

Data Transfer Service

Product Introduction

Product Documentation



Copyright Notice

©2013-2025 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Product Introduction

- Overview

- Data Migration

- Data Sync

- Data Subscription (Kafka Edition)

- Strengths

- Supported Regions

Specification Description

- Data Migration Specification

- Data Synchronization Specification

Product Introduction

Overview

Last updated : 2022-09-02 15:04:24

Overview

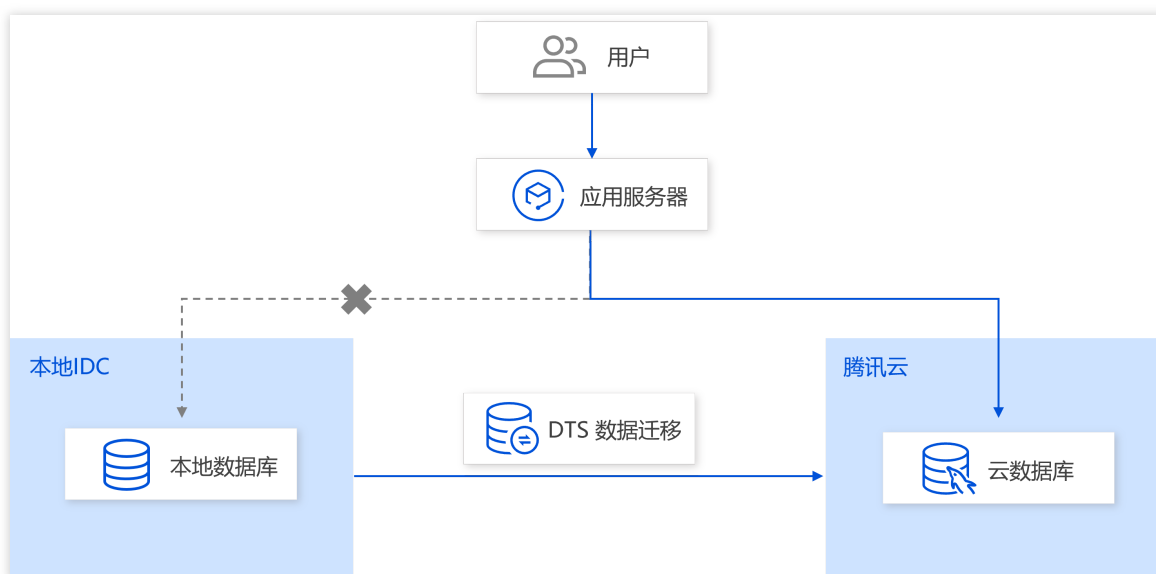
Data Transfer Service (DTS) supports multiple types of databases, such as MySQL, MariaDB, Percona, TDSQL-C, TDSQL for MySQL, PostgreSQL, Redis, MongoDB, and SQL Server. It helps you migrate your databases to the cloud without interrupting your business and build a high-availability database disaster recovery architecture through real-time sync channels. It provides the data subscription feature to meet your requirements for commercial data mining and async business decoupling.

DTS also provides a special edition: [DTS-DBbridge](#), which can be independently deployed on the private cloud. DTS-DBbridge supports data migration and sync between heterogeneous and homogeneous databases, enabling you to migrate an entire database, such as an Oracle database.

Product Features

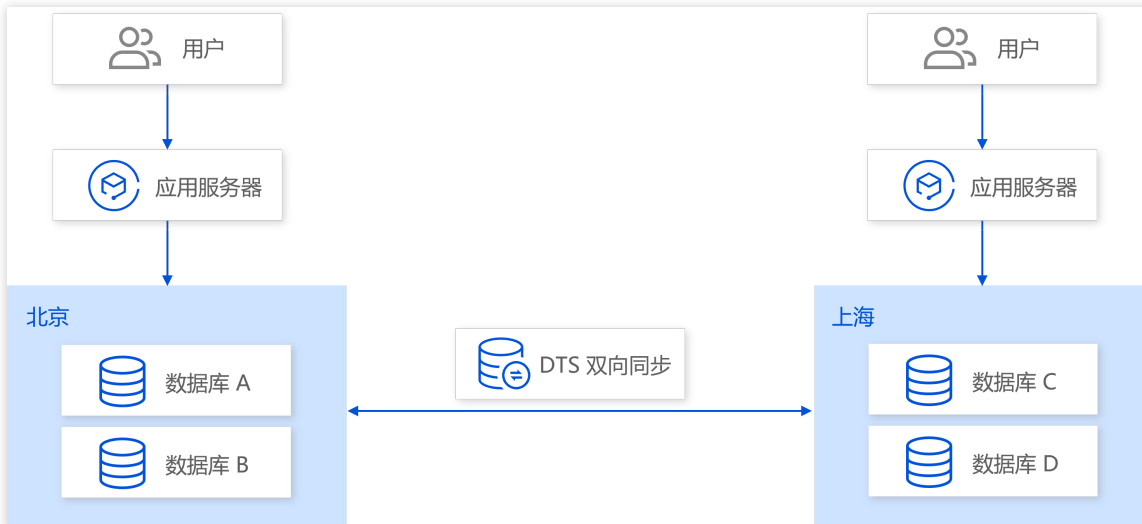
Data migration

Data migration refers to the process of replicating data between different data sources. DTS supports non-stop data migration to minimize the impact of database downtime on the business. This feature can be used to migrate data from local databases, TencentDB databases, and third-party cloud databases to TencentDB.



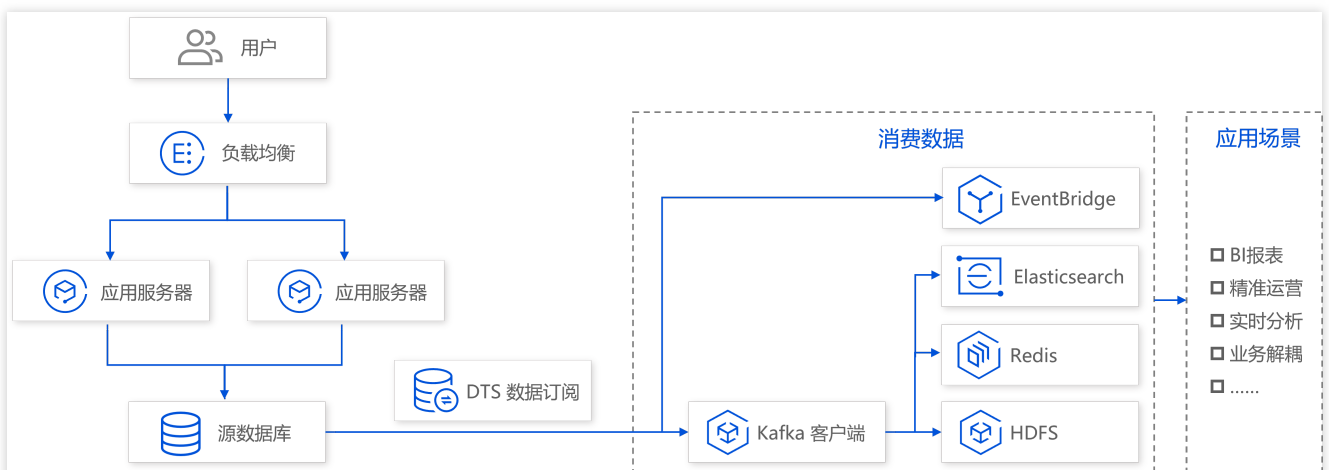
Data sync

Data sync refers to the process of syncing data between two database sources in real time. It is suitable for cloud-local active-active, multi-site active-active, and cross-border data sync as well as real-time data warehousing. While data migration involves moving the entire database in one go, data sync is an ongoing process that will keep the data synced between the source and target databases to ensure consistency.



Data subscription

Data subscription refers to the process where DTS gets the data change information of a key business in the database, converts it into message objects, and pushes them to Kafka for the downstream businesses to subscribe to, obtain, and consume. DTS allows you to directly consume data through a Kafka client, so that you can implement data sync between TencentDB databases and heterogeneous systems, such as cache update, real-time ETL (a data warehousing technology) sync, and async business decoupling.



Data Migration

Last updated : 2023-03-03 14:57:18

Feature Overview

The data migration feature refers to data replication between different data sources. It is a short-term one-time task that migrates the entire database. After migration, manual cutover will be performed to continue the task in the new database. DTS supports non-stop data migration without locks to minimize the impact of database downtime on the business.

DTS supports migration of self-built, TencentDB, and third-party cloud source databases.

Self-built database migration to the cloud: This means to migrate data from a self-built database based on a local IDC, CVM instance, or Lighthouse instance to TencentDB.

Cross-TencentDB instance migration: This includes database version upgrade, cross-region migration (including cross-border migration), and cross-account instance migration.

Third-party database migration: This means to migrate data from a database in another cloud such as Alibaba Cloud or AWS to Tencent Cloud.

How It Works

The following takes MySQL as an example to describe the data migration process:

Source database export: All existing data in the source database is exported.

Data import: Existing data is imported into the target database.

Incremental data sync and data check: Binlog takeover will begin when a migration task is started. During migration, SQL operations in the source database are written to the binlog, which DTS parses to write incremental data generated during existing data migration into the target database.



Typical Use Cases

Data migration to Tencent Cloud

It simply takes a few steps in DTS to set up data migration to Tencent Cloud without any complicated configuration. The migration process does not interrupt the service provided by your source database, thereby minimizing the impact of cloudification on your business.



Restrictions

Basic tables, views, functions, triggers, procedures, and events can be migrated (unsupported migration objects will be gradually supported in the future).

Correlated data objects must be migrated together; otherwise, migration will fail. Common correlations include table referenced by views, view referenced by views, view/table referenced by procedures/functions/triggers, and tables correlated through primary/foreign keys.

Supported Migration Types

DTS supports the following three migration types:

Structural migration: The structure of the migration objects in the source database is migrated to the target database.

Full migration: All data except system tables in the source database is migrated to the target database at a time. Full migration is one-time migration and is applicable to scenarios where the source database has no data writes.

Full + incremental migration: Full migration is performed first to initialize the target database, and then the incremental data is migrated. Technical methods such as log parsing are used to keep the data between the source and target databases consistent. Full + incremental migration is applicable to scenarios where the source instance has data writes.

Notes

If you select full data migration, do not write new data into the source instance during migration; otherwise, the data in the source and target instances will be inconsistent. In scenarios with data writes, to ensure the data consistency in real time, we recommend you select full + incremental migration.

Supported Database Types

For more information on the source and target database types, versions, and migration types supported for data migration, see [Databases Supported by Data Migration](#).

Supported Advanced Features

Feature	Description	Documentation
Heterogeneous migration and migration from third-party cloud databases	Data migration between databases in different types is supported, such as migration from MySQL to TDSQL-C. Currently supported third-party cloud databases include Alibaba Cloud and AWS databases.	Databases Supported by Data Migration
Cross-account migration	Data migration between different Tencent Cloud accounts is supported.	Cross-Account TencentDB Instance Migration
Cross-version migration of most databases	The target database version should be equal to or later than the source database version; for example, data on v5.5.x can be migrated to v5.5.x, v5.6.x, or later. The last digit in the version number is the minor version number, which is not restricted.	Databases Supported by Data Migration
Migration of user information, views,	User information, views, procedures, functions, triggers, and events can be migrated.	-

and advanced objects		
Data consistency check	The data in the source and target databases can be checked for consistency.	Creating Data Consistency Check Task
Task progress visualization	Information such as migration steps and progress can be displayed.	Viewing Task
Metric monitoring and default alarm policy	Data migration metrics can be monitored. Default configuration is supported for data migration event monitoring to automatically notify you of abnormal events.	Supported Events and Metrics
Migration without locks	Migration is implemented without locks, during which no global lock (the FTWRL lock) is added, and only tables without a primary key are locked.	-
Database/Table mapping	Databases/Tables migrated from the source database to the target database can be renamed.	Database/Table Renaming
Instance restart or upgrade	During incremental migration, the instances can be restarted or upgraded.	-
HA switch	HA switch from the source instance (with GTID enabled) to the target instance is supported.	-
No requirements for the super privileges	The operator does not need to have the super privileges of the source instance account (only for certain databases currently).	-

Data Sync

Last updated : 2023-03-03 14:41:40

Feature overview

The data sync feature refers to real-time data sync between two database sources. It is suitable for cloud-local active-active, multi-site active-active, and cross-border data sync as well as real-time data warehousing.

Data migration is a one-time short-term task to migrate the entire database and requires a manual cutover operation after migration for business connection to the new database. In contrast, data sync is a continuous task. After a task is created, the data will be continuously synced (almost in real time) to keep consistency between the source and target databases.

DTS supports the sync of self-built, TencentDB, and third-party cloud databases to TencentDB.

Cloud-local sync: DTS can sync an IDC-based self-built database to a TencentDB instance and vice versa.

Cross-cloud sync: DTS can sync a third-party cloud database to TencentDB instance.

Cross-TencentDB instance sync: DTS can sync TencentDB instances in various scenarios, such as multi-site active-active, cross-border, and cross-Tencent Cloud account sync.

How it works

The following takes MySQL sync as an example to describe the data sync feature, where data is exported from the source instance and imported into the target instance, and key steps include structure initialization, full data initialization, and incremental data processing.

Structure initialization

Structure initialization is to create the same table structure information in the target instance as that in the source instance. When configuring a sync task, you can select whether to sync the table structure. If the target database already contains the same structural information, you simply need to sync the data; otherwise, you'll need to sync the information as well.

Full data initialization

After structure initialization is completed, DTS will initialize the existing data, i.e., exporting all existing data from the source instance and import it into the target instance.

Incremental data processing

In incremental data processing, DTS gets the incremental data continuously through the source instance's binlog and persistently stores such data in the intermediate storage after a series filtering and conversion operations. After importing the full data, it continuously replays the changed incremental data stored in the intermediate storage into the target database in order to ensure data consistency between the source and target databases.



Data conflict resolution

During a DTS sync task, the data in the source and target databases may have conflicts. DTS can check for duplicate table name and primary key conflicts.

Duplicate table name conflict

If the source and target databases have tables with the same name, the task can report an error (**Precheck and report error**) or append the data in the source database to the table with the same name in the target database (**Ignore and execute**).

Primary key conflict

DTS can check for primary key conflicts and provides the following conflict resolution methods. For specific implementation examples, see [Selecting Data Sync Conflict Resolution Policy](#).

Report: During a sync task, if an INSERT statement in the source database has a primary key conflict with the data in the target database, the task will report an error and pause. You need to handle the conflict manually first before proceeding.

Ignore: During a sync task, if an INSERT statement in the source database has a primary key conflict with the data in the target database, the data inserted into the source database will be ignored, and the data in the target database will prevail.

Overwrite: During a sync task, if an INSERT or UPDATE statement has a primary key conflict with the data in the target database, the data in the target database will be overwritten by the inserted or updated data in the source database.

Supported topologies

The basic unit of the sync service is one-way sync. During configuration, you can choose to use data definition language (DDL) or data manipulation language (DML) for sync. By combining one-way sync tasks, you can customize various complex topologies.

In a complex topology, technical measures will be used for DML operations to avoid data loop. However, for DDL, the data sync service will check for loop during configuration to avoid forming DDL loop.

Below are some common topologies, which you can customize by purchasing multiple sync instances. For detailed directions on how to create a complex topology, see [Creating Two-Way Sync Data Structure](#), [Creating Many-to-One Sync Data Structure](#), or [Creating Multi-Site Active-Active IDC Architecture](#).

One-to-one one-way sync



Cascaded one-way sync



One-to-many one-way sync



Many-to-one one-way sync



Two-way sync



Cascaded two-way sync



Typical use cases

By using DTS, you can sync data between MySQL databases in multiple regions to implement multi-site active-active deployment. Database instances in each region can run in the cloud or your self-built IDC.



Restrictions

You can implement two-way sync by combining one-way sync tasks, but there are the following restrictions:

Your business should cooperate in terms of data consistency and cannot update the data record with the same primary key on two nodes; otherwise, a primary key conflict or mutual overwrite may occur. For example, you can choose to update the data records with primary keys `1` , `3` , and `5` on node A and data records with primary keys `2` , `4` , and `6` on node B.

If a data sync conflict occurs, DTS will process the data strictly according to the selected conflict resolution policy. You need to confirm whether the corresponding policy meets your business expectation during configuration.

DML statements support two-way sync, but DDL statements support only one-way sync. To create two-way sync, ensure that DDL sync is disabled in one of the one-way instances.

Supported database types

For more information on the source and target database types, versions, and sync types supported by data sync, see [Databases Supported by Data Sync](#).

Supported advanced features

Feature	Description	Documentation
Two-way sync, ring sync, and many-to-one sync	Complex sync topologies such as two-way sync, ring sync, and many-to-one sync are supported.	Databases Supported by Data Sync
Cross-account sync	Data sync between instances under different Tencent Cloud accounts is supported.	Cross-Account TencentDB Instance Sync
Cross-version sync of most databases	The target database version should be equal to or later than the source database version; for example, data on v5.5.x can be synced to v5.5.x, v5.6.x, or later. The last digit in the version number is the minor version number, which is not restricted.	-
Table conflict check	The duplicate table name check policy is provided.	-
Primary key conflict check	The following three primary key conflict resolution policies are supported: Report: If a primary key conflict of tables is found during data sync, the system will report an error and pause the data sync task. Ignore: If a primary key conflict is found during data sync, the system will keep the primary key record in the target database. Overwrite: If a primary key conflict is found during data sync, the system will use the primary key record in the source database to overwrite that in the target database.	Selecting Data Sync Conflict Resolution Policy
DML and DDL filtering	You can select the data types to be synced, including INSERT, UPDATE, and DELETE. You can select the specific DDL operation, such as <code>CREATE TABLE</code> , <code>CREATE VIEW</code> , and <code>DROP INDEX</code> .	Setting SQL Filter Policy
WHERE conditional filter	You can customize a filter for a single table.	Setting SQL Filter Policy
Sync of views and advanced	Views, procedures, and functions can be synced.	-

objects		
Task progress visualization	Information such as sync steps and progress can be displayed.	-
Metric monitoring and default alarm policy	Data sync metrics can be monitored. Default configuration is supported for data sync event monitoring to automatically notify you of abnormal events.	Supported Events and Metrics
Instance restart or upgrade	During incremental data sync, the source and target instances can be restarted or upgraded.	-
HA switch	HA switch of the source instance is supported if GTID is enabled. HA switch of the target database is supported.	-

Data Subscription (Kafka Edition)

Last updated : 2023-03-02 11:41:52

Feature overview

Data subscription refers to the process where DTS gets the data change information of a key business in the database, converts it into message objects, and pushes them to Kafka for the downstream businesses to subscribe to, get, and consume. DTS allows you to directly consume data through a Kafka/Flink client, so you can build data sync features between TencentDB databases and heterogeneous systems, such as cache update, real-time ETL (data warehousing technology) sync, and async business decoupling.

How it works

The following takes MySQL as an example to describe how data subscription pulls the incremental binlog from the source database in real time, parses the incremental data into Kafka messages, and then stores them on the Kafka server. You can consume the data through a Kafka client. As an open-source messaging middleware, Kafka supports multi-channel data consumption and SDKs for multiple programming languages to reduce your use costs.



Typical use cases

Data archiving

By using the data subscription feature of DTS, you can push the updated incremental data in TencentDB to an archive database or data warehouse as a stream in real time.



Restrictions

Currently, the subscribed message content is retained for 1 day by default. Once expired, the data will be cleared. Therefore, you need to consume the data promptly.

The region where the data is consumed should be the same as that of the subscribed instance.

Data subscription to MySQL, MariaDB, and TDSQL for MySQL does not support geometry data types.

Performance description

In the subscription link, the data parsed by the source database is first written into the Kafka instance built in DTS and then consumed by the client. The performance of write and consumption is as follows:

Scenario	Reference Performance Cap
Data write to built-in Kafka (MySQL/MariaDB/Percona/TDSQL-C for MySQL/TDSQL for MySQL single-shard)	10 MB/s
Data write to built-in Kafka (TDSQL for MySQL multi-shard)	10 MB/s * shard quantity
Data consumption from built-in Kafka	20 MB/s (single-consumer group)
	50 MB/s (multi-consumer group)

The above performance data is for reference only. The actual performance may be compromised by various factors such as high load or network delay in the source database.

Supported subscription types

DTS allows you to subscribe to databases and tables. Specifically, the following three subscription types are supported:

Data update: Subscription to DML operations.

Structure update: Subscription to DDL operations.

Full: Subscription to the DML and DDL operations of all tables.

Consumable data formats

Subscribed data in **ProtoBuf**, **Avro**, or **JSON** formats can be consumed. ProtoBuf and Avro adopt the binary format with a higher consumption efficiency, while JSON adopts the easier-to-use lightweight text format.

Supported advanced features

--	--	--

Feature	Description	Documentation
SDKs for various programming languages	DTS uses the Kafka protocol and supports Kafka client SDKs for multiple programming languages.	-
Metric monitoring and default alarm policy	Data subscription metrics can be monitored. Default configuration is supported for data subscription event monitoring to automatically notify you of abnormal events.	Supported Events and Metrics
Multi-channel data consumption	DTS allows creating multiple data channels for a single database, which can be consumed concurrently through a consumer group.	-
Partitioned consumption	DTS supports partitioned storage of data in a single topic for concurrent consumption of data in multiple partitions, improving the consumption efficiency.	-
Custom routing policy	DTS supports routing data fields to Kafka partitions according to custom rules.	-
Consumption offset change	DTS supports modifying the consumption offset.	Managing Consumer Group

Strengths

Last updated : 2020-06-29 16:49:29

DTS supports data transfer between RDBMS (relational) and NoSQL (non-relational) data sources and offers a wide range of data transfer methods such as data migration, real-time data subscription, and real-time data sync. Compared with third-party data flow tools, DTS provides a wider selection of more secure, reliable, and high-performance transfer linkages as well as convenient features, greatly facilitating the creation and management of transfer linkages.

Data Sync

DTS helps migrate your databases to TencentDB with virtually no need for shutdown during data replication. All the changes made to data in the source database during migration will be replicated to the target database, so the service provided by the source database will not be affected during the process. After the replication is completed, the source and target databases will stay in sync, and you can choose the business switch time as needed.

High Transfer Performance

DTS uses high-spec servers to ensure that every migration or sync linkage has outstanding transfer performance. In terms of data migration, its underlying layer adopts a variety of performance optimizations that make it outperform traditional data migration tools.

Auto Recovery from Failure

DTS boasts an extremely high availability where each node is capable of high-efficiency recovery and self-healing that enable automatic failover in a matter of seconds. Thanks to its high reliability, DTS helps you migrate your data to the cloud with no concerns over data consistency.

Visual Operations

DTS requires no separate drivers or applications or major changes to the source database. To enable DTS, you just need to complete some simple configurations in the visual management interface presented by Tencent Cloud.

Easy and Quick Setup

A migration task can be set within minutes in the DTS Console, where you can specify various parameters as needed to perform migration, including setting the connection between source and target databases, migration type, and objects to be migrated.

Supported Regions

Last updated : 2024-09-02 14:48:40

Supported Regions

DTS is supported in the following regions:

Region	Data migration	Data sync	Data subscription
China	Guangzhou, Shenzhen, Shenzhen Finance, Shanghai, Shanghai Finance, Hangzhou, Nanjing, Beijing, Beijing Finance, Tianjin, Chengdu, Chongqing, Hong Kong (China)	Guangzhou, Shenzhen, Shenzhen Finance, Shanghai, Shanghai Finance, Hangzhou, Nanjing, Beijing, Beijing Finance, Tianjin, Chengdu, Chongqing, Hong Kong (China)	Guangzhou, Shenzhen, Shenzhen Finance, Shanghai, Shanghai Finance, Nanjing, Beijing, Beijing Finance, Tianjin, Chengdu, Chongqing, Hong Kong (China)
Other countries and regions	Singapore, Jakarta, Bangkok, Mumbai, Seoul, Tokyo, Silicon Valley, Virginia, Frankfurt	Singapore, Jakarta, Bangkok, Mumbai, Seoul, Tokyo, Silicon Valley, Virginia, Frankfurt	Singapore, Jakarta, Bangkok, Mumbai, Seoul, Tokyo, Silicon Valley, Virginia, Frankfurt

Region Selection

When purchasing a DTS task, you need to select regions for the source and target instances from the above table of supported regions. The region selection rules differ in different scenarios.

If your database is a TencentDB instance, you need to select the region where the database resides for the purchased DTS task. In this scenario, you don't need to select any AZs because data can be transferred across AZs in the same region.

If your database is self-built or provided by other cloud vendors, you need to select a region closest to the region where the database is deployed when purchasing a DTS task, so that DTS can choose the most time-saving data transfer link.

Below are some sample cases of region selection:

Data transfer direction	Source database region	Target database region	Source instance region selected for DTS task	Target instance region selected for DTS task
Self-built database in IDC >	Xi'an	Shanghai	Chengdu (the region closest to)	Shanghai

TencentDB			Xi'an)	
Self-built database in IDC > TencentDB	Shanghai	Shanghai	Shanghai	Shanghai
Database of other cloud vendors > TencentDB	Hangzhou	Beijing	Hangzhou	Beijing
TencentDB > TencentDB	Guangzhou	Guangzhou	Guangzhou	Guangzhou
TencentDB > TencentDB	Beijing	Guangzhou	Beijing	Guangzhou

Specification Description

Data Migration Specification

Last updated : 2024-12-30 15:33:31

Parameter Description

When purchasing a DTS task, users need to select the specification of the task. Different specifications correspond to different transfer performances. Transfer performance is represented by RPS.

RPS represents the number of rows per second that DTS incrementally migrates from the source database to the target database. For example, if 2000 rows of data are incrementally migrated from the source database to the target database per second, then the RPS is 2000.

Specification

The following specifications are available for the data migration link.

Note :

The RPS (Requests Per Second) may be influenced by several factors including the configuration of the source database, network latency, and the complexity of the data in the source database. The following RPS performance reference for DTS (Data Transfer Service) is based on a benchmark of 1KB per data record. Utilizing larger data records in actual scenarios may result in a decrease in RPS.

Specification	Performance Limit (RPS)
small	2000
medium	5000
large	6000
xlarge	7000
2xlarge	11000

Test Model

Test procedure: Create an incremental migration task between two TencentDB MySQL instances, conduct stress testing on the source database MySQL, and monitor the performance of incremental data migration.

Instance	MySQL Instance Configuration	Reference Limit Performance
Source Database	Instance Specifications: 16-core 32000 MB Memory: 200 GB	Maximum QPS: 12000 Maximum IOPS: 32000
Target Database	Instance Specifications: 16-core 32000 MB Memory: 200 GB	Maximum QPS: 12000 Maximum IOPS: 32000

Test Model:

Number of test tables is 20. Each table has about 2 million records. Each record is about 1 KB.

All test tables have a primary key.

Each Transaction generally contains two DML operations, one is COMMIT, with the ratio of INSERT, UPDATE, DELETE being 1:1:1.

Note :

The test results in this section are based on a benchmark of 1KB per data record. In practical applications, larger data records may result in a decrease in Requests Per Second (RPS).

Test Results

After stress testing the small, medium, and large specifications, the results show that the RPS provided by DTS can be achieved under all specifications. Even in scenarios with the maximum network latency, the RPS performance can reach the upper limit in cross-region (China to global) migration results.

Migration Region Type	Maximum Network Latency	Minimum Network Latency	Specification	RPS
Intra-region (China): Guangzhou > Guangzhou	3.46 ms	2.14 ms	small	2000
	2.51 ms	2.21 ms	medium	5000
	10.1 ms	2.20 ms	large	6000
Cross-region (China): Guangzhou > Beijing	37.6 ms	33 ms	small	2000
	68.2 ms	33.2 ms	medium	5000
	48.2 ms	36.4 ms	large	6000
Intra-region (Global): Singapore > Singapore	697 μ s	240 μ s	small	2000

	376 μ s	273 μ s	medium	5000
	241 μ s	2.11 ms	large	6000
Cross-region (global): Singapore > Silicon Valley	200 ms	174 ms	small	2000
	201 ms	175 ms	medium	5000
	204 ms	175 ms	large	6000
Cross-region (China to global): Guangzhou > Silicon Valley	255 ms	162 ms	small	2000
	309 ms	151 ms	medium	5000
	174 ms	152 ms	large	6000

Data Synchronization Specification

Last updated : 2024-12-30 15:35:45

Parameter description

When purchasing a DTS task, users need to select the specification of the task; different specifications correspond to different transfer performances. Transfer performance is represented by RPS.

RPS represents the number of rows DTS synchronizes from the source database to the target database per second incrementally. For instance, if 2000 rows of data are incrementally synchronized from the source database to the target database per second, then the RPS is 2000.

Specification Description

The following specifications are available for data synchronization links.

Note :

Due to the fact that RPS is influenced by multiple factors such as the configuration of the source database, network latency, and the complexity of the data in the source database, the following RPS performance reference for DTS is based on a benchmark of 1KB per data record. Actual usage with larger data records may result in a decrease in RPS.

Specification	Performance Limit (RPS) Reference
micro	1000
small	2000
medium	5000
large	11000