

Tencent Cloud TCHouse-D

FAQs

Product Documentation



Copyright Notice

©2013-2024 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

FAQs

Common Operational Issues

Common Errors

FAQs

Common Operational Issues

Last updated : 2024-07-31 09:20:06

Does Tencent Cloud TCHouse-D support renaming columns?

Since version 1.2.0, if you enable `"light_schema_change"="true"` . Optionally, you can modify the column name.

Before version 1.2.0 or when the `"light_schema_change"="true"` option is not enabled, you cannot modify the column name.

In the Doris kernel used by Tencent Cloud TCHouse-D, column names are directly written into the data file, and queries find the corresponding columns via column names. Therefore, renaming a column is not just a simple metadata modification but also involves rewriting data, which is a heavy operation. Renaming columns is supported after enabling light schema change, but this feature is currently immature and should be used with caution.

Can the tables of the Unique Key model support the creation of materialized views?

Not support.

The table of the Unique Key model is business-friendly because of its unique feature of deduplication based on primary keys, which can easily synchronization business databases with frequently changing data. Therefore, many users will first consider using the Unique Key model when connecting data.

Unfortunately, **it is not possible to create a materialized view for a table in the Unique Key model.** The reason is that the essence of materialized views is to pre-calculated data through pre-calculation so that the calculated data can be directly returned during the query to speed up the query. In a materialized view, the pre-calculated data is usually some aggregated metrics, such as sum and count. In this case, if the data changes, such as being updated or deleted, the pre-calculated data cannot be updated synchronously because the detailed information has been lost. For example, a sum of 5 may be 1+4 or 2+3. Because of the loss of detailed information, we cannot know how the sum is calculated and cannot meet the update requirements.

Is the information viewed through show backends/frontends incomplete?

When executing statements such as `show backends/frontends` , you may find that some columns in the results are incomplete. For example, information such as disk capacity cannot be viewed in the result of `show backends` .

Usually, this problem occurs when the cluster has multiple FEs. If the user connects to a non-Master FE node to execute these statements, the information seen will be incomplete. This is because such information only exists on the Master FE node. For example, BE's disk usage information. Therefore, only the direct connection to Master FE allows you to get complete information.

Of course, users can also execute `set forward_to_master=true;` before executing these statements. After

this session variable is set to true, some subsequent information viewing statements will be automatically forwarded to Master FE to obtain results. In this way, no matter which FE the user is connected to, the complete results can be obtained.

Unique Key model query results are inconsistent?

In some cases, when a user uses the same SQL to query a table of the Unique Key model, inconsistent query results may occur. The query results always vary between 2-3 types.

This may be because data with the same key but different values appear in the same batch of imported data, which will lead to inconsistent results between different replicas due to the uncertain order of data overwriting.

For example, the table definition is k1, v1. A batch of imported data is as follows:



```
1, "abc"  
1, "def"
```

Then, the result of replica 1 may be `1, "abc"` , and the result of replica 2 is `1, "def"` . This results in data inconsistency. To ensure that the data sequence between different replicas is unique, see the [Sequence column](#) feature.

Common Errors

Last updated : 2024-07-31 09:20:19

Query error: Failed to get scan range, no queryable replica found in tablet: xxxx.

This is because the corresponding tablet cannot find a replica to query. The usual reasons may be BE downtime, replica missing, and so on. First, you can use the `show tablet tablet_id` statement and then execute the following `show proc` statement to view the replica information of this tablet and check whether the replica is complete. You can also use `show proc "/cluster_balance"` information to query the progress in replica scheduling and repair within the cluster.

For commands related to data replica management, see [Data Replica Management](#).

tablet writer write failed, tablet_id=27306172, txn_id=28573520, err=-235 or -215 or -238

This error usually occurs during data import. The error code is -235 or -238.

The -235 error means that the data version of the corresponding tablet exceeds the maximum limit (the default is 500, controlled by the BE parameter `max_tablet_version_num`), and subsequent writes will be rejected. For example, the error in the question means that the data version of tablet 27306172 exceeds the limit. This is usually because that import is too frequent, which is greater than the compaction speed of the backend data, resulting in version accumulation and eventually exceeding the limit. At this time, you can first use the `show tablet 27306172` statement and then execute the `show proc in the result` statement to view the status of each tablet replica. The `versionCount` in the result indicates the number of versions. If you find that a replica has too many versions, you need to reduce the import frequency or stop import and observe whether the number of versions decreases. If the version number still does not decrease after the import is stopped, you need to view the `be.INFO` log of the corresponding BE node, search for the tablet ID and the `Compaction` keyword, and check whether the `Compaction` is running normally.

The -238 error usually occurs when the amount of data imported in the same batch is too large, resulting in too many segment files for a tablet (the default is 200, controlled by the BE parameter `max_segment_num_per_rowset`). At this time, it is recommended to reduce the amount of data imported in a batch, or appropriately increase the BE configuration parameter value to solve the problem.

tablet 110309738 has few replicas: 1, alive backends: [10003]

This error can occur during a query or import operation. Usually, it means that an exception has occurred in the corresponding tablet replica.

In this case, you can first check whether the BE node is down, for example, the `isAlive` field is false, or the `LastStartTime` is a recent time (indicating that it has been restarted recently). If BE is down, you need to submit a ticket to [Contact Us](#) for troubleshooting. If no BE node is down, you need to use the `show tablet 110309738`

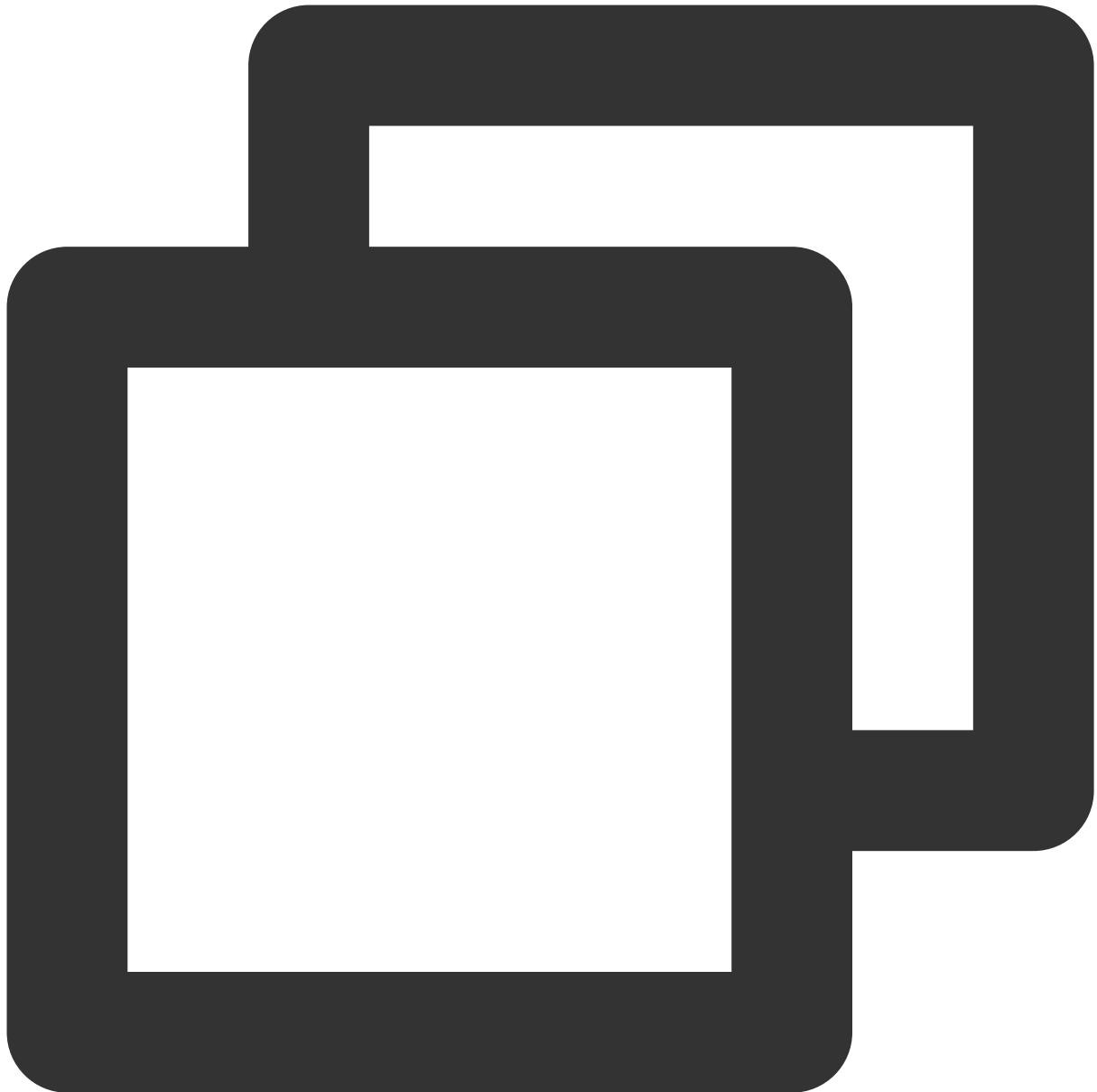
statement and then execute the `show proc in the result` statement to view the status of each tablet replica for further troubleshooting.

disk xxxxx on backend xxx exceed limit usage

Usually, it appears in operations such as Import and Alter. This error means that the usage of the disk of the corresponding BE exceeds the threshold (95% by default). In this case, you can first execute the `show backends` command, where `MaxDiskUsedPct` shows the disk with the highest utilization on the corresponding BE. If it exceeds 95%, this error will be reported. You can choose to manually delete some data to free up the space or scale out the cloud disk to solve this problem. If the disk utilization increases exceptionally, you can submit a ticket [Contact Us](#) for troubleshooting.

-214 Error

When performing operations such as import and query, you may encounter the following errors:



```
failed to initialize storage reader. tablet=63416.1050661139.aa4d304e7a7aff9c-f0fa7
```

The -214 error means that the data version for the corresponding tablet is missing. For example, the above error indicates that the data version of the replica of tablet 63416 on BE 192.168.100.10 is missing. (There may be other similar error codes, which can be troubleshot and repaired in the following way).

Usually, if your data has multiple replicas, the system will automatically repair the problematic replicas. You can perform troubleshooting by following these steps:

1. By using the `show tablet 63416` statement and executing the result `show proc xxx` statement, you can view the status of each replica of the corresponding tablet. Usually, we need to care about the data in the `Version`

column.

Normally, the versions of multiple replicas of a tablet should be the same, and it is the same as the VisibleVersion version of the corresponding partition.

2. You can use `show partitions from tblx` to view the corresponding partition version (the partition corresponding to the tablet can be found in the `show tablet` statement.)

3. You can also visit the URL (open it in the browser) in CompactionStatus column of the `show proc` statement to view the concrete version information and find out which versions are lost.

If there has been no automatic repair for a long time, you need to use the `show proc "/cluster_balance"` statement to view the tablet repair and scheduling tasks currently being executed by the system. This may be because there are a large number of tablets waiting to be scheduled, resulting in a lengthy repair time. You can follow the records in `pending_tablets` and `running_tablets` .

4. Furthermore, you can use the `admin repair` statement to specify a table or partition to be repaired first. For details, see `help admin repair` .

If the problem still cannot be fixed, then in the case of multiple replicas, we can execute the `admin set replica status` command to force the problematic replica to go offline. For details, see `help admin set replica status` for the example of setting the replica status to bad. (After being set to bad, the replica will no longer be visited. But before the operation, ensure that other replicas are normal).

Not connected to 192.168.100.1:8060 yet, server_id=384

This error may be encountered when users import or query. If you check the corresponding BE log, you may also find similar errors. This is an RPC error, and there are two possible causes:

1. The corresponding BE node is down.
2. RPC is congested or other errors occur.

If the BE node is down, you need to check the specific cause of the downtime. Here are some suggestions for solving the problem of RPC congestion:

One case is OVERCROWDED, which means that the RPC source has a large amount of unsent data that exceeds the threshold.

1. `brpc_socket_max_unwritten_bytes` : The default value is 1 GB. If the unsent data exceeds this value, an error will be reported. This value can be modified appropriately to avoid OVERCROWDED errors. (But this is only a temporary solution and congestion still occurs).
 2. `tablet_writer_ignore_eovercrowded` : The default value is false. If it is set to true, OVERCROWDED errors during import will be ignored. This parameter is mainly used to avoid import failure and improve import stability.
- The second case is that the rpc packet size exceeds `max_body_size`. This problem may occur if the query contains a very large string type or a bitmap type. This can be avoided by modifying the following BE parameters:
- `brpc_max_body_size` : The default value is 3 GB.