Data Transmission Service

Product Introduction

Product Introduction

Data Transmission Service (DTS) is a data exchange service that streamlines data migration, data synchronization, and data subscription. It is applicable for databases running through Alibaba Cloud. DTS takes care of most widely used commercial and open-sourced databases, including RDBMS (Relational Database Management System) and NoSQL.

You can use the rich data transmission capacities of DTS to build a secure, scalable, and high-available data architecture. Applicable scenarios include **zero downtime data migration**, **off-site disaster recovery**, **cross region data synchronization**, and **cache update**.

DTS will take over the complex and heavy work of data transaction, allowing you to concentrate on developing business opportunities. DTS guarantees a 99.5% connection stability, and a 99.999% data reliability.

DTS supports a multitude of database engines, including:

Relational databases, such as Oracle, MySQL, SQLServer, PostgreSQL

NoSQL, such as Redis

DTS supports data transmission between database engines, such as RDBMS, NoSQL, and OLAP, and provides data migration, data subscription, and data synchronization for different scenarios. The following features are available for creating and managing DTS instances.

Zero downtime

Data migration enables you to migrate data between databases with zero downtime. All data changes that occur to the source database during migration are replicated to the target database. After the data migration is completed, the target database remains synchronous with the source database for a period of time. This allows you to switch over the databases at your convenience.

DTS supports data migration and synchronization between homogeneous and heterogeneous data sources, for example, from Oracle to MySQL and from Oracle to Postgres Plus Advanced Server. For migration between heterogeneous data sources, DTS supports the conversion of structural object definition. For example, the schema in Oracle is converted according to the structural definition in Postgres Plus Advanced Server.

Zero downtime data migration minimizes the impact of data migration on applications, streamlining

your data migration to cloud.

High performance

DTS adopts a high configured server to ensure excellent performance of each migration synchronization link.

For data migration, DTS performs a variety of transmission optimization measures on the service base. The transmission speed of existing data migration at its peak can reach 70 MB per second, or 200,000 records per second.

Compared with traditional data synchronization tools, the real-time synchronization function of DTS refines the concurrency granularity to the transaction level. It also supports multiple network optimization methods, such as compressed transmission and packet retransmission, greatly improving the synchronization performance. At its peak performance, the synchronization speed can reach 30,000 records per second.

Connection reliability

DTS guarantees a 99.95% link reliability due to the following features:

DTS is built on service clusters. Any node down or fault in the cluster triggers DTS to immediately switch all tasks on that node to other nodes within the cluster.

DTS provides 24*7 data accuracy validation for some core transmission links to accurately locate and correct abnormal transmission data.

Secure transmission protocols and secure token authentication are also enabled between units for reliable transmission.

Ease of use

DTS provides a visual management interface, as well as wizards to guide you through establishing transmission links. You can easily create your own DTS instance on the console.

The DTS console displays the transmission status and progress, transmission performance, and other information for you to manage your transmission links.

To avoid link interruption resulting from network or system exceptions, DTS provides breakpoint recoverable transmission, and regularly monitors the statuses of all links. On detecting any link exception, DTS reacts to repair or restart the instance automatically. In cases where manual operation is needed, you can also directly fix the link and restart the instance on the DTS console.

Features

Data migration helps you quickly and easily migrate data between different databases. It is applicable for scenarios such as on-premises data to cloud migration, Alibaba Cloud cross-instance migration, and database split and expansion. Both homogeneous migration, such as MySQL to MySQL, and heterogeneous migration, such as Oracle to Alibaba Cloud RDS MySQL, are attainable. DTS also provides basic ETL functions, including multi-level object mapping and data filtering.

Multiple migration types

The complete data migration process consists of three sequential stages. When creating a migration task, you can determine which stages are most applicable to your environment. These stages are:

Schema migration: Migrates structural definition from the source instance to the target instance.

Existing data migration: Migrates the existing data from the source instance to the target instance.

Incremental replication: Replicates the source instance data changes to the target instance. The data changes are captured from the start of the existing data migration until the end of the whole migration progress.

Note: All three stages of data migration are crucial if you want to perform zero downtime data migration.

Migration task

Migration task is the basic unit of data migration. To carry out data migration, you must create a migration task on the DTS console. With a migration task, you need to specify the connection between the source and target instances, the objects to be migrated, migration stages, and additional parameters. You can also perform migration task creation, management, suspension, and deletion on the DTS console.

The migration task displays different status during creation and progress.

Migration status	Description	Available operation
Not started	The migration task has been	Pre-check

	configured, but is not started.	Delete
Pre-checking	The migration task is performing pre-check.	Delete
Pre-check passes	The migration task passes pre-check.	Start Delete
Migrating	The data is being migrated.	Pause Stop Delete
Migration Failed	The migration failed.	Delete
Paused	The migration is halted.	Start Delete
Finished	The migration is completed successfully or cancelled.	Delete

Migration specification

Data migration supports the following list of migration specifications, in form of source database engine to target database engine.

Migration spec.	Schema migration	Existing data migration	Ongoing data replication
Oracle > RDS For MySQL	Support	Support	Not support
MySQL > RDS For MySQL	Support	Support	Support
SQLServer > RDS For SQLServer	Support	Support	Support
PostgreSQL > RDS For PostgreSQL	Support	Support	Support
Redis > Redis	Support	Support	Support

Source instance

You can use the following databases as a source instance for data migration.

- RDS instance
- On-premises databases
- ECS built-in database

Target instance

You can use the following databases as a target instance for data migration.

- RDS instance
- ECS built-in database
- Redis instance

Multiple ETL functions

Data migration supports multiple ETL functions, including:

Multi-level object mapping: Data migration can be conducted between source and target objects with different names. You can configure the mapping relationship through the management console. The object granularity supports database, table, and column.

Data filtering: You can filter the table to be migrated by configuring a standard SQL condition. For example, by specifying the time period, you can choose to migrate the latest data.

DTS monitor

DTS monitors the operating status of migration tasks and detects abnormalities, such as lag or abended processes. If an abnormality is detected during the data migration process, you will immediately receive an alarm message instructing you how to best handle the issue.

Data subscription helps obtain real-time incremental data of specific RDS instances. It is applicable for scenarios such as cache update, message notification, real-time data synchronization between heterogeneous data sources, and real-time data synchronization with complex ETL.

Features

Key features of DTS subsciption include:

Obtain data changes from a RDS instance in a classic network.

Obtain data changes from a RDS instance in a VPC network.

Subscription object

The subscription object includes database and table. You can subscribe incremental data of selected tables as needed.

Incremental data is divided into DML (Data Manipulation Language) and DDL (Data Definition Language). You determine the specific data change type to be subscribed during configuration.

Subscription channel

To subscribe data from a specific RDS instance, you must create a subscription channel (that is, the basic unit for data subscription) and allocate it to the RDS instance through the DTS console.

Upon creation, the subscription channel starts to capture incremental data from the source RDS instance, and stores the captured data locally. Consequently, you can use the DTS SDK to subscribe and consume incremental data from the subscription channel.

One subscription channel can be subscribed to only one downstream SDK. In the case that incremental data of one RDS instance is requested by multiple SDKs, you must create a corresponding subscription channel for each SDK. The subscription target of these channels are the same RDS instance.

Subscription channel displays different status on creation and progress.

Channel status	Description	Operations available
Pre-checking	Subscription instance is performing pre-check.	- Delete - Configure monitor
Not started	Subscription channel has been configured, but not started.	- Start - Delete - Configure monitor
Initializing	The subscription channel is initializing. The process may take up to 1 minute.	- Delete - Configure monitor
Normal	Subscription channel is running.	- View subscription data - Delete - Configure monitor
Abnormal	Subscription channel has encountered an issue.	- View sample code - Delete - Configure monitor

Benefits

DTS subscription offers convenient design features such as:

Subscription object modification

You can add or delete the subscription object when the subscription channel is running.

Visualized subscription data

You can view the data in the subscription channel on the DTS console.

Consumption time modification

You can modify the consumption time of the subscription channel on the DTS console according to your requirements.

Monitoring system

DTS monitors the performance of all tasks and detects problems, such as lag or abended processes. If a task fails, DTS immediately sends an alarm messages to the task owner.

Data synchronization helps you replicate data between two databases in real time. It is applicable for scenarios such as off-site disaster recovery, cross region data synchronization, and data synchronization between OLTP and OLAP.

Features

Key features of DTS synchronization include:

Real-time data synchronization between two RDS instances.

Data synchronization between a RDS instances in a classic network and a RDS instances in a VPC network.

Data synchronization between two RDS instances under different Alibaba Cloud accounts.

Synchronization task

To synchronize data between two instances, you must create a synchronization task (that is, the basic unit of data synchronication) on the DTS console.

Synchronization task displays different status on creation and progress.

Synchronization status	Description	Operations available

Not started	Synchronization task has been configured, but is not started.	- Modify configuration - Start - Delete - Configure monitor
Pre-checking	Synchronization task is performing pre-check.	- Delete - Configure monitor
Pre-check failed	The pre-check failed.	- Modify configuration - Start - Delete - Configure monitor
Initializing	Synchronization task is migrating the existing data from the source instance to the target instance.	- Delete - Configure monitor
Initialize Failed	The synchronization task failed to migrate existing data.	- Modify synchronization objects - Delete - Configure monitor
Synchronizing	The synchronization task is replicating incremental data from the source instance to the target instance.	- Modify synchronization object - Delete - Configure monitor
Failed	Synchronization task failed to replicate incremental data.	- Modify synchronization object - Delete - Configure monitor
Paused	Synchronization task is halted.	- Start - Modify synchronization object - Delete - Configure monitor

Benefits

DTS synchronization offers convenient design features such as:

Synchronization object modification

You can add or delete synchronization objects during replication.

Synchronization exception monitoring

DTS monitors the performance of all tasks and detects problems, such as lag or abended processes. When a task fails, DTS immediately sends an alarm message to the task owner.

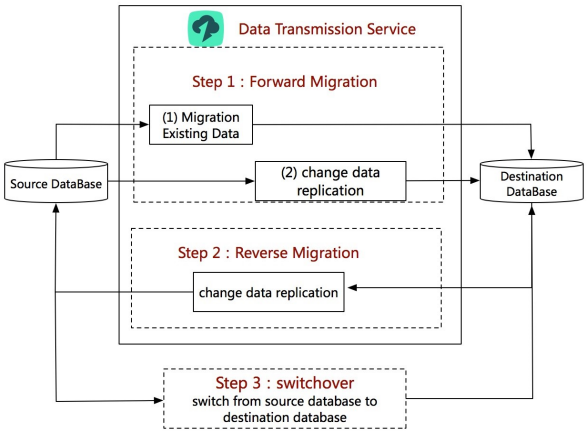
Synchronization performance visualization

DTS console displays crucial information, such as replication lag and task status in real time, for better insight into your key business applications.

DTS delivers data migration, data subscription, and data synchronization processes for the following scenarios.

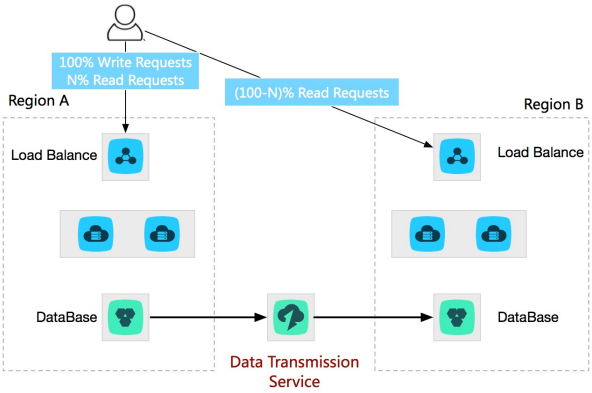
Zero downtime data migration

DTS supports data migration with zero downtime. All data changes that occur to the source database during data migration are continuously replicated to the target database. After the data migration is completed, the target database will remain synchronous with the source database for a period of time. This allows you to switch over the database at your convenience. The following figure demonstrates the basic concept:



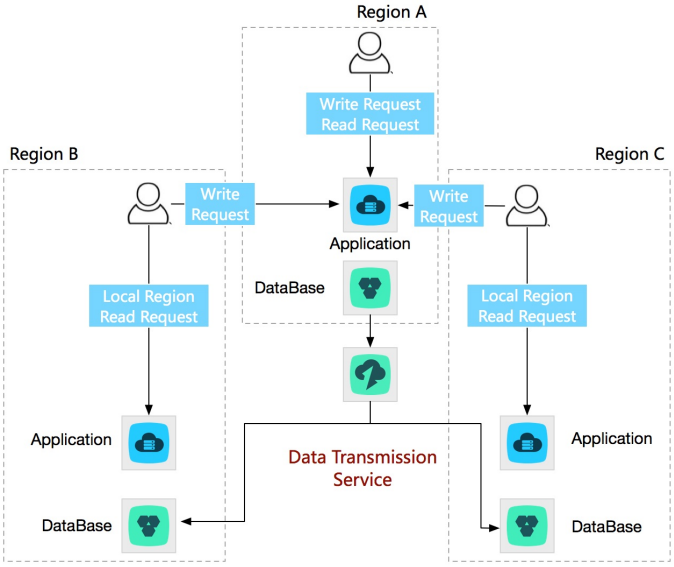
Off-site disaster recovery

With DTS, you can replicate data between two RDS instances in different regions in real time. The off-site disaster recovery instance is then a backup of the primary instance. If a disaster occurs, you can switch all requests from the primary instance to the backup instance to ensure business availability. The following figure demonstrates how DTS can enhance data recover:



Distributed access architecture

When accessing services across extended geographical distances, latency may occur, leading to poor transmission access experience. To improve access experience, DTS provides a distributed access architecture as illustrated in the following figure:



Write requests from all regions are routed to the central unit, where they are synchronized to each local unit in real time through a data transmission service.

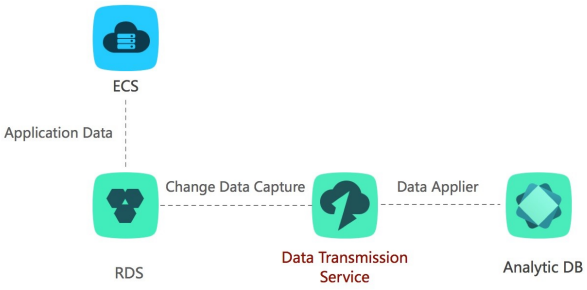
Read requests from each local region are routed to the nearest local unit, which clears the

latency of long-distance access.

Through these request processes, the overall access speed is greatly improved.

Real-time data analysis

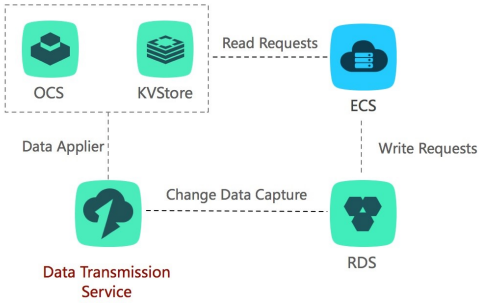
Data subscription helps obtain real-time data information relating to your business without impacting actual online service. The data obtained can be synchronized to the analysis system for real-time analysis via SDK. The following figure demonstrates the basic concept:



Cache update

Integrating a cache layer into your business architecture enables concurrent read access, which typically improves access speed. However, the temporary storage of cache layers can cause data loss in memory during an abnormal exit. DTS mitigates this scenario by coordinating the needs of persistent storage and cache update.

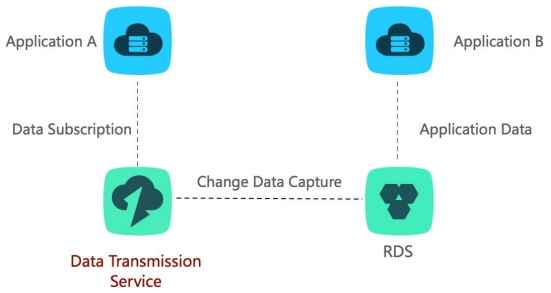
DTS subscribes real time data changes of the RDS instance and then updates the cached data. With data subscription, you can perform low latency cache refreshes without degrading the performance of the RDS instance. The following figure demonstrates the basic concept:



Message notification

Data subscription can be used between coupled services to enable asynchronous coupling through a real time message notification manner. The previously interrelated services can be concurrently started, without degrading the performance. Through asynchronously coupling the core business

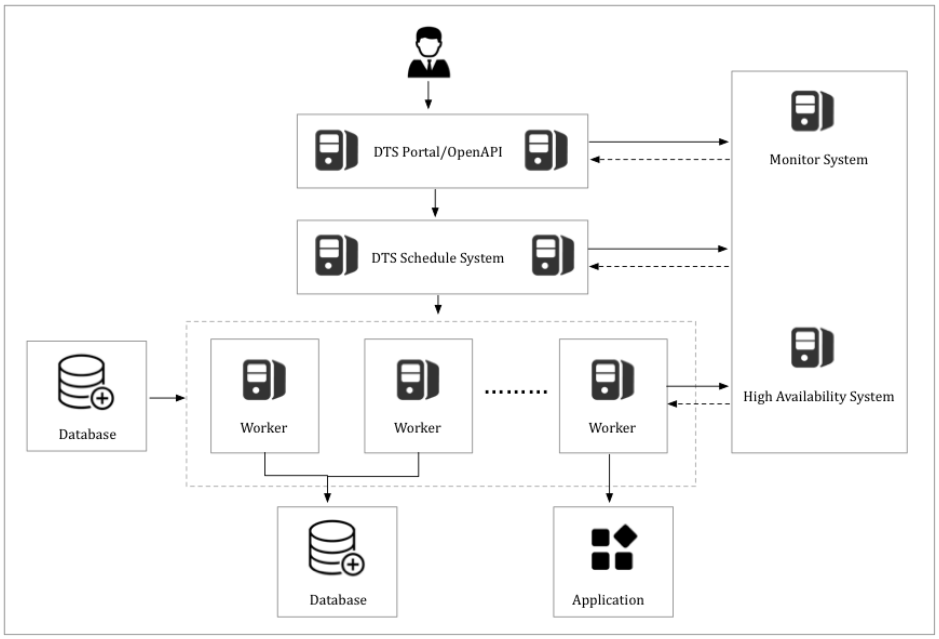
with downstream services, the core business benefits from higher stability and reliability. The following figure demonstrates the basic concept:



This section introduces the system architecture of DTS and the basic design concepts of its main functions.

System architecture

The following figure demonstrates the system architecture of DTS.



System availability

Each unit in DTS owns a set of workers, which includes primary and backup devices to ensure high availability of the whole system. The Disaster Recovery System detects the status of every primary point in real time. If interruption occurs, DTS immediately switches the task to the backup points, and continues the task flow from where it is stopped.

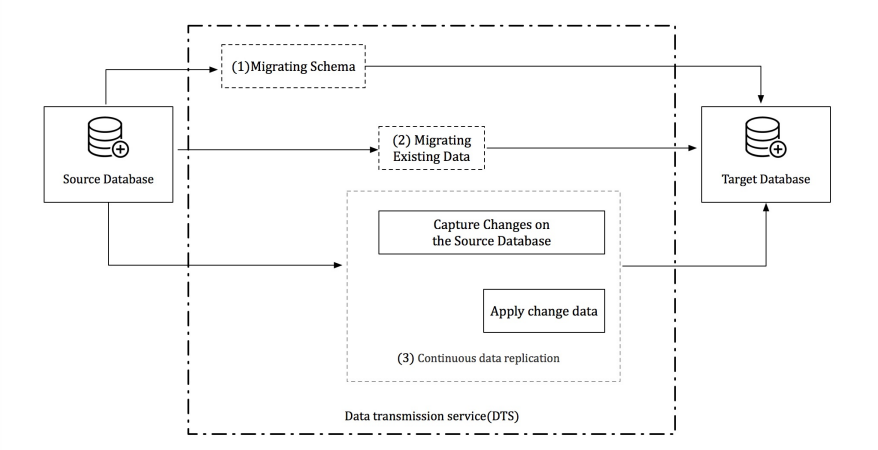
Connection reliability

For data subscription and synchronization instance, the Disaster Recovery System also monitors the source and target endpoints. If connection address changes are discovered, DTS dynamically adapts the new data source connection, and automatically allocates a new connection address for the endpoint.

Design concept of data migration

A complete data migration process in DTS consists of three stages: schema migration, existing data migration, and incremental replication. If all three stages are performed when creating a data migration task, zero downtime data migration is achieved.

The design concept of data migration is as follows:



Schema migration

DTS handles schema conversion in schema migration. When migrating between heterogeneous databases, DTS obtains the syntax definition of the source scheme from the source database, converts the schema into the target database’ s syntax definition format, and imports the modified schema to the target instance.

Existing data migration

In this stage, DTS migrates all the existing data from source to target. To ensure migration consistency, the change data capture unit starts to capture change data on the source database from when the migration starts. The captured change data is then resolved, encapsulated, and stored in local queue.

Incremental replication

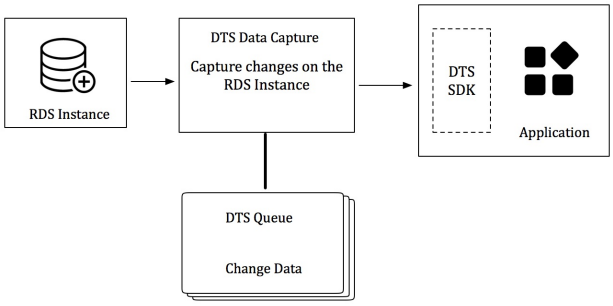
After running data migration, DTS starts the data replication unit. The replication unit pulls change data and replicates it to the target instance after performing mapping, filtering, and

encapsulating processes.

Design concept of data subscription

Data subscription is used to capture the incremental log from specific RDS instances in real time. With the DTS SDK, you can subscribe to the captured data and consume it according to your requirements.

The design concept of data subscription is as follows:



Data subscription instance uses the log capture unit to capture change data from RDS instances. The captured data is stored in a local queue after performing data resolution, filtering, and standard formatting processes. The log capture unit connects the source instance based on the database protocol. For example, if the source instance is RDS MySQL, the log capture unit connects to the source instance through the Binlog dump protocol.

Both the log capture unit and the SDK data consumption process are highly available.

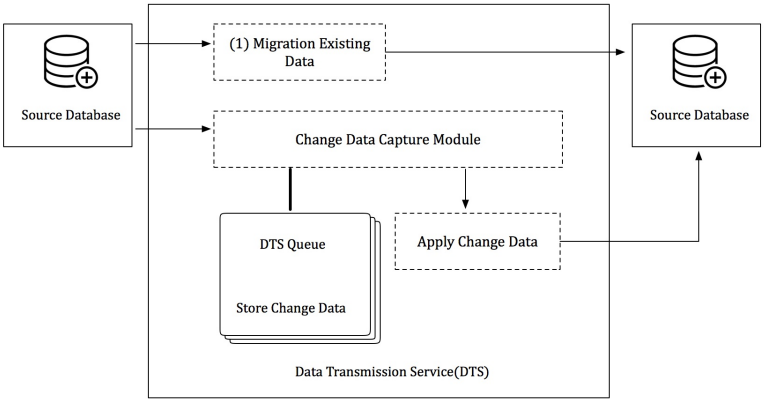
> The Disaster Recovery System continually monitors the subscription instance. In detection of abnormality in the log capture unit, it restarts the log capture unit on another health point from where the process is stopped.

> If multiple SDK consumptions are started for the same subscription instance, DTS only pushes change data to one consumer. In the case of an interruption, DTS switches to another health consumer, and continues to push data from where the process is stopped.

Design concept of data synchronization

Data synchronization supports real-time synchronization between RDS instances and RDS/ADS instances.

The design concept of data synchronization is as follows:



The creation of a synchronization link includes:

Existing data initialization

The existing data of the source instance is replicated to the target instance.

Incremental data synchronization

The incremental data of the source database is synchronized to the target database in real time. Eventually, the source and target databases are dynamically synchronized.

The incremental data synchronization is built on the following units:

Change data capture unit

The change data capture unit reads raw data from the source instance, and stores the data in the local queue after performing data resolution, filtering, and standard formatting processes. It connects the source instance based on the database protocol and reads the incremental data. For example, if the source instance is RDS MySQL, the change data capture unit connects to the source instance through the Binlog dump protocol.

Change data replay unit

The change data replay unit requests incremental data from the change data capture unit, and filters the obtained data according to the customized synchronization object. After that, the replay unit starts to synchronize the log record to the target instance.

Both the change data capture unit and the change data replay unit are highly available.

When detecting link abnormality, the disaster recovery system restarts the link on another health

point from where the process is stopped.

Specification description

Note: This test report only serves as a performance reference for data migration. It is not the criteria of the product SLA.

Introduction

This performance test report provides a basic reference to the performance of different specifications of migration instances. You can use this report as a reference when selecting a specification for your migration instance.

Glossary

Specification

DTS provides different levels of specification for distinguishing the performance of data migration. The specification is based on the incremental data replication performance.

Table count

The total number of tables in the test model.

Record size

The size of each record.

RPS

The amount of records being replicated per second. The record refers to SQL statements, such as begin, commit, insert, and delete.

Test model

Test process

Create a migration task with incremental data replication between two RDS instances.

Perform stress tests on the source RDS instance to check the performance of incremental data replication.

Test environment

Instance	RDS configuration	Reference performance limit
Source instance	- Specification: rds.mys2.8xlarge - Memory (MB): 48000 - Max connections: 2000	- Max QPS: 18000 - Max IOPS: 14000
Target instance	- Specification: rds.mys2.8xlarge - Memory (MB): 48000 - Max connections: 2000	- Max QPS: 18000 - Max IOPS: 14000

Test condition

Number of test tables: 20

Record size: 1 KB

All test tables have primary keys.

Each transaction contains two DML operations and one commit statement on average. The proportional amount between insert, delete, and update statement is 3:1:2.

Test results

Source instance region > Target instance region	Network latency between the source and target instances (Unit:	Specification	TPS	QPS

	millisecond)			
East China 1 (Hangzhou) > East China 1 (Hangzhou)	0.26	small	2566	8981
East China 1 (Hangzhou) > East China 1 (Hangzhou)	0.26	medium	4726	16541
East China 1 (Hangzhou) > East China 1 (Hangzhou)	0.26	large	6378	23204
East China 1 (Hangzhou) > North China 1 (Qingdao)	26	small	2469	8641
East China 1 (Hangzhou) > North China 1 (Qingdao)	26	medium	4856	16996
East China 1 (Hangzhou) > North China 1 (Qingdao)	26	large	5439	20400
East China 1 (Hangzhou) > North China 2 (Beijing)	26	small	2533	8866
East China 1 (Hangzhou) > North China 2 (Beijing)	26	medium	5038	17633
East China 1 (Hangzhou) > North China 2 (Beijing)	26	large	6829	26100
East China 1 (Hangzhou) > West US (Silicon Valley)	175	small	1753	6135
East China 1 (Hangzhou) > West US (Silicon Valley)	175	medium	2837	9929
East China 1 (Hangzhou) > West US (Silicon Valley)	175	large	3884	15500

Asia-Pacific (Singapore) > West US (Silicon Valley)	198	small	1104	4000
Asia-Pacific (Singapore) > West US (Silicon Valley)	198	medium	1724	6334
Asia-Pacific (Singapore) > West US (Silicon Valley)	198	large	2256	8300

The test result indicates the ultimate performance of different kinds of specifications.

Note: To obtain ultimate performance, the following prerequisites must be met:

- Each synchronization table under the synchronization instance has a primary key.
- No update hotspot is found.
- No performance bottlenecks between the source and target instances are found.

Note: This test report only serves as a performance reference for data synchronization. It is not the criteria of the product SLA.

Glossary

Specification

DTS provides different levels of specification for distinguishing the performance of data synchronization. The specification is based on the amount of synchronized records.

RPS

The amount of records being synchronized per second. The record refers to SQL statements, such as begin, commit, insert, delete, update, and DDL.

Data synchronization specifications

Four levels of specification are defined based on the upper limit of synchronization performance (in RPS): micro, small, medium, and large.

The following table illustrates the performance limit of each specification.

Specification	Performance limit (in RPS)
micro	Less than 200

small	200-1999
medium	2000-4999
large	5000-10000

To meet the performance limit for each specification, the following prerequisites must be met:

The performance stress of the source instance is not less than the upper limit performance of the corresponding specification.

The write performance of the target instance meets the performance stress of the corresponding specification.

The network latency between source instance, target instance, and DTS is not more than 2 ms.

No batch update, such as using a SQL statement, is used to update millions of rows.

The RPS includes: begin, commit, DML statements (insert, delete, and update), and DDL operations.

Version	Release date	Content
DTS 4.5.0	2016.12.25	- Real-time synchronization between RDS instance and AnalyticDB is supported.
DTS 4.4.15	2016.11.24	- Online migration of Redis is supported.
DTS 4.4.0	2016.11.15	- RAM (Resource Access Management) is supported.
DTS 4.1.11	2016.5.20	- Online migration of PostgreSQL is supported.

DTS 4.1.1	2016.5.20	- Online migration of SQLServer is supported.
DTS 3.0.0	2015.10.30	- Real time synchronization between RDS instances is supported. - Offline migration of SQLServer is supported.
DTS 2.4.3	2015.8.11	- Automatic creation of migration database is supported. - Automatic collection of the server whitelist of servers that are added to the RDS instance during migration is supported.
DTS 2.4.0	2015.7.28	- Real-time incremental data subscription for RDS MySQL is supported.
DTS V2.3.0	2015.6.5	- SQL Server to SQL Server schema migration and existing data migration is supported.
DTS V2.2.2	2015.5.28	- Incremental data replication between RDS for MySQL instances is supported.

DTS V2.2.1	2015.4.28	The following features are supported: - Oracle to MySQL schema migration and existing data migration. - MySQL to MySQL schema migration, existing data migration and incremental data replication. - Migration at database, table, and column granularity. - Object name mappings for database, table, and column. - Conditioned filtering.

Pre-check

Pre-check is a necessary stage before the start of a migration task. It involves checking pre-conditions that may affect a successful migration, such as the connectivity of the source and target instances, and permissions of migration accounts. If a pre-check fails, you can troubleshoot the issues, and re-run the pre-check.

Schema migration

Schema migration refers to the migration of structure object definition syntax, including tables, views, triggers, stored procedures, storage functions, synonyms, and other structure object syntax. For migration between heterogeneous databases, data types should be mapped in the schema migration stage, and object definition syntax should be adjusted according to the source and target instance definition syntax. For example, in Oracle to MySQL migration, the number type in Oracle will be mapped to the decimal type in MySQL.

Existing data migration

Existing data migration refers to migration of all the existing data in the source instance database

(except for the definition of structure syntax) to the target instance. If you only choose existing data migration when creating a migration task, without selecting incremental data replication, during the migration process the new data written to the source instance will not be migrated to the target instance.

Incremental data replication

Incremental data replication refers to synchronizing the incremental data written to the source instance during the migration process to the target instance. If you choose existing data migration and incremental data replication when creating a migration task, DTS will first implement static snapshot on the source instance, migrate the snapshot data to the target instance, and then synchronize the incremental data written to the source instance during the migration process to the target instance. Incremental data replication is a process of real-time data synchronization between the source instance and target instance. It will not automatically stop. If you want to stop the migration, you need to manually stop the task on the console.

Synchronization initialization

Synchronization initialization refers to initializing the historical data of the object to be synchronized to the target instance before incremental data synchronization of the synchronization link.

The synchronization initialization types can be divided into structural initialization and full-data initialization. Structural initialization is the initialization of the structure definition of the object to be synchronized. Full-data initialization is the initialization of the data of the object to be synchronized.

Synchronization performance

Synchronization performance refers to the number of records synchronized to the target instance per second, and the unit is RPS. Synchronization performance is a specification used for data synchronization service. Different specifications of performance have different number or records synchronized per second. Detailed definitions can be found in Synchronization Specification Definition.

Synchronization latency

Synchronization latency refers to the time difference between the timestamp of the latest data synchronized to the target instance in the source database, and the current timestamp in the source instance. Synchronization latency reflects the data time difference between the target and source instances. When the synchronization latency is 0, it indicates the source instance and target instance are fully consistent.

Subscription channel ID

Subscription channel ID is the unique identity of the subscription channel. After you purchase a subscription channel, DTS will automatically generate a subscription channel ID. When using SDK to consume incremental data, you need to configure the corresponding subscription channel ID. The subscription channel ID corresponding to each subscription channel is displayed in the subscription list of the DTS Console.

Data update

DTS divides data in the database into two types: data update and structure update. Data update modifies the data, but does not modify the structure object definition, such as insert, update, and delete.

Structure update

DTS divides data in the database into two types: data update and structure update. Structure update refers to the modification of the syntax defined by the structure object, such as create table, alter table, and drop view. You can choose whether to subscribe to the structure update when creating a subscription channel.

Data scope

Data scope is the range of incremental data timestamp stored in the subscription channel, and the timestamp of incremental data is the timestamp when the incremental data is applied and written to the transaction log in the RDS instance. By default, only incremental data of the most recent day is retained in the subscription channel. DTS will regularly clean expired incremental data and update the data scope of the subscription channel.

Consumption time point

The consumption time point is the timestamp corresponding to the latest consumed incremental data subscribed to downstream SDKs. The SDK reports the ACK to DTS for every piece of data it consumed. The server then updates and saves the consumption time corresponding to the SDK. If the SDK suffers an exception and restarts, the server automatically pushes the subscription data from the last consumption point.