Alibaba Cloud
Data Transmission Service

Product Introduction

Issue: 20191101
Legal disclaimer

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

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

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

3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent
ial, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

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

6. Please contact Alibaba Cloud directly if you discover any errors in this document.
# Document conventions

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

Legal disclaimer.................................................................................................I
Document conventions.......................................................................................I
1 Benefits........................................................................................................... 1
2 Features.......................................................................................................... 3
3 Scenarios......................................................................................................... 4
4 System architecture and design concepts.....................................................10
5 Specifications.................................................................................................15
   5.1 Specifications of data migration channels...............................................15
   5.2 Specifications of data synchronization channels.....................................17
1 Benefits

DTS supports data transmission between different data sources, such as RDBMS, NoSQL, and OLAP databases. Multiple data transmission features are available in DTS, including data migration, real-time change tracking, and real-time data synchronization. Compared with third-party data streaming tools, DTS provides multiple types of transmission channels with high performance, security, and reliability. DTS also makes it easy to create and manage transmission channels.

High compatibility

DTS supports data migration between multiple types of homogeneous and heterogeneous data sources, such as from Oracle to MySQL, or from Oracle to Postgres Plus Advanced Server (PPAS). For migration between heterogeneous data sources, DTS supports schema conversion. For example, you can convert a synonym in Oracle to a synonym in PPAS.

DTS supports multiple data transmission features, including data migration, real-time change tracking, and real-time data synchronization. In real-time change tracking and real-time data synchronization, data is transmitted in real time.

Real-time data synchronization supports one-way and two-way synchronization between two data sources. This feature applies to the following scenarios: geo-disaster recovery, active geo-redundancy, nearby application access, query load balancing, and real-time data warehousing.

DTS supports data migration with minimized downtime to ensure availability. The application downtime during data migration is reduced to minutes.

High performance

DTS uses servers with high specifications to ensure the performance of each synchronization or migration channel.

The infrastructure of DTS has been optimized to ensure high-speed and reliable data migration. The peak rate of full data migration can reach 70 MB per second or 200,000 transactions per second (TPS).

DTS provides better synchronization performance compared to traditional data synchronization tools. With the real-time synchronization feature, you can
simultaneously synchronize the incremental data for one table. During peak hours, the data synchronization performance can reach 30,000 records per second (RPS).

**Note:**
The performance indicators are for reference only. The data migration and data synchronization performance of DTS is affected by various factors such as the performance of the source and destination databases, network latency, network bandwidth, and channel specifications.

**Security and reliability**

DTS is implemented based on clusters. If a node in a cluster is down or faulty, the control center moves all tasks from this node to another healthy node in the cluster within seconds.

DTS provides a 24 x 7 mechanism for validating data accuracy in some transmission channels to discover and rectify inaccurate data. This helps ensure reliable data transmission.

Secure transmission protocols and tokens are used for authentication across DTS modules to ensure reliable data transmission. DTS also supports resumable transmission.

**Ease of use**

The DTS console is a visual management interface that provides a wizard-like process to assist you in creating data transmission channels.

*The DTS console* also visualizes information about transmission channels to facilitate channel management, such as transmission status, progress, and performance.

DTS supports resumable transmission and regularly monitors channel status. If DTS detects an error such as network failure or system exception, it automatically fixes the error and restarts the channel. If the error persists, you need to manually repair and restart the channel in the DTS console.
2 Features
3 Scenarios

Data Transmission Service (DTS) supports data migration, change tracking, and real-time data synchronization in various scenarios.

Database migration with minimized downtime

To ensure data consistency, traditional migration requires that you stop writing data to the source database during data migration. Depending on the data volume and network conditions, the migration may take several hours or even days, which has a great impact on your businesses.

DTS provides migration with minimized downtime. Services are always available except when they are switched from the source instance to the destination instance. The service downtime is minimized to minutes. The following figure shows the architecture of data migration.

The data migration process includes schema migration, full data migration, and incremental data migration. During incremental data migration, the data in the source instance is synchronized to the destination instance in real time. You can verify businesses in the destination database. After the verification succeeds, you can migrate businesses to the destination database.
Geo-disaster recovery

If your businesses are deployed in a single zone, service interruption may occur because of force majeure factors such as power failure and network disconnection. In this case, you can build a geo-disaster recovery center in another zone to ensure service availability. DTS synchronizes data between the geo-disaster recovery center and the business center in real time to ensure data consistency. If a failure occurs in the business center, you can switch the traffic to the geo-disaster recovery center.

Active geo-redundancy

With the rapid development of businesses and the increase in the number of users, you may encounter the following issues if you deploy businesses in a single region:

- Users are distributed across a wide range of geographical locations, and distant users have high access latency, which affects user experience.
- The scalability is limited by the capacity of infrastructure in a single region, such as power supply and network bandwidth.

To solve the preceding issues, you can build multiple business units in the same city or different cities. DTS enables two-way real-time data synchronization between business units to ensure global data consistency. If a failure occurs in a business unit, you only need to switch the traffic of this business unit to another business unit. The businesses can be recovered within seconds. The redundancy of multiple business units ensures high availability.
You can also distribute traffic across business units based on a specific dimension. For example, you can reschedule the traffic of each business unit based on the region to allow users to access the nearest node. This reduces network latency and improves user experience. The scalability is no longer limited by the capacity of infrastructure because business units are distributed across different regions.

Custom BI system built with more efficiency

User-created BI systems cannot meet increasing requirements for real-time capability. Alibaba Cloud provides complete BI systems. DTS allows you to synchronize data in real time from user-created databases to Alibaba Cloud BI storage systems, such as MaxCompute. DTS helps you build a custom BI system that meets your business requirements on Alibaba Cloud.

Real-time data analysis

Data analysis is essential in improving enterprise insights and user experience. With real-time data analysis, enterprises can adjust marketing strategies to adapt to changing markets and higher demands for better user experience.

With the change tracking feature provided by DTS, you can acquire real-time incremental data without affecting online businesses. You can use the DTS SDK to synchronize the subscribed incremental data to the analysis system for real-time analysis.
Lightweight cache update policies

To accelerate access speed and improve concurrent read performance, a cache layer is used in the business architecture to receive all read requests. The memory read mechanism of the cache layer can help to improve read performance. The data in the cache memory is not persistent. If the cache memory fails, the data in the cache memory will be lost.

With the change tracking feature provided by DTS, you can subscribe to the incremental data in databases and update the cached data to implement lightweight cache update policies.

Benefits
• Quick update with low latency

The business returns data after the database update is complete. For this reason, you do not need to consider the cache invalidation process, and the entire update path is short with low latency.

• Simple and reliable applications

The complex doublewrite logic is not required for the applications. You only need to start the asynchronous thread to monitor the incremental data and update the cached data.

• Application updates without extra performance consumption

DTS retrieves incremental data by parsing incremental logs in the database, which does not affect the performance of businesses and databases.

Business decoupling

The e-commerce industry involves many different types of business logic such as ordering, inventory, and logistics. If all of these types of business logic are included in the ordering process, the order result can be returned only after all the changes are complete. However, this may cause the following issues:

• The ordering process consumes a long period of time and results in poor user experience.
• The business system is unstable and downstream faults will affect service availability.

With the change tracking feature provided by DTS, you can optimize your business system and receive notifications in real time. You can decouple different types of business logic and asynchronously process data. This makes the core business logic simpler and more reliable. The following figure shows the architecture of business decoupling.
In this scenario, the ordering system returns the result after the buyer places an order. The underlying layer obtains the data changes that are generated in the ordering system in real time by using the change tracking feature. You can subscribe to these data changes by using the DTS SDK, which triggers different types of downstream business logic such as inventory and logistics. This ensures that the entire business system is simple and reliable.

This scenario has been applied to a wide range of businesses in Alibaba Group. Tens of thousands of downstream businesses in the Taobao ordering system are using the change tracking feature to retrieve real-time data updates and trigger business logic every day.

Scalable read capability

A single database instance may not have sufficient resources to deal with a large number of read requests. You can use the real-time synchronization feature of DTS to build read-only instances and distribute read requests across these read-only instances. This allows you to scale out the read capability and relieve the pressure on the primary database instance.
Data Transmission Service

Product Introduction / 4 System architecture and design concepts

4 System architecture and design concepts

This topic describes the system architecture of Data Transmission Service (DTS) and the basic design concepts of its main features.

System architecture

Architecture description

• High availability

Each module in DTS has primary and secondary nodes to ensure high availability. The disaster recovery module runs a health check on each node in real time. When a node exception is detected, the module only requires a few seconds to switch the channel to another healthy node.

• Connection reliability

To ensure the connection reliability of change tracking and data synchronization channels, the disaster recovery module checks for any changes, such as changes of data source address. When a change of data source address is detected, the module dynamically allocates a new connection method to ensure the stability of the channel.
Data migration supports schema migration, full data migration, and incremental data migration. To migrate data with minimized downtime, you must select schema migration, full data migration, and incremental data migration.

For a migration between heterogeneous databases, DTS reads the source database schema, translates the schema into the syntax of the destination database, and imports the schema to the destination instance.

A full data migration requires a long period of time. In this process, incremental data is continuously written into the source instance. To ensure data consistency, DTS starts the incremental data reading module before full data migration. This module obtains the incremental data from the source instance and parses, encapsulates, and locally stores the data.

When a full data migration is complete, DTS starts the incremental data loading module. This module retrieves the incremental data from the incremental data reading module. After reverse parsing, filtering, and encapsulation, the incremental data is migrated to the destination instance.
The real-time data synchronization feature can be used to synchronize incremental data between any two data sources in real time. The real-time data synchronization from OLTP databases to OLAP databases is supported.

The process of creating a data synchronization channel is as follows:

• **Initial synchronization**

  DTS synchronizes the schemas and historical inventory data of the objects that you have selected from the source instance to the destination instance.

• **Real-time synchronization of incremental data**

  After initial synchronization is complete, DTS synchronizes incremental data from the source instance to the destination instance in real time.

DTS uses the following modules for real-time synchronization of incremental data:

• **Incremental data reading module**

  The incremental data reading module obtains raw data from the source instance. After parsing, filtering, and syntax conversion, the data is locally stored. The incremental data reading module connects to the source instance by using a database protocol and obtains incremental data from the source instance. If the source database is ApsaraDB RDS for MySQL, the incremental data reading module connects to the source database by using the binary log dump protocol.


- **Incremental data loading module**

  The incremental data loading module requests incremental data from the incremental data reading module and filters data based on the specified objects to be synchronized. Then, the incremental data loading module synchronizes data to the destination instance without compromising transactional sequence and consistency.

  DTS ensures high availability of the incremental data reading module and incremental data loading module. When a channel exception is detected, the disaster recovery module switches the channel to a healthy node. This ensures high availability of the synchronization channel.

**Design concept of change tracking**

The change tracking feature allows you to obtain incremental data from an RDS instance in real time. You can subscribe to the incremental data on the change tracking server by using DTS SDKs. You can also customize data consumption rules based on your business requirements.

The incremental data reading module on the server side of DTS obtains raw data from the source instance. After parsing, filtering, and syntax conversion, the incremental data is locally stored.

The incremental data reading module connects to the source instance by using a database protocol and obtains the incremental data from the source instance in real time. If the source database is ApsaraDB RDS for MySQL, the incremental data...
reading module connects to the source database by using the binary log dump protocol.

DTS ensures high availability of the incremental data reading module and consumption SDK processes.

- When an exception is detected in the incremental data reading module, the disaster recovery module restarts the incremental data reading module on a healthy service node. This ensures high availability of the incremental data reading module.

- DTS ensures high availability of consumption SDK processes on the server. If you start multiple consumption SDK processes for the same channel, the server pushes incremental data to only one process at a time. If an exception occurs on a process, the server pushes data to another healthy consumption process.
5 Specifications

5.1 Specifications of data migration channels

This topic describes the specifications of each data migration channel and provides performance testing results.

Precautions

The performance indicators provided in this document are only used for reference and are not used as a criterion for product SLA evaluation.

Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>specification</td>
<td>DTS provides data migration channels that have different specifications. The performance of these channels depends on the performance of incremental data migration.</td>
</tr>
<tr>
<td>table quantity</td>
<td>The total number of tables in the test model.</td>
</tr>
<tr>
<td>record size</td>
<td>The size of each record that is migrated during incremental data migration.</td>
</tr>
<tr>
<td>RPS</td>
<td>The number of records per second (RPS) that are changed by INSERT, UPDATE, and DELETE operations in the source database.</td>
</tr>
</tbody>
</table>

Note:

- If an SQL statement contains operations on multiple rows of data, DTS identifies the operations as multiple data changes. If you perform INSERT, UPDATE, and DELETE operations on a data record multiple times, DTS also identifies the operations as multiple data changes.
- DTS identifies each COMMIT operation as a data change.

Test model

Test procedure: Create an incremental migration task between two ApsaraDB RDS for MySQL instances. Then, perform a stress test on the source ApsaraDB RDS for MySQL instance to view the performance of incremental data migration.
Table 5-1: Test environment

<table>
<thead>
<tr>
<th>Instance</th>
<th>RDS instance configuration</th>
<th>Maximum performance</th>
</tr>
</thead>
</table>
| **Source instance** | • Instance type: rds.mys2.8xlarge  
• Memory: 48,000 MB  
• Maximum connections: 2,000 | • Maximum QPS: 18,000  
• Maximum IOPS: 14,000 |
| **Destination instance** | • Instance type: rds.mys2.8xlarge  
• Memory: 48,000 MB  
• Maximum connections: 2,000 | • Maximum QPS: 18,000  
• Maximum IOPS: 14,000 |

Test model:

- The number of test tables is 20.
- Each test table has a primary key.
- The record size is 1 KB.
- Each transaction has an average of two DML operations and one COMMIT operation. The ratio of INSERT, UPDATE, and DELETE operations is 3:1:2.

Test results

<table>
<thead>
<tr>
<th>Source instance region</th>
<th>Destination instance region</th>
<th>Network latency between instances (milliseconds)</th>
<th>Specification</th>
<th>TPS</th>
<th>QPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>China (Hangzhou)</td>
<td>China (Hangzhou)</td>
<td>0.26</td>
<td>small</td>
<td>2,566</td>
<td>8,981</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>China (Hangzhou)</td>
<td>0.26</td>
<td>medium</td>
<td>4,726</td>
<td>16,541</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>China (Hangzhou)</td>
<td>0.26</td>
<td>large</td>
<td>6,378</td>
<td>23,204</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>China (Qingdao)</td>
<td>26</td>
<td>small</td>
<td>2,469</td>
<td>8,641</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>China (Qingdao)</td>
<td>26</td>
<td>medium</td>
<td>4,856</td>
<td>16,996</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>China (Qingdao)</td>
<td>26</td>
<td>large</td>
<td>5,439</td>
<td>20,400</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>China (Beijing)</td>
<td>26</td>
<td>small</td>
<td>2,533</td>
<td>8,866</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>China (Beijing)</td>
<td>26</td>
<td>medium</td>
<td>5,038</td>
<td>17,633</td>
</tr>
<tr>
<td>Source instance region</td>
<td>Destination instance region</td>
<td>Network latency between instances (milliseconds)</td>
<td>Specification</td>
<td>TPS</td>
<td>QPS</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------</td>
<td>---------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>China (Beijing)</td>
<td>26</td>
<td>large</td>
<td>6,829</td>
<td>26,100</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>US (Silicon Valley)</td>
<td>175</td>
<td>small</td>
<td>1,753</td>
<td>6,135</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>US (Silicon Valley)</td>
<td>175</td>
<td>medium</td>
<td>2,837</td>
<td>9,929</td>
</tr>
<tr>
<td>China (Hangzhou)</td>
<td>US (Silicon Valley)</td>
<td>175</td>
<td>large</td>
<td>3,884</td>
<td>15,500</td>
</tr>
<tr>
<td>Singapore</td>
<td>US (Silicon Valley)</td>
<td>198</td>
<td>small</td>
<td>1,104</td>
<td>4,000</td>
</tr>
<tr>
<td>Singapore</td>
<td>US (Silicon Valley)</td>
<td>198</td>
<td>medium</td>
<td>1,724</td>
<td>6,334</td>
</tr>
<tr>
<td>Singapore</td>
<td>US (Silicon Valley)</td>
<td>198</td>
<td>large</td>
<td>2,256</td>
<td>8,300</td>
</tr>
</tbody>
</table>

Note:
The preceding test results show the maximum performance of data migration channels that are configured with different specifications. The performance of incremental data migration cannot be guaranteed in the following cases: The table to be migrated does not have a primary key, the network latency is high, an update hotspot exists, or the source and destination instances have performance bottlenecks.

5.2 Specifications of data synchronization channels

This topic describes the specifications of each data synchronization channel and provides performance testing results.

Precautions

The performance indicators provided in this document are only used for reference and are not used as a criterion for product SLA evaluation.

Terms

- specification: DTS provides data synchronization channels that are configured with different specifications. The performance of these channels depends on the number of synchronized records.
• RPS: the number of SQL statements that are synchronized per second, including BEGIN operation, COMMIT operation, DML statements (INSERT, UPDATE, and DELETE operations), and DDL statements.

Specifications of data synchronization channels

DTS offers four specifications for data synchronization channels based on their maximum performance: micro, small, medium, and large. The synchronization performance of each specification can reach the maximum performance if the following conditions are met:

• The pressure on the source instance must be greater than or equal to the maximum performance that corresponds to each specification.
• The destination instance does not have bottlenecks in write performance and supports the performance pressure that corresponds to each specification.

• The network latency between the DTS server and the source or destination instance is less than or equal to 2 milliseconds.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Maximum performance (RPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>micro</td>
<td>Less than 200</td>
</tr>
<tr>
<td>small</td>
<td>200 to 2,000</td>
</tr>
<tr>
<td>medium</td>
<td>2,000 to 5,000</td>
</tr>
<tr>
<td>large</td>
<td>Unlimited</td>
</tr>
</tbody>
</table>

Note: The online running performance of the large specification depends on the network environment and the performance of the source and destination instances.