



文件存储HDFS 最佳实践

文档版本: 20220329



## 法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。 如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用 于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格 遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或 提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文 档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
- 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有 任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时 发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠 道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含"阿里云"、"Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

# 通用约定

格式	说明	样例
⚠ 危险	该类警示信息将导致系统重大变更甚至故 障,或者导致人身伤害等结果。	⚠ 危险 重置操作将丢失用户配置数据。
⚠ 警告	该类警示信息可能会导致系统重大变更甚 至故障,或者导致人身伤害等结果。	警告 重启操作将导致业务中断,恢复业务 时间约十分钟。
〔〕) 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	大意 权重设置为0,该服务器不会再接受新 请求。
? 说明	用于补充说明、最佳实践、窍门等,不是 用户必须了解的内容。	<ul><li>⑦ 说明</li><li>您也可以通过按Ctrl+A选中全部文件。</li></ul>
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 <b>结果确认</b> 页面,单击 <b>确定</b> 。
Courier字体	命令或代码。	执行    cd /d C:/window    命令,进入 Windows系统文件夹。
斜体	表示参数、变量。	bae log listinstanceid
[] 或者 [alb]	表示可选项,至多选择一个。	ipconfig [-all -t]
{} 或者 {a b}	表示必选项,至多选择一个。	switch {act ive st and}

## 目录

1.文件存储HDFS版和对象存储OSS双向数据迁移	05
2.文件存储HDFS版和数据库MySQL双向数据迁移	11
3.迁移开源HDFS的数据到文件存储HDFS版	20
4.在文件存储HDFS版上使用CDH6	23
4.1. CDH6数据迁移	23
4.2. 配置CDH6使用文件存储HDFS版	26
4.3. 卸载并释放CDH6 HDFS服务使用的云盘	40
5.在文件存储HDFS版上使用E-MapReduce	41
5.1. E-MapReduce数据迁移	41
5.2. 配置E-MapReduce服务使用文件存储HDFS版	44
5.3. 卸载并释放E-MapReduce HDFS使用的云盘	54
6.性能调优	55
6.1. 性能优化最佳实践	55
6.2. 集群吞吐性能测试	56
6.3. 元数据性能测试	60
7.在文件存储HDFS版上使用Apache Spark	66
8.在文件存储HDFS版上使用Apache Flink	70
9.使用Fuse-DFS挂载文件存储HDFS版	74
10.在文件存储HDFS版上使用Presto	77
11.使用云企业网跨VPC访问文件存储HDFS版	84
12.在文件存储HDFS版上使用TensorFlow	88
13.文件存储HDFS版和MaxCompute双向数据迁移	90
14.在文件存储HDFS版上使用Apache HBase	99
15.在文件存储HDFS版上使用Apache Tez	102

# 1.文件存储HDFS版和对象存储OSS双向 数据迁移

本文档介绍**文件存储HDFS版**和对象存储OSS之间的数据迁移操作过程。您可以将**文件存储HDFS版**数据迁移到对象存储OSS,也可以将对象存储OSS的数据迁移到**文件存储HDFS版**。

## 前提条件

- 已开通文件存储HDFS版服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入门。
- 已搭建Hadoop集群并且所有集群节点安装JDK, JDK版本不低于1.8。建议您使用的Hadoop版本不低于 2.7.2,本文档中使用的Hadoop版本为Apache Hadoop 2.7.2。

## 背景信息

阿里云**文件存储HDFS版**是面向阿里云ECS实例及容器服务等计算资源的文件存储服务。**文件存储HDFS** 版允许您就像在Hadoop的分布式文件系统中管理和访问数据,并对热数据提供高性能的数据访问能力。对 象存储OSS是海量、安全、低成本、高可靠的云存储服务,提供标准型、归档型等多种存储类型。您可以 在**文件存储HDFS版**和对象存储OSS之间双向数据迁移,从而实现热、温、冷数据合理分层,在实现对热数 据的高性能访问的同时,有效控制存储成本。

## 步骤一: Hadoop集群挂载文件存储HDFS版实例

在Hadoop集群中配置文件存储HDFS版实例。具体操作,请参见挂载管理。

## 步骤二:安装OSS客户端JindoFS SDK

在进行数据迁移前,你还需要为Hadoop集群安装OSS客户端JindoFS SDK。JindoFS SDK详细介绍请参见JindoFS SDK。

1. 下载jindofs-sdk.jar, 并将sdk包安装到hadoop的claclasspath下。

cp ./jindofs-sdk-\*.jar \${HADOOP\_HOME}/share/hadoop/hdfs/lib/jindofs-sdk.jar

2. 配置JindoFS OSS实现类。

将JindoFS OSS实现类配置到Hadoop的core-site.xml中。

```
<configuration>
<property>
<name>fs.AbstractFileSystem.oss.impl</name>
<value>com.aliyun.emr.fs.oss.OSS</value>
</property>
<property>
<name>fs.oss.impl</name>
<value>com.aliyun.emr.fs.oss.JindoOssFileSystem</value>
</property>
</configuration>
```

#### 3. 配置OSS AccessKey。

将OSS的AccessKey、AccessKeySecret、Endpoint等预先配置在Hadoop的core-site.xml中。

<configuration></configuration>
<property></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>
ECS<b 环境推荐使用内网OSS Endpoint <b>,即</b> oss-cn-xxx-internal.aliyuncs.com>
<value>oss-cn-xxx.aliyuncs.com</value>

4. 在Hadoop集群使用OSS客户端。

\${HADOOP HOME}/bin/hadoop fs -ls oss://<bucket>/<path>

### 步骤三:数据迁移

为Hadoop集群挂载好HDFS实例和安装OSS客户端JindoFS SDK后,使用Hadoop MapReduce任务即可实现数 据迁移。迁移数据的操作如下所示。

## 实践一:将文件存储HDFS版数据迁移到对象存储OSS

- 1. 准备测试数据。
  - i. 执行以下命令,在HDFS实例中生成1TB测试数据。

```
${HADOOP_HOME}/bin/hadoop jar \
${HADOOP_HOME}/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar \
randomtextwriter \
-D mapreduce.randomtextwriter.totalbytes=1099511627776 \
-D mapreduce.randomtextwriter.bytespermap=5368709120 \
dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/dfs2oss/data/data_1t/
```

其中, f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290 为文件存储HDFS版挂载点域名, /dfs2oss/data/data\_1t为本文测试数据的文件目录。请根据实际情况进行替换。

ii. 执行 \${HADOOP\_HOME}/bin/hadoop fs -du -s -h dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyunc s.com:10290/dfs2oss/data/data 1t/ 命令,查看生成的测试数据大小。

froot@cluster-header-1 ~]# /opt/module/hadoop-2.7.2/bin/hadoop fs -du -s -h dfs://f-(
 .0 ] dfs://f-(
 .0 dfs://f-

2. 启动Hadoop MapReduce任务(DistCp)将测试数据迁移至对象存储OSS。有关DistCp工具使用说明, 请参见Hadoop Distcp工具官方说明。

```
${HADOOP_HOME}/bin/hadoop distcp \
dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/dfs2oss \
oss://<bucket>/<path>
```

其中, <bucket>/<path>为测试数据迁移至OSS的文件目录。请根据实际情况进行替换。

3. 任务执行完成后,查看迁移结果。

.cn-zhangjiakou.dfs.aliyuncs.com:10290/dfs2oss/data/data 1

#### 如果回显包含如下类似信息,说明迁移成功。

```
21/11/11 16:17:55 INFO mapreduce.Job: Job job 1636613902785 0001 completed successfully
21/11/11 16:17:55 INFO mapreduce.Job: Counters: 38
        File System Counters
                DFS: Number of bytes read=1124279173244
                DFS: Number of bytes written=0
                DFS: Number of read operations=742
                DFS: Number of large read operations=0
                DFS: Number of write operations=42
                FILE: Number of bytes read=0
                FILE: Number of bytes written=2596976
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                OSS: Number of bytes read=0
                OSS: Number of bytes written=1124279123564
                OSS: Number of read operations=0
                OSS: Number of large read operations=0
                OSS: Number of write operations=0
        Job Counters
                Launched map tasks=21
                Other local map tasks=21
                Total time spent by all maps in occupied slots (ms)=1930874048
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=60339814
                Total vcore-milliseconds taken by all map tasks=60339814
                Total megabyte-milliseconds taken by all map tasks=61787969536
        Map-Reduce Framework
                Map input records=206
                Map output records=0
                Input split bytes=2814
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=131013
                CPU time spent (ms)=10430520
                Physical memory (bytes) snapshot=9110585344
                Virtual memory (bytes) snapshot=83361333248
                Total committed heap usage (bytes)=15489564672
        File Input Format Counters
                Bytes Read=46866
        File Output Format Counters
                Bytes Written=0
        org.apache.hadoop.tools.mapred.CopyMapper$Counter
                BYTESCOPIED=1124279123564
                BYTESEXPECTED=1124279123564
                COPY=206
```

#### 4. 验证迁移结果。

在迁移完后,执行以下命令,检查对象存储OSS的测试数据是否与HDFS待迁移数据一致。

\${HADOOP HOME}/bin/hadoop fs -du -s -h oss://<bucket>/<path>

如果两者数据一致,则表示迁移成功。

[root@cluster-header-1 ~]# /opt/module/hadoop-2.7.2/bin/hadoop fs -du -s -h oss:/// i i _ i / i / i f 2/	
21/11/11 16:30:31 INFO jboot.JbootLogger: jboot_jni_logging.cpp:34] Jboot logger level is info	
21/11/11 16:30:31 INFO jboot.JbootLogger: AliyunCloudAuth.cpp:61] Default credential provider is not config	
21/11/11 16:30:31 INFO jboot.JbootLogger: HdfsBackend.cpp:25] No conf dir env configured, will start HDFS backend without hdfs-site.xml	
21/11/11 16:30:31 INFO jboot.JbootLogger: NamespaceManager.cpp:49] Initializing namespace connectors	
21/11/11 16:30:31 INFO jboot.JbootLogger: ConcurrencyManager.cpp:36] Init FlushLocal ConcurrencyManager with 100 parallelism	
21/11/11 16:30:31 INFO jboot.JbootLogger: LibclientMainImpl.hpp:152] Enabled Tmpfile Cleaner Thread.	
21/11/11 16:30:31 INFO jboot.JbootLogger: LibclientMainImpl.hpp:100] Client Service Metrics System Start	
21/11/11 16:30:31 INFO jboot.JbootLogger: MemoryBufferManager.cpp:23] Memory buffer capacity 6442450944, low watermark 1932735360, high watermark 418759315	
21/11/11 16:30:31 INFO jboot.JbootLogger: BigbootMainBaseImpl.hpp:130] main doRun() return value True	
21/11/11 16:30:31 INFO jboot.JbootLogger: jboot_native.cpp:39] Successfully initialized libjboot native	
21/11/11 16:30:31 INFO jboot.JbootLogger: BigbootMainBaseImpl.hpp:130] main doRun() return value True	
1.0 T oss://	

## 实践二:将对象存储OSS数据迁移至文件

- 1. 准备测试数据。
  - i. 执行以下命令,在OSS上生成1TB测试数据。

```
${HADOOP_HOME}/bin/hadoop jar \
${HADOOP_HOME}/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar \
randomtextwriter \
-D mapreduce.randomtextwriter.totalbytes=1099511627776 \
-D mapreduce.randomtextwriter.bytespermap=5368709120 \
oss://<bucket>/<path>
```

#### 其中, <bucket>/<path>为测试数据OSS Bucket目录。请根据实际情况进行替换。

ii. 执行 \${HADOOP\_HOME}/bin/hadoop fs -du -s -h oss://<bucket>/<path> 命令, 查看已生成的1 TB测试数据。

[root@cluster-header-1	~]# /opt/module/hadoop-2.7.2/bin/hadoop fs -du -s -h oss:// ////////////////////////////////
21/11/11 10:14:58 INFO	jboot.JbootLogger: jboot_jni_logging.cpp:34] Jboot logger level is info
21/11/11 10:14:58 INFO	jboot.JbootLogger: AliyunCloudAuth.cpp:61] Default credential provider is not config
21/11/11 10:14:58 INFO	jboot.JbootLogger: HdfsBackend.cpp:25] No conf dir env configured, will start HDFS backend without hdfs-site.xml
21/11/11 10:14:58 INFO	jboot.JbootLogger: NamespaceManager.cpp:49] Initializing namespace connectors
21/11/11 10:14:58 INFO	jboot.JbootLogger: MemoryBufferManager.cpp:23] Memory buffer capacity 6442450944, low watermark 1932735360, high watermark 4187593152
21/11/11 10:14:58 INFO	jboot.JbootLogger: ConcurrencyManager.cpp:36] Init FlushLocal ConcurrencyManager with 100 parallelism
21/11/11 10:14:58 INFO	jboot.JbootLogger: LibclientMainImpl.hpp:100] Client Service Metrics System Start
21/11/11 10:14:58 INFO	jboot.JbootLogger: BigbootMainBaseImpl.hpp:130] main doRun() return value True
21/11/11 10:14:58 INFO	jboot.JbootLogger: BigbootMainBaseImpl.hpp:130] main doRun() return value True
21/11/11 10:14:58 INFO	jboot.JbootLogger: LibclientMainImpl.hpp:152] Enabled Tmpfile Cleaner Thread.
21/11/11 10:14:58 INFO	jboot.JbootLogger: jboot_native.cpp:39] Successfully initialized libjboot native
1.0 T oss://	data_1t

2. 启动Hadoop MapReduce任务(DistCp)将测试数据迁移至文件存储HDFS版。

```
${HADOOP_HOME}/bin/hadoop distcp \
oss://<bucket>/<path>
dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/oss2dfs
```

其中, *f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290*为HDFS文件系统挂载点域名, /oss2dfs为测 试数据迁移至HDFS的文件目录。请根据实际情况进行替换。

3. 任务执行完成后,查看迁移结果。

如果回显包含如下类似信息,说明迁移成功。

```
21/11/11 10:37:18 INFO mapreduce.Job: Job job 1636535499203 0007 completed successfully
21/11/11 10:37:18 INFO mapreduce.Job: Counters: 38
        File System Counters
               DFS: Number of bytes read=37907
                DFS: Number of bytes written=1124279506814
               DFS: Number of read operations=1397
               DFS: Number of large read operations=0
               DFS: Number of write operations=453
                FILE: Number of bytes read=0
               FILE: Number of bytes written=2598446
               FILE: Number of read operations=0
               FILE: Number of large read operations=0
               FILE: Number of write operations=0
               OSS: Number of bytes read=1124279506814
               OSS: Number of bytes written=0
               OSS: Number of read operations=0
                OSS: Number of large read operations=0
               OSS: Number of write operations=0
        Job Counters
               Launched map tasks=21
                Other local map tasks=21
               Total time spent by all maps in occupied slots (ms)=783323456
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=24478858
                Total vcore-milliseconds taken by all map tasks=24478858
               Total megabyte-milliseconds taken by all map tasks=25066350592
        Map-Reduce Framework
               Map input records=206
               Map output records=0
               Input split bytes=2793
               Spilled Records=0
               Failed Shuffles=0
               Merged Map outputs=0
                GC time elapsed (ms)=112771
                CPU time spent (ms)=9003330
               Physical memory (bytes) snapshot=7519907840
               Virtual memory (bytes) snapshot=83900002304
               Total committed heap usage (bytes)=14357626880
        File Input Format Counters
               Bytes Read=35114
        File Output Format Counters
               Bytes Written=0
        org.apache.hadoop.tools.mapred.CopyMapper$Counter
               BYTESCOPIED=1124279506814
                BYTESEXPECTED=1124279506814
                COPY=206
```

#### 4. 验证迁移结果。

在迁移完成后,执行以下命令,检查迁移到**文件存储HDFS版**的测试数据是否与源OSS待迁移数据一致。

\${HADOOP\_HOME}/bin/hadoop fs -du -s dfs://f-xxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290
/oss2dfs

如果两者数据一致,则表示迁移成功。

[root@cluster-header-1 ~]#	/opt/module/hadoop-2.7.2/bin/hadoop fs -du -s dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/oss2dfs
124279506814 dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/oss2dfs	
root@cluster-header-1 ~]#	/opt/module/hadoop-2.7.2/bin/hadoop fs -du -s -h dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/oss2dfs
.0 T dfs://f-	.cn-zhangjiakou.dts.aliyuncs.com:10290/oss2dts	

## 常见问题

对于正在写入的文件,进行迁移时会遗漏最新写入的数据吗?

Hadoop兼容文件系统提供单写者多读者并发语义,针对同一个文件,同一时刻可以有一个写者写入和多个读者读出。以**文件存储HDFS版**到对象存储OSS的数据迁移为例,数据迁移任务打开**文件存储HDFS版**的文件F,根据当前系统状态决定文件F的长度L,将L字节迁移到对象存储OSS。如果在数据迁移过程中,有并发的写者写入,文件F的长度将超过L,但是数据迁移任务无法感知到最新写入的数据。因此,建议当您在做数据迁移时,请避免往迁移的文件中写入数据。

## 2.文件存储HDFS版和数据库MySQL双向 数据迁移

本文介绍如何使用Sqoop工具实现文件存储HDFS版和关系型数据库MySQL之间的双向数据迁移。

#### 前提条件

- 已开通文件存储HDFS版服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入门。
- 已搭建Hadoop集群。建议您使用的Hadoop版本不低于2.7.2,本文使用的Hadoop版本为Apache Hadoop 2.8.5。
- 已为Hadoop集群所有节点安装JDK, 且JDK版本不低于1.8。更多信息, 下载JDK。

## 背景信息

Sqoop是一款开源的工具,主要用于在Hadoop和结构化数据存储(例如关系数据库)之间高效传输批量数据。既可以将一个关系型数据库(MySQL、Oracle、Postgres等)中的数据导入**文件存储HDFS版**中,也可以将**文件存储HDFS版**的数据导入到关系型数据库中。

### 步骤一: Hadoop集群挂载文件存储HDFS版实例

在Hadoop集群中配置文件存储HDFS版实例。具体操作,请参见挂载文件系统。

## 步骤二:安装Sqoop

目前Sqoop分为Sqoop1和Sqoop2两个版本,且两个版本并不兼容。本文以Sqoop1的稳定版本Sqoop 1.4.7版本为例进行介绍。

#### 1. 下载Sqoop 1.4.7版本。

2. 执行以下命令, 解压安装包。

tar -zxf sqoop-1.4.7.bin\_hadoop-2.6.0.tar.gz -C /usr/local/

#### 3. 配置环境变量。

i. 执行 vim /etc/profile 命令,打开配置文件,添加如下内容:

export SQOOP\_HOME=/usr/local/sqoop-1.4.7.bin\_hadoop-2.6.0
export PATH=\$PATH:\$SQOOP HOME/bin

ii. 执行 source /etc/profile 命令, 使配置生效。

#### 4. 添加数据库驱动。

#### i. 下载MySQL链接包。

wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.48.ta
r.gz

#### ii. 解压链接包。

tar -zxf mysql-connector-java-5.1.48.tar.gz

#### iii. 将MySQL链接包存放到Sqoop安装目录的lib目录下。

cp ./mysql-connector-java-5.1.48/mysql-connector-java-5.1.48.jar \${SQOOP\_HOME}/lib/

#### 5. 修改配置文件。

i. 执行如下命令复制sqoop-env-template.sh,并命名为sqoop-env.sh。

cp \${SQOOP\_HOME}/conf/sqoop-env-template.sh \${SQOOP\_HOME}/conf/sqoop-env.sh

ii. 执行 vim \${SQOOP HOME}/conf/sqoop-env.sh 命令打开配置文件, 添加如下内容。

```
export HADOOP_COMMON_HOME=/usr/local/hadoop-2.8.5
export HADOOP_MAPRED_HOME=$HADOOP_COMMON_HOME
export HIVE_HOME=/usr/local/apache-hive-2.3.9-bin #若没有安装hive可不必添加此配置
```

ⅲ.执行 cp \${HIVE HOME}/lib/hive-common-2.3.9.jar \${SQOOP HOME}/lib/ 命令复制文件。

#### 6. 执行如下命令验证数据库是否连接成功。

sqoop list-databases --connect jdbc:mysql://<dburi> --username 'username' --password 'p
assword'

参数	说明
dburi	数据库的访问连接。例如 jdbc:mysql://0.0.0.0:3306/ 。
username	数据库登录用户名。
password	用户密码。

如果回显信息中显示MySQL数据库的名称,则表示连接成功。

19/08/21 11:07:12 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider	using -P	instead.
19/08/21 11:07:12 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.		
information_schema		
hivemeta		
hue		
mysql		
performance_schema		
tost		

### 步骤三:数据迁移

下面介绍数据迁移方式及迁移后的结果验证。

## 实践一:将文件存储HDFS版的数据迁移到MySQL

将**文件存储HDFS版**的数据迁移到MySQL,需要先在MySQL上创建好对应数据结构的表,然后在集群Sqoop 节点上使用 sqoop export 命令进行迁移。

此处以迁移文件存储HDFS版/sqoop2mysql/table/mysqltest.txt中的数据为例,mysqltest.txt中已写入如下数据。

- 6,测试用户6,2019-08-10,男 7,测试用户7,2019-08-11,男 8,测试用户8,2019-08-12,男 9,测试用户9,2019-08-13,女 10,测试用户10,2019-08-14,女
- 1. 执行 create database sqoop migrate; 命令, 创建MySQL数据库。
- 2. 执行以下命令, 创建表。

```
use sqoop_migrate;
CREATE TABLE `employee` (
 `e_id` varchar(20) NOT NULL DEFAULT '',
 `e_name` varchar(20) NOT NULL DEFAULT '',
 `e_birth` varchar(20) NOT NULL DEFAULT '',
 `e_sex` varchar(10) NOT NULL DEFAULT '',
 PRIMARY KEY (`e_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

3. 执行以下命令,迁移数据。

```
sqoop export --connect jdbc:mysql://localhost:3306/sqoop_migrate --username 'userName
' --password 'userPW' --num-mappers 1 --table employee --columns "e_id,e_name,e_birt
h,e_sex" --export-dir '/sqoop2mysql/table/mysqltest.txt' --fields-terminated-by ','
```

#### 迁移命令格式:

sqoop export --connect jdbc:mysql://<dburi>/<dbname> --username <username> --password
<password> --table <tablename> --export-dir <hdfs-dir>

参数	说明		
dburi	数据库的访问连接。例如 jdbc:mysql://localhost:3306/ 。如果 您的访问连接中含有参数,则请加上单引号,例如 'jdbc:mysql://localhost:3306/mydatabase? useUnicode=true&characterEncoding=UTF-8'		
dbname	数据库的名字,例如 <i>user</i> 。		
username	数据库登录用户名。		
password	用户密码。		
tablename	MySQL数据库中表的名称。		

参数	说明
hdfs-dir	存放待迁移数据的 <b>文件存储HDFS版</b> 目录。

4. 验证迁移结果。

执行 select \* from employee; 命令查看表数据。如果表中有如下数据,则表示迁移成功。

<pre>mysql&gt; select * from employee;</pre>				
e_id	e_name	e_birth	e_sex	
10   6   7   8   9	测试用户10   测试用户6   测试用户7   测试用户8   测试用户9	2019-08-14 2019-08-10 2019-08-11 2019-08-12 2019-08-13	女       男       男	
5 rows :	in set (0.00 sec)			

## 实践二:将MySQL的数据迁移到HDFS

在集群Sqoop节点上,使用 sqoop import 命令将MySQL中的数据迁移到文件存储HDFS版上。

<pre>mysql&gt; select * from employee;</pre>				
e_id	e_name	e_birth	e_sex	
10   6   7   8   9	测试用户10   测试用户6   测试用户7   测试用户8   测试用户9	2019-08-14 2019-08-10 2019-08-11 2019-08-12 2019-08-13	女       男 男   男 女	
+ 5 rows	in set (0.00 sec)	+	+	

此处以迁移MySQL中的employee表为例, employee表中已写入如下数据。

1. 执行以下命令迁移数据。

sqoop import --connect jdbc:mysql://localhost:3306/sqoop\_migrate --username 'userid' -password 'userPW' --table employee --target-dir /mysql2sqoop/table/sqoop\_migrate --nu
m-mappers 1 --columns "e\_id,e\_name,e\_birth,e\_sex" --direct

#### 命令格式:

sqoop import --connect jdbc:mysql://<dburi>/<dbname> --username <username> --password
<password> --table <tablename> --check-column <col> --incremental <mode> --last-value
<value> --target-dir <hdfs-dir>

#### 参数说明如下所示,更多信息,请参见Sqoop Import。

参数	说明				
dburi	数据库的访问连接。例如 jdbc:mysql://172.x.x.x:3306/ 。 如果您的访问连接中含有参数,则请加上单引号。例如 'jdbc:mysql://172.x.x.x.235:3306/mydatabase? useUnicode=true&characterEncoding=UTF-8' 。				
dbname	数据库的名字。例如 <i>user</i> 。				
username	数据库登录用户名。				
password	用户密码。				
tablename	MySQL数据库中表的名称。				
col	迁移表中列的名称。				
mode	该模式决定Sqoop如何定义哪些行为新的行。取值为 <i>append</i> 或 <i>lastmodified</i> 。				
value	前一个导入中检查列的最大值。				
hdfs-dir	文件存储HDFS版的写入目录。				

#### 2. 检查迁移结果。

i. 执行 hadoop fs -ls /mysql2sqoop/table/sqoop\_migrate 命令,获取迁移文件信息。

[root@master1 Found 2 items	test]#	# /usr/local,	/hado	oop-2.8.5/bi	in/hado	op fs -ls	/mysql2sqoop/table/s	qoop_migrate
-rwxrwxrwx	3 root	root	0	2021-11-09	11:34	/mysql2sq	oop/table/sqoop_migra	at /_SUCCESS
-rwxrwxrwx	3 root	root	157	2021-11-09	11:34	/mvsal2sa	oop/table/sgoop migra	ate/part-m-00000

ii. 执行 hadoop fs -cat /mysql2sqoop/table/sqoop\_migrate/part-m-00000 命令查看文件中的内

容。 如果part-m-00000文件中有如下内容,则表示迁移成功。

[root@master1	. test]# /usr/local/hadoop-2.8.5/bin/hadoo	o fs -cat	/mysql2sqoop/table/sqoop_migrate/part-m-00000
10,测试用户10	,2019-08-14,女		
6,测试用户6,2	.019-08-10,男		
7,测试用户7,2	019-08-11,男		
8,测试用户8,2	.019-08-12,男		
9,测试用户9,2	.019-08-13,女		

## 实践三:将MySQL的数据迁移到Hive

在集群Sqoop节点上使用 sqoop import 命令可以将MySQL上的数据迁移到Hive上。

<pre>mysql&gt; select * from employee;</pre>				
e_id	e_name	e_birth	e_sex	
10   6   7   8   9	测试用户10   测试用户6   测试用户7   测试用户8   测试用户9	2019-08-14 2019-08-10 2019-08-11 2019-08-12 2019-08-13	女       男男   女	
+ 5 rows :	in set (0.00 sec)	+	++	

此处以迁移MySQL中的employee表为例, employee表中已写入如下数据。

#### 1. 执行以下命令迁移数据。

sqoop import --connect jdbc:mysql://localhost:3306/sqoop\_migrate --username 'userid' -password 'PW' --table employee --hive-import --hive-database default --create-hive
-table --hive-overwrite -m 1

#### 迁移命令格式:

sqoop import --connect jdbc:mysql://<dburi>/<dburi> --username <username> --password
<password> --table <tablename> --fields-terminated-by "\t" --lines-terminated-by "\n" -hive-import --target-dir <hdfs-dir> --hive-table <hive-tablename>

参数	说明			
dburi	数据库的访问连接。例如 jdbc:mysql://localhost:3306/ 。如果您的访问连 接中含有参数,则请加上单引号,例如 'jdbc:mysql://localhost:3306/mydatabase? useUnicode=true&characterEncoding=UTF-8'			
dbname	数据库的名字。例如 <i>user</i> 。			
username	数据库登录用户名。			
password	用户密码。			
tablename	MySQL数据库中表的名称。			

参数	说明
hdfs-dir	文件存储HDFS版的写入目录。
hive-tablename	对应的Hive中的表名。

#### 2. 验证迁移结果。

执行 select \* from default.employee; 命令查看表数据,如果表中有如下数据,则表示迁移成功。

#### 实践四:将Hive的数据迁移到MySQL

将Hive的数据迁移到MySQL上,需要先在MySQL上创建好对应Hive数据结构的表,然后在集群Sqoop节点上 使用 sqoop export 命令进行迁移。

此处以迁移Hive上default.employee中的数据为例,该表中已写入如下数据。

hive>	selec	t *	fro	m defaul	t.emp	loye	e;		
OK									
10	测证	代用丿	户 <b>10</b>	20	19-08	3-14		女	
6	测证	代用ノ	户6	20	19-08	3-10		男	
7	测证	、用ノ	户7	20	19-08	3-11		男	
8	测证	代用ノ	户8	20	19-08	3-12		男	
9	测证	、用)	户 <b>9</b>	20	19-08	3-13		女	
Time	taken:	1.1	153	seconds,	Feto	:hed:	5	row(s)	

1. 在MySQL上的sqoop\_migrate库中创建要导入的表。

```
use sqoop_migrate;
CREATE TABLE `employee_from_hive`(
  `id` VARCHAR(20),
  `name` VARCHAR(20) NOT NULL DEFAULT '',
  `birth` VARCHAR(20) NOT NULL DEFAULT '',
  `sex` VARCHAR(10) NOT NULL DEFAULT '',
  PRIMARY KEY(`id`)
);
```

2. 执行 DESCRIBE FORMATTED default.employee; 命令查看表信息。

hive> DESCRIBE FORMATTE	D default.employee;			
ОК				
# col_name	data_type comment			
e_id	string			
e_name	string			
e_birth	string			
e_sex	string			
# Detailed Table Inform	ation			
Database:	default			
Owner:	root			
CreateTime:	Tue Nov 09 15:08:51 CST 2021			
LastAccessTime:	UNKNOWN			
Retention:	e			
Location:	dfs://fcn-beijing.dfs.aliyuncs.com:10290/user/hive/warehouse/employee			
Table Type:	MANAGED_TABLE			
Table Parameters:				
comment	Imported by sqoop on 2021/11/09 15:08:42			
numFiles	1			
numRows	0			
rawDataSize	0			
totalSize	157			
transient_lastD	dlTime 1636441731			
# Storage Information				
SerDe Library:	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe			
InputFormat:	org.apache.hadoop.mapred.TextInputFormat			
OutputFormat:	org.apache.hadoop.hive.gl.io.HiveIgnoreKevTextOutputFormat			
Compressed:	No			
Num Buckets:	-1			
Bucket Columns:	[]			
Sort Columns:				
Storage Desc Params:				
field.delim	\u0001			
line.delim	n			
serialization.f	ormat \u0001			
Time taken: 0.038 secon	ds, Fetched: 35 row(s)			

3. 执行以下命令迁移数据。

sqoop export --connect jdbc:mysql://localhost:3306/sqoop\_migrate --username 'userid' -password 'userPW' --table employee\_from\_hive -m 1 --fields-terminated-by '\0001' --exp
ort-dir /user/hive/warehouse/employee

#### 迁移命令格式:

sqoop export --connect jdbc:mysql://<dburi>/<dbname> --username <username> --password
<password> --table <tablename> --export-dir <hive-dir> --fields-terminated-by <Splitter>

参数	说明
dburi	数据库的访问连接。例如 jdbc:mysql://localhost:3306/ 。如果您的访问连 接中含有参数,则请加上单引号,例如 'jdbc:mysql://localhost:3306/mydatabase? useUnicode=true&characterEncoding=UTF-8'
dbname	数据库的名字,例如user。

参数	说明
username	数据库登录用户名。
password	用户密码。
tablename	MySQL数据库中表的名称。
hive-dir	存放待迁移数据的 <b>文件存储HDFS版</b> 目录。
Splitter	Hive表数据使用的分隔符。

#### 4. 验证迁移结果。

执行 select \* from sqoop\_migrate.employee\_from\_hive; 命令查看表数据。如果表中有如下数据,

则表示迁移成功。

mysql>	select * from	sqoop_migrate.	employee_from_hive;
id	name	birth	sex
10     6     7     8     9	测试用户10 测试用户6 测试用户7 测试用户8 测试用户9	2019-08-14 2019-08-10 2019-08-11 2019-08-12 2019-08-13	女     男     男     女
5 rows	in set (0.00 s	sec)	

## 3.迁移开源HDFS的数据到文件存储HDFS 版

本文档介绍如何将开源HDFS的数据平滑地迁移到文件存储HDFS版。

## 背景信息

当前业界有很多公司是以Hadoop技术构建数据中心,而越来越多的公司和企业希望将业务顺畅地迁移到云上。**文件存储HDFS版**可以帮助您实现将开源HDFS的数据迁移到云上,并允许您在云上就像在Hadoop分布 式文件系统中管理和访问数据。

## 准备工作

- 1. 开通**文件存储HDFS版**服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入门。
- 2. 基于阿里云ECS搭建Hadoop集群(下称"迁移集群"),用于访问**文件存储HDFS版**实例和迁移数据,并 满足以下条件:
  - 迁移集群与**文件存储HDFS版**实例在相同区域、相同可用区。
  - 迁移集群与文件存储HDFS版实例的挂载点使用相同阿里云VPC网络和交换机。
  - 。 迁移集群上安装的JDK版本不低于1.8。
  - 迁移集群上安装的Hadoop版本不低于2.7.2。

⑦ 说明 如果原集群满足上述条件且计算资源充足,可以直接将原集群当作迁移集群使用,而不必额外创建新集群。

- 3. 在迁移集群中配置文件存储HDFS版实例。具体操作,请参见挂载管理。
- 4. 验证迁移集群和文件存储HDFS版实例之间的连通性。
  - i. 执行以下命令,在文件存储HDFS版上创建目录(例如/dfs\_links)。

hadoop fs -mkdir /dfs\_links

ii. 执行以下命令, 验证连通性。

hadoop fs -ls dfs://f-xxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com:10290/dfs links

其中 f-xxxxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为文件存储HDFS版挂载点域名,请根据您的实际情况进行修改。

如果命令正常执行无输出结果,则表示连通成功。如果连通失败,请排查连通性问题。具体操作, 请参见创建文件系统实例后,为什么无法访问文件存储HDFS版?。

5. 准备迁移工具。

您可以通过Hadoop社区标准的Dist Cp工具实现全量或增量的HDFS数据迁移。更多关于Dist Cp工具使用 说明,请参见Hadoop Dist Cp工具官方说明文档。 ⑦ 说明 使用 hadoop distep 命令将原集群数据迁移至文件存储HDFS版时,请注意文件存储 HDFS版不支持以下参数,其它参数使用和Hadoop DistCp工具官方说明文档一致。文件存储HDFS 版及命令行存在限制的更多信息,请参见使用限制。

参数	描述	状态
-p[rbpax]	r: replication, b: block- size, p: permission, a: ACL, x: XATTR	不可用

## 数据迁移

下面介绍直接迁移和数据中转两种迁移方式,推荐使用直接迁移。

## 方法一: 直接迁移

直接从原集群迁移数据到**文件存储HDFS版**,需要先连通原集群和迁移集群的网络环境,再执行数据迁移任务。

- 1. 连通原集群与迁移集群的网络环境。
  - 场景A: 原集群在阿里云ECS实例上(包括EMR)。

如果在准备工作中将原集群当作迁移集群来使用,请直接执行步骤2中的迁移命令。

- 原集群处于经典网络时,建立ClassicLink连接。
- 集群处于VPC网络环境,但无法访问**文件存储HDFS版**实例时,请参见使用云企业网跨VPC访问文 件存储HDFS版。
- 场景B: 原集群在其他平台上。

使用阿里云高速通道产品连通原集群和迁移集群的网络环境。具体操作,请参见方案介绍。

2. 在已配置文件存储HDFS版的集群上参考以下命令迁移数据。

hadoop distcp hdfs://x.x.x.x:9000/user/hive/warehouse dfs://f-xxxxxxxxxxxxx.cn-xxxxx xx.dfs.aliyuncs.com:10290/user/hive/warehouse

## 方法二:数据中转

如果原集群与迁移集群的网络无法连通,可以参考以下步骤迁移数据。

- 1. 将原集群数据迁移到对象存储OSS。具体操作,请参见离线迁移。
- 2. 将对象存储OSS数据迁移到**文件存储HDFS版**。具体操作,请参见文件存储HDFS版和对象存储OSS双向数据迁移。

#### 常见问题

- 整体迁移速度受Hadoop集群与文件存储HDFS版之间的带宽、集群规模影响。同时传输文件越多, checksum需要的时间越长。如果迁移数据量大,建议先尝试迁移几个目录评估整体时间。如果只能在指定时间段内迁移数据,可以将目录切为几个小目录,依次迁移。
- 一般全量数据同步时,需要一个短暂的业务停写过程,用来启用双写双算。
- 迁移过程出现异常提示: Cannot obtain block length for LocatedBlock。

从原生的HDFS往对象存储OSS或**文件存储HDFS版**迁移数据时,可能会遇到这个问题。遇到该问题时,请 执行 hdfs fsck / -openforwrite 命令,检查当前是否有文件处于写入状态尚未关闭。如果有处于写入 状态的文件时,需判断文件是否有效。

• 如果文件无效,则直接删除文件。

hdfs rm <path-of-the-file>

• 如果文件有效,则不能直接删除,请考虑恢复问题文件租约。

hdfs debug recoverLease -path <path-of-the-file> -retries <retry times>

## 4.在文件存储HDFS版上使用CDH6

## 4.1. CDH6数据迁移

本文介绍如何将CDH中本地HDFS的数据迁移到**文件存储HDFS版**,实现存储计算分离。

## 前提条件

- 1. 已开通**文件存储HDFS版**服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入 门。
- 2. 已在阿里云上创建ECS实例并安装CDH6集群。具体操作,请参见CDH官方文档。

## 背景信息

CDH(Cloudera's Dist ribution, including Apache Hadoop)是众多Hadoop发行版本中的一种,您可以使用文件存储HDFS版替换CDH6原有的本地HDFS服务,通过CDH6和文件存储HDFS版实现大数据计算在云上的存储与计算分离,应对灵活多变的业务需求的挑战。

## 步骤一: 配置文件存储HDFS版

- 1. 登录CDH6的Cloudera Manager管理页面。
- 2. 配置文件存储HDFS版实现类。
  - i. 在主页页面,选择配置 > 高级配置代码段,进入高级配置代码段页面。
  - ii. 在搜索框中输入 core-site.xml 进行搜索,在搜索结果HDFS的core-site.xml的群集范围高级
     配置代码段(安全阀)区域中,单击<☐添加文件存储HDFS版配置项。</li>
    - a. 配置fs.dfs.impl。
      - 名称: fs.dfs.impl。
      - 值:: com.alibaba.dfs.DistributedFileSystem。
    - b. 配置fs.AbstractFileSystem.dfs.impl。
      - 名称: fs.AbstractFileSystem.dfs.impl。
      - 值:: com.alibaba.dfs.DFS。

高级配置代码段 core-site.xml 的群集范围高级配置代码段(安 Cluster 1 > HDFS(服务范围) ヘ 今阀) ? 以 XML 格式查看 名称 fs.dfs.imp 值 com.alibaba.dfs.DistributedFileSystem 说明 □ 最终 E 名称 fs.AbstractFileSystem.dfs.impl com.alibaba.dfs.DFS 值 说明 □ 最终

- iii. 单击保存更改。
- 3. 配置mapreduce.application.classpath。
  - i. 返回**主页**页面,选择**状态**页签。

ii. 在左侧集群组件列表中,单击YARN(MR2 Included)右侧 图标 > 配置,进入YARN(MR2

Included)配置页面。

iii. 在搜索框中输入mapreduce.application.classpath进行搜索,在搜索结果的MR应用程序 Classpath区域中,单击+示加\$HADOOP\_HDFS\_HOME/\*配置。

<sup>Cluster</sup> 1 ♥ YARN (MR2 I	ncluded	) 操作 -			11月 19, 晚上6点11分 CST
状态 实例 配置	命令	应用程序 资源池 图录	長库 审核	Web UI ▼ 快速链接 ▼	
mapreduce.application	ı.classpath				角色组
					显示所有说明
筛选器	清除全部	MR 应用程序 Classpath		Gateway Default Group 👆	?
▼范围		mapreduce.application.classpa	th	\$HADOOP_CLIENT_CONF_DIR	
YARN (MR2 Included)	(服务范 0			\$PWD/mr-framework/*	
Gateway JobHistory Server	1 0			\$MR2_CLASSPATH	<b>⊕</b> ⊕
NodeManager ResourceManager	0			\$HADOOP_HDFS_HOME/*	
✔ 类别				L	
主要	1				25 🗸 每页
代理	0				

- ⅳ. 单击保存更改。
- 4. 配置文件存储HDFS版Java SDK。
  - i. 下载最新的文件存储HDFS版Java SDK (aliyun-sdk-dfs-x.y.z.jar)。
  - ii. 将下载的文件存储HDFS版Java SDK复制到CDH HDFS服务的CLASSPATH路径下。

cp aliyun-sdk-dfs-x.y.z.jar /opt/cloudera/parcels/CDH/lib/hadoop-hdfs/

⑦ 说明 集群中的每台机器都需要在相同位置添加文件存储HDFS版 Java SDK。

- 5. 部署客户端配置。
  - Ⅰ. 返回主页页面,选择状态页签。在左侧集群组件中,单击HDFS右侧的≧图标,进入过期配置页面。
  - ii. 在过期配置页面, 单击重启过时服务。
  - iii. 在重启过时服务页面,选中重新部署客户端配置后,单击立即重启。
  - iv. 在服务全部重启完成,并重新部署客户端配置后,单击完成。

(可选)步骤二:HBase快照迁移

如果原HDFS集群中已部署HBase服务且存在业务数据,您需要将HBase中的数据以快照方式迁移到**文件存储** HDFS版。

本文以HBase服务的两个模拟表(mock\_table\_0和mock\_table\_1),每个表的中的模拟数据有10万条为例,介绍将HBase中的数据以快照方式迁移到**文件存储HDFS版**。

```
hbase(main):001:0> list
TABLE
mock table 0
mock table 1
2 row(s)
Took 0.4213 seconds
=> ["mock table 0", "mock table 1"]
hbase(main):002:0> count 'mock table 1'
Current count: 1000, row: 001637311398
. . . . .
Current count: 100000, row: 991637311398
100000 row(s)
Took 4.3696 seconds
=> 100000
hbase(main):003:0> count 'mock table 0'
Current count: 1000, row: 001637311398
. . . . . .
Current count: 100000, row: 991637311398
100000 row(s)
Took 4.0691 seconds
=> 100000
```

1. 在HBase Shell中执行以下命令,为mock\_table\_0和mock\_table\_1创建快照。

```
snapshot 'mock_table_0', 'mock_table_0_snapshot'
snapshot 'mock table 1', 'mock table 1 snapshot'
```

2. 通过刚创建的快照将HDFS中的数据拷贝到文件存储HDFS版。

```
sudo -u hdfs hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot \
-snapshot mock_table_0_snapshot \
-copy-to dfs://f-xxxxxxxxx.cn-xxxxx.dfs.aliyuncs.com:10290/hbase
sudo -u hdfs hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot \
-snapshot mock_table_1_snapshot \
-copy-to dfs://f-xxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com:10290/hbase
```

其中 f-xxxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为您的文件存储HDFS版挂载点地址,请根据实际情况进行修改。

#### 步骤三:数据迁移

1. 暂停服务。

为了保证在更换文件存储系统的过程中文件数据不丢失,需要暂停数据处理服务(例如:YARN服务、 Hive服务、Spark服务、HBase服务等),HDFS服务仍需保持运行。此处以停止Hive服务为例进行说明。

- i. 在CDH Web主页,找到Hive服务,在右侧的操作项中,单击停止。
- ii. 在停止确认框中,单击停止。当Hive前的图标变成灰色,表示该服务完全停止。
- iii. 重复上述步骤,停止剩余服务。

⑦ 说明 建议只保留HDFS服务正常运行,以方便进行数据迁移。但是如果要迁移的数据量大,请 开启YARN服务,以便使用数据迁移工具Hadoop Dist Cp并行执行迁移任务。

#### 2. 迁移数据。

建议将/user等服务目录和相关数据目录全量迁移至文件存储HDFS版。

- 如果涉及将云下集群的数据迁移到云上。具体操作,请参见迁移开源HDFS的数据到文件存储HDFS 版。
- HBase服务只迁移快照,其他数据目录不迁移。
- 如果CDH原HDFS文件系统上的数据量较小,可以使用 hadoop fs -cp 命令进行数据迁移。

为了避免因为权限问题导致数据迁移失败,建议切换到hdfs用户执行命令。

sudo -u hdfs hadoop fs -cp hdfs://oldclusterip:8020/user dfs://f-xxxxxxxxxxxxxxx.cnxxxxxx.dfs.aliyuncs.com:10290/

其中 f-xxxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为您的文件存储HDFS版挂载点地址,请 根据实际情况进行修改。

 如果CDH HDFS文件系统上的数据量较大,需要使用数据迁移工具Hadoop Dist cp进行数据迁移。具体 操作,请参见迁移开源HDFS的数据到文件存储HDFS版。

sudo -u hdfs hadoop distcp hdfs://oldclusterip:8020/user dfs://f-xxxxxxxxxxxxxxxx.cnxxxxxx.dfs.aliyuncs.com:10290/

其中 f-xxxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为您的文件存储HDFS版挂载点地址,请 根据实际情况进行修改。

#### 后续步骤

配置CDH6使用文件存储HDFS版

## 4.2. 配置CDH6使用文件存储HDFS版

数据迁移完成后,您还需要配置CDH上的HDFS服务、Hive服务、Spark服务、HBase服务,才能使用**文件存** 储HDFS版。

#### 配置Cloudera Management服务

1. 执行以下命令,将最新的**文件存储HDFS版J**ava SDK复制到Cloudera Management服务的lib目录下。集 群中的每台机器都需要在相同位置添加文件存储HDFS版Java SDK。

cp aliyun-sdk-dfs-x.y.z.jar /opt/cloudera/cm/lib/cdh6/

其中, x.y.z 为文件存储HDFS版Java SDK版本号,请根据实际填写。重启服务。

- 2. 重启服务。
  - i. 登录CDH6的Cloudera Manager管理页面。
  - ii. 在**主页**页面,选择**状态**页签。
  - iii. 在左侧Cloudera Management Service区域,单击Cloudera Management Service右侧的 🔺

**图标 > 重启**, 重启服务。

## 配置HDFS服务

- 1. 登录CDH6的Cloudera Manager管理页面。
- 2. 在主页页面,选择配置 > 高级配置代码段,进入高级配置代码段页面。
- 3. 在搜索框中输入core-site.xml进行搜索,在搜索结果HDFS的core-site.xml的群集范围高级配置代码 段(安全阀)区域中,单击⊕,添加如下配置。
  - i. 名称: fs.defaultFS。
  - ii. 值: dfs://f-xxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com:10290

其中 f-xxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为您的文件存储HDFS版挂载点地址,

请根据实际情况进行修改。

core-site.xml 的群集范围高	Cluster 1 > HDFS(服务范围) <b>つ</b>		
级配置代码段(安全阀)		以 XML	格式查看
	名称	fs.AbstractFileSystem.dfs.impl	Θ
	值	com.alibaba.dfs.DFS	
	说明	说明	
		Final	
			-
	名称	fs.dfs.impl	Ξ
	值	com.alibaba.dfs.DistributedFileSystem	
	说明	说明	
		Final	
	名称	fs.defaultFS	
	值	dfs://	
	说明	说明	
-		Einal	_

#### 4. 单击保存更改。

## 配置YARN服务

- 1. 检查mapreduce.application.classpath配置。
  - i. 登录CDH6的Cloudera Manager管理页面。
  - ii. 在**主页**页面,选择**状态**页签。
  - iii. 在左侧集群组件列表中,单击YARN(MR2 Included)右侧 ▲ 图标 > 配置,进入YARN(MR2 Included)配置页面。

iv. 在搜索框中输入mapreduce.application.classpath进行搜索,在搜索结果的MR应用程序 Classpath区域中,检查确认已添加*\$HADOOP\_HDFS\_HOME/\**配置。

vster 1 YARN (MR2 In	ncluded	) 操作▼		11月 19, 晚上6点11分 CST
状态 实例 配置	命令	应用程序 资源池 图表库 审	核 Web UI ▼ 快速链接 ▼	
mapreduce.application.c	classpath			角色组
				显示所有说即
筛选器	清除全部	MR 应用程序 Classpath	Gateway Default Group 👆	?
✔ 范围		mapreduce.application.classpath	\$HADOOP_CLIENT_CONF_DIR	$\ominus$ $\oplus$
YARN (MR2 Included) (	服务范 0		\$PWD/mr-framework/*	
Gateway	1			
JobHistory Server NodeManager	0		\$MR2_CLASSPATH	
ResourceManager	0		\$HADOOP_HDFS_HOME/*	⊟⊕
✔ 类别				
主要	1			25 🗸 毎月
代理	0			

如果上述区域中没有*\$HADOOP\_HDFS\_HOME/\**配置,请单击──添加,然后单击**保存更改**。

- 2. 配置mapred-site.xml。
  - i. 在搜索框中输入mapred-site.xml进行搜索。在搜索结果的YARN(MR2 Included)的YARN服务
     MapReduce高级配置代码段(安全阀)区域中,单击
     →添加如下配置项。
    - 名称: mapreduce.application.framework.path。
    - 值: dfs://f-xxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com:10290/user/yarn/mapreduce/mrframework/3.x.x-cdh6.x.x-mr-framework.tar.gz#mr-framework。其中,

f-xxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为您的文件存储HDFS版挂载点地址;

3.x.x-cdh6.x.x-mr-framework 为CDH MR框架文件,请根据实际情况进行修改。

? 说明

如果文件存储HDFS版实例中的 3.x.x-cdh6.x.x-mr-framework.tar.gz 文件不存在, 可

能是因为此文件还没有从CDH的HDFS服务同步到**文件存储HDFS版**。您需要将HDFS服务/use r/yam目录下的所有内容同步到**文件存储HDFS版**上。具体操作,请参见CDH6数据迁移。

′ARN 服务 MapReduce 高级配置代码段(安全 <sup>羽</sup> )	YARN (MR2 Inc	luded) (服务范围) ● 以XML ł	格式查看	?
	名称	mapreduce.application.framework.path	Ξ	
	值	dfs://f-		
	说明	说明		
	ŧ			

- ii. 单击保存更改。
- 3. 部署新配置并重启服务。
  - i. 返回主页页面,选择状态页签。在左侧集群组件中,单击YARN(MR2 Included)右侧的 已图标,进入过期配置页面。
  - ii. 在过期配置页面, 单击重启过时服务。
  - iii. 在重启过时服务页面,选中重新部署客户端配置后,单击立即重启。

iv. 在服务全部重启完成,并重新部署客户端配置后,单击完成。

## 配置Hive服务

## ◯ 注意

配置HDFS服务完成后,才能配置Hive服务。

在配置Hive服务之前,请确认/*user/hive/*目录中的数据已完成全量迁移。具体操作,请参见<mark>步骤三:数据</mark> 迁移。

- 1. 修改元数据。本文以修改Hive服务元数据存储在MySQL中的数据为例,修改DBS表和SDS表相应的存储系统的URL。
  - i. 执行 use metastore; 命令,进入存储Hive元数据的MySQL数据库。
  - ii. 修改表DBS中的数据。
    - a. 执行 SELECT \* FROM DBS LIMIT 5; 命令, 查询表DBS中的数据。返回结果示例如下:

+	+
	+++++
+	
DB TD   DESC	DB LOCATION URI
NAME	OWNED NAME   OWNED TYDE   OPRATE TIME
	OWNER_WARE   OWNER_TITE   CREATE_TIME
++	
· · · · · · · · · · · · · · · · · · ·	
+	
1   Default Hive database	e   hdfs://cdh6-master:8020/user/hive/warehouse
default	public   ROLE   1629164990
1826   NULL	hdfs://cdh6-master:8020/user/hive/warehouse/t
pcds_text_2.db	tpcds_text_2   hive   USER
1629702940	
1828   NULL	hdfs://cdh6-master:8020/user/hive/warehouse/t
pcds_bin_partitioned_orc_2.db	tpcds_bin_partitioned_orc_2   hive   USER
1629703145	
+	+
	+++++
+	
3 rows in set (0.00 sec)	

#### b. 更新DBS表中原HDFS地址为文件存储HDFS版挂载点地址。示例命令如下:

UPDATE DBS SET DB\_LOCATION\_URI=REPLACE(DB\_LOCATION\_URI,'hdfs://cdh6-master:8020
','dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290');

c. 执行 SELECT \* FROM DBS LIMIT 5; 命令, 查看地址是否已更换。

+		
	·	
		+
DB_ID   DESC	DB_LOCATION_ORI	
NAME	OWNER_NAME   OWNER_TY	PE   CREATE_TIME
+	+	
		+
+	++++	+
1   Default Hive d	atabase   dfs://f-xxxxxxx.cm	-zhangjiakou.dfs.aliyuncs.
com:10290/user/hive/ware	house	default
public   ROLE	1629164990	
1826   NULL	dfs://f-xxxxxxx.cn	-zhangjiakou.dfs.aliyuncs.
com:10290/user/hive/ware	house/tpcds_text_2.db	tpcds_text_2
hive   USER	1629702940	
1828   NULL	dfs://f-xxxxxxx.cn	-zhangjiakou.dfs.aliyuncs.
com:10290/user/hive/ware	house/tpcds bin partitioned c	prc 2.db   tpcds bin partit
ioned orc 2   hive	USER   1629703145	
++		
· · ·	·	
2	+++	+
S LOWS TH SET (0.00 SEC)		

#### iii. 修改表SDS中的数据。

a. 执行 SELECT \* FROM SDS LIMIT 5; 命令, 查询表SDS中的数据。返回结果示例如下:

+	-+		+	
+				
+				
+				
SD_ID   CD_ID   INPUT_FORMAT	IS_	COMPRE	SSED	IS
_STOREDASSUBDIRECTORIES   LOCATION				
NUM_BUCKETS   OUTPUT_FORMAT				SE
RDE_ID				
+++	-+		+	
+				
++				
+				
5423   1846   org.apache.hadoop.mapred.TextInputFormat	1			
hdfs://cdh6-master:8020/tmp/tpcds-generate/2/store_sales		I	-1	1
org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat	1	5423		
5424   1847   org.apache.hadoop.mapred.TextInputFormat	1			
hdfs://cdh6-master:8020/tmp/tpcds-generate/2/store return	ns	I	-1	1
org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat	1	5424		
5425   1848   org.apache.hadoop.mapred.TextInputFormat	1			
hdfs://cdh6-master:8020/tmp/tpcds-generate/2/catalog sale	es	I	-1	1
org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat	1	5425		
5426   1849   org.apache.hadoop.mapred.TextInputFormat	1			
hdfs://cdh6-master:8020/tmp/tpcds-generate/2/catalog_retu	ırns	I	-1	1
org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat	1	5426		
5427   1850   org.apache.hadoop.mapred.TextInputFormat	1			
hdfs://cdh6-master:8020/tmp/tpcds-generate/2/web_sales		I	-1	1
org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat	1	5427		
+	-+		+	
+++				
+				
5 rows in set (0.00 sec)				

## b. 更新SDS表中原HDFS地址为**文件存储HDFS版**挂载点地址。示例命令如下:

UPDATE SDS SET LOCATION=REPLACE(LOCATION, 'hdfs://cdh6-master:8020', 'dfs://f-xxx
xxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290');

c. 执行 SELECT \* FROM SDS LIMIT 5; 命令, 查看地址是否已更换。

------\_\_\_\_\_ \_\_\_\_\_\_ | SD ID | CD ID | INPUT FORMAT | IS COMPRESSED | IS STOREDASSUBDIRECTORIES | LOCATION | NUM BUCKETS | OUTPUT FORMAT | SE RDE ID | \_\_\_\_\_ \_\_\_\_\_ -----+ 5423 | 1846 | org.apache.hadoop.mapred.TextInputFormat | | dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/tpcds-generate/2/s tore sales | -1 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOu tputFormat | 5423 | | 5424 | 1847 | org.apache.hadoop.mapred.TextInputFormat | | dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/tpcds-generate/2/s tore returns | -1 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOu 5424 | tputFormat | 5425 | 1848 | org.apache.hadoop.mapred.TextInputFormat | 1 | dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/tpcds-generate/2/c atalog sales | -1 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOu tputFormat | 5425 | | 5426 | 1849 | org.apache.hadoop.mapred.TextInputFormat | | dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/tpcds-generate/2/c -1 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOu atalog returns | tputFormat | 5426 | 5427 | 1850 | org.apache.hadoop.mapred.TextInputFormat | | dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/tpcds-generate/2/w eb sales | -1 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOu tputFormat | 5427 | 5 rows in set (0.00 sec)

#### 2. 重启服务。

- i. 登录CDH6的Cloudera Manager管理页面。
- ii. 在**主页**页面,选择**状态**页签。
- iii. 在左侧集群组件列表中,单击Hive右侧的 🔺 图标 > 启动。
- iv. 在启动确认框中, 单击**启动**。

## 配置Spark服务

#### ↓ 注意

配置HDFS服务完成后,才能配置Spark服务。

配置Spark服务前,请确认/*user/spark*和/*user/history*目录中的数据已经完成了全量迁移。具体操作, 请参见CDH6数据迁移。

执行以下命令将最新的**文件存储HDFS版**Java SDK复制到Spark服务的jars目录下。其中, x.y.z 为**文件存** 储HDFS版Java SDK版本号,请根据实际填写。

cp aliyun-sdk-dfs-x.y.z.jar /opt/cloudera/parcels/CDH/lib/spark/jars/

? 说明

集群中的每台机器都需要在相同位置添加文件存储HDFS版Java SDK。

## 配置HBase服务

#### ○ 注意

配置HDFS服务完成后,才能配置HBase服务。

配置HBase服务前,请确认原HDFS集群HBase中的快照已迁移至**文件存储HDFS版**。具体操作,请参见HBase快照迁移。

- 1. 登录CDH6的Cloudera Manager管理页面。
- 2. 在主页页面,选择配置 > 高级配置代码段,进入高级配置代码段页面。
- 3. 在搜索框中输入hbase-site.xml进行搜索。在搜索结果的hbase-site.xml的HBase服务高级配置代码 段(安全阀)和hbase-site.xml的HBase客户端高级配置代码段(安全阀)区域中,均添加如下配 置项。
  - i. 配置hbase.rootdir。
    - 名称: hbase.rootdir
    - 值: dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/hbase

其中 f-xxxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为您的文件存储HDFS版挂载点地

址,请根据实际情况进行修改。

- ii. 配置hbase.unsafe.stream.capability.enforce。
  - 名称: hbase.unsafe.stream.capability.enforce。
  - 值: false

hbase-site.xml 的 HBase 服务高级配置代码段 (安全阀)	Cluster 1 >	HBase (服务范围) 🕤
	名称	hbase.rootdir
	值	dfs://f-
	说明	说明
		□ 最终
	名称	hbase.unsafe.stream.capability.enforce
	值	false
	说明	说明
		□ 最终
hbase-site.xml 的 HBase 客户端高级配置代码 段(安全阀)	Cluster 1 >	HBase > Gateway Default Group 🤸
	名称	hbase.rootdir
	值	dfs://f- cn-zhangjiakou.dfs.aliyuncs.com:10290/hbase
	说明	说明
		□ 最终

hbase.unsafe.stream.capability.enforce

□ 最終 4. 配置zookeeper.znode.parent。zookeeper.znode.parent默认配置为/hbase, HBase服务配置文件存 储HDFS版时需修改该配置,否则会与存储在Zookeeper中的数据冲突。

false

说明

- i. 在**主页**页面,选择**状态**页签。
- ii. 在左侧集群组件列表中,单击HBase右侧的 🔺 图标 > 配置,进入HBase配置页面。

名称

值

说明

iii. 在搜索框中输入zookeeper.znode.parent进行搜索,在搜索结果的ZooKeeper Znode父级区域中,配置新的HBase根Znode(例如/hbase\_dfs)。

ZooKeeper Znode 父级	HBase (服务范围	) 🔦
zookeeper.znode.parent	/hbase_dfs	

- iv. 单击保存更改。
- 5. 部署新配置并重启服务。
  - i. 返回状态页签,在左侧集群组件列表中,单击HBase右侧的已图标,进入过期配置页面。

- ii. 在过期配置页面, 单击重启过时服务。
- iii. 在重启过时服务页面,单击立即重启。
- iv. 等待服务全部重启完成, 并重新部署客户端配置后, 单击完成。
- 6. (可选)快照恢复。将存储在原HDFS集群的(可选)步骤二: HBase快照迁移到文件存储HDFS版后,可通 过在HBase Shell中执行以下命令查看和恢复文件存储HDFS版中已经迁移的快照。
  - 执行 list snapshots 命令, 查看快照信息。

```
SNAPSHOT TABLE + CREATION TIME
mock_table_0_snapshot mock_table_0 (2021-11-19 17:25:
38 +0800)
mock_table_1_snapshot mock_table_1 (2021-11-19 17:25:
47 +0800)
2 row(s)
Took 0.4619 seconds
=> ["mock table 0 snapshot", "mock table 1 snapshot"]
```

• 执行以下命令,恢复快照。

restore\_snapshot 'mock\_table\_0\_snapshot'

restore\_snapshot 'mock\_table\_1\_snapshot'

## 停止HDFS服务

完成上述各项服务配置后,集群的HDFS服务已经替换为文件存储HDFS版,建议您停止集群的HDFS服务。

您可以在主页页面,选择状态页签。在左侧集群组件中,单击HDFS右侧的 🔺 图标 > 停止,停止HDFS服

务。

#### ? 说明

停止HDFS服务前,请确认HDFS所有需要迁移的数据都已被迁移,停止HDFS服务后将无法迁移数据。具体操作,请参见CDH6数据迁移。

## 验证服务正确性

## YARN服务验证

1. 使用CDH Hadoop中自带包hadoop-mapreduce-examples-3.x.x-cdh6.x.x.jar进行测试,在/tmp/rando mtextwriter目录下生成大小约为128 MB的文件,示例命令如下。

```
yarn jar \
/opt/cloudera/parcels/CDH/jars/hadoop-mapreduce-examples-*-cdh*.jar \
randomtextwriter \
-D mapreduce.randomtextwriter.totalbytes=134217728 \
/tmp/randomtextwriter
```

1. 执行以下命令验证YARN服务与文件存储HDFS版实例的连通性。

hadoop fs -ls dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/randomtextwrit
er

如果看到\_SUCCESS和part-m-00000两个文件,表示YARN服务与文件存储HDFS版实例连通成功。

[root@cdh6-master ~]# hadoop fs -ls dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/randomtextwriter						
Found Z Items						
-rwxrwxrwx 3 root	t root 0	2021-11-22 15:35 c	dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/randomtextwriter/	_SUCCESS	
-rwxrwxrwx 3 root	t ro <mark>o</mark> t 136664696	2021-11-22 15:35 c	lfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/randomtextwriter/	part-m-00000	

## Hive服务验证

• 数据迁移验证

本文以原HDFS集群TPC-DS的数据已迁移到**文件存储HDFS版**为例,进行数据迁移验证。

i. 登录Hue管理控制台,查看store\_sales表详细信息中的Location是否已变更为配置Hive服务时修改的文件存储HDFS版挂载点地址。



ii. 执行一条SQL语句,验证Hive服务与文件存储HDFS版实例之间的连通性。
```
SELECT *
FROM
  (SELECT i manager id ,
          sum(ss sales price) sum sales ,
          avg(sum(ss sales price)) over (partition BY i manager id) avg monthly sales
   FROM item ,
       store sales ,
       date dim ,
       store
   WHERE ss item sk = i item sk
    AND ss sold date sk = d date sk
    AND ss store sk = s store sk
    AND d month seq IN (1211,
                         1211+1,
                         1211+2,
                         1211+3,
                         1211+4,
                         1211+5,
                         1211+6,
                         1211+7,
                         1211+8,
                         1211+9,
                         1211+10,
                         1211+11)
     AND ((i category IN ('Books',
                          'Children',
                          'Electronics')
           AND i_class IN ('personal',
                            'portable',
                           'reference',
                           'self-help')
           AND i_brand IN ('scholaramalgamalg #14',
                            'scholaramalgamalg #7',
                           'exportiunivamalg #9',
                           'scholaramalgamalg #9')) or(i category IN ('Women','Music'
,'Men')
                                                        AND i class IN ('accessories',
'classical','fragrances','pants')
                                                       AND i brand IN ('amalgimporto
#1', 'edu packscholar #1', 'exportiimporto #1', 'importoamalg #1')))
  GROUP BY i manager id,
           d moy) tmp1
WHERE CASE
         WHEN avg monthly sales > 0 THEN ABS (sum sales - avg monthly sales) / avg m
onthly sales
         ELSE NULL
     END > 0.1
ORDER BY i_manager_id ,
        avg monthly sales ,
        sum sales
LIMIT 100;
```

iii. 查看SQL结果。如果回显正常,则表示Hive服务与**文件存储HDFS版**实例之间的连通性正常。

#### • 新建数据表验证

i. 在Hue中创建表并查看表信息。

```
DROP TABLE IF EXISTS user_info_test;
CREATE TABLE user_info_test(user_id BIGINT, firstname STRING, lastname STRING, countr
y STRING);
INSERT INTO user_info_test VALUES(1,'Dennis','Hu','CN'),(2,'Json','Lv','Jpn'),(3,'Mik
e','Lu','USA');
```

<sup>Hive</sup> ▼ 数据库 > defa	ult > user_info_test	▶ 查询 ▲ 导入 × 丢弃 ② 刷新
Overview 样本 (3)	详细信息	
DETAILED TABLE INFORMAT	ION	
Database:	default	
OwnerType:	USER	
Owner:	hdfs	
CreateTime:	Tue Nov 23 10:10:45 CST 2021	
LastAccessTime:	UNKNOWN	
Retention:	0	
Location:	dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/user/hive/warehouse/user_info_test	
Table Type:	MANAGED_TABLE	
Table Parameters:		
	COLUMN_STATS_ACCURATE	{\"BASIC_STATS\":\"true\"}
	numFiles	1
	numRows	3
	rawDataSize	40
	totalSize	43
	transient_lastDdlTime	1637633463
STORAGE INFORMATION		
SerDe Library:	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe	
InputFormat:	org.apache.hadoop.mapred.TextInputFormat	
OutputFormat:	org.apache.hadoop.hive.ql.io.HivelgnoreKeyTextOutputFormat	

#### ii. 执行以下命令, 查询SQL。

SELECT country, count(\*) AS country count FROM user info test GROUP BY country;

如果回显正常,则表示Hive服务与**文件存储HDFS版**实例之间的连通性正常。

#### Spark服务验证

1. 使用CDH Spark中自带包spark-examples\_\*-cdh\*.jar从本地读取文件写入文件存储HDFS版实例。

```
spark-submit \
--master yarn \
--executor-memory 2G \
--executor-cores 2 \
--class org.apache.spark.examples.DFSReadWriteTest \
/opt/cloudera/parcels/CDH/jars/spark-examples_*-cdh*.jar \
/etc/profile /tmp/sparkwrite
```

2. 执行以下命令, 查看是否写入成功。

hadoop fs -ls dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/sparkwrite/\*

如果看到返回信息中包含如下信息,则表示写入成功。



HBase服务验证

• 数据迁移验证

查看迁移到文件存储HDFS版两张表的数据行数(表mock\_table\_0和表mock\_table\_1)。示例如下:

```
hbase(main):001:0> list
TABLE
mock table 0
mock table 1
2 row(s)
Took 0.3587 seconds
=> ["mock table 0", "mock table 1"]
hbase(main):002:0> count 'mock table 0'
Current count: 1000, row: 001637311398
. . . . . . . .
Current count: 100000, row: 991637311398
100000 row(s)
Took 7.2724 seconds
=> 100000
hbase(main):003:0> count 'mock table 1'
Current count: 1000, row: 001637311398
. . . . . . . .
Current count: 100000, row: 991637311398
100000 row(s)
Took 3.9399 seconds
=> 100000
```

#### • 新建数据表验证

i. 使用HBase Shell创建测试表(例如hbase\_test)并插入数据。示例如下:

```
hbase(main):001:0> create 'hbase_test','info'
Created table hbase test
Took 1.7688 seconds
=> Hbase::Table - hbase test
hbase(main):002:0> list
TABLE
hbase test
mock table 0
mock table 1
3 row(s)
Took 0.0147 seconds
=> ["hbase_test", "mock_table_0", "mock_table_1"]
hbase(main):003:0> put 'hbase_test','1', 'info:name' ,'Sariel'
Took 0.2415 seconds
hbase(main):004:0> put 'hbase test','1', 'info:age' ,'22'
Took 0.0053 seconds
hbase(main):005:0> put 'hbase test','1', 'info:industry','IT'
Took 0.0631 seconds
hbase(main):006:0> scan 'hbase_test'
ROW
                                                      COLUMN+CELL
1
                                                      column=info:age, timestamp=1637
636289233, value=22
                                                      column=info:industry, timestamp
1
=1637636299912, value=IT
1
                                                      column=info:name, timestamp=163
7636279553, value=Sariel
1 row(s)
Took 0.0227 seconds
```

ii. 在文件存储HDFS版上的HBase数据目录查看是否有新建的测试表。

[root@cdh6-m	iast	ter ~	]# hadoop :	fs -ls	dfs://f-			hangjiakou.dfs.aliyuncs.com:10290/hbase/data/default/	
Found 3 items									
drwxrwxrwx		root	root		2021-11-23	11:03	dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/hbase/data/default/hbase_test	
drwxrwxrwx		root	root		2021-11-22	16:03	dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/hbase/data/default/mock_table_0	
drwxrwxrwx		root	root		2021-11-22	16:03	dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/hbase/data/default/mock_table_1	

至此, CDH中的服务已经迁移到文件存储HDFS版。

### 常见问题

1. SecondaryNameNode服务停止。

根据文档操作步骤配置**文件存储HDFS版**后,CDH中的HDFS服务会报错,报错信息: SecondaryNameNode服务无法启动。这是由于SecondaryNameNode服务需要通过http get方式获取 NameNode的fsimage与edits文件进行工作。而**文件存储HDFS版**不提供HTTP服务,所以 SecondaryNameNode服务无法启动。此时,CDH内的Spark\Hive\Hbase服务已经使用阿里云**文件存储** HDFS版读写数据,CDH自身的HDFS服务问题不会影响系统运行和数据安全。

2. Hue服务文件浏览器不可用。

Hue是通过WebHDFS REST API访问HDFS上的文件。文件存储HDFS版暂不支持WebHDFS REST API,因此不可以在Hue上直接操作文件存储HDFS版实例上的文件。

### 后续步骤

#### 卸载并释放CDH6 HDFS服务使用的云盘

# 4.3. 卸载并释放CDH6 HDFS服务使用的云盘

本文介绍在配置CDH完成后,如何卸载并释放CDH HDFS服务使用的云盘。

### 前提条件

- 1. 已完成数据迁移。具体操作,请参见CDH6数据迁移。
- 2. 已配置CDH使用。具体操作,请参见配置CDH6使用文件存储HDFS版。

### 背景信息

当CDH已经成功运行在阿里云**文件存储HDFS版**上时,ECS挂载的云盘只用来存储运算中的临时Shuffle文件,可以选择卸载原来用于构建CDH HDFS服务的云盘,降低集群的拥有成本。

↓ 注意 从数据安全性考虑,数据迁移后建议进行数据完整性校验并让CDH系统在文件存储HDFS 版上正常运行一段时间后再卸载和释放云盘。云盘释放以后原有数据将无法找回。

# 操作步骤

- 1. 卸载数据盘。具体操作,请参见卸载数据盘。
- 2. 释放云盘。具体操作,请参见释放云盘。

# 5.在文件存储HDFS版上使用E-MapReduce

# 5.1. E-MapReduce数据迁移

本文介绍如何将E-MapReduce HDFS上的数据迁移到文件存储HDFS版。

# 背景信息

阿里云E-MapReduce是构建在阿里云云服务器ECS上的开源Hadoop、Spark、Hive、Flink生态大数据PaaS产品。提供用户在云上使用开源技术建设数据仓库、离线批处理、在线流式处理、即时查询、机器学习等场景下的大数据解决方案。

# 准备工作

1. 开通并创建E-MapRedece集群。具体操作,请参见创建集群。

⑦ 说明 当使用阿里云文件存储HDFS版替换E-MapReduce HDFS服务时,您可以选择使用高效 云盘、SSD云盘或者本地盘作为Shuffle数据的临时本地存储。关于存储规划的更多信息,请参 见:存储说明。

2. 开通**文件存储HDFS版**服务,并创建文件系统实例、添加挂载点,创建权限组和权限组规则。具体操作,请参见文件存储HDFS版快速入门。

⑦ 说明 配置挂载点时选择的专有网络和交换机要与E-MapReduce集群侧的配置保持一致。您可以通过以下方法获取:

i. 登录阿里云E-MapReduce控制台。

- ii. 在集群管理页面,找到需要挂载文件存储HDFS版的目标E-MapReduce集群,单击管理。
- iii. 单击**集群基础信息**,在网络信息区域中获取专有网络和交换机信息。

# 数据迁移

- 1. 登录阿里云E-MapReduce控制台。
- 2. 在集群管理页面,找到需要挂载文件存储HDFS版的目标E-MapReduce集群,单击管理。
- 3. 配置链接。
  - i. 选择集群服务 > HDFS, 单击配置。
  - ii. 在服务配置中,选择core-site,并单击自定义配置。

#### iii. 新增如下配置项, 单击**确定**。

- (必须配置)配置项fs.dfs.impl,其值为com.alibaba.dfs.DistributedFileSystem。
- (必须配置)配置项fs.AbstractFileSystem.dfs.impl,其值为com.alibaba.dfs.DFS。
- (可选)配置项io.file.buffer.size,其值为4194304。
- (可选)配置项dfs.connection.count,其值为1。

新增配置项			×
* Key	* Value	描述	操作
fs.dfs.impl	com.alibaba.dfs.Distribu		删除
fs.AbstractFileSystem.dfs	com.alibaba.dfs.DFS		删除
添加			
			确定取消

iv. 确认自定义配置成功后, 单击保存, 在确认保存对话框中, 输入执行原因, 单击确定。

首页 > 集群管理 > 集群(( )	) 、服务	HDFS	✓ 添加自定义配置	1成功,位于读文件配置项末	æ				
< 返回 正常 @ HDFS - 当前	集群:	/	'EMR-HDFS-SH	H-Cluster		9 查看	操作历史	Ⅰ 快捷链接	▼ ■ 操作 >
状态 部署拓扑 配置 配置修改	历史								
配置过滤	服务配置	置						? 部署	客户端配置保存
配置搜索:	全部	hdfs-site	kms-site	httpfs-site	hadoop-env	httpfs-env	core-site		自定义配置
请输入 Q			hadaan sa	aurity auth to loca					
配置范围:			nauoop.see	curity.autri_to_loca	RULE:[1:\$1]	:[2:\$1]			
集群默认配置 ~					DEFF				
配置类型									
基础配置 高级配置 只读配置									
数据路径 日志路径 日志相关									
JVM相关 数据相关									
数据库相关 性能相关									
时间相关 编解码相关									
OSS相关 地址端口 内存配置									R
磁盘相关 网络相关 文件路径			ipc.	client.idlethreshold	4000				0
URL或URI		ha	doop.proxyuse	r.flowagent.group	*				0

- v. 单击部署客户端配置, 在确认保存对话框中, 输入执行原因, 单击确定。
- 4. 执行以下命令将**文件存储HDFS版**最新的SDK包,放置到E-MapReduce HDFS服务存放jar包的路径下。 更多信息,下载最新版本SDK安装包。

cp ~/aliyun-sdk-dfs-1.0.2-beta.jar /opt/apps/ecm/service/hadoop/2.8.5-1.3.1/package/ hadoop-2.8.5-1.3.1/share/hadoop/hdfs/ 在E-MapReduce服务中,对应的路径为/opt/apps/ecm/service/hadoop/x.x.x-x.x.x/package/hadoo p-x.x.x-x.x.x/share/hadoop/hdfs。

⑦ 说明 集群中的每台机器都需要在相同位置添加该SDK包。

5. 暂停服务。

为了保证在数据迁移过程中数据不丢失,需要暂停数据处理服务(如: YARN服务、Hive服务、Spark服务、HBase服务等),HDFS服务仍需保持运行。此处以暂停Spark服务为例进行说明。

- i. 选择集群服务 > Spark。
- ii. 在页面右侧的操作栏中,单击停止All Components。
- iii. 在执行集群操作对话框中,填写执行原因,并单击确定。

当组件前面的图标变成红色,表示该组件服务已停止。直到Spark服务运行的组件全部停止后,再 进行其他服务的组件停止操作。无需关注xxx Client组件的状态。

< 返回 正常 😽 Spark 🔻	当前集群: C-262E3014187A6	51EE / EMR-HDFS-SH-C	luster	0	查看操作历史	Ⅰ 伊捷链接 ~	■操作 ∨
<u>状态</u> 部署拓扑 配置 間	配置修改历史						
组件列表							刷新
组件	正常	异常停止	手动停止	总数	警告	操作	
ThriftServer	0	0	1	1	-	重启 停止	
Spark Client	3	-	-	3	-		
SparkHistory	0	0	1	1	-	重启 停止	

ⅳ. 重复上述步骤,停止剩余服务。

⑦ 说明 只保留HDFS服务正常运行,以方便进行数据迁移。但是如果要迁移的数据量大,请 开启YARN服务,以便使用hadoop的数据迁移工具hadoop dist cp进行快速地数据迁移。

6. 迁移数据。

建议将/user、/hbase、/spark-history、/apps等服务目录和相关的数据目录全量迁移至**文件存储** HDFS版。

- 如果涉及将云下集群的数据迁移到云上,具体操作,请参见迁移开源HDFS的数据到文件存储HDFS 版。
- 如果E-MapReduce HDFS文件系统上的数据量较小,可以使用 hadoop fs -cp 命令进行数据迁移。

为了避免因为权限问题导致数据迁移失败,建议使用root用户执行命令。

hadoop fs -cp /user dfs://f-xxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com:10290/

其中 f-xxxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为您的文件存储HDFS版挂载点域名,请 根据实际情况进行修改。

 如果E-MapReduce HDFS文件系统上的数据量较大,需要使用数据迁移工具hadoop distcp进行数据 迁移。具体操作,请参见迁移开源HDFS的数据到文件存储HDFS版。

hadoop distcp hdfs://emr-header-1.cluster-xxxx:9000/ dfs://f-xxxxxxxxxxxxxxxx.cn-xxxx
xxx.dfs.aliyuncs.com:10290/

其中 f-xxxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为您的文件存储HDFS版挂载点域名,需要根据实际情况进行修改。

# 后续步骤

配置E-MapReduce服务使用文件存储HDFS版

# 5.2. 配置E-MapReduce服务使用文件存储HDFS

# 版

本文介绍如何配置E-MapReduce上的HDFS服务、HIVE服务、SPARK服务、HBase服务来使用**文件存储HDFS**版。

# 前提条件

已完成数据迁移。具体操作,请参见E-MapReduce数据迁移。

# 配置HDFS服务

- 1. 登录阿里云E-MapReduce控制台。
- 2. 在集群管理页面,找到需要挂载文件存储HDFS版的目标E-MapReduce集群,单击管理。
- 3. 更改配置。
  - i. 选择集群服务 > HDFS, 单击配置。
  - ii. 在服务配置中, 单击core-site。
  - iii. 找到配置项fs.defaultFS,将其值替换为您的**文件存储HDFS版**挂载点域名(dfs://fxxxxxxxxxxx.cn-xxxxx.dfs.aliyuncs.com:10290)。

全部	hdfs-site	kms-site	httpfs-site	hadoop-env	httpfs-env	core-site		自定义配置
		hadoop.pro:	xyuser.hue.group	s *			?	
			fs.du.interva	600000			?	
	hadoo	p.security.auth	entication.use.ha	s true				
		io.co	mpression.codec	s com.hadoop.c	compression.lzo.l	.zoCodec,com.h	nadoop.coi	
io.serializations				s org.apache.ha	doop.io.serialize	r.WritableSeriali	ization,org	?
			fs.defaultF	S dfs://	topic tay	dfs.aliyuncs	.com:1029	2 0

- iv. 单击**保存**,在确认保存对话框中,输入执行原因,单击确定。
- v. 单击部署客户端配置,在确认保存对话框中,输入执行原因,单击确定。
- 4. 重启YARN服务。
  - i. 选择**集群服务 > YARN**。
  - ii. 在页面右侧的操作栏中,单击重启All Components。

# 配置Hive服务

#### ? 说明

配置HDFS服务完成后,才能配置Hive服务。

在配置Hive服务之前,请确认/*user/hive/*目录中的数据已完成全量迁移。具体操作,请参见迁移开源 HDFS的数据到文件存储HDFS版。

#### 1. 更改配置。

- i. 选择集群管理 > HIVE, 单击配置。
- ii. 在服务配置中, 单击hive-site。
- iii. 找到配置项hive.metastore.warehouse.dir, 删除其对应值中的E-MapReduce HDFS文件系统域名, 只保留/user/hive/warehouse。

hive.opt	imize.dynamic.partition.hashjoin	false	?
	hive.groupby.skewindata	false	?
	hive.metastore.warehouse.dir	/user/hive/warehouse	2
	hive.compactor.initiator.on	false	-
	hive.execution.engine	spark	2

- iv. 单击保存, 在确认保存对话框中, 输入执行原因, 单击确定。
- v. 单击部署客户端配置, 在确认保存对话框中, 输入执行原因, 单击确定。
- 2. 修改元数据。
  - i. 在hivemetastore-site中, 获取数据库相关信息。
    - 在配置项javax.jdo.option.ConnectionURL中,获取MySQL服务的主机名和元数据存储的数据 库。
    - 在配置项javax.jdo.option.ConnectionUserName中,获取MySQL服务的用户名。
    - 在配置项javax.jdo.option.ConnectionPassword中,获取MySQL服务的用户密码。
  - ii. Hive的元数据存储在MySQL,进入存储Hive元数据的MySQL数据库hivemeta,修改CTLGS表、DBS 表和SDS表相应的值。
    - a. 执行 use hivemeta 命令,进入存储Hive元数据的MySQL数据库hivemeta。

#### b. 修改表CTLGS中的数据。

a. 执行 select \* from CTLGS 命令, 查询表CTLGS中的数据。返回结果示例如下:

+----+
+ ----+
CTLG\_ID | NAME | DESC | LOCATION\_URI
|
+----+
----+
| 1 | hive | Default catalog, for Hive | hdfs://emr-header-1.cluster125428:9000/user/hive/warehouse |
+----+
1 row in set (0.00 sec)

b. 修改LOCATION\_URI为文件存储HDFS版的挂载点域名,请根据实际情况进行修改。

修改CTLG\_ID为1的LOCATION\_URI,示例如下:

```
MariaDB [hivemeta]> UPDATE CTLGS
    -> SET LOCATION_URI = 'dfs://f-xxxxxxxxxx.cn-xxxxx.dfs.ali
yuncs.com:10290/user/hive/warehouse'
    -> WHERE CTLG_ID = 1;
```

#### 若返回如下信息,则表示修改成功。

Query OK, 0 rows affected (0.00 sec) Rows matched: 1 Changed: 0 Warnings: 0

c. 修改表DBS中的数据。

- a. 执行 select \* from DBS 命令, 查询表DBS中的数据。返回结果示例如下:
  - +----+ | DB\_ID | DESC | DB LOCATION URI | OWNER NAME | OWNER TYPE | CTLG NAME | | NAME +--------------+ 1 | Default Hive database | hdfs://emr-header-1.cluster-125428:9000/u 1 | default ser/hive/warehouse public | ROLE | hive | | 2 | NULL | hdfs://emr-header-1.cluster-125428:9000/u ser/hive/warehouse/analysis\_logs.db | analysis\_logs | root | USER | hive | | 3 | NULL | hdfs://emr-header-1.cluster-125428:9000/u ser/hive/warehouse/analysis\_logs\_report.db | analysis\_logs\_report | root | USER | hive | | 4 | NULL | hdfs://emr-header-1.cluster-125428:9000/u ser/hive/warehouse/analysis logs report old.db | analysis logs report old | root | USER | hive | +-----\_\_\_\_\_ 4 rows in set (0.00 sec)
- b. 修改DB\_LOCATION\_URI为**文件存储HDFS版**的挂载点域名,请根据实际情况进行修改。

修改DB\_ID为1的DB\_LOCATION\_URI, 示例如下:

```
MariaDB [hivemeta]>UPDATE DBS
    -> SET    DB_LOCATION_URI = 'dfs://f-xxxxxxxxxx.cn-xxxxx.dfs.ali
yuncs.com:10290/user/hive/warehouse'
    -> WHERE    DB_ID = 1;
```

若返回如下信息,则表示修改成功。

Query OK, 1 row affected (0.00 sec) Rows matched: 1 Changed: 1 Warnings: 0

- d. 修改表SDS中的数据。
  - a. 执行 select \* from SDS 命令, 查询表SDS中的数据。返回结果示例如下:

\_\_\_\_\_ \_\_\_\_\_ 1 | org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFo 1 1 | | hdfs://emr-header-1.clus rmat | ter-125428:9000/user/hive/warehouse/analysis logs.db/original log bj partit | -1 | org.apache.hadoop.hive.ql.io ioned .parquet.MapredParquetOutputFormat | 1 | | 2 | 2 | org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFo | hdfs://emr-header-1.clus 1 rmat | ter-125428:9000/user/hive/warehouse/analysis\_logs.db/original\_log\_hz\_partit | -1 | org.apache.hadoop.hive.ql.io ioned .parquet.MapredParquetOutputFormat | 2 | 3 | 3 | org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFo rmat | | | hdfs://emr-header-1.clus ter-125428:9000/user/hive/warehouse/analysis logs.db/original log sh partit ioned | -1 | org.apache.hadoop.hive.ql.io .parquet.MapredParquetOutputFormat | 3 | | 29 | 22 | org.apache.hadoop.mapred.TextInputFormat 1 | hdfs://emr-header-1.cluster-1 25428:9000/user/hive/warehouse/analysis logs report.db/hz writethroughput t op\_daily | -1 | org.apache.hadoop.hive.ql.io.Hive IgnoreKeyTextOutputFormat | 29 | . . . . . . . . 548 | 1 80 | org.apache.hadoop.mapred.TextInputFormat 1 1 | hdfs://emr-header-1.cluster-1 25428:9000/user/hive/warehouse/analysis\_logs\_report.db/hz\_readthroughput\_to p\_yearly | -1 | org.apache.hadoop.hive.ql.io.Hive 548 I IgnoreKeyTextOutputFormat 549 81 org.apache.hadoop.mapred.TextInputFormat | hdfs://emr-header-1.cluster-1 1 25428:9000/user/hive/warehouse/analysis logs report old.db/hz writethroughp ut\_top\_yearly\_20190709 | -1 | org.apache.hadoop.hive.ql.io.Hive 549 I IgnoreKeyTextOutputFormat 1 | 550 | 82 | org.apache.hadoop.mapred.TextInputFormat 1 | hdfs://emr-header-1.cluster-1 25428:9000/user/hive/warehouse/analysis\_logs\_report.db/hz\_writethroughput\_t op yearly | -1 | org.apache.hadoop.hive.ql.io.Hive IgnoreKeyTextOutputFormat 550 | \_\_\_\_\_ \_\_\_\_\_ 536 rows in set (0.00 sec)

b. 修改LOCATION属性中的值为**文件存储HDFS版**的挂载点域名,请根据实际情况进行修改。

#### 修改SD\_ID为1的LOCATION, 示例如下:

MariaDB [hivemeta]> UPDATE SDS SET LOCATION = "dfs://f-xxxxxxxxxxxxxxxx.c n-xxxxxx.dfs.aliyuncs.com:10290/user/hive/warehouse/analysis\_logs.db/origi nal\_log\_bj\_partitioned" WHERE SD\_ID = 1;

若返回如下信息,则表示修改成功。

Query OK, 0 rows affected (0.01 sec) Rows matched: 1 Changed: 0 Warnings: 0

3. 在页面右侧的操作栏,单击重启All Components,重启服务。

#### 配置Spark服务

#### ? 说明

配置HDFS服务完成后,才能配置Spark服务。

配置Spark服务前,请确认/*spark-hist ory*目录中的数据已经完成了全量迁移。具体操作,请参见<del>迁移开源</del> HDFS的数据到文件存储HDFS版。

#### 1. 更改配置。

- i. 选择集群服务 > Spark, 单击配置。
- ii. 在服务配置中, 单击spark-defaults。
- iii. 找到配置项spark\_eventlog\_dir,将其对应的值替换为您文件存储HDFS版挂载点域名。

	spark_eventlog_dir	dfs://i.dfs.aliyuncs.com:1029	2
spark.fil	es.openCostInBytes	4194304	?
spark.memo	ory.useLegacyMode	false	?
:	spark.history.ui.port	18080	
spark.executor.logs	.rolling.time.interval	daily	?
spark.sp	eculation.multiplier	1.5	?

- iv. 单击保存, 在确认保存对话框中, 输入执行原因, 单击确定。
- v. 单击部署客户端配置, 在确认保存对话框中, 输入执行原因, 单击确定。
- 2. 放置SDK包。

将**文件存储HDFS版**的SDK包(aliyun-sdk-dfs-1.0.2-beta.jar),放置到E-MapReduce Spark服务存放 jar包的目录下。

```
cp ~/aliyun-sdk-dfs-1.0.2-beta.jar /opt/apps/ecm/service/spark/2.4.3-1.0.0/package/s
park-2.4.3-1.0.0-bin-hadoop2.8/jars/
```

一般情况下,放置到/opt/apps/ecm/service/spark/x.x.x-x.x.x/package/spark-x.x.x-x.x.sbin-hadoo px.x/jars目录。

⑦ 说明 集群中的每台机器都需要添加该SDK包。

3. 在页面右侧的操作栏,单击重启All Components,重启服务。

# 配置HBase服务

#### ? 说明

配置HDFS服务完成后,才能配置HBase服务。

配置HBase服务前,请确认/hbase目录中的数据已经完成了全量迁移。具体操作,请参见迁移开源HDFS的数据到文件存储HDFS版。

#### 1. 更改配置。

- i. 选择集群服务 > Hbase, 单击配置。
- ii. 在服务配置中, 单击Hbase-site。
- iii. 找到配置项hbase.rootdir, 删除其对应值中的E-MapReduce HDFS文件系统域名, 只保留/hbase。

hbase.hstore.useEx	ploringCompation	true	
hbase.coprocessor.enabled		true	•
	hbase.rootdir	/hbase	∠ ?
hbase.zookeeper.dns.nameserver		default	?
hbase.hregior	.majorcompaction	864000000	?

iv. 单击保存,在确认保存对话框中,输入执行原因,单击确定。

v. 单击部署客户端配置, 在确认保存对话框中, 输入执行原因, 单击确定。

2. 在页面右侧的操作栏,单击重启All Components,重启服务。

#### 关闭HDFS服务

② 说明 关闭HDFS服务前,请确认原来E-MapReduce HDFS上存储的数据都已经迁移到文件存储 HDFS版。具体操作,请参见迁移开源HDFS的数据到文件存储HDFS版。

- 1. 选择集群服务 > HDFS。
- 2. 在页面右侧的操作栏,单击重启All Components,重启服务。
- 3. 在执行集群操作对话框中,输入执行原因,单击确定。

#### 验证服务正确性

• hadoop的验证

使用E-MapReduce hadoop中自带的测试包hadoop-mapreduce-examples-2.x.x.jar进行测试。该测试包 默认放置在/opt/apps/ecm/service/hadoop/2.x.x-1.x.x/package/hadoop-2.x.x-1.x.x/share/hadoop/ mapreduce/目录下。

i. 执行以下命令,在/tmp/randomtextwriter目录下生成128 MB大小的文件。

```
hadoop jar /opt/apps/ecm/service/hadoop/2.8.5-1.3.1/package/hadoop-2.8.5-1.3.1/share/
hadoop/mapreduce/hadoop-mapreduce-examples-2.8.5.jar randomtextwriter -D mapreduce.
randomtextwriter.totalbytes=134217728 -D mapreduce.job.maps=2 -D mapreduce.job.reduc
es=2 /tmp/randomtextwriter
```

其中 hadoop-mapreduce-examples-2.8.5.jar 为E-MapReduce的测试包,请根据实际情况修改。

ii. 执行以下命令验证文件是否生成成功,从而验证文件系统实例的连通性。

hadoop fs -ls dfs://f-xxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com:10290/tmp/randomtex
twriter

其中 f-xxxxxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com 为您的文件存储HDFS版挂载点域名,请 根据实际情况修改。

■ 如果看到\_SUCCESS和part-m-00000两个文件, 表示连通成功。如下所示:

```
-rwxrwxrwx 3 root root 0 2019-07-26 11:05 dfs://f-xxxxxxxxxx.cn-xxx
xxxx.dfs.aliyuncs.com:10290/tmp/randomtextwriter2/_SUCCESS
-rwxrwxrwx 3 root root 137774743 2019-07-26 11:05 dfs://f-xxxxxxxxxxxx.cn-xxx
xxxx.dfs.aliyuncs.com:10290/tmp/randomtextwriter2/part-m-000000
```

- 如果连通失败(例如,报错No such file or directory),请排查连通性问题。具体操作,请参见创建文件系统实例后,为什么无法访问文件存储HDFS版?
- Spark的验证

使用E-MapReduce Spark中自带的测试包spark-examples\_2.x-2.x.x.jar进行测试。该测试包默认放置在/o pt/apps/ecm/service/spark/2.x.x-1.0.0/package/spark-2.x.x-1.0.0-bin-hadoop2.8/examples/jars下。

i. 执行以下命令, 在/tmp/randomtextwriter 目录下生成128M大小的文件。

```
hadoop jar /opt/apps/ecm/service/hadoop/2.8.5-1.3.1/package/hadoop-2.8.5-1.3.1/share/
hadoop/mapreduce/hadoop-mapreduce-examples-2.8.5.jar randomtextwriter -D mapreduce.
randomtextwriter.totalbytes=134217728 -D mapreduce.job.maps=2 -D mapreduce.job.reduc
es=2 /tmp/randomtextwriter
```

其中 hadoop-mapreduce-examples-2.8.5.jar 为E-MapReduce的测试包,请根据实际情况修改。

#### ii. 使用spark测试包从文件存储HDFS版上读取测试文件并按照word count的格式展示。

```
spark-submit --master yarn --executor-memory 2G --executor-cores 2 --class org.apa
che.spark.examples.JavaWordCount /opt/apps/ecm/service/spark/2.4.3-1.0.0/package/spa
rk-2.4.3-1.0.0-bin-hadoop2.8/examples/jars/spark-examples_2.11-2.4.3.jar /tmp/rando
mtextwriter
```

如果回显信息类似如下图所示,表示配置成功。



- Hive的验证
  - i. 执行以下命令进入Hive命令界面。

```
[root@hadoop9 ~]# hive
Hive Session ID = 9dlaeaf3-19d8-461b-952f-6fcfed900e69
Logging initialized using configuration in file:/etc/ecm/hive-conf-3.1.1-1.1.6/hive-l
og4j2.properties Async: true
Hive Session ID = 9c564244-96b1-4865-8e0f-21631a2e97f1
hive>
```

### ii. 执行以下命令创建测试表。

```
hive> create table default.testTable(id int , name string ) row format delimited f
ields terminated by '\t' lines terminated by '\n';
OK
Time taken: 0.536 seconds
```

#### iii. 执行以下命令查看测试表。

如果回显信息中的Location属性对应的值为**文件存储HDFS版**的路径,则表示配置Hive成功。如果不 是,请重新配置。具体操作,请参见<mark>配置Hive服务</mark>。

```
hive> desc formatted default.testTable ;
OK
2019-07-26 11:23:25,133 INFO [9dlaeaf3-19d8-461b-952f-6fcfed900e69 main] mapred.File
InputFormat: Total input files to process : 1
# col name
                       data type
                                              comment
id
                       int
name
                       string
# Detailed Table Information
Database:
                      default
                      USER
OwnerType:
Owner:
                      root
CreateTime: Fri Jul 26 11:23:12 CST 2019
LastAccessTime: UNKNOWN
Retention:
                      0
Location:
                      dfs://f-xxxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com:10290/use
r/hive/warehouse/testtable
Table Type:
                      MANAGED TABLE
Table Parameters:
       COLUMN STATS ACCURATE {\"BASIC STATS\":\"true\",\"COLUMN STATS\":{\"id\":\"
true\",\"name\":\"true\"}}
       bucketing version
                               2
       numFiles
                               0
       numRows
                              0
       rawDataSize
                              0
       totalSize
                               0
       transient_lastDdlTime 1564111392
# Storage Information
SerDe Library:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDeInputFormat:org.apache.hadoop.mapred.TextInputFormat
                   org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
OutputFormat:
Compressed:
                      No
                       -1
Num Buckets:
Bucket Columns:
                       []
Sort Columns:
                      []
Storage Desc Params:
                               \t
       field.delim
       line.delim
                                ∖n
       serialization.format
                               \uparrow +
Time taken: 0.134 seconds, Fetched: 34 row(s)
```

- HBase的验证
  - i. 执行 hbase shell 命令进入hbase shell命令界面。
  - ii. 在HBase中创建测试表。

```
hbase(main):001:0> create 'hbase_test','info'
0 row(s) in 1.5620 seconds
=> Hbase::Table - hbase_test
hbase(main):002:0> put 'hbase_test','l', 'info:name' ,'Sariel'
0 row(s) in 0.2990 seconds
hbase(main):003:0> put 'hbase_test','l', 'info:age' ,'22'
0 row(s) in 0.0230 seconds
hbase(main):004:0> put 'hbase_test','l', 'info:industry' ,'IT'
0 row(s) in 0.0740 seconds
```

iii. 执行以下命令查看**文件存储HDFS版**的/hbase/data/default/路径,如果/hbase/data/default/路 径下有hbase\_test目录,则证明配置链接成功。

hadoop fs -ls /hbase/data/default

]# hadoop fs -ls /hbase/data/default

Found 2 items drwxrwxrwx - root root 0 1970-01-01 08:00 /hbase/data/default/Noi.ine\_MAXThroughput drwxrwxrwx - root root 0 1970-01-01 08:00 /hbase/data/default/hbase\_test

# 后续步骤

卸载并释放E-MapReduce HDFS使用的云盘

# 5.3. 卸载并释放E-MapReduce HDFS使用的云 盘

本文介绍在配置E-MapReduce完成后,如何卸载并释放E-MapReduce HDFS服务使用的云盘。

#### 前提条件

- 1. 已完成数据迁移。具体操作,请参见E-MapReduce数据迁移。
- 2. 已配置E-MapReduce使用**文件存储HDFS版**。具体操作,请参见配置E-MapReduce服务使用文件存储 HDFS版。
- 3. 在卸载磁盘前,请停止E-MapReduce集群中的所有服务,等到卸载磁盘操作完成后再启动。

### 背景信息

当E-MapReduce已经成功运行在阿里云**文件存储HDFS版**上时,ECS挂载的云盘只用来存储运算中的临时 Shuffle文件,可以选择卸载原来用于构建E-MapReduce HDFS服务的云盘,降低集群的拥有成本。

○ 注意 从数据安全性考虑,数据迁移后建议进行数据完整性校验并让E-MapReduce系统在文件存储HDFS版上正常运行一段时间后再卸载和释放云盘。云盘释放以后原有数据将无法找回。集群中的每台机器至少需要保留一块数据盘,通常是/mnt/disk1上挂载的磁盘。

### 操作步骤

- 1. 卸载数据盘。具体操作,请参见卸载数据盘。
- 2. 释放云盘。具体操作,请参见释放云盘。

# 6.性能调优

# 6.1. 性能优化最佳实践

您可以通过调整core-site.xml配置、TestDFSIO配置或避免使用小文件来优化集群性能。

• 建议一: 调整core-site.xml配置

```
在测试集群吞吐性能之前建议在core-site.xml文件中增加或修改如下配置,同步到所有依赖hadoop-common的节点上并重启集群服务。
```

```
<property>
     <name>alidfs.default.write.buffer.size</name>
    <value>8388608</value>
    <description>To achieve high write throughput, no less than 1MB, no more than 8MB</d>
escription>
</property>
<property>
    <name>alidfs.default.read.buffer.size</name>
    <value>8388608</value>
     <description>To achieve high read throughput, no less than 1MB, no more than 8MB</de</pre>
scription>
</property>
<property>
     <name>alidfs.use.buffer.size.setting</name>
    <value>false</value>
</propertv>
<property>
     <name>dfs.connection.count</name>
     <value>16</value>
    <description>If multi threads in the same process will read/write to DFS, set to cou
nt of threads</description>
</property>
```

- alidfs.use.buffer.size.setting:该值配置成false表示文件存储HDFS版使用 alidfs.default.write.buffer.size和alidfs.default.read.buffer.size配置的值。该值配置成true表示文件 存储HDFS版使用hadoop配置的值。
- o alidfs.default.write.buffer.size:表示写缓存区的大小,单位为Byte。适当的缓存大小可以提供更高的 吞吐,建议配置在1 MB和8 MB之间。
- o alidfs.default.read.buffer.size:表示读缓存区的大小,单位Byte。适当的缓存大小可以提供更高的吞
   吐,建议配置在1 MB和8 MB之间。
- 。 dfs.connection.count:表示单SDK内的连接池数目,建议配置为16。
- 建议二:调整TestDFSIO配置

在使用TestDFSIO测试集群吞吐性能时建议您将-nrFiles参数值设置为500。

-nrFiles: 该参数表示读/写文件的个数和测试TestDFSIO时的并发度,提高并发度可以更好的测试集群的 吞吐性能,该参数建议配置为500。

• 建议三: 尽量避免使用小文件

处理小文件并非Hadoop的设计目标,Hadoop分析引擎处理大量小文件的速度远远小于处理同等数据量的大文件的速度。每一个小文件都会占用一个task,而task启动将耗费大量时间,造成作业的大部分时间都耗费在启动task和释放task上。将存储在**文件存储HDFS版**上的小文件聚合成大文件会对整体的分析性能有较大帮助。

# 6.2. 集群吞吐性能测试

本文介绍集群在顺序写、顺序读、随机写等方面的性能测试方法。

### 注意事项

性能测试前,请注意以下事项。

- 吞吐最大不会超过ECS带宽。如果您的ECS带宽只有 1.5 Gbps,则吞吐最高可达到187.5 MB/s。
- 文件存储HDFS版的吞吐能力和购买的存储空间相关。
- 测试集群吞吐性能所使用的TestDFSIO是一个分布式任务,存在任务调度及结果汇总阶段,计算集群吞吐 均值时会略低于**文件存储HDFS版**吞吐限速。
- 在进行顺序读与随机读的测试之前需要确保**文件存储HDFS版**之上已有指定的待测数据,如果没有待测数 据请使用顺序写先生成待测数据再进行顺序读与随机读的测试。

### 测试环境

配置名称	配置说明
	CPU核数: 4核
计值//M型器	内存: 16 GB
り 异 V M 比 直	机器数量: 6台
	网络带宽: 1.5 Gbps
文 <i>바 方 랴</i> 니DEC ᄠ ᆔ 풀	实例大小: 10 TB
义计仔细印53版配直	吞吐限速: 1000 MB/s
软件配置	Apache Hadoop: Hadoop 2.7.6

### 测试工具

Test DFSIO是Hadoop系统自带的基准测试组件,用于测试DFS的IO吞吐性能。

TestDFSIO的jar包位于开源hadoop版本的\$HADOOP\_HOME/share/hadoop/mapreduce目录下,其中 \$HADOOP\_HOME为测试机器中的Hadoop安装目录,jar包名为hadoop-mapreduce-client-jobclient-x.x.xtests.jar,TestDFSIO使用方法如下所示。

```
[root@nodel bin]# ./hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-
2.7.6-tests.jar TestDFSIO
Usage: TestDFSIO [genericOptions] -read [-random | -backward | -skip [-skipSize Size]] | -w
rite | -append | -truncate | -clean [-compression codecClassName] [-nrFiles N] [-size Size[
B|KB|MB|GB|TB]] [-resFile resultFileName] [-bufferSize Bytes] [-rootDir]
```

# 顺序写性能测试

预估值: 1000 MB/s

1. 将文件写入到文件存储HDFS版。

将500个大小为4 GB的文件按顺序写入到**文件存储HDFS版**,读写数据的缓存大小为8 MB,并将统计数据写入 /tmp/TestDFSIOwrite.log中。

./hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.6-tests.ja
r TestDFSIO -write -nrFiles 500 -size 4GB -bufferSize 8388608 -resFile /tmp/TestDFSIOwr
ite.log

2. 执行以下命令查看文件存储HDFS版已写入的文件数量及大小。

./hadoop fs -count -q -h /benchmarks/TestDFSIO/io\_data

3. 执行以下命令查看生成的统计信息文件。

cat /tmp/TestDFSIOwrite.log

测试结果如下所示。

- Total MBytes processed: 写入的文件的总大小。
- Test exectime sec: 消耗的时长。



4. 计算集群吞吐。

您可以通过以下两种方式计算集群吞吐。

• 通过统计信息文件计算吞吐。

集群的吞吐约为Total MBytes processed / Test exec time sec = 985.16 MB/s

○ 通过ECS监控的带宽流出速率,计算集群吞吐。

进行顺序写时集群中每个节点的带宽流出速率大约为1.4 Gb/s、1.4 Gb/s、1.4 Gb/s、1.1 Gb/s、1.1 Gb/s、1.1 Gb/s、1.4 Gb/s。

集群吞吐=(1.4+1.4+1.4+1.1+1.1+1.45)\*1000÷8=981.25 MB/s

云服与器CS(bits/s)	云服务器ECS(bits/s)	云服約器ECS(bits/s)		
1.576 1.405 953.67M 476.84M 0.00 20:10:15 20:30:00 20:46:40 20:59:00 ● (Apert) Host.netout.rate一平均圈—apsara-hd	1.536 1.466 ///////////////////////////////////	1.556         1.406         1.4           953.67M         1.4           476.84M         20:30:00           20:10:15         20:30:00           20:46:40         20:59:00           ● (Agent) Host netout.rate=\\$\$\$#\$\$\$#\$\$\$#\$\$\$#\$\$\$#\$\$\$\$		
云服务器ECS(bits/s)	云服务器ECS(bits/s)	云服务器ECS(bits/s)		
1.416 953.67M 476.84M	1.49G 953.67M 476.84M	1.566 1.406 953.67M 476.84M		
20:10:15 20:30:00 20:46:40 20:59:00 ④ (Agent) Host.netout.rate一平均值—apsara-hd	20:10:15 20:30:00 20:46:40 20:59:00 (Agent) Host.netout.rate—平均值—apsara-hd	20:10:15 20:30:00 20:46:40 20:59:00 ● (Agent) Host.netout.rate—平均值—apsara-hd		

# 顺序读性能测试

预估值: 1000 MB/s

1. 按顺序读取文件存储HDFS版上的文件。

顺序读取**文件存储HDFS版**上面500个大小为4 GB的文件,读写数据的缓存大小为8 MB,并将统计数据 写入/*tmp/TestDFSIOread.log*中。

```
./hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.6-tests.ja
r TestDFSIO -read -nrFiles 500 -size 4GB -bufferSize 8388608 -resFile /tmp/TestDFSIOrea
d.log
```

2. 执行以下命令查看生成的统计信息文件。

cat /tmp/TestDFSIOread.log

#### 测试结果如下所示。

- Total MBytes processed: 写入的文件的总大小。
- Test exec time sec: 消耗的时长。

read				
Tue Aug	27	21:24:15	CST	2019
500				
2048000				
47.03				
47.16				
2.56	_			
2091.61				
	read Tue Aug 500 2048000 47.03 47.16 2.56 2091.61	read Tue Aug 27 500 2048000 47.03 47.16 2.56 2091.61	read Tue Aug 27 21:24:15 500 2048000 47.03 47.16 2.56 2091.61	read Tue Aug 27 21:24:15 CST 500 2048000 47.03 47.16 2.56 2091.61

3. 计算集群吞吐。

您可以通过以下两种方式计算集群吞吐。

。 通过统计信息文件计算吞吐。

集群的吞吐约为Total MBytes processed / Test exec time sec = 989.15 MB/s

• 通过ECS监控的带宽流出速率,计算集群吞吐。

进行顺序读时集群中每个节点的带宽流入速率大约为1.4 Gb/s, 1.45 Gb/s, 1.1 Gb/s, 1.4 Gb/s, 1.45 Gb/s, 1.1 Gb/s, 1.4

集群吞吐= (1.4+1.45+1.1+1.4+1.45+1.1) \*1000÷8=987.5 MB/s

云服务器ECS(bits/s)	云舰务器ECS(bits/s)	云服务鞧ECS(bits/s)
1.385 953.67M 476.84M 20:40:15 21:03:20 21:20:00 21:39:00 ● (Agent) Host.netin.rate一平均值—apsara-hdf	1.576 1.406 953.67M 476.84M 0.00 20:40:15 21:03:20 21:20:00 21:39:00 ● (Agent) Host.netin.rate一平均面apsara-hdf	1.50G 953.67M 476.84M 0.00 20:40:15 21:03:20 21:20:00 21:20:00 21:39:00 (Agent) Host neth.rate—T+5fff—apsara-hdf
云振务器ECS(bits/s)	云服务器ECS(bits/s)	云服务器ECS(bits/s)
1.50G         1.700           953.67M         1.4           476.84M         0.00           20040:15         21:03:20         21:20:00         21:39:00	1.576 1.405 953.67M 476.84M 0.00 20/40:15 21:03:20 21:20:00 21:39:00	1.546 953.67M 476.84M 0.00 20:90:15 21:03:20 21:20:00 21:20
● (Agent) Host.netin.rate—平均值—apsara-hdf	(Agent) Host.netin.rate—平均值—apsara-hdf	● (Agent) Host.netin.rate—平均值—apsara-hdf

### 随机读性能测试

预估值: 1000 MB/s

1. 随机读取文件存储HDFS版上的文件。

随机读取**文件存储HDFS版**上面的大小为1 GB的500个文件,读写数据的缓存大小为8 MB,并将统计数据写入/*tmp/TestDFSIOrandomread.log*中。

```
./hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.6-tests.ja
r TestDFSIO -read -random -nrFiles 500 -size 1GB -bufferSize 8388608 -resFile /tmp/Test
DFSIOrandomread.log
```

2. 执行以下命令查看生成的统计信息文件。

cat /tmp/TestDFSIOrandomread.log

#### 测试结果如下所示。

- 。 Total MBytes processed: 写入的文件的总大小。
- Test exec time sec: 消耗的时长。



3. 计算集群吞吐。

您可以通过以下两种方式计算集群吞吐。

• 通过统计信息文件计算吞吐。

集群的吞吐约为Total MBytes processed / Test exec time sec = 962.07 MB/s

○ 通过ECS监控的带宽流出速率, 计算集群吞吐。

进行随机读时集群中每个节点的带宽流入速率大约为1.35 Gb/s, 1.2 Gb/s, 1.4 Gb/s, 1.1 Gb/s, 1.4 Gb/s, 1.4 Gb/s, 1.4 Gb/s, 1.4 Gb/s, 1.35 Gb/s

集群吞吐= (1.35+1.2+1.4+1.1+1.4+1.35) \*1000÷8=975 MB/s



# 6.3. 元数据性能测试

本文介绍元数据在create\_write、open\_read、rename和delete等操作上的性能测试方法。

# 注意事项

在进行open\_read、rename、delete测试之前要确保**文件存储HDFS版**上已存在待测数据。如果不存在待测数据,请先使用create\_open创建待测数据。

# 测试环境

配置名称	配置说明		
	CPU核数: 4核		
计算VM配置	内存: 16 GB		
	机器数量: 6台		
	网络带宽: 1.5 Gbps		
文件友健HDES版型署	实例大小: 10 TB		
义计仔细印0-3放乱直	吞吐限速: 1000 MB/s		
软件配置	Apache Hadoop: Hadoop 2.7.6		

# 测试工具

NNbench是Hadoop系统自带的基准测试组件, NNbench用于测试元数据操作的性能。

NNbench的jar包位于\$HADOOP\_HOME/share/hadoop/mapreduce目录下,\$HADOOP\_HOME为测试机器中的Hadoop 安装目录,NNbench的jar包名为hadoop-mapreduce-client-jobclient-x.x.x-tests.jar,使用方法如下。

[root@nodel bin]# ./hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.6-tests.jar nnbench Usage: nnbench <options> Options: -operation <可用操作是create\_write open\_read rename delete.必选项> -maps <map数量.默认值为1。非必选项> -reduces <reduce数量.默认值为1.非必选项> -blockSize <块大小 (以字节为单位) .默认值为1.非必选项> -bytesToWrite <要写入的字节数.默认值为0.非必选项> -bytesPerChecksum <文件的每个校验和的字节数.默认值为1.非必选项> -numberOfFiles <要创建的文件数.默认值为1.非必选项> -baseDir <DFS存储路径.默认是/becnhmarks/NNBench.非必选项> -readFileAfterOpen <true或false.如果为true,则读取文件并报告平均读取时间.默认为false.非必选项>

#### create\_write性能测试

1. 使用20个map, 10个reduce的MapReduce作业, 创建1000个文件。

./hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.6-tests.ja
r nnbench -maps 20 -reduces 10 -numberOfFiles 1000 -operation create\_write

2. 执行如下命令查看在**文件存储HDFS版**上面创建的文件数量及大小。

./hadoop fs -count -q -h /benchmarks/NNBench/io\_data

3. 查看生成的统计信息。

统计信息在当前测试目录下的NNBench\_results.log文件中。

TPS表示每秒处理的事务个数,测试结果为1987。

<pre>NNBench</pre>
Version: NameNode Benchmark 0.4 Date & time: 2019-08-28 15:23:56,949 Test Operation: create_write Start time: 2019-08-28 15:23:44,972 Maps to run: 20 Reduces to run: 10 Block Size (bytes): 1 Bytes to write: 0 Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Date & time: 2019-08-28 15:23:56,949 Test Operation: create_write Start time: 2019-08-28 15:23:44,972 Maps to run: 20 Reduces to run: 10 Block Size (bytes): 1 Bytes to write: 0 Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Test Operation: create_write Start time: 2019-08-28 15:23:44,972 Maps to run: 20 Reduces to run: 10 Block Size (bytes): 1 Bytes to write: 0 Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Test Operation: create_write Start time: 2019-08-28 15:23:44,972 Maps to run: 20 Reduces to run: 10 Block Size (bytes): 1 Bytes to write: 0 Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Start time: 2019-08-28 15:23:44,972 Maps to run: 20 Reduces to run: 10 Block Size (bytes): 1 Bytes to write: 0 Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Maps to run: 20 Reduces to run: 10 Block Size (bytes): 1 Bytes to write: 0 Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Reduces to run: 10 Block Size (bytes): 1 Bytes to write: 0 Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Block Size (bytes): 1 Bytes to write: 0 Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Bytes to write: 0 Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Bytes per checksum: 1 Number of files: 200 Replication factor: 1
Number of files: 200 Replication factor: 1
Replication factor: 1
Successful file operations: 4000
# maps that missed the barrier: 0
# exceptions: 0
TPS: Create/Write/Close: 1987
Avg exec time (ms). Create/Write/Close, 19 7415
Avg Lat (ms): Create/Write: 7,7955
Avg Lat (ms): Close: 11,8985
RAW DATA: AL Total #1: 31182
RAW DATA: AL Total #2: 47594
RAW DATA: TPS Total (ms): 78966
RAW DATA: Longest Man Time (ms): 4025 0
RAW DATA: Late maps: 0
RAW DATA: # of exceptions: 0

# open\_read性能测试

1. 使用20个map、10个reduce的MapReduce作业打开1000个文件。

./hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.6-tests.ja
r nnbench -maps 20 -reduces 10 -numberOfFiles 1000 -operation open\_read

2. 查看生成的统计信息。

统计信息在当前测试目录下的NNBench\_results.log文件中。

TPS表示每秒处理的事务个数,测试结果为7174。

NNBench :	
Version:	NameNode Benchmark 0.4
Date & time:	2019-08-28 15:38:08,168
Test Operation:	open read
Start time:	2019-08-28 15:37:58,173
Maps to run:	20
Reduces to run:	10
Block Size (bytes):	1
Bytes to write:	Θ
Bytes per checksum:	1
Number of files:	1000
Replication factor:	1
Successful file operations:	20000
<pre># maps that missed the barrier:</pre>	0
# exceptions:	0
TPS: Open/Read:	7147
Avg Exec time (ms). Open/Read.	2.2103
Avg Lat (ms): Open:	1.3914
null	
RAW DATA: AL Total #1:	27828
RAW DATA: AL Total #1: RAW DATA: AL <u>Total #2:</u>	27828 0
RAW DATA: AL Total #1: RAW DATA: AL Total #2: RAW DATA: TPS T <u>otal (ms):</u>	27828 0 44206
RAW DATA: AL Total #1: RAW DATA: AL Total #2: RAW DATA: TPS Total (ms): RAW DATA: Longest Map Time (ms):	27828 0 44206 2798.0
RAW DATA: AL Total #1: RAW DATA: AL Total #2: RAW DATA: TPS Total (ms): RAW DATA: Longest Map Time (ms): RAW DATA: Late maps:	27828 0 44206 2798.0 0

# rename性能测试

1. 使用20个map, 10个reduce的MapReduce作业来重命名1000个文件。

./hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.6-tests.ja
r nnbench -maps 20 -reduces 10 -numberOfFiles 1000 -operation rename

2. 查看生成的统计信息。

统计信息在当前测试目录下的NNBench\_results.log文件中。

TPS表示每秒处理的事务个数,测试结果为3469。

NNDonch	
NNBench :	NameNede Renchmark 0 4
version. Data & time:	Namenoue Benchmark 0.4
	2019-08-28 13.40.34,830
Test Operation.	500.000
Test operation.	2010 00 20 15,40,40 770
Mana ta runi	2019-08-28 15.40.40,770
Maps to run:	20
Reduces to run:	10
BLOCK Size (bytes):	1
Bytes to write:	1
Bytes per checksum:	1
Number of files:	1000
Replication factor:	1
Successful file operations:	20000
# word that wiseed the boundary	0
# maps that missed the barrier:	0
# exceptions:	0
TDS: Ponomo:	3460
Avg Exec time	5 50/1
Avg Lite (ms): Rename:	5.5541
Avg Lat (ms): Kename:	3.30373
RAW DATA: AL Total #1:	111275
RAW DATA: AL Total #2:	0
RAW DATA: TPS Total (ms):	111882
RAW DATA: Longest Map Time (ms):	5765 0
RAW DATA: Late maps:	0
RAW DATA: # of exceptions:	0
NAW DATA. # OT exceptions.	0

# delete性能测试

1. 使用20个map, 10个reduce的MapReduce作业来删除1000个文件。

./hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.6-tests.ja
r nnbench -maps 20 -reduces 10 -numberOfFiles 1000 -operation delete

2. 查看生成的统计信息。

统计信息在当前测试目录下的NNBench\_results.log文件中。

TPS表示每秒处理的事务个数,测试结果为10887。

······ NNBench ······ Version: Version: Date & time:	NameNode Benchmark 0.4 2019-08-28 15:43:26,45
Test Operation: Start time: Maps to run: Reduces to run: Block Size (bytes): Bytes to write: Bytes per checksum: Number of files: Beplication factor:	delete 2019-08-28 15:43:17,123 20 10 1 0 1 1000 1
Successful file operations:	20000
<pre># maps that missed the barrier:</pre>	0 0
TPS: Delete: Avg Exec time (ms): Delete: Avg Lat (ms): Delete: null	10887 1.5899 1.56395
RAW DATA: AL Total #1: RAW DATA: AL Total #2: RAW DATA: TPS Total (ms): RAW DATA: Longest Map Time (ms): RAW DATA: Late maps: RAW DATA: # of exceptions:	31279 0 31798 1837.0 0

# 7.在文件存储HDFS版上使用Apache Spark

本文主要介绍如何在挂载文件存储HDFS版的Hadoop集群上安装及使用Apache Spark。

# 前提条件

- 已开通**文件存储HDFS版**服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入 门。
- 已为Hadoop集群所有节点安装JDK,且JDK版本不低于1.8。
- 下载Apache Hadoop压缩包。建议您选用的Hadoop版本不低于2.7.2,本文中使用的Hadoop版本为 Apache Hadoop 2.7.2。
- 下载Apache Spark压缩包。本文中使用的Apache Spark版本为2.4.8。

# 步骤一: 配置Hadoop

1. 执行以下命令解压Hadoop压缩包到指定目录。

tar -zxf hadoop-2.7.2.tar.gz -C /usr/local/

- 2. 修改hadoop-env.sh配置文件。
  - i. 执行以下命令打开hadoop-env.sh配置文件。

vim /usr/local/hadoop-2.7.2/etc/hadoop/hadoop-env.sh

ii. 配置JAVA\_HOME目录,如下所示。

export JAVA\_HOME=/usr/java/default

#### 3. 修改core-site.xml配置文件。

i. 执行以下命令打开core-site.xml配置文件。

vim /usr/local/hadoop-2.7.2/etc/hadoop/core-site.xml

ii. 在core-site.xml配置文件中,配置如下信息。更多信息,请参见挂载管理。

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>dfs://x-xxxxxxx.cn-xxxxx.dfs.aliyuncs.com:10290</value>
<!-- 该地址填写您的挂载点地址 -->
</property>
<property>
<name>fs.dfs.impl</name>
<value>com.alibaba.dfs.DistributedFileSystem</value>
</property>
<property>
<name>fs.AbstractFileSystem.dfs.impl</name>
<value>com.alibaba.dfs.DFS</value>
</property>
</property>
```

#### 4. 修改yarn-site.xml配置文件。

i. 执行如下命令打开yarn-site.xml配置文件。

vim /usr/local/hadoop-2.7.2/etc/hadoop/yarn-site.xml

#### ii. 在yarn-site.xml配置文件中, 配置如下信息。

```
<configuration>
   <property>
       <name>yarn.resourcemanager.hostname</name>
       <value>xxxx</value>
       <!-- 该地址填写集群中ResourceManager的Hostname -->
   </property>
    <property>
       <name>yarn.nodemanager.aux-services</name>
       <value>mapreduce shuffle</value>
   </property>
    <property>
       <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
       <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>
    <property>
       <name>yarn.nodemanager.resource.memory-mb</name>
       <value>16384</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
    </property>
    <property>
       <name>yarn.nodemanager.resource.cpu-vcores</name>
       <value>4</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
   </property>
    <property>
       <name>yarn.scheduler.maximum-allocation-vcores</name>
       <value>4</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
    </property>
    <property>
       <name>yarn.scheduler.minimum-allocation-mb</name>
       <value>3584</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
   </property>
    <property>
       <name>yarn.scheduler.maximum-allocation-mb</name>
       <value>14336</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
    </property>
</configuration>
```

### 5. 修改slaves配置文件。

#### i. 执行如下命令打开slaves配置文件。

vim /usr/local/hadoop-2.7.2/etc/hadoop/slaves

ii. 在slaves配置文件中, 配置集群计算节点的Host name。

```
cluster-header-1
cluster-worker-1
```

#### 6. 配置环境变量。

i. 执行如下命令打开/etc/profile配置文件。

vim /etc/profile

ii. 在/etc/profile配置文件中, 配置HADOOP HOME。

```
export HADOOP_HOME=/usr/local/hadoop-2.7.2
export HADOOP_CLASSPATH=$($HADOOP_HOME/bin/hadoop classpath)
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$HADOOP HOME/bin:$HADOOP HOME/sbin:$PATH
```

iii. 执行如下命令使配置生效。

source /etc/profile

7. 配置文件存储HDFS版的Java SDK。

您可以下载最新的文件存储HDFS版的Java SDK,将其部署在Hadoop生态系统组件的CLASSPATH上,具体操作,请参见挂载管理。

cp aliyun-sdk-dfs-x.y.z.jar /usr/local/hadoop-2.7.2/share/hadoop/hdfs

8. 执行如下命令将\${HADOOP\_HOME}文件夹同步到集群的其他节点的相同目录下,并按照步骤6对集群其 他节点配置Hadoop的环境变量。

scp -r hadoop-2.7.2/ root@cluster-worker-1:/usr/local/

# 步骤二:验证Hadoop配置

完成Hadoop配置后,不需要格式化NameNode,也不需要使用start-dfs.sh来启动HDFS相关服务。如需使用YARN服务,只需在ResourceManager节点启动YARN服务。具体验证Hadoop配置成功的方法,请参见验证安装。

#### 步骤三:配置Apache Spark

1. 执行如下命令解压Apache Spark压缩包到指定目录。

tar -zxf spark-2.4.8-bin-hadoop2.7.tgz -C /usr/local/

2. 将文件存储HDFS版Java SDK放到Spark的jars目录下。

cp aliyun-sdk-dfs-x.y.z.jar /usr/local/hadoop-2.7.2/share/hadoop/hdfs

☑ 注意 如果您还需要对Spark进行额外配置。具体操作,请参见Spark官方文档。

# 步骤四:验证Apache Spark配置

使用Spark读取文件存储HDFS版上面的文件进行WordCount计算,并将计算结果写入文件存储HDFS版。

1. 执行以下命令, 在**文件存储HDFS版**上生成测试数据。

hadoop jar \${HADOOP\_HOME}/share/hadoop/mapreduce/hadoop-mapreduce-examples-\*.jar \
randomtextwriter \
-D mapreduce.randomtextwriter.totalbytes=10240 \
-D mapreduce.randomtextwriter.bytespermap=1024 \

dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/input

其中, f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com 为文件存储HDFS版挂载点地址,请根据实际 情况替换。

#### 2. 启动spark-shell执行WordCount。

```
${SPARK_HOME}/bin/spark-shell --master yarn
scala> val res = sc.textFile("dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/inp
ut").flatMap(_.split(" ")).map((_,1)).reduceByKey(_+_)
scala> res.collect.foreach(println)
scala> res.saveAsTextFile("dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/output
")
```

#### 3. 执行以下命令, 查看写入文件存储HDFS版的结果数据。

hadoop fs -ls dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/output

#### 如果返回以下信息,则表示配置Apache Spark成功。

[root@cluste	r-h	neader	r-1 ~]#	hadoop fs	-ls dfs	//f-	and the first of the	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output
Found 11 items								
-rwxrwxrwx		root	root		2021-11-2	5 14:14	dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/_SUCCESS
-rwxrwxrwx		root	root	1215	2021-11-2	5 14:14	dfs://f-(	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00000
-rwxrwxrwx		root	root	1171	2021-11-2	5 14:14	dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00001
-rwxrwxrwx		root	root	1405	2021-11-2	5 14:14	dfs://f-(	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00002
-rwxrwxrwx		root	root	1532	2021-11-2	5 14:14	dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00003
-rwxrwxrwx		root	root	1008	2021-11-2	5 14:14	dfs://f-0	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00004
-rwxrwxrwx		root	root	1061	2021-11-2	5 14:14	dfs://f-0	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00005
-rwxrwxrwx		root	root	1381	2021-11-2	5 14:14	dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00006
-rwxrwxrwx		root	root	1497	2021-11-2	5 14:14	dfs://f-0	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00007
-rwxrwxrwx		root	root	1439	2021-11-2	5 14:14	dfs://f-(	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00008
-rwxrwxrwx		root	root	1294	2021-11-2	5 14:14	dfs://f-	.cn-zhangjiakou.dfs.aliyuncs.com:10290/output/part-00009
[root@cluste	ir-h	neader	r-1 ~]#					

# 8.在文件存储HDFS版上使用Apache Flink

本文介绍如何在挂载文件存储HDFS版的Hadoop集群上安装及使用Apache Flink。

# 前提条件

- 已开通**文件存储HDFS版**服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入 门。
- 已为Hadoop集群所有节点安装JDK,且JDK版本不低于1.8。
- 下载Apache Hadoop压缩包。建议您选用的Hadoop版本不低于2.7.2,本文中使用的Hadoop版本为 Apache Hadoop 2.7.2。
- 下载Apache Flink压缩包。本文使用的版本为官方提供的预编译版本Apache Flink 1.12.5。

# 步骤一: 配置Hadoop

1. 执行以下命令解压Hadoop压缩包到指定目录。

tar -zxf hadoop-2.7.2.tar.gz -C /usr/local/

- 2. 修改hadoop-env.sh配置文件。
  - i. 执行以下命令打开hadoop-env.sh配置文件。

vim /usr/local/hadoop-2.7.2/etc/hadoop/hadoop-env.sh

ii. 配置JAVA\_HOME目录,如下所示。

export JAVA\_HOME=/usr/java/default

#### 3. 修改core-site.xml配置文件。

i. 执行以下命令打开core-site.xml配置文件。

vim /usr/local/hadoop-2.7.2/etc/hadoop/core-site.xml

ii. 在core-site.xml配置文件中,配置如下信息。更多信息,请参见挂载管理。

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>dfs://x-xxxxxxx.cn-xxxxx.dfs.aliyuncs.com:10290</value>
<!-- 该地址填写您的挂载点地址 -->
</property>
<property>
<name>fs.dfs.impl</name>
<value>com.alibaba.dfs.DistributedFileSystem</value>
</property>
<property>
<name>fs.AbstractFileSystem.dfs.impl</name>
<value>com.alibaba.dfs.DFS</value>
</property>
</property>
```

#### 4. 修改yarn-site.xml配置文件。

i. 执行如下命令打开yarn-site.xml配置文件。

vim /usr/local/hadoop-2.7.2/etc/hadoop/yarn-site.xml

#### ii. 在yarn-site.xml配置文件中, 配置如下信息。

```
<configuration>
   <property>
       <name>yarn.resourcemanager.hostname</name>
       <value>xxxx</value>
       <!-- 该地址填写集群中ResourceManager的Hostname -->
   </property>
    <property>
       <name>yarn.nodemanager.aux-services</name>
       <value>mapreduce shuffle</value>
   </property>
    <property>
       <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
       <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>
    <property>
       <name>yarn.nodemanager.resource.memory-mb</name>
       <value>16384</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
    </property>
    <property>
       <name>yarn.nodemanager.resource.cpu-vcores</name>
       <value>4</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
   </property>
    <property>
       <name>yarn.scheduler.maximum-allocation-vcores</name>
       <value>4</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
    </property>
    <property>
       <name>yarn.scheduler.minimum-allocation-mb</name>
       <value>3584</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
   </property>
    <property>
       <name>yarn.scheduler.maximum-allocation-mb</name>
       <value>14336</value>
       <!-- 根据您当前的集群能力进行配置此项 -->
    </property>
</configuration>
```

### 5. 修改slaves配置文件。

#### i. 执行如下命令打开slaves配置文件。

vim /usr/local/hadoop-2.7.2/etc/hadoop/slaves

ii. 在slaves配置文件中, 配置集群计算节点的Host name。

```
cluster-header-1
cluster-worker-1
```

#### 6. 配置环境变量。

i. 执行如下命令打开/etc/profile配置文件。

vim /etc/profile

ii. 在/etc/profile配置文件中, 配置HADOOP HOME。

```
export HADOOP_HOME=/usr/local/hadoop-2.7.2
export HADOOP_CLASSPATH=$($HADOOP_HOME/bin/hadoop classpath)
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$HADOOP HOME/bin:$HADOOP HOME/sbin:$PATH
```

iii. 执行如下命令使配置生效。

source /etc/profile

7. 配置文件存储HDFS版的Java SDK。

您可以下载最新的文件存储HDFS版的Java SDK,将其部署在Hadoop生态系统组件的CLASSPATH上,具体操作,请参见挂载管理。

cp aliyun-sdk-dfs-x.y.z.jar /usr/local/hadoop-2.7.2/share/hadoop/hdfs

8. 执行如下命令将\${HADOOP\_HOME}文件夹同步到集群的其他节点的相同目录下,并按照步骤6对集群其 他节点配置Hadoop的环境变量。

scp -r hadoop-2.7.2/ root@cluster-worker-1:/usr/local/

# 步骤二:验证Hadoop配置

完成Hadoop配置后,不需要格式化NameNode,也不需要使用start-dfs.sh来启动HDFS相关服务。如需使用YARN服务,只需在ResourceManager节点启动YARN服务。具体验证Hadoop配置成功的方法,请参见验证安装。

# 步骤三: 配置Flink

执行以下命令, 解压Flink压缩包到指定目录。

tar -zxf flink-1.12.5-bin-scala\_2.11.tgz -C /usr/local/

↓ 注意

- 在使用Flink前,必须在您的集群环境变量中配置HADOOP\_HOME,HADOOP\_CLASSPATH和 HADOOP\_CONF\_DIR。具体操作,请参见配置环境变量。
- 如果您需要对Flink进行额外的配置,请参见配置操作指南。

# 步骤四:验证Flink配置
使用Flink自带的WordCount.jar对**文件存储HDFS版**上的数据进行读取,并将计算结果写入到**文件存储HDFS**版。

## 1. 在**文件存储HDFS版**上生成测试数据。

```
${HADOOP_HOME}/bin/hadoop jar ${HADOOP_HOME}/share/hadoop/mapreduce/hadoop-mapreduce-e
xamples-2.7.2.jar \
randomtextwriter \
-D mapreduce.randomtextwriter.totalbytes=10240 \
-D mapreduce.randomtextwriter.bytespermap=1024 \
dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/input
```

其中, f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com 为您的文件存储HDFS版的挂载点,请根据实际情况替换。

## 2. 执行以下命令,检查环境变量中是否包含HADOOP\_CLASSPATH。

echo \$HADOOP CLASSPATH

如果环境变量中不包含HADOOP\_CLASSPATH,执行以下命令,配置HADOOP\_CLASSPATH。

export HADOOP\_CLASSPATH=\$HADOOP\_CLASSPATH:\$(\$HADOOP\_HOME/bin/hadoop classpath)

#### 3. 执行以下命, 启动Flink Session on YARN。

./flink-1.12.5/bin/yarn-session.sh --detached

### 4. 执行WordCount.jar。

```
./flink-1.12.5/bin/flink run \
```

./flink-1.12.5/examples/batch/WordCount.jar \

--input dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/flink-test/input \

--output dfs://f-xxxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/flink-test/output

其中, f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com 为您的文件存储HDFS版的挂载点地址,请根 据实际情况替换。

## 5. 查看输出在文件存储HDFS版实例上的部分结果。

```
${HADOOP_HOME}/bin/hadoop fs -cat dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290
/flink-test/output | tail -20
```

#### 如果返回以下类似信息,则表示Flink配置成功。



# 9.使用Fuse-DFS挂载文件存储HDFS版

本文主要介绍如何使用Fuse-DFS工具实现文件存储HDFS版在本地文件系统的映射。

## 前提条件

- 已开通文件存储HDFS版服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入门。
- 已为Hadoop集群所有节点安装JDK,且JDK版本不低于1.8。建议您使用的Hadoop版本不低于2.7.2,本文 使用的Hadoop版本为Apache Hadoop 2.8.5。
- 已下载与Hadoop集群版本相同的Hadoop源码包。具体下载地址请参见下载Hadoop源码包。

## 背景信息

Fuse-DFS是Hadoop项目自带的一个功能模块,提供了使用FUSE(用户空间中的文件系统)在UNIX文件系统 上映射HDFS的功能,在官方提供的Hadoop预编译版本中不包含Fuse-DFS的功能模块,如需使用该功能需要 手动编译该功能模块并添加到Hadoop客户端中。Fuse-DFS官方文档请参见Mount ableHDFS。

⑦ 说明 Fuse-DFS工具在阿里云文件存储HDFS版上使用时需要额外的配置。具体操作,请参见步骤
 二:配置Fuse-DFS。

## 步骤一: Hadoop集群挂载文件存储HDFS版实例

在Hadoop集群中配置文件存储HDFS版实例。具体操作,请参见挂载管理。

## 步骤二:配置Fuse-DFS

- 1. 在Hadoop客户端上安装依赖并载入FUSE模块。
  - i. 执行以下命令, 安装依赖。

yum -y install fuse fuse-devel fuse-libs

ii. 执行以下命令,载入FUSE模块。

modprobe fuse

2. 解压Hadoop源码包。

tar -zxvf hadoop-2.8.5-src.tar.gz

3. 修改代码并编译。

使用Fuse-DFS挂载**文件存储HDFS版**到本地文件系统时,系统会将挂载点地址的前缀\_dfs:// 修改为 hdfs:// ,导致挂载失败,因此需要修改代码重新编译,操作如下。 i. 执行以下命令, 打开*fuse\_options.c*文件, 并将 #define NEW\_HDFS\_URI\_LOCATION "hdfs://" 修 改为 #define NEW HDFS URI LOCATION "dfs://" 。

vim hadoop-2.8.5-src/hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/ fuse-dfs/fuse options.c



ii. 执行以下命令编译Hadoop源码中hadoop-hdfs-project模块下的hadoop-hdfs-native-client子模 块。

```
cd hadoop-2.8.5-src/
mvn clean package -pl hadoop-hdfs-project/hadoop-hdfs-native-client -Pnative -Dskip
Tests
```

○ 注意 不同的Hadoop版本中子模块的hadoop-hdfs-native-client命名及位置可能不同。 在Hadoop 2.7.x版本中该模块包含在hadoop-hdfs-project/hadoop-hdfs子模块中, Hadoop 2.8.x及之后的版本都在模块hadoop-hdfs-project下。

## 4. 配置Fuse-DFS。

## 将编译出来的Fuse-DFS包拷贝到Hadoop客户端的bin目录下。示例命令如下:

```
cp hadoop-2.8.5-src/hadoop-hdfs-project/hadoop-hdfs-native-client/target/main/native/fu se-dfs/fuse dfs ${HADOOP HOME}/bin
```

## 5. 为Hadoop客户端配置环境变量。

i.执行 vim /etc/profile 命令,打开配置文件,添加如下内容。

```
export OS_ARCH=amd64
export LD_LIBRARY_PATH=${JAVA_HOME}/jre/lib/${OS_ARCH}/server:${HADOOP_HOME}/lib/na
tive
export CLASSPATH=$CLASSPATH:`${HADOOP_HOME}/bin/hadoop_classpath --glob`
```

ii. 执行以下命令使配置生效。

source /etc/profile

## 步骤三:使用Fuse-DFS

1. 创建目录。

mkdir /mnt/dfs\_mount

## 2. 挂载文件存储HDFS版至本地文件系统。

fuse\_dfs dfs://f-xxxxx.cn-xxx.dfs.aliyuncs.com:10290/ /mnt/dfs\_mount

其中, dfs://f-xxxxx.cn-xxx.dfs.aliyuncs.com 为您的文件存储HDFS版的挂载点地址,请根据实际情况替换。

3. 验证挂载结果。

如果您在本地目录中可以查看文件存储HDFS版中的文件和目录,则表示挂载成功。

[root@cluster-header-1 ~]# ls -l /mnt/	
cotal 4	
drwxrwxrwx 13 root root 4096 Dec 1 10:56 <mark>dfs_mount</mark>	
[root@cluster-header-1 ~]# ls -l /mnt/dfs_mount/	
total 44	
drwxrwxrwx 5 root root 4096 Nov 5 15:55 apps	
drwxrwxrwx 3 root root 4096 Nov 10 15:03 <mark>benchmarks</mark>	
drwxrwxrwx 3 root root 4096 Nov 10 17:26 <mark>dfs2oss</mark>	
drwxrwxrwx 4 root root 4096 Nov 11 16:53 <mark>flink-test</mark>	
drwxrwxrwx 14 root root 4096 Dec 1 11:22 <mark>hbase</mark>	
drwxrwxrwx 13 root root 4096 Nov 8 16:26 <mark>input</mark>	
drwxrwxrwx 207 root root 4096 Nov 11 10:37 <mark>oss2dfs</mark>	
drwxrwxrwx 13 root root 4096 Nov 25 14:14 output	
drwxrwxrwx 2 root root 4096 Nov 19 15:05 <mark>cestsnapshot</mark>	
drwxrwxrwx 11 root root 4096 Dec 2 11:17 tmp_	
drwxrwxrwx 5 root root 4096 Nov 15 11:07 <mark>Jeen</mark>	
root@cluster-header-1 ~]# hadoop fs -ls dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/	
SLF4J: Class path contains multiple SLF4J bindings.	
5LF4J: Found binding in [jar:file:/opt/module/hadoop-2.8.5/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/S	a
5LF4J: Found binding in [jar:file:/opt/module/tez-0.9.2/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class	l l
5LF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.	
5LF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]	
Found 11 items	
drwxrwxrwx - root root 0 2021-11-05 15:55 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/apps	
drwxrwxrwx - root root 0 2021-11-10 15:03 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/benchmarks	
drwxrwxrwx - root root 0 2021-11-10 17:26 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/dfs2oss	
drwxrwxrwx - root root 0 2021-11-11 16:53 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/flink-test	
drwxrwxrwx - root root 0 2021-12-01 11:22 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/hbase	
drwxrwxrwx - root root 0 2021-11-08 16:26 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/input	
drwxrwxrwx - root root 0 2021-11-11 10:37 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/oss2dfs	
drwxrwxrwx - root root 0 2021-11-25 14:14 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/output	
drwxrwxrwx - root root 0 2021-11-19 15:05 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/testsnapshot	
drwxrwxrwx - root root 0 2021-12-02 11:17 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/tmp	
drwxrwxrwx - root root 0 2021-11-15 11:07 dfs://fcn-zhangjiakou.dfs.aliyuncs.com:10290/user	

4. 挂载成功后,您可以在本地访问文件存储HDFS版,执行读取或写入操作。

执行以下命令在已映射**文件存储HDFS版**的本地目录中创建文件,创建成功后,您可以在**文件存储** HDFS版上查看到该文件。

cd /mnt/dfs mount

mkdir fuse test

echo "hello dfs" > fuse test/fuse.txt

[root@cluster-header-1 dfs\_mount]# hadoop fs -cat dfs://f-( .cn-zhangjiakou.dfs.aliyuncs.com:10290/fuse\_test/fuse.txt 2>/dev/nul hello dfs [root6c.uster border 1 dfc\_mount]#

5. 卸载挂载目录。

fusermount -u /mnt/dfs\_mount

↓ 注意 客户端重启之后,您需要重新挂载文件存储HDFS版到本地,建议您配置开机自动挂载。

# 10.在文件存储HDFS版上使用Presto

本文主要介绍如何在文件存储HDFS版上搭建及使用Presto。

## 前提条件

- 已开通文件存储HDFS版服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入门。
- •
- 已在集群中安装Hive。本文使用的Hive版本为2.3.7。
- 已下载Presto安装包和presto-cli-xxx-executable.jar。本文使用的Presto版本为0.265.1。

## 背景信息

Presto是一个开源的分布式SQL查询引擎,适用于交互式分析查询。

② 说明 在本文中Presto是通过连接Hive的元数据服务来读取文件存储HDFS版上的数据,在文件存储HDFS版上使用Presto时需要额外配置一些依赖包。具体操作,请参见步骤二:配置Presto。

## 步骤一: Hadoop集群挂载文件存储HDFS版实例

在Hadoop集群中配置文件存储HDFS版实例。具体操作,请参见挂载管理。

## 步骤二: 配置Presto

1. 执行以下命令, 解压Presto安装包。

tar -zxf presto-server-0.265.1.tar.gz

2. 执行以下命令, 在Presto解压目录下创建目录(例如/etc)。

mkdir presto-server-0.265.1/etc

## 3. 配置节点环境。

i. 创建etc/node.properties文件。

vim presto-server-0.265.1/etc/node.properties

ii. 在etc/node.properties文件中添加如下内容。

## 4. 配置JVM参数。

## i. 创建etc/jvm.config文件。

vim presto-server-0.265.1/etc/jvm.config

## ii. 在etc/jvm.config文件中添加如下内容。

-server
-Xmx8G
-XX:+UseG1GC
-XX:G1HeapRegionSize=32M
-XX:+UseGCOverheadLimit
-XX:+ExplicitGCInvokesConcurrent
-XX:+HeapDumpOnOutOfMemoryError

-XX:+ExitOnOutOfMemoryError

### 5. 配置Presto Server。

## 本文以将Coordinator和Worker配置在同一台机器上为例进配置。您也可以将Coordinator和Worker配置 在不同的机器中,具体操作,请参见Presto官方文档。

#### i. 创建etc/config.properties文件。

vim presto-server-0.265.1/etc/config.properties

#### ii. 在etc/config.properties中添加如下内容。

```
coordinator=true
node-scheduler.include-coordinator=true
http-server.http.port=8080
query.max-memory=5GB
query.max-memory-per-node=1GB
query.max-total-memory-per-node=2GB
discovery-server.enabled=true
discovery.uri=http://xx.xx.xx:8080 #xx.xx.xx.xx为当前机器的ip地址
```

### 6. 配置日志级别。

#### i. 创建etc/log.properties文件。

vim presto-server-0.265.1/etc/log.properties

#### ii. 在etc/log.properties文件中添加如下内容。

com.facebook.presto=INFO

#### 7. 配置Presto数据源。

## i. 创建etc/catalog目录。

mkdir presto-server-0.265.1/etc/catalog

### ii. 创建etc/catalog/hive.properties文件。

vim presto-server-0.265.1/etc/catalog/hive.properties

## iii. 在etc/catalog/hive.properties中添加如下内容。

```
connector.name=hive-hadoop2
hive.metastore.uri=thrift://xxxx:9083 #xxxx为启动 hive 元数据服务的ip地址
hive.config.resources=/path/to/core-site.xml #请替换为该节点上已挂载文件存储HDFS版的Had
oop core-site.xml路径
```

### 8. 编译并替换JAR包。

Presto使用maven-shade-plugin插件打包,对引入的Hadoop依赖进行了重命名,文件存储HDFS 版Java SDK和Hadoop共用了protobuf-xxx.jar包,Presto通过Hive Metastore读取文件存储HDFS版上的数据时,文件存储HDFS版获取不到Presto重命名地址后的protobuf-xxx.jar包而报错。为了避免兼容性问题,文件存储HDFS版的Java SDK需要作为Presto Hadoop的依赖项引入,并对Presto中引入的Hadoop Jar包hadoop-apache2-xxx.jar重新编译。



## i. 查看安装的Presto中的presto-hadoop-apache2版本。

以0.265.1版本为例,通过源码POM文件可以看到引入的hadoop-apache2版本为2.7.4-9。

611	<dependency></dependency>
612	<pre><groupid>com.fasterxml.jackson.module</groupid></pre>
613	<artifactid>jackson-module-afterburner</artifactid>
614	<version>\${dep.jackson.version}</version>
615	
616	
617	<dependency></dependency>
618	<pre><groupid>com.facebook.presto.hadoop</groupid></pre>
619	<artifactid>hadoop-apache2</artifactid>
620	<version>2.7.4-9</version>
621	
622	
623	<dependency></dependency>

#### ii. 下载presto-hadoop-apache2对应版本的源码。

git clone -b 2.7.4-9 https://github.com/prestodb/presto-hadoop-apache2.git

 iii. 在源码中的POM文件中添加文件存储HDFS版最新Java SDK的依赖项。本文使用的Java SDK版本为 1.0.5。



iv. 编译presto-hadoop-apache2。

cd presto-hadoop-apache2

mvn clean package -DskipTests

- v. 查看生成的hadoop-apache2-2.7.4-9.jar。
  - a. 执行以下命令, 进入上一级目录。

cd ..

b. 执行以下命令, 查看hadoop-apache2-2.7.4-9.jar是否符合预期。

ll -h presto-hadoop-apache2/target/

-rw-rr	1	root	root	1.7K	Nov	16	15:42						
- pw-pp	1	root	root	11K	Nov	16	15.42						
-pw-pp	1	root	root	1 84	Nov	16	15.42						
	5	noot	root	1 01	Nov	16	15.42						
	ח	poot	noot	4.00	Nov	16	15.42						
dnuwn-xn-x	n N	noot	noot	4.00	Nov	16	15:42						
	1		t	4.0K	Mov	10	15:42	-		L-2 2	7 4 0		
- rw-rr	4	root	root	1 74	NOV	10	15:42	nadoor	o-apac	nez-z.	7.4-9	.jar	
-rw-rr	T	root	root	1/14	NOV	16	15:43						
-rw-rr	1	root	root	9.4K	Nov	16	15:42						

## vi. 替换旧JAR包。

## a. 移除旧JAR包。

mv presto-server-0.265.1/plugin/hive-hadoop2/hadoop-apache2-2.7.4-9.jar prestoserver-0.265.1/plugin/hive-hadoop2/hadoop-apache2-2.7.4-9.jar.bak

b. 将编译后的hadoop-apache2-2.7.4-9.jar依赖包拷贝到对应目录下。

cp presto-hadoop-apache2/target/hadoop-apache2-2.7.4-9.jar presto-server-0.265
.1/plugin/hive-hadoop2/

## 9. 配置presto-cli-xxx-executable.jar。

i. 将下载的presto-cli-xxx-executable.jar复制到presto-server-0.265.1/bin/目录。

cp presto-cli-0.265.1-executable.jar presto-server-0.265.1/bin/

ii. 重命名 prest o-server-0.265.1/bin/目录中的 prest o-cli-xxx-execut able.jar。

mv presto-server-0.265.1/bin/presto-cli-0.265.1-executable.jar presto-server-0.265
.1/bin/presto

## iii. 为重新命名的文件添加可执行权限。

chmod +x presto-server-0.265.1/bin/presto

# 步骤三: 验证Presto

## 1. 执行以下命令,启动Hive的元数据服务。

hive --service metastore

#### 2. 启动Presto Server并连接Hive Metastore。

i. 启动Presto Server。

presto-server-0.265.1/bin/launcher start

ii. 连接Hive Metastore。

presto-server-0.265.1/bin/presto --server localhost:8080 --catalog hive

### 3. 通过Presto在Hive中创建数据库。

i. 在文件存储HDFS版实例上创建测试目录。

```
hadoop fs -mkdir dfs://f-xxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/presto_te
st
```

#### ii. 在Hive中创建数据库。

```
Query 20211117_093732_00032_sm7iu, FINISHED, 1 node
Splits: 19 total, 19 done (100.00%)
187ms [3 rows, 51B] [16 rows/s, 272B/s]
```

oresto> use prosto\_test;

- 4. 在刚创建的数据库中新建表并添加数据。
  - i. 进入刚创建的数据库。

use prosto\_test;

## ii. 创建表。

```
CREATE TABLE user_info_test (
   user_id bigint,
   firstname varchar,
   lastname varchar,
   country varchar
)
WITH (
   format = 'TEXTFILE'
);
```

### iii. 在刚创建表中插入数据。

INSERT INTO user\_info\_test VALUES(1,'Dennis','Hu','CN'),(2,'Json','Lv','Jpn'),(3,'M
ike','Lu','USA');

## 5. 查看文件存储HDFS版实例上是否有刚创建的表数据。

hadoop fs -ls dfs://f-xxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/tmp/presto\_test/\*

如果返回信息显示刚创建的表数据,则表示Presto可以向文件存储HDFS版写入数据。

## 6. 进行Word Count计算,检验Presto能否读取文件存储HDFS版上的数据并计算。

SELECT country,count(\*) FROM user info test GROUP BY country;

如果返回信息与创建表中的信息一致,则表示Presto可读取文件存储HDFS版上的数据并计算。

# 11.使用云企业网跨VPC访问文件存储 HDFS版

本文档主要介绍如何使用云企业网(CEN)跨VPC访问文件存储HDFS版。

# 背景信息

阿里云提供云企业网服务实现跨VPC访问**文件存储HDFS版**,主要场景如下所示。

场景	描述
	访问 <b>文件存储HDFS版</b> 的计算实例(例如ECS)与 <b>文件存储HDFS版</b> 属于相同 的账号且在同一地域。
同账号同地域跨VPC访问 <b>文件存储</b> HDFS版	⑦ 说明 此场景也可以使用文件存储HDFS版的多挂载功能实现单文件系统实例的多VPC访问。
同账号跨地域跨VPC访问 <b>文件存储</b> HDFS版	访问 <b>文件存储HDFS版</b> 的计算实例(例如ECS)与 <b>文件存储HDFS版</b> 属于相同 的账号但不在同一区域下。

本文以北京地域的ECS访问杭州地域的文件存储HDFS版为例。

? 说明

- 使用云企业网实现跨地域跨VPC访问文件存储HDFS版时,计算实例(例如ECS)与文件存储 HDFS版挂载点所在VPC的交换机网段必须不相同。
- 文件存储HDFS版的权限组规则中的授权地址必须包含计算实例(例如ECS)所在的VPC地址。

# 准备工作

- 1. 开通**文件存储HDFS版**服务并创建文件系统实例和挂载点,具体操作,请参见HDFS快速入门。
- 2. 在需要访问文件存储HDFS版的ECS上配置Hadoop环境。具体操作,请参见配置Hadoop和部署依赖。

## 创建云企业网实例

创建云企业网实例,并将北京地域ECS所在的VPC加载到云企业网实例中。

- 1. 登录云企业网管理控制台。
- 2. 在云企业网实例页面, 单击创建云企业网实例。
- 3. 在创建云企业网实例对话框中, 配置相关参数。

创建云企业网实例	?	×
本企业网		
• 名称 💿		
access_to_different_regions_test 32/128		
描述 ②		
北京区域ECS访问杭州区域文件存储HDFS		
21/256		
加载网络实例		
同账号		
(1) 注: 已加载到云企业网的实例不允许重复加载		
• 实例类型 💿		
专有网络(VPC) ~		
• 地域 💿		
华北2 (北京) ~		
• 网络实例 📀		
cn-beijing ~		
确定 取消		

重要参数说明如下所示。

配置	说明
实例类型	选择要互通的实例。本文选择 <b>专有网络(VPC)</b> 。
地域	选择所选实例的地域。本文选择 <b>华北2(北京)</b> 。
网络实例	选择要加载的实例。此处选择ECS所在的VPC实例。

4. 单击**确定**。

加载网络实例

将杭州地域的文件存储HDFS版挂载点所在的VPC实例加载到云企业网实例。

- 1. 在云企业网实例页面,找到目标云企业网实例,单击管理。
- 2. 在网络实例管理页签下,单击加载网络实例。
- 3. 在加载网络实例对话框中, 配置如下参数。

加载网络实例	?	$\times$
同账号 跨账号		
(1) 注: 已加载到云企业网的实例不允许重复加载		
• 实例类型 😨		
专有网络(VPC) ~		
• 地域 🕐		
华东1 (杭州) ~		
• 网络实例 📀		
dfs-test/vr		

重要参数说明如下所示。

配置	说明
账号	选择 <b>同账号</b> 。
实例类型	选择要互通的实例。本文选择 <b>专有网络(VPC)</b> 。
地域	选择所选实例的地域。本文选择 <b>华东1(杭州)</b> 。
实例	选择要加载的实例。本文选择 <b>文件存储HDFS版</b> 挂载点所在的VPC实例。

4. 单击确定。

## 设置跨地域互通带宽

为云企业网实例设置跨地域互通带宽。

② **说明** 如果您是使用云企业网实现同账号同地域跨VPC访问**文件存储HDFS版**,则无需设置跨地域 互通带宽。

1. 在云企业网实例页面,找到目标云企业网实例,单击管理。

- 2. 购买带宽包。
  - i. 在带宽包管理页签下,单击购买带宽包(预付费)。
  - ii. 根据需求购买带宽包。具体操作,请参见步骤四:设置跨地域互通带宽。

- 3. 设置跨地域互通带宽。
  - i. 在跨地域互通带宽管理页签下, 单击设置跨地域带宽。
  - ii. 根据页面提示配置相关参数,具体操作,请参见步骤四:设置跨地域互通带宽。

## 验证结果

- 1. 登录北京地域的ECS。
- 2. 验证ECS和文件存储HDFS版之间的连通性。

```
ping f-xxxxx.cn-xxxx.dfs.aliyuncs.com
```

f-xxxxx.cn-xxxx.dfs.aliyuncs.com 为文件存储HDFS版的挂载点域名。

3. 验证ECS是否能访问文件存储HDFS版挂载点的10290端口。

telnet f-xxxxx.cn-xxxx.dfs.aliyuncs.com 10290

4. 验证ECS是否能通过文件存储HDFS版挂载点访问文件存储HDFS版上的文件。

⑦ 说明 如果创建的云企业网实例未配置带宽包,默认带宽为1 kbps,仅可以测试连通性,不能 访问**文件存储HDFS版**实例上的文件或进行其他操作。

```
hadoop fs -ls dfs://f-xxxxx.cn-xxx.dfs.aliyuncs.com:10290/
hadoop fs -mkdir dfs://f-xxxxx.cn-xxxx.dfs.aliyuncs.com:10290/test_dir
hadoop fs -touchz dfs://f-xxxxx.cn-xxxx.dfs.aliyuncs.com:10290/test_dir/words
echo "hello world" | hadoop fs -appendToFile - dfs://f-xxxxx.cn-xxxx.dfs.aliyuncs.com/
test_dir/words
hadoop fs -cat dfs://f-xxxxx.cn-xxxx.dfs.aliyuncs.com/test_dir/words
```

# 12.在文件存储HDFS版上使用 TensorFlow

本文介绍如何在文件存储HDFS版上使用TensorFlow。

# 背景信息

TensorFlow是一个采用数据流图(data flow graphs),用于数值计算的开源软件库。

⑦ 说明 由于TensorFlow对文件系统的schema有特别的指定,因此在文件存储HDFS版上使用 TensorFlow时需要修改TensorFlow源码。具体操作,请参见配置TensorFlow支持文件存储HDFS版。

# 准备工作

- 1. 开通**文件存储HDFS版**服务并创建文件系统实例和挂载点,具体操作,请请参见HDFS快速入门。
- 2. 在Hadoop集群所有节点上安装JDK,版本不能低于1.8。
- 3. 在Hadoop集群中配置文件存储HDFS版实例,具体操作,请参见挂载管理。
- 根据您目前使用的TensorFlow版本下载源码,具体下载地址请参见官方链接。
   本文中使用的TensorFlow版本为1.15.0。

# 配置TensorFlow支持文件存储HDFS版

- 1. 在TensorFlow源码中添加对**文件存储HDFS版**的支持,具体操作,请参见tensorflow support aliyun HDFS。
- 2. 编译TensorFlow。

编译步骤1中已修改的TensorFlow源码,具体操作,请参见Build from source。

- 3. 验证配置结果。
  - i. 在文件存储HDFS版上创建测试文件。

```
hadoop fs -mkdir dfs://f-xxxx.cn-xxxx.dfs.aliyuncs.com:10290/test_tensorflow
hadoop fs -touchz dfs://f-xxxx.cn-xxxx.dfs.aliyuncs.com:10290/test_tensorflow/word
s
echo "hello tensorflow" | hadoop fs -appendToFile - dfs://f-xxxx.cn-xxxx.dfs.aliyun
cs.com:10290/test tensorflow/words
```

f-xxxx.cn-xxxx.dfs.aliyuncs.com 为文件存储HDFS版的挂载点域名,请根据实际情况替换。

ii. 使用TensorFlow的apigfile查看文件存储HDFS版上的数据。

如果显示HDFS上的数据,则表示配置成功。

```
(base) root@zb3 ~# python
Python 3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 19:07:31)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> tf.io.gfile.listdir('dfs://f-xxxx.cn-xxxx.dfs.aliyuncs.com:10290/test_tensorflo
w')
['words']
>>> with tf.gfile.Open('dfs://f-xxxx.cn-xxxx.dfs.aliyuncs.com:10290/test_tensorflow
/words') as rf:
... rf.read()
'hello tensorflow\n'
>>>
```

# 13.文件存储HDFS版和MaxCompute双 向数据迁移

本文介绍如何使用DataWorks实现MaxCompute与**文件存储HDFS版**之间的双向数据同步。您可以将 MaxCompute数据同步至**文件存储HDFS版**,也可以将**文件存储HDFS版**数据同步至MaxCompute。

☑ 注意 请确保Maxcompute、文件存储HDFS版、Dataworks在同一个地域(region)下。

## 准备工作

- 开通文件存储HDFS版服务并创建文件系统实例和挂载点,具体操作,请参见HDFS快速入门。
- 在Hadoop集群所有节点上安装JDK,版本不能低于1.8。
- 在Hadoop集群中配置文件存储HDFS版实例,具体操作,请参见挂载管理。
- 开通MaxCompute,具体操作,请参见配置MaxCompute。
- 开通DataWorks,具体操作,请参见配置DataWorks。

## 配置步骤

1. 配置DataWorks独享数据集成资源组。

开通配置及使用,请参见购买独享资源组。在购买时需要您注意以下几方面的配置:

- 配置的独享数据集成资源组,需要与文件存储HDFS版在同一region的同一可用区(AZ)下。
- 如果文件存储HDFS版相对应的AZ下的DataWorks无资源,或者文件存储HDFS版与独享数据集成资源组在相同region不同AZ下,请提交工单联系DataWorks工作人员获取支持。
- 在配置DataWorks专有网络及交换机地址时,要与文件存储HDFS版挂载点所绑定的专有网络和交换 机地址一致,具体操作,请参见操作独享资源组。

类型: 数据集成资源组	可用区: cn-shanghai-e	剩余可绑定的专有网络个数: 1	
* 专有网络: 🧕			创建专有网络
请选择			~
* 交换机: 🧕			创建交换机
请选择			~
请选择需要同步的数据源所领	- 邦定的交换机		
* 安全组: 🛛			创建安全组
请选择			~

2. 在独享数据集成资源组下, 放置**文件存储HDFS版**依赖。

在放置**文件存储HDFS版**依赖时,请提交工单联系DataWorks工作人员获取支持。

3. 配置数据同步任务。

配置数据任务的具体操作,请参见通过向导模式配置任务。详细参数设置如下:

- i. 配置数据走向。
  - 如果配置从MaxCompute数据同步到文件存储HDFS版时,配置的数据去向的数据源为HDFS。

J KALADIGA DI WALADI MANA			
	数据去向		收起
* 数据源	敗病源美型 へ 法择振调率 、 AnalyticDB for MySQL2.0 OTS	0	
	220		收起
	HDFS		
 才会显示字段映射	FTP HBase HBase11veal		
	LogHub		收起
别整个数据同步过程;	OpenSearch 数据同步文档		

■ 如果配置从文件存储HDFS版同步数据到MaxCompute时,配置的数据来源的数据源为HDFS。

				在这里配置数据的来	源端和写入端:	可以是默	认的数
01 选	择数据源	数	据来	源			
	* 数据源	HDFS A		dfs		?	
		OTS					
	此数据源不到	OTS Stream		同步任务,点击转			
		OSS					
		FTP					
(02) <del>-</del>	容段映射	HBase		源			
		Logi lub					
		✓ HDFS			请	先选择数	据源
		Kaika					
		AnalyticDB for MySQL2.0					

ii. 根据提示使用脚本模式配置同步任务。



iii. 在配置脚本中,添加文件存储HDFS版配置参数。

- 同步数据到文件存储HDFS版时,需要配置HDFS Writer。具体操作,请参见HDFS Writer。
- 从文件存储HDFS版同步数据时,需要配置HDFS Reader。具体操作,请参见HDFS Reader。

在配置HDFS Writer和HDFS Reader时,需在同步脚本中添加hadoopConfig参数并配置**文件存储** HDFS版的实现类。如下图所示。

? 说明

- defaultFS需配置为文件存储HDFS版的挂载点地址。
- hadoopConfig需添加文件存储HDFS版的实现类。

"hadoopConfig": {

```
"fs.dfs.impl": "com.alibaba.dfs.DistributedFileSystem",
    "fs.AbstractFileSystem.dfs.impl" : "com.alibaba.dfs.DFS"
}
```



4. 执行数据脚本。

执行数据脚本时,需要选用在步骤1中配置的独享数据集成资源组。保存数据同步脚本并运行。



# 验证MaxCompute数据同步至文件存储HDFS版

以下示例用来验证MaxCompute数据是否同步到了文件存储HDFS版上。

1. 在MaxCompute中创建测试表。

在MaxCompute中的创建表。具体操作,请参见在MaxCompute创建表。

```
CREATE TABLE IF NOT EXISTS maxcompute2df
(
id BIGINT COMMENT '编号',
name STRING COMMENT '姓名',
gender STRING COMMENT '性别',
age BIGINT COMMENT '性别',
birth STRING COMMENT '生日'
);
```

2. 在测试表中插入测试数据。

```
insert into maxcompute2dfs values(1,'测试用户1','男',20,'2000-1-1');
insert into maxcompute2dfs values(2,'测试用户2','男',20,'2000-1-1');
insert into maxcompute2dfs values(3,'测试用户3','女',20,'2000-1-1');
insert into maxcompute2dfs values(4,'测试用户4','女',20,'2000-1-1');
```

## 3. 在文件存储HDFS版上创建目录。

hadoop fs -mkdir dfs://f-xxxxx.cn-xxxx.dfs.aliyuncs.com:10290/maxcompute2dfs

## 4. 在DataWorks中编写数据同步脚本。

配置MaxCompute Reader和HDFS Writer脚本,具体操作,请参见MaxCompute Reader和HDFS Writer。

```
"column": [
                "*"
            ],
            "guid": null,
            "emptyAsNull": false,
            "table": "maxcompute2dfs"
        },
        "name": "Reader",
        "category": "reader"
    },
    {
        "stepType": "hdfs",
        "parameter": {
            "path": "/maxcompute2dfs",
            "fileName": "maxcompute2dfs",
            "datasource": "xxx",
            "column": [
                {
                    "name": "id",
                    "type": "long"
                },
                {
                    "name": "name",
                    "type": "string"
                },
                {
                    "name": "gender",
                    "type": "string"
                },
                {
                    "name": "age",
                    "type": "long"
                },
                {
                    "name": "birth",
                    "type": "string"
                }
            ],
            "defaultFS": "dfs://f-xxxxx.cn-xxx.dfs.aliyuncs.com:10290",
            "writeMode": "append",
            "encoding": "UTF-8",
            "fieldDelimiter": ",",
            "fileType": "text",
            "hadoopConfig": {
                "fs.dfs.impl": "com.alibaba.dfs.DistributedFileSystem",
                "fs.AbstractFileSystem.dfs.impl" : "com.alibaba.dfs.DFS"
            }
        },
        "name": "Writer",
        "category": "writer"
    }
],
"version": "2.0",
"order": {
  "hono". [
```

```
nops : [
           {
                "from": "Reader",
                "to": "Writer"
           }
       ]
   },
   "setting": {
       "errorLimit": {
           "record": ""
       },
       "speed": {
           "concurrent": 2,
           "throttle": false
       }
   }
}
```

5. 加载创建的独享数据集成资源组并执行数据脚本。



6. 查看MaxCompute是否成功同步数据到文件存储HDFS版。

```
hadoop fs -cat dfs://f-xxxx.cn-xxx.dfs.aliyuncs.com:10290/maxcompute2dfs/*

[root@izuf6hcv8qx066x6xgjd8gz ~]# hadoop fs -ls /maxcompute2dfs
Found 2 items

-rwxrwxrwx 3 root root 64 2020-01-02 18:02 /maxcompute2dfs/maxcompute2dfs_3cbcf0cf_0dc3_4439_9efc_aabffc8c3327

-rwxrwxrwx 3 root root 64 2020-01-02 18:02 /maxcompute2dfs/maxcompute2dfs_b594eb6a_0861_4302_8fc6_54e466b694ed

[root@izuf6hcv8qx066x6xgjd8gz ~]# hadoop fs -cat /maxcompute2dfs/*

3,测试用户3,女,20,2000-1-1

4,测试用户1,男,20,2000-1-1

2,测试用户2,男,20,2000-1-1
```

# 验证文件存储HDFS版数据同步至MaxCompute

以下示例用来验证文件存储HDFS版数据是否同步到了MaxCompute上。

⑦ 说明 在验证文件存储HDFS版数据同步到MaxCompute中时,文件存储HDFS版上的测试数据是 使用验证MaxCompute数据同步至文件存储HDFS版章节中由MaxCompute同步过去的数据,将该数据再 同步到MaxCompute的另外一张表中。

1. 在MaxCompute中创建新的测试表。

在MaxCompute中的创建表。具体操作,请参见在MaxCompute创建表。

```
CREATE TABLE IF NOT EXISTS dfs2maxcompute
(
id BIGINT COMMENT '编号',
name STRING COMMENT '姓名',
gender STRING COMMENT '性別',
age BIGINT COMMENT '年龄',
birth STRING COMMENT '生日'
);
```

2. 在DataWorks中编写数据同步脚本。

配置MaxCompute Reader和HDFS Writer脚本。具体操作,请参见MaxCompute Reader和HDFS Writer。

```
{
   "type": "job",
   "steps": [
       {
            "stepType": "hdfs",
            "parameter": {
                "path": "/maxcompute2dfs",
                "fileName": "maxcompute2dfs*",
                "datasource": "xxx",
                "column": [
                   {
                        "index": 0,
                        "type": "long"
                    },
                    {
                        "index": 1,
                        "type": "string"
                    },
                    {
                        "index": 2,
                        "type": "string"
                    },
                    {
                        "index": 3,
                        "type": "long"
                    },
                    {
                        "index": 4,
                        "type": "string"
                   }
                ],
                "defaultFS": "dfs://f-xxxxx.cn-xxx.dfs.aliyuncs.com:10290",
                "encoding": "UTF-8",
```

```
"fieldDelimiter": ",",
            "fileType": "text",
            "hadoopConfig": {
                "fs.dfs.impl": "com.alibaba.dfs.DistributedFileSystem",
                "fs.AbstractFileSystem.dfs.impl" : "com.alibaba.dfs.DFS"
            }
        },
        "name": "Reader",
        "category": "reader"
    },
    {
        "stepType": "odps",
        "parameter":{
            "partition":"",
            "truncate":true,
            "compress":false,
            "datasource":"odps first",
            "column": [
                    "id",
                    "name",
                    "gender",
                    "age",
                    "birth"
            ],
            "quid": null,
            "emptyAsNull": false,
            "table": "dfs2maxcompute"
        },
        "name": "Writer",
        "category": "writer"
    }
],
"version": "2.0",
"order": {
   "hops": [
       {
           "from": "Reader",
            "to": "Writer"
        }
    ]
},
"setting": {
   "errorLimit": {
       "record": ""
   },
    "speed": {
        "concurrent": 2,
        "throttle": false
    }
}
```

3. 加载创建的独享数据集成资源组并执行数据脚本。

}

2020-01-02 18:20:20.626 [job-198484218] INFO MetricReportUtil - reportJobMetric is turn off					
2020-01-02 18:20:20.627 [job-198484218] INFO LocalJobContainerCommunicator - Total 4 records, 68 bytes   Speed 38/s, 0 records/s   Error 0 records, 0 bytes   All Task					
WaitWriterTime 0.000s   All Task WaitReaderTime 0.000s   Percentage 100.00%					
2020-01-02 18:20:20.628 [job-198484218] INFO LogReportUtil - report datax log is turn off					
2020-01-02 18:20:20.628 [job-198484218] INFO JobContainer -					
任务启动时刻 : 2020-01-02 18:19:57					
任务结束时刻 : 2020-01-02 18:20:20					
任务总计耗时 : 23s					
任务平均流量 : 38/s					
记录写入速度 : Brec/s					
读出记录总数 : 4					
读写失败总数 : 0					
2020-01-02 18:20:20 INFO					
2020-01-02 18:20:20 INFO Exit code of the Shell command 0					
2020-01-02 18:20:20 INFO Invocation of Shell command completed					
2020-01-02 18:20:20 INFO Shell run successfully!					
2020-01-02 18:20:20 INFO Current task status: FINISH					
2020-01-02 18:20:20 INFO Cost time is: 24.827s					
/home/admin/alisatasknode/taskinfo//20200102/diide/18/19/53/xbk5gekdaru0f5fe2plwk3l6/T3_1191826449.log-END-EOF					
Exit with SUCCESS.					
2020-01-02 18:20:24 [INFO] Sandbox context cleanup temp file success.					
2020-01-02 18:20:24 [INFO] Data synchronization ended with return code: [0].					
2020-01-02 18:20:24 INFO ====================================					
2020-01-02 18:20:24 INFO Exit code of the Shell command 0					
2020-01-02 18:20:24 INFO Invocation of Shell command completed					
2020-01-02 18:20:24 INFO Shell run successfully!					
2020-01-02 18:20:24 INFO Current task status: FINISH					
2020-01-02 18:20:24 INFO Cost time is: 33.315s					
/home/admin/alisatasknode/taskinfo//20200102/datastudio/18/19/49/5u7x8invirwe6f1p7baelcxs/T3 1191826420.log-END-EOF					

4. 查看**文件存储HDFS版**是否成功同步数据到MaxCompute。

select * from dfs2maxcompute; text vie							
			gender	age	birth		
		测试用户1	男		2000-1-1		
		测试用户2	男		2000-1-1		
		测试用户3			2000-1-1		
		测试用户4			2000-1-1		

# 14.在文件存储HDFS版上使用Apache HBase

本文主要介绍在文件存储HDFS版上使用Apache HBase的方法。

## 前提条件

- 已开通文件存储HDFS版服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入门。
- 已为Hadoop集群所有节点安装JDK,且JDK版本不低于1.8。
- 已部署Apache HBase分布式集群,且版本必须与Hadoop版本兼容。具体操作,请参见Apache HBase Reference Guide。

本文使用Hadoop-2.10.1和HBase-2.3.7测试验证。更多信息,请参见HBase官网Hadoop版本兼容性说明。

② 说明 如果您使用的是CDH或者阿里云E-MapReduce中内置的HBase,请参见在文件存储HDFS版上 使用CDH6和在文件存储HDFS版上使用E-MapReduce。

# 步骤一: Hadoop集群挂载文件存储HDFS版实例

在Hadoop集群中配置**文件存储HDFS版**实例。具体操作,请参见挂载管理。

# 步骤二:配置Apache HBase

1. 将Hadoop中配置的core-site.xml复制到\${HBASE\_HOME}/conf目录下。

cp \${HADOOP\_HOME}/etc/hadoop/core-site.xml \${HBASE\_HOME}/conf

2. 配置 \${HBASE\_HOME}/conf/hbase-site.xml 。

```
<property>
   <!-- 使用分布式模式运行HBase -->
   <name>hbase.cluster.distributed</name>
   <value>true</value>
</property>
<property>
   <!-- 指定HBase存储目录,f-xxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com为您的挂载点域名,
根据实际情况修改 -->
   <name>hbase.rootdir</name>
   <value>dfs://f-xxxxxxxxxxx.cn-xxxxxx.dfs.aliyuncs.com:10290/hbase</value>
</property>
<property>
   <!-- 在文件存储HDFS版上使用HBase时该配置必须设置为false -->
   <name>hbase.unsafe.stream.capability.enforce</name>
   <value>false</value>
</property>
<property>
   <!-- 依赖的ZooKeeper配置,请根据实际情况修改 -->
   <name>hbase.zookeeper.quorum</name>
   <value>hostname:2181</value>
</property>
```

## 3. 配置\${HBASE\_HOME}/conf/hbase-env.sh。

```
## HBase不使用自己的ZooKeeper
export HBASE_MANAGES_ZK=false
## 在HBASE_CLASSPATH中添加文件存储HDFS版Java SDK
export HBASE_CLASSPATH=/path/to/aliyun-sdk-dfs-x.y.z.jar:${HBASE_CLASSPATH}
```

# 步骤三:验证Apache HBase

1. 执行以下命令,启动HBase。

 ${HBASE\_HOME}/bin/start-hbase.sh$ 

↓ 注意 启动分布式HBase集群前确保ZooKeeper已经启动。

2. 创建测试文件(例如dfs\_test.txt)并写入如下内容。

```
create 'dfs_test', 'cf'
for i in Array(0..9999)
  put 'dfs_test', 'row'+i.to_s , 'cf:a', 'value'+i.to_s
end
list 'dfs_test'
scan 'dfs_test', {LIMIT => 10, STARTROW => 'row1'}
get 'dfs_test', 'row1'
```

3. 执行以下命令,在HBase中创建表并写入10000条数据。

\${HBASE\_HOME}/bin/hbase shell dfs\_test.txt

其中, dfs\_test.txt 为刚创建的测试文件,请根据实际情况替换。

ROW	COLUMN+CELL
row1	column=cf:a, timestamp=2021-12-01T15:02:43.900, value=value1
row10	column=cf:a, timestamp=2021-12-01T15:02:43.926, value=value10
row100	<pre>column=cf:a, timestamp=2021-12-01T15:02:44.592, value=value100</pre>
row1000	column=cf:a, timestamp=2021-12-01T15:02:52.516, value=value1000
row1001	<pre>column=cf:a, timestamp=2021-12-01T15:02:52.517, value=value1001</pre>
row1002	column=cf:a, timestamp=2021-12-01T15:02:52.517, value=value1002
row1003	column=cf:a, timestamp=2021-12-01T15:02:52.518, value=value1003
row1004	column=cf:a, timestamp=2021-12-01T15:02:52.557, value=value1004
row1005	column=cf:a, timestamp=2021-12-01T15:02:52.558, value=value1005
row1006	column=cf:a, timestamp=2021-12-01T15:02:52.559, value=value1006
10 row(s)	
Took 0.0395 seconds	
COLUMN	CELL
cf:a	timestamp=2021-12-01T15:02:43.900, value=value1

- 4. 使用MapReduce计算行数。
  - 执行命令

```
HADOOP_CLASSPATH=`${HBASE_HOME}/bin/hbase mapredcp` \
${HADOOP_HOME}/bin/hadoop jar ${HBASE_HOME}/lib/hbase-mapreduce-2.3.7.jar \
rowcounter dfs_test
```

• 返回信息

如果计算行数与写入数据一致,则表示Apache HBase配置成功。



如果在环境变量中已配置HADOOP\_HOME及HADOOP\_CLASSPATH也可以执行以下命令进行RowCounter 计算。

\${HBASE\_HOME}/bin/hbase org.apache.hadoop.hbase.mapreduce.RowCounter dfs\_test

↓ 注意 在YARN上执行MapReduce计数前需要先在Hadoop集群中启动YARN服务。

# 15.在文件存储HDFS版上使用Apache Tez

本文主要介绍在挂载文件存储HDFS版的Hadoop集群上安装及使用Apache Tez。

## 前提条件

- 已开通文件存储HDFS版服务并创建文件系统实例和挂载点。具体操作,请参见文件存储HDFS版快速入门。
- 已为Hadoop集群所有节点安装JDK,且JDK版本不低于1.8。更多信息,下载JDK。
- 已根据集群Hadoop版本对Tez重新编译。具体操作,请参见Tez安装。
- 已根据Hadoop版本对Tez重新编译。

本文使用的Apache Hadoop版本为2.8.5, Tez版本为0.9.2, 编译后的两个压缩包如下。

tez-dist/target/tez-0.9.2-minimal.tar.gz //最小压缩包,仅包含Tez依赖 tez-dist/target/tez-0.9.2.tar.gz //完整压缩包,包含Tez和Hadoop依赖

## 背景信息

Apache Tez是构建在Hadoop YARN之上支持分布式DAG(Directed Acyclic Graph)的计算框架,支持通过 复杂的DAG描述并处理大数据任务。Tez主要使用在Apache Hive中,作为Hive的一种执行引擎。与Hive on MR(MapReduce)相比, Hive on Tez具有更好的查询性能和稳定性。

## 步骤一: Hadoop集群挂载文件存储HDFS版实例

在Hadoop集群中配置文件存储HDFS版实例。具体操作,请参见挂载文件系统。

## 步骤二:安装Tez到Hadoop集群

安装Tez共有三种安装方式,差异点如下。

安装方式	说明
(推荐)方式一:安装Tez时使用本身预打包的Hadoop 依赖	<ul> <li>需要将文件存储HDFS版Java SDK放到Tez的依赖中。</li> <li>支持Hadoop集群滚动升级。</li> </ul>
方式二:安装Tez时使用集群环境中的Hadoop依赖	<ul> <li>如果Hadoop环境中已配置<b>文件存储HDFS版</b>Java SDK,则不需要额外配置。</li> <li>不支持Hadoop集群滚动升级。</li> </ul>
方式三:安装Tez时使用额外的Hadoop依赖	<ul> <li>需要将<b>文件存储HDFS版</b>Java SDK放到Tez的依赖中或 者额外的Hadoop依赖中。</li> <li>支持Hadoop集群滚动升级。</li> </ul>

(推荐)方式一:安装Tez时使用本身预打包的Hadoop依赖

### 1. 将编译好的完整压缩包上传至文件存储HDFS版目录下。

i. 执行以下命令, 解压完整压缩包到指定目录(例如tez-0.9.2/)。

```
mkdir tez-0.9.2
tar -zxvf tez-0.9.2.tar.gz -C tez-0.9.2/
```

ii. 执行以下命令,将已下载的文件存储HDFS版Java SDK放到指定目录中(例如tez-0.9.2/lib)。

cp aliyun-sdk-dfs-x.y.z.jar tez-0.9.2/lib

iii. 执行以下命令, 重新压缩文件。

tar -zcvf tez-0.9.2.tar.gz -C tez-0.9.2/ .

iv. 执行以下命令, 查看压缩包中包含**文件存储HDFS版**Java SDK。

```
tar -ztvf tez-0.9.2.tar.gz | grep aliyun-sdk-dfs-x.y.z.jar
```

v. 执行以下命令,将重新压缩的压缩包放到文件存储HDFS版实例目录下。

```
hadoop fs -mkdir -p /apps/tez-0.9.2
hadoop fs -put tez-0.9.2.tar.gz /apps/tez-0.9.2
```

2. 在\${HADOOP\_HOME}/etc/hadoop下新建文件tez-site.xml并同步到Hadoop集群所有节点。

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>tez.lib.uris</name>
<value>${fs.defaultFS}/apps/tez-0.9.2/tez-0.9.2.tar.gz</value>
</property>
<name>tez.use.cluster.hadoop-libs</name>
<value>false</value>
</property>
</configuration>
```

3. 为客户端节点添加Tez依赖库。

将编译好的最小压缩包解压到本地目录,同时将该目录同步到Hadoop集群所有节点。

```
mkdir -p /opt/module/tez-0.9.2
tar -xvzf tez-0.9.2-minimal.tar.gz -C /opt/module/tez-0.9.2
```

- 4. 在Hadoop Classpath中添加tez-site.xml配置和最小压缩包解压后的Jar文件,并同步修改到Hadoop集群所有节点。
  - i. 执行以下命令, 打开hadoop-env.sh文件。

vim \${HADOOP\_HOME}/etc/hadoop/hadoop-env.sh

ii. 在hadoop-env.sh文件末尾添加如下内容。

```
export TEZ_CONF_DIR=${HADOOP_HOME}/etc/hadoop/tez-site.xml
export TEZ_JARS=/opt/module/tez-0.9.2
export HADOOP_CLASSPATH=${HADOOP_CLASSPATH}:${TEZ_CONF_DIR}:${TEZ_JARS}/*:${TEZ_JAR
S}/lib/*
```

# 方式二:安装Tez时使用集群环境中的Hadoop依赖

1. 将编译好的Tez最小压缩包上传至文件存储HDFS版目录下。

```
hadoop fs -mkdir -p /apps/tez-0.9.2-minimal
hadoop fs -put tez-0.9.2-minimal.tar.gz /apps/tez-0.9.2-minimal
```

2. 在\${HADOOP\_HOME}/etc/hadoop下新建文件tez-site.xml并同步修改到Hadoop集群所有节点。

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>tez.lib.uris</name>
<value>${fs.defaultFS}/apps/tez-0.9.2-minimal/tez-0.9.2-minimal.tar.gz</value>
</property>
<property>
<name>tez.use.cluster.hadoop-libs</name>
<value>true</value>
</property>
</configuration>
```

3. 后续操作,请参见方式一中的步骤3和步骤4。

## 方式三:安装Tez时使用额外的Hadoop依赖

使用该方式需要重新编译Hadoop,获取hadoop-dist/target/hadoop-x.y.z.tar.gz压缩包。

- 1. 将编译好的Tez最小压缩包上传至文件存储HDFS版目录下。
  - i. 执行以下命令, 解压最小压缩包到指定目录(例如tez-0.9.2-minimal/)。

```
mkdir tez-0.9.2-minimal-new tez-0.9.2-minimal
tar -zxvf tez-0.9.2-minimal.tar.gz -C tez-0.9.2-minimal/
```

ii. 执行以下命令,将**文件存储HDFS版**Java SDK放到指定目录(例如tez-0.9.2-minimal/lib)。

cp aliyun-sdk-dfs-x.y.z.jar tez-0.9.2-minimal/lib

iii. 执行以下命令, 重新压缩文件。

tar -zcvf tez-0.9.2-minimal-new/tez-0.9.2-minimal.tar.gz -C tez-0.9.2-minimal/ .

压缩完成后,你还可以执行以下命令,查看压缩包中包含文件存储HDFS版Java SDK。

tar -ztvf tez-0.9.2-minimal-new/tez-0.9.2-minimal.tar.gz | grep aliyun-sdk-dfs-x.y.
z.jar

## iv. 执行以下命令,将新压缩包上传至文件存储HDFS版实例目录中。

```
hadoop fs -mkdir -p /apps/tez-0.9.2-minimal
hadoop fs -put tez-0.9.2-minimal-new/tez-0.9.2-minimal.tar.gz /apps/tez-0.9.2-minim
al
```

## 2. 将编译后的Hadoop压缩包上传至文件存储HDFS版目录。

```
hadoop fs -mkdir -p /apps/hadoop-2.8.5
hadoop fs -put hadoop-dist/target/hadoop-2.8.5.tar.gz /apps/hadoop-2.8.5
```

3. 在\${HADOOP\_HOME}/etc/hadoop下新建文件tez-site.xml,并将文件同步到所有Hadoop集群节点。

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>tez.lib.uris</name>
        <value>${fs.defaultFS}/apps/tez-0.9.2-minimal/tez-0.9.2-minimal.tar.gz#tez,${fs
.defaultFS}/apps/hadoop-2.8.5/hadoop-2.8.5.tar.gz#hadoop-mapreduce</value>
    </property>
    <property>
        <name>tez.use.cluster.hadoop-libs</name>
        <value>false</value>
    </property>
    <property>
        <name>tez.lib.uris.classpath</name>
        <value>./tez/*:./tez/lib/*:./hadoop-mapreduce/hadoop-2.8.5/share/hadoop/common/
*:./hadoop-mapreduce/hadoop-2.8.5/share/hadoop/common/lib/*:./hadoop-mapreduce/hadoop-2
.8.5/share/hadoop/hdfs/*:./hadoop-mapreduce/hadoop-2.8.5/share/hadoop/hdfs/lib/*:./hado
op-mapreduce/hadoop-2.8.5/share/hadoop/yarn/*:./hadoop-mapreduce/hadoop-2.8.5/share/had
oop/yarn/lib/*:./hadoop-mapreduce/hadoop-2.8.5/share/hadoop/mapreduce/*:./hadoop-mapred
uce/hadoop-2.8.5/share/hadoop/mapreduce/lib/*</value>
    </property>
</configuration>
```

4. 后续操作,请参见方式一中的步骤3和步骤4。

## 步骤三: 验证Tez

- 1. 检测Hadoop集群Tez引擎是否可运行。
  - i. 执行以下命令, 在**文件存储HDFS版**中生成测试数据。

```
${HADOOP_HOME}/bin/hadoop jar ${HADOOP_HOME}/share/hadoop/mapreduce/hadoop-mapredu
ce-examples-2.8.5.jar \
randomtextwriter \
-D mapreduce.randomtextwriter.totalbytes=10240 \
-D mapreduce.randomtextwriter.bytespermap=1024 \
dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/input
```

其中, f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com 为文件存储HDFS版的挂载点,请根据实际情况替换。

### ii. 执行以下命令,使用Tez对测试数据进行Word Count运算。

\${HADOOP\_HOME}/bin/hadoop jar /opt/module/tez-0.9.2/tez-examples-0.9.2.jar orderedw ordcount dfs://f-xxxxxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/input dfs://f-xxxx xxx.cn-zhangjiakou.dfs.aliyuncs.com:10290/output

#### 其中, tez-examples-0.9.2.jar在Tez最小压缩包的本地解压目录下。

#### iii. 执行以下命令, 查看执行任务引擎是否为Tez。

/opt/module/hadoop-2.8.5/bin/yarn application -status <Application ID>

其中, <Application ID> 为执行任务的Application ID, 请根据实际情况替换。

如果显示如下信息,则表示执行任务的引擎为Tez。



- 2. 测试Hadoop集群MapReduce引擎是否受Tez引擎影响。
  - i. 执行以下命令, 提交MapReduce测试作业。

\${HADOOP\_HOME}/bin/hadoop jar \${HADOOP\_HOME}/share/hadoop/mapreduce/hadoop-mapreduc
e-client-jobclient-2.8.5-tests.jar sleep -mt 1 -rt 1 -m 1 -r 1

## ii. 执行以下命令, 查看任务详情。

/opt/module/hadoop-2.8.5/bin/yarn application -status <Application ID>

其中, <Application ID> 为执行任务的Application ID, 请根据实际情况替换。

如果显示如下信息,则表示MapReduce引擎不受Tez引擎影响。

Application Report :				
_Application-Id : application 1636359861012_0003				
Application-Name : Sleep job				
Application-Type : MAPREDUCE				
User : root				
Queue : default				
Application Priority : 0				
Start-Time : 1636362123547				
Finish-Time : 1636362135975				
Progress : 100%				
State : FINISHED				
Final-State : SUCCEEDED				
Tracking-URL : http://cluster-header-1:19888/jobhistory/job/job_1636359861012_0003				
KPC Port : 3806/				
AM Host : cluster-worker-1				
Aggregate Resource Allocation : 98626 MB-seconds, 23 vcore-seconds				
Aggregate Resource Preempted : 0 MB-seconds, 0 vcore-seconds				
Log Aggregation Status : SUCCEEDED				
Diagnostics :				
Unmanaged Application : false				
Application Node Label Expression : <not set=""></not>				
AM container Node Label Expression : <default_partition></default_partition>				

3. Hive服务验证。

## i. 检测Hive服务是否可以正常调用MapReduce引擎。

a. 执行以下命令, 配置Hive使用MapReduce引擎, 然后进行Count运算。

```
hive> SET hive.execution.engine=mr;
hive> DROP TABLE IF EXISTS user_info_test;
hive> CREATE TABLE user_info_test(user_id BIGINT, firstname STRING, lastname ST
RING, country STRING);
hive> INSERT INTO user_info_test VALUES(1,'Dennis','Hu','CN'),(2,'Json','Lv','J
pn'),(3,'Mike','Lu','USA');
hive> SELECT country,count(*) FROM user_info_test GROUP BY country;
```

## b. 执行以下命令, 查看任务详情。

```
${HADOOP_HOME}/bin/yarn application -status <Application ID>
```

#### 其中, <Application ID> 为执行任务的Application ID, 请根据实际情况替换。

如果显示如下信息,则表示Hive服务可以正常调用MapReduce引擎。


## ii. 检测Hive服务是否可以正常调用Tez引擎。

# a. 执行以下命令, 配置Hive使用Tez引擎, 然后进行Count运算。

```
hive> SET hive.execution.engine=tez;
hive> DROP TABLE IF EXISTS user_info_test;
hive> CREATE TABLE user_info_test(user_id BIGINT, firstname STRING, lastname ST
RING, country STRING);
hive> INSERT INTO user_info_test VALUES(1,'Dennis','Hu','CN'),(2,'Json','Lv','J
pn'),(3,'Mike','Lu','USA');
hive> SELECT country,count(*) FROM user info test GROUP BY country;
```

## b. 执行以下命令, 查看任务详情。

```
${HADOOP_HOME}/bin/yarn application -status <Application ID>
```

#### 其中, <Application ID> 为执行任务的Application ID, 请根据实际情况替换。

# 如果显示如下信息,则表示Hive服务可正常调用Tez引擎。

