

TDSQL-H LibraDB

内核能力介绍



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

内核能力介绍

分析引擎功能

轻量级删除

全局系统表

连接管理

支持 DDL 语句默认的 on cluster

CDC 特性

事务最终一致性

无 FINAL 一致性查询

内核能力介绍

分析引擎功能

轻量级删除

最近更新时间：2024-10-17 20:41:22

特性描述

原生 ClickHouse 的 MergeTree 系列引擎数据删除操作需要重建底层数据，操作执行代价高，无法在生产环境频繁使用。

LibraDB 提供了轻量级的标记删除，无需重建底层数据，查询时根据删除的 bitmap 快速过滤删除数据，在基本不影响查询效率的情况下，大量降低了删除操作的执行代价。

轻量级删除语句：

```
ALTER TABLE [db.]table LIGHT WEIGHT DELETE WHERE filter_expr;
```

使用限制

需要使用轻量级删除特性的 MergeTree 系列引擎在建表时需要增加 `allow_fast_delete_columnn = 1` 参数（默认 CDC 同步的表已经添加此参数）。

使用说明

以下将构造数据测试轻量级删除和原生 ClickHouse `DELETE` 的删除性能。

创建库表

1. 创建数据库。

```
c [1 :]) DROP DATABASE IF EXISTS tencent_testzm3;

DROP DATABASE IF EXISTS tencent_testzm3

Query id: cc4f2285-125c-41c2-8e6e-b17907cf8607

Ok.

0 rows in set. Elapsed: 0.002 sec.

c [ :]) CREATE DATABASE tencent_testzm3;

CREATE DATABASE tencent_testzm3

Query id: 22eb8263-3a75-45ab-99ae-27df2daa08ca

Ok.
```

2. 创建表 t_1 时增加 `allow_fast_delete_column = 1` 参数。

```
c [ :]) CREATE TABLE tencent_testzm3.t_1
:-] (
:-]   `order_0` UInt64,
:-]   `ordinary_1` UInt32,
:-]   `ordinary_2` String,
:-]   `ordinary_3` UInt32,
:-]   `p_time` Date
:-] )
:-] ENGINE = MergeTree
:-] PARTITION BY toYYYYMM(p_time)
:-] ORDER BY order_0
:-] SETTINGS allow_fast_delete_column = 1;

CREATE TABLE tencent_testzm3.t_1
(
  `order_0` UInt64,
  `ordinary_1` UInt32,
  `ordinary_2` String,
  `ordinary_3` UInt32,
  `p_time` Date
)
ENGINE = MergeTree
PARTITION BY toYYYYMM(p_time)
ORDER BY order_0
SETTINGS allow_fast_delete_column = 1

Query id: 1d96c565-387a-4ae0-9d40-c45b122d5e0c

Ok.

0 rows in set. Elapsed: 0.006 sec.
```

3. 创建随机表。

```
c                                     :) CREATE TABLE tencent_testzm3.t_random_1
:-] (
:-]   `ordinary_1` UInt32,
:-]   `ordinary_2` String,
:-]   `ordinary_3` UInt32
:-] )
:-] ENGINE = GenerateRandom(1, 5, 3);

CREATE TABLE tencent_testzm3.t_random_1
(
  `ordinary_1` UInt32,
  `ordinary_2` String,
  `ordinary_3` UInt32
)
ENGINE = GenerateRandom(1, 5, 3)

Query id: 382b14dd-7dd6-4fbe-9d97-e3bc9597cdab

Ok.

0 rows in set. Elapsed: 0.005 sec.
```

4. 创建表 t_2 时不增加 `allow_fast_delete_column = 1` 参数。

```
c                                     :) CREATE TABLE tencent_testzm3.t_2
:-] (
:-]   `order_0` UInt64,
:-]   `ordinary_1` UInt32,
:-]   `ordinary_2` String,
:-]   `ordinary_3` UInt32,
:-]   `p_time` Date,
:-]   `sign` Int8,
:-]   `ver` UInt64
:-] )
:-] ENGINE = ReplacingMergeTree(ver)
:-] PARTITION BY toYYYYMM(p_time)
:-] ORDER BY order_0
:-] SETTINGS index_granularity = 8192;

CREATE TABLE tencent_testzm3.t_2
(
  `order_0` UInt64,
  `ordinary_1` UInt32,
  `ordinary_2` String,
  `ordinary_3` UInt32,
  `p_time` Date,
  `sign` Int8,
  `ver` UInt64
)
ENGINE = ReplacingMergeTree(ver)
PARTITION BY toYYYYMM(p_time)
ORDER BY order_0
SETTINGS index_granularity = 8192

Query id: 65ce31a8-3d57-4b4b-949e-82476a09c267

Ok.

0 rows in set. Elapsed: 0.007 sec.
```

5. 分别将随机表中的数据写入 t_1 和 t_2 中。

```
c : insert into tentcent_testzm3.t_1 select rowNumberInAllBlocks(),*,NOW() from tentcent_testzm2.t_random_1 limit 50000000;

INSERT INTO tentcent_testzm3.t_1 SELECT
  rowNumberInAllBlocks(),
  *,
  NOW()
FROM tentcent_testzm2.t_random_1
LIMIT 50000000

Query id: d9928f65-590d-40a3-86ec-4682984f6e6d

Ok.

0 rows in set. Elapsed: 5.582 sec. Processed 50.33 million rows, 981.44 MB (9.02 million rows/s., 175.81 MB/s.)

c : insert into tentcent_testzm3.t_2 select rowNumberInAllBlocks(),*,NOW(),1,1 from tentcent_testzm2.t_random_1 limit 50000000;

INSERT INTO tentcent_testzm3.t_2 SELECT
  rowNumberInAllBlocks(),
  *,
  NOW(),
  1,
  1
FROM tentcent_testzm2.t_random_1
LIMIT 50000000

Query id: 4dc31cbb-2144-4090-ba45-27389cc13974

Ok.
```

删除数据

- **轻量级删除**：删除 t_1 中数据。

```
ALTER TABLE [db.]table LIGHT WEIGHT DELETE WHERE filter_expr;
```

示例：

```
c : alter table tentcent_testzm3.t_1 LIGHT WEIGHT DELETE where order_0 < 10000;

ALTER TABLE tentcent_testzm3.t_1
  LIGHT WEIGHT DELETE WHERE order_0 < 10000

Query id: 6ea85069-5b29-4034-bc77-1426b061dfe8

Ok.

0 rows in set. Elapsed: 0.296 sec.

c : select count(order_0) from tentcent_testzm3.t_1;

SELECT count(order_0)
FROM tentcent_testzm3.t_1

Query id: e11a7b03-df5b-48f5-ad03-58234de21110

+-----+
|count(order_0)|
+-----+
|         49990000|
+-----+

1 rows in set. Elapsed: 0.164 sec. Processed 50.00 million rows, 800.00 MB (304.12 million rows/s., 4.87 GB/s.)
```

- **原生 Clickhouse 删除**：删除 t_2 中数据。

```
ALTER TABLE [db.]table DELETE WHERE filter_expr;
```

示例：

```
c . :) alter table tencent_testzm3.t_2 DELETE where order_0 < 10000;

ALTER TABLE tencent_testzm3.t_2
  DELETE WHERE order_0 < 10000

Query id: b77e0c4a-fc13-489d-aeef-66a4e0d29b78

Ok.

0 rows in set. Elapsed: 2.473 sec.

c . :) select count(order_0) from tencent_testzm3.t_2;

SELECT count(order_0)
FROM tencent_testzm3.t_2

Query id: beab4715-32e3-446d-816b-3338382ed6ba

+-----+
| count(order_0) |
+-----+
| 49990000       |
+-----+

1 rows in set. Elapsed: 0.132 sec. Processed 49.99 million rows, 399.92 MB (377.75 million rows/s., 3.02 GB/s.)
```

测试结果：轻量级删除耗时0.296sec，原生 ClickHouse `DELETE` 语句删除耗时2.473sec，整体性能提升8倍以上。

全局系统表

最近更新时间：2024-10-17 20:41:22

特性描述

原生 ClickHouse 的 `system database` 中包含大量的审计、日志和元数据信息。对于多节点实例，这些信息在每个节点中各自单独存储着，查询这些信息需要人工登录到每个节点中分别进行，对于日常运维和故障排除效率较低。

因此 LibraDB 在一些关键的系统表上建立基于集群的全局系统表，可在单节点上进行全局信息的查询，同时在这些全局系统表中加入名为 `_host` 的虚拟列，可以快速分辨出不同实例的信息。

在 `system database` 中，全局系统表以关联的本地表 + `_all` 来进行命名。

注意事项

全局系统表可以使用 `_host` 虚拟列获取信息对应的节点。

使用说明

1. 连接到 LibraDB 分析引擎。
2. 查看 `system.tables` 中内置的全局系统表。

```
SELECT database,name FROM system.tables WHERE (database = 'system')
AND (name LIKE '%_all');
```

示例：

```
c:\> SELECT database,name FROM system.tables WHERE (database = 'system') AND (name LIKE '%_all');

SELECT
  database,
  name
FROM system.tables
WHERE (database = 'system') AND (name LIKE '%_all')

Query id: 34c732f8-5bea-42b5-bc28-ffec04c3ad5b

+-----+-----+
| database | name |
+-----+-----+
| system   | columns_all |
| system   | connections_all |
| system   | current_roles_all |
| system   | databases_all |
| system   | detached_parts_all |
| system   | disks_all |
| system   | distributed_ddl_queue_all |
| system   | distribution_queue_all |
| system   | errors_all |
| system   | events_all |
| system   | grants_all |
| system   | macros_all |
| system   | merges_all |
| system   | metrics_all |
| system   | mutations_all |
| system   | part_log_all |
| system   | parts_all |
| system   | parts_columns_all |
| system   | query_log_all |
| system   | query_thread_log_all |
| system   | quotas_all |
| system   | replicas_all |
| system   | replication_queue_all |
| system   | role_grants_all |
| system   | roles_all |
| system   | row_policies_all |
| system   | server_dynamic_configurations_all |
| system   | settings_profiles_all |
| system   | tables_all |
| system   | trace_log_all |
| system   | users_all |
+-----+-----+

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

3. 在具体的全局系统表中使用与查询本地表相同的方式查询。

使用 `SELECT * FROM ...` 从全局系统表中查询数据时，是不带 `_host` 列的。如果需要使用

`SELECT ... FROM ...` 查询表中列并带上 `_host` 列，需要显示添加 `_host` 列。

- 添加 `_host` 列

示例：

```
SELECT _host,query,query_id FROM system.query_log_all;
```

```
c:\> SELECT SUM(data_uncompressed_bytes) AS bytes_inserted, COUNT() AS active_data_parts, SUM(rows) AS rows_inserted FROM system.parts WHERE active = 1
b1865c0a-50e4-4cf4-a79e-5ec6c22e11d8

c:\> SELECT SUM(data_uncompressed_bytes) AS bytes_inserted, COUNT() AS active_data_parts, SUM(rows) AS rows_inserted FROM system.parts WHERE active = 1
b1865c0a-50e4-4cf4-a79e-5ec6c22e11d8
```

- 不添加 `_host` 列

示例：

```
SELECT query,query_id FROM system.query_log_all;
```

```
| SELECT SUM(data_uncompressed_bytes) AS bytes_inserted, COUNT() AS active_data_parts, SUM(rows) AS rows_inserted FROM system.parts WHERE active = 1  
| 8a89d70f-372c-4cff-b1ea-2ba6f0c76adf |  
| SELECT SUM(data_uncompressed_bytes) AS bytes_inserted, COUNT() AS active_data_parts, SUM(rows) AS rows_inserted FROM system.parts WHERE active = 1  
| 9ccd65c6-2050-41ec-af34-a95d5d782323 |
```

连接管理

最近更新时间：2024-10-17 20:41:22

特性描述

在实例连接数过多等运维场景下，我们需要查看连接并删除连接。TDSQL-H LibraDB 支持查看和删除连接。

- 查看当前节点连接：`show connections` 或访问 `system.connections` 表。
- 删除当前节点连接：

```
kill connection where ... (sync);
```

- 查看全局连接：`show connections global` 或访问 `system.connections_all` 表。
- 删除全局连接：

```
kill connection where ... (sync) global;
```

使用限制

如果需要清除连接，登录账号需具有 KILL QUERY 权限。

使用说明

查看当前节点连接

- `show connections`

示例：

```
c [root@localhost ~]# show connections;
SHOW CONNECTIONS
Query id: 5d38da83-0d35-4580-8273-2ec7ba20400d
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | query_id | state | query | connection_type | user | host | db | time | idle |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 5d38da83-0d35-4580-8273-2ec7ba20400d | 5d38da83-0d35-4580-8273-2ec7ba20400d | Sleep | show connections; | TCP | tencentroot_manual | 192.168.1.103 | default | 1446 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set. Elapsed: 0.003 sec.
```

- 访问 `system.connections` 表

```
SELECT * FROM system.connections ...;
```

示例:

```
c [redacted] ; :) SELECT * FROM system.connections where state = 'Sleep';

SELECT *
FROM system.connections
WHERE state = 'Sleep'

Query id: 0310c4da-2bad-4888-b25a-1de430b3cbdc

+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | state | query | connection_type | user | host | db | time | idle |
+-----+-----+-----+-----+-----+-----+-----+-----+
| c | Sleep | | n.svc.cluster.local_2 | TCP | tencentroot_inner | default | 1721005 | 166 | |
| c | Sleep | | n.svc.cluster.local_42993 | TCP | tencentroot_auto | 061 | default | 46 | 16 |
| c | Sleep | | n.svc.cluster.local_42994 | TCP | tencentroot_auto | 085 | default | 46 | 6 |
+-----+-----+-----+-----+-----+-----+-----+-----+

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

删除当前节点连接

- 同步删除

```
kill connection where connection_id=' ' sync;
```

示例:

```
[redacted] :) show connections;

SHOW CONNECTIONS

Query id: dfb6455c-25ad-4b9f-b301-eceb5ca05eb

+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | state | query | connection_type | user | host | db | time | idle |
+-----+-----+-----+-----+-----+-----+-----+-----+
| dfb6455c-25ad-4b9f-b301-eceb5ca05eb | Sleep | | n.svc.cluster.local_2 | TCP | tencentroot_inner | 4 | default | 1722539 | 138 |
| dfb6455c-25ad-4b9f-b301-eceb5ca05eb | Sleep | | n.svc.cluster.local_42988 | TCP | tencentroot_manual | | default | 1720 | 0 |
| dfb6455c-25ad-4b9f-b301-eceb5ca05eb | Query | show connections; | n.svc.cluster.local_43032 | TCP | tencentroot_auto | 821 | default | 18 | 18 |
| dfb6455c-25ad-4b9f-b301-eceb5ca05eb | Sleep | | n.svc.cluster.local_43033 | TCP | tencentroot_auto | 278 | default | 18 | 8 |
+-----+-----+-----+-----+-----+-----+-----+-----+

4 rows in set. Elapsed: 0.003 sec.

[redacted] :) kill connection where connection_id='n.svc.cluster.local_43033' sync;

KILL CONNECTION WHERE connection_id = 'n.svc.cluster.local_43033' SYNC

Query id: 5ffab545-0046-42d4-bf2f-0131ac20c2a2

+-----+-----+-----+-----+-----+-----+
| connection_status | connection_id | user | query_id | query | query_status |
+-----+-----+-----+-----+-----+-----+
| killed | n.svc.cluster.local_43033 | tencentroot_auto | | | |
+-----+-----+-----+-----+-----+-----+

1 rows in set. Elapsed: 5.004 sec.
```

- 异步删除

```
kill connection where connection_id=' ';
```

示例：

```

:~) show connections;

SHOW CONNECTIONS
Query id: 521d7a44-8157-479b-8414-e796b538c235

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | state | query | connection_type | user | host | db | time | idle |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 521d7a44-8157-479b-8414-e796b538c235 | Sleep | show connections; | TCP | tencentroot_inner | 192.168.1.2 | 4 | default | 1722612 | 211 |
| 521d7a44-8157-479b-8414-e796b538c235 | Sleep | show connections; | TCP | tencentroot_manual | 192.168.1.2 | 4 | default | 1793 | 0 |
| 521d7a44-8157-479b-8414-e796b538c235 | Sleep | show connections; | TCP | tencentroot_auto | 192.168.1.2 | 821 | default | 91 | 61 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

3 rows in set. Elapsed: 0.002 sec.

:~) kill connection where connection_id='n.svc.cluster.local_43032';

KILL CONNECTION WHERE connection_id = 'clickhouse-0.khaos.tdach-ip75is8n.svc.cluster.local_43032' ASYNC
Query id: f8137949-f0b4-4f8c-9868-a644e10efc04

+-----+-----+-----+-----+-----+-----+
| connection_status | connection_id | user | query_id | query | query_status |
+-----+-----+-----+-----+-----+-----+
| waiting | n.svc.cluster.local_43032 | tencentroot_auto | | | |
+-----+-----+-----+-----+-----+-----+

1 rows in set. Elapsed: 0.003 sec.

```

查看连接已删除成功。

```

:~) show connections;

SHOW CONNECTIONS
Query id: 958f1580-875e-4c67-a3a6-4d0a92947da9

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | state | query | connection_type | user | host | db | time | idle |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 958f1580-875e-4c67-a3a6-4d0a92947da9 | Sleep | show connections; | TCP | tencentroot_inner | 192.168.1.2 | 34 | default | 1722655 | 254 |
| 958f1580-875e-4c67-a3a6-4d0a92947da9 | Sleep | show connections; | TCP | tencentroot_manual | 192.168.1.2 | 3 | default | 1836 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

2 rows in set. Elapsed: 0.003 sec.

```

查看全局连接

- `show connections global`

示例：

```

c:~) show connections global;

SHOW CONNECTIONS GLOBAL
Query id: 8428a475-b26d-437a-b0c4-703e3bf09e8d

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | connection_type | user | host | db | time | idle | query_id | state | query |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| c | TCP | tencentroot_manual | 192.168.1.2 | 4 | default | 0 | 8428a475-b26d-437a-b0c4-703e3bf09e8d | Query | show connections global; |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| c | TCP | tencentroot_inner | 192.168.1.2 | 2 | default | 0 | f80d4024-d210-469a-8ca6-4a779b05b7f4 | Query | SELECT connections.connection_id, connections.connect |
| ion_type, connections.user, connections.host, connections.db, connections.time, connections.idle, connections.query_id, connections.state, connections.query FROM system.connections |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

2 rows in set. Elapsed: 0.006 sec.

```

- 访问 `system.connections_all` 表

```
SELECT * FROM system.connections_all;
```

示例：

```
c:) select * from system.connections_all;

SELECT *
FROM system.connections_all

Query id: 6fdbabe4-beb6-45e1-9122-39b7acd8ce3e

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | .svc.cluster.local_64 | TCP | tencentroot_manual | 9 | 0 | default | 737 | 0 | 6fdbabe4-beb6-45e1-9122-39b7acd8ce3e | Sleep | select * from system.connections_all; |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | connection_type | user | host | db | time | idle | query_id | state | query |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| c | .svc.cluster.local_5 | TCP | tencentroot_inner | 9 | 2 | default | 655 | 0 | 76c03a41-e3c0-43d3-9793-02136facbeb3 | Query | SELECT connections.connection_id, connections.connect |
| ion_type, connections.user, connections.host, connections.db, connections.time, connections.idle, connections.query_id, connections.state, connections.query FROM system.connections |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|
```

删除全局连接

- 同步删除

```
kill connection where connection_id=' ' sync global;
```

示例：

```
c:) show connections global;

SHOW CONNECTIONS GLOBAL

Query id: 967df9e4-bd11-48d2-a7ab-daec67190097

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | .local_68 | TCP | tencentroot_manual | 9 | 12 | default | 1376 | 663 | 967df9e4-bd11-48d2-a7ab-daec67190097 | Sleep | show connections global; |
| c | .local_70 | TCP | tencentroot_manual | 9 | 0 | default | 25 | 0 | | Query | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | connection_type | user | host | db | time | idle | query_id | state | query |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| c | .local_6 | TCP | tencentroot_inner | 9 | 10 | default | 1376 | 0 | a15f05c5-8613-4911-949a-d25512ad8b14 | Query | SELECT connections.connection_id, connections.connect |
| ion_type, connections.user, connections.host, connections.db, connections.time, connections.idle, connections.query_id, connections.state, connections.query FROM system.connections |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|

3 rows in set. Elapsed: 0.005 sec.

clickhouse-0.khaos.tdach-515s5ges.svc.cluster.local :) kill connection where connection_id='c' .local_6' sync global;

KILL CONNECTION ON CLUSTER all_shard_cluster WHERE connection_id = 'c' .local_6' SYNC

Query id: a3872454-dadb-4a3a-9ece-8a7d47de66a1

+-----+-----+-----+-----+-----+-----+-----+-----+
| host | port | status | error | num_hosts_remaining | num_hosts_active |
+-----+-----+-----+-----+-----+-----+-----+-----+
| c | s.t | .svc.cluster.local | 9000 | 0 | 1 | 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| host | port | status | error | num_hosts_remaining | num_hosts_active |
+-----+-----+-----+-----+-----+-----+-----+-----+
| c | s.t | .svc.cluster.local | 9000 | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+

2 rows in set. Elapsed: 6.132 sec.

c:) show connections global;

SHOW CONNECTIONS GLOBAL

Query id: fb948c7b-4cba-4c02-8d23-cc85134e2c6a

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | .local_68 | TCP | tencentroot_manual | 9 | 2 | default | 1417 | 705 | fb948c7b-4cba-4c02-8d23-cc85134e2c6a | Sleep | show connections global; |
| c | .local_70 | TCP | tencentroot_manual | 9 | 0 | default | 66 | 0 | | Query | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|

/Progress: 0.00 rows, 0.00 B (0.00 rows/s., 0.00 B/s.)
2 rows in set. Elapsed: 0.003 sec.

Received exception from server (version 21.3.12):
Code: 2006. DB::Exception: Received from 9. . DB::Exception: Received from c1 .svc.cluster.local:9000. DB::Exception: Connection killed by others.: While executing Remote.
```

- 异步删除

```
kill connection where connection_id=' ' global;
```

示例：

```
c1 : show connections global;

SHOW CONNECTIONS GLOBAL

Query id: 63651a4b-a9fa-44cd-b465-06a59b6e7768

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | connection_type | user           | host | db | time | idle | query_id | state | query |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| c              | .local_68       | TCP           | 9     | 2 | default | 2028 | 1315 | Sleep | show connections global; |
| c              | .local_72       | TCP           | 9     | 2 | default | 37    | 0    | Query | show connections global; |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | connection_type | user           | host | db | time | idle | query_id | state | query |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| c              | .local_7        | TCP           | 9     | 10 | default | 29    | 0    | Query | SELECT connections.connection_id, connections.connect |
| ion_type, connections.user, connections.host, connections.db, connections.time, connections.idle, connections.query_id, connections.state, connections.query FROM system.connections |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

3 rows in set. Elapsed: 0.005 sec.

c : kill connection where connection_id='s.svc.cluster.local_7' global;

KILL CONNECTION ON CLUSTER all_shard_cluster WHERE connection_id = 's.svc.cluster.local_7' ASYNC

Query id: 0f7de18d-7e52-467a-a8cd-66b3d65dc9f

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| host | port | status | error | num_hosts_remaining | num_hosts_active |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| c     | 9000 | s.t    | 0     | 1                   | 0                 |
| c     | 9000 | s.t    | 0     | 0                   | 0                 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

2 rows in set. Elapsed: 0.119 sec.

c : show connections global;

SHOW CONNECTIONS GLOBAL

Query id: 9f1baff3-3481-4082-aade-c665ac7c7060

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| connection_id | .local_68       | connection_type | user           | host | db | time | idle | query_id | state | query |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| c              | .local_72       | TCP           | tencentroot_manual | 9     | 2 | default | 65    | 0    | 9f1baff3-3481-4082-aade-c665ac7c7060 | Sleep | show connections global; |
| c              | .local_72       | TCP           | tencentroot_manual | 9     | 2 | default | 65    | 0    | 9f1baff3-3481-4082-aade-c665ac7c7060 | Query | show connections global; |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

Progress: 0.00 rows, 0.00 B (0.00 rows/s., 0.00 B/s.)
2 rows in set. Elapsed: 0.004 sec.

Received exception from server (version 21.3.12):
Code: 2006. DB::Exception: Received from s.svc.cluster.local:9000. DB::Exception: Connection killed by others.: While executing Remote.
```

支持 DDL 语句默认的 on cluster

最近更新时间：2024-10-17 20:41:22

特性描述

默认情况下，原生 ClickHouse 的 DDL 操作只在链接所在的节点上生效。如果需要 DDL 变更在整个集群维度生效，需要显示在 DDL 语句中添加 `ON CLUSTER default_cluster` 语句。

LibraSQL 提供了默认 DDL 集群配置：`ddl_default_oncluster`。打开该配置后，DDL 操作会默认在 `default_cluster` 上执行，效果相当于在 DDL 语句中添加 `ON CLUSTER default_cluster`。该配置默认关闭。

LibraSQL 支持会话级别打开配置。打开前，执行 DDL 操作仅在当前节点执行。打开后，执行 DDL 操作将在集群中各节点执行。

使用限制

暂不支持在 TDSQL-H LibraDB 控制台修改该参数，如需全局开启该配置，请 [提交工单](#)。

使用说明

打开 DDL 集群配置

```
set ddl_default_oncluster = 'default_cluster';
```

打开 DDL 集群配置 `ddl_default_oncluster`，执行 DDL 操作将在集群各节点执行。

示例：

```
c 0.k s.svc.cluster.local :) set ddl_default_oncluster = 'default_cluster';
```

```
SET ddl_default_oncluster = 'default_cluster'
```

```
Query id: f6a7a1aa-2181-428e-8a41-15ec00ce8bd2
```

```
Ok.
```

```
0 rows in set. Elapsed: 0.002 sec.
```

```
c 0.k s.svc.cluster.local :) create database testzm4;
```

```
CREATE DATABASE testzm4
```

```
Query id: 9543b57f-3acc-4afe-bd34-0815a86be120
```

host	port	status	error	num_hosts_remaining	num_hosts_active
c 0.k s.svc.cluster.local	9000	0		1	0
c 1.k s.svc.cluster.local	9000	0		0	0

```
2 rows in set. Elapsed: 0.115 sec.
```

关闭 DDL 集群配置

```
set ddl_default_oncluster = '';
```

关闭 DDL 集群配置 `ddl_default_oncluster`，执行 DDL 操作仅在当前节点执行。

示例：

```
c 0.k s.svc.cluster.local :) set ddl_default_oncluster = '';
```

```
SET ddl_default_oncluster = ''
```

```
Query id: de5572e9-df48-49f1-a496-1fd4cceaec86
```

```
Ok.
```

```
0 rows in set. Elapsed: 0.001 sec.
```

```
c 0.k s.svc.cluster.local :) create database testzm5;
```

```
CREATE DATABASE testzm5
```

```
Query id: b791cdc2-65dd-43b2-99b8-957322efea7c
```

```
Ok.
```

```
0 rows in set. Elapsed: 0.005 sec.
```

CDC 特性

事务最终一致性

最近更新时间：2024-10-17 20:41:22

特性描述

对于 MySQL 这种 OLTP 型数据库同步数据时，如果一个事务涉及了多行数据或者是多个表的数据的操作时，原生 ClickHouse（MaterializedMySQL）不支持事务的原子可见，可能出现幻读等异常数据。

TDSQL-H LibraDB 的 MVCC 版本针对该问题进行了优化，对于从 MySQL 这种 OLTP 型数据库同步，一个事务的变更数据在 TDSQL-H LibraDB 的多个分片、多个分区上都是原子可见的，保证的数据的严格一致。即 LibraDB 的 CDC 任务保证了源端数据库事务的最终一致性。

使用限制

仅 TDSQL-H LibraDB 的 MVCC 版本支持。

前提条件

- 已创建上线 MVCC 版本的 TDSQL-H LibraDB 实例。
- 已创建 CDC 任务，同步类型为表结构 + 全量数据 + 增量数据。

使用说明

同一批删除多行数据，非 MVCC 版本不支持数据的原子可见，MVCC 版本数据原子可见。多个表的原子可见与同一批删除多行数据原理类似。在 CDC 任务目标端（即 TDSQL-H LibraDB）查询非 MVCC 版本和 MVCC 版本同步的数据。

下文以同一批删除多行数据为例。

- 非 MVCC 版本：**

1.1 查询 t3 表中的数据。

```
select * from t3;
```

返回结果

Query id: 5312f276-027e-44cc-9ff3-7e3aa752705d

id	name	birth	_sign	_version
1	dd	2022-06-21 14:29:08	1	5295
2	cc	2022-06-22 14:29:22	1	5296
4	xx	2022-06-24 14:29:49	1	5298
8	tt	2022-06-23 14:29:34	1	5297

```
4 rows in set. Elapsed: 0.002 sec.
```

1.2 在 CDC 任务源端同时删除 id=1、id=2、id=4 的数据后，并且迅速在目的端 TDSQL-H LibraDB 中查询数据。因为不支持数据的原子可见性，故可能返回删除过程中的任意中间状态。

返回结果示例

```
SELECT count(*) AS cnt FROM t3 FINAL WHERE _sign > 0;
```

Query id: f7dba001-a0cc-41f7-bdfa-04b9eec12e0a

cnt
2

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

在短时间内，上述查询结果可能是 4、3、2、1 中的任意值（部分数据没有完全同步），但最终查询结果为 1（所有数据均已同步至目标端，并且可见）。

● MVCC 版本：

1.1 查询 t5 表中的数据。

返回结果

```
SELECT count(*) AS cnt FROM t5
```

Query id: 95cdaac0-31ee-41ef-a6dc-b58208dbb8c4

cnt
4

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

1.2 在 CDC 任务源端同时删除 id=1、id=2、id=4 的数据后，在目的端 TDSQL-H LibraDB 中查询数据，数据严格实时一致（事务中所有变更同时生效）。

```
select count(*) as cnt from t5;
```

返回结果

Query id: 6420d6a8-5ae6-407e-a080-f706a4509a16

```
┌cnt┐  
└  1┘
```

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

因为数据严格实时一致，故查询结果只可能是4（所有数据均未同步到目标端）或者是1（所有数据均已同步到目标端），即为数据原子可见。

无 FINAL 一致性查询

最近更新时间：2024-10-17 20:41:22

特性描述

原生 ClickHouse 实时 update 和 delete 性能较弱。传统的做法是针对同步的数据表，自动映射成分布式表和本地表，业务层实际通过分布式表来进行数据读写。查询获取一致性的数据需要使用 FINAL，查询性能差。而 LibraDB 优化的内容则是将本地表和分布式表中除原表字段外，额外为每个表新增字段 `_sign` 和 `_version`，使用 insert 代替 update 和 delete，通过删除标志和版本来替代查询等待。

- 查询时，需带上条件 `_sign=1`。
- 具体转换逻辑如下：
 - 新增字段含义
 - `_version` 是单调递增的版本数据。下文以初始值1为例。
 - `_sign=1` 代表有效数据，`_sign=-1` 为删除的数据。
 - INSERT
 - 插入新的一行数据。
 - 其他数据不变，`_sign=1`，`_version=1`。
 - DELETE
 - 插入新的一行数据。
 - 其他数据不变，`_sign=-1`，`_version = _version+1`。
 - UPDATE
 - 对于修改的列中，不包含排序键的情况：
 - 插入新的一行数据。
 - 使用后镜像，`_sign=1`，`_version = _version+1`。
 - 对于修改的列中，包含了排序键的情况：
 - 插入两行数据。
 - 第一行，使用前镜像，`_sign=-1`，`_version = _version+1`。
 - 第二行，使用新镜像，`_sign=1`，`_version = _version+1`。

优化后

TDSQL-H LibraDB 提供 MVCC 版本，对于 CDC 同步场景，将 update/delete 转换为 insert+delete 模式，无需 FINAL 即可保证目标端的数据与源端完全一致，且几乎没有查询性能的损失。同时在本地表原有字段基础上增加 `_dversion` 隐藏字段，用于删除操作标记。

基于 MVCC 版本，update/delete 转换与上述内置字段值变化关系如下：

- **update**：转换成 delete + insert 模式，delete 操作会对原数据打删除标记并更新 `_dversion` 的值，对于删除标记的主键记录会单独进行缓存，以便于异步聚合删除，insert 会新插入一条数据，并更新 `_version`

值，`_dversion` 取值为0。

- **delete:** 转换成 delete 模式，对原数据打删除标记，更新 `_dversion` 值，并记录到缓存中，包括删除的主键和删除的时间戳，`_version` 值不变。

前提条件

- 已创建上线 MVCC 版本的 TDSQL-H LibraDB 实例。
- 已创建 CDC 任务，同步类型为表结构 + 全量数据 + 增量数据。

使用说明

下文以一个示例来演示数据一致性查询，源端 OLTP 库表为 migrate.t1。
在源端更新 id=8 的数据。

```
update t1 set birth=now() where id=8
```

在 CDC 任务目标端（即 TDSQL-H LibraDB）查看优化前和优化后的 MVCC 版本同步的数据。

优化前：

#查询数据时，与原生 ClickHouse 的 VersionedCollapsingMergeTree 引擎相同，数据出现三条

```
select * from t1 where id=8;
```

Query id: d8b66e67-ba54-4b0e-850b-b2b54ceb1ec6

id	name	birth	_sign	_version
8	tt	2022-05-29 12:32:26	1	194
8	tt	2022-05-29 12:28:20	1	66
8	tt	2022-05-29 12:28:20	-1	194

3 rows in set. Elapsed: 0.004 sec.

#如果要获取正确的数据效果，通过用 final 实现，达到数据最终一致目标

```
select * from t1 final where id=8;
```

Query id: 72e7fe14-0b1e-456f-bcbc-4bc91db4fd12

id	name	birth	_sign	_version
8	tt	2022-05-29 12:32:26	1	194

1 rows in set. Elapsed: 0.003 sec.

优化后:

基于 MVCC 版本的内核，直接显示最新数据，保证数据一致

```
select * from t1 where id=8;
```

Query id: 9f6107b5-f45b-47d1-93ea-5f139a943bbe

id	name	birth	_sign	_version
8	tt	2022-05-29 12:32:26	1	370

1 rows in set. Elapsed: 0.007 sec.