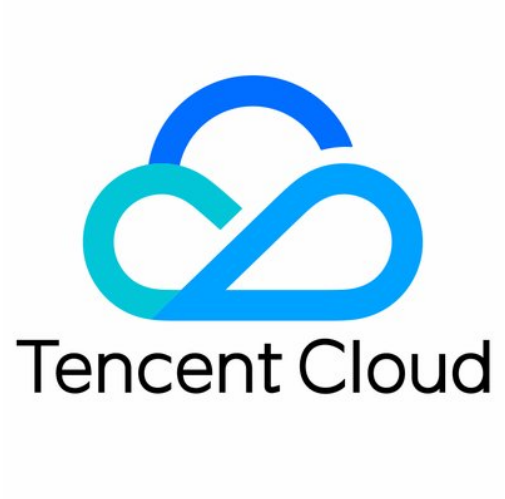
TDSQL for MySQL

FAQs

Product Documentation



Contents

FAQs
InnoDB
General

Last updated：2024-01-06 17:35:36

How do I select the instance specification?

For functional testing in TDSQL for MySQL with no special performance requirements: 2 shards with 2 GB of memory and 25 GB of disk capacity each.
In initial business stage when the total size of data is small but grows fast: 2 shards with 16 GB of memory and 200 GB of disk capacity each.
In stable development stage when sharding is based on actual business demands: 4 shards, and the specification for each equals current business peak x growth rate / 4.
For more information on instance specifications, please see Selecting TDSQL for MySQL Instance and Shard Configurations.

What are the differences between TDSQL for MySQL syntax and traditional MySQL syntax?

Currently, you cannot configure user permissions with command lines. Instead, you need to log in to the TDSQL for MySQL console to do so.
Currently, TDSQL for MySQL does not support custom functions, views, triggers, foreign keys, etc.
For more information on the compatibility with MySQL's syntax, please see Use Limits.

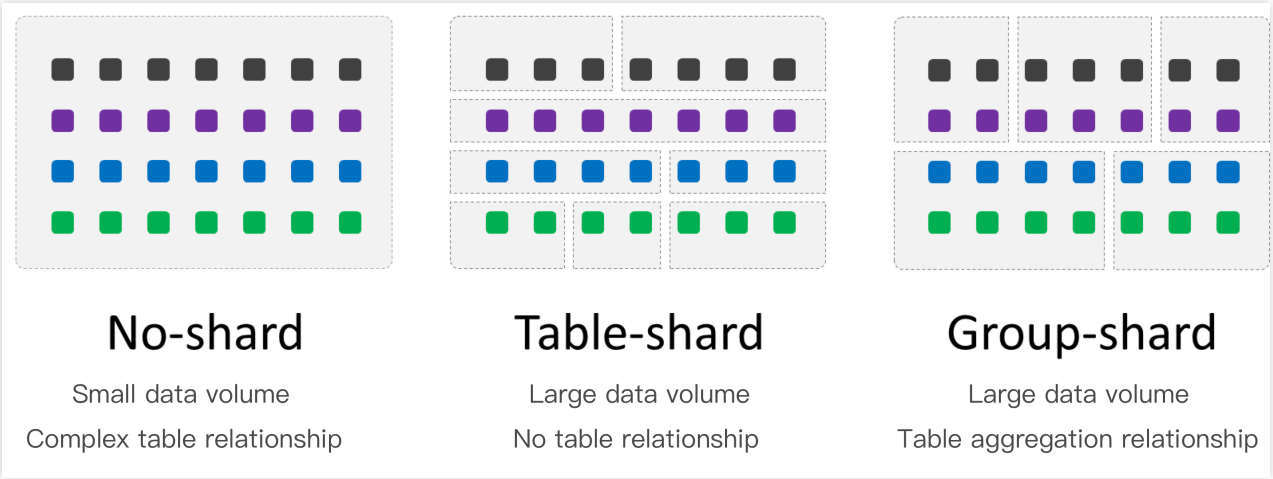
What does the shardkey do?

We recommend that you specify the shardkey field in a SELECT statement when querying a sharded table. The proxy can route a SQL request to the corresponding shard based on the hash value of the shardkey field; otherwise, the request needs to be sent to all shards in the cluster for execution, and then the proxy aggregates all the result sets returned by the shards, which lowers the efficiency of execution.
You need to specify the shardkey field in an INSERT or REPLACE statement when querying a sharded table; otherwise, the SQL statement is not allowed to execute, because the proxy does not know which shard this SQL statement should be routed to.
You need to specify the shardkey field in the WHERE condition in a DELETE or UPDATE statement when querying a sharded table; otherwise, for security reasons, this SQL statement is not allowed to execute.

How do I select the shardkey?

The shardkey is a data table field used to generate the sharding rules during horizontal sharding, which should be specified when creating a table. In TDSQL for MySQL, we recommend that the shardkey be the field of the data on

which most (or core) database operations are performed. Such a table sharding solution is called group-shard, as shown below:



Group-shard solution can ensure that some of the associated data and complex business logical operations of different sharded tables can be aggregated into one physical shard. For example, if both the order table and user table of an e-commerce platform are sharded based on the `UserID`, it is quick to calculate how many orders a user has recently placed through join queries (without cross-node joins or distributed transactions).

Some typical cases of shardkey selection are as follows:

For user-based Internet applications, most (or core) database operations are based on users, so the field corresponding to user data can be used as a shardkey.

For e-commerce applications or O2O applications, most (or core) database operations are based on sellers and buyers, so the field corresponding to seller or buyer data can be used as a shardkey. In the case where super sellers account for the vast majority of transactions, the load and pressure on some shards will increase significantly.

For game applications, most (or core) database operations are based on players, so the field corresponding to player data can be used as a shardkey.

For Internet of Things (IoT) applications, most (or core) database operations are based on IoT information, so the field corresponding to the data of sensors, independent devices, or IMEIs of SIM cards can be used as a shardkey.

For taxation/industry and commerce/social insurance applications, most (or core) database operations of frontend businesses are based on the information of taxpayers, legal representatives, and residents, so the field corresponding to the data of taxpayers or legal representatives can be used as a shardkey.

In most other types of cases, the field of the data on which most (or core) database operations are based can be found in the same way, but there are certain restrictions on the selection of shardkey. For more information, please see Shardkey Selection Restrictions.

Can the shardkey be changed?

Once selected, the shardkey cannot be changed. If you want to modify the shardkey of a table, you need to create a new table.

To modify a shardkey value in a row of a sharded table, you need to INSERT a new value and DELETE the old one. You cannot UPDATE it.

Disk Overuse

Last updated：2024-01-06 17:35:36

Overview

Starting from August 3, 2020, TDSQL for MySQL will lock instances with disk utilization of 150% or more to prevent data loss. Locked disks cannot be written to. Please organize your instance's storage space, and clean up or expand in advance the disks that run the risk of being overused.

Disk space usage

Data space: the space which your data takes up.
System file space: the space which system tablespace files, redolog, undolog, and temp files take up.
Note:
Tencent Cloud offers space for binlog for free, therefore binlog does not take up any purchased disk space.

Causes of Disk Overuse

The following may cause disk overuse:
Too much data: as businesses expand, new data is constantly inserted, resulting in data space growth.
Temp files too large: temp tables are generated when complex query statements with `order by` or `group by` and the `alter table` statements are executed. Small temp tables are stored in memory, while large temp tables in disks.
System files too large: during the database installation, some system files will be initialized to maintain normal operation. If transactions are not submitted for an extended time and a large number of UPDATE, INSERT and DELETE operations are executed, the log recording transaction information may be too big.

Solutions

If a disk is overused, we recommend that you identify the cause and troubleshoot the issue as follows:
If the cause is data taking up too much space, you can delete legacy tables that are no longer in use to free up space. You can also expand your instance disk specification in the console; the instance can be read from and written to after the expansion is completed.
If the cause is temp files taking up too much space, you can optimize the `order by` or `group by` query statements for your application, and monitor and clear sessions and transactions that take a long time to execute. These will reduce the number of temp files.

If the cause is system files taking up too much space, it may be due to existing extended queries and the ibdata1 file is oversized. You can monitor and clear sessions as well as transactions that take a long time to execute to reduce system file redundancy.

IO Metric Exception

Last updated：2024-01-06 17:35:36

Exception

Log in to the TDSQL console and click an instance ID to enter the instance management page. On the **Monitoring and Alarms** tab, you can view three IO metrics.

Metric	Category	Description
IOUsageRate	Instance monitoring	Maximum IO utilization of the source node
IOUsageRateShard	Shard monitoring	IO utilization
IOUsageRateNode	Node monitoring	IO utilization

In actual use, the above three IO metrics may increase abnormally even during off-peak hours. Alarms configured for them, if any, will be triggered.

Exception causes

Possible causes of the abnormal IO increase include:
There is a calculation problem with the IO metrics, and the actually displayed values are IO values of the server where the instance and node reside.
The IO levels of other instances on the same server increase, which may affect the IO monitoring of the current instance.

Solution

This issue involves alarms. We have located it and are fixing it. You can try the following workarounds for the time being:
Do not configure alarms for these three IO metrics. Or, increase their alarm thresholds to avoid the impact of abnormal alarms.
Ignore these three IO metrics when they increase abnormally and don't match the business usage.
We are monitoring and handling the IO usage issue uniformly to ensure the normal operations of instances.