TDSQL-H LibraDB

常见问题

腾讯云

文档目录

常见问题
产品介绍

最近更新时间：2023-08-02 15:44:12

什么是 TDSQL-H LibraDB?

TDSQL-H LibraDB 是腾讯云自主研发的分布式 HTAP 数据库。基于可插拔式引擎设计、强大的数据融合能力和云原生系统架构，为用户提供一体化产品体验。支持海量数据处理，无论是高负载事务，或是复杂分析，均能出色完成。

什么是 LibraSQL 分析引擎?

LibraSQL 分析引擎是 TDSQL-H LibraDB 的核心组件之一。LibraSQL 分析引擎高度兼容 ClickHouse 引擎，提供了 HTAP 一体化体验和能力。并针对 ClickHouse 不足，研发了新特性和增强能力，例如，支持 update/delete 实时一致性、支持 Read Committed 级别的一致可见性、提供会话管理等。

什么是 TDSQL-H LibraDB 可插拔设计?

TDSQL-H LibraDB 通过 CDC 高速链路与分析引擎，可为您已有的 OLTP 数据库提供一体化分析体验，让您无需重建 OLTP 实例。一个分析引擎可绑定多个 OLTP，并能按需解除绑定，满足数据多合一的需求，充分发挥分析引擎对海量数据的支持能力。

支持的 OLTP 和分析引擎有哪些?

TDSQL-H LibraDB 从架构上支持多种 OLTP 和分析引擎的选择。目前支持的 OLTP 为 MySQL，分析引擎为 LibraSQL，更多的支持能力会陆续推出。

什么是 CDC?

CDC（Change Data Capture）是连接 OLTP 与分析引擎的数据高速通路。您在 OLTP 的数据，会通过 CDC 实时同步到分析引擎。CDC 提供了数据过滤，数据映射，数据多合一以及自动化的异构数据转换等能力。

什么是分区?

分区（Partition）是表的分区，具体的 DDL 操作关键词是 PARTITION BY，指的是一个表按照某一列数据（例如，按月，按日或按事件类型）进行分区。为了减少需要操作的数据，每个分区都是分开存储的。访问数据时，TDSQL-H LibraDB 尽量使用这些分区的最小子集。

什么是分片?

TDSQL-H LibraDB 中的实例节点都可称为 一个分片（Shard）。为了提高查询效率，LibraSQL 引擎将数据分散在您所选择的多个分片上，从而降低单节点的数据扫描数量，提高查询性能。

帐号权限

最近更新时间：2023-12-01 10:06:41

用户创建 CDC 任务时报源端账号权限错误?

在源数据库对 CDC 任务的整个实例或指定对象授权，需要的源端账号权限请参见 添加 CDC 中的注意事项。

如何将 LibraSQL 账号权限调整为只读?

通过控制台的修改权限功能将账号权限调整为只读。具体操作请参见 管理账号 中的修改权限部分，可调整全局权限和对象级权限。

运维管理

最近更新时间：2022-07-19 18:04:12

为何新建的库表（新写入的数据）不见了?

初次使用 TDSQL-H LibraDB 的用户，常反馈库表或是数据丢失。该问题存在于多副本或多分片实例，这是由于没有正确的创建库表。创建库表需要遵循以下原则：

- 必须使用 Replicated 引擎族，即您选择的引擎，需要支持副本（请注意这里不是指 Replacing 引擎族）。
- 必须使用 Distributed 表做数据操作。同时，创建表语句时，请通过以下任一方式保证 Distributed 表在集群纬度创建。
  - 使用 `ON CLUSTER default_cluster` 语法。
  - 使用 TDSQL-H LibraDB 内核的优化功能，即 LibraSQL 提供默认 DDL 集群配置：`ddl_default_oncluster`，自动在集群纬度执行DDL。具体说明请参见 支持 DDL 语句默认的 on cluster。

这是因为，您所连上的网络地址，是按照负载均衡原则，随机接入集群任意节点的。如果只使用不带副本的本地表，当下一个会话被派发到其他节点时，是无法获取到上一个节点的库表。
为方便用户使用 LibraSQL 分析引擎，提供操作 Demo 如下：

```
# 建立 online_smoke Database
CREATE DATABASE online_smoke ON CLUSTER default_cluster;

# 创建 ReplicatedMergeTree 本地表
create table online_smoke.online_smoke_local on cluster default_cluster ( i Int32, d
Date ) ENGINE = ReplicatedMergeTree() PARTITION BY toYYYYMM(d) order by  (i, d);

# 创建本地表 online_smoke_local的 Distributed 表
CREATE TABLE online_smoke.online_smoke_global on cluster default_cluster
as`online_smoke`. `online_smoke_local` ENGINE = Distributed(default_cluster,
online_smoke, online_smoke_local, i);

# 插入数据
insert into online_smoke.online_smoke_global (i, d) values(1, '2020-01-01');
insert into online_smoke.online_smoke_local (i, d) values(1, '2020-01-02');

# 获取数据
select * from online_smoke.online_smoke_global;
select * from online_smoke.online_smoke_local;

# 删除分布式表
drop table  online_smoke.online_smoke_global on cluster default_cluster;
```

```
# 删除本地表
drop table online_smoke.online_smoke_local  on cluster default_cluster;
```

如何查询当前正在执行的查询?

通过 system.processes 可以查询当前正在执行的查询。
示例如下:

```
SELECT
    query_id,
    user,
    address,
    query
FROM system.processes
ORDER BY query_id ASC

Query id: a557f763-cf83-4cc1-afd8-148c5263cefb

┌─query_id─────────────────────────────┬─user─┬─address──────────────┬─q
uery─────────────────────────────────────────────────────────────────────
──────────────────────┐
│ a557f763-cf83-4cc1-afd8-148c5263cefb │ meral │ ::ffff:172.xx.xx.27 │ SELECT
    query_id,
    user,
    address,
    query
FROM system.processes
ORDER BY query_id ASC │
└──────────────────────────────────────┴───────┴──────────────────────┴─
──────────────────────────────────────────────────────────────────────
──────────────────────┘

1 rows in set. Elapsed: 0.004 sec.
```

如何终止正在执行的查询?

通过 KILL QUERY 可以终止正在执行的查询。
示例如下:

```
KILL QUERY WHERE query_id = 'a557f763-cf83-4cc1-afd8-148c5263cefb';

KILL QUERY WHERE query_id = 'a557f763-cf83-4cc1-afd8-148c5263cefb' ASYNC

Query id: e22b1074-b7ad-490e-95ce-b89a369fb226
```

```
Ok.

0 rows in set. Elapsed: 0.004 sec.
```

如何查询 update、delete 操作?

通过 `system.mutations` 可以查询 update、delete 操作。
示例如下:

```
SELECT
    database,
    table,
    mutation_id,
    command,
    create_time,
    is_done
FROM system.mutations

Query id: 48e4faff-5286-4fac-be06-b8f460676baf

┌─database─┬─table────┬─mutation_id─┬─command─────────────────────────────────────────────────────────────
─────────────────────────────────────────────────────────────────────────────────
─────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────────────┬───────────create_time─┬─
is_done─┐
│ zm2      │ t5_local │ 0000000000  │ LIGHT WEIGHT DELETE WHERE (_dversion = 0)
AND (_version < 416) AND (id IN (SELECT id FROM file('c9d7c5b4-17f8-4287-b1d0-
6676405b8694_d5dee8b2-d548-4b9a-bf6b-ed6c40f742a8', 'CSV', 'id Int32') SETTINGS
format_csv_delimiter = '')) │ 2022-06-24 14:50:00 │       1 │
└──────────┴──────────┴─────────────┴───────────────────────────────────────────────────────────────
─────────────────────────────────────────────────────────────────────────────────
─────────────────────────────────────────────────────────────────────────────────
───────────────────────────────────────────────────────┴─────────────────────────
┴─────────┘

1 rows in set. Elapsed: 0.003 sec.
```

如何终止 update、delete 操作?

通过 `KILL MUTATION` 可以终止 update、delete 操作。
示例如下:

```
KILL MUTATION WHERE mutation_id = '0000000000';
```

```
KILL MUTATION WHERE mutation_id = '0000000000' ASYNC

Query id: d815f1d7-c62b-4dec-a363-ef01a288703f

Ok.

0 rows in set. Elapsed: 0.003 sec.
```

如何查询存储空间?

通过 `system.disks` 查询各存储路径空间。
示例如下:

```
SELECT
    name,
    path,
    formatReadableSize(free_space) AS free,
    formatReadableSize(total_space) AS total,
    formatReadableSize(keep_free_space) AS reserved
FROM system.disks

Query id: 8924a6d2-aaf1-430e-a6f1-6936fd1c44ac

┌─name────┬─path────────────────┬─free─────┬─total────┬─reserved─┐
│ default │ /data/clickhouse/ │ 116.22 GiB │ 117.56 GiB │ 0.00 B   │
└─────────┴─────────────────────┴────────────┴────────────┴──────────┘

1 rows in set. Elapsed: 0.003 sec.
```

如何查询各数据库已占用存储空间?

通过 `system.parts` 查询。
示例如下:

```
SELECT
    database,
    formatReadableSize(sum(bytes_on_disk)) AS on_disk
FROM system.parts
GROUP BY database

Query id: 196085d3-4b01-46e1-9733-2f81b9c82460

┌─database────────┬─on_disk────┐
```

```
| zm2          | 497.00 B   |
| system       | 107.97 MiB |
| __tencentdb__ | 366.84 KiB |
└──────────────┴────────────┘

3 rows in set. Elapsed: 0.003 sec.
```

如何查询各列字段占用空间统计数据?

通过 system.parts_columns 查询。
示例如下:

```
SELECT
    database,
    table,
    column,
    any(type),
    sum(column_data_compressed_bytes) AS compressed,
    sum(column_data_uncompressed_bytes) AS uncompressed,
    round(uncompressed / compressed, 2) AS ratio,
    compressed / sum(rows) AS bpr,
    sum(rows)
FROM system.parts_columns
WHERE active AND (database != 'system')
GROUP BY
    database,
    table,
    column
ORDER BY
    database ASC,
    table ASC,
    column ASC

┌─database─┬─table──────────┬─column──┬─any(type)─┬─compressed─┬─uncompressed─┬─ratio─┬──────────────bpr─┬─sum(rows)─┐
| ch_test_1 | table_1_local | _sign | Int8 | 39 | 976 | 25.03 | 0.039959016393442626 | 976 |
| ch_test_1 | table_1_local | _version | UInt64 | 70 | 7808 | 111.54 | 0.07172131147540983 | 976 |
| ch_test_1 | table_1_local | cint32 | Int32 | 67 | 3904 | 58.27 | 0.06864754098360656 | 976 |
| ch_test_1 | table_1_local | float32 | Float32 | 61 | 3904 | 64 | 0.0625 | 976 |
```

```
| ch_test_1     | table_1_local       | id          | Int64          |      3934 |       7808 |
1.98 |    4.030737704918033 |      976 |
| ch_test_1     | table_1_local       | interface   | String         |        46 |       1952 |
42.43 |   0.0471311475409836 |      976 |
| ch_test_1     | table_1_local       | status      | Int8           |        40 |        976 |
24.4 | 0.040983606557377046 |      976 |
| ch_test_1     | table_1_local       | uin         | Int64          |        74 |       7808 |
105.51 |  0.07581967213114754 |      976 |
| ch_test_1     | table_1_local       | updateTime  | String         |      4899 |
19520 |   3.98 |    5.019467213114754 |      976 |
| db_d10_0221   | tab_t10_local       | _sign       | Int8           |        30 |          4 |
0.13 |            7.5 |        4 |
| db_d10_0221   | tab_t10_local       | _version    | UInt64         |        40 |
32 |    0.8 |             10 |        4 |
| db_d10_0221   | tab_t10_local       | c1          | Int32          |        43 |         16 |
0.37 |           10.75 |        4 |
| db_d10_0221   | tab_t10_local       | c2          | Int32          |        43 |         16 |
0.37 |           10.75 |        4 |
| db_d10_0221   | tab_t10_local       | c4          | Nullable(String) |        86 |
84 |   0.98 |           21.5 |        4 |
| db_d10_0221   | tab_t10_local       | c5          | Nullable(String) |        76 |
48 |   0.63 |             19 |        4 |
| db_d10_0221   | tab_t10_local       | c6          | Nullable(String) |        86 |
84 |   0.98 |           21.5 |        4 |
| db_d1_0221    | tab_t1_local        | _sign       | Int8           |        30 |          4 |
0.13 |            7.5 |        4 |
| db_d1_0221    | tab_t1_local        | _version    | UInt64         |        40 |         32
|    0.8 |             10 |        4 |
| db_d1_0221    | tab_t1_local        | c1          | Int32          |        43 |         16 |
0.37 |           10.75 |        4 |
| db_d1_0221    | tab_t1_local        | c2          | Int32          |        43 |         16 |
0.37 |           10.75 |        4 |
| db_d1_0221_02 | tab_t1_local        | _sign       | Int8           |        30 |          4 |
0.13 |            7.5 |        4 |
| db_d1_0221_02 | tab_t1_local        | _version    | UInt64         |        40 |
32 |    0.8 |             10 |        4 |
| db_d1_0221_02 | tab_t1_local        | c1          | Int32          |        43 |         16 |
0.37 |           10.75 |        4 |
| db_d1_0221_02 | tab_t1_local        | c2          | Int32          |        43 |         16 |
0.37 |           10.75 |        4 |
| test1_0222_01 | bu_bidding_ent_local | _sign       | Int8           |        27 |          1
|   0.04 |             27 |        1 |
| test1_0222_01 | bu_bidding_ent_local | _version    | UInt64         |        34 |
8 |   0.24 |             34 |        1 |
```

```
| test1_0222_01 | bu_bidding_ent_local | bid_type     | Int32          |       30 |
4 |   0.13 |            30 |       1 |
| test1_0222_01 | bu_bidding_ent_local | contact      | Nullable(String) |       54 |
2 |   0.04 |            54 |       1 |
| test1_0222_01 | bu_bidding_ent_local | createdDate  | Nullable(Int64)  |       61 |
9 |   0.15 |            61 |       1 |
| test1_0222_01 | bu_bidding_ent_local | email        | Nullable(String) |       54 |
2 |   0.04 |            54 |       1 |
| test1_0222_01 | bu_bidding_ent_local | ent_name     | String         |       28 |
2 |   0.07 |            28 |       1 |
| test1_0222_01 | bu_bidding_ent_local | fax          | Nullable(String) |       54 |
2 |   0.04 |            54 |       1 |
| test1_0222_01 | bu_bidding_ent_local | id           | String         |       28 |        2
|   0.07 |            28 |       1 |
| test1_0222_01 | bu_bidding_ent_local | isDeleted    | Nullable(Int8)   |       54 |
2 |   0.04 |            54 |       1 |
| test1_0222_01 | bu_bidding_ent_local | mobile       | Nullable(String) |       54 |
2 |   0.04 |            54 |       1 |
| test1_0222_01 | bu_bidding_ent_local | modifiedDate | Nullable(Int64)  |       61 |
9 |   0.15 |            61 |       1 |
| test1_0222_01 | bu_bidding_ent_local | pid          | String         |       28 |        2
|   0.07 |            28 |       1 |
| test1_0222_01 | bu_bidding_ent_local | project_id   | String         |       28 |
2 |   0.07 |            28 |       1 |
| test1_0222_01 | bu_bidding_ent_local | pub_date     | Int64          |       34 |
8 |   0.24 |            34 |       1 |
| test1_0222_01 | bu_bidding_ent_local | tel          | Nullable(String) |       54 |
2 |   0.04 |            54 |       1 |
| test1_0222_01 | bu_bidding_ent_local | type_desc    | Nullable(String) |       54 |
2 |   0.04 |            54 |       1 |
| test1_0222_01 | bu_bidding_ent_local | updatedDate  | Nullable(Int64)  |       61 |
9 |   0.15 |            61 |       1 |
| test1_0222_01 | t1_local             | _sign        | Int8           |       36 |       64 |
1.78 |         0.5625 |      64 |
| test1_0222_01 | t1_local             | _version     | UInt64         |       41 |      512
|  12.49 |        0.640625 |      64 |
| test1_0222_01 | t1_local             | c1           | Int32          |      283 |      256 |
0.9 |        4.421875 |      64 |
└───────────────┴──────────────────────┴──────────────┴────────────────┴──────────┴──────────
──┴──────────────┴─────────┴──────────────────────────┴──────
──────┘

45 rows in set. Elapsed: 0.154 sec.
```

如何查询慢查询?

通过 system.query_log 查询。
示例如下：

```
SELECT
    user,
    client_hostname AS host,
    client_name AS client,
    formatDateTime(query_start_time, '%T') AS started,
    query_duration_ms / 1000 AS sec,
    round(memory_usage / 1048576) AS MEM_MB,
    result_rows AS RES_CNT,
    result_bytes / 1048576 AS RES_MB,
    read_rows AS R_CNT,
    round(read_bytes / 1048576) AS R_MB,
    written_rows AS W_CNT,
    round(written_bytes / 1048576) AS W_MB,
    query
FROM system.query_log
WHERE type = 2
ORDER BY query_duration_ms DESC
LIMIT 10

Query id: de49a470-2b40-46d0-b238-31b3927ebe08

Row 1:
──────
user:    tencentroot_auto
host:    VM_46_163_centos
client:  Golang SQLDriver
started: 14:44:45
sec:     0.128
MEM_MB:  0
RES_CNT: 2
RES_MB:  0.000118255615234375
R_CNT:   2
R_MB:    0
W_CNT:   0
W_MB:    0
query:   CREATE DATABASE IF NOT EXISTS __tencentdb__ ON CLUSTER default_cluster

......
```

如何对副本进行预警监控?

通过以下 SQL 语句对副本进行预警监控，其中各个预警的变量可以根据具体情况调整。

```
SELECT
    database,
    table,
    is_leader,
    total_replicas,
    active_replicas
FROM system.replicas
WHERE is_readonly OR is_session_expired OR (future_parts > 30) OR (parts_to_check
> 20) OR (queue_size > 30) OR (inserts_in_queue > 20) OR ((log_max_index -
log_pointer) > 20) OR (total_replicas < 2) OR (active_replicas < total_replicas)
```