters are accessing the Tablestore instance.

2. Create an EMR cluster.

   Create an EMR cluster that resides in the same region as the Tablestore instance. Suspend all SmartData components. For more information, see Step 1 in Configure a Raft instance as the local storage backend.

3. Configure parameters for metadata recovery.

   On the **bigboot** tab for the SmartData service, configure the parameters described in the following table.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	false
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	true

4. Save the configurations.

i. In the upper-right corner of the Service Configuration section, click **Save**.

ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Select **Start All Components** from the **Actions** drop-down list in the upper-right corner.
6. After the SmartData service of the new EMR cluster is activated, EMR automatically recovers metadata from the Tablestore instance to the local Raft-RocksDB. You can run the following command to view the recovery progress:

```
jindo jfs -metaStatus -detail
```

If the state of the LEADER node is FINISH, the recovery is completed, as shown in the following figure.

```
[RaftPeerImpl]
peer_id: [redacted]:8103:0
state: LEADER
readonly: 0
term: 2
conf_index: 1
peers: [redacted]
changing_conf: NO    stage: STAGE_NONE
election_timer: timeout(5000ms) STOPPED
vote_timer: timeout(5000ms) STOPPED
stepdown_timer: timeout(5000ms) SCHEDULING(in 3382ms)
snapshot_timer: timeout(600000ms) SCHEDULING(in 474855ms)
storage: [1, 153]
disk_index: 153
known_applied_index: 153
last_log_id: (index=153,term=2)
first_index_pinned: 1
state_machine: Idle
last_committed_index: 153
last_snapshot_index: 1
last_snapshot_term: 2
snapshot_status: IDLE
replicator_1116691496965@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0
replicator_3311419785217@[redacted]: next_index=154  flying_append_entries_size=0 idle hc=262 ac=154 ic=0

[Recovery From OTS Status]
state: FINISH
total rows: 1484409
table `jfs_block_test` 2 rows.
table `jfs_namespace_cache_ns` 1 rows.
table `jfs_namespace_test` 1484406 rows.
```

7. (Optional)Check whether the numbers of files and folders of the new EMR cluster are consistent with those of the original EMR cluster.

   The new EMR cluster is in recovery mode and is read-only.

```
# Count the numbers of files and folders for the new EMR cluster. The results are consi
stent with those of the original EMR cluster.
[hadoop@emr-header-1 ~]$ hadoop fs -count jfs://test/
        1596      1482809                 25 jfs://test/
# Use the CAT or GET command to read data from the files.
[hadoop@emr-header-1 ~]$ hadoop fs -cat jfs://test/testfile
this is a test file
# View file directories.
[hadoop@emr-header-1 ~]$ hadoop fs -ls jfs://test/
Found 3 items
drwxrwxr-x   - root   root            0 2020-03-25 14:54 jfs://test/emr-header-1.cluste
r-50087
-rw-r-----   1 hadoop hadoop          5 2020-03-25 14:50 jfs://test/haha-12096RANDOM.tx
t
-rw-r-----   1 hadoop hadoop         20 2020-03-25 15:07 jfs://test/testfile
# Remove a file. A read-only error is returned.
[hadoop@emr-header-1 ~]$ hadoop fs -rm jfs://test/testfile
java.io.IOException: ErrorCode : 25021 , ErrorMsg: Namespace is under recovery mode, an
d is read-only.
```

8. Enable asynchronous upload to Tablestore and disable recovery from Tablestore.

   On the **bigboot** tab for the SmartData service, configure the parameters described in the following table.

Parameter	Description	Required value
namespace.backend.raft.async.ots.enabled	Specifies whether to enable asynchronous upload to Tablestore. Valid values: ○ true ○ false	true
namespace.backend.raft.recovery.mode	Specifies whether to enable metadata recovery from Tablestore. Valid values: ○ true ○ false	false

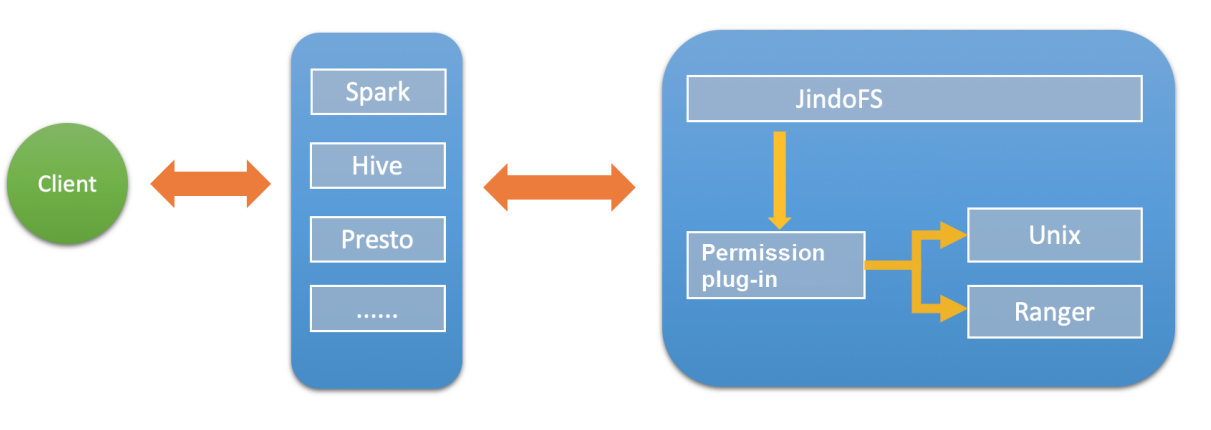
9. Restart the new EMR cluster.
   i. Click the **Cluster Management** tab.
   ii. On the **Cluster Management** page, find the cluster. In the Actions column of this cluster, click **More** and select **Restart**.

12.2.5. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode. You can run UNIX commands or use Ranger to manage permissions.

Context

To use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of E-MapReduce (EMR) and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

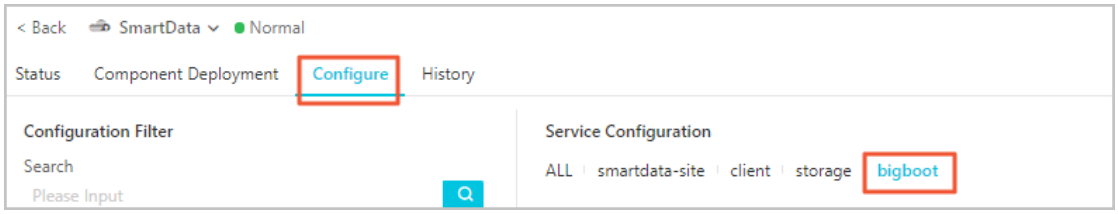


In block storage mode, you can run UNIX commands or use Ranger to manage permissions:

- UNIX allows you to grant the rwxrwxrwx permission on files.
- Ranger allows you to use wildcards in paths when you configure a permission.

Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Configure bigboot parameters.
   i. Click the **Configure** tab.
   ii. Click the **bigboot** tab in the Service Configuration section.



3. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Restart Namespace Service.

   i. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   ii. In the Cluster Activities dialog box, specify Description and click **OK**.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the following error is returned:

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

1. Configure Ranger as a permission management method in JindoFS.

   i. On the **bigboot** tab for the SmartData service, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger and click OK.

   iii. Save the configurations.

      a. In the upper-right corner of the Service Configuration section, click **Save**.

      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

      c. Click **OK**.

   iv. Restart Namespace Service.

      a. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

      b. In the Cluster Activities dialog box, specify Description and click **OK**.

2. Add the HDFS service on the web UI of Ranger and configure the required parameters.

   i. Log on to the Ranger web UI.

      For more information, see Overview.

   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}*.
Authorization Enabled	Retain default value No.
Authentication Type	Retain default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you also need to enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information of user groups synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **bigboot** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters listed in the following table and click **OK**.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping

Parameter	Example
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

Note Configure the parameters based on the configurations in open source HDFS.

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Select **Restart All Components** from the **Actions** drop-down list in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

12.3. JindoFS in cache mode

12.3.1. Use JindoFS in cache mode

When you use JindoFS in cache mode, files are stored as objects in Object Storage Service (OSS), and the frequently used files are cached in an EMR cluster to improve the data access efficiency. In cache mode, JindoFS can access files in OSS without the need to convert the file formats, and JindoFS is fully compatible with OSS clients. This topic describes how to use JindoFS in cache mode.

Context

In cache mode, JindoFS supports the object semantics of OSS and is fully compatible with various OSS clients. This ensures that you can access files in OSS without the need to migrate data or convert data formats. In cache mode, JindoFS caches frequently used files in an EMR cluster. This improves the read and write performance and relieves pressure on bandwidth.

Methods to access files in OSS

In cache mode, you can use one of the following methods to access files in OSS based on your business requirements:

- OSS Scheme

  For more information, see (Recommended) Configure OSS Scheme.

- JFS Scheme

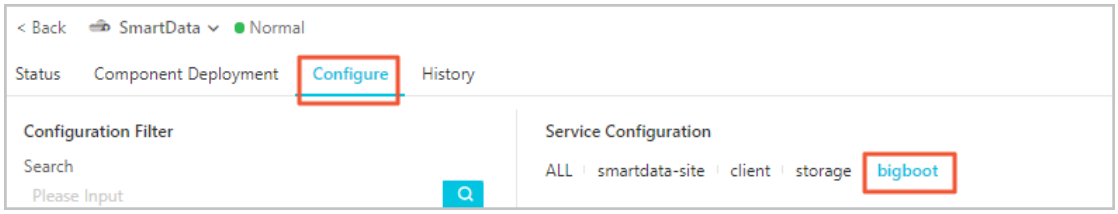
  For more information, see Configure JFS Scheme.

(Recommended) Configure OSS Scheme

OSS Scheme refers to the original method of accessing files in OSS. You can use the `oss://<bucket_name>/<path_of_your_file>` command to access files in OSS. After you create an EMR cluster, you can use this method to access files in OSS without additional configurations. You can also run existing jobs to read or write data from or to OSS without the need to modify the configurations.

Configure JFS Scheme

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Configure bigboot parameters.
   i. Click the **Configure** tab.
   ii. Click the **bigboot** tab in the Service Configuration section.



3. Configure the required parameters.

   JindoFS allows you to configure multiple namespaces. A namespace named test is used in this topic.

   i. Set **jfs.namespaces** to **test**.

      If you configure multiple namespaces, separate them with commas (,).

ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click OK.

Parameter	Description	Example
jfs.namespaces.test.oss.uri	The storage backend of the test namespace.	*oss://<oss_bucket>/<oss_dir>/* **Note** Set this parameter to a directory for a specific OSS bucket or the root directory.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

4. In the upper-right corner of the Service Configuration section, click **Save**.
5. Select **Restart Jindo Namespace Service** from the **Actions** drop down list in the upper-right corner.

   After the service is restarted, you can use the `jfs://test/<path_to_your_file>` command to access files in OSS. The files for the test namespace are organized based on the setting of the **jfs.namespaces.test.oss.uri** parameter. For example, `jfs://test/hello.txt` corresponds to *oss://<oss_bucket>/<oss_dir>/hello.txt*.

Enable local cache

After you enable local cache, hot data blocks are cached on local disks. By default, this feature is disabled, and EMR directly reads data from OSS.

1. In the left-side navigation pane, click **Cluster Service** and then **SmartData**. On the SMARTDATA page, click the **Configure** tab. In the Service Configuration section, click the **client** tab.
2. Set **jfs.cache.data-cache.enable** to **1** to enable local cache.

   The configuration immediately takes effect on the client, without the need to restart the SmartData service.

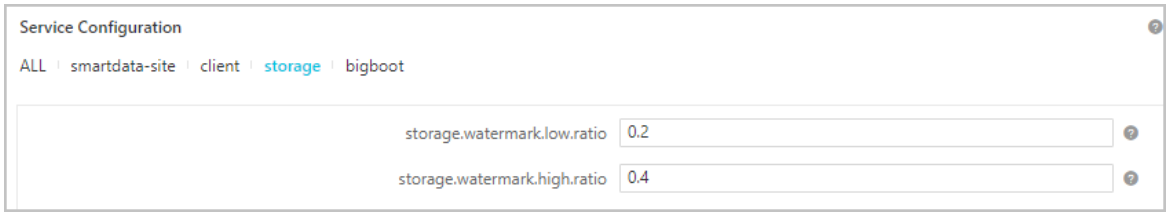
After you enable local cache, Jindo automatically manages cached data. It clears cache based on the high and low watermarks that you configured. For more information about how to configure the watermarks, see Control disk space usage.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. The `storage.watermark.high.ratio` and `storage.watermark.low.ratio` parameters are used to adjust the space usage of local disks. You can set the parameters to decimal numbers between 0 and 1.

1. Modify disk usage configurations.

In the **Service Configuration** section for the SmartData service, click the **storage** tab and configure the parameters described in the following table.



Parameter	Description
storage.watermark.high.ratio	The upper limit of disk usage. When the disk usage of JindoFS data exceeds this limit, JindoFS automatically deletes data in the disk. Default value: 0.4.
storage.watermark.low.ratio	The lower limit of disk usage. After automatic data deletion is triggered, JindoFS starts to delete data until the disk usage of JindoFS data is reduced to this limit. Default value: 0.2.

> **Note** You can configure the upper limit and lower limit to adjust the disk space assigned to JindoFS. Make sure that the upper limit is greater than the lower limit.

2. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
3. Restart Jindo Storage Service to apply the configurations.
   i. Select **Restart Jindo Storage Service** from the **Actions** drop-down list in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify related parameters.
   iii. Click **OK**.
   iv. In the **Confirm** message, click **OK**.

Access an OSS bucket

If you access an OSS bucket that is under the same Alibaba Cloud account and in the same region as your EMR cluster, you do not need to configure an AccessKey pair. In other cases, you must configure an AccessKey pair and an OSS bucket endpoint. Configure the parameters based on the method that you use to access files in OSS:

- OSS Scheme
  i. In the left-side navigation pane, click **Cluster Service** and then **SmartData**. On the SMARTDATA page, click the **Configure** tab. In the Service Configuration section, click the **smartdata-site** tab.
  ii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the

parameters described in the following table and click **OK**.

Parameter	Description
fs.jfs.cache.oss-accessKeyId	The AccessKey ID of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss-accessKeySecret	The AccessKey secret of the OSS bucket that serves as the storage backend.
fs.jfs.cache.oss-endpoint	The endpoint of the OSS bucket that serves as the storage backend.

- JFS Scheme
  i. In the left-side navigation pane, click **Cluster Service** and then **SmartData**. On the SMARTDATA page, click the **Configure** tab. In the Service Configuration section, click the **bigboot** tab.
  ii. Set **jfs.namespaces** to **test**.
  iii. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, configure the parameters described in the following table and click **OK**.

Parameter	Description
jfs.namespaces.test.oss.uri	The storage backend of the test namespace. Example: *oss://<oss_bucket.endpoint>/<oss_dir>*. The OSS bucket endpoint is specified in this parameter.
jfs.namespaces.test.oss.access.key	The AccessKey ID of the OSS bucket that serves as the storage backend.
jfs.namespaces.test.oss.access.secret	The AccessKey secret of the OSS bucket that serves as the storage backend.

Advanced configurations

You can configure some advanced parameters to optimize cache performance. After you configure the parameters, you do not need to restart the SmartData service. The configurations take effect immediately on the client.

- In the **Service Configuration** section, click the **client** tab and configure the parameters described in the following table.

Parameter	Description
client.oss.upload.threads	The number of OSS upload threads for each data write stream. Default value: 4.

Parameter	Description
client.oss.upload.max.parallelism	The maximum number of concurrent OSS upload threads of a process. This parameter prevents upload threads from occupying an excessive amount of bandwidth and memory. Default value: 16.

- In the **Service Configuration** section, click the **smartdata-site** tab and configure the parameters described in the following table.

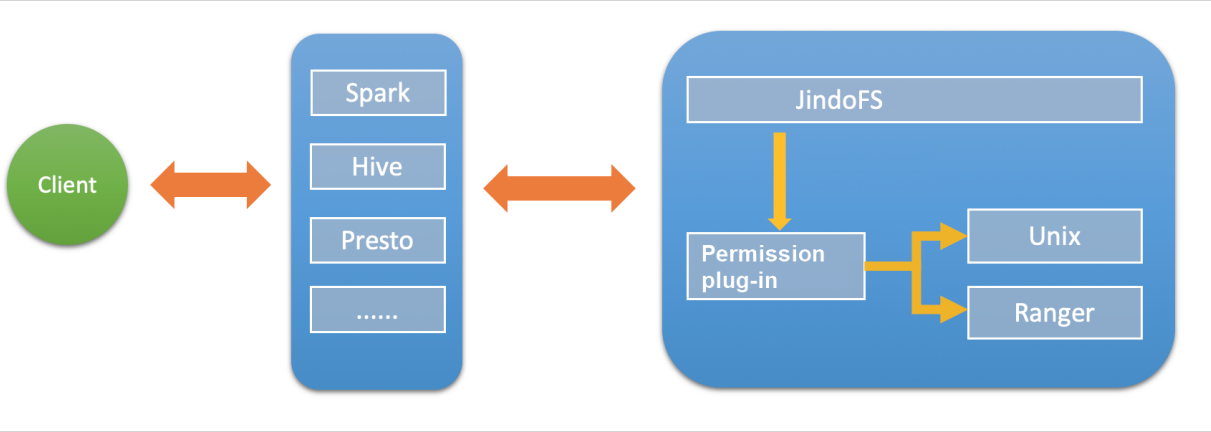
Parameter	Description
fs.jfs.cache.copy.simple.max.byte	The threshold for the size of a file that is renamed over a common copy interface. If the size of a file is smaller than this threshold, a common copy interface is used. If the size is larger than this threshold, the Multipart Copy interface is used to improve copy efficiency. **Note** If you have enabled the fast copy feature of OSS, set this parameter to **-1**. This value indicates that all files are renamed over a common copy interface. This way, you can obtain the optimal rename performance.
fs.jfs.cache.write.buffer.size	The buffer size of data write streams. Unit: bytes. You must set this parameter to a power of 2. The maximum value is 8388608 (8 MB). If too much memory is occupied by write streams, we recommend that you set this parameter to a small value. Default value: 1048576.
fs.oss.committer.magic.enabled	Specifies whether to enable Jindo Job Committer. This Job Committer does not require rename operations and improves job commit performance. Default value: true. **Note** In cache mode, the performance of renaming files in OSS is less than standard. We recommend that you use Jindo Job Committer.

12.3.2. Manage JindoFS permissions

This topic describes how to manage the permissions of JindoFS in block storage mode. You can run UNIX commands or use Ranger to manage permissions.

Context

To use Ranger to manage permissions, you must first configure permissions in the Apache Ranger component of E-MapReduce (EMR) and activate the Ranger plug-in in JindoFS. Then, you can manage JindoFS permissions in Ranger by using the same method as you manage permissions on other components.

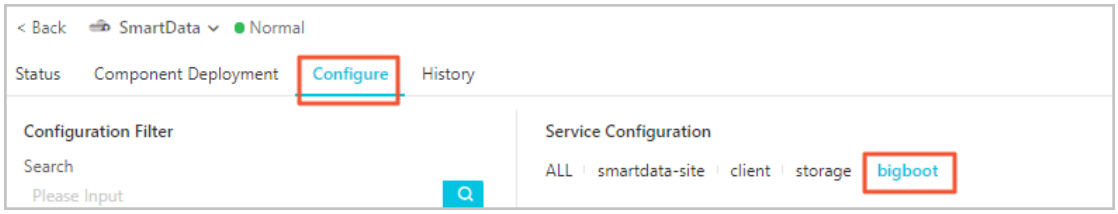


In block storage mode, you can run UNIX commands or use Ranger to manage permissions:

- UNIX allows you to grant the rwxrwxrwx permission on files.
- Ranger allows you to use wildcards in paths when you configure a permission.

Enable UNIX-based permission management

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Configure bigboot parameters.
   i. Click the **Configure** tab.
   ii. Click the **bigboot** tab in the Service Configuration section.



3. Click **Custom Configuration**. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to unix and click **OK**.
4. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

iii. Click **OK**.

5. Restart Namespace Service.

   i. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

   ii. In the Cluster Activities dialog box, specify Description and click **OK**.

After the service is restarted, you can run UNIX commands to manage JindoFS permissions by using the same method as you manage HDFS permissions. You can use the following commands:

```
hadoop fs -chmod 777 jfs://{namespace_name}/dir1/file1
hadoop fs -chown john:staff jfs://{namespace_name}/dir1/file1
```

If a user does not have permissions on a file, the following error is returned:

```
[root@emr-header-1 ~]# hadoop fs -ls jfs://test/user/
ls: java.security.AccessControlException: Permission denied. Server Exception
```

Enable Ranger-based permission management

1. Configure Ranger as a permission management method in JindoFS.

   i. On the **bigboot** tab for the SmartData service, click **Custom Configuration**.

   ii. In the **Add Configuration Item** dialog box, set **Key** to jfs.namespaces.<namespace>.permission.method and **Value** to ranger and click OK.

   iii. Save the configurations.

      a. In the upper-right corner of the Service Configuration section, click **Save**.

      b. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

      c. Click **OK**.

   iv. Restart Namespace Service.

      a. Select **Restart Jindo Namespace Service** from the **Actions** drop-down list in the upper-right corner.

      b. In the Cluster Activities dialog box, specify Description and click **OK**.

2. Add the HDFS service on the web UI of Ranger and configure the required parameters.

   i. Log on to the Ranger web UI.

   For more information, see Overview.

   ii. Add the HDFS service on the web UI of Ranger.

iii. Configure the parameters described in the following table.

Parameter	Description
Service Name	Set this parameter in the format of jfs-{namespace_name}.
Username	Customize a username.
Password	Customize a password.
Namenode URL	Set this parameter in the format of *jfs://{namespace_name}*.
Authorization Enabled	Retain default value No.
Authentication Type	Retain default value Simple.
dfs.datanode.kerberos.principal	Leave this parameter empty.
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Add New Configurations	

iv. Click **Add**.

Enable synchronization of user groups from an LDAP server in JindoFS

If you have enabled synchronization of user groups from an LDAP server in Ranger Usersync, you also need to enable this feature in JindoFS. Otherwise, JindoFS cannot obtain the information of user groups synchronized from the LDAP server and cannot verify the permissions of the user groups.

1. On the **bigboot** tab for the SmartData service, click **Custom Configuration**.
2. In the **Add Configuration Item** dialog box, configure the LDAP parameters listed in the following table and click **OK**.

Parameter	Example
hadoop.security.group.mapping	org.apache.hadoop.security.CompositeGroupsMapping
hadoop.security.group.mapping.providers	shell4services,ad4users
hadoop.security.group.mapping.providers.combined	true
hadoop.security.group.mapping.provider.shell4services	org.apache.hadoop.security.ShellBasedUnixGroupsMapping

Parameter	Example
hadoop.security.group.mapping.provider.ad4users	org.apache.hadoop.security.LdapGroupsMapping
hadoop.security.group.mapping.ldap.url	ldap://emr-header-1:10389
hadoop.security.group.mapping.ldap.search.filter.user	(&(objectClass=person)(uid={0}))
hadoop.security.group.mapping.ldap.search.filter.group	(objectClass=groupOfNames)
hadoop.security.group.mapping.ldap.base	o=emr

Note Configure the parameters based on the configurations in open source HDFS.

3. Save the configurations.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Restart all components of the SmartData service.
   i. Select **Restart All Components** from the **Actions** drop-down list in the upper-right corner.
   ii. In the Cluster Activities dialog box, specify Description and click **OK**.
5. Log on to the emr-header-1 node of the EMR cluster in SSH mode and connect Ranger Usersync to the LDAP server.

   For more information, see Integrate Ranger UserSync with an LDAP server.

12.3.3. Use Jindo Job Committer

This topic describes how to use Jindo Job Committer, which is also called JindoOssCommitter.

Context

Job Committer is a basic component of distributed computing frameworks such as MapReduce and Spark. It is used to handle data consistency issues of write operations for distributed tasks.

Jindo Job Committer is an effective Job Committer developed by the Alibaba Cloud E-MapReduce (EMR) team. It is dedicated for job commits in the OSS scenario. Jindo Job Committer is combined with the multipart upload and file system customization features of OSS. If Jindo Job Committer is used, the output data of tasks is directly written in the final output directory without the need to perform rename operations. Before the job commit is completed, intermediate data is invisible to users. This ensures data consistency.

Notice

- The OSS bandwidth and the enabling of some advanced features may affect the data copy performance of OSS. Therefore, the data copy performance for different users or buckets may vary. If you have any question, contact the technical support personnel of OSS.
- After all tasks are completed, MapReduce Application Master or Spark Driver commits the job. This triggers a short time window in which only a part of the result files appear in the final output directory. The length of the time window is positively correlated with the number of files. You can set the `fs.oss.committer.threads` parameter to a larger value to speed up concurrent processing.
- Hive and Presto jobs do not use Hadoop Job Committer.
- In EMR clusters, Jindo Job Committer is enabled by default.

Use Jindo Job Committer in MapReduce jobs

1. Go to the **mapred-site** tab for the YARN service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides and select a resource group based on your business requirements.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page, find your cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, choose **Cluster Service > YARN**.
   vi. Click the **Configure** tab.
   vii. In the **Service Configuration** section, click the **mapred-site** tab.
2. Configure the Job Committer parameter based on your Hadoop version:
   - Hadoop 2.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter.
   - Hadoop 3.X

     On the **mapred-site** tab, set the **mapreduce.outputcommitter.factory.scheme.oss** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitterFactory.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

> **Note** After you set the **mapreduce.outputcommitter.class** parameter to com.aliyun.emr.fs.oss.commit.JindoOssCommitter, you can use the **fs.oss.committer.magic.enabled** parameter to determine which Job Committer is used. If you set this parameter to true, MapReduce jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

Use Jindo Job Committer in Spark jobs

1. Go to the **spark-defaults** tab for the Spark service.
   i. In the left-side navigation pane, choose **Cluster Service > Spark**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **spark-defaults** tab.
2. On the **spark-defaults** tab, configure the parameters listed in the following table.

Parameter	Value
spark.sql.sources.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.parquet.output.committer.class	com.aliyun.emr.fs.oss.commit.JindoOssCommitter
spark.sql.hive.outputCommitterClass	com.aliyun.emr.fs.oss.commit.JindoOssCommitter

   These parameters specify the Job Committers used to write data to Spark data source tables, Spark Parquet data source tables, and Hive tables.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.
4. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
5. On the **smartdata-site** tab, set **fs.oss.committer.magic.enabled** to true.

> **Note** You can use the `fs.oss.committer.magic.enabled` parameter to determine which Job Committer is used. If you set this parameter to true, Spark jobs use Jindo Oss Magic Committer, which does not require rename operations. If you set this parameter to false, JindoOssCommitter functions the same as FileOutputCommitter.

6. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Optimize Jindo Job Committer

If your MapReduce or Spark tasks write a large number of files, you can adjust the number of threads that can be concurrently executed for job commit-related tasks to improve job commit performance.

1. Go to the **smartdata-site** tab for the SmartData service.
   i. In the left-side navigation pane, choose **Cluster Service > SmartData**.
   ii. Click the **Configure** tab.
   iii. In the **Service Configuration** section, click the **smartdata-site** tab.
2. On the **smartdata-site** tab, set the **fs.oss.committer.threads** parameter to 8.

   The default value is 8.
3. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

12.4. JindoTable

12.4.1. Use JindoCube

JindoCube is supported in E-MapReduce (EMR) V3.24.0 and later. This topic describes how to install, deploy, and use JindoCube in EMR.

Prerequisites

A table or view is created.

Overview

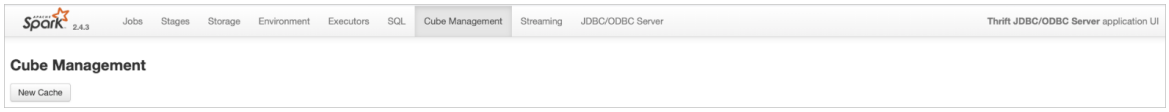
JindoCube is an advanced feature of EMR Spark. It implements pre-calculation to accelerate data processing. It improves query performance by dozens or even hundreds of times. You can persist data of a view to a data source supported by EMR Spark, such as HDFS or Object Storage Service (OSS). EMR Spark automatically discovers available persistent data and optimizes the execution plans for the data. This process is completely visible to users. JindoCube applies to business scenarios where a fixed query mode is used. JindoCube pre-calculates and pre-organizes data to accelerate data queries. For example, you can use JindoCube to implement multidimensional online analytical processing (MOLAP), generate data reports, create data dashboards, and synchronize data across clusters.

Install and deploy JindoCube

JindoCube is an advanced feature of EMR Spark. You can use JindoCube to accelerate all Dataset, DataFrame API, and SQL tasks that are submitted by using EMR Spark. No additional component deployment or maintenance is required.

1. Manage JindoCube on the Spark web UI.

   You can manage JindoCube on the Spark web UI. For example, you can create, delete, or update JindoCube caches. After you create a JindoCube cache, it automatically accelerates all Spark queries in the cluster. You can use the **spark.sql.cache.tab.display** parameter to specify whether to display the Cube Management tab on the Spark web UI. You can specify this parameter on the configuration page of the Spark service in the EMR console or in a Spark task submission command. The default value of this parameter is **false**.

   You can also use the **spark.sql.cache.useDatabase** parameter to create a database based on your business requirements and store the view for which you want to create a cache in this database. If partitioned tables are used, you can use the **spark.sql.cache.cacheByPartition** parameter to specify whether to cache data by partition.

Parameter	Description	Example
spark.sql.cache.tab.display	Specifies whether to display the Cube Management tab.	true
spark.sql.cache.useDatabase	The databases where cubes are stored.	db1,db2,dbn
spark.sql.cache.cacheByPartition	Specifies whether to store cubes by partition.	true

2. Optimize Spark queries.

   You can use the **spark.sql.cache.queryRewrite** parameter to specify whether to use a JindoCube cache to accelerate Spark queries. You can specify this parameter at the cluster, session, or SQL statement level. The default value is **true**.

Use JindoCube

1. Create a JindoCube cache.
   i. Log on to the Alibaba Cloud EMR console by using your Alibaba Cloud account.
   ii. Click the **Cluster Management** tab.

iii. Find the cluster where you want to create a JindoCube cache and click the cluster ID.

iv. In the left-side navigation pane, click **Connect Strings**.

v. On the **Public Connect Strings** page, click the link of YARN UI to go to the YARN web UI as a Knox user.

For more information about Knox, see Knox.

vi. Click **ApplicationMaster** in the row where **Application Type** is **SPARK** and **Name** is **Thrift JDBC/ODBC Server**.

vii. On the Spark web UI, click the **Cube Management** tab.

viii. Click **New Cache**.

Select a table or view, and click raw cache or cube cache in the action column. You can create two types of caches:

- Raw cache: persists the data of a table or view based on the cache configurations.

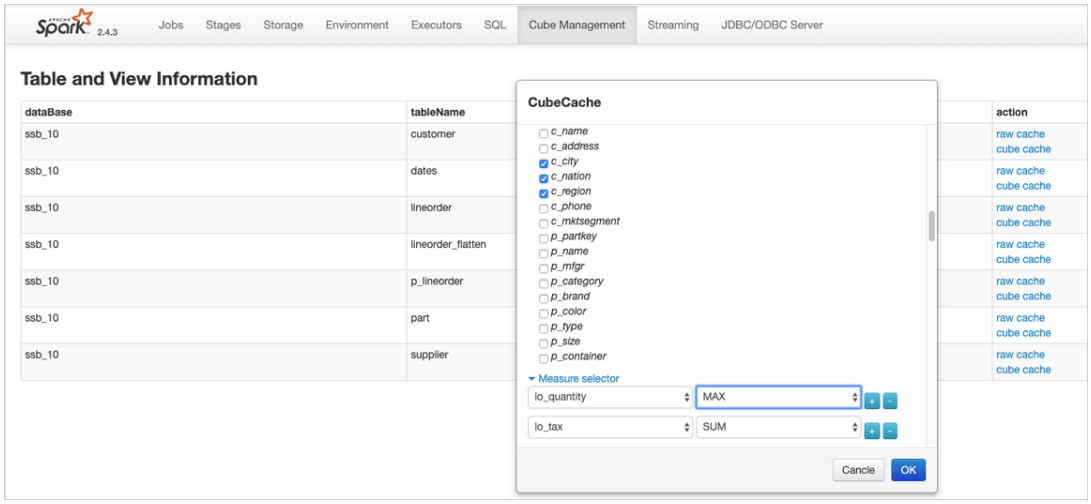
  When you create a raw cache, specify the parameters described in the following table.

Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Column Selector	The columns from which you want to cache data.	Yes
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No

- Cube cache: builds a cube based on the raw data of a table or view and persists the cube data.

When you create a cube cache, specify the parameters described in the following table.

Parameter	Description	Required
Cache Name	The name of the cache. The name can contain only letters, digits, hyphens (-), and underscores (_).	Yes
Dimension Selector	The dimension fields that are used to build a cube.	Yes
Measure Selector	The measure fields and pre-calculation functions that are used to build a cube.	Yes
Rewrite	Specifies whether to use the cache to optimize the execution plans of subsequent queries.	Yes
Provider	The storage format of the cached data. You can set this parameter to a format supported by Spark, such as JSON, Parquet, or ORC.	Yes
Partition Columns	The partition fields by which data is cached.	No
ZOrder Columns	The Z-order columns that are used to filter data. Z-order is a multi-column sorting method. After cached data is sorted by using this method, the data is filtered based on the Z-order columns during Spark queries. This further accelerates Spark queries.	No



JindoCube builds a cube based on specified dimensions and measures. The following SQL statement shows the configurations of the cube cache in the preceding figure:

```
SELECT c_city, c_nation, c_region, MAX(lo_quantity), SUM(lo_tax)
FROM lineorder_flatten
GROUP BY c_city, c_nation, c_region;
```

JindoCube calculates the finest-grained combination of dimensions. Powered by the onsite computing capability of Spark, JindoCube implements re-aggregation to optimize the queries where a coarse-grained combination of dimensions is used. You must use the pre-calculation functions supported by JindoCube to define a cube cache. The following table lists the supported pre-calculation functions and the aggregate functions that correspond to them.

Aggregate function	Pre-calculation function
COUNT	COUNT
SUM	SUM
MAX	MAX
MIN	MIN
AVG	COUNT and SUM
COUNT (DISTINCT)	PRE_COUNT_DISTINCT
APPROX_COUNT_DISTINCT	PRE_APPROX_COUNT_DISTINCT

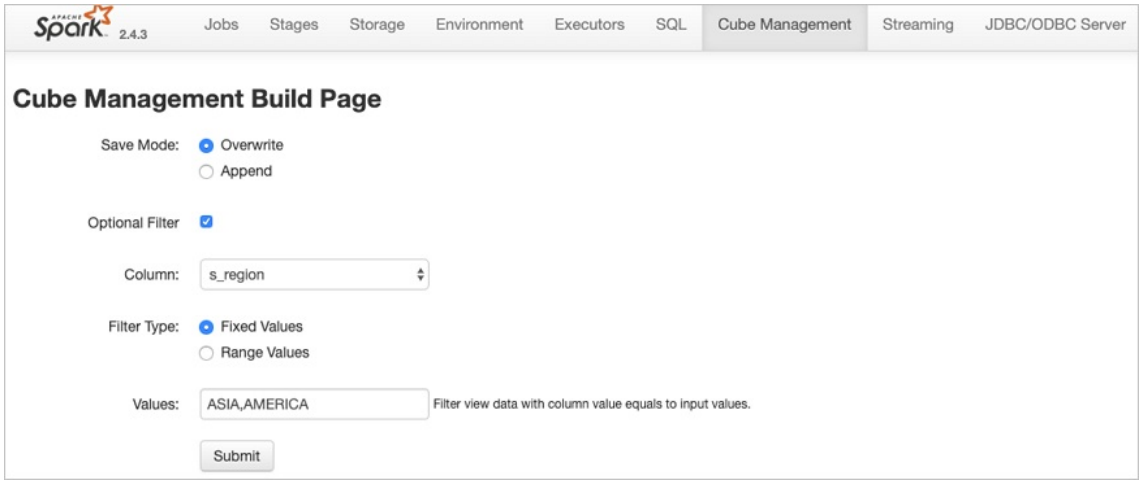
All created caches are displayed on the **Cube Management** tab. You can click **Detail** for a cache to view its details, including its basic information, partition information, building information, and building history.

2. Build a JindoCube cache.

   During the creation of a cache, metadata is prepared, but no data is persisted. Therefore, after you create a cache, you must continue to build the cache. This way, the related data can be persisted to permanent storage, such as HDFS or OSS. In addition, the source table of a cache may be appended with new data or updated. In this case, you must update the data in the cache to ensure data consistency. You can use one of the following methods to build a cache:

   - Manually build a cache

     Click Build Cache to manually build a cache. On the page that appears, configure the parameters, as shown in the following figure.



The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.

Optional Filter	You can select this parameter to add filter conditions. This way, JindoCube filters data based on the filter conditions before it persists the data.
	■ Column: the field based on which data is filtered.
	■ Filter Type: specifies whether to filter data by fixed values or by value range.
	  ■ Fixed Values: If this option is selected, specify one or more values. Separate values with commas (,).
	  ■ Range Values: If this option is selected, specify a minimum value and a maximum value. You can leave the maximum value empty. In this case, the value range is the minimum value and all values greater than it.

The following SQL statement shows the configurations of the raw cache for the lineorder_flatten view in the preceding figure:

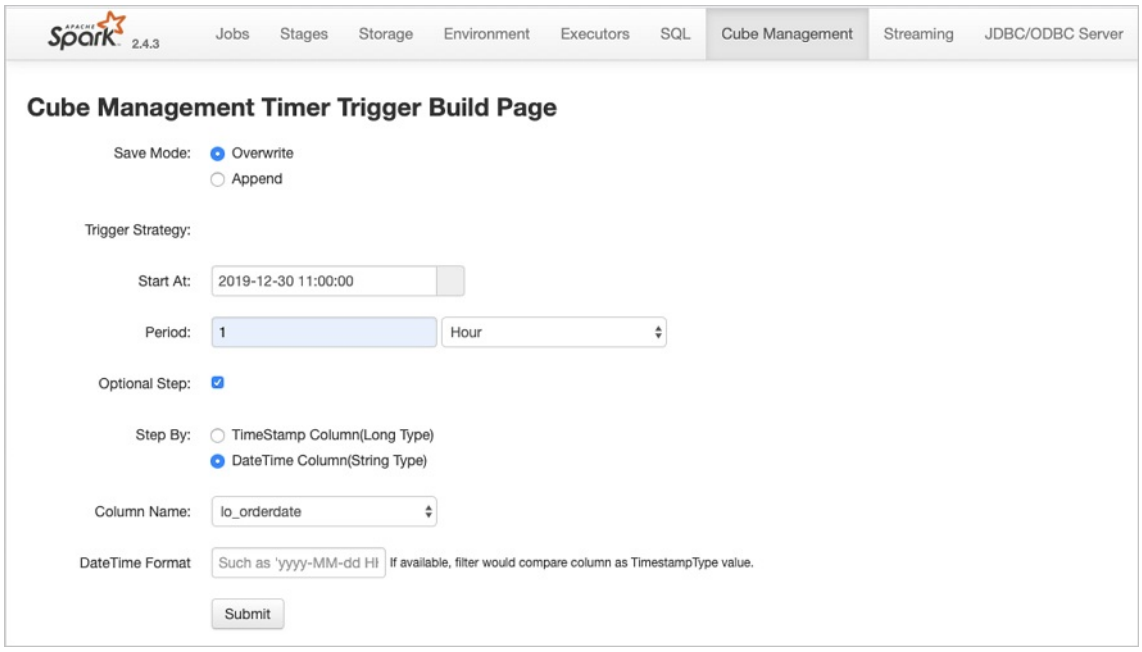
```
SELECT * FROM lineorder_flatten
WHERE s_region == 'ASIA' OR s_region == 'AMERICA';
```

Click **Submit** to submit the cache building task to a Spark cluster. You can view the task status in the Build Information section of the cache details page. After the task is complete, you can view the cache information in the Build History section.

> **Note** Spark writes the cached data to a specified directory in the same way common Spark tasks write data to a table or directory. If multiple users concurrently build a cache whose data format is Parquet, JSON, or ORC, the cached data may be inaccurate and invalid. If concurrent cache building operations or updates are inevitable, we recommend that you use a data format that supports concurrent writes, for example, Delta.

- Configure periodic cache building

  Click Trigger Period Build to configure a periodic cache update policy to ensure data consistency between the cache and source table. On the page that appears, configure the parameters, as shown in the following figure.



The following table describes the parameters.

Parameter	Description
Save Mode	The following modes are supported: ■ Overwrite: overwrites exiting data in the cache. ■ Append: adds new data to the cache.

Parameter	Description
Trigger Strategy	Specifies the start time of the first building task, and the building interval. ■ Start At: Select or enter the time you want to trigger the first building task. Format: yyyy-MM-dd hh:mm:ss. ■ Period: Specify the interval at which cache building tasks are triggered.
Optional Step	Specifies the filter conditions for cache building. JindoCube updates the cache in incremental mode based on the filter conditions and interval. If you do not select Optional Step, JindoCube updates the entire cache each time. ■ Step By: Specify the type of the field that you want to update in incremental mode. TimeStamp Column(Long Type) indicates a timestamp field of the LONG type. DateTime Column(String Type) indicates a datetime field of the STRING type. ■ Column Name: Specify the name of the field that you want to update in incremental mode.

On the cache details page, you can view the configured cache update policy. You can click Cancel Period Build to cancel periodic updates. You can view the information of a complete building task in the Build History section.

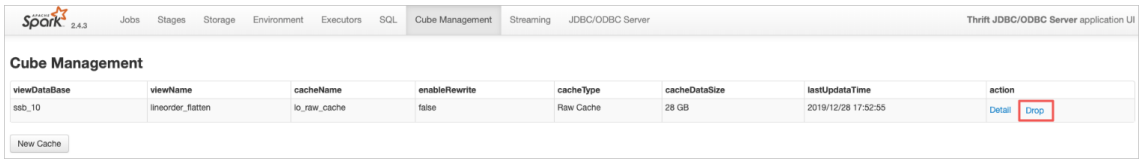
> **Note**
>
> - Periodic building tasks run on a Spark cluster. The related configurations are all saved in SparkContext, and the tasks are periodically triggered by Spark Driver. If you release the Spark cluster, periodic building is disabled.
> - The Build History section displays the following information of each complete task in the current Spark cluster: start time, end time, save mode, building conditions, and final status. After you release the Spark cluster, the information in the Build History section is cleared.

3. Manage a JindoCube cache.

   After you create and build a JindoCube cache, you can manage it on the Cube Management page.

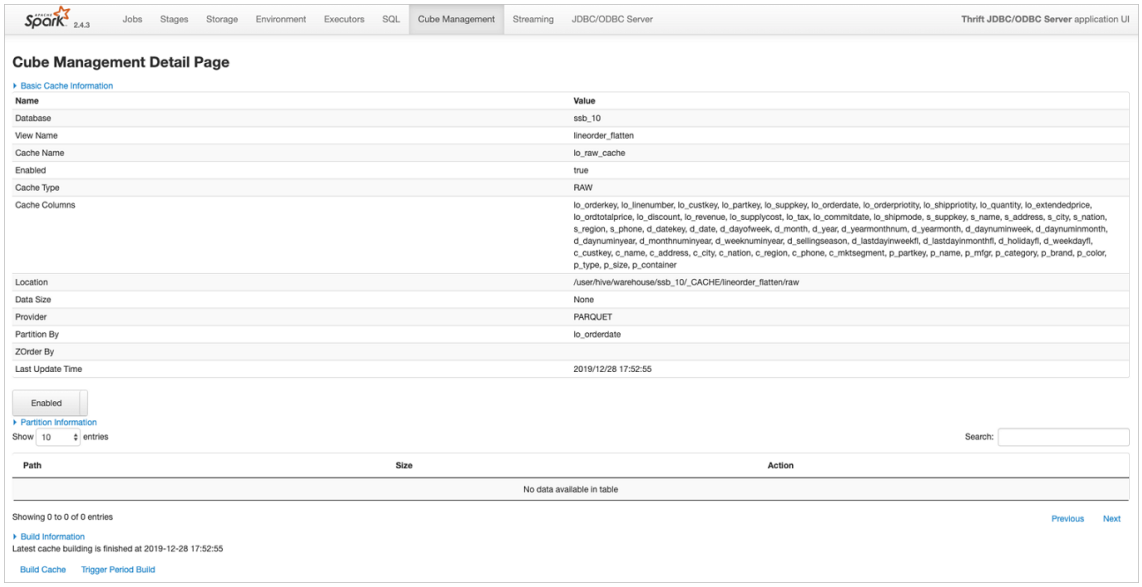
   - Delete a cache

     On the Cube Management page, find the cache you want to delete, and click **Drop** in the action column. All the metadata and data of the cache are deleted.



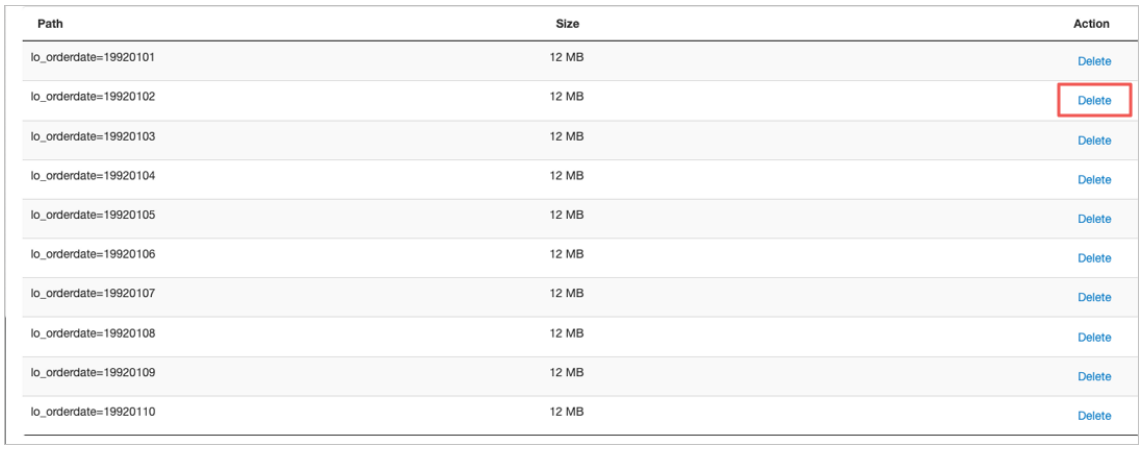
- Enable and disable optimization of Spark queries

  JindoCube allows you to specify whether to optimize Spark queries at the cache level. On the details page of a cache, click **Enabled** or **Disabled** in the lower part of the Basic Cache Information section to enable or disable the optimization.



- Delete partition data from a cache

  If data in a cache is stored by partition, you can delete the unnecessary partition data from the cache to release more space. On the details page of the cache, find the partition whose data you want to delete, and click **Delete** in the Action column.



Path	Size	Action
lo_orderdate=19920101	12 MB	Delete
lo_orderdate=19920102	12 MB	Delete
lo_orderdate=19920103	12 MB	Delete
lo_orderdate=19920104	12 MB	Delete
lo_orderdate=19920105	12 MB	Delete
lo_orderdate=19920106	12 MB	Delete
lo_orderdate=19920107	12 MB	Delete
lo_orderdate=19920108	12 MB	Delete
lo_orderdate=19920109	12 MB	Delete
lo_orderdate=19920110	12 MB	Delete

> **Note** Before you delete the data of a partition, make sure that the data is no longer needed. If you delete data that is required in subsequent queries, the queries return invalid results.

4. Optimize Spark queries.

JindoCube optimizes view-based queries. After you have created a raw cache or cube cache for a view, EMR Spark overwrites the execution plan of queries on this view. The new execution plan still uses the original logical semantics but can directly access the data in the cache. This accelerates Spark queries. Use the lineorder_flatten view as an example. It is a wide table view generated by associating the lineorder table with other dimension tables. The following figure shows the information about this view.

```
0: jdbc:hive2://localhost:10001/ssb_10> desc extended lineorder_flatten;
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:08:16 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+-------------------------------+----------------------------------------------------+----------+
|           col_name            |                     data_type                      | comment  |
+-------------------------------+----------------------------------------------------+----------+
| lo_orderkey                   | bigint                                             | NULL     |
| lo_linenumber                 | bigint                                             | NULL     |
| lo_custkey                    | int                                                | NULL     |
| lo_partkey                    | int                                                | NULL     |
| lo_suppkey                    | int                                                | NULL     |
| lo_orderdate                  | int                                                | NULL     |
| lo_orderpriotity              | string                                             | NULL     |
| lo_shippriotity               | int                                                | NULL     |
| lo_quantity                   | bigint                                             | NULL     |
| lo_extendedprice              | bigint                                             | NULL     |
| lo_ordtotalprice              | bigint                                             | NULL     |
| lo_discount                   | bigint                                             | NULL     |
| lo_revenue                    | bigint                                             | NULL     |
| lo_supplycost                 | bigint                                             | NULL     |
| lo_tax                        | bigint                                             | NULL     |
| lo_commitdate                 | int                                                | NULL     |
| lo_shipmode                   | string                                             | NULL     |
| s_suppkey                     | int                                                | NULL     |
| s_name                        | string                                             | NULL     |
| s_address                     | string                                             | NULL     |
| s_city                        | string                                             | NULL     |
| s_nation                      | string                                             | NULL     |
| s_region                      | string                                             | NULL     |
| s_phone                       | string                                             | NULL     |
| d_datekey                     | int                                                | NULL     |
| d_date                        | string                                             | NULL     |
| d_dayofweek                   | string                                             | NULL     |
| d_month                       | string                                             | NULL     |
| d_year                        | int                                                | NULL     |
| d_yearmonthnum                | int                                                | NULL     |
| d_yearmonth                   | string                                             | NULL     |
| d_daynuminweek                | int                                                | NULL     |
| d_daynuminmonth               | int                                                | NULL     |
| d_daynuminyear                | int                                                | NULL     |
| d_monthnuminyear              | int                                                | NULL     |
| d_weeknuminyear               | int                                                | NULL     |
| d_sellingseason               | string                                             | NULL     |
| d_lastdayinweekfl             | int                                                | NULL     |
| d_lastdayinmonthfl            | int                                                | NULL     |
| d_holidayfl                   | int                                                | NULL     |
| d_weekdayfl                   | int                                                | NULL     |
| c_custkey                     | int                                                | NULL     |
| c_name                        | string                                             | NULL     |
| c_address                     | string                                             | NULL     |
| c_city                        | string                                             | NULL     |
| c_nation                      | string                                             | NULL     |
| c_region                      | string                                             | NULL     |
| c_phone                       | string                                             | NULL     |
| c_mktsegment                  | string                                             | NULL     |
| p_partkey                     | int                                                | NULL     |
| p_name                        | string                                             | NULL     |
| p_mfgr                        | string                                             | NULL     |
| p_category                    | string                                             | NULL     |
| p_brand                       | string                                             | NULL     |
| p_color                       | string                                             | NULL     |
| p_type                        | string                                             | NULL     |
| p_size                        | int                                                | NULL     |
| p_container                   | string                                             | NULL     |
|                               |                                                    |          |
| # Detailed Table Information  |                                                    |          |
| Database                      | ssb_10                                             |          |
| Table                         | lineorder_flatten                                  |          |
| Owner                         | hadoop                                             |          |
| Created Time                  | Sat Dec 28 17:30:02 CST 2019                       |          |
| Last Access                   | Thu Jan 01 08:00:00 CST 1970                       |          |
| Created By                    | Spark 2.2 or prior                                 |          |
| Type                          | VIEW                                               |          |
| View Text                     | SELECT `lineorder`.`lo_orderkey`, `lineorder`.`lo_linenumber`, `lineorder`.`lo_custkey`, `lineorder`.`lo_partkey`, `
lineorder`.`lo_suppkey`, `lineorder`.`lo_orderdate`, `lineorder`.`lo_orderpriotity`, `lineorder`.`lo_shippriotity`, `lineorder`.`lo_quantity`, `lineor
der`.`lo_extendedprice`, `lineorder`.`lo_ordtotalprice`, `lineorder`.`lo_discount`, `lineorder`.`lo_revenue`, `lineorder`.`lo_supplycost`, `lineorder`
.`lo_tax`, `lineorder`.`lo_commitdate`, `lineorder`.`lo_shipmode`, `supplier`.`s_suppkey`, `supplier`.`s_name`, `supplier`.`s_address`, `supplier`.`s_
city`, `supplier`.`s_nation`, `supplier`.`s_region`, `supplier`.`s_phone`, `dates`.`d_datekey`, `dates`.`d_date`, `dates`.`d_dayofweek`, `dates`.`d_mo
nth`, `dates`.`d_year`, `dates`.`d_yearmonthnum`, `dates`.`d_yearmonth`, `dates`.`d_daynuminweek`, `dates`.`d_daynuminmonth`, `dates`.`d_daynuminyear`
, `dates`.`d_monthnuminyear`, `dates`.`d_weeknuminyear`, `dates`.`d_sellingseason`, `dates`.`d_lastdayinweekfl`, `dates`.`d_lastdayinmonthfl`, `dates`
.`d_holidayfl`, `dates`.`d_weekdayfl`, `customer`.`c_custkey`, `customer`.`c_name`, `customer`.`c_address`, `customer`.`c_city`, `customer`.`c_nation`
, `customer`.`c_region`, `customer`.`c_phone`, `customer`.`c_mktsegment`, `part`.`p_partkey`, `part`.`p_name`, `part`.`p_mfgr`, `part`.`p_category`, `
part`.`p_brand`, `part`.`p_color`, `part`.`p_type`, `part`.`p_size`, `part`.`p_container` FROM `ssb_10`.`lineorder`, `ssb_10`.`supplier`, `ssb_10`.`da
tes`, `ssb_10`.`customer`, `ssb_10`.`part`
WHERE `lineorder`.`lo_orderdate` = `dates`.`d_datekey` AND `lineorder`.`lo_custkey` = `customer`.`c_custkey` AND `lineorder`.`lo_suppkey` = `supplier`
.`s_suppkey` AND `lineorder`.`lo_partkey` = `part`.`p_partkey` |          |
| Table Properties              | [transient_lastDdlTime=1577677599]                 |          |
| Serde Library                 | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe |          |
| InputFormat                   | org.apache.hadoop.mapred.TextInputFormat           |          |
| OutputFormat                  | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat |          |
| Storage Properties            | [serialization.format=1]                           |          |
+-------------------------------+----------------------------------------------------+----------+
73 rows selected (0.057 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

The following figure shows the execution plan of simple queries on the lineorder_flatten view.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:19:32 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(11) SortMergeJoin [lo_partkey#313], [p_partkey#359], Inner
   :- *(8) Sort [lo_partkey#313 ASC NULLS FIRST], false, 0
   :  +- Exchange hashpartitioning(lo_partkey#313, 200)
   :     +- *(7) SortMergeJoin [lo_custkey#312], [c_custkey#351], Inner
   :        :- *(4) Sort [lo_custkey#312 ASC NULLS FIRST], false, 0
   :        :  +- Exchange hashpartitioning(lo_custkey#312, 200)
   :        :     +- *(3) BroadcastHashJoin [lo_orderdate#315], [d_datekey#334], Inner, BuildRight
   :        :        :- *(3) BroadcastHashJoin [lo_suppkey#314], [s_suppkey#327], Inner, BuildRight
   :        :        :  :- *(3) Filter (((isnotnull(lo_suppkey#314) && isnotnull(lo_orderdate#315)) && isnotnull(lo_custkey#312)) && isnotnull(lo_part
key#313))
   :        :        :  :  +- Scan hive ssb_10.lineorder [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_ord
erdate#315, lo_orderpriotity#316, lo_shippriotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#32
2L, lo_supplycost#323L, lo_tax#324L, lo_commitdate#325, lo_shipmode#326], HiveTableRelation `ssb_10`.`lineorder`, org.apache.hadoop.hive.serde2.lazy.L
azySimpleSerDe, [lo_orderkey#310L, lo_linenumber#311L, lo_custkey#312, lo_partkey#313, lo_suppkey#314, lo_orderdate#315, lo_orderpriotity#316, lo_ship
priotity#317, lo_quantity#318L, lo_extendedprice#319L, lo_ordtotalprice#320L, lo_discount#321L, lo_revenue#322L, lo_supplycost#323L, lo_tax#324L, lo_c
ommitdate#325, lo_shipmode#326]
   :        :        :  +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :        :     +- *(1) Filter isnotnull(s_suppkey#327)
   :        :        :        +- Scan hive ssb_10.supplier [s_suppkey#327, s_name#328, s_address#329, s_city#330, s_nation#331, s_region#332, s_phone#
333], HiveTableRelation `ssb_10`.`supplier`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [s_suppkey#327, s_name#328, s_address#329, s_city#330
, s_nation#331, s_region#332, s_phone#333]
   :        :        +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)))
   :        :           +- *(2) Filter isnotnull(d_datekey#334)
   :        :              +- Scan hive ssb_10.dates [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearm
onth#340, d_daynuminweek#341, d_daynuminmonth#342, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweek
fl#347, d_lastdayinmonthfl#348, d_holidayfl#349, d_weekdayfl#350], HiveTableRelation `ssb_10`.`dates`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSe
rDe, [d_datekey#334, d_date#335, d_dayofweek#336, d_month#337, d_year#338, d_yearmonthnum#339, d_yearmonth#340, d_daynuminweek#341, d_daynuminmonth#34
2, d_daynuminyear#343, d_monthnuminyear#344, d_weeknuminyear#345, d_sellingseason#346, d_lastdayinweekfl#347, d_lastdayinmonthfl#348, d_holidayfl#349,
 d_weekdayfl#350]
   :        +- *(6) Sort [c_custkey#351 ASC NULLS FIRST], false, 0
   :           +- Exchange hashpartitioning(c_custkey#351, 200)
   :              +- *(5) Filter ((isnotnull(c_region#356) && (c_region#356 = ASIA)) && isnotnull(c_custkey#351))
   :                 +- Scan hive ssb_10.customer [c_custkey#351, c_name#352, c_address#353, c_city#354, c_nation#355, c_region#356, c_phone#357, c_mk
tsegment#358], HiveTableRelation `ssb_10`.`customer`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [c_custkey#351, c_name#352, c_address#353, c
_city#354, c_nation#355, c_region#356, c_phone#357, c_mktsegment#358]
   +- *(10) Sort [p_partkey#359 ASC NULLS FIRST], false, 0
      +- Exchange hashpartitioning(p_partkey#359, 200)
         +- *(9) Filter isnotnull(p_partkey#359)
            +- Scan hive ssb_10.part [p_partkey#359, p_name#360, p_mfgr#361, p_category#362, p_brand#363, p_color#364, p_type#365, p_size#366, p_conta
iner#367], HiveTableRelation `ssb_10`.`part`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [p_partkey#359, p_name#360, p_mfgr#361, p_category#3
62, p_brand#363, p_color#364, p_type#365, p_size#366, p_container#367] |
+----------------------------------------------------+
1 row selected (0.435 seconds)
0: jdbc:hive2://localhost:10001/ssb_10>
```

After a raw cache is created and built for the lineorder_flatten view, the same queries are performed on the view. EMR Spark automatically uses the cache to optimize the original execution plan. The following figure shows the new execution plan.

```
0: jdbc:hive2://localhost:10001/ssb_10> explain select * from lineorder_flatten where c_region = 'ASIA' limit 10;
19/12/30 14:17:47 INFO [main] HiveConf: Found configuration file file:/etc/ecm/spark-conf-2.4.3-hadoop2.8-1.4.2/hive-site.xml
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.enforce.bucketing does not exist
19/12/30 14:17:47 WARN [main] HiveConf: HiveConf of name hive.support.sql11.reserved.keywords does not exist
+----------------------------------------------------+
|                        plan                        |
+----------------------------------------------------+
| == Physical Plan ==
CollectLimit 10
+- *(1) Project [lo_orderkey#128L, lo_linenumber#129L, lo_custkey#130, lo_partkey#131, lo_suppkey#132, lo_orderdate#133, lo_orderpriotity#134, lo_ship
priotity#135, lo_quantity#136L, lo_extendedprice#137L, lo_ordtotalprice#138L, lo_discount#139L, lo_revenue#140L, lo_supplycost#141L, lo_tax#142L, lo_c
ommitdate#143, lo_shipmode#144, s_suppkey#145, s_name#146, s_address#147, s_city#148, s_nation#149, s_region#150, s_phone#151, ... 34 more fields]
   +- *(1) Filter (isnotnull(c_region#174) && (c_region#174 = ASIA))
      +- *(1) FileScan parquet [lo_orderkey#128L,lo_linenumber#129L,lo_custkey#130,lo_partkey#131,lo_suppkey#132,lo_orderpriotity#134,lo_shippriotity#
135,lo_quantity#136L,lo_extendedprice#137L,lo_ordtotalprice#138L,lo_discount#139L,lo_revenue#140L,lo_supplycost#141L,lo_tax#142L,lo_commitdate#143,lo_
shipmode#144,s_suppkey#145,s_name#146,s_address#147,s_city#148,s_nation#149,s_region#150,s_phone#151,d_datekey#152,... 34 more fields] Batched: true,
Format: Parquet, Location: InMemoryFileIndex[/user/hive/warehouse/ssb_10/_CACHE/lineorder_flatten/raw], PartitionCount: 2406, PartitionFilters: [], Pu
shedFilters: [IsNotNull(c_region), EqualTo(c_region,ASIA)], ReadSchema: struct<lo_orderkey:bigint,lo_linenumber:bigint,lo_custkey:int,lo_partkey:int,l
o_suppkey:int,lo_or... |
+----------------------------------------------------+
```

The new execution plan no longer uses the calculation logic of the lineorder_flatten view. Instead, it directly accesses the cached data in HDFS.

Usage notes

1. JindoCube does not ensure data consistency between the cache and source table. You must manually trigger data updates or configure a periodic update policy based on your requirements for data consistency.
2. JindoCube determines whether to use a cache to optimize execution plans based on the metadata of views. If cached data is incomplete when JindoCube optimizes an execution plan, the query results may be invalid or incomplete. The following operations or configurations may cause incomplete cached data: 1. Delete partition data that is required in subsequent queries on the cache details page. 2. Data required in subsequent queries is filtered out based on the filter conditions specified when you build or update the cache. 3. Required data has not been updated to the cache when the query is performed.

12.5. Toolset

12.5.1. Use Jindo DistCp

This topic describes how to use the data copy tool Jindo DistCp.

Prerequisites

An E-MapReduce (EMR) cluster of the 3.28.0 version or later is created. For information about how to create a cluster, see Create a cluster.

Use Jindo DistCp

1. Log on to the master node of the EMR cluster in SSH mode.

   For more information, see Log on to a cluster.

2. Run the following command to obtain the help information:

   ```
   jindo distcp --help
   ```

   The following information is returned:

   ```
       --help   -   Print help text
       --src=VALUE   -   Directory to copy files from
       --dest=VALUE   -   Directory to copy files to
       --parallelism=VALUE   -   Copy task parallelism
       --outputManifest=VALUE   -   The name of the manifest file
       --previousManifest=VALUE   -   The path to an existing manifest file
       --requirePreviousManifest=VALUE   -   Require that a previous manifest is present
   if specified
       --copyFromManifest   -   Copy from a manifest instead of listing a directory
       --srcPrefixesFile=VALUE   -   File containing a list of source URI prefixes
       --srcPattern=VALUE   -   Include only source files matching this pattern
       --deleteOnSuccess   -   Delete input files after a successful copy
       --outputCodec=VALUE   -   Compression codec for output files
       --groupBy=VALUE   -   Pattern to group input files by
       --targetSize=VALUE   -   Target size for output files
       --enableBalancePlan   -   Enable plan copy task to make balance
       --enableDynamicPlan   -   Enable plan copy task dynamically
       --enableTransaction   -   Enable transation on Job explicitly
       --diff   -   show the difference between src and dest filelist
   ```

When you use Jindo DistCp, you can specify the following options to copy data:

- --src and --dest
- --parallelism
- --srcPattern
- --deleteOnSuccess
- --outputCodec
- --outputManifest and --requirePreviousManifest
- --outputManifest and --previousManifest
- --copyFromManifest

- --srcPrefixesFile
- --groupBy and -targetSize
- --enableBalancePlan
- --enableDynamicPlan
- --enableTransaction
- --diff
- Check DistCp Counters
- Use an AccessKey pair to access OSS
- Write data to OSS IA or Archive storage
- Clean up residual files

--src and --dest

`--src` specifies the source directory. `--dest` specifies the destination directory.

By default, Jindo DistCp copies all files in the directory specified by `--src` to the directory specified by `--dest`. If you do not specify a root directory, Jindo DistCp automatically creates a root directory.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp
```

> **Note** yourBucketName is the name of your OSS bucket.

--parallelism

`--parallelism` specifies the mapreduce.job.reduces parameter for the MapReduce job that is run to copy files. The default value is 7. You can customize `--parallelism` based on your available cluster resources. This allows you to determine how many reduce tasks can be run in parallel.

For example, you can run the following command to copy files in the */opt/tmp* directory of HDFS to an OSS bucket:

```
jindo distcp --src /opt/tmp --dest oss://<yourBucketName>/tmp --parallelism 20
```

--srcPattern

`--srcPattern` specifies a regular expression that filters files for the copy operation. The regular expression must match a full path.

For example, if you need to copy all log files in the */data/incoming/hourly_table/2017-02-01/03* directory, set `--srcPattern` to .*\.log.

Run the following command to view the files in the */data/incoming/hourly_table/2017-02-01/03* directory:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 6 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to copy the log files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPattern .*\.log --parallelism 20
```

Run the following command to view files in the destination OSS bucket:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Only log files in the source directory are copied.

```
Found 2 items
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log
-rw-rw-rw-   1       4891 2020-04-17 20:52 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log
```

--deleteOnSuccess

`--deleteOnSuccess` enables Jindo DistCp to delete the copied files from the source directory after a copy operation succeeds.

For example, you can run the following command to copy files in */data/incoming/hourly_table* to an OSS bucket and delete the copied files from the source directory:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-deleteOnSuccess --parallelism 20
```

--outputCodec

`--outputCodec` specifies the compression codec that is used to compress copied files online. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputCodec=gz --parallelism 20
```

Run the following command to view files in the destination directory:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03
```

The following information is returned. Files in the destination directory are compressed in the GZ format.

```
Found 6 items
-rw-rw-rw-   1        938 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/000151.sst.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/1.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/2.log.gz
-rw-rw-rw-   1       1956 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/OPTIONS-000109.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp01.txt.gz
-rw-rw-rw-   1        506 2020-04-17 20:58 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp06.txt.gz
```

You can set this parameter to gzip, gz, lzo, lzop, snappy, none, or keep. Default value: keep. Descriptions of none and keep:

- none: Jindo DistCp does not compress copied files. If the files have been compressed, Jindo DistCp decompresses them.
- keep: Jindo DistCp copies files with no change in the compression.

> **Note** If you want to use the LZO codec in an open source Hadoop cluster, you must install the native library of gplcompression and the hadoop-lzo package.

--outputManifest and --requirePreviousManifest

`--outputManifest` generates a manifest file that contains the information about all the files copied by Jindo DistCp. The information includes the destination files, source files, and file sizes.

If you want to generate a manifest file, set `--requirePreviousManifest` to `false`. By default, the file is compressed in the GZ format. This is the only supported format.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-outputManifest=manifest-2020-04-17.gz --requirePreviousManifest=false --parallelism 20
```

Run the following command to view the content of the file:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz > before.lst
cat before.lst
```

The following information is returned:

```
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/000151.sst","baseName":"2017-02-01/03/000151.sst","srcDir":"oss://<yourBucketName>/hourly_table","size":2252}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/1.log","baseName":"2017-02-01/03/1.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/2.log","baseName":"2017-02-01/03/2.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/OPTIONS-000109","baseName":"2017-02-01/03/OPTIONS-000109","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp01.txt","baseName":"2017-02-01/03/emp01.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
{"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/emp06.txt","baseName":"2017-02-01/03/emp06.txt","srcDir":"oss://<yourBucketName>/hourly_table","size":1016}
```

--outputManifest and --previousManifest

`--outputManifest` generates a manifest file that contains a list of both previously and newly copied files. `--previousManifest` generates a manifest file that contains a list of previously copied files. This way, you can recreate the full history of operations and see what files are copied by the current job.

For example, two files are added to the source directory. Run the following command to copy the newly added files:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table --outputManifest=manifest-2020-04-18.gz --previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --parallelism 20
```

Run the following command to view the copied files:

```
hadoop fs -text oss://<yourBucketName>/hourly_table/manifest-2020-04-18.gz > current.lst
diff before.lst current.lst
```

The following information is returned:

```
3a4,5
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/5.log","baseName":"2017-02-01/03/5.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
> {"path":"oss://<yourBucketName>/hourly_table/2017-02-01/03/6.log","baseName":"2017-02-01/03/6.log","srcDir":"oss://<yourBucketName>/hourly_table","size":4891}
```

--copyFromManifest

You can use `--copyFromManifest` to specify a manifest file that was previously generated by `--outputManifest` and copy the files listed in the manifest file to the destination directory. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table --previousManifest=oss://<yourBucketName>/hourly_table/manifest-2020-04-17.gz --copyFromManifest --parallelism 20
```

--srcPrefixesFile

`--srcPrefixesFile` enables Jindo DistCp to copy files in multiple folders at a time.

Run the following command to view the sub-folders under *hourly_table*:

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table
```

The following information is returned:

```
Found 4 items
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-01
drwxrwxrwx   -          0 1970-01-01 08:00 oss://<yourBucketName>/hourly_table/2017-02-02
```

Run the following command to copy all the files under *hourly_table* to the destination folder. A file named *folders.txt* is generated. It contains prefixes that correspond to each of the sub-folders under hourly_table.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-srcPrefixesFile file:///opt/folders.txt --parallelism 20
```

Run the following command to view the content of the *folders.txt* file:

```
cat folders.txt
```

The following information is returned:

```
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-01
hdfs://emr-header-1.cluster-50466:9000/data/incoming/hourly_table/2017-02-02
```

--groupBy and -targetSize

Reading a large number of small files from HDFS affects the data processing performance. Therefore, we recommend that you use Jindo DistCp to merge small files into large files of a specified size. This optimizes analysis performance and reduces costs.

Run the following command to view the files in the specified folder:

```
hdfs dfs -ls /data/incoming/hourly_table/2017-02-01/03
```

The following information is returned:

```
Found 8 items
-rw-r-----   2 root hadoop       2252 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/000151.sst
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/1.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/2.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/5.log
-rw-r-----   2 root hadoop       4891 2020-04-17 21:08 /data/incoming/hourly_table/2017-02-
01/03/6.log
-rw-r-----   2 root hadoop       4891 2020-04-17 20:42 /data/incoming/hourly_table/2017-02-
01/03/OPTIONS-000109
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp01.txt
-rw-r-----   2 root hadoop       1016 2020-04-17 20:47 /data/incoming/hourly_table/2017-02-
01/03/emp06.txt
```

Run the following command to merge the TXT files in the folder into files each with a size of no more than 10 MB:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-targetSize=10 --groupBy='.*/([a-z]+).*.txt' --parallelism 20
```

Run the following command to view the files in the destination directory. The two TXT files are merged into one.

```
hdfs dfs -ls oss://<yourBucketName>/hourly_table/2017-02-01/03/
Found 1 items
-rw-rw-rw-   1       2032 2020-04-17 21:18 oss://<yourBucketName>/hourly_table/2017-02-01/0
3/emp2
```

--enableBalancePlan

If both small and large files are to be copied but the file sizes are not significantly different among small files and among large files, you can use `--enableBalancePlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp. If you do not specify an application plan, files are randomly allocated.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableBalancePlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableDynamicPlan

If the files to be copied are significantly different in size and most of the files are small files, you can use `--enableDynamicPlan` to optimize the job allocation plan. This improves the copy performance of Jindo DistCp.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableDynamicPlan --parallelism 20
```

> **Note** You cannot use this option in the same command as `--groupBy` or `--targetSize`.

--enableTransaction

`--enableTransaction` ensures the integrity of job levels and transaction support among jobs. Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-enableTransaction --parallelism 20
```

--diff

After files are copied, you can use `--diff` to check the differences between the source and destination directories.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-diff
```

If all files are copied, the following information is returned:

```
INFO distcp.JindoDistCp: distcp has been done completely
```

If some files are not copied, a *manifest* file that contains a list of these files is generated in the destination directory. Then, you can use `--copyFromManifest` and `--previousManifest` to copy the files in the list to the destination directory. This way, the data volume and file quantity are verified. If Jindo DistCp has performed compression or decompression operations during the copy process, `--diff` does not return accurate file size differences.

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-dest oss://<yourBucketName>/hourly_table --previousManifest=file:///opt/manifest-2020-04-1
7.gz --copyFromManifest --parallelism 20
```

> **Note** If your destination directory is an HDFS directory, you must specify `--dest` in the format of */path*, *hdfs://hostname:ip/path*, or *hdfs://headerIp:ip/path*. Other formats, such as *hdfs:///path* and *hdfs:/path*, are not supported.

Check DistCp Counters

Run the following commands to check DistCp Counters in the counter information of MapReduce:

```
Distcp Counters
        Bytes Destination Copied=11010048000
        Bytes Source Read=11010048000
        Files Copied=1001
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
```

Note If Jindo DistCp has performed compression or decompression operations during the copy process, the values of `Bytes Destination Copied` and `Bytes Source Read` may be different.

Use an AccessKey pair to access OSS

If you want to access OSS from an instance outside EMR or AccessKey-free access is not supported, you can use an AccessKey pair to access OSS. Set the --key, --secret, and --endPoint parameters in the command to specify an AccessKey pair.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table --key <yourAccessKeyId> --secret <yourAccessKeySecret> --endPoint oss-cn-hangzhou.aliyuncs.com --parallelism 20
```

yourAccessKeyId is the AccessKey ID of your Alibaba Cloud account. yourAccessKeySecret is the AccessKey secret of your Alibaba Cloud account.

Write data to OSS IA or Archive storage

When you use a DistCp task to write data to OSS, you can write data to OSS Archive or Infrequent Access (IA) storage.

- Example of the Archive storage class (`--archive`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table --policy archive --parallelism 20
  ```

- Example of the IA storage class (`--ia`):

  ```
  jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table --policy ia --parallelism 20
  ```

Clean up residual files

When you run a DistCp task, files that are not correctly uploaded may be generated in your destination directory. The files are managed by OSS based on uploadId and may be invisible to users. In this case, you can specify the --cleanUpPending parameter in the command. This way, the system automatically cleans up the residual files after the task is complete. Alternatively, you can also clean up the files in the OSS console.

Example:

```
jindo distcp --src /data/incoming/hourly_table --dest oss://<yourBucketName>/hourly_table -
-cleanUpPending --parallelism 20
```

13.SmartData 2.2.x and earlier

13.1. Use JindoFS in EMR V3.20.0 to V3.22.0 (V3.22.0 excluded)

This topic describes how to configure and use JindoFS, and its use scenarios.

Overview

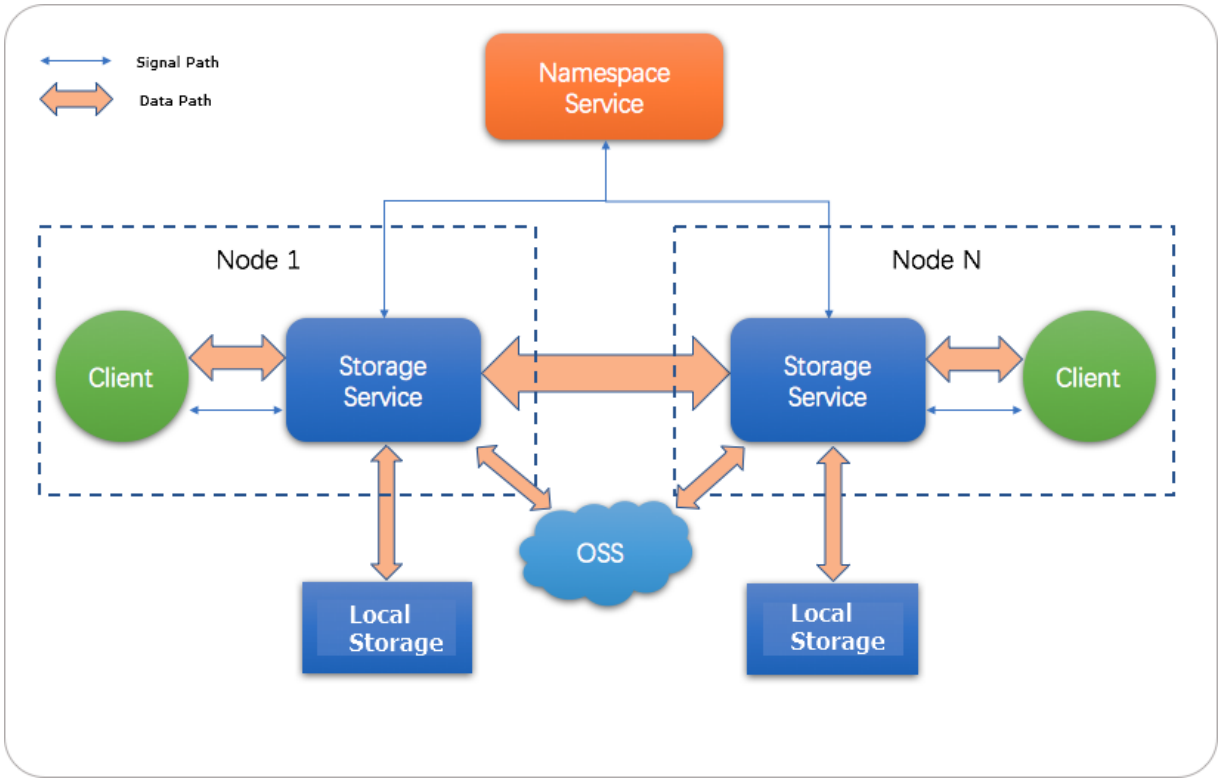
JindoFS is a cloud-native file system that combines the advantages of Object Storage Service (OSS) and local storage. JindoFS is also a next-generation storage system that provides efficient and reliable storage services for cloud computing in E-MapReduce (EMR).

JindoFS supports the block storage mode and cache mode.

JindoFS adopts a heterogeneous multi-backup mechanism. Storage Service provides the data storage capability. Data is stored in OSS to ensure high reliability. Redundant backups are stored in the local cluster to accelerate read operations. Namespace Service manages metadata of JindoFS. In this case, metadata is queried from Namespace Service instead of OSS, which improves query performance. This query method of JindoFS is similar to that of Hadoop Distributed File System (HDFS).

Note

- EMR V3.20.0 and later support JindoFS. To use JindoFS, select the related services when you create an EMR cluster.
- This topic describes how to use JindoFS in EMR V3.20.0 to V3.22.0 (V3.22.0 excluded). For more information about how to use JindoFS in EMR V3.22.0 or later, see Use JindoFS in EMR V3.22.0 to V3.26.3.



Scenarios

EMR has three storage systems: EMR OssFileSystem, EMR HDFS, and EMR JindoFS. Among them, OssFileSystem and JindoFS store data in the cloud. The following table compares the features of three EMR storage systems and Hadoop support for Alibaba Cloud OSS.

Feature	Hadoop support for Alibaba Cloud OSS	E-MapReduce OssFileSystem	E-MapReduce HDFS	E-MapReduce JindoFS
Storage capacity	Tremendous	Tremendous	Depending on the EMR cluster scale	Tremendous
Reliability	High	High	High	High
Factor that affects throughput	Server	I/O performance of caches on disks in the EMR cluster	I/O performance of disks in the EMR cluster	I/O performance of disks in the EMR cluster
Metadata query efficiency	Low	Medium	High	High
Scale-out operation	Easy	Easy	Easy	Easy
Scale-in operation	Easy	Easy	Node decommission required	Easy
Data locality	None	Weak	Strong	Medium

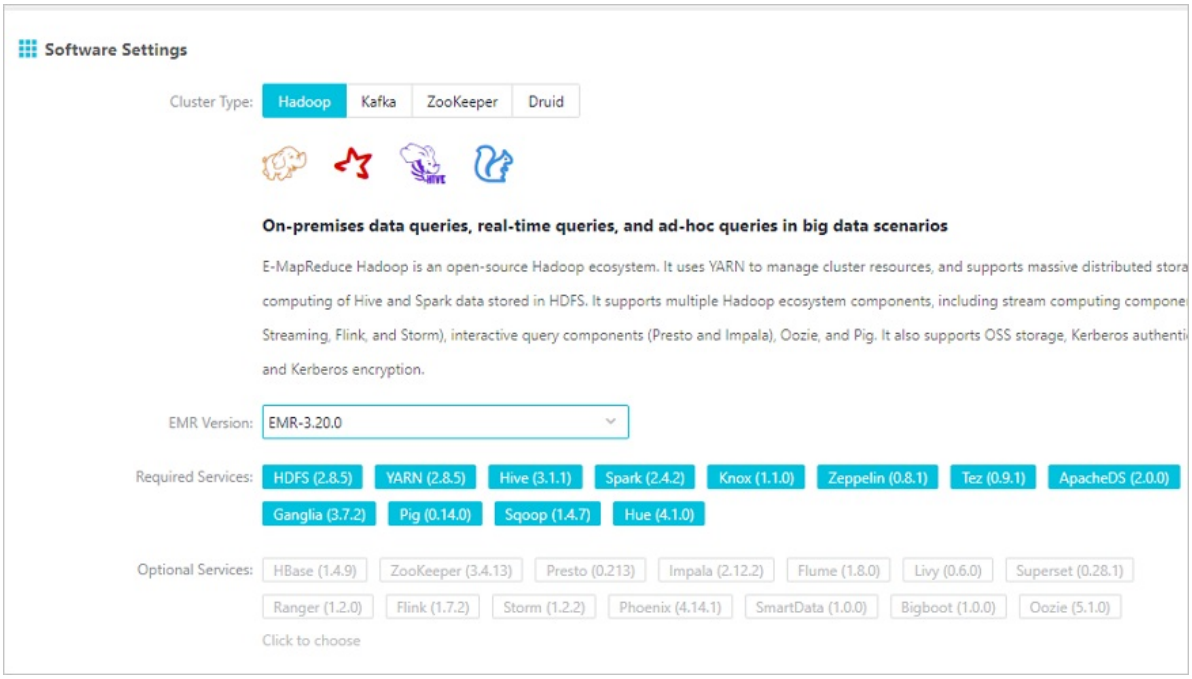
The block storage mode of JindoFS has the following features:

- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.
- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

Prepare the environment

- Create an EMR cluster

  Select a version from EMR V3.20.0 to V3.22.0 (V3.22.0 excluded). Select **SmartData** and **Bigboot** from the optional services. For more information about how to create an EMR cluster, see Create a cluster. Bigboot provides distributed data management and component management services in EMR. Based on Bigboot, SmartData provides JindoFS for the application layer.



- Configure JindoFS

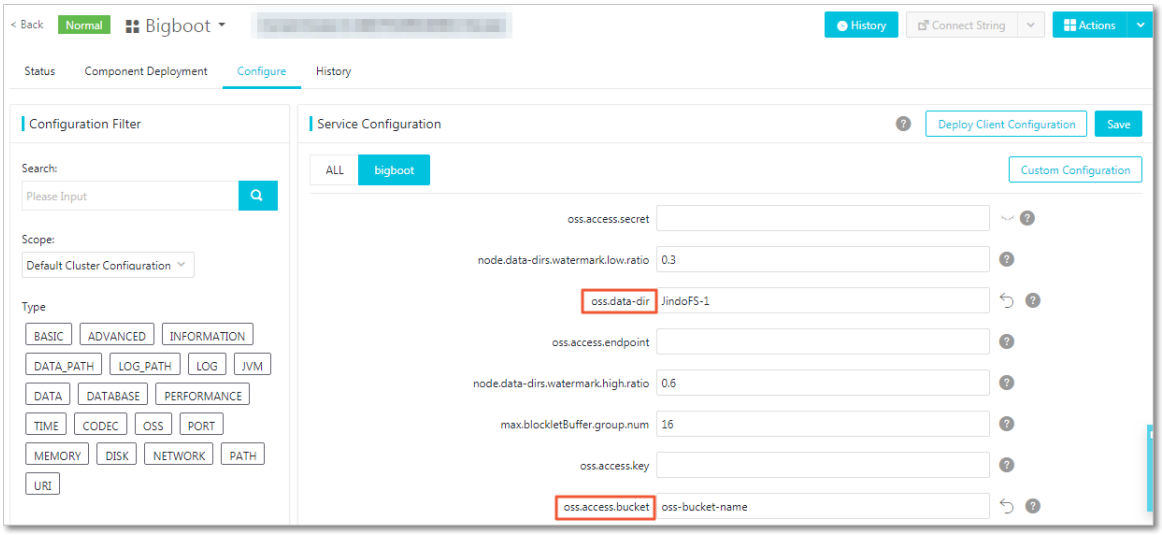
  JindoFS provided by SmartData uses OSS as the storage backend. Therefore, you must configure parameters related to OSS before you use JindoFS. EMR provides the following two configuration methods: 1. Modify parameters related to Bigboot after you create an EMR cluster, and then restart SmartData for the configurations to take effect. 2. Add custom configurations when you create an EMR cluster. In this case, the related services are restarted based on custom parameters after the EMR cluster is created.

  - Initialize parameters after the EMR cluster is created
    - `oss.access.bucket` specifies the name of the OSS bucket.

- `oss.data-dir` specifies the directory of JindoFS in the OSS bucket. The directory only serves as the storage backend for JindoFS. The data generated in the directory cannot be damaged. The directory is automatically created when JindoFS writes data. You do not need to create the directory in advance.
- `oss.access.endpoint` specifies the region where the OSS bucket resides.
- `oss.access.key` specifies the AccessKey ID used to access the OSS bucket.
- `oss.access.secret` specifies the AccessKey secret used to access the OSS bucket.

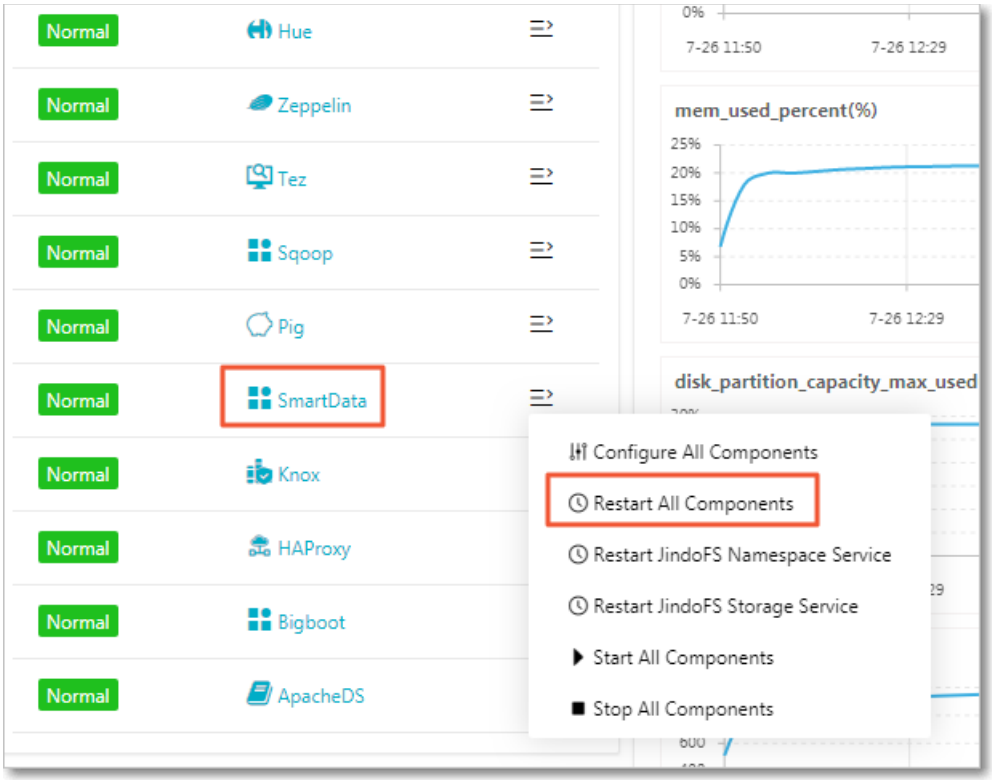
We recommend that you store data in an OSS bucket that is in the same region as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.

You can configure all parameters related to JindoFS in Bigboot, as shown in the following figure. The parameters framed in red are required.



Note JindoFS supports multiple namespaces. A namespace named test is used in this topic.

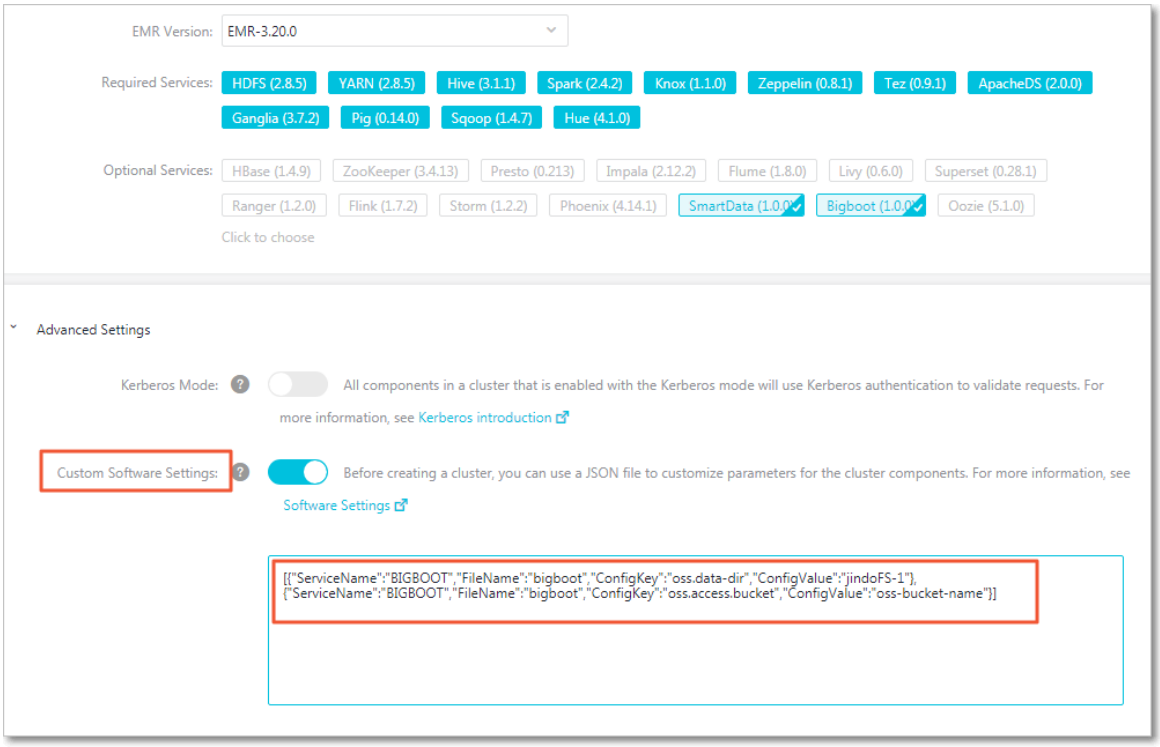
Save and deploy the JindoFS configuration. Restart all components in SmartData to use JindoFS.


Normal
Hue
Normal
Zeppelin
Normal
Tez
Normal
Sqoop
Normal
Pig
Normal
SmartData
Normal
Knox
Normal
HAProxy
Normal
Bigboot
Normal
ApacheDS
Configure All Components
Restart All Components
Restart JindoFS Namespace Service
Restart JindoFS Storage Service
Start All Components
Stop All Components
mem_used_percent(%)
disk_partition_capacity_max_used
7-26 11:50
7-26 12:29

- Add custom configurations when you create an EMR cluster

  You can add custom configurations when you create an EMR cluster. Assume that you want to create an EMR cluster in the same region as an OSS bucket to access OSS without using an AccessKey pair. Turn on **Custom Software Settings** as shown in the following figure. Add the following configurations including `oss.data-dir` and `oss.access.bucket` to the field in the Advanced Settings section:

```
[
    {
    "ServiceName":"BIGBOOT",
    "FileName":"bigboot",
    "ConfigKey":"oss.data-dir",
    "ConfigValue":"jindoFS-1"
    },
    {
    "ServiceName":"BIGBOOT",
    "FileName":"bigboot",
    "ConfigKey":"oss.access.bucket",
    "ConfigValue":"oss-bucket-name"
    }
]
```



Use JindoFS

The use of JindoFS is similar to that of HDFS. JindoFS also provides a prefix. To use JindoFS, you only need to replace the hdfs prefix with the jfs prefix. Example:

```
hadoop fs -ls jfs:/// hadoop fs -mkdir jfs:///test-dirhadoop fs -put test.log jfs:///test-d
ir/
```

Data can be read from JindoFS only when Hadoop, Hive, and Spark jobs are running in the EMR cluster.

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. Alibaba Cloud uses the node.data-dirs.watermark.high.ratio and node.data-dirs.watermark.low.ratio parameters to adjust the space usage of local disks. The values of both parameters are in the range of 0 to 1 to indicate the percentage of space usage. JindoFS uses the total storage capacity of all data disks by default. The node.data-dirs.watermark.high.ratio parameter specifies the upper limit of space usage on each disk. Less frequently accessed data stored on a disk is released if the space used by JindoFS reaches the upper limit. The node.data-dirs.watermark.low.ratio parameter specifies the lower limit of space usage on each disk. After the space usage of a disk reaches the upper limit, less frequently accessed data is released until the space usage of the disk reaches the lower limit. You can configure the upper limit and lower limit to adjust and assign disk space to JindoFS. Make sure that the upper limit is greater than the lower limit.

Configure the storage policy

JindoFS provides multiple storage policies to meet different storage needs. The following table lists four available storage policies for a directory.

Policy	Description
COLD	Data has a backup in OSS but no backups in the local cluster. This policy is suitable for storing cold data.
WARM	The default storage policy. Data has a backup in OSS and a backup in the local cluster. The local backup can accelerate read operations.
HOT	Data has a backup in OSS and multiple backups in the local cluster. Local backups can accelerate read operations on hot data.
TEMP	Data has a backup in the local cluster but no backups in OSS. This policy is suitable for storing temporary data. The local backup can accelerate read and write operations on the temporary data. However, this may lower data reliability.

JindoFS provides a command-line tool Admin to configure the storage policy of a directory. The default storage policy is WARM. New files are stored based on the storage policy configured for the parent directory. Run the following command to configure the storage policy:

```
jindo dfsadmin -R -setStoragePolicy [path] [policy]
```

Run the following command to obtain the storage policy configured for a directory:

```
jindo dfsadmin -getStoragePolicy [path]
```

> **Note** The *[path]* parameter specifies the directory. The *-R* option specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.

The Admin tool provides the archive command to archive cold data.

This command allows you to explicitly evict local blocks. Assume that Hive partitions a table by day. If the data generated a week ago in partitioned tables is infrequently accessed, you can run the archive command on the directory that stores such data on a regular basis. Then, the backups stored in the local cluster are evicted, whereas the backups in OSS are retained.

Run the following archive command:

```
jindo dfsadmin -archive [path]
```

> **Note** The *[path]* parameter specifies the directory in which the data is to be archived.

13.2. Use JindoFS in EMR V3.22.0 to V3.26.3

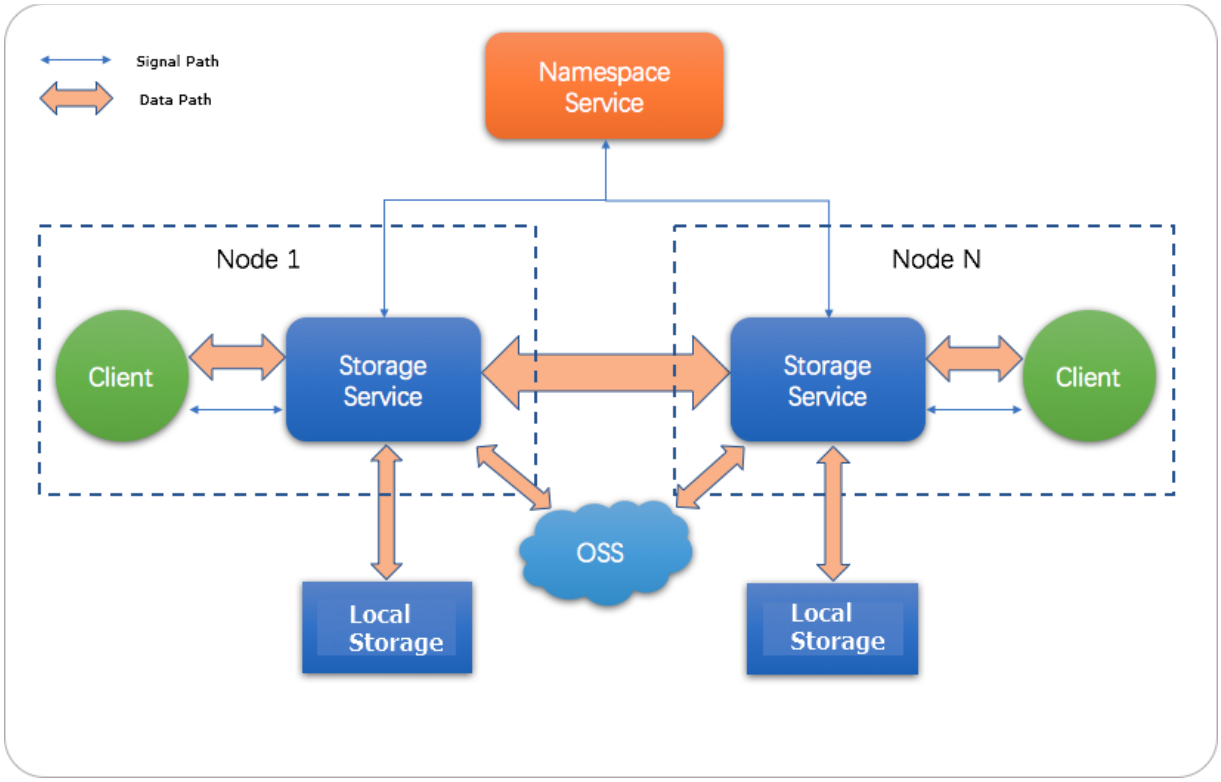
JindoFS is a cloud-native file system that combines the advantages of Object Storage Service (OSS) and local storage. JindoFS is also a next-generation storage system that provides efficient and reliable storage services for cloud computing in E-MapReduce (EMR). This topic describes how to configure and use JindoFS, and its use scenarios.

Overview

JindoFS supports the block storage mode and cache mode.

JindoFS adopts a heterogeneous multi-backup mechanism. Storage Service provides the data storage capability. Data is stored in OSS to ensure high reliability. Redundant backups are stored in the local cluster to accelerate read operations. Namespace Service manages metadata of JindoFS. In this case, metadata is queried from Namespace Service instead of OSS, which improves query performance. This query method of JindoFS is similar to that of Hadoop Distributed File System (HDFS).

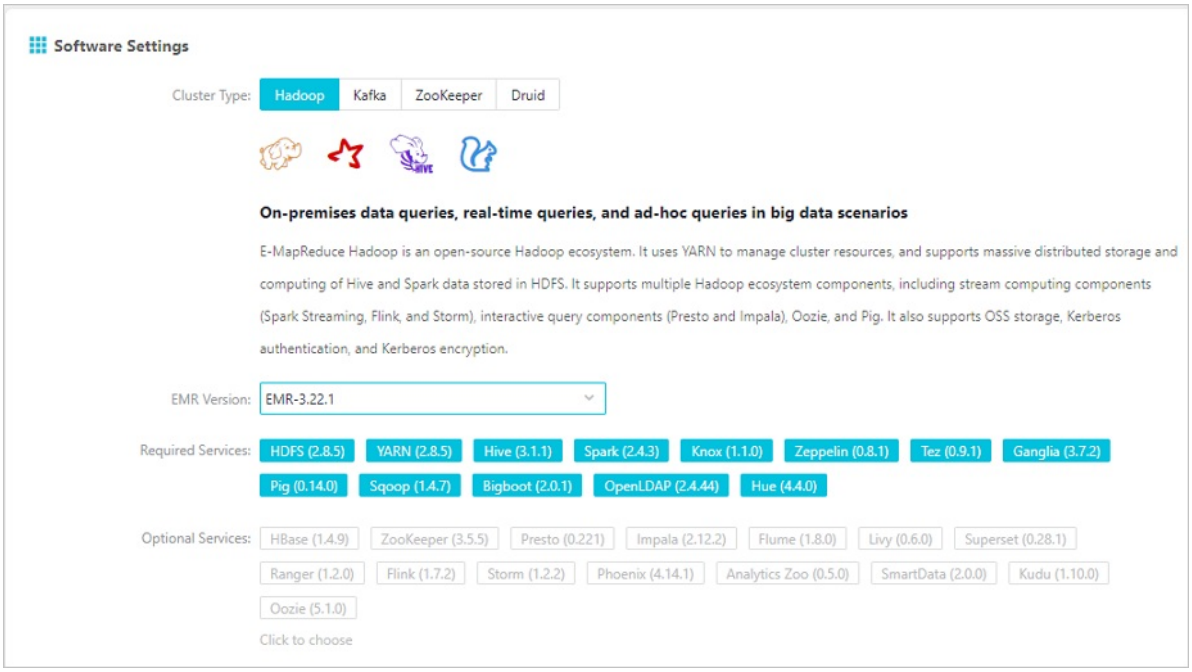
> **Note**
>
> - EMR V3.20.0 and later support JindoFS. To use JindoFS, select the related services when you create an EMR cluster.
> - This topic describes how to use JindoFS in EMR V3.22.0 or later. For more information about how to use JindoFS in EMR V3.20.0 to V3.22.0 (V3.22.0 excluded), see Use JindoFS in EMR V3.20.0 to V3.22.0 (V3.22.0 excluded).



Prepare the environment

- Create an EMR cluster

  Select EMR V3.22.0 or a later version. Select **SmartData** from the optional services. For more information about how to create an EMR cluster, see Create a cluster.



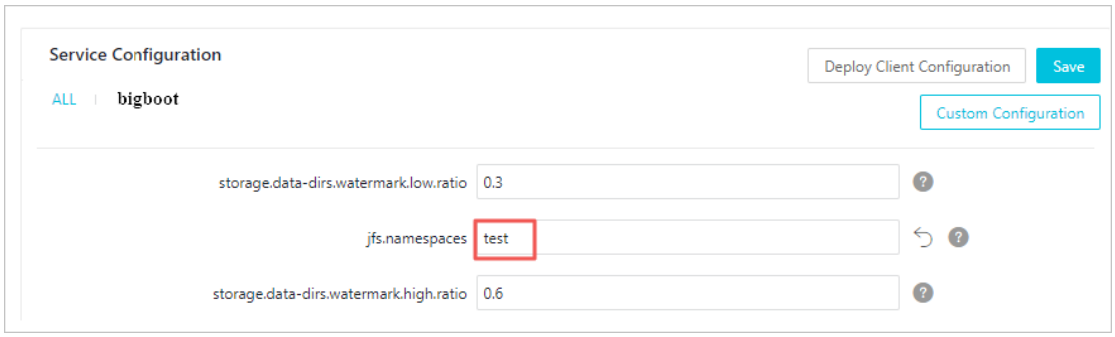
- Configure JindoFS

JindoFS provided by SmartData uses OSS as the storage backend. Therefore, you must configure parameters related to OSS before you use JindoFS. EMR provides the following two configuration methods: 1. Modify parameters related to Bigboot after you create an EMR cluster, and then restart SmartData for the configurations to take effect. 2. Add custom configurations when you create an EMR cluster. In this case, the related services are restarted based on custom parameters after the EMR cluster is created.

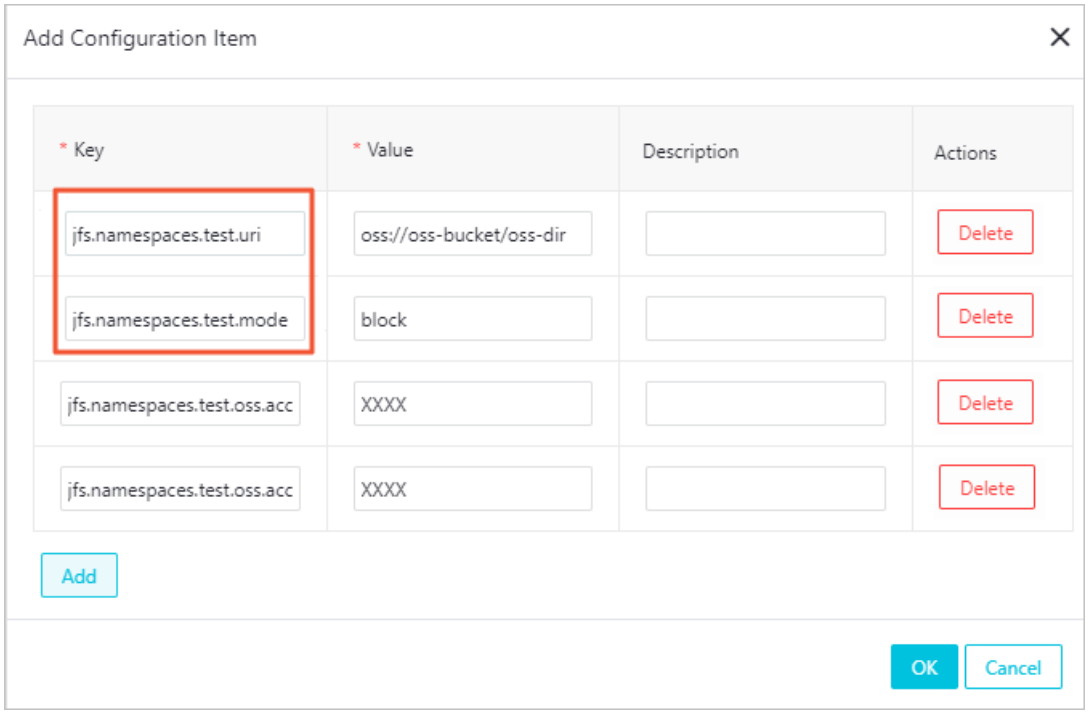
- Initialize parameters after the EMR cluster is created

  You can configure all parameters related to JindoFS in Bigboot, as shown in the following figures.

  a. In the **Service Configuration** section, click the **bigboot** tab.



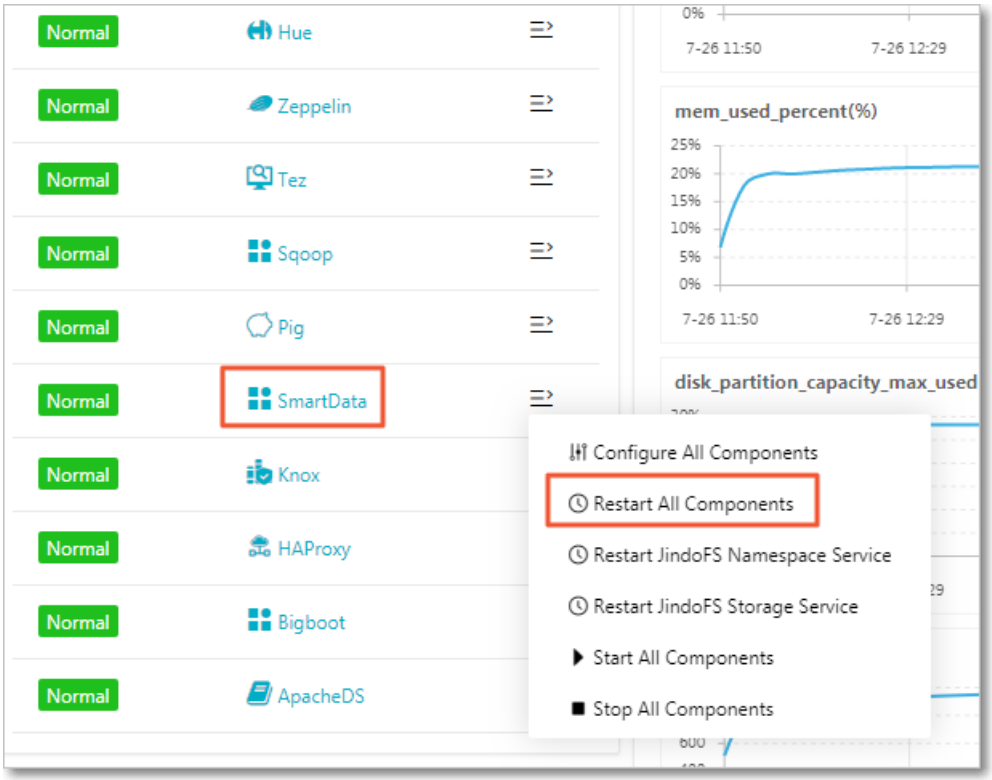
  b. Click **Custom Configuration** in the upper-right corner.



> **Note**
> - The parameters framed in red in the preceding figures are required.
> - JindoFS supports multiple namespaces. A namespace named test is used in this topic.

Parameter	Description	Example
jfs.namespaces	The namespace supported by JindoFS. Separate multiple namespaces with commas (,).	test
jfs.namespaces.test.uri	The storage backend of the test namespace.	*oss://oss-bucket/oss-dir* **Note** You can set the value to a directory in an OSS bucket. In this case, this directory serves as the root directory, in which the test namespace reads and writes data.
jfs.namespaces.test.mode	The storage mode of the test namespace.	block **Note** JindoFS supports the block storage mode and cache mode.
jfs.namespaces.test.oss.access.key	The AccessKey ID used to access the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret used to access the OSS bucket that serves as the storage backend.	

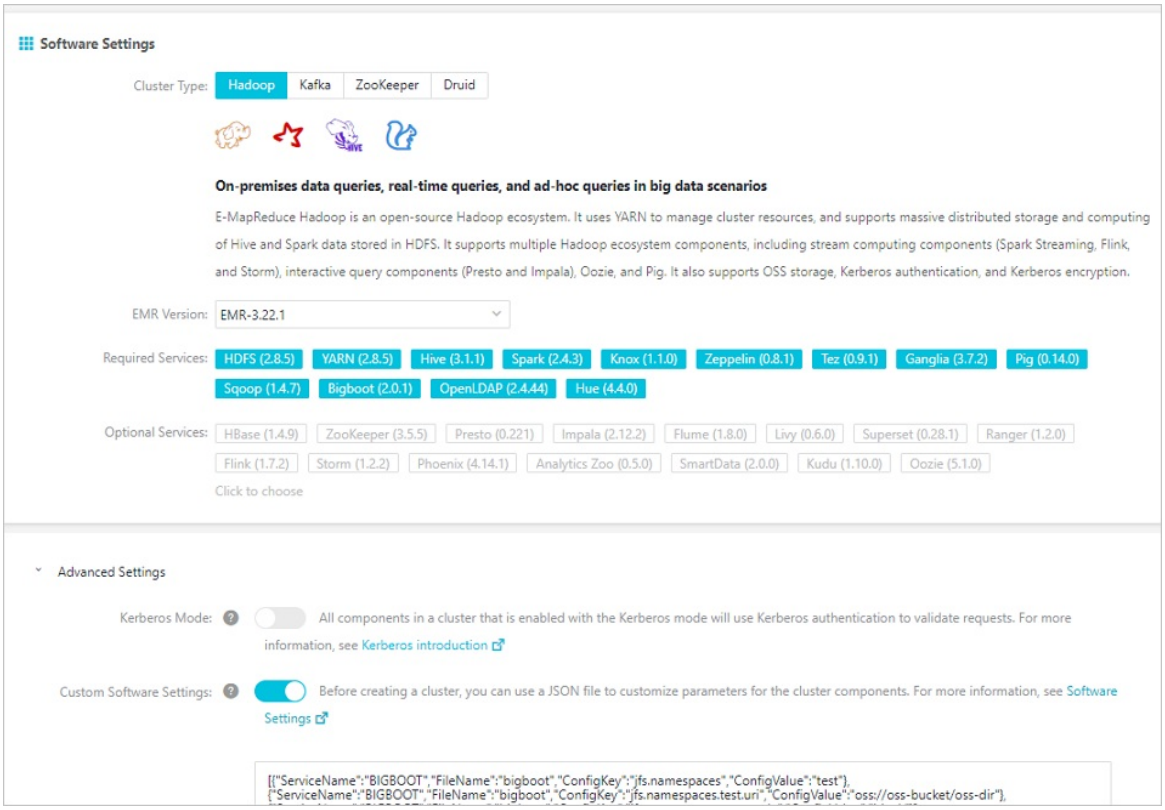
Save and deploy the JindoFS configuration. Restart all components in SmartData to use JindoFS.



- Add custom configurations when you create an EMR cluster

  You can add custom configurations when you create an EMR cluster. Assume that you want to create an EMR cluster in the same region as an OSS bucket to access OSS without using an AccessKey pair. Turn on **Custom Software Settings** as shown in the following figure. Add the following configuration to the field in the Advanced Settings section to customize parameters for the test namespace:

```
[
{
"ServiceName":"BIGBOOT",
"FileName":"bigboot",
"ConfigKey":"jfs.namespaces","ConfigValue":"test"
},{
"ServiceName":"BIGBOOT",
"FileName":"bigboot",
"ConfigKey":"jfs.namespaces.test.uri",
"ConfigValue":"oss://oss-bucket/oss-dir"
},{
"ServiceName":"BIGBOOT",
"FileName":"bigboot",
"ConfigKey":"jfs.namespaces.test.mode",
"ConfigValue":"block"
}
]
```



Use JindoFS

The use of JindoFS is similar to that of HDFS. JindoFS also provides a prefix. To use JindoFS, you only need to replace the hdfs prefix with the jfs prefix.

JindoFS supports most of the computing components in the EMR cluster, including Hadoop, Hive, Spark, Flink, Presto, and Impala.

Examples:

- Run shell commands

```
hadoop fs -ls jfs://your-namespace/
hadoop fs -mkdir jfs://your-namespace/test-dir
hadoop fs -put test.log jfs://your-namespace/test-dir/
hadoop fs -get jfs://your-namespace/test-dir/test.log . /
```

- Run a MapReduce job

```
hadoop jar /usr/lib/hadoop-current/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.5
.jar teragen -Dmapred.map.tasks=1000 10737418240 jfs://your-namespace/terasort/input
hadoop jar /usr/lib/hadoop-current/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.5
.jar terasort -Dmapred.reduce.tasks=1000 jfs://your-namespace/terasort/input jfs://your-n
amespace/terasort/output
```

- Run a Spark SQL test

```
CREATE EXTERNAL TABLE IF NOT EXISTS src_jfs (key INT, value STRING) location 'jfs://your-
namespace/Spark_sql_test/';
```

Control disk space usage

JindoFS uses OSS as the data storage backend, which allows you to store large volumes of data. However, the capacity of local disks is limited. JindoFS automatically deletes cold data in local disks. Alibaba Cloud uses the node.data-dirs.watermark.high.ratio and node.data-dirs.watermark.low.ratio parameters to adjust the space usage of local disks. The values of both parameters are in the range of 0 to 1 to indicate the percentage of space usage. JindoFS uses the total storage capacity of all data disks by default. The node.data-dirs.watermark.high.ratio parameter specifies the upper limit of space usage on each disk. Less frequently accessed data stored on a disk is released if the space used by JindoFS reaches the upper limit. The node.data-dirs.watermark.low.ratio parameter specifies the lower limit of space usage on each disk. After the space usage of a disk reaches the upper limit, less frequently accessed data is released until the space usage of the disk reaches the lower limit. You can configure the upper limit and lower limit to adjust and assign disk space to JindoFS. Make sure that the upper limit is greater than the lower limit.

Configure the storage policy

JindoFS provides multiple storage policies to meet different storage needs. The following table lists four available storage policies for a directory.

Policy	Description
COLD	Data has only a backup in OSS but no backups in the local cluster. This policy is suitable for storing cold data.
WARM	The default storage policy. Data has a backup in OSS and a backup in the local cluster. The local backup can accelerate read operations.
HOT	Data has a backup in OSS and multiple backups in the local cluster. Local backups can accelerate read operations on hot data.
TEMP	Data has only a backup in the local cluster. This policy is suitable for storing temporary data. The local backup can accelerate read and write operations on the temporary data. However, this may lower data reliability.

JindoFS provides a command-line tool Admin to configure the storage policy of a directory. The default storage policy is WARM. New files are stored based on the storage policy configured for the parent directory. Run the following command to configure the storage policy:

```
jindo dfsadmin -R -setStoragePolicy [path] [policy]
```

Run the following command to obtain the storage policy configured for a directory:

```
jindo dfsadmin -getStoragePolicy [path]
```

> **Note** The *[path]* parameter specifies the directory. The *-R* option specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.

Use the Admin tool

JindoFS supports the archive and jindo commands of the Admin tool.

- The Admin tool provides the archive command to archive cold data.

  This command allows you to explicitly evict local blocks. Assume that Hive partitions a table by day. If the data generated a week ago in partitioned tables is infrequently accessed, you can run the archive command on the directory that stores such data on a regular basis. Then, the backups stored in the local cluster are evicted, whereas the backups in OSS are retained.

  Run the following archive command:

  ```
  jindo dfsadmin -archive [path]
  ```

  Note The *[path]* parameter specifies the directory in which the data is to be archived.

- The Admin tool provides the jindo command to manage metadata of JindoFS for Namespace Service.

  ```
  jindo dfsadmin [-options]
  ```

  Note You can run the `jindo dfsadmin --help` command to obtain help information.

The Admin tool provides the diff and sync commands for the cache mode.

- The diff command is used to display the difference between the data stored in the local cluster and that in OSS.

  ```
  jindo dfsadmin -R -diff [path]
  ```

  Note By default, you can use the diff command to display the difference between the metadata stored in the local cluster and that in the sub-directories of the directory specified by the `[path]` parameter. The `-R` option specifies that a recursive operation is performed to compare the metadata stored in the local cluster with that stored in all sub-directories of the directory specified by the `[path]` parameter.

- The sync command is used to synchronize the metadata between the local cluster and OSS.

  ```
  jindo dfsadmin -R -sync [path]
  ```

  Note The `[path]` parameter specifies the directory in which the metadata is to be synchronized. By default, you can use the sync command to synchronize the metadata in the sub-directories of the directory specified by the `[path]` parameter to the local cluster. The `-R` option specifies that a recursive operation is performed to synchronize the metadata in all sub-directories of the directory specified by the `[path]` parameter.

13.3. Use the block storage mode

This topic describes the block storage mode of JindoFS and its use scenarios.

Overview

Block storage is the most efficient mode to read and write data and query metadata. In addition, it supports Hadoop Distributed File System (HDFS) semantics related to data locality. JindoFS also provides an external client so that you can access JindoFS from the outside of an E-MapReduce (EMR) cluster.

JindoFS uses Object Storage Service (OSS) as the storage backend. In block storage mode, JindoFS stores data as blocks in OSS and uses Namespace Service to maintain metadata. This ensures high performance when you read and write data or query metadata.

Scenarios

EMR has three storage systems: EMR OssFileSystem, EMR HDFS, and EMR JindoFS. Among them, OssFileSystem and JindoFS store data in the cloud. The following table compares the features of three EMR storage systems and Hadoop support for Alibaba Cloud OSS.

Feature	Hadoop support for Alibaba Cloud OSS	E-MapReduce OssFileSystem	E-MapReduce HDFS	E-MapReduce JindoFS
Storage capacity	Tremendous	Tremendous	Depending on the EMR cluster scale	Tremendous
Reliability	High	High	High	High
Factor that affects throughput	Server	I/O performance of caches on disks in the EMR cluster	I/O performance of disks in the EMR cluster	I/O performance of disks in the EMR cluster
Metadata query efficiency	Low	Medium	High	High
Scale-out operation	Easy	Easy	Easy	Easy
Scale-in operation	Easy	Easy	Node decommission required	Easy
Data locality	None	Weak	Strong	Medium

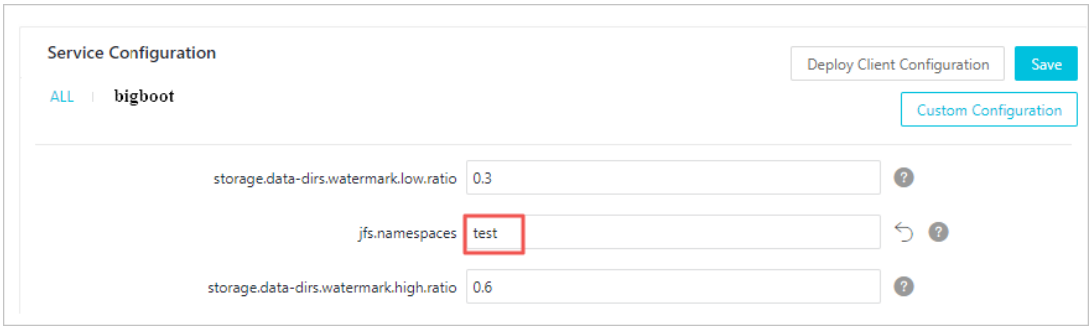
The block storage mode of JindoFS has the following features:

- JindoFS offers tremendous and scalable storage capacity by using OSS as the storage backend. The storage capacity is independent of the EMR cluster scale. The local cluster can be scaled in or out as required.
- JindoFS stores some backup data in the local cluster to accelerate read operations. This improves the throughput by using limited local storage capacity, especially for Write Once Read Many (WORM) solutions.
- JindoFS provides efficient metadata query similar to HDFS. Compared with OssFileSystem, JindoFS saves much time in metadata query. In addition, JindoFS avoids system instability when data and metadata are frequently accessed.
- JindoFS ensures maximal data locality when jobs are executed in the EMR cluster. This reduces the load on network transmission and improves the read performance.

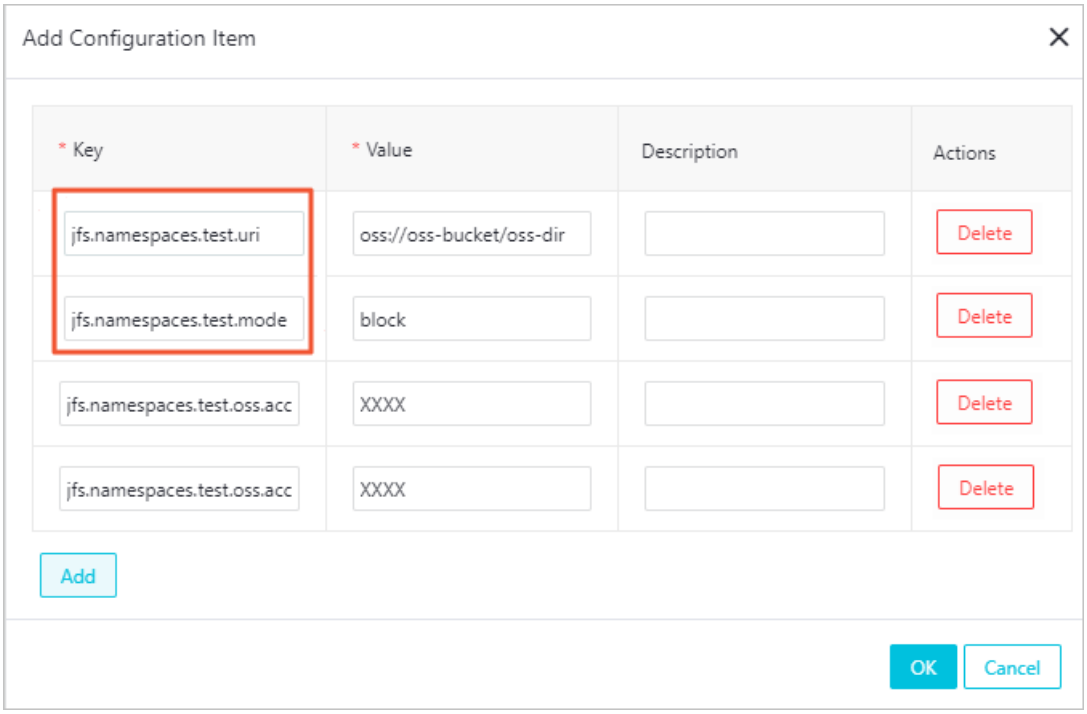
Configure JindoFS

You can set all parameters related to JindoFS in Bigboot, as shown in the following figures.

Modify a parameter



Add parameters



Note

- The parameters framed in red in the preceding figures are required.
- JindoFS supports multiple namespaces. A namespace named test is used in this topic.

Parameter	Description	Example
jfs.namespaces	The namespace supported by JindoFS. Separate multiple namespaces with commas (,).	test

Parameter	Description	Example
jfs.namespaces.test.uri	The storage backend of the test namespace.	*oss://oss-bucket/oss-dir* **Note** You can set the value to a directory in an OSS bucket. In this case, this directory serves as the root directory, in which the test namespace reads and writes data.
jfs.namespaces.test.mode	The storage mode of the test namespace.	block
jfs.namespaces.test.oss.access.key	The AccessKey ID used to access the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret used to access the OSS bucket that serves as the storage backend.	

Save and deploy the JindoFS configuration. Restart Namespace Service in SmartData to use JindoFS.

Configure the storage policy

JindoFS provides multiple storage policies to meet different storage needs. The following table lists four available storage policies for a directory.

Policy	Description
COLD	Data has a backup in OSS but no backups in the local cluster. This policy is suitable for storing cold data.
WARM	The default storage policy. Data has a backup in OSS and a backup in the local cluster. The local backup can accelerate read operations.
HOT	Data has a backup in OSS and multiple backups in the local cluster. Local backups can accelerate read operations on hot data.
TEMP	Data has a backup in the local cluster but no backups in OSS. This policy is suitable for storing temporary data. The local backup can accelerate read and write operations on the temporary data. However, this may lower data reliability.

JindoFS provides a command-line tool Admin to configure the storage policy of a directory. The default storage policy is WARM. New files are stored based on the storage policy configured for the parent directory. Run the following command to configure the storage policy:

```
jindo dfsadmin -R -setStoragePolicy [path] [policy]
```

Run the following command to obtain the storage policy configured for a directory:

```
jindo dfsadmin -getStoragePolicy [path]
```

> **Note** The *[path]* parameter specifies the directory. The *-R* option specifies that a recursive operation is performed to configure the same storage policy for all sub-directories of the directory.

The Admin tool provides the archive command to archive cold data.

This command allows you to explicitly evict local blocks. Assume that Hive partitions a table by day. If the data generated a week ago in partitioned tables is infrequently accessed, you can run the archive command on the directory that stores such data on a regular basis. Then, the backups stored in the local cluster are evicted, whereas the backups in OSS are retained.

Run the following archive command:

```
jindo dfsadmin -archive [path]
```

> **Note** The *[path]* parameter specifies the directory in which the data is to be archived.

13.4. Use the cache mode

This topic describes the cache mode of JindoFileSystem (JindoFS) and its use scenarios.

Overview

In cache mode, JindoFS stores data files as objects in Object Storage Service (OSS) and caches data and metadata of these files in the local cluster based on the requirements for accessing these files. This accelerates read and write operations on data and metadata. In addition, the cache mode provides multiple policies for you to synchronize metadata as required.

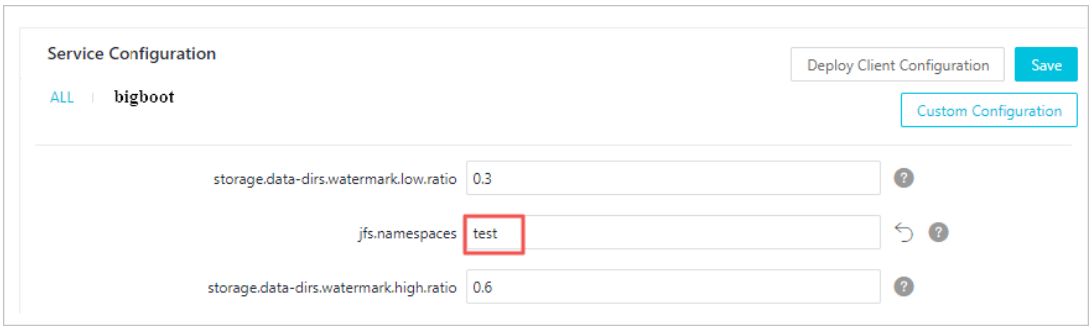
Scenarios

The cache mode is compatible with original OSS semantics. In cache mode, JindoFS stores data files as objects in OSS and caches data and metadata in the local cluster. This ensures that JindoFS is compatible with the OSS client, E-MapReduce (EMR) OssFileSystem, and other OSS interactive applications. You can also access data that is stored in OSS before you configure JindoFS, without the need to migrate or convert data. In addition, local caches accelerate read and write operations on data and metadata.

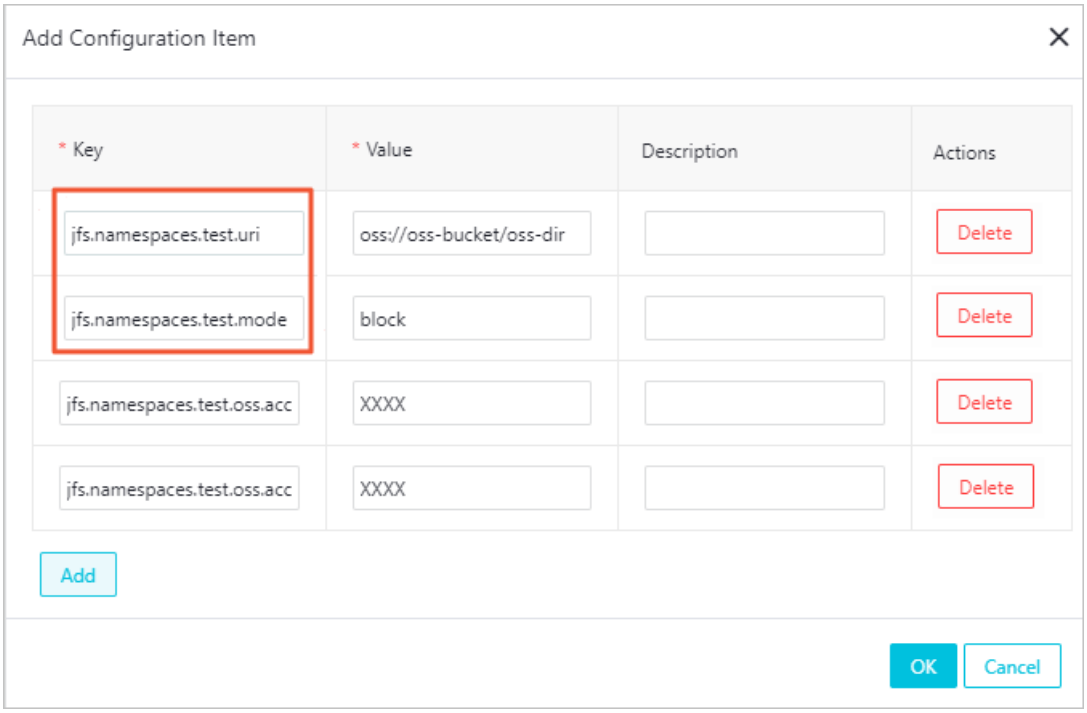
Configure JindoFS

You can configure all parameters related to JindoFS in Bigboot, as shown in the following figure.

Update a ConfigMap



Add Configuration Item



Note

- The parameters framed in red in the preceding figure are required.
- JindoFS supports multiple namespaces. A namespace named test is used in this topic.

Parameter	Description	Example
jfs.namespaces	The namespace supported by JindoFS. Separate multiple namespaces with commas (,).	test
jfs.namespaces.test.uri	The storage backend of the test namespace.	*oss://oss-bucket/* **Note** You can set the value to a directory in an OSS bucket. In this case, this directory serves as the root directory, in which the test namespace reads and writes data. Generally, you can set the value to an OSS bucket to ensure that the path is the same as that in OSS.
jfs.namespaces.test.mode	The storage mode of the test namespace. Set this parameter to cache.	cache

Parameter	Description	Example
jfs.namespaces.test.oss.access.key	The AccessKey ID used to access the OSS bucket that serves as the storage backend.	*xxxx* **Note** We recommend that you store data in an OSS bucket that is in the same region and under the same account as your EMR cluster. This ensures high performance and stability. In this case, you do not need to configure the AccessKey ID and AccessKey secret because the OSS bucket allows password-free access from the EMR cluster.
jfs.namespaces.test.oss.access.secret	The AccessKey secret used to access the OSS bucket that serves as the storage backend.	

Save and deploy the JindoFS configuration. Restart Namespace Service in SmartData to use JindoFS.

Configure a metadata synchronization policy

In cache mode, some data may already exist in OSS before you configure JindoFS. In this scenario, after JindoFS is configured, the data and metadata are synchronized to JindoFS for future access. The data synchronization policy is that data is cached in the local cluster each time data is accessed. JindoFS supports two types of metadata synchronization policies: interval policy and loading policy.

- Interval policy:

  You can set the **namespace.sync.interval** parameter to specify the synchronization interval. The default value is **-1**, which indicates that JindoFS does not synchronize metadata from OSS.

  - If you set this parameter to **0**, JindoFS synchronizes metadata from OSS each time data is accessed.
  - If you set this parameter to a value greater than **0**, JindoFS synchronizes metadata from OSS at intervals of the set value in units of seconds.

    Note For example, if you set this parameter to **5**, JindoFS synchronizes metadata from OSS every 5 seconds.

- Loading policy:

You can set the **namespace.sync.loadtype** parameter to specify the loading policy. Valid values are never, once, and always. **never** indicates that JindoFS never synchronizes metadata from OSS. **once** indicates that JindoFS synchronizes metadata from OSS only once. This is the default value. **always** indicates that JindoFS synchronizes metadata from OSS each time data is accessed.

> **Note** The **namespace.sync.loadtype** parameter takes effect only when you do not specify the **namespace.sync.interval** parameter.

13.5. Use the password-free feature of JindoFS SDK

This topic describes how to access JindoFileSystem (JindoFS) in password-free mode from an ECS instance (not an instance in EMR clusters) when you use JindoFS SDK.

Prerequisites

1. You have ECS instances other than those for EMR. 2. The Hadoop ecosystem is used. 3. JindoFS SDK for Java is obtained.

Context

Before you can use JindoFS SDK, you must remove packages related to Jindo from your environment, such as *jboot.jar* and *smartdata-aliyun-jfs-*.jar*. To use Spark, you must also remove the packages in the */opt/apps/spark-current/jars/* directory.

Step 1: Create an instance RAM role

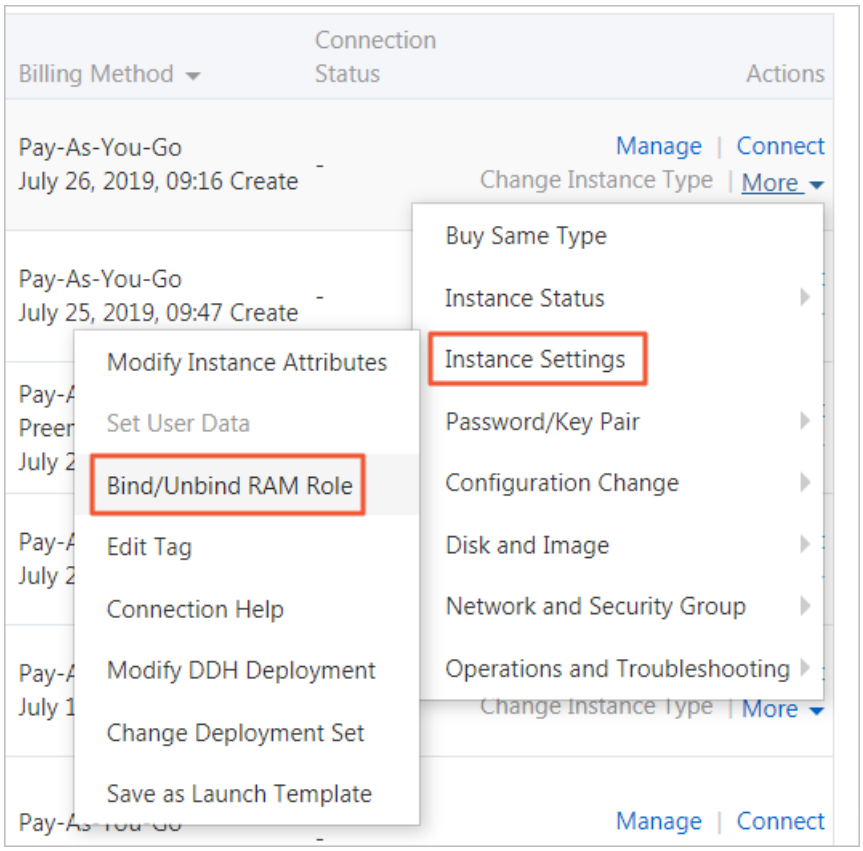
1. Log on to the RAM console with an Alibaba Cloud account.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click **Create RAM Role**. In the Create RAM Role panel, select **Alibaba Cloud Service** for Trusted entity type.
4. Click **Next**.
5. Enter a role name in the **RAM Role Name** field and select **Elastic Compute Service** from the **Select Trusted Service** drop-down list.
6. Click **OK**.

Step 2: Grant permissions to the RAM role

1. Log on to the RAM console with an Alibaba Cloud account.
2. Optional. If you do not use system permissions, you can create custom permissions. For more information, see the "(Optional) Create a custom authorization policy" section in Implement access control by using RAM.
3. In the left-side navigation pane, click **RAM Roles**.
4. Find the created RAM role and click **Input and Attach** in the Actions column.
5. In the panel that appears, select **System Policy** or **Custom Policy** for Type.
6. Enter the policy name.
7. Click **OK**.

Step 3: Bind the RAM role to an ECS instance

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images > Instances**.
3. In the top navigation bar, select a region.
4. Find the ECS instance to which you want to bind the RAM role and choose **More > Instance Settings > Bind/Unbind RAM Role** in the Actions column.



5. In the Bind/Unbind RAM Role dialog box, select the RAM role and click **OK**.

Step 4: Configure environment variables on ECS

Run one of the following commands to configure environment variables on ECS:

```
export CLASSPATH=/xx/xx/jindofs-2.5.0-sdk.jar
```

and

```
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/xx/xx/jindofs-2.5.0-sdk.jar
```

Step 5: Access JindoFS in password-free mode

1. Use Shell to access OSS.

```
hdfs dfs -ls/-mkdir/-put/..... oss://<ossPath>
```

2. Access OSS by using the FileSystem interface of Hadoop.

Example:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import java.net.URI;
public class test {
    public static void main(String[] args) throws Exception {
        FileSystem fs = FileSystem.get(new URI("ossPath"), new Configuration());
        RemoteIterator<LocatedFileStatus> iterator = fs.listFiles(new Path("ossPath"),
false);
        while (iterator.hasNext()){
            LocatedFileStatus fileStatus = iterator.next();
            Path fullPath = fileStatus.getPath();
            System.out.println(fullPath);
        }
    }
}
```

13.6. Use the external client

This topic describes the external client of JindoFS and its use scenarios.

Overview

JindoFS provides an external client so that you can access JindoFS from the outside of an E-MapReduce (EMR) cluster. If you want to access JindoFS from the external client, make sure that JindoFS is in block storage mode. You cannot access JindoFS from the external client if JindoFS is in cache mode. To access JindoFS in cache mode from the outside of an EMR cluster, use the common Object Storage Service (OSS) client because the cache mode is compatible with original OSS semantics.

Scenarios

The external client of JindoFS is compatible with Hadoop Distributed File System (HDFS). To access data stored in JindoFS from the external client, make sure that your application is connected to Namespace Service of JindoFS. However, you cannot access cached data in the local cluster from the external client. In this case, the performance of data access from the external client is not as efficient as that from the inside of an EMR cluster.

Configure the external client

Make sure that a namespace supported by JindoFS in block storage mode is configured. For more information, see Use the block storage mode.

1. Obtain the Bigboot package.

   Access the */usr/lib/bigboot-current* directory in the EMR cluster to obtain the Bigboot package.

   > **Note** The Bigboot package is developed based on native code, which may be incompatible with your operating system. Click to submit applications.

2. Set up the environment.

   Set the **BIGBOOT_HOME** variable to the root directory for installing Bigboot on your device. Add the *ext* and *lib* directories in the root directory to the **classpath** parameter of your big data processing component, such as Hadoop or Spark.

3. Copy the configuration file bigboot.cfg.external from the */usr/lib/bigboot-current/conf/* directory in the EMR cluster to the installation directory *conf/* on your device.

4. Configure Namespace Service.
   - `client.namespace.rpc.port` : the port for listening on Namespace Service.
   - `client.namespace.rpc.address` : the endpoint for listening on Namespace Service.

   > **Note** By default, the preceding two parameters are configured in the Bigboot configuration file of the EMR cluster.

5. Configure data access parameters.
   - `client.namespaces.{ YourNamespace}.oss.access.bucket` : the OSS bucket to be accessed.
   - `client.namespaces.{ YourNamespace}.oss.access.endpoint` : the endpoint of the OSS bucket.
   - `client.namespaces.{ YourNamespace}.oss.access.key` : the AccessKey ID used to access the OSS bucket.
   - `client.namespaces.{ YourNamespace}.oss.access.secret` : the AccessKey secret used to access the OSS bucket.

   > **Note** In the preceding parameters, `{YourNamespace}` specifies the namespace that you want to access from the external client. In this topic, a namespace named test is used.

   Example:

   ```
   client.namespace.rpc.port = 8101
   client.namespace.rpc.address = {RPC_Address}
   client.namespaces.test.oss.access.bucket = {YourOssBucket}
   client.namespaces.test.oss.access.endpoint = {YourOssEndpoint}
   client.namespaces.test.oss.access.key = {YourOssAccessKeyID}
   client.namespaces.test.oss.access.secret = {YourOssAccessKeySecret}
   ```

Verify the configuration

Perform the following operations:

- Run the following command to check whether the test namespace is correctly configured:

  ```
  hdfs dfs -ls jfs://test/
  ```

- Run the following commands to check whether data can be uploaded to or downloaded from the test namespace:

  ```
  hdfs dfs -put /etc/hosts  jfs://test/
  hdfs dfs -get jfs://test/hosts
  ```

14.Best Practices

14.1. Migrate data from HDFS to JindoFS

This topic describes how to migrate data from Hadoop Distributed File System (HDFS) to JindoFileSystem (JindoFS) that stores data in Object Storage Service (OSS).

Migrate data

- Use Hadoop FS shell commands

  You can run File System (FS) shell commands to migrate a small amount of data:

  ```
  hadoop dfs -cp hdfs://emr-cluster/README.md jfs://emr-jfs/
  ```

  ```
  hadoop dfs -cp oss://oss_bucket/README.md jfs://emr-jfs/
  ```

- Use Hadoop DistCp

  You can use DistCp, a built-in tool of Hadoop, to migrate a large amount of data:

  ```
  hadoop distcp hdfs://emr-cluster/files jfs://emr-jfs/output/
  ```

  ```
  hadoop distcp oss://oss_bucket/files jfs://emr-jfs/output/
  ```

  Note For more information about DistCp parameters, see DistCp Version2 Guide.

Use the cache mode

In cache mode, JindoFS stores data files as objects in OSS without changing the metadata and data. When you access these OSS objects, JindoFS can cache data and metadata of these OSS objects in the local cluster so that you can quickly access them next time. For more information, see Use the cache mode.

14.2. Use MapReduce to process data in JindoFS

This topic describes how to use MapReduce to read and write data in JindoFileSystem (JindoFS).

JindoFS configuration

For example, a namespace named emr-jfs is created with the following configuration:

- jfs.namespaces=emr-jfs
- jfs.namespaces.emr-jfs.oss.uri=oss://oss-bucket/oss-dir
- jfs.namespaces.emr-jfs.mode=block

MapReduce introduction

Generally, Hadoop MapReduce jobs read and write data through Hadoop Distributed File System (HDFS). JindoFS is compatible with HDFS. You can set the input and output directories of MapReduce jobs to directories in JindoFS. In this case, MapReduce jobs can read and write data in JindoFS.

Hadoop MapReduce is a software framework for easily writing applications. Applications based on Hadoop MapReduce can process multi-terabyte datasets in parallel in large clusters that consist of thousands of nodes in a reliable and fault-tolerant manner. A MapReduce job usually splits the input dataset into independent blocks that are processed by map tasks in parallel. The MapReduce framework sorts the output of map tasks and writes the sorted output to reduce tasks. Both the input and the output of the job are stored in a file system. The MapReduce framework is responsible for scheduling tasks, monitoring tasks, and rerunning failed tasks.

Job input and output

In applications, the input and output directories of MapReduce jobs are specified. Map and reduce functions are implemented based on appropriate methods or abstract classes. The Hadoop job client submits MapReduce jobs and their configuration to ResourceManager. Then, ResourceManager schedules tasks. In this case, you can modify the input and output directories of MapReduce jobs to directories in JindoFS to read and write data in JindoFS.

Examples

The following examples show how to modify the input and output directories of a MapReduce job to read and write data in JindoFS.

- Teragen

  In this MapReduce job, use Teragen to generate data in a specified number of rows in the specified directory:

  ```
  hadoop jar /usr/lib/hadoop-current/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar
   teragen <num rows> <output dir>
  ```

  Replace the output directory with a directory in JindoFS to write data to JindoFS:

  ```
  hadoop jar /usr/lib/hadoop-current/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar
   teragen 100000 jfs://emr-jfs/teragen_data_0
  ```

- Terasort

  In this MapReduce job, use Terasort to sort data in the specified input directory and write the sorted data to the specified output directory:

  ```
  hadoop jar /usr/lib/hadoop-current/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar
  terasort <in> <out>
  ```

  Replace the input and output directories with directories in JindoFS to process data in JindoFS:

  ```
  hadoop jar /usr/lib/hadoop-current/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar
  terasort jfs://emr-jfs/teragen_data_0/  jfs://emr-jfs/terasort_data_0
  ```

14.3. Use Hive to query data in JindoFS

Apache Hive is a distributed SQL query engine that is widely used in the Hadoop ecosystem. Hive manages data in databases, tables, and partitions. You can specify the storage location to query data at the storage backend.

JindoFS configuration

For example, a namespace named emr-jfs is created with the following configuration:

- jfs.namespaces=emr-jfs
- jfs.namespaces.emr-jfs.oss.uri=oss://oss-bucket/oss-dir
- jfs.namespaces.emr-jfs.mode=block

Specify the storage location for data warehouses, databases, tables, or partitions

- Specify the storage location for a data warehouse

  The hive-site configuration file contains the hive.metastore.warehouse.dir parameter. This parameter specifies the directory in which a Hive data warehouse stores data. For example, set this parameter to `jfs://emr-jfs/user/hive/warehouse`.

- Specify the storage location for a database

  A Hive database has a storage location. The location is also used as the default storage location of tables in the database. The parameter that specifies the storage location is optional when you create a database. By default, the storage location of a database is the value of the hive.metastore.warehouse.dir parameter in the hive-site file added with the database name. You can execute the following SQL statements to specify the storage location:

  - Set the storage location to a directory in JindoFS when you create a database:

    ```
    CREATE DATABASE database_name
    LOCATION
    'jfs://namespace/database_dir';
    ```

    For example, to create a Hive database named database_on_jindofs and set the storage location to `jfs://emr-jfs/warehouse/database_on_jindofs`, execute the following statement:

    ```
    CREATE DATABASE database_on_jindofs
    LOCATION
    'jfs://emr-jfs/hive/warehouse/database_on_jindofs';
    ```

- Change the storage location of a database to a directory in JindoFS:
  a. Execute the following SHOW CREATE statement to query the storage location of a database:

```
SHOW CREATE DATABASE database_name;
```

  b. View the returned query result. By default, the storage location of a database is that of its data warehouse added with the database name.

```
CREATE DATABASE `database_name`
LOCATION
'hdfs://emr-jfs/user/hive/warehouse/database_name.db'
```

  c. Change the storage location to a directory in JindoFS. This operation does not affect existing tables. If you do not specify the storage location for a new table, the changed location is used as the default storage location of the table.

     For example, to query data in a partition in the jfs_table_name table, execute the following statement:

```
ALTER DATABASE database_name SET LOCATION jfs_path;
```

- Specify the storage location for a table or partition

  Similar to the storage location of a database, the storage location of a table or partition is also specified based on the upper-level storage location. Data in a non-partitioned table is stored in the storage location of the table. Data in a partitioned table is stored in the storage location of respective partitions. You can execute the following SQL statements to specify the storage location:

  - Set the storage location to a directory in JindoFS when you create a table:

```
CREATE [EXTERNAL] TABLE table_name
  [(col_name data_type,...)]
LOCATION 'jfs://emr-jfs/database_dir/table_dir';
```

  - Change the storage location of a table or partition to a directory in JindoFS:
    a. Execute the following DESCRIBE statement to query the storage location of a table or partition:

```
DESCRIBE FORMATTED [PARTITION partition_spec] table_name;
```

    b. Change the storage location to a directory in JindoFS:

```
ALTER TABLE table_name [PARTITION partition_spec] SET LOCATION "jfs_path";
```

    For example, to query data in a partition in the jfs_table_name table, execute the following statement:

```
DESCRIBE FORMATTED jfs_table_name PARTITION (partition_key1=123,partition_key2='xxx
x');
```

Query data in a scratch directory

Hive stores temporary output files and job plans to a scratch directory. You can set the hive.exec.scratchdir parameter in the hive-site configuration file to a directory in JindoFS. You can also run one of the following commands to specify this parameter:

```
bin/hive --hiveconf hive.exec.scratchdir=jfs://emr-jfs/scratch_dir
```

and

```
set hive.exec.scratchdir=jfs://emr-jfs/scratch_dir;
```

14.4. Use Spark to process data in JindoFS

Spark processes data in JindoFileSystem (JindoFS) by using one of the following methods: call methods and use Spark SQL to read data from tables stored in JindoFS.

JindoFS configuration

For example, a namespace named emr-jfs is created with the following configuration:

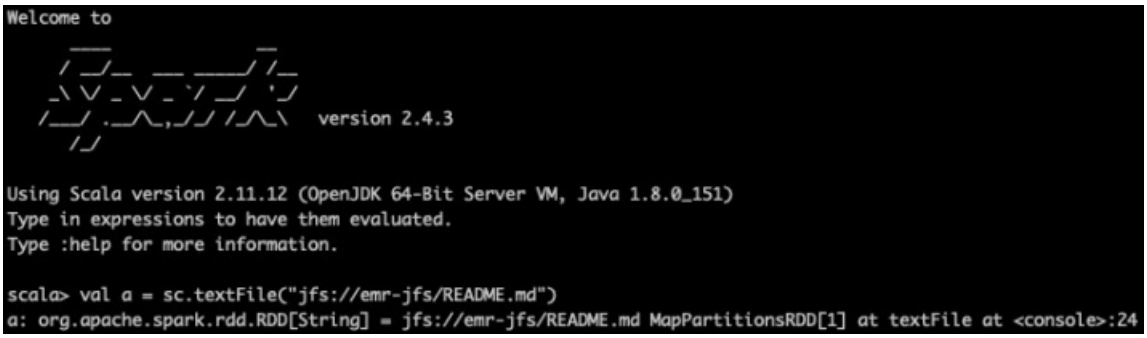
- jfs.namespaces=emr-jfs
- jfs.namespaces.emr-jfs.oss.uri=oss://oss-bucket/oss-dir
- jfs.namespaces.emr-jfs.mode=block

Process data in JindoFS

- Call methods

  The read and write operations performed by Spark in JindoFS are similar to those in other file systems. For example, to access data in JindoFS, use a directory with the jfs prefix in the following Resilient Distributed Dataset (RDD) operation:

  ```
  val a = sc.textFile("jfs://emr-jfs/README.md")
  ```

  To write data to JindoFS, call the following method:

  ```
  scala> a.collect().saveAsTextFile("jfs://emr-jfs/output")
  ```

- Use Spark SQL

  Configure the parameter that sets the storage location to a directory in JindoFS when you create databases, tables, or partitions. For more information, see Use Hive to query data in JindoFS. Then, you can query data from tables stored in JindoFS.

14.5. Use Flink to process data in JindoFS

This topic describes how to use Flink to process data in JindoFileSystem (JindoFS).

JindoFS configuration

For example, a namespace named emr-jfs is created with the following configuration:

- jfs.namespaces=emr-jfs
- jfs.namespaces.emr-jfs.oss.uri=oss://oss-bucket/oss-dir
- jfs.namespaces.emr-jfs.mode=block

Use JindoFS

You can set the input and output directories of Flink jobs to directories in a namespace supported by JindoFS. In this case, Flink jobs can read and write data in JindoFS.

For example, to store job data in Hadoop Distributed File System (HDFS), run the following command:

```
flink run -m yarn-cluster -yD taskmanager.network.memory.fraction=0.4 -yD akka.ask.timeout=
60s -yjm 2048 -ytm 2048 -ys 4 -yn 14 -c xxx.xxx.FlinkWordCount -p 56 XXX.jar --input hdfs:/
//test//large-input-flink --output hdfs:///runjob/test/large-output-flink"
```

To store job data in JindoFS, run the following command:

```
flink run -m yarn-cluster -yD taskmanager.network.memory.fraction=0.4 -yD akka.ask.timeout=
60s -yjm 2048 -ytm 2048 -ys 4 -yn 14 -c xxx.xxx.FlinkWordCount -p 56 XXX.jar --input jfs://
emr-jfs/test/large-input-flink --output jfs://emr-jfs/test/large-output-flink"
```

14.6. Use Impala or Presto to query data in JindoFS

This topic describes how to use Impala or Presto to query data in JindoFileSystem (JindoFS).

JindoFS configuration

For example, a namespace named emr-jfs is created with the following configuration:

- jfs.namespaces=emr-jfs
- jfs.namespaces.emr-jfs.oss.uri=oss://oss-bucket/oss-dir
- jfs.namespaces.emr-jfs.mode=block

Use JindoFS

Currently, Impala, and Presto services for E-MapReduce V3.22.0 and later can read Hive metadata from Hive tables stored in JindoFS.

To store table data in JindoFS, you can set the parameter that specifies the storage location in the CREATE TABLE statement to a directory in JindoFS.

For example, to store table data in Hadoop Distributed File System (HDFS), use the following CREATE TABLE statement:

```
Create external table lineitem (L_ORDERKEY INT, L_PARTKEY INT, L_SUPPKEY INT, L_LINENUMBER
INT, L_QUANTITY DOUBLE, L_EXTENDEDPRICE DOUBLE, L_DISCOUNT DOUBLE, L_TAX DOUBLE, L_RETURNFL
AG STRING, L_LINESTATUS STRING, L_SHIPDATE STRING, L_COMMITDATE STRING, L_RECEIPTDATE STRIN
G, L_SHIPINSTRUCT STRING, L_SHIPMODE STRING, L_COMMENT STRING) ROW FORMAT DELIMITED FIELDS
TERMINATED BY '|' STORED AS TEXTFILE LOCATION 'hdfs:///tpch_impala/lineitem';
```

To store table data in JindoFS, use the following CREATE TABLE statement:

```
Create external table lineitem (L_ORDERKEY INT, L_PARTKEY INT, L_SUPPKEY INT, L_LINENUMBER
INT, L_QUANTITY DOUBLE, L_EXTENDEDPRICE DOUBLE, L_DISCOUNT DOUBLE, L_TAX DOUBLE, L_RETURNFL
AG STRING, L_LINESTATUS STRING, L_SHIPDATE STRING, L_COMMITDATE STRING, L_RECEIPTDATE STRIN
G, L_SHIPINSTRUCT STRING, L_SHIPMODE STRING, L_COMMENT STRING) ROW FORMAT DELIMITED FIELDS
TERMINATED BY '|' STORED AS TEXTFILE LOCATION 'jfs://emr-jfs/tpch_impala/lineitem';
```

14.7. Use JindoFS as the storage backend of HBase

This topic describes how to use JindoFS as the storage backend of HBase.

Background information

HBase is a real-time database in the Hadoop ecosystem and provides high write performance. In E-MapReduce (EMR), you can use JindoFS or Object Storage Service (OSS) as the storage backend of HBase. JindoFS and OSS are more flexible than HDFS.

> **Note** We recommend that you use EMR clusters of V3.36.0 or later.

Configuration of JindoFS

In the example of this topic, a namespace named *emr-jfs* is created for an EMR V3.36.0 cluster, and an OSS bucket named *oss-bucket* is configured for the namespace:

- jfs.namespaces=emr-jfs
- jfs.namespaces.emr-jfs.oss.uri=oss://<oss-bucket>/oss-dir
- jfs.namespaces.emr-jfs.mode=block

Specify a storage path for HBase

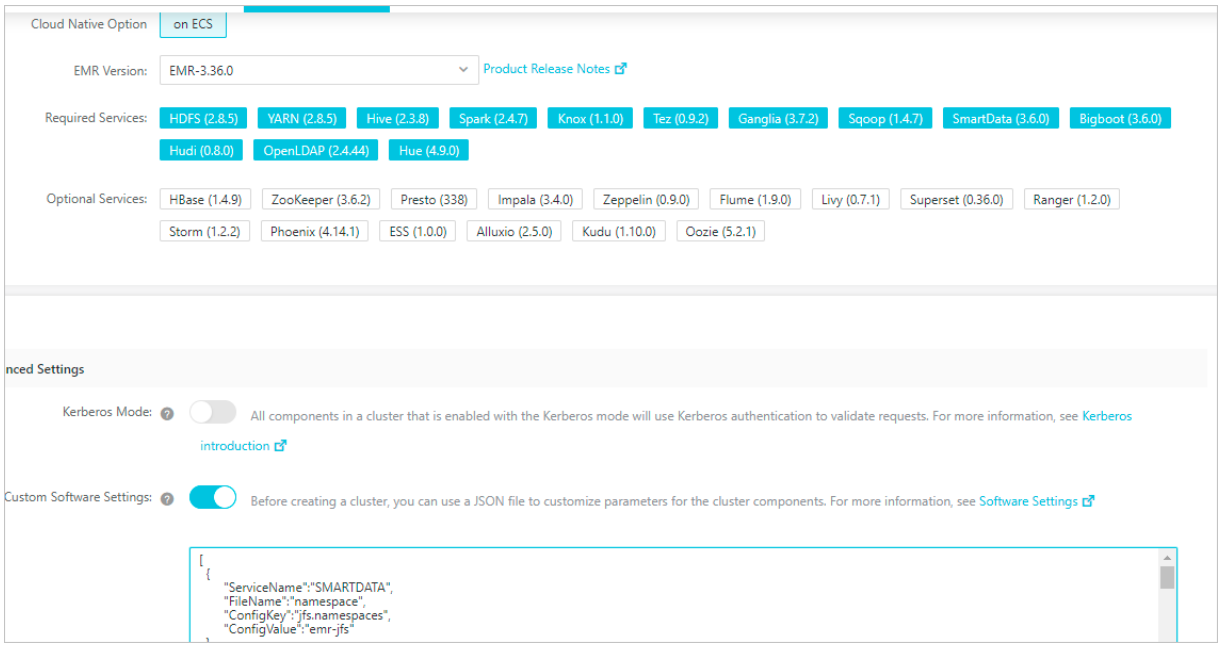
To specify a storage path for HBase, you must set the **hbase.rootdir** parameter in the **hbase-site** configuration file to a JindoFS or OSS path and set the **hbase.wal.dir** parameter in the file to a local HDFS path. This way, the HDFS service of your EMR cluster is used to store write-ahead logging (WAL) files. To release the cluster, you must disable the tables in the cluster and make sure that the updates in WAL files are written to HFile.

Parameter	Description

Parameter	Description
hbase.rootdir	The root directory of HBase in JindoFS or OSS. Set this parameter to *jfs://emr-jfs/hbase-root-dir*. **Note** *emr-jfs* is the namespace that you configured.
hbase.wal.dir	The local HDFS path in which the WAL files of HBase are stored. Value values: • For a high-availability (HA) cluster, set this parameter to *hdfs://emr-cluster/hbase*. • For a non-HA cluster, set this parameter to *hdfs://emr-header-1:9000/hbase*.

Create a cluster

When you create a cluster, turn on Custom Software Settings and add custom software configurations. For more information about how to create a cluster, see Create a cluster.



For example, to use JindoFS as the storage backend of HBase, add the following custom configurations. You must replace `oss_bucket` and the OSS path with the actual OSS bucket name and OSS path.

```
[
  {
        "ServiceName":"SMARTDATA",
        "FileName":"namespace",
        "ConfigKey":"jfs.namespaces",
        "ConfigValue":"emr-jfs"
  },
  {
        "ServiceName":"SMARTDATA",
        "FileName":"namespace",
        "ConfigKey":"jfs.namespaces.emr-jfs.oss.uri",
        "ConfigValue":"oss://oss-bucket/jindoFS"
  },
  {
        "ServiceName":"SMARTDATA",
        "FileName":"namespace",
        "ConfigKey":"jfs.namespaces.emr-jfs.mode",
        "ConfigValue":"block"
  },
  {
        "ServiceName":"HBASE",
        "FileName":"hbase-site",
        "ConfigKey":"hbase.rootdir",
        "ConfigValue":"jfs://emr-jfs/hbase-root-dir"
  },
  {
        "ServiceName":"HBASE",
        "FileName":"hbase-site",
        "ConfigKey":"hbase.wal.dir",
        "ConfigValue":"hdfs://emr-cluster/hbase"
  }
]
```

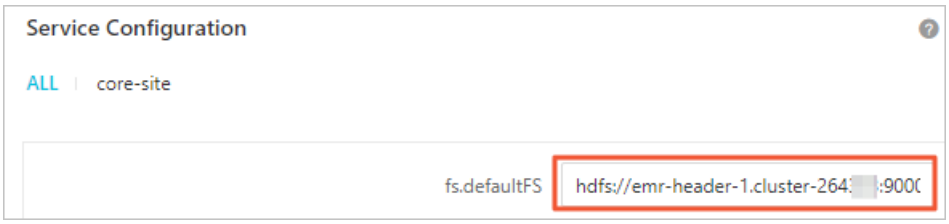
FAQ

- Problem description: When TableSnapshotInputFormat is used in a MapReduce program to read HBase data, the following error message is returned:

```
java.lang.IllegalArgumentException: Wrong FS: jfs://emr-jfs/tmp/..., expected: hdfs://emr
-header-1.cluster-*:9000
    at org.apache.hadoop.fs.FileSYstem.checkPath(FileSystem.java:666)
    at org.apache.hadoop.hbase.regionServer.HRegionFileSystem.createRegionOnFileSystem(HR
egionFileSYstem.java:875)
```

- Cause: An open source MapReduce program that is developed based on HBase has a defect. When the program reads data, it checks whether the data path of HBase is the same as the path that is specified by the fs.defaultFS parameter for HDFS. The error is returned because the paths do not match.
- Solution:
  - You can use ExportSnapshot to read and export HBase data.

- If you use TableSnapshotInputFormat to read HBase data, you must go to the HDFS service page of your cluster in the Alibaba Cloud EMR console and click the **Configure** tab. Then, change the value of the **fs.defaultFS** parameter on the **core-site** tab in the Service Configuration section to a root directory whose prefix is jfs. For example, you can change the value of the **fs.defaultFS** parameter to **jfs://emr-jfs/**, which is configured in the sample code in this topic.



14.8. Store the logs of YARN MapReduce and Spark jobs

This topic describes how to store the logs of MapReduce and Spark jobs to JindoFileSystem (JindoFS) or Object Storage Service (OSS).

Overview

E-MapReduce (EMR) clusters support the pay-as-you-go and subscription billing methods to meet different needs. Pay-as-you-go clusters can be released at any time. Hadoop clusters store logs in the Hadoop Distributed File System (HDFS) by default. If a pay-as-you-go cluster is released, you cannot query job logs of the cluster. In this case, you may have difficulties in troubleshooting job issues. This topic describes how to store the logs of MapReduce and Spark jobs to JindoFS or OSS so that you can query the previous logs of jobs.

Configure JindoFS, YARN Container logs, and Spark History Server

- JindoFS configuration

Configuration file	Parameter	Description	Example
bigboot	jfs.namespaces	The namespace supported by JindoFS. Separate multiple namespaces with commas (,).	emr-jfs
	jfs.namespaces.emr-jfs.oss.uri	The storage backend of the emr-jfs namespace.	*oss://oss-bucket/oss-dir*

Configuration file	Parameter	Description	Example
	jfs.namespaces.test.mode	The storage mode of the emr-jfs namespace.	block **Note** JindoFS supports the block storage mode and cache mode.

- Configuration for YARN Container logs

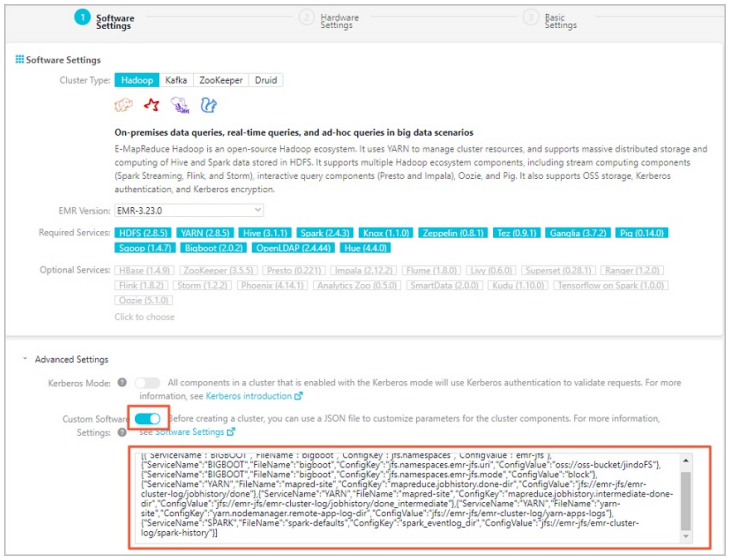
Configuration file	Parameter	Description	Example
yarn-site	yarn.nodemanager.remote-app-log-dir	The directory in which YARN aggregates and stores logs after your application stops running. The log aggregation feature of YARN is enabled by default.	*jfs://emr-jfs/emr-cluster-log/yarn-apps-logs* or *oss://${oss-bucket}/emr-cluster-log/yarn-apps-logs*
mapred-site	mapreduce.jobhistory.done-dir	The directory in which JobHistory stores the logs of Hadoop jobs that are completed.	*jfs://emr-jfs/emr-cluster-log/jobhistory/done* or *oss://${oss-bucket}/emr-cluster-log/jobhistory/done*
	mapreduce.jobhistory.intermediate-done-dir	The directory in which JobHistory stores the logs that are not archived for Hadoop jobs.	*jfs://emr-jfs/emr-cluster-log/jobhistory/done_intermediate* or *oss://${oss-bucket}/emr-cluster-log/jobhistory/done_intermediate*

- Configuration for Spark History Server

Configuration file	Parameter	Description	Example
spark-defaults	spark_eventlog_dir	The directory in which Spark History Server stores the logs of Spark jobs.	*jfs://emr-jfs/emr-cluster-log/spark-history* or *oss://${oss-bucket}/emr-cluster-log/spark-history*

Create a cluster

You can add custom software configurations when you create an EMR cluster, as shown in the following figure.



Custom configuration example

For example, to store logs in JindoFS, use the following custom configuration and replace the OSS bucket and relevant directories:

```
[
  {
        "ServiceName":"BIGBOOT",
        "FileName":"bigboot",
        "ConfigKey":"jfs.namespaces",
        "ConfigValue":"emr-jfs"
  },
  {
        "ServiceName":"BIGBOOT",
        "FileName":"bigboot",
        "ConfigKey":"jfs.namespaces.emr-jfs.oss.uri",
        "ConfigValue":"oss://oss-bucket/jindoFS"
  },
  {
        "ServiceName":"BIGBOOT",
        "FileName":"bigboot",
        "ConfigKey":"jfs.namespaces.emr-jfs.mode",
        "ConfigValue":"block"
  },
  {
        "ServiceName":"YARN",
        "FileName":"mapred-site",
        "ConfigKey":"mapreduce.jobhistory.done-dir",
        "ConfigValue":"jfs://emr-jfs/emr-cluster-log/jobhistory/done"
  },
  {
        "ServiceName":"YARN",
        "FileName":"mapred-site",
        "ConfigKey":"mapreduce.jobhistory.intermediate-done-dir",
        "ConfigValue":"jfs://emr-jfs/emr-cluster-log/jobhistory/done_intermediate"
  },
  {
        "ServiceName":"YARN",
        "FileName":"yarn-site",
        "ConfigKey":"yarn.nodemanager.remote-app-log-dir",
        "ConfigValue":"jfs://emr-jfs/emr-cluster-log/yarn-apps-logs"
  },
  {
        "ServiceName":"SPARK",
        "FileName":"spark-defaults",
        "ConfigKey":"spark_eventlog_dir",
        "ConfigValue":"jfs://emr-jfs/emr-cluster-log/spark-history"
  }
]
```

For example, to store logs in OSS, use the following custom configuration and replace the OSS bucket and relevant directories:

```
[
  {
        "ServiceName":"YARN",
        "FileName":"mapred-site",
        "ConfigKey":"mapreduce.jobhistory.done-dir",
        "ConfigValue":"oss://oss_bucket/emr-cluster-log/jobhistory/done"
  },
  {
        "ServiceName":"YARN",
        "FileName":"mapred-site",
        "ConfigKey":"mapreduce.jobhistory.intermediate-done-dir",
        "ConfigValue":"oss://oss_bucket/emr-cluster-log/jobhistory/done_intermediate"
  },
  {
        "ServiceName":"YARN",
        "FileName":"yarn-site",
        "ConfigKey":"yarn.nodemanager.remote-app-log-dir",
        "ConfigValue":"oss://oss_bucket/emr-cluster-log/yarn-apps-logs"
  },
  {
        "ServiceName":"SPARK",
        "FileName":"spark-defaults",
        "ConfigKey":"spark_eventlog_dir",
        "ConfigValue":"oss://oss_bucket/emr-cluster-log/spark-history"
   }
]
```

14.9. Import data from Kafka to JindoFS

When you collect logs and aggregate monitoring data, you can use Apache Kafka to process offline data and streaming data and analyze data in real time. This topic describes how to import data from Kafka to JindoFileSystem (JindoFS).

Import methods

- Use Flume

  Apache Flume is a system used for moving data into Hadoop Distributed File System (HDFS). We recommend that you use Flume to import data from Kafka to JindoFS. To implement this feature, set the jfs.type parameter to hdfs and the jfs.hdfs.path parameter to a directory in JindoFS:

  ```
  a1.sinks = emr-jfs
  ...
  a1.sinks.emr-jfs.type = hdfs
  a1.sinks.emr-jfs.hdfs.path = jfs://emr-jfs/kafka/%{topic}/%y-%m-%d
  a1.sinks.emr-jfs.hdfs.rollInterval = 10
  a1.sinks.emr-jfs.hdfs.rollSize = 0
  a1.sinks.emr-jfs.hdfs.rollCount = 0
  a1.sinks.emr-jfs.hdfs.fileType = DataStream
  ```

- Call a Kafka API

Some engines like MapReduce and Spark call a Kafka API to export data from Kafka to HDFS. You only need to reference HDFS and set the export path to a directory in JindoFS.

- Use Kafka HDFS Connector

  You can also use Kafka HDFS Connector to export data from Kafka to HDFS by setting the sink export path to a directory in JindoFS.

14.10. Access JindoFS across clusters

E-MapReduce (EMR) clusters are independent of each other. The client of a cluster can access only the namespaces that are configured in the cluster. If you have multiple clusters, you can configure JindoFS parameters to implement cross-cluster access to JindoFS. This topic describes how to implement cross-cluster access to JindoFS. Cluster A and Cluster B are used as an example.

Prerequisites

- EMR clusters A and B of the 3.30.0 version or later are deployed in the same VPC. For information about how to create a cluster, see Create a cluster.
- The */etc/hosts* file is configured to synchronize the data of the hosts of all nodes in Cluster B to Cluster A.

Modify configurations

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Go to the **client** tab.
   i. Click the **Configure** tab.
   ii. In the **Service Configuration** section, click the **client** tab.
3. Modify configurations to enable cross-cluster access.

   Configure Cluster A based on the settings of the **namespace.backend.type** parameter specified in Cluster B.

   - If **namespace.backend.type** is set to **rocksdb**, perform the following operations:
     a. Click **Custom Configuration** in the upper-right corner.
     b. In the **Add Configuration Item** dialog box, set Key to **client.external.namespace.rpc.addresses** and Value to `emr-header-1.<cluster-B>:8101`, and click **OK**.

     > **Note** <cluster-B> is the ID of Cluster B.

   - If **namespace.backend.type** is set to **raft**, perform the following operations:

a. Click **Custom Configuration** in the upper-right corner.

b. In the **Add Configuration Item** dialog box, set Key to **client.external.namespace.rpc.addresses** and Value to `emr-header-1.<cluster-B>:8101,emr-header-2.<cluster-B>:8101,emr-header-3.<cluster-B>:8101`, and click **OK**.

4. Save the configurations.

   i. In the upper-right corner of the Service Configuration section, click **Save**.

   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.

   iii. Click **OK**.

Associate multiple clusters

To associate multiple clusters, use **client.external.namespace.rpc.addresses** to configure remote addresses for these clusters. Separate the addresses with semicolons (;).

For example, the address of Cluster B is `emr-header-1.<cluster-B>:8101`, and namespace.backend.type is set to rocksdb for Cluster B. The address of Cluster C is `emr-header-1.<cluster-C>:8101,emr-header-2.<cluster-C>:8101,emr-header-3.<cluster-C>:8101`, and namespace.backend.type is set to raft for Cluster C. To associate Cluster B and Cluster C with Cluster A, set Key to client.external.namespace.rpc.addresses and Value to emr-header-1.<cluster-B>:8101;emr-header-1.<cluster-C>:8101,emr-header-2.<cluster-C>:8101,emr-header-3.<cluster-C>:8101 in the Add Configuration Item dialog box for Cluster A.

14.11. Change the HDFS client path for the Jindo system

In SmartData 3.1.X, you can change the HDFS client path for the Jindo system. This way, you do not need to change the path when you migrate cluster data. For example, after you change the HDFS client path for the Jindo system to an Object Storage Service (OSS) path, you can access the data that you migrated from HDFS to OSS. You do not need to change the data storage path that is defined in the business logic.

Limits

This feature is supported only in Hadoop 2.x.

Enable the path change feature

1. Go to the SmartData service.

   i.

   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.

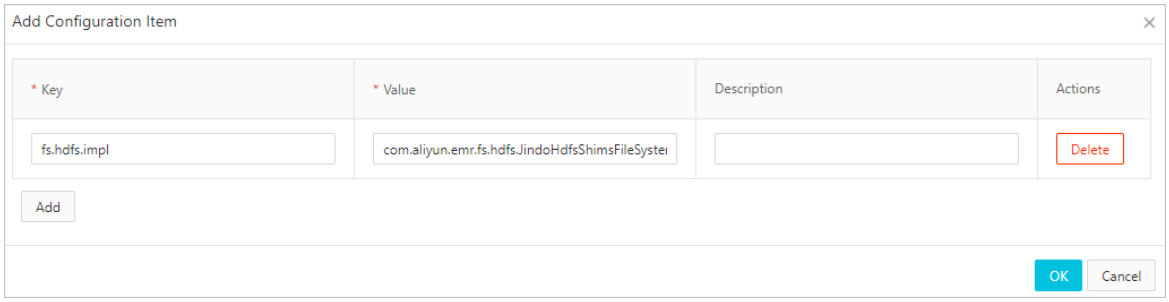
   iii. Click the **Cluster Management** tab.

   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.

   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.

2. Click the **Configure** tab.

3. In the **Service Configuration** section, click the **smartdata-site** tab.
4. In the **Service Configuration** section, click **Custom Configuration** in the upper-right corner.
5. In the **Add Configuration Item** dialog box, add a configuration item whose **Key** is set to fs.hdfs.impl and **Value** is set to com.aliyun.emr.fs.hdfs.JindoHdfsShimsFileSystem.



6. Click **OK**.
7. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Change the HDFS client path for the Jindo system

1. Go to the SmartData service.
   i.
   ii. In the top navigation bar, select the region where your cluster resides. Select the resource group as required. By default, all resources of the account appear.
   iii. Click the **Cluster Management** tab.
   iv. On the **Cluster Management** page that appears, find the target cluster and click **Details** in the Actions column.
   v. In the left-side navigation pane, click **Cluster Service** and then **SmartData**.
2. Click the **Configure** tab.
3. In the **Service Configuration** section, click the **smartdata-site** tab.
4. In the **Service Configuration** section, click **Custom Configuration** in the upper-right corner.
5. In the **Add Configuration Item** dialog box, add the configuration items that are described in the following table.

Parameter	Description	Value

Parameter	Description	Value
fs.jindo.shim.path-rewrite.<RULE-NAME>.source	The HDFS client path that you want to change.	○ High availability (HA) cluster hdfs://emr-cluster/<osspath> ○ Non-HA cluster hdfs://<your_hostname>:9000/<osspath> **Note** To obtain the hostname that you can use to replace <your_hostname> in the parameter value, you can log on to the master node of your EMR cluster in SSH mode and run the `hostname` command. For more information, see Log on to a cluster.
fs.jindo.shim.path-rewrite.<RULE-NAME>.target	The path you actually access.	oss://<your_bucket>/<testpath>

You must specify RULE-NAME.

6. Click **OK**.
7. Save the configuration.
   i. In the upper-right corner of the Service Configuration section, click **Save**.
   ii. In the **Confirm Changes** dialog box, specify Description and turn on **Auto-update Configuration**.
   iii. Click **OK**.

Example

For an HA cluster, you can access data that is stored in *oss://jindo-bucket/<testpath>* instead of *hdfs://emr-cluster/osspath* after you add the parameters on the **smartdata-site** tab. The following table describes the parameters.

Parameter	Value
fs.jindo.shim.path-rewrite.testrule.source	hdfs://emr-cluster/osspath
fs.jindo.shim.path-rewrite.testrule.target	oss://jindo-bucket/<testpath>

To view the path change, you can log on to the master node of your EMR cluster in SSH mode and run the following command:

```
hadoop fs -ls /
```

The following output shows that osspath is mounted on the root directory.

```
[root@emr-header-1 ~]# hadoop fs -ls /
Found 7 items
drwxr-xr-x   - hadoop    hadoop          0 2020-11-24 21:12 /apps
drwxrwxrwx   - flowagent hadoop          0 2020-11-24 21:11 /emr-flow
drwxr-x--x   - root      hadoop          0 2020-11-24 21:12 /emr-sparksql-udf
drwxrwxrwx   -                           0 2020-11-25 16:58 /osspath
drwxr-x--x   - hadoop    hadoop          0 2020-11-25 16:56 /spark-history
drwxrwxrwx   - root      hadoop          0 2020-11-25 16:55 /tmp
drwxr-x--t   - hadoop    hadoop          0 2020-11-24 21:14 /user
```

15.SmartData FAQ

This topic provides answers to some frequently asked questions about SmartData.

Basic concepts

- What is JindoFS?
- Why do I need to use JindoFS when OSS is available for me?
- What storage modes does JindoFS support? What are the use scenarios of each mode?
- What are the differences between the client-only mode and cache mode?
- What are the differences between the cache mode and block storage mode?
- Can JindoFS in block storage mode read data over the OSS API?
- OSS does not support rename operations. Does JindoFS support rename operations?
- Does JindoFS provide high rename performance?
- Does JindoFS support magic committers, such as Hadoop S3A Magic Committer?
- How does JindoFS support a directory that contains millions of files or even tens of millions of files?
- How does JindoFS ensure data reliability?
- Does JindoFS provide strong consistency of operations on files and directories?
- Does JindoFS support atomicity of operations on files and directories?
- How does JindoFS in block storage mode ensure high availability?
- JindoFS in block storage mode stores file metadata in each cluster. How do I recover the metadata of a stopped cluster to a newly created cluster?
- Why do I need to use JindoFS in block storage mode while I can use HDFS in EMR?
- What are the technical differences between JindoFS and Alluxio? What are the advantages of JindoFS over Alluxio?
- Are JindoFS and OSS more cost-effective than HDFS?
- Open source Hadoop can also access OSS. What are the advantages of JindoFS over open source Hadoop?
- Does JindoFS provide Fuse? What are the advantages of Fuse provided by JindoFS over Fuse provided by OSS?
- What is the relationship between SmartData and JindoFS in EMR?
- What is the relationship between Bigboot and JindoFS in EMR?

Open source ecosystems

- What open source components does JindoFS support?
- Does JindoFS deliver a high data throughput that is required by Spark or Hive large-scale data processing?
- Does JindoFS provide high write performance? How do I deal with the bottleneck encountered when I use Flume or Kafka to write data?
- Does JindoFS support Realtime Compute for Apache Flink?
- Can I use Presto for interactive analysis of data in OSS based on JindoFS?
- How do I migrate data in HDFS to JindoFS?
- Can I query data in OSS by using JindoFS when I use Impala?

- Can I store data in OSS by using JindoFS when I use Delta Lake, Hudi, or Iceberg?
- Does JindoFS support machine learning training for data stored in OSS?
- How does JindoFS support machine learning training for data in MaxCompute?
- How does JindoFS support machine learning training for data in Hive?
- What do I do if Bigboot logs occupy too much space?

Upgrade and data migration

- How do I migrate data in HDFS to JindoFS?
- The version of my EMR cluster does not support JindoFS. How do I use JindoFS?
- Which Hadoop versions and Hadoop vendors does JindoFS support?
- Can I use JindoFS in a self-managed cluster that is deployed on Elastic Compute Service (ECS) instances?
- Can I use JindoFS in Alibaba Cloud Container Service for Kubernetes (ACK)?
- Is JindoFS bound to EMR?
- Can I use JindoFS in a Hadoop cluster of a data center?

Known version-specific issues

- What do I do if the disk space occupied by the /opt/bignode directory is extremely large and continues to increase?
- What do I do if a data error occurs when I read cached data?

OSS

- How do I view the data volume in JindoFS?
- What do I do if the data volume queried in JindoFS is inconsistent with the data volume displayed in the OSS console?
- In which scenarios do I need to enable OSS bucket versioning?
- What is the impact of OSS bucket versioning on EMR and JindoFS?
- The Archive storage class of OSS helps reduce storage costs. Does JindoFS support the Archive storage class?

Security

- Is it possible for my AccessKey pair to be disclosed when I use JindoFS?
- What is AccessKey-free access?
- How do I grant different users different permissions when I use the AccessKey-free access feature?
- How do I use different AccessKey pairs to access different OSS buckets from JindoFS?
- What do I do if I want to use a self-managed cluster in a data center but do not want to configure an AccessKey pair on the cluster nodes?
- Does JindoFS support the AuditLog feature?
- Does JindoFS support the Ranger service?

What is JindoFS?

JindoFS is a Hadoop file system provided by Alibaba Cloud E-MapReduce (EMR). JindoFS provides multi-level encapsulation and optimization features to enable Hadoop and Spark to better use Alibaba Cloud Object Storage Service (OSS).

JindoFS is adapted to OSS and allows you to use JindoFS SDK to access OSS. You can use JindoFS in cache mode to cache OSS data in distributed mode. You can also use JindoFS in block storage mode together with some other advanced features of JindoFS to meet your business requirements in some special scenarios.

Why do I need to use JindoFS when OSS is available for me?

OSS is an object storage system that provides object-based RESTful APIs and SDKs for various programming languages. JindoFS provides Hadoop Compatible File System (HCFS) interfaces that you can use to access OSS. JindoFS also supports cache-based acceleration and allows you to customize advanced optimization features. Hadoop and Spark components rely on abstract HCFS interfaces. Therefore, JindoFS is required.

What storage modes does JindoFS support? What are the use scenarios of each mode?

You can use JindoFS in client-only, cache, or block storage mode. In client-only mode, JindoFS SDK (*jindo-sdk_xxx.jar*) is used.

Use scenarios of each mode:

- Client-only mode: If you do not need to change the manner in which objects are organized in OSS, use this mode. To use JindoFS in this mode, you need only to upload the JAR package of JindoFS SDK to the classpath directory of your big data computing component.
- Cache mode: If computing performance is limited by IOPS and storage throughput, use this mode. In cache mode, you can add disks or expand the capacity of disks on the core nodes of your EMR cluster and enable data caching.
- Block storage mode: This mode is suitable for some special scenarios. For example, if you have high requirements for the performance of metadata-related operations and the consistency of metadata, use this mode.

What are the differences between the client-only mode and cache mode?

Both of the modes are fully compatible with OSS. Both of the modes allow you to read data from and write data to OSS by using the API and SDK provided by OSS.

To use JindoFS in cache mode, you must deploy and configure the Jindo distributed caching service and enable data caching. If you use JindoFS in client-only mode, you do not need to perform these operations. If the caching service fails, the system switches to the client-only mode and directly reads data from or writes data to OSS.

What are the differences between the cache mode and block storage mode?

In block storage mode, you can manage the metadata of files, organize data blocks, and use OSS as disks in a similar way to HDFS. In block storage mode, you must use the JindoFS SDK client to read data from or write data to OSS.

In cache mode, JindoFS is compatible with OSS. Therefore, you can directly read data from and write data to OSS. For example, in cache mode, after you write a large file to OSS, you can find the file in an OSS directory in the OSS console. In block storage mode, you can find many file blocks, which you can combine into a large file only by using the JindoFS SDK client.

How can the consistency between data cached in JindoFS and data stored in OSS be ensured after data in OSS is updated?

When JindoFS reads data from the OSS object, JindoFS checks the metadata and MD5 information of the OSS object and compares the information with the cached data. If data in the OSS object is modified, overwritten, or deleted, JindoFS directly reads data from OSS instead of from the cache and updates data in the cache.

Can JindoFS in block storage mode read data over the OSS API?

No, JindoFS in block storage mode cannot read data over the OSS API. JindoFS in block storage mode can read data only by using the JindoFS SDK client.

OSS does not support rename operations. Does JindoFS support rename operations?

Yes, JindoFS supports rename operations. You can use JindoFS to rename files and directories because JindoFS supports HCFS interfaces.

You cannot directly rename directories or files in OSS. JindoFS allows you to perform rename operations by simulating a file system. To rename an object, you must first copy the object to a different location and then delete the original object.

Does JindoFS provide high rename performance?

The rename performance of JindoFS is better than that of open source solutions. OSS provides the Fast Copy feature to optimize the performance of copying large files. This enables JindoFS to rename a file much faster than the open source solutions. Each directory may contain a large number of files. JindoFS has fully optimized the parallelism of rename operations. This ensures that JindoFS can rename a directory much faster than the open source solutions.

Does JindoFS support magic committers, such as Hadoop S3A Magic Committer?

JindoFS supports magic committers that do not require rename operations.

How does JindoFS support a directory that contains millions of files or even tens of millions of files?

JindoFS supports parallel access to and memory optimization for large directories. Parallel access to a large directory does not cause out of memory (OOM) errors or task suspension.

How does JindoFS ensure data reliability?

Data reliability is ensured by OSS. JindoFS stores data in OSS no matter which storage mode is used, and local disks are used only to cache data.

Does JindoFS provide strong consistency of operations on files and directories?

Yes, JindoFS in block storage mode implements HDFS semantics, which supports strong consistency.

Does JindoFS support atomicity of operations on files and directories?

JindoFS in cache mode does not support atomicity. JindoFS in cache mode is compatible with OSS and is limited by the storage features of OSS. Therefore, JindoFS in cache mode does not support atomicity of cross-object operations. For example, if you perform a rename operation, at least a source object and a destination object are involved. If you perform the operation on a directory, more objects are involved.

JindoFS in block storage mode strictly implements HDFS semantics and supports atomicity of various operations such as rename operations.

How does JindoFS in block storage mode ensure high availability?

JindoFS in block storage mode allows you to deploy multiple Jindo Namespace Service nodes based on the distributed consistency protocol Raft. In addition, the metadata of a cluster can be asynchronously synchronized to an Alibaba Cloud Tablestore instance.

JindoFS in block storage mode stores file metadata in each cluster. How do I recover the metadata of a stopped cluster to a newly created cluster?

JindoFS in block storage mode can asynchronously synchronize the metadata of a cluster to an Alibaba Cloud Tablestore instance. If you stop a cluster after the cluster stops updating data and synchronizes all updates to Tablestore, all metadata of the cluster is stored in Tablestore. When you create a cluster, you can restore data from OSS and restore metadata from Tablestore to the cluster.

> **Note** You must take note of the version compatibility when you create a cluster. For example, you can recover the metadata of an EMR V2.7.X cluster to another EMR V2.7.X cluster. However, the same operation between an EMR V2.6.X cluster and an EMR V2.7.X cluster may fail. If such a failure occurs, we recommend that you use the Jindo DistCp tool to synchronize data to OSS.

Why do I need to use JindoFS in block storage mode while I can use HDFS in EMR?

The technical architecture and features of JindoFS in block storage mode are similar to those of HDFS. Both of JindoFS in block storage mode and HDFS can be used to manage file metadata and organize data, and both of them deliver strong consistency.

However, JindoFS in block storage mode has the following advantages over HDFS: JindoFS in block storage mode backs up data to OSS, supports auto scaling, is cost-effective, and does not require disk maintenance.

What are the technical differences between JindoFS and Alluxio? What are the advantages of JindoFS over Alluxio?

Item	JindoFS	Alluxio

Item	JindoFS	Alluxio
Similarities	The technical architecture of JindoFS in cache mode is similar to that of Alluxio. Both JindoFS in cache mode and Alluxio support cache-based acceleration for OSS data and work on one master node and multiple worker nodes. The master node is used to maintain the location information of cache blocks, and the worker nodes are used to manage cache blocks and read and write data.	
Differences	To access an *oss://* path by using JindoFS, you need only to enable data caching.	To access an oss:// path by using Alluxio, you must mount an OSS bucket to a namespace and use *alluxio://* in a command to access the OSS path.
	JindoFS supports only OSS data sources, and all performance optimizations are dedicated to OSS.	Alluxio supports many types of data sources and allows you to mount multiple data sources to the same namespace.
	JindoFS provides the SDK-based client-only mode to support access from various open source engines to OSS.	N/A.

Are JindoFS and OSS more cost-effective than HDFS?

HDFS does not support auto scaling. In some cases, your storage space may be insufficient or purchased space may be wasted.

OSS allows you to store large volumes of data, supports auto scaling, and allows you to archive cold data at low costs. JindoFS supports the tiered storage of hot and cold data and data archiving based on OSS. This way, JindoFS can help you reduce costs.

Open source Hadoop can also access OSS. What are the advantages of JindoFS over open source Hadoop?

Access to OSS from open source Hadoop is limited by the community. Only some basic features are supported.

JindoFS has the following advantages:

- More comprehensive: Various open source engines can connect to OSS.
- More active: JindoFS is updated or upgraded to support new features provided by OSS.
- More advanced: JindoFS supports advanced cache-based acceleration, and JindoFS in block storage mode can meet your requirements for advanced customization.
- More performant: The core code of JindoFS is developed based on C++ native code, which ensures the higher performance of basic operations.

Does JindoFS provide Fuse? What are the advantages of Fuse provided by JindoFS over Fuse provided by OSS?

Yes, JindoFS provides Fuse. Fuse provided by JindoFS can fully use the capabilities enabled by the cache mode and block storage mode.

What open source components does JindoFS support?

JindoFS supports components that read and write data over HCFS interfaces. The components include Hadoop, Hive, Spark, Flink, Presto, HBase, Impala, Druid, Kafka, and Flume.

Does JindoFS deliver a high data throughput that is required by Spark or Hive large-scale data processing?

JindoFS supports asynchronous concurrent read operations based on the high concurrency capability of OSS. JindoFS allows you to use the concurrent multipart upload feature of OSS to split a large object into multiple parts and upload them concurrently. JindoFS has large advantages over open source Hadoop in terms of read and write throughput.

Both JindoFS in cache mode and JindoFS in block storage mode can cache data in local disks or memory of clusters. This feature significantly accelerates the queries of new data and data that is repeatedly read. Under the same cluster conditions, JindoFS delivers an equivalent or even higher throughput for Spark or Hive large-scale data processing than HDFS.

Does JindoFS provide high write performance? How do I deal with the bottleneck encountered when I use Flume or Kafka to write data?

HDFS needs to write three replicas. JindoFS does not need to write a backup to OSS. Therefore, in most cases, JindoFS has higher write performance than HDFS.

If you encounter a bottleneck when you use Flume or Kafka to write data, submit a ticket.

Does JindoFS support Realtime Compute for Apache Flink?

Yes, JindoFS supports Realtime Compute for Apache Flink. JindoFS allows you to use Realtime Compute for Apache Flink to read data from OSS and use the checkpoint and sink mechanisms of Realtime Compute for Apache Flink to write data to OSS. Exactly-Once semantics are supported.

Can I use Presto for interactive analysis of data in OSS based on JindoFS?

Yes, both JindoFS in cache mode and JindoFS in block storage mode allow you to use Presto to perform interactive analysis of data in OSS.

Can I query data in OSS by using JindoFS when I use Impala?

Yes, Impala 3.4 and later support JindoFS. You can use JindoFS to read and write data.

Can I store data in OSS by using JindoFS when I use Delta Lake, Hudi, or Iceberg?

Yes, you can store data in OSS by using JindoFS when you use Delta Lake, Hudi, or Iceberg.

Does JindoFS support machine learning training for data stored in OSS?

Yes, JindoFS supports machine learning training for data stored in OSS. You can use JindoFS in cache mode to preload data in OSS to the memory or an SSD of your cluster. This way, a training engine can use JindoFuse to read the data from the memory or SSD.

How does JindoFS support machine learning training for data in MaxCompute?

You can use one of the following methods:

- MaxCompute jobs write data to OSS by using external tables created in MaxCompute. Then, the training cluster uses JindoFS in cache mode and JindoFuse to load training data.
- Use JindoTable to pull data from MaxCompute and write the data to JindoFS in cache mode. Then, use JindoFuse to load training data.

How does JindoFS support machine learning training for data in Hive?

The methods are similar to those for data in MaxCompute. For more information, see How does JindoFS support machine learning training for data in MaxCompute?.

How do I migrate data in HDFS to JindoFS?

You can use Jindo DistCp to synchronize data in HDFS to JindoFS or OSS. Jindo DistCp has higher performance than Hadoop DistCp and can archive data in OSS.

The version of my EMR cluster does not support JindoFS. How do I use JindoFS?

If the cluster scale is small, we recommend that you create a cluster of a version that supports JindoFS and use JindoFS in the new cluster. If the cluster scale is large, submit a ticket.

Which Hadoop versions and Hadoop vendors does JindoFS support?

The JindoFS SDK provides the OSS adaptation feature and supports minor versions later than Hadoop 2.7 and Hadoop 3.X.

Hortonworks Data Platform (HDP) developed by Hortonworks and Cloudera's Distribution Including Apache Hadoop (CDH) developed by Cloudera can be used, but a conflict may occur. To use HDP or CDH, you must set `fs.oss.impl` to JindoOssFileSystem.

Can I use JindoFS in a self-managed cluster that is deployed on Elastic Compute Service (ECS) instances?

Yes, you can use JindoFS in a self-managed cluster that is deployed on ECS instances. You need only to download JindoFS SDK and manually deploy JindoFS in the cluster. If you want to use JindoFS in cache mode or block storage mode, we recommend that you log on to the EMR console and use an EMR cluster.

Can I use JindoFS in Alibaba Cloud Container Service for Kubernetes (ACK)?

Yes, you can use JindoFS in Alibaba Cloud ACK.

Is JindoFS bound to EMR?

No, JindoFS is not bound to EMR. JindoFS provides standard HCFS interfaces and is fully compatible with open source ecosystems.

Can I use JindoFS in a Hadoop cluster of a data center?

Yes, you can use JindoFS in a Hadoop cluster of a data center. You can download the open source JindoFS SDK and deploy it by following instructions in the documentation. If the cluster is incompatible with JindoFS, submit a ticket.

How do I view the data volume in JindoFS?

You can run the following command to view the data volume:

```
hadoop dfs -du/count
```

What do I do if the data volume queried in JindoFS is inconsistent with the data volume displayed in the OSS console?

In most cases, this issue occurs because the OSS bucket versioning feature is enabled and the data of historical versions is not cleared. Another possible cause is that the version of JindoFS in block storage mode is too old, which results in memory leak and residual data. You can submit a ticket to solve this issue.

In which scenarios do I need to enable OSS bucket versioning?

We recommend that you enable OSS bucket versioning for important data. OSS bucket versioning ensures that data is not lost if you delete data by mistake.

What is the impact of OSS bucket versioning on EMR and JindoFS?

We recommend that you do not enable OSS bucket versioning for intermediate result data of Hive or Spark and data that is frequently modified. OSS bucket versioning affects computing performance.

The Archive storage class of OSS helps reduce storage costs. Does JindoFS support the Archive storage class?

Yes, JindoFS supports the Archive storage class. JindoFS in block storage mode provides specific storage policies to support the Archive storage class.

Is it possible for my AccessKey pair to be disclosed when I use JindoFS?

JindoFS allows you to configure and use an AccessKey pair on a cluster. However, the AccessKey pair may be disclosed. If a node of an EMR cluster or in an ECS environment is bound to an ECS role, you can access the node based on your permissions instead of using an AccessKey pair.

What is AccessKey-free access?

EMR clusters support AccessKey-free access. This feature allows you to obtain an Alibaba Cloud Security Token Service (STS) token and use this token to access Alibaba Cloud resources, such as OSS buckets.

How do I grant different users different permissions when I use the AccessKey-free access feature?

AccessKey-free access is not suitable in some scenarios.

You can use one of the following methods to grant permissions to different users:

- Use RAM to implement access control. If this method is used, RAM users are used to access OSS.
- Use JindoFS to implement access control. If this method is used, Ranger is used to grant permissions.

> **Notice** JindoFS supports access control only on namespaces.

How do I use different AccessKey pairs to access different OSS buckets from JindoFS?

You can use multiple namespaces in JindoFS and configure the information about a single OSS bucket and the corresponding AccessKey pair for each namespace.

What do I do if I want to use a self-managed cluster in a data center but do not want to configure an AccessKey pair on the cluster nodes?

You can use a Hadoop credential provider. For more information, see Use a credential provider.

Does JindoFS support the AuditLog feature?

Yes, JindoFS supports the AuditLog feature. JindoFS allows you to configure multiple namespaces and enable the AuditLog feature on each namespace. The AuditLog feature is disabled by default.

Does JindoFS support the Ranger service?

Yes, JindoFS supports the Ranger service. JindoFS allows you to configure multiple namespaces and enable the Ranger service on each namespace. The Ranger service is disabled by default.

What is the relationship between SmartData and JindoFS in EMR?

SmartData is an EMR component, which provides the JindoFS service.

What is the relationship between Bigboot and JindoFS in EMR?

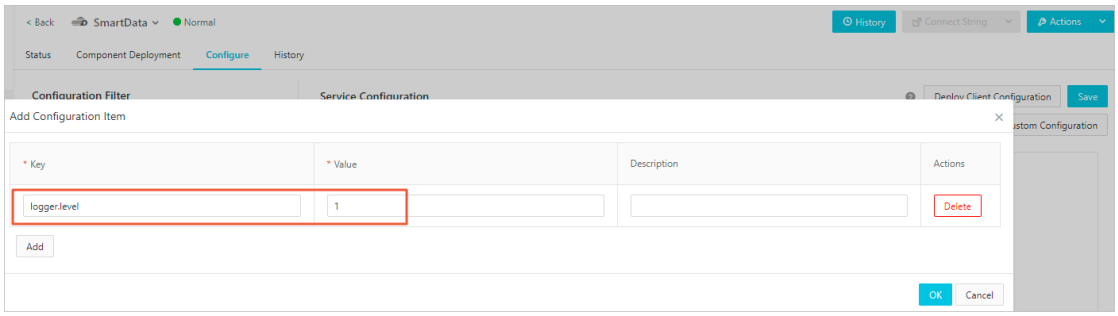
Bigboot is the infrastructure of the SmartData component. Bigboot provides features such as millisecond-level process monitoring and log cleaning for the services contained in the component.

What do I do if Bigboot logs occupy too much space?

This issue may occur in a minor version earlier than EMR V3.36.1 or EMR V5.2.1. You must manually delete some log files. You can perform the following steps to change the log level from INFO to WARN. This way, the number of displayed logs is reduced.

1. Add a configuration item in the EMR console.
   i. On the **Configure** tab of the SmartData service page, click the **namespace** tab.
   ii. Click **Custom Configuration** in the upper-right corner.

iii. In the **Add Configuration Item** dialog box, add the logger.level parameter and set it to 1.



> **Note** The default value is 0, which indicates INFO. The value 1 indicates WARN.

iv. Click **OK**.

2. Save the configuration.
   i. Click **Save** in the upper-right corner.
   ii. In the **Confirm Changes** dialog box, specify **Description** and click **OK**.
3. Restart Jindo Namespace Service.
   i. Choose **Actions > Restart Jindo Namespace Service** in the upper-right corner.
   ii. In the **Cluster Activities** dialog box, specify **Description** and click **OK**.
   iii. In the **Confirm** message, click **OK**.

What do I do if the disk space occupied by the /opt/bignode directory is extremely large and continues to increase?

- Problem description: In most cases, the */opt/bignode* directory occupies several gigabytes of disk space. However, in SmartData 2.X, the disk space occupied by the */opt/bignode* directory is extremely large and continues to increase.
- Possible cause: The monitoring process unexpectedly exits. As a result, the Jindo Storage Service process is deemed abnormal and is repeatedly started.
- Solution: Perform the following steps to troubleshoot the issue:

  Find the node on which the issue occurs and run the following command as the root user to view information about the processes:

  ```
  ps -aux | grep b2-storageservice | grep -v grep
  ```

  In most cases, the monitoring process xxxx/services/b2-storageservice.spec and the Jindo Storage Service process xxxx/bin/b2-storageservice are started. If no information about the xxxx/services/b2-storageservice.spec process is displayed, the process unexpectedly exits and causes the issue. In this case, you must run the following command to manually kill the xxxx/bin/b2-storageservice process:

  ```
  kill -9 <PID of b2-storageservice>
  ```

  > **Note** <PID of b2-storageservice> specifies the ID of the xxxx/bin/b2-storageservice process. The process ID is obtained when you view process information.

After you run the preceding command, the monitoring process and the Jindo Storage Service process are automatically recovered. The disk space occupied by the /opt/bignode directory becomes normal. If the issue persists after you perform the preceding steps, submit a ticket.

What do I do if a data error occurs when I read cached data?

- Problem description: In SmartData 3.0 to SmartData 3.7, if JindoFS in block storage mode or cache mode is used and data caching is enabled, data can be generated, but data is contaminated during read operations. (In block storage mode, data caching is enabled by default.) For example, a data format error is reported when you read data from ORC or Parquet files, or an HFile format error is reported when you read HBase data.
- Possible cause: Invalid cached data blocks are read due to a known defect in the program that you use to read cached data blocks.
- Solution: Use one of the following methods to implement a workaround. Then, contact EMR technical support engineers to update the program.

  Use one of the following methods based on your business requirements to avoid the continuous impact of the issue on your business:

  - Method 1: Disable data caching.

    In most cases, the defect is caused by data caching. You can disable data caching to implement a workaround. To disable data caching, perform the following steps: Go to the SmartData service page and click the Configure tab. Then, click the client tab in the Service Configuration section and configure related parameters. If you use JindoFS in block storage mode, set jfs.data-cache.enable to false. If you use JindoFS in cache mode, set jfs.cache.data-cache.enable to false.

  - Method 2: Clear cached data of a file in which an error occurs.

    If Method 1 is not applicable due to considerations such as performance, you can use this method.

    a. Run the following command to submit a cache clearing task:

    ```
    jindo jfs -uncache <Full path of the file in which an error occurs>
    ```

    b. Run the following command to view the task status. Make sure that the cached data of the file is cleared.

    ```
    jindo jfs -status <Full path of the file in which an error occurs>
    ```

    c. On the client tab in the Service Configuration section of the Configure tab on the SmartData service page, add a custom parameter whose name is storage.compaction.enable and value is false.

    For more information, see Add parameters.

    d. Restart Jindo Storage Service.

    For more information, see Restart a service.