数据库知识库

一、TIDB数据库相关问题

1.查看集群版本时，会出现timeout问题

```
[root@wbjdnchzttdb01 tidb_source]# tiup cluster --version
tiup is checking updates for component cluster ...timeout!
Starting component cluster: /root/.tiup/components/cluster/v1.11.1/tiup-cluster --version
tiup version 1.11.1 tiup
Go Version: go1.19.2
Git Ref: v1.11.1
GitHash: b95172df211e4f9b643590f2dd8436ad60c72b38
```

解决办法：

```
[root@wbjdnchzttdb01 tidb_source]# rm -rf ~/.tiup/manifests/*
[root@wbjdnchzttdb01 tidb_source]# tiup cluster --version
tiup is checking updates for component cluster ...
Starting component `cluster`: /root/.tiup/components/cluster/v1.11.1/tiup-cluster --version
tiup version 1.11.1 tiup
Go Version: go1.19.2
Git Ref: v1.11.1
GitHash: b95172df211e4f9b643590f2dd8436ad60c72b38
```

2.执行Br备份命令时，报错：

```
[root@wbjdnchzttdb01 tidb_source]# tiup br backup full \
>     --pd "10.3.8.230:2379" \
>     --filter 'bj_sjzt_db.zhcx*incoming_alldata' \
>     --storage "local:///acdata/backup/backup_`date +%F`_bj_sjzt_db" \
>     --ratelimit 128 \
>     --log-file bj_sjzt_db_2tabs_backup.log
tiup is checking updates for component br ...timeout!
Error: timestamp manifest has a version number < the old manifest (1, 21)
```

解决办法：

```
[root@host-pro-tidb-02 cdc-to-sjzt-kafaka]# br --version
Release Version: v5.4.0
Git Commit Hash: 55f3b24c1c9f506bd652ef1d162283541e428872
Git Branch: heads/refs/tags/v5.4.0
Go Version: go1.16.4
UTC Build Time: 2022-01-25 08:36:34
Race Enabled: false
[root@host-pro-tidb-02 cdc-to-sjzt-kafaka]# which br
/usr/bin/br
[root@host-pro-tidb-02 cdc-to-sjzt-kafaka]# rm -rf /usr/bin/br
[root@host-pro-tidb-02 cdc-to-sjzt-kafaka]# br --version
Release Version: v5.4.0
Git Commit Hash: 55f3b24c1c9f506bd652ef1d162283541e428872
Git Branch: heads/refs/tags/v5.4.0
Go Version: go1.16.4
UTC Build Time: 2022-01-25 08:36:34
Race Enabled: false
[root@host-pro-tidb-02 cdc-to-sjzt-kafaka]# which br
/usr/bin/br
[root@host-pro-tidb-02 cdc-to-sjzt-kafaka]# rm -rf /usr/bin/br
```

3.创建cdc同步任务时，报错：

```
[root@wbjdnchzttdb01 bj_sjzt_to_bb]# tiup cdc cli changefeed create --pd="http://10.3.8.230:2379" --sink-uri="mysql://root:SJZT_tidb@10.3.8.227:4000/" --changefeed-id="bj-sjzt-to-bb-01" --sort-engine="unified" --config="./bj_sjzt_to_bb_01.toml" --start-ts='439919660410863622'
tiup is checking updates for component cdc ...
Starting component `cdc`: /root/.tiup/components/cdc/v6.5.0/cdc cli changefeed create --pd=http://10.3.8.230:2379 --sink-uri=mysql://root:SJZT_tidb@10.3.8.227:4000/ --changefeed-id=bj-sjzt-to-bb-01 --sort-engine=unified --config=./bj_sjzt_to_bb_01.toml --start-ts=439919660410863622
Error: election: no leader
```

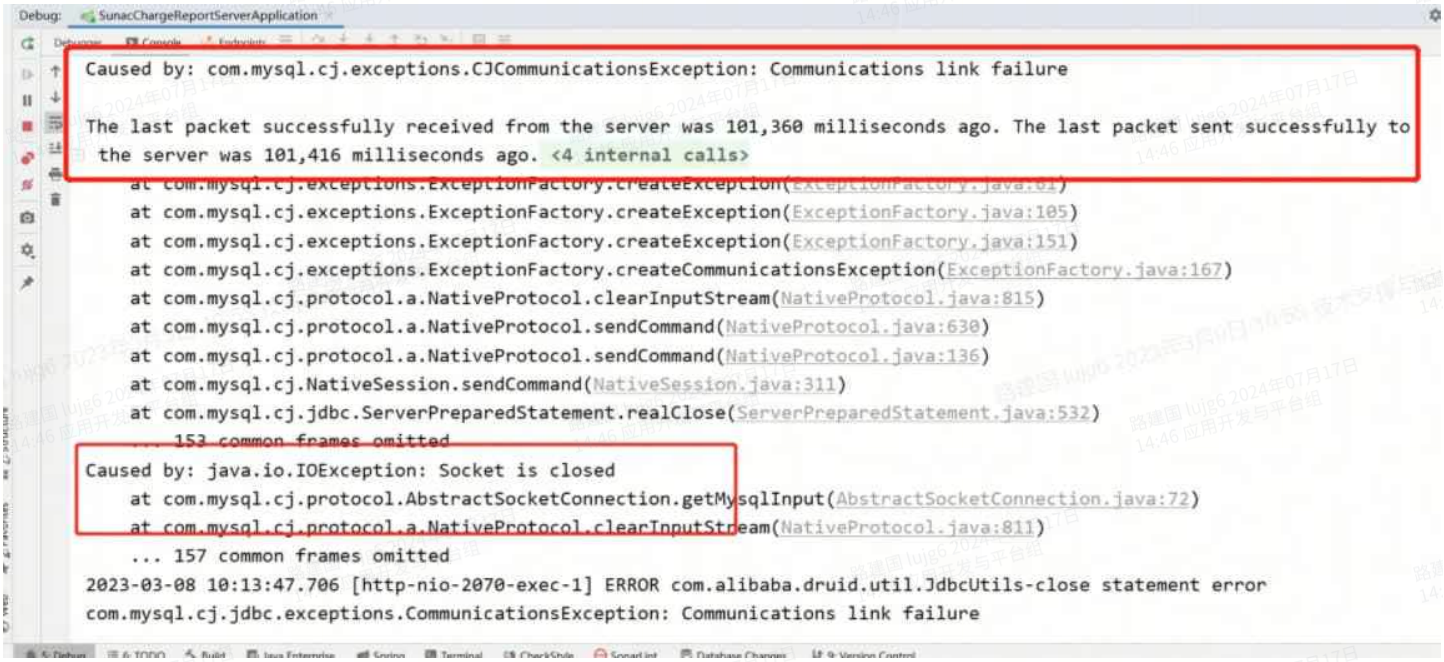
解决办法：（是因为没有找到cdc的leader组件，检查是否安装，并进行安装）

```
[root@wbjdnchzttdb01 tidb_source]# tiup cluster scale-out bj_sjzt_tidb scale-out.yaml --user root
```

4.添加同步任务时，报错：

```
[root@wbjdnchzttdb01 bj_sjzt_to_bb]# tiup cdc cli changefeed create --server=http://10.3.8.231:8300 --sink-uri="mysql://root:SJZT_tidb@10.3.8.227:4000/" --changefeed-id="bj-sjzt-to-bb-01" --sort-engine="unified" --config="./bj_sjzt_to_bb_01.toml" --start-ts='439919660410863622'
tiup is checking updates for component cdc ...
Starting component `cdc`: /root/.tiup/components/cdc/v6.5.0/cdc cli changefeed create --server=http://10.3.8.231:8300 --sink-uri=mysql://root:SJZT_tidb@10.3.8.227:4000/ --changefeed-id=bj-sjzt-to-bb-01 --sort-engine=unified --config=./bj_sjzt_to_bb_01.toml --start-ts=439919660410863622
Error: [CDC:ErrAPIInvalidParam]invalid api parameter: [CDC:ErrInternalServerError]internal server error%!(EXTRA string=changefeed with same ID was just removed, please wait 11.287818702s)
```

5.tidb集群升级版本后，主从库集群版本不同，从库为v6.1.5版本。出现如下报错：



解决办法：升级jdbc版本> 8.0.29或直接使用tidb-jdbc驱动。

问题里面那个：https://docs.pingcap.com/zh/tidb/stable/dev-guide-choose-driver-or-orm 9
其实就是一个官方说明，其中提到了几点明显的区别：
强烈建议使用 JDBC 5.1 的最后一个版本 5.1.49。因为当前 8.0.29 版本有未合并的 Bug 修复，在与 TiDB 共同使用时可能会导致线程卡死。在 MySQL JDBC 8.0 未合并此修复之前，建议不要升级至 8.0

版本。
TiDB-JDBC 4 是基于 MySQL 8.0.29 的定制版本。TiDB-JDBC 基于 MySQL 官方 8.0.29 版本编译，修复了原 JDBC 在 prepare 模式下多参数、多字段 EOF 的错误，并新增 TiCDC snapshot 自动维护和 SM3 认证插件等功能。
基于 SM3 的认证仅在 TiDB 版本的 MySQL Connector/J 中支持。

探活配置

连接池维护到 TiDB 的长连接，TiDB 默认不会主动关闭客户端连接（除非报错），但一般客户端到 TiDB 之间还会有 LVS 或 HAProxy 之类的网络代理，它们通常会在连接空闲一定时间（由代理的 idle 配置决定）后主动清理连接。除了注意代理的 idle 配置外，连接池还需要进行保活或探测连接。

如果常在 Java 应用中看到以下错误：

```
The last packet sent successfully to the server was 3600000 milliseconds ago. The driver has not received any pack
```

如果 `n milliseconds ago` 中的 `n` 如果是 0 或很小的值，则通常是执行的 SQL 导致 TiDB 异常退出引起的报错，推荐查看 TiDB stderr 日志；如果 n 是一个非常大的值（比如这里的 3600000），很可能是因为这个连接空闲太久然后被中间 proxy 关闭了，通常解决方式除了调大 proxy 的 idle 配置，还可以让连接池执行以下操作：

- 每次使用连接前检查连接是否可用。
- 使用单独线程定期检查连接是否可用。
- 定期发送 test query 保活连接。

不同的连接池实现可能会支持其中一种或多种方式，可以查看所使用的连接池文档来寻找对应配置。

4.tidb数据库表健康度检查及配置调整：**(北京主库，从库，准实时，三套集，目前已经改回0.5)**

```

SHOW STATS_HEALTHY where healthy <=70 and healthy >=60;
show variables like 'tidb_auto_analyze_ratio';   -- 0.5
show variables like 'tidb_enable_auto_analyze';  -- ON
set global tidb_auto_analyze_ratio=0.7;


-- 北京主库和北京准实时从库最新调整参数：
set global tidb_enable_auto_analyze=1;
set global tidb_enable_fast_analyze=1;
set global tidb_auto_analyze_ratio=0.3;
set global tidb_auto_analyze_start_time='00:00 +0800';
set global tidb_auto_analyze_end_time ='08:00 +0800';


-- 主库新修改为0.3，代表表的数据变化超过30%时，将触发自动收集
mysql> set global tidb_auto_analyze_ratio=0.3;
Query OK, 0 rows affected (0.02 sec)

```

```
-- 需要开启自动analyze table功能，并且，设定执行时间在业务低峰时间段

mysql> show variables like 'tidb_auto_analyze_start_time';
+-------------------------------+-------------+
| Variable_name                 | Value       |
+-------------------------------+-------------+
| tidb_auto_analyze_start_time | 00:00 +0000 |
+-------------------------------+-------------+
1 row in set (0.01 sec)

mysql> show variables like 'tidb_auto_analyze_end_time';
+-----------------------------+-------------+
| Variable_name               | Value       |
+-----------------------------+-------------+
| tidb_auto_analyze_end_time | 08:00 +0000 |
+-----------------------------+-------------+
1 row in set (0.00 sec)

-- 修改为北京起始时间
mysql> set global tidb_auto_analyze_start_time='00:00 +0800';
Query OK, 0 rows affected (0.01 sec)

mysql> set global tidb_auto_analyze_end_time ='08:00 +0800';
Query OK, 0 rows affected (0.03 sec)

-- 10.3.8.231 准实时集群收集统计信息配置参数调整:
set global tidb_enable_auto_analyze=1;
set global tidb_enable_fast_analyze=1;
set global tidb_auto_analyze_ratio=0.3;
set global tidb_auto_analyze_start_time='00:00 +0800';
set global tidb_auto_analyze_end_time ='08:00 +0800';


MySQL [maindb]>  show variables like '%analyze%';
+-------------------------------+-------------+
| Variable_name                 | Value       |
+-------------------------------+-------------+
| tidb_analyze_version          | 2           |
| tidb_auto_analyze_end_time    | 23:59 +0000 |
| tidb_auto_analyze_ratio       | 0.5         |
| tidb_auto_analyze_start_time  | 00:00 +0000 |
| tidb_enable_analyze_snapshot  | OFF         |
| tidb_enable_auto_analyze      | ON          |
| tidb_enable_fast_analyze      | ON          |
| tidb_max_auto_analyze_time    | 43200       |
| tidb_mem_quota_analyze        | -1          |
| tidb_persist_analyze_options  | ON          |
```

```
+------------------------------+-------------+
10 rows in set (0.00 sec)


-- 暂时不调整的参数，主库集群默认12个小时
mysql> show variables like 'tidb_max_auto_analyze_time';
+------------------------------+-------+
| Variable_name                | Value |
+------------------------------+-------+
| tidb_max_auto_analyze_time | 43200 |
+------------------------------+-------+
1 row in set (0.00 sec)

自动统计信息收集最大执行时间受参数 tidb_max_auto_analyze_time限制，我记得有个bug，对分区表可能不能正确限制，会导致窗口外还有analyze语句发起的情况出现。

-- 查看有哪些表进行了自动收集
从 TiDB v7.3.0 起，你可以通过系统表 mysql.analyze_jobs 或者 SHOW ANALYZE STATUS 查看当前 ANALYZE 任务的进度。
mysql> show analyze status;
+--------------+-----------------------------------------+----------------+----------------------------------------------------------------------------------------------------------------------------+---------------+---------------------+-------------+----------+-------------+-----------------+------------+
| Table_schema | Table_name                              | Partition_name | Job_info                                                                                                                   | Processed_rows | Start_time          | End_time            | State    | Fail_reason | Instance        | Process_ID |
+--------------+-----------------------------------------+----------------+----------------------------------------------------------------------------------------------------------------------------+---------------+---------------------+-------------+----------+-------------+-----------------+------------+
| maindb       | es_platform_domain_view                 |                | auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate                                                    |          3978 | 2024-06-12 04:50:42 | 2024-06-12 04:50:46 | finished | NULL        | 10.3.8.193:4000 |       NULL |
| maindb       | es_charge_handin_statistics             |                | auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate                                                    |          1694 | 2024-06-12 03:59:42 | 2024-06-12 03:59:47 | finished | NULL        | 10.3.8.193:4000 |       NULL |
| maindb       | es_charge_handin_statistics             |                | auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate                                                    |          1692 | 2024-06-12 03:55:42 | 2024-06-12 03:55:45 | finished | NULL        | 10.3.8.193:4000 |       NULL |
| maindb       | es_charge_handin_statistics             |                | auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
```

```
                                |          1692 | 2024-06-12 03:52:42 | 2024-06-
12 03:52:46 | finished | NULL        | 10.3.8.193:4000 |       NULL |
 95 | maindb       | es_charge_handin_statistics             |                |
auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                |          1693 | 2024-06-12 03:48:45 | 2024-06-
12 03:48:50 | finished | NULL        | 10.3.8.193:4000 |       NULL |
 96 | maindb       | es_charge_handin_statistics             |                |
auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                |          1693 | 2024-06-12 03:45:42 | 2024-06-
12 03:45:48 | finished | NULL        | 10.3.8.193:4000 |       NULL |
 97 | maindb       | es_charge_handin_statistics             |                |
auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                |          1693 | 2024-06-12 03:41:45 | 2024-06-
12 03:41:51 | finished | NULL        | 10.3.8.193:4000 |       NULL |
 98 | maindb       | es_charge_handin_statistics             |                |
auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                |          1693 | 2024-06-12 03:37:42 | 2024-06-
12 03:37:47 | finished | NULL        | 10.3.8.193:4000 |       NULL |
 99 | maindb       | es_charge_handin_statistics             |                |
auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                |          1693 | 2024-06-12 03:33:45 | 2024-06-
12 03:33:49 | finished | NULL        | 10.3.8.193:4000 |       NULL |
100 | maindb       | es_charge_incoming_data_monitor         |                |
auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                |          1684 | 2024-06-12 03:00:47 | 2024-06-
12 03:00:48 | finished | NULL        | 10.3.8.193:4000 |       NULL |
101 | maindb       | es_charge_incoming_data_monitor         |                |
auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                |          1684 | 2024-06-12 02:00:42 | 2024-06-
12 02:00:45 | finished | NULL        | 10.3.8.193:4000 |       NULL |
102 | maindb       | es_platform_domain_view                 |                |
auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                |          3978 | 2024-06-12 00:22:45 | 2024-06-
12 00:22:49 | finished | NULL        | 10.3.8.193:4000 |       NULL |
103 | maindb       | xxl_job_log                             |                |
auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                |         12425 | 2024-06-12 00:04:51 | 2024-06-
12 00:04:54 | finished | NULL        | 10.3.8.193:4000 |       NULL |
104 | maindb       | es_charge_object_progress_on_collection |                |
auto analyze table all columns with 256 buckets, 500 topn, 0.8524488530688159
samplerate                     |        129040 | 2024-06-12 00:04:45 | 2024-06-
12 00:04:51 | finished | NULL        | 10.3.8.193:4000 |       NULL |
105 | maindb       | es_charge_cost_object_temp              |                |
auto analyze table all columns with 256 buckets, 500 topn,
0.024932064280180556 samplerate                |        142691 | 2024-06-12
00:04:42 | 2024-06-12 00:04:44 | finished | NULL        | 10.3.8.193:4000 |
   NULL |
```

```
106 | maindb       | es_charge_bill_owner_info               |                 |
    auto analyze table all columns with 256 buckets, 500 topn,
    0.050606472567841426 samplerate                  |        2178201 | 2024-06-12
    00:04:00 | 2024-06-12 00:04:42 | finished | NULL        | 10.3.8.193:4000 |
       NULL |
107 | maindb       | es_charge_cost_owner_temp               |                 |
    auto analyze table all columns with 256 buckets, 500 topn, 0.02242780042487225
    samplerate                   |         179737 | 2024-06-12 00:03:58 | 2024-06-
    12 00:04:00 | finished | NULL        | 10.3.8.193:4000 |       NULL |
108 | maindb       | es_charge_owner_fee                     |                 |
    merge global stats for maindb.es_charge_owner_fee's index
    es_charge_owner_fee_index2                       |              0 | 2024-06-
    12 00:03:57 | 2024-06-12 00:03:58 | finished | NULL        | 10.3.8.193:4000 |
          NULL |
109 | maindb       | es_charge_owner_fee                     |                 |
    merge global stats for maindb.es_charge_owner_fee's index
    es_charge_owner_fee_name                         |              0 | 2024-06-
    12 00:03:52 | 2024-06-12 00:03:57 | finished | NULL        | 10.3.8.193:4000 |
          NULL |
110 | maindb       | es_charge_owner_fee                     |                 |
    merge global stats for maindb.es_charge_owner_fee's index idx_fld_project_guid
                                     |              0 | 2024-06-12 00:03:52 | 2024-06-
    12 00:03:52 | finished | NULL        | 10.3.8.193:4000 |       NULL |
111 | maindb       | es_charge_owner_fee                     |                 |
    merge global stats for maindb.es_charge_owner_fee columns
                                     |              0 | 2024-06-12 00:03:01 | 2024-06-
    12 00:03:52 | finished | NULL        | 10.3.8.193:4000 |       NULL |
112 | maindb       | es_charge_owner_fee                     |                 |
    merge global stats for maindb.es_charge_owner_fee's index idx_biz_id
                                     |              0 | 2024-06-12 00:03:00 | 2024-06-
    12 00:03:01 | finished | NULL        | 10.3.8.193:4000 |      NULL |
113 | maindb       | es_charge_owner_fee                     |                 |
    merge global stats for maindb.es_charge_owner_fee's index
    es_charge_owner_fee_index1                       |              0 | 2024-06-
    12 00:02:59 | 2024-06-12 00:03:00 | finished | NULL        | 10.3.8.193:4000 |
          NULL |
114 | maindb       | es_charge_owner_fee                     |                 |
    merge global stats for maindb.es_charge_owner_fee's index
    idx_fld_object_guid_fld_project_guid             |              0 | 2024-06-
    12 00:02:53 | 2024-06-12 00:02:59 | finished | NULL        | 10.3.8.193:4000 |
          NULL |
115 | maindb       | es_charge_owner_fee                     |                 |
    merge global stats for maindb.es_charge_owner_fee's index
    idx_fld_area_guid_fld_adjust_guid_fld_object_guid |              0 | 2024-06-
    12 00:02:48 | 2024-06-12 00:02:53 | finished | NULL        | 10.3.8.193:4000 |
          NULL |
```

```
116 | maindb       | es_charge_owner_fee                        |                |
    merge global stats for maindb.es_charge_owner_fee's index idx_fld_owner_guid
                                 |              0 | 2024-06-12 00:02:41 | 2024-06-
    12 00:02:48 | finished | NULL        | 10.3.8.193:4000 |       NULL |
117 | maindb       | es_charge_owner_fee                        |                |
    merge global stats for maindb.es_charge_owner_fee's index idx_fld_allot_date
                                 |              0 | 2024-06-12 00:02:41 | 2024-06-
    12 00:02:41 | finished | NULL        | 10.3.8.193:4000 |       NULL |
118 | maindb       | es_charge_owner_fee                        |                |
    merge global stats for maindb.es_charge_owner_fee's index primary
                                 |              0 | 2024-06-12 00:01:41 | 2024-06-
    12 00:02:41 | finished | NULL        | 10.3.8.193:4000 |       NULL |
119 | maindb       | es_charge_owner_fee                        | p55            |
    auto analyze table all columns with 256 buckets, 500 topn, 0.03052027073144881
    samplerate                   |        3606706 | 2024-06-12 00:00:03 | 2024-06-
    12 00:01:40 | finished | NULL        | 10.3.8.193:4000 |       NULL |
120 | maindb       | es_charge_handin_statistics                |                |
    auto analyze table all columns with 256 buckets, 500 topn, 1 samplerate
                                 |           1692 | 2024-06-12 00:00:00 | 2024-06-
    12 00:00:03 | finished | NULL        | 10.3.8.193:4000 |       NULL |
121 +--------------+--------------------------------------------+----------------+----
    ------------------------------------------------------------------------------
    -------------------------+---------------+---------------------+-------------
    --------+----------+-------------+-----------------+------------+
122 30 rows in set (0.00 sec)
```

在 TiDB 中 `auto analyze` 的触发策略是怎样的?

当一张表或分区表的单个分区达到 1000 条记录，且表或分区的（修改数/当前总行数）比例大于 `tidb_auto_analyze_ratio` 的时候，会自动触发 `ANALYZE` 语句。

`tidb_auto_analyze_ratio` 的默认值为 `0.5`，即默认开启触发 `auto analyze`。注意该变量值不建议大于等于 `pseudo-estimate-ratio`（默认值为 `0.8`），否则优化器可能会使用 pseudo 统计信息。TiDB 从 v5.3.0 开始引入 `tidb_enable_pseudo_for_outdated_stats` 变量，当设置为 `OFF` 时，即使统计信息过期也不会使用 pseudo 统计信息。

你可以用系统变量 `tidb_enable_auto_analyze` 关闭 `auto analyze`。

5.使用v6.5.0版本导出单个表数据时，tiup dumpling报错：

```
1 [2023/03/28 10:46:41.520 +08:00] [INFO] [client.go:779] ["[pd] stop fetching
  the pending tso requests due to context canceled"] [dc-location=global]
2 [2023/03/28 10:46:41.520 +08:00] [INFO] [client.go:716] ["[pd] exit tso
  dispatcher"] [dc-location=global]
```

```
[2023/03/28 10:46:41.520 +08:00] [ERROR] [main.go:77] ["dump failed error stack info"] [error="sql: SELECT * FROM `maindb`.`es_charge_generate_record_detail1` LIMIT 1, args: []: Error 1105: DB::TiFlashException: Failed to register MPP Task MPP<query:440394475472683015,task:4>, reason: query is being aborted, error message = Receive cancel request from TiDB"] [errorVerbose="Error 1105: DB::TiFlashException: Failed to register MPP Task MPP<query:440394475472683015,task:4>, reason: query is being aborted, error message = Receive cancel request from TiDB\nsql: SELECT * FROM `maindb`.`es_charge_generate_record_detail1` LIMIT 1, args: []\ngithub.com/pingcap/tidb/dumpling/export.simpleQueryWithArgs\n\tgithub.com/pingcap/tidb/dumpling/export/sql.go:1147\ngithub.com/pingcap/tidb/dumpling/export.
(*BaseConn).QuerySQL.func1\n\tgithub.com/pingcap/tidb/dumpling/export/conn.go:42\ngithub.com/pingcap/tidb/br/pkg/utils.WithRetry\n\tgithub.com/pingcap/tidb/br/pkg/utils/retry.go:52\ngithub.com/pingcap/tidb/dumpling/export.
(*BaseConn).QuerySQL\n\tgithub.com/pingcap/tidb/dumpling/export/conn.go:34\ngithub.com/pingcap/tidb/dumpling/export.GetColumnTypes\n\tgithub.com/pingcap/tidb/dumpling/export/sql.go:492\ngithub.com/pingcap/tidb/dumpling/export.dumpTableMeta\n\tgithub.com/pingcap/tidb/dumpling/export/dump.go:1187\ngithub.com/pingcap/tidb/dumpling/export.
(*Dumper).dumpDatabases\n\tgithub.com/pingcap/tidb/dumpling/export/dump.go:423\ngithub.com/pingcap/tidb/dumpling/export.
(*Dumper).Dump\n\tgithub.com/pingcap/tidb/dumpling/export/dump.go:295\nmain.main\n\t./main.go:74\nruntime.main\n\truntime/proc.go:250\nruntime.goexit\n\truntime/asm_amd64.s:1594"]

dump failed: sql: SELECT * FROM `maindb`.`es_charge_generate_record_detail1` LIMIT 1, args: []: Error 1105: DB::TiFlashException: Failed to register MPP Task MPP<query:440394475472683015,task:4>, reason: query is being aborted, error message = Receive cancel request from TiDB
```

4. v6.5.0版本设置内存控制参数，但是，发现不生效。

```
tidb:
   alter-primary-key: true
   log.file.max-days: 30
   log.level: info
   mem-quota-query: 16106127360
   new_collations_enabled_on_first_bootstrap: true
   oom-action: cancel
   oom-use-tmp-storage: true
   performance.max-procs: 30
```

```
10      performance.txn-total-size-limit: 4221225472
```

从v6.1.0版本新增 `tidb_mem_oom_action` 变量，由TiDB 配置项 oom-action 转化而来。

tidb_mem_oom_action	新增	由 TiDB 配置项 oom-action 转化而来。

```
1 mysql> show variables like '%oom%';
2 +---------------------------------+-------+
3 | Variable_name                   | Value |
4 +---------------------------------+-------+
5 | tidb_enable_tmp_storage_on_oom | ON    |
6 | tidb_mem_oom_action            | LOG   |
7 +---------------------------------+-------+
8 2 rows in set (0.00 sec)
```

即使通过# tiup cluster edit-config pro-sunac-tidb-newfee，调整配置文件，需要将oom-action:cancel改为 `tidb_mem_oom_action` : cancel。仍然不生效。这是因为从v6.5.0版本，此参数，只支持在系统参数命令行调整生效，不支持修改配置文件生效。

此时，只需要使用命令行登录tidb，执行如下语句：

```
1 mysql> set global  tidb_mem_oom_action='cancel';
2 Query OK, 0 rows affected (0.02 sec)
3
4 mysql> show variables like '%oom%';
5 +---------------------------------+--------+
6 | Variable_name                   | Value  |
7 +---------------------------------+--------+
8 | tidb_enable_tmp_storage_on_oom | ON     |
9 | tidb_mem_oom_action            | CANCEL |
10 +---------------------------------+--------+
11 2 rows in set (0.01 sec)
12
```

再次，拿大SQL进行查询，超过mem-quota-query，就会抛出OOM。

```
1 mysql> select
  inc.fld_owner_guid,inc.fld_object_guid,inc.fld_project_guid,inc.fld_object_name
```

```
    ,inc.fld_amount,inc.fld_total,inc.fld_desc, IF((inc.fld_busi_type = 4 or
    inc.fld_busi_type = 13 or (inc.fld_busi_type = 0 and inc.fld_pay_mode_guid =
    inc.fld_project_guid)), inc.fld_project_name, '') AS fld_project_name,
    IF((inc.fld_busi_type = 4 or inc.fld_busi_type = 13 or (inc.fld_busi_type = 0
    and inc.fld_pay_mode_guid = inc.fld_project_guid)), 'ConstituteFeeDesc', '')
    AS fld_fee_desc, IF((inc.fld_busi_type = 4 or inc.fld_busi_type = 13 or
    (inc.fld_busi_type = 0 and inc.fld_pay_mode_guid = inc.fld_project_guid)),
    inc.fld_allot_date, null) AS fld_allot_date, ifnull(fee.fld_resource, '') AS
    fld_resource, IF(inc.fld_back_fee_guid = '', IF(inc.fld_busi_type = 13, '返销',
    '正常'), '退款') AS fld_cancel_name,inc.fld_modify_date AS
    fld_modify_date,inc.fld_modify_user AS fld_user_name,
    inc.fld_create_date,inc.fld_incoming_fee_guid,inc.fld_operate_date from
    es_charge_incoming_data inc left join es_charge_owner_fee fee on
    inc.fld_area_guid = fee.fld_area_guid and inc.fld_owner_fee_guid =
    fee.fld_guid inner join es_info_area_info a on a.fld_guid=inc.fld_area_guid
    where a.fld_dq='重庆大区' and inc.fld_cancel <> 1 union select
    mppi.fld_owner_guid,mppi.fld_object_guid,mppi.fld_pay_mode_guid,o.fld_name AS
    fld_object_name,mppi.fld_amount,mppi.fld_amount AS fld_total,concat('初始化金
    额: ', mppi.fld_amount) AS fld_desc, '' AS fld_project_name,'' AS fld_fee_desc,
    null AS fld_allot_date,'初始化' AS fld_resource,' 常 ' AS fld_cancel_name,'1900-
    01-01' AS fld_modify_date, 'autouser' AS fld_user_name,'1900-01-01 00:00:00'
    AS fld_create_date,'' AS fld_incoming_fee_guid,'1900-01-01 00:00:00' AS
    fld_operate_date from es_charge_pay_mode_pre_pay_init mppi left join
    es_info_object o on mppi.fld_area_guid = o.fld_area_guid and
    mppi.fld_object_guid = o.fld_guid inner join es_info_area_info a on
    a.fld_guid=mppi.fld_area_guid where a.fld_dq='重庆大区';
2   ERROR 1105 (HY000): Out Of Memory Quota![conn_id=1863700796382642699]
```

相关参数说明：

在 v6.5.0 之前的版本中，该变量用来设置单条查询的内存使用限制。

在 v6.5.0 及之后的版本中，该变量用来设置单个会话整体的内存使用限制，如果某个会话执行过程中使用的内存量超过该阈值，会触发系统变量 `tidb_mem_oom_action` 中指定的行为。

如果变量值为 LOG，那么当一条 SQL 的内存使用超过一定阈值（由 session 变量 tidb_mem_quota_query 控制）后，这条 SQL 会继续执行，但 TiDB 会在 log 文件中打印一条 LOG。

5.查询未走索引的最大SQL：

```
1  SELECT
2          FLOOR(
3                  UNIX_TIMESTAMP(
4                  MIN( summary_begin_time ))) AS agg_begin_time,
5          FLOOR(
```

```
        UNIX_TIMESTAMP(
        MAX( summary_end_time ))) AS agg_end_time,
    ANY_VALUE ( digest_text ) AS agg_digest_text,
    ANY_VALUE ( digest ) AS agg_digest,
    SUM( exec_count ) AS agg_exec_count,
    SUM( sum_latency ) AS agg_sum_latency,
    MAX( max_latency ) AS agg_max_latency,
    MIN( min_latency ) AS agg_min_latency,
    CAST( SUM( exec_count * avg_latency ) / SUM( exec_count ) AS SIGNED ) AS agg_avg_latency,
    CAST( SUM( exec_count * avg_mem ) / SUM( exec_count ) AS SIGNED ) AS agg_avg_mem,
    MAX( max_mem ) AS agg_max_mem,
    ANY_VALUE ( schema_name ) AS agg_schema_name,
    ANY_VALUE ( plan_digest ) AS agg_plan_digest,
    query_sample_text,
    index_names
FROM
    `INFORMATION_SCHEMA`.`CLUSTER_STATEMENTS_SUMMARY_HISTORY`
WHERE
    index_names IS NULL
GROUP BY
    schema_name,
    digest
ORDER BY
    agg_sum_latency DESC
    LIMIT 2;
```

6.tidb集群多个ddl任务出现阻塞问题处理：

```
在TiDB数据库中,可以通过以下步骤取消正在执行的job:1. 查看当前正在执行的job列表:
admin show ddl jobs;
```

该语句会显示正在运行的DDL job列表,包含job ID,job类型,状态等信息。2. 根据job ID取消指定的job:

```

admin cancel ddl jobs <job_id>;
```

将<job_id>替换为要取消的job的ID。例如,如果要取消ID为10的job,命令 would be:

```
admin cancel ddl jobs 10;
```

1. 重新检查job列表,会看到对应的job状态变为“cancelled”:

```
admin show ddl jobs;
```

```
示例输出:
```

```
+------+------------------+---------+--------------------+-----------+
| ID   | Type             | Schema | Table              | State     |
+------+------------------+---------+--------------------+-----------+
| 10   | add index        | test    | t1                 | cancelled |
+------+------------------+---------+--------------------+-----------+
```

2. 如果需要重新执行被取消的job,可以使用 `admin resume ddl jobs <job_id>` 重新启动。这就是在TiDB中取消正在执行的DDL job的步骤。总而言之,主要是:1. 查看当前job列表和状态
3. 使用 `admin cancel ddl jobs <job_id>` 取消指定job
4. 重新检查job列表,确认job状态变为cancelled
5. 如需重新执行,使用 `admin resume ddl jobs <job_id>` 启动job

7.cdc创建任务时报错:

```
Error: [CDC:ErrMetaListDatabases]meta store list databases: [tikv:9006]GC life time is shorter than transaction duration, transaction starts at 2023-05-09 15:00:01 +0800 CST, GC safe point is 2023-05-09 15:34:45.625 +0800 CST
```

解决办法:

```
新生成一个当前时间的Tso，再次创建任务时，指定Tso:

mysql> SELECT conv(concat(bin(unix_timestamp('2023-05-09 15:54:01') * 1000),'0000000000000001'),2,10);
```

```
+-----------------------------------------------------------------------------------------------------------+
| conv(concat(bin(unix_timestamp('2023-05-09 15:54:01') * 1000),'0000000000000001'),2,10) |
+-----------------------------------------------------------------------------------------------------------+
| 441350577455104001                                                                                        |
+-----------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)


[root@wyvunchtdb03 bj_pre_to_test]# tiup cdc cli changefeed create --pd="http://10.3.72.95:2379" --sink-uri="mysql://root:9CMv3k%25a%246wyD%3D5*81@10.3.72.97:4000/" --changefeed-id="bj-pre-to-test-13" --sort-engine="unified" --config="./bj_pre_to_test_13.toml" --start-ts='441350577455104001'
tiup is checking updates for component cdc ...
Starting component `cdc`: /root/.tiup/components/cdc/v6.5.0/cdc cli changefeed create --pd=http://10.3.72.95:2379 --sink-uri=mysql://root:9CMv3k%25a%246wyD%3D5*81@10.3.72.97:4000/ --changefeed-id=bj-pre-to-test-13 --sort-engine=unified --config=./bj_pre_to_test_13.toml --start-ts=441350577455104001
Create changefeed successfully!
ID: bj-pre-to-test-13
Info: {"upstream_id":7189544195128704238,"namespace":"default","id":"bj-pre-to-test-13","sink_uri":"mysql://root:xxxxx@10.3.72.97:4000/","create_time":"2023-05-09T15:54:54.654326795+08:00","start_ts":441350577455104001,"engine":"unified","config":{"case_sensitive":true,"enable_old_value":true,"force_replicate":false,"ignore_ineligible_table":false,"check_gc_safe_point":true,"enable_sync_point":false,"bdr_mode":false,"sync_point_interval":600000000000,"sync_point_retention":8640000000000,"filter":{"rules":["maindb.es_charge_owner_fee_partion_news"],"event_filters":null},"mounter":{"worker_num":16},"sink":{"protocol":"","schema_registry":"","csv":{"delimiter":",","quote":"\"","null":"\\N","include_commit_ts":false},"column_selectors":null,"transaction_atomicity":"none","encoder_concurrency":16,"terminator":"\r\n","date_separator":"none","enable_partition_separator":false},"consistent":{"level":"none","max_log_size":64,"flush_interval":2000,"storage":""}},"state":"normal","creator_version":"v6.5.0"}
```

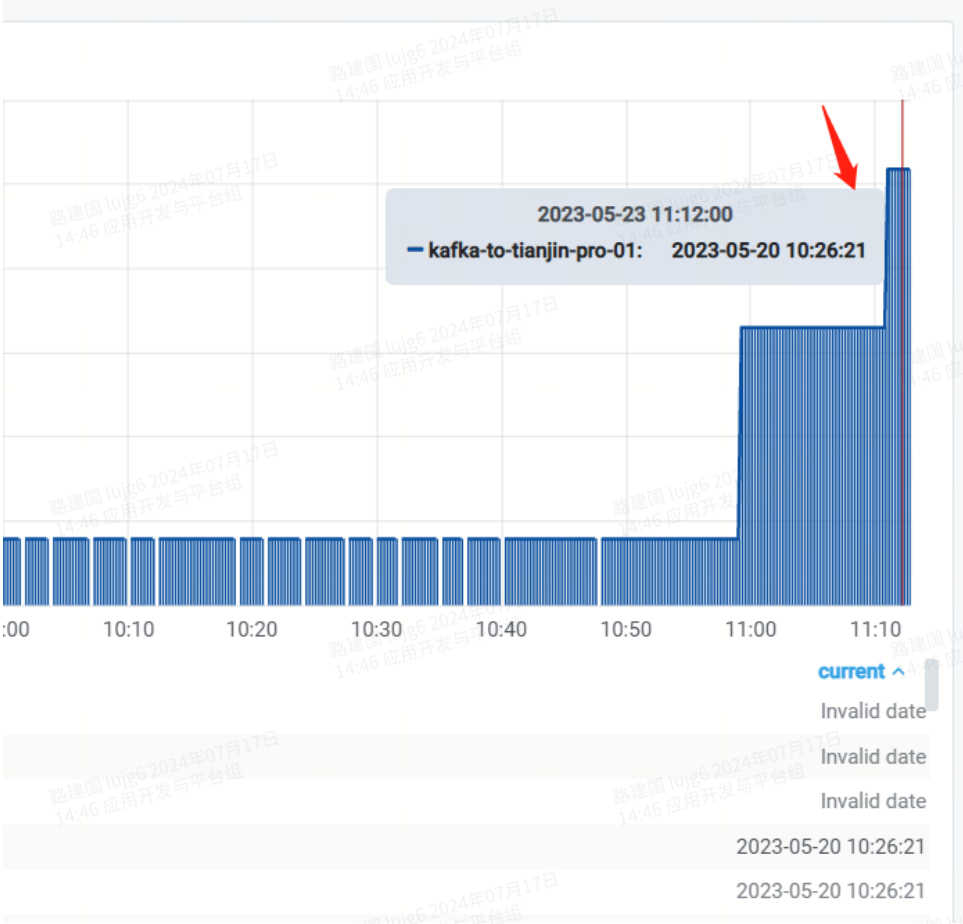
8.cdc的某个kafka任务同步出现延迟卡住问题：

```
1  [2023/05/19 15:25:53.787 +08:00] [ERROR] [feed_state_manager.go:393]
["processor report an error"] [namespace=
2        default] [changefeed=kafka-to-tianjin-pro-05] [captureID=c7922775-9af0-
46cf-8e8f-0cb28e29c0c3] [error="{\"addr
3
\":\"10.3.8.203:8300\",\"code\":\"CDC:ErrKafkaAsyncSendMessage\",\"message\":\"
[CDC:ErrKafkaAsyncSendMessage]k
4        afka async send message failed: kafka: Failed to produce message to
topic charge: write tcp 10.3.8.203:27180-\
5        \u003e172.17.44.108:9092: i/o timeout\"}"]
```

问题原因： cdc服务10.3.8.203:27180到172.17.44.108:9092: i/o timeout，需要排查kakfa侧的问题。

解决办法：重启kafka服务，cdc任务同步正常追踪，问题解决。

```
1 [2023/05/23 11:11:43.792 +08:00] [WARN] [collector.go:96] ["Get Kafka brokers
failed, use historical brokers to collect kafka broker level metrics"]
[namespace=default] [changefeed=kafka-to-tianjin-pro-06] [role=processor]
[duration=48.858µs] [error="write tcp 10.3.8.203:19675->172.17.44.108:9092:
write: broken pipe"]
2 [2023/05/23 11:11:44.083 +08:00] [WARN] [collector.go:96] ["Get Kafka brokers
failed, use historical brokers to collect kafka broker level metrics"]
[namespace=default] [changefeed=kafka-to-tianjin-pro-07] [role=processor]
[duration=121.034µs] [error="write tcp 10.3.8.203:63676->172.17.44.108:9092:
write: broken pipe"]
3 [2023/05/23 11:11:44.324 +08:00] [WARN] [collector.go:96] ["Get Kafka brokers
failed, use historical brokers to collect kafka broker level metrics"]
[namespace=] [changefeed=] [role=owner] [duration=99.472µs] [error="write tcp
10.3.8.203:63552->172.17.44.108:9092: write: broken pipe"]
4 [2023/05/23 11:11:44.930 +08:00] [WARN] [collector.go:96] ["Get Kafka brokers
failed, use historical brokers to collect kafka broker level metrics"]
[namespace=] [changefeed=] [role=owner] [duration=60.778µs] [error="write tcp
10.3.8.203:63680->172.17.44.108:9092: write: broken pipe"]
```

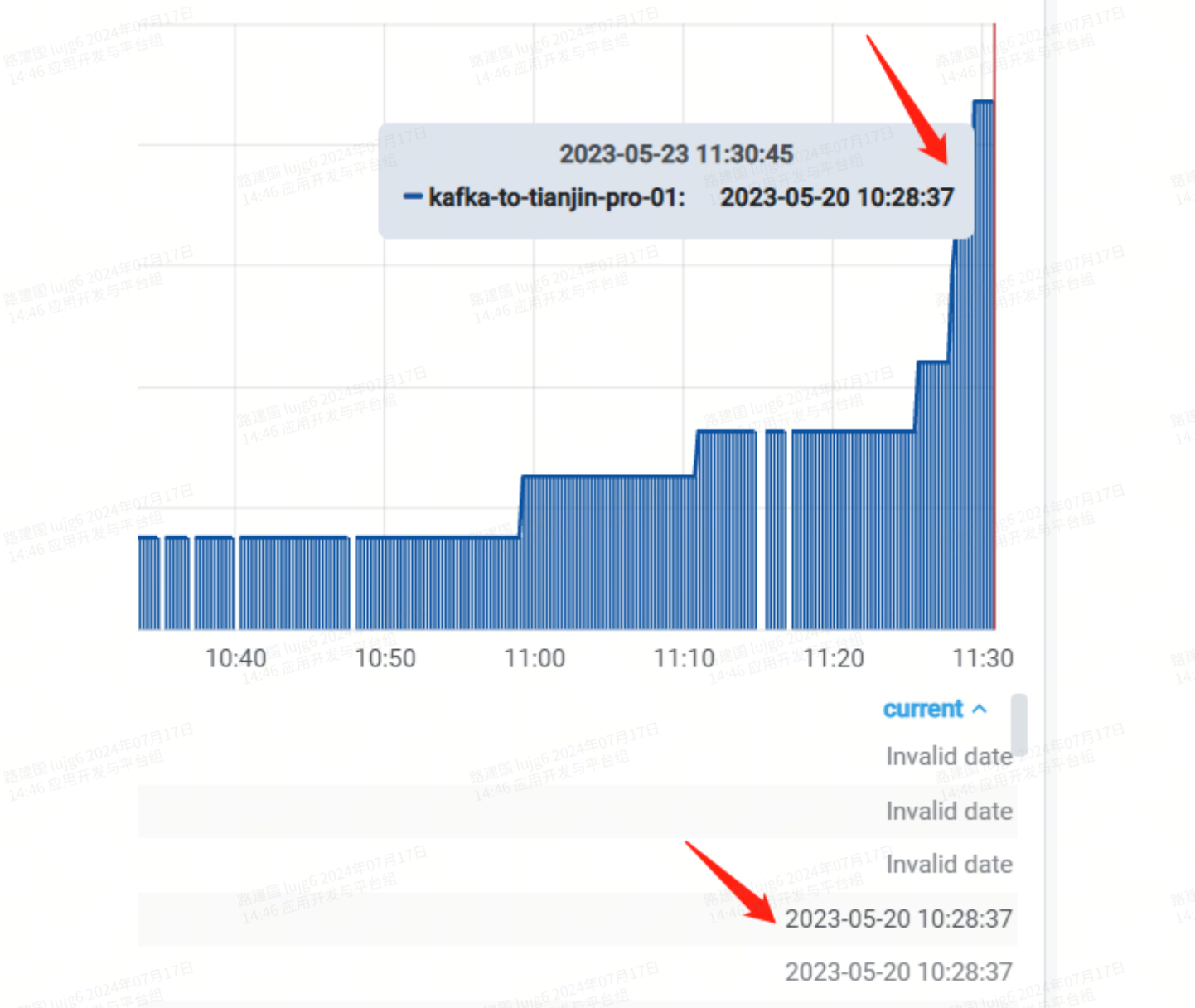


这个出现的原因是：

连接建立后，如果读端或者写端关闭连接，具体分两种情况：

如果读端关闭连接，写端继续写，第一次写，会收到RST，再写，报错broken pipe

如果写端关闭连接，读端继续读，报错EOF



9.tikv节点内存一直没有任何波动，内存使用率不高，想要提高内存使用率，操作如下：

```
show config where Name like '%storage.block-cache.capacity%'; -- 本次调整参数：总内存60%
tikv        10.3.8.226:20163        storage.block-cache.capacity        38GiB
tikv        10.3.8.223:20163        storage.block-cache.capacity        38GiB
tikv        10.3.8.224:20163        storage.block-cache.capacity        38GiB

# tiup cluster edit-config news_sjzt_tidb
  tikv:
    storage.block-cache.capacity: 38912MiB
```

10.在使用tidb-lightning工具导入csv文件到tidb集群时，报错：

```
[2023/08/10 18:13:03.870 +08:00] [ERROR] [restore.go:489] ["the whole procedure failed"] [takeTime=33.650083523s] [error="[Lighting:Restore:ErrChecksumMismatch]checksum mismatched remote vs local => (checksum: 15481151244998829301 vs 14011016464524084535) (total_kvs: 12160985 vs 12161055) (total_bytes:1746760846 vs 1746771416)"]
[2023/08/10 18:13:03.870 +08:00] [ERROR] [restore.go:171] ["tables failed to be imported"] [count=1]
[2023/08/10 18:13:03.870 +08:00] [ERROR] [restore.go:173] [-] [table=`idx`.`zhcx_idx_owe_es_charge_settle_accounts_main`] [status=checksum] [error="[Lighting:Restore:ErrChecksumMismatch]checksum mismatched remote vs local => (checksum: 15481151244998829301 vs 14011016464524084535) (total_kvs: 12160985 vs 12161055) (total_bytes:1746760846 vs 1746771416)"]
[2023/08/10 18:13:03.906 +08:00] [INFO] [checksum.go:459] ["service safe point keeper exited"]
[2023/08/10 18:13:03.906 +08:00] [ERROR] [main.go:103] ["tidb lightning encountered error stack info"] [error="[Lighting:Restore:ErrChecksumMismatch]checksum mismatched remote vs local => (checksum: 15481151244998829301 vs 14011016464524084535) (total_kvs: 12160985 vs 12161055) (total_bytes:1746760846 vs 1746771416)"] [errorVerbose="[Lighting:Restore:ErrChecksumMismatch]checksum mismatched remote vs local => (checksum: 15481151244998829301 vs 14011016464524084535) (total_kvs: 12160985 vs 12161055) (total_bytes:1746760846 vs 1746771416)
```

解决方法：

从源头控制数据导出生成csv时，保证控制数据去重操作，这样，再使用tidb-lightning导入即可。

11.同一套Tidb集群，连接集群中的不同tidb-server，执行同样的SQL语句，发现执行计划不同。

```
SELECT
        f.*
FROM
        es_charge_incoming_fee f
INNER JOIN es_charge_incoming_data d ON d.fld_incoming_fee_guid = f.fld_guid
INNER JOIN es_charge_voucher_project_set s ON s.fld_project_guid = d.fld_project_guid
INNER JOIN es_charge_incoming_data nb ON nb.fld_guid = d.fld_back_fee_guid
LEFT JOIN es_charge_pay_mode pm ON d.fld_pay_mode_guid = pm.fld_guid
LEFT JOIN es_info_object_and_owner DO
```

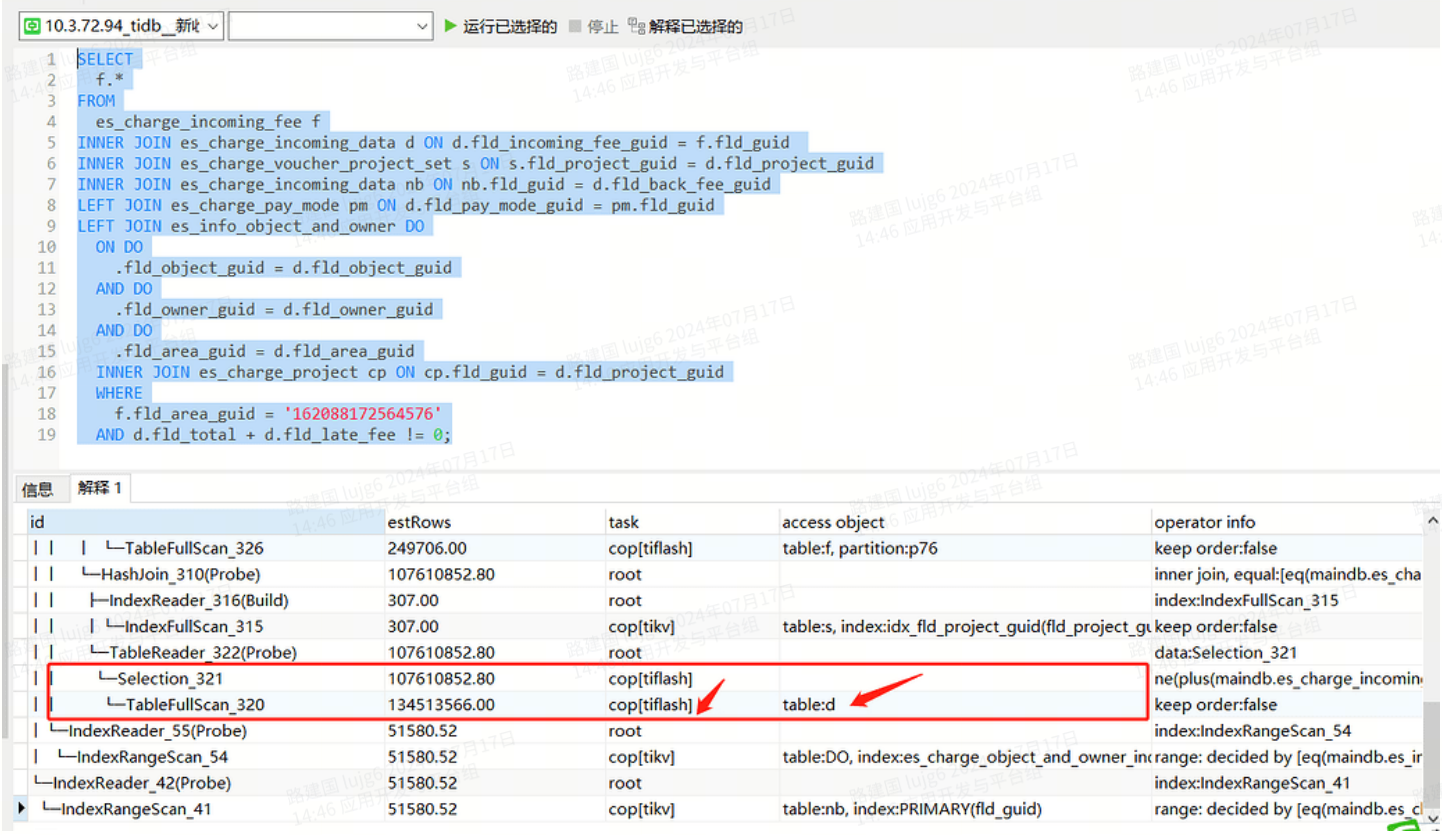
```
10          ON DO
11                  .fld_object_guid = d.fld_object_guid
12          AND DO
13                  .fld_owner_guid = d.fld_owner_guid
14          AND DO
15                  .fld_area_guid = d.fld_area_guid
16          INNER JOIN es_charge_project cp ON cp.fld_guid = d.fld_project_guid
17          WHERE
18                  f.fld_area_guid = '162088172564576'
19          AND d.fld_total + d.fld_late_fee != 0;
```

10.3.72.95_tidb_新 ▶ 运行 ■ 停止 解释

```
1  SELECT
2    f.*
3  FROM
4    es_charge_incoming_fee f
5  INNER JOIN es_charge_incoming_data d ON d.fld_incoming_fee_guid = f.fld_guid
6  INNER JOIN es_charge_voucher_project_set s ON s.fld_project_guid = d.fld_project_guid
7  INNER JOIN es_charge_incoming_data nb ON nb.fld_guid = d.fld_back_fee_guid
8  LEFT JOIN es_charge_pay_mode pm ON d.fld_pay_mode_guid = pm.fld_guid
9  LEFT JOIN es_info_object_and_owner DO
10   ON DO
11     .fld_object_guid = d.fld_object_guid
12   AND DO
13     .fld_owner_guid = d.fld_owner_guid
14   AND DO
15     .fld_area_guid = d.fld_area_guid
16   INNER JOIN es_charge_project cp ON cp.fld_guid = d.fld_project_guid
17   WHERE
18     f.fld_area_guid = '162088172564576'
19   AND d.fld_total + d.fld_late_fee != 0;
```

信息 | 解释 1

id	estRows	task	access object	operator info
├─TableReader_293(Build)	249.71	root		data:Selection_292
│ └─Selection_292	249.71	cop[tiflash]		eq(maindb.es_charge_incoming_fee
│ └─TableFullScan_291	249706.00	cop[tiflash]	table:f, partition:p76	keep order:false
└─IndexLookUp_266(Probe)	1061.22	root		
├─IndexRangeScan_263(Build)	1326.52	cop[tikv]	table:d, index:idx1(fld_incoming_fee_guid, fld_ba	range: decided by [eq(maindb.es_cl
└─Selection_265(Probe)	1061.22	cop[tikv]		ne(plus(maindb.es_charge_incomin
└─TableRowIDScan_264	1326.52	cop[tikv]	table:d	keep order:false
└─IndexReader_55(Probe)	1658.16	root		index:IndexRangeScan_54
└─IndexRangeScan_54	1658.16	cop[tikv]	table:DO, index:es_charge_object_and_owner_in	range: decided by [eq(maindb.es_ir
└─IndexReader_42(Probe)	1658.16	root		index:IndexRangeScan_41
└─IndexRangeScan_41	1658.16	cop[tikv]	table:nb, index:PRIMARY(fld_guid)	range: decided by [eq(maindb.es_cl

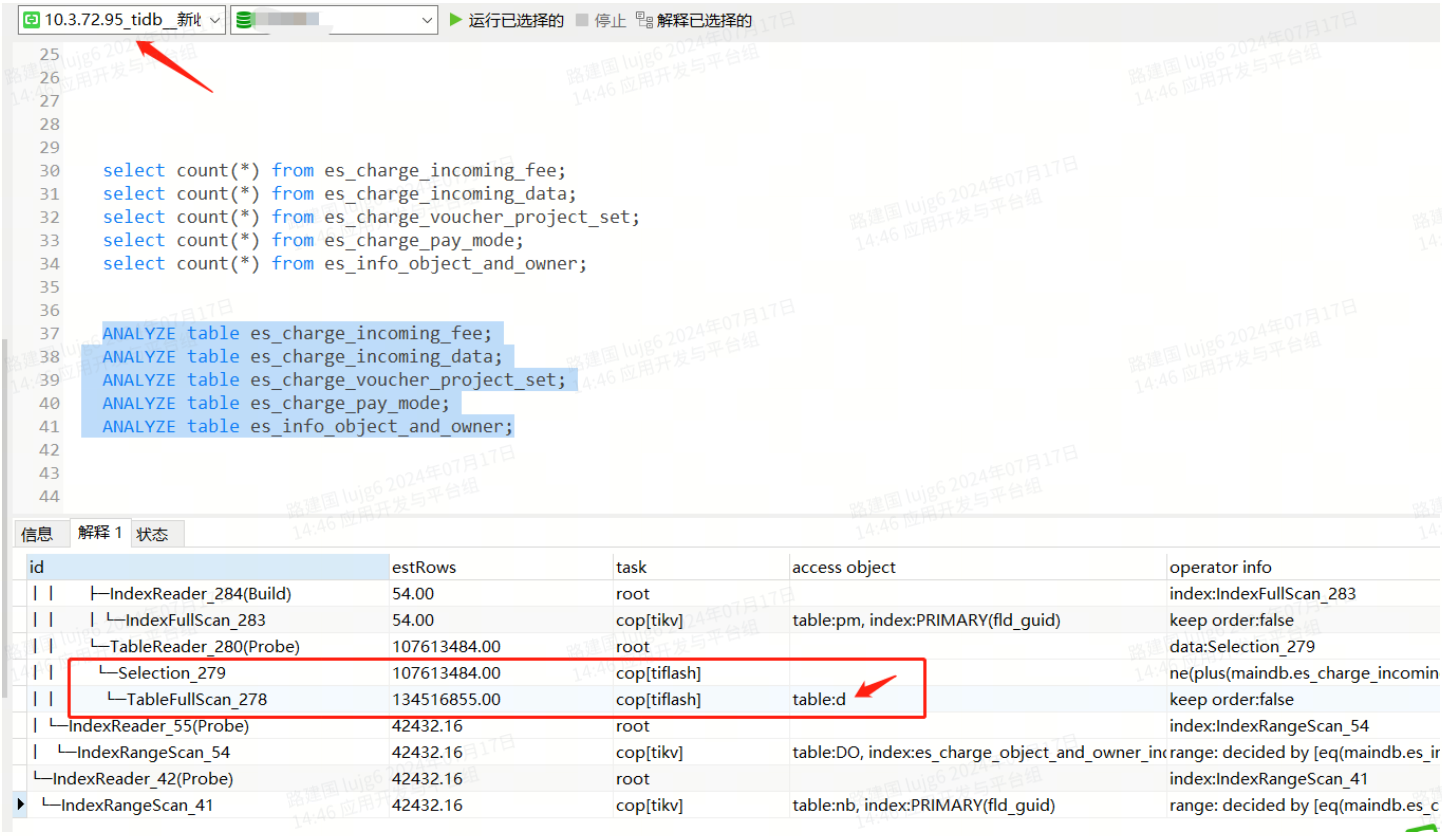


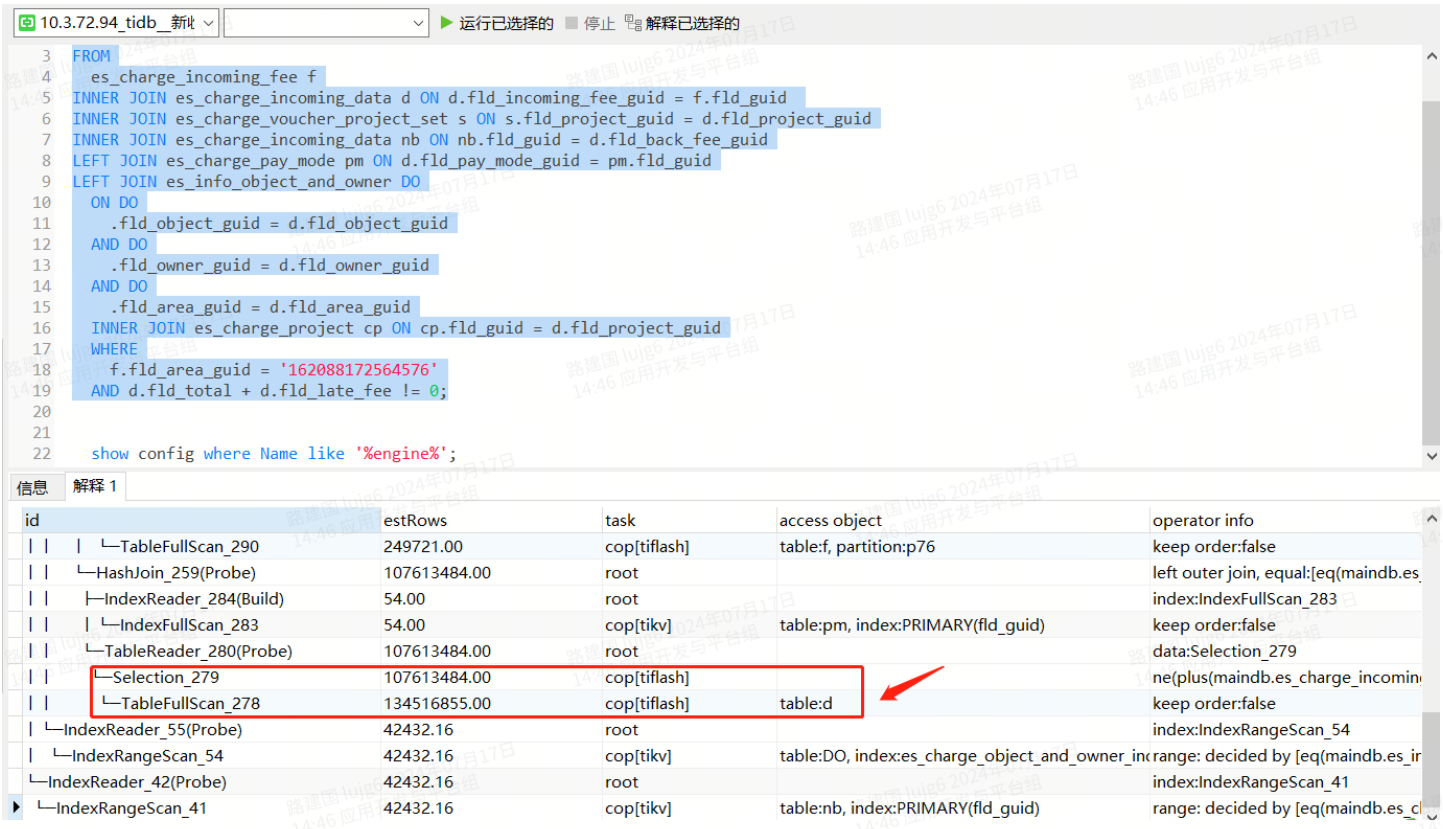
解决方法：对sql中的表进行统计信息收集：

10.3.72.95的tidb-server上，把5张表都执行了下：

```
1  ANALYZE table es_charge_incoming_fee;
2  ANALYZE table es_charge_incoming_data;
3  ANALYZE table es_charge_voucher_project_set;
4  ANALYZE table es_charge_pay_mode;
5  ANALYZE table es_info_object_and_owner;
```

再次查询：





12、开启PITR日志备份报错：

```
Error: failed to check gc safePoint, ts 443883795406651398: GC safepoint
443903291409825792 exceed TS 443883795406651398:
[BR:Backup:ErrBackupGCSafepointExceeded]backup GC safepoint exceeded
```

```
mysql> SELECT TIDB_PARSE_TSO('443883795406651398');
+--------------------------------------+
| TIDB_PARSE_TSO('443883795406651398') |
+--------------------------------------+
| 2023-08-29 12:11:40.593000           |
+--------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT TIDB_PARSE_TSO('443903291409825792');
+--------------------------------------+
| TIDB_PARSE_TSO('443903291409825792') |
+--------------------------------------+
| 2023-08-30 08:51:11.943000           |
+--------------------------------------+
1 row in set (0.01 sec)

mysql>
```

问题原因：

暂停日志备份任务后，备份程序为了防止生成变更日志的 MVCC 数据被 GC，暂停任务程序会自动将当前备份点 checkpoint 设置为 service safepoint，允许保留最近 24 小时内的 MVCC 数据。当超过 24 小时后，备份点 checkpoint 的 MVCC 数据已经被 GC，此时程序会拒绝恢复备份任务。

此场景的处理办法是：

1、先执行 `br log stop` 命令来删除当前的任务，然后执行 `br log start` 重新创建新的日志备份任务，同时做一个全量备份，便于后续做 PITR 恢复操作。

2、或者删除之前的log备份路径下的backup.lock和backupmeta：

```
#  rm -rf /br_backup/backup_8.231_data/backup_data/log_backup/*
```

13.普通用户回收权限后，登录tidb-dashboard报错：

解决办法：TiDB Dashboard 用户管理

```
创建一个最小权限 SQL 用户用于登录 TiDB Dashboard

CREATE USER 'owner_for_sjzt'@'%' IDENTIFIED BY '<YOUR_PASSWORD>';
GRANT PROCESS, CONFIG ON *.* TO 'owner_for_sjzt'@'%';
GRANT SHOW DATABASES ON *.* TO 'owner_for_sjzt'@'%';
GRANT DASHBOARD_CLIENT ON *.* TO 'owner_for_sjzt'@'%';

-- 如果要使自定义的 SQL 用户能修改 TiDB Dashboard 界面上的各项配置，可以增加以下权限
GRANT SYSTEM_VARIABLES_ADMIN ON *.* TO 'dashboardAdmin'@'%';

-- 如果要使用快速绑定执行计划（具体参见 https://docs.pingcap.com/zh/tidb/v7.1/dashboard-statement-details#快速绑定执行计划）功能，可以增加以下权限
GRANT SYSTEM_VARIABLES_ADMIN ON *.* TO 'dashboardAdmin'@'%';
GRANT SUPER ON *.* TO 'dashboardAdmin'@'%';
```

14、时间转换问题：

```
把时间转换成TSO:
mysql> SELECT conv(concat(bin(unix_timestamp('2024-07-12 00:00:01') * 1000),'0000000000000001'),2,10) as tso;
+--------------------+
| tso                |
+--------------------+
| 443229615161344001 |
```

```
+--------------------+
1 row in set (0.01 sec)

把TSO转换为时间：
mysql>  SELECT TIDB_PARSE_TSO('450405353943203983');
+--------------------------------------+
| TIDB_PARSE_TSO('443229615161344001') |
+--------------------------------------+
| 2023-07-31 15:00:01.000000           |
+--------------------------------------+
1 row in set (0.01 sec)
```

15、br做定时任务备份时，BackupTS -- 创建备份时的快照 TSO，会存在比较备份开始时间还要提前的问题

问题原因：pd节点时间不同步，进行备份时，会以pd leader的时间作为BackupTS起始时间。

解决办法：

```
# timedatectl set-ntp yes
# timedatectl status
      Local time: Mon 2023-02-13 16:23:18 CST
  Universal time: Mon 2023-02-13 08:23:18 UTC
        RTC time: Mon 2023-02-13 08:23:18
       Time zone: Asia/Shanghai (CST, +0800)
     NTP enabled: yes
NTP synchronized: no
 RTC in local TZ: no
      DST active: n/a
```

16.提升添加索引效率：

查看 ddl job结果是用的传统方式加的索引，6.5 有快速加索引了 tidb_ddl_enable_fast_reorg 先看下默认值是不是On 。加索引时有没有其他ddl操作:

tidb_ddl_reorg_batch_size

- 作用域：GLOBAL
- 是否持久化到集群：是
- 类型：整数型
- 默认值：`256`
- 范围：`[32, 10240]`
- 单位：行
- 这个变量用来设置 DDL 操作 `re-organize` 阶段的 batch size。比如 `ADD INDEX` 操作，需要回填索引数据，通过并发 `tidb_ddl_reorg_worker_cnt` 个 worker 一起回填数据，每个 worker 以 batch 为单位进行回填。
  - 如果 `ADD INDEX` 操作时有较多 `UPDATE` 操作或者 `REPLACE` 等更新操作，batch size 越大，事务冲突的概率也会越大，此时建议调小 batch size 的值，最小值是 32。
  - 在没有事务冲突的情况下，batch size 可设为较大值（需要参考 worker 数量，见线上负载与 `ADD INDEX` 相互影响测试），这样回填数据的速度更快，但是 TiKV 的写入压力也会变大。

tidb_ddl_reorg_priority

- 作用域：SESSION
- 类型：枚举型
- 默认值：`PRIORITY_LOW`
- 可选值：`PRIORITY_LOW`、`PRIORITY_NORMAL`、`PRIORITY_HIGH`
- 这个变量用来设置 `ADD INDEX` 操作 `re-organize` 阶段的执行优先级，可设置为 `PRIORITY_LOW` / `PRIORITY_NORMAL` / `PRIORITY_HIGH`。

tidb_ddl_reorg_worker_cnt

- 作用域：GLOBAL
- 是否持久化到集群：是
- 类型：整数型
- 默认值：`4`
- 范围：`[1, 256]`
- 单位：线程

```
1 SET @@tidb_ddl_enable_fast_reorg=on
2 SET @@global.tidb_ddl_reorg_batch_size = 1024;
3 SET @@global.tidb_ddl_reorg_worker_cnt = 16;
```

17.br导入报错：

```
1 [root@wtj1nchsync01 tidb_source]# br restore db     --pd "172.17.44.84:2379"
    --db maindb     --storage "local:///acdata/backup/backup_2023-09-05_maindb"
     --ratelimit 512     --log-file restore_maindb_date +%F.log
2 Detail BR log in restore_maindb_2023-09-07.log
3 DataBase Restore <---------------------------------------------------------
  ----------------------------> 100.00%
```

```
4 [2023/09/07 01:02:15.688 +08:00] [INFO] [collector.go:69] ["DataBase Restore
  failed summary"] [total-ranges=37170] [ranges-succeed=37170] [ranges-failed=0]
  [split-region=51.341264834s] [restore-ranges=28271]
5 Error: failed to validate checksum:
  [BR:Restore:ErrRestoreChecksumMismatch]restore checksum mismatch
```

18、tidb-cluster集群版本为：v6.1.5，发现通过tiup cluster edit-config 方式修后，重新reload失败报错：

```
+ Refresh instance configs
  - Generate config pd -> 172.17.44.84:2379 ... Done
  - Generate config pd -> 172.17.44.94:2379 ... Done
  - Generate config pd -> 172.17.44.95:2379 ... Done
  - Generate config tikv -> 172.17.44.85:20160 ... Done
  - Generate config tikv -> 172.17.44.86:20160 ... Done
  - Generate config tikv -> 172.17.44.94:20160 ... Done
  - Generate config tidb -> 172.17.44.84:4000 ... Error
  - Generate config tidb -> 172.17.44.95:4000 ... Error
  - Generate config tiflash -> 172.17.44.129:9000 ... Done
  - Generate config prometheus -> 172.17.44.84:9090 ... Done
  - Generate config grafana -> 172.17.44.84:3000 ... Done
  - Generate config alertmanager -> 172.17.44.84:9093 ... Done

Error: init config failed: 172.17.44.84:4000: executor.ssh.execute_failed: Failed to execute command over SSH for 'tidb@172.
17.44.84:22' {ssh_stderr: , ssh_stdout: [2023/09/11 14:54:21.101 +08:00] [FATAL] [terror.go:292] ["unexpected error"] [error
="config file /acdata/tidb-cluster/tidb-deploy/tidb-4000/conf/tidb.toml contained invalid configuration options: tidb_mem_qu
ota_query; check TiDB manual to make sure this option has not been deprecated and removed from your TiDB version if the opti
on does not appear to be a typo"] [stack="github.com/pingcap/tidb/parser/terror.MustNil\n\t/home/jenkins/agent/workspace/bui
ld-common/go/src/github.com/pingcap/tidb/parser/terror/terror.go:292\ngithub.com/pingcap/tidb/config.InitializeConfig\n\t/ho
me/jenkins/agent/workspace/build-common/go/src/github.com/pingcap/tidb/config/config.go:1009\nmain.main\n\t/home/jenkins/age
nt/workspace/build-common/go/src/github.com/pingcap/tidb/tidb-server/main.go:170\nruntime.main\n\t/usr/local/go/src/runtime/
proc.go:250"] [stack="github.com/pingcap/tidb/parser/terror.MustNil\n\t/home/jenkins/agent/workspace/build-common/go/src/git
hub.com/pingcap/tidb/parser/terror/terror.go:292\ngithub.com/pingcap/tidb/config.InitializeConfig\n\t/home/jenkins/agent/wor
kspace/build-common/go/src/github.com/pingcap/tidb/config/config.go:1009\nmain.main\n\t/home/jenkins/agent/workspace/build-c
ommon/go/src/github.com/pingcap/tidb/tidb-server/main.go:170\nruntime.main\n\t/usr/local/go/src/runtime/proc.go:250"]
, ssh_command: export LANG=C; PATH=$PATH:/bin:/sbin:/usr/bin:/usr/sbin /acdata/tidb-cluster/tidb-deploy/tidb-4000/bin/tidb-s
erver --config-check --config=/acdata/tidb-cluster/tidb-deploy/tidb-4000/conf/tidb.toml }, cause: Process exited with status
 1: check config failed
```

官网写的很清楚6.1这个参数转化为系统变量不在通过配置文件配置，mem-quota-query参数也已经删除了

`tidb_mem_quota_query`

- 作用域：SESSION | GLOBAL
- 是否持久化到集群：是
- 默认值：`1073741824` (1 GiB)
- 范围：`[-1, 9223372036854775807]`
- 单位：字节
- 这个变量用来设置一条查询语句的内存使用阈值。
- 如果一条查询语句执行过程中使用的内存空间超过该阈值，会触发系统变量 `tidb_mem_oom_action` 中指定的行为。
- 在 v6.1.0 之前这个开关通过 TiDB 配置文件（`mem-quota-query`）进行配置，且作用域为 `SESSION`。升级到 v6.1.0 时会自动继承原有设置，作用域变更为 `SESSION | GLOBAL`。

只需要通过set命令修改配置session会话级别和global全局级别即可：

```
mysql> set global tidb_mem_quota_query=32424509440;
```

19.扩容tiflash新节点时，报错：（集群版本：v6.1.5）

```
[2023/09/11 15:54:56.711 +08:00] [WARN] [server.rs:523] ["failed to remove space holder on starting: No such file or directory (os error 2)"]
[2023/09/11 15:54:56.711 +08:00] [INFO] [mod.rs:118] ["encryption: none of key dictionary and file dictionary are found."]
[2023/09/11 15:54:56.711 +08:00] [INFO] [mod.rs:479] ["encryption is disabled."]
[2023/09/11 15:54:56.711 +08:00] [INFO] [server.rs:236] ["set raft-store proxy helper"]
[2023/09/11 15:54:56.711 +08:00] [INFO] [server.rs:236] ["wait for engine-store server to start"]
[2023/09/11 15:55:18.123 +08:00] [INFO] [server.rs:236] ["engine-store server is started"]
[2023/09/11 15:55:19.055 +08:00] [WARN] [config.rs:251] ["Memory allocator nodump is not supported, continue with default allocator"]
```

解决办法：

```
# tiup cluster edit-config sjzt_tidb
修改配置：
server_configs:
```

```
  tiflash-learner:
    storage.reserve-space: 0MB

保存退出，重新reload集群
# tiup cluster reload sjzt_tidb
```

20、tidb只导出业务库的数据库字典

```
# tiup dumpling -h 10.3.8.231 -P 4000 -u root  -p'xxxxxx' -B  bj_sjzt_db -d --filter "bj_sjzt_db.*_alldata"  --filter "bj_sjzt_db.*_data" --filter "bj_sjzt_db.*_summary" -o ./bj_sjzt_db_struct_dict
# cd ./bj_sjzt_db_struct_dict
# cat *.sql > ./bj_sjzt_db_struct_dict_`date +%F`.sql
```

21、加速tidb-server的内存回收

`tidb_server_memory_limit` *从 v6.4.0 版本开始引入*

- 作用域：GLOBAL
- 是否持久化到集群：是
- 默认值：`80%`
- 取值范围：
  - 你可以将该变量值设为百分比格式，表示内存用量占总内存的百分比，取值范围为 `[1%, 99%]`。
  - 你还可以将变量值设为内存大小，取值范围为 `0` 以及 `[536870912, 9223372036854775807]`，单位为 Byte。支持带单位的内存格式 "KB|MB|GB|TB"。`0` 值表示不设内存限制。
  - 当设置的内存值小于 512 MB 且不为 0 时，TiDB 将会使用 512 MB 作为替代。
- 该变量指定 TiDB 实例的内存限制。TiDB 会在内存用量达到该限制时，对当前内存用量最高的 SQL 语句进行取消 (Cancel) 操作。在该 SQL 语句被成功 Cancel 掉后，TiDB 会尝试调用 Golang GC 立刻回收内存，以最快速度缓解内存压力。
- 只有内存使用大于 `tidb_server_memory_limit_sess_min_size` 的 SQL 语句会被选定为最优先被 Cancel 的 SQL 语句。
- 目前 TiDB 一次只能 Cancel 一条 SQL 语句。如果 TiDB 完全 Cancel 掉一条 SQL 语句并回收资源后，内存使用仍然大于该变量所设限制，TiDB 会开始下一次 Cancel 操作。

`tidb_server_memory_limit_gc_trigger` *从 v6.4.0 版本开始引入*

- 作用域：GLOBAL
- 是否持久化到集群：是
- 默认值：`70%`
- 取值范围：`[50%, 99%]`
- TiDB 尝试触发 GC 的阈值。当 TiDB 的内存使用达到 `tidb_server_memory_limit` 值 * `tidb_server_memory_limit_gc_trigger` 值时，则会主动触发一次 Golang GC。在一分钟之内只会主动触发一次 GC。

`tidb_server_memory_limit_sess_min_size` *从 v6.4.0 版本开始引入*

- 作用域：GLOBAL
- 是否持久化到集群：是
- 默认值：`134217728`（即 128 MB）
- 取值范围：`[128, 9223372036854775807]`，单位为 Byte。支持带单位的内存格式“KB|MB|GB|TB”。
- 开启内存限制后，TiDB 会终止当前实例上内存用量最高的 SQL 语句。本变量指定此情况下 SQL 语句被终止的最小内存用量。如果 TiDB 实例的内存超限是由许多内存使用量不明显的会话导致的，可以适当调小该变量值，使得更多会话成为 Cancel 的对象。

```
1
2
3  mysql> SHOW variables LIKE '%tidb_server_memory_limit%';
4  +----------------------------------------+-------------+
5  | Variable_name                          | Value       |
6  +----------------------------------------+-------------+
7  | tidb_server_memory_limit               | 68719476736 |
8  | tidb_server_memory_limit_gc_trigger    | 0.7         |
9  | tidb_server_memory_limit_sess_min_size | 67108864    |
10 +----------------------------------------+-------------+
11 3 rows in set (0.00 sec)
12
```

22.生产环境10.3.8.202的tiflash中断服务报错：

```
[2023/10/26 20:09:06.288 +08:00] [ERROR] [observer.rs:290] ["transfer leader won't exec"] [req="cmd_type: TransferLeader transfer_leader { peer { id: 170926 store_id: 5 } }"] [region="id: 170923 start_key: 7480000000000005FFBA5F728000000003FFCEEFB00000000000FA end_key: 7480000000000005FFBB00000000000000F8 region_epoch { conf_ver: 37 version: 1494 } peers { id: 170924 store_id: 1 } peers { id: 170925 store_id: 4 } peers { id: 170926 store_id: 5 } peers { id: 447403 store_id: 439786 role: Learner } peers { id: 448061 store_id: 439795 role: Learner }"]
```

23、查看tidb集群的配置参数

```
# tiup cluster  show-config  pro-sunac-tidb-newfee
```

24、分区裁剪后在tiflash查询有时很慢，hashagg没有正常在tiflash中开启mpp问题

参考文档：**1.动态分区裁剪**

https://docs.pingcap.com/zh/tidb/stable/partitioned-table#%E5%8A%A8%E6%80%81%E8%A3%81%E5%89%AA%E6%A…

分区表

欢迎来到 PingCAP 文档中心！我们为您提供了丰富的操作指南和详实的参考资料，助您轻松上手 TiDB 产品，顺利完成数据迁移和基于数据库的应用开发等操作。

2.使用MPP模式

https://docs.pingcap.com/zh/tidb/stable/use-tiflash-mpp-mode

使用 MPP 模式

了解如何使用 MPP 模式。

这两个变量所有取值对应的结果如下：

	tidb_allow_mpp=off	tidb_allow_mpp=on（默认）
tidb_enforce_mpp=off（默认）	不使用 MPP 模式。	优化器根据代价估算选择。（默认）
tidb_enforce_mpp=on	不使用 MPP 模式。	TiDB 无视代价估算，选择 MPP 模式。

`tidb_auto_analyze_ratio`

- 作用域：GLOBAL
- 是否持久化到集群：是
- 默认值：`0.5`
- 这个变量用来设置 TiDB 在后台自动执行 `ANALYZE TABLE` 更新统计信息的阈值。`0.5` 指的是当表中超过 50% 的行被修改时，触发自动 ANALYZE 更新。可以指定 `tidb_auto_analyze_start_time` 和 `tidb_auto_analyze_end_time` 来限制自动 ANALYZE 的时间

例如，如果你不想使用 MPP 模式，可以通过以下语句来设置：

```
mysql>  show variables like 'tidb_enable_fast_analyze';
+---------------------------+-------+
| Variable_name             | Value |
+---------------------------+-------+
| tidb_enable_fast_analyze | OFF   |
+---------------------------+-------+
1 row in set (0.00 sec)

mysql> select @@session.tidb_partition_prune_mode;
+-------------------------------------+
| @@session.tidb_partition_prune_mode |
+-------------------------------------+
| static                              |
+-------------------------------------+
1 row in set (0.00 sec)

select @@session.tidb_partition_prune_mode;

SHOW STATS_HEALTHY where healthy <=70 and healthy >=60;
```

```
show variables like 'tidb_enable_fast_analyze';
select @@tidb_partition_prune_mode;
select @@tidb_allow_mpp;
select @@tidb_enforce_mpp;


如果是static，可以SET tidb_partition_prune_mode='dynamic';
先修改当前session成dynamic，重新explain anlzye跑一下这个sql，看下执行计划

先在会话级别执行如下操作：
SET tidb_partition_prune_mode='dynamic';
set tidb_allow_mpp=on;
set tidb_enforce_mpp=on;


-- 全局级别支持分区动态裁剪，自动快速优化，触自动analyze收集表信息数据
SET global tidb_enable_fast_analyze=1;
SET global tidb_partition_prune_mode='dynamic';
set global tidb_allow_mpp=on;
set global tidb_enforce_mpp=off;

-- 生产环境4套集群，收集统计信息配置参数调整(开启自动analyze收集功能，设置每张表数据超过30%数据变更，每天的0点到8点时间段，进行此操作的触自动analyze动作)
set global tidb_enable_auto_analyze=1;
set global tidb_auto_analyze_ratio=0.3;
set global tidb_auto_analyze_start_time='00:00 +0800';
set global tidb_auto_analyze_end_time ='08:00 +0800';




explain
SELECT
  fee.fld_dq,
  fee.fld_ywdy,
  fee.fld_company,
  fee.fld_xm,
  fee.fld_area_name,
  fee.fld_area_guid,
  fee.fld_yt AS fldObjectClassName,
  fee.butler_guid AS fldButlerGuid,
  fee.fld_grid_no,
  fee.fld_grid_name,
  fee.fld_employee_guid,
  fee.fld_employee_name,
  "curMonDebtReceivableParkFee" AS fldKey,
  sum(fee.fld_amount) fldIncomeAmount,
```

```
  NOW() fldUpdateDate,
  "2024-03-27 23:59:59" fldPointDate
FROM
  zhcx_owner_incoming_alldata fee
WHERE
  fee.fld_income <> 1
  AND fee.fld_allot_date >= "1900-01-01 00:00:00"
  AND fee.fld_allot_date <= "2023-12-31 23:59:59"
  AND fee.fld_point_date <= "2024-03-27 23:59:59"
  AND IFNULL(fee.fld_settle_status, 1) != 0
  AND fee.fld_project_type <> 0
  AND fee.fld_project_guid IN ( "bb3144d4-210d-11ec-a455-a4bb6d58a26d",
"5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-210e-11ec-a455-
a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d")
  AND fee.fld_is_owner IN (0, 1)
GROUP BY
  fee.fld_area_guid,
  fldKey,
  fee.butler_guid;

  [arguments: ("curMonDebtReceivableParkFee", "2024-03-27 23:59:59", "1900-01-
01 00:00:00", "2023-12-31 23:59:59", "2024-03-27 23:59:59", "bb3144d4-210d-
11ec-a455-a4bb6d58a26d", "5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-
210e-11ec-a455-a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d", 0, 1)];

+----------------------------------------+------------+-------------------+----
----------------------+---------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------+
| id                                     | estRows    | task              |
access object              | operator info
```

```
                                |
88 +---------------------------------------+------------+-------------------+----
   ----------------------+-----------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   ----------------------------------------------------------------------------------
   -------------+
89 | Projection_208                         | 287.10     | root              |
                          | bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt,
   bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid,
   bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name,
   curMonDebtReceivableParkFee->Column#64, Column#63, 2024-03-27 10:46:58-
   >Column#65, 2024-03-27 23:59:59->Column#66

                |
90 | └─HashAgg_212                         | 287.10     | root              |
                          | group
   by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
```

```
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:sum(Column#70)->Column#63, funcs:firstrow(Column#71)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:firstrow(Column#72)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name, funcs:firstrow(Column#73)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq, funcs:firstrow(Column#74)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy, funcs:firstrow(Column#75)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company, funcs:firstrow(Column#76)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm, funcs:firstrow(Column#77)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt, funcs:firstrow(Column#78)->bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid, funcs:firstrow(Column#79)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no, funcs:firstrow(Column#80)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name, funcs:firstrow(Column#81)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid, funcs:firstrow(Column#82)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name |
91 |   └─PartitionUnion_214                | 287.10     | root             |
                        |
                        |
92 |   ├─HashAgg_228                      | 2.68       | root             |
                        | group by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid, bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:sum(Column#83)->Column#70, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)->Column#71, funcs:firstrow(Column#85)->Column#72, funcs:firstrow(Column#86)->Column#73, funcs:firstrow(Column#87)->Column#74, funcs:firstrow(Column#88)->Column#75, funcs:firstrow(Column#89)->Column#76, funcs:firstrow(Column#90)->Column#77, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)->Column#78, funcs:firstrow(Column#92)->Column#79, funcs:firstrow(Column#93)->Column#80, funcs:firstrow(Column#94)->Column#81, funcs:firstrow(Column#95)->Column#82, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
```

```
>bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
>bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid
                                                                  |
93 |     | └─TableReader_229               | 2.68      | root              |
                        | data:HashAgg_216
                                                                  |
94 |     |   └─HashAgg_216                 | 2.68      | batchCop[tiflash] |
                        | group
by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:sum(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_amount)->Column#83,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name)-
>Column#85, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq)-
>Column#86, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy)-
>Column#87, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company)-
>Column#88, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm)-
>Column#89, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt)-
>Column#90, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no)-
>Column#92,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name)-
>Column#93,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid)-
>Column#94,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name)-
>Column#95                                                         |
95 |     |     └─Selection_227             | 181806.43 | batchCop[tiflash] |
                        |
ge(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 1900-01-01
00:00:00.000000), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_is_owner, 0,
1), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_guid, "bb3144d4-210d-
11ec-a455-a4bb6d58a26d", "5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-
210e-11ec-a455-a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d"),
le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 2023-12-31
23:59:59.000000), le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_point_date,
```

```
2024-03-27 23:59:59.000000),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_income, 1),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_type, 0),
ne(ifnull(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_settle_status, 1), 0)

                                                          |
|      |       └─TableFullScan_226        | 3257850.00 | batchCop[tiflash] |
table:fee, partition:p0  | keep order:false

                                        |
|      ├─HashAgg_252                      | 2.74        | root              |
                        | group
by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:sum(Column#100)-
>Column#70,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
>Column#71, funcs:firstrow(Column#102)->Column#72, funcs:firstrow(Column#103)-
>Column#73, funcs:firstrow(Column#104)->Column#74, funcs:firstrow(Column#105)-
>Column#75, funcs:firstrow(Column#106)->Column#76, funcs:firstrow(Column#107)-
>Column#77, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
>Column#78, funcs:firstrow(Column#109)->Column#79, funcs:firstrow(Column#110)-
>Column#80, funcs:firstrow(Column#111)->Column#81, funcs:firstrow(Column#112)-
>Column#82,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
>bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
>bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid
                                                            |
|      │ └─TableReader_253                | 2.74        | root              |
                        | data:HashAgg_240
```

```
                                  |
|      |    └─HashAgg_240                       | 2.74         | batchCop[tiflash] |
                                  | group
by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:sum(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_amount)->Column#100,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name)-
>Column#102, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq)-
>Column#103, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy)-
>Column#104,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company)-
>Column#105, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm)-
>Column#106, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt)-
>Column#107,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no)-
>Column#109,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name)-
>Column#110,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid)-
>Column#111,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name)-
>Column#112                                                 |
|      |      └─Selection_251                   | 196628.41    | batchCop[tiflash] |
                                  |
ge(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 1900-01-01
00:00:00.000000), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_is_owner, 0,
1), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_guid, "bb3144d4-210d-
11ec-a455-a4bb6d58a26d", "5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-
210e-11ec-a455-a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d"),
le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 2023-12-31
23:59:59.000000), le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_point_date,
2024-03-27 23:59:59.000000),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_income, 1),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_type, 0),
ne(ifnull(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_settle_status, 1), 0)



                                                       |
```

```
|    |        └─TableFullScan_250          | 4810376.00 | batchCop[tiflash] |
table:fee, partition:p1  | keep order:false

                                                                                    |
|      ├─HashAgg_276                       | 1.41       | root              |
                       | group
by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:sum(Column#117)-
>Column#70,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
>Column#71, funcs:firstrow(Column#119)->Column#72, funcs:firstrow(Column#120)-
>Column#73, funcs:firstrow(Column#121)->Column#74, funcs:firstrow(Column#122)-
>Column#75, funcs:firstrow(Column#123)->Column#76, funcs:firstrow(Column#124)-
>Column#77, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
>Column#78, funcs:firstrow(Column#126)->Column#79, funcs:firstrow(Column#127)-
>Column#80, funcs:firstrow(Column#128)->Column#81, funcs:firstrow(Column#129)-
>Column#82,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
>bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
>bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid
                                                             |
|      │ └─TableReader_277                 | 1.41       | root              |
                       | data:HashAgg_264
```

```
                          |
|     |    └─HashAgg_264                     | 1.41       | batchCop[tiflash] |
                          | group
by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:sum(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_amount)->Column#117,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name)-
>Column#119, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq)-
>Column#120, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy)-
>Column#121,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company)-
>Column#122, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm)-
>Column#123, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt)-
>Column#124,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no)-
>Column#126,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name)-
>Column#127,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid)-
>Column#128,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name)-
>Column#129                                                  |
|     |      └─Selection_275                 | 50543.91   | batchCop[tiflash] |
                          |
ge(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 1900-01-01
00:00:00.000000), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_is_owner, 0,
1), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_guid, "bb3144d4-210d-
11ec-a455-a4bb6d58a26d", "5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-
210e-11ec-a455-a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d"),
le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 2023-12-31
23:59:59.000000), le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_point_date,
2024-03-27 23:59:59.000000),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_income, 1),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_type, 0),
ne(ifnull(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_settle_status, 1), 0)



                                             |
|     |        └─TableFullScan_274           | 1179881.00 | batchCop[tiflash] |
table:fee, partition:p2  | keep order:false
```

```
                         |
|     ├─HashAgg_300                       | 2.53         | root            |
                      | group
by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:sum(Column#134)-
>Column#70,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
>Column#71, funcs:firstrow(Column#136)->Column#72, funcs:firstrow(Column#137)-
>Column#73, funcs:firstrow(Column#138)->Column#74, funcs:firstrow(Column#139)-
>Column#75, funcs:firstrow(Column#140)->Column#76, funcs:firstrow(Column#141)-
>Column#77, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
>Column#78, funcs:firstrow(Column#143)->Column#79, funcs:firstrow(Column#144)-
>Column#80, funcs:firstrow(Column#145)->Column#81, funcs:firstrow(Column#146)-
>Column#82,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
>bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
>bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid
                                                          |
|     │ └─TableReader_301                  | 2.53         | root            |
                      | data:HashAgg_288

                      |
|     │   └─HashAgg_288                    | 2.53         | batchCop[tiflash] |
                      | group
by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:sum(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_amount)->Column#134,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name)-
```

```
>Column#136, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq)-
>Column#137, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy)-
>Column#138,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company)-
>Column#139, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm)-
>Column#140, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt)-
>Column#141,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no)-
>Column#143,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name)-
>Column#144,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid)-
>Column#145,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name)-
>Column#146                                          |
110 |     |     └─Selection_299              | 130810.88  | batchCop[tiflash] |
                        |
ge(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 1900-01-01
00:00:00.000000), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_is_owner, 0,
1), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_guid, "bb3144d4-210d-
11ec-a455-a4bb6d58a26d", "5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-
210e-11ec-a455-a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d"),
le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 2023-12-31
23:59:59.000000), le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_point_date,
2024-03-27 23:59:59.000000),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_income, 1),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_type, 0),
ne(ifnull(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_settle_status, 1), 0)
                                                |
111 |     |       └─TableFullScan_298        | 2218896.00 | batchCop[tiflash] |
table:fee, partition:p3  | keep order:false
                        |
```

```
112 |   ├─HashAgg_324                    | 3.85       | root                |
                        | group
    by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:sum(Column#151)-
    >Column#70,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
    >Column#71, funcs:firstrow(Column#153)->Column#72, funcs:firstrow(Column#154)-
    >Column#73, funcs:firstrow(Column#155)->Column#74, funcs:firstrow(Column#156)-
    >Column#75, funcs:firstrow(Column#157)->Column#76, funcs:firstrow(Column#158)-
    >Column#77, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
    >Column#78, funcs:firstrow(Column#160)->Column#79, funcs:firstrow(Column#161)-
    >Column#80, funcs:firstrow(Column#162)->Column#81, funcs:firstrow(Column#163)-
    >Column#82,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid
                                                                        |
113 |   |   └─TableReader_325               | 3.85       | root                |
                        | data:HashAgg_312
                                                                        |
114 |   |     └─HashAgg_312                 | 3.85       | batchCop[tiflash] |
                        | group
    by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
    funcs:sum(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_amount)->Column#151,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name)-
    >Column#153, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq)-
    >Column#154, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy)-
    >Column#155,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company)-
    >Column#156, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm)-
    >Column#157, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt)-
    >Column#158,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no)-
```

```
>Column#160,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name)-
>Column#161,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid)-
>Column#162,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name)-
>Column#163                                                    |
115 |   |     └─Selection_323          | 210444.22  | batchCop[tiflash] |
                        |
ge(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 1900-01-01
00:00:00.000000), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_is_owner, 0,
1), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_guid, "bb3144d4-210d-
11ec-a455-a4bb6d58a26d", "5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-
210e-11ec-a455-a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d"),
le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 2023-12-31
23:59:59.000000), le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_point_date,
2024-03-27 23:59:59.000000),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_income, 1),
ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_type, 0),
ne(ifnull(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_settle_status, 1), 0)
                                                |
116 |     |     └─TableFullScan_322        | 4374306.00 | batchCop[tiflash] |
table:fee, partition:p4  | keep order:false
                                |
117 |     ├─HashAgg_348                   | 3.78       | root              |
                        | group
by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:sum(Column#168)-
>Column#70,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
>Column#71, funcs:firstrow(Column#170)->Column#72, funcs:firstrow(Column#171)-
>Column#73, funcs:firstrow(Column#172)->Column#74, funcs:firstrow(Column#173)-
```

```
>Column#75, funcs:firstrow(Column#174)->Column#76, funcs:firstrow(Column#175)-
>Column#77, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
>Column#78, funcs:firstrow(Column#177)->Column#79, funcs:firstrow(Column#178)-
>Column#80, funcs:firstrow(Column#179)->Column#81, funcs:firstrow(Column#180)-
>Column#82,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
>bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
>bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid
                                                              |
118 |     | └─TableReader_349                | 3.78       | root              |
                    | data:HashAgg_336
                    |
119 |     |   └─HashAgg_336                  | 3.78       | batchCop[tiflash] |
                    | group
by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
funcs:sum(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_amount)->Column#168,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name)-
>Column#170, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq)-
>Column#171, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy)-
>Column#172,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company)-
>Column#173, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm)-
>Column#174, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt)-
>Column#175,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no)-
>Column#177,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name)-
>Column#178,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid)-
>Column#179,
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name)-
>Column#180                                                    |
```

```
120 |   |     └─Selection_347              | 148455.72  | batchCop[tiflash] |
                                                                                |
    ge(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 1900-01-01
    00:00:00.000000), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_is_owner, 0,
    1), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_guid, "bb3144d4-210d-
    11ec-a455-a4bb6d58a26d", "5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-
    210e-11ec-a455-a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d"),
    le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 2023-12-31
    23:59:59.000000), le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_point_date,
    2024-03-27 23:59:59.000000),
    ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_income, 1),
    ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_type, 0),
    ne(ifnull(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_settle_status, 1), 0)
                                                                                |
121 |   |       └─TableFullScan_346        | 2706768.00 | batchCop[tiflash] |
    table:fee, partition:p5  | keep order:false
                                                                                |
122 |   ├─HashAgg_372                      | 3.71       | root              |
                                                                                | group
    by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:sum(Column#185)-
    >Column#70,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
    >Column#71, funcs:firstrow(Column#187)->Column#72, funcs:firstrow(Column#188)-
    >Column#73, funcs:firstrow(Column#189)->Column#74, funcs:firstrow(Column#190)-
    >Column#75, funcs:firstrow(Column#191)->Column#76, funcs:firstrow(Column#192)-
    >Column#77, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
    >Column#78, funcs:firstrow(Column#194)->Column#79, funcs:firstrow(Column#195)-
    >Column#80, funcs:firstrow(Column#196)->Column#81, funcs:firstrow(Column#197)-
    >Column#82,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
```

```
funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)->bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid                                                    |
123 |     | └─TableReader_373                | 3.71       | root              |                  | data:HashAgg_360
124
125
126 mysql> show warnings;
127 +---------+------+-------------------------------------------------------------------------------------------------------------------------------------------------------------+
128 | Level   | Code | Message                                                                                                                                                     |
129 +---------+------+-------------------------------------------------------------------------------------------------------------------------------------------------------------+
130 | Warning | 1105 | disable dynamic pruning due to zhcx_owner_incoming_alldata has no global stats                                                                               |
131 | Warning | 1105 | MPP mode may be blocked because table `zhcx_owner_incoming_alldata`is a partition table which is not supported when `@@tidb_partition_prune_mode=static`. |
132 | Warning | 1105 | MPP mode may be blocked because table `zhcx_owner_incoming_alldata`is a partition table which is not supported when `@@tidb_partition_prune_mode=static`. |
133 | Warning | 1105 | MPP mode may be blocked because table `zhcx_owner_incoming_alldata`is a partition table which is not supported when `@@tidb_partition_prune_mode=static`. |
134 | Warning | 1105 | MPP mode may be blocked because table `zhcx_owner_incoming_alldata`is a partition table which is not supported when `@@tidb_partition_prune_mode=static`. |
135 | Warning | 1105 | MPP mode may be blocked because table `zhcx_owner_incoming_alldata`is a partition table which is not supported when `@@tidb_partition_prune_mode=static`. |
136 | Warning | 1105 | MPP mode may be blocked because table `zhcx_owner_incoming_alldata`is a partition table which is not supported when `@@tidb_partition_prune_mode=static`. |
137 | Warning | 1105 | MPP mode may be blocked because table `zhcx_owner_incoming_alldata`is a partition table which is not supported when `@@tidb_partition_prune_mode=static`. |
138
139
140 找一个业务低峰时间段，对大表进行analyze table zhcx_owner_incoming_alldata;
141
142 再次，查看执行计划。
143
```

```
explain
SELECT
  fee.fld_dq,
  fee.fld_ywdy,
  fee.fld_company,
  fee.fld_xm,
  fee.fld_area_name,
  fee.fld_area_guid,
  fee.fld_yt AS fldObjectClassName,
  fee.butler_guid AS fldButlerGuid,
  fee.fld_grid_no,
  fee.fld_grid_name,
  fee.fld_employee_guid,
  fee.fld_employee_name,
  "curMonDebtReceivableParkFee" AS fldKey,
  sum(fee.fld_amount) fldIncomeAmount,
  NOW() fldUpdateDate,
  "2024-03-27 23:59:59" fldPointDate
FROM
  zhcx_owner_incoming_alldata fee
WHERE
  fee.fld_income <> 1
  AND fee.fld_allot_date >= "1900-01-01 00:00:00"
  AND fee.fld_allot_date <= "2023-12-31 23:59:59"
  AND fee.fld_point_date <= "2024-03-27 23:59:59"
  AND IFNULL(fee.fld_settle_status, 1) != 0
  AND fee.fld_project_type <> 0
  AND fee.fld_project_guid IN ( "bb3144d4-210d-11ec-a455-a4bb6d58a26d",
"5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-210e-11ec-a455-
a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d")
  AND fee.fld_is_owner IN (0, 1)
GROUP BY
  fee.fld_area_guid,
  fldKey,
  fee.butler_guid;

  +----------------------------------+--------------+--------------+-----
----------+-----------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-----------------+
| id                                     | estRows      | task         |
access object | operator info

              |
+----------------------------------------+--------------+--------------+-------
--------+----------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
---------------+
| TableReader_55                         | 1.00         | root         |
partition:all | data:ExchangeSender_54
```

```
                |
182 | └─ExchangeSender_54                   | 1.00         | mpp[tiflash] |
        | ExchangeType: PassThrough


          |
183 |   └─Projection_6                      | 1.00         | mpp[tiflash] |
        | bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt,
    bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name,
    curMonDebtReceivableParkFee->Column#64, Column#63, 2024-03-28 10:38:33-
    >Column#65, 2024-03-27 23:59:59->Column#66


            |
184 |   └─Projection_50                     | 1.00         | mpp[tiflash] |
        | Column#63, bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm,
```

```
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt,
    bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name
                                                                |
185 |       └─HashAgg_51                    | 1.00        | mpp[tiflash] |
        | group by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, funcs:sum(Column#111)-
    >Column#63,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
    funcs:firstrow(Column#113)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name,
    funcs:firstrow(Column#114)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq,
    funcs:firstrow(Column#115)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy,
    funcs:firstrow(Column#116)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company,
    funcs:firstrow(Column#117)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm,
    funcs:firstrow(Column#118)->bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
    funcs:firstrow(Column#120)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no,
    funcs:firstrow(Column#121)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name,
    funcs:firstrow(Column#122)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid,
    funcs:firstrow(Column#123)-
    >bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name |
186 |         └─ExchangeReceiver_53          | 1.00        | mpp[tiflash] |
            |
```

```
        |
187 |           └─ExchangeSender_52          | 1.00         | mpp[tiflash] |
        | ExchangeType: HashPartition, Hash Cols: [name:
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid, collate: binary], [name:
    bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid, collate:
    utf8mb4_general_ci]
                                                                                  |
188 |             └─HashAgg_48                | 1.00         | mpp[tiflash] |
        | group by:bj_sjzt_db.zhcx_owner_incoming_alldata.butler_guid,
    bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_guid,
    funcs:sum(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_amount)->Column#111,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_area_name)-
    >Column#113, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_dq)-
    >Column#114, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_ywdy)-
    >Column#115,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_company)-
    >Column#116, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_xm)-
    >Column#117, funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_yt)-
    >Column#118,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_no)-
    >Column#120,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_grid_name)-
    >Column#121,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_guid)-
    >Column#122,
    funcs:firstrow(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_employee_name)-
    >Column#123                                                                   |
189 |               └─Selection_35            | 6.74         | mpp[tiflash] |
        | ge(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 1900-01-01
    00:00:00.000000), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_is_owner, 0,
    1), in(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_guid, "bb3144d4-210d-
    11ec-a455-a4bb6d58a26d", "5910b216-210e-11ec-a455-a4bb6d58a26d", "23aee470-
    210e-11ec-a455-a4bb6d58a26d", "6f1a4cec-210d-11ec-a455-a4bb6d58a26d"),
    le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_allot_date, 2023-12-31
    23:59:59.000000), le(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_point_date,
    2024-03-27 23:59:59.000000),
```

```
    ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_income, 1),
    ne(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_project_type, 0),
    ne(ifnull(bj_sjzt_db.zhcx_owner_incoming_alldata.fld_settle_status, 1), 0)
                                                  |
190 |               └─TableFullScan_34     | 286019747.00 | mpp[tiflash] |
    table:fee     | keep order:false, PartitionTableScan:true
                  |
191 +---------------------------------------+--------------+--------------+-------
    --------+-----------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ------------------------------------------------------------------------------
    ---------------+
192 10 rows in set (0.00 sec)
193
194
195
196 打开全局模式参数配置:
197
198 SET global tidb_partition_prune_mode='dynamic';
199 set global  tidb_allow_mpp=on;
```

```
200 set global  tidb_enforce_mpp=on;
```

23、慢查询相关参数（可以添加到tidb巡检中）

- tidb_enable_slow_log： 用于控制是否开启 slow log 功能。
- tidb_slow_log_threshold： 设置慢日志的阈值，执行时间超过阈值的 SQL 语句将被记录到慢日志中。默认值是 300 ms。
- tidb_query_log_max_len： 设置慢日志记录 SQL 语句的最大长度。默认值是 4096 byte。
- tidb_redact_log： 设置慢日志记录 SQL 时是否将用户数据脱敏用 ? 代替。默认值是 0 ，即关闭该功能。
- tidb_enable_collect_execution_info： 设置是否记录执行计划中各个算子的物理执行信息，默认值是 1 。

- 生产环境修改调整：

```
1  set global tidb_enable_slow_log=1;
2  set global tidb_slow_log_threshold=1000;
3
4
5  mysql> set global tidb_slow_log_threshold=1000;
6  Query OK, 0 rows affected (0.00 sec)
7
8  mysql> show variables like '%tidb_slow_log_threshold%';
9  +---------------------------+--------+
10 | Variable_name             | Value |
11 +---------------------------+--------+
12 | tidb_slow_log_threshold | 1000  |
13 +---------------------------+--------+
14 1 row in set (0.01 sec)
```

24、手工清理prometheus监控数据后，启动提示成功，实际启动失败问题

通过 TiUP 部署 TiDB 集群的拓扑文件配置

- `storage_retention`： Prometheus 监控数据保留时间，默认 "30d"

```
1  解决办法，先进行缩容，再进行扩容：（历史监控数据会丢失，防止监控数据过大，可以设置保留策略）
2
3  # cat ./scale-out-new-tidb.yaml
```

```
4
5 monitoring_servers:
6 - host: 10.3.8.222
7   ssh_port: 22
8   port: 9090
9   ng_port: 12020
10  deploy_dir: /acdata/tidb-deploy/prometheus-9090
11  data_dir: /acdata/tidb-data/prometheus-9090
12  log_dir: /acdata/tidb-deploy/prometheus-9090/log
13  external_alertmanagers: []
14  storage_retention: "15"
15  arch: amd64
16  os: linux
17
18
19 # tiup cluster scale-in  news_sjzt_tidb --node 10.3.8.222:9090
20 # tiup cluster scale-out  news_sjzt_tidb ./scale-out-new-tidb.yaml --user root
```

25、empty region数量非常大的问题

```
1  -- 查询region的分配大小与数量的关系
2  select db_name,table_name,sum(APPROXIMATE_SIZE)/count(*)   from
   information_schema.TIKV_REGION_STATUS   where TABLE_NAME IS NOT NULL order by
   3 desc;
3
4  -- 查询某张表的某个region_id的状态
5    select
6  region_id,
7  table_id,
8  DB_NAME,
9  table_name
10   from  information_schema.TIKV_REGION_STATUS
11   where table_id='14815'
12   and REGION_ID=16381494;
13
14 -- 查询出现报错的region_id对应的表的数据
15 select
16 -- region_id,
17 table_id,
18 -- DB_NAME,
19 table_name,
20 count(*)
21   from  information_schema.TIKV_REGION_STATUS
22   where REGION_ID in
23   (
```

```
16373822
,16373518
,16373766
,16373814
,16383394
,16374068
,16373682
,18008487
,16383378
,16380622
,16380046
,16377630
,16374240
,16374124
,16373802
,16373654
,16377758
,17970704
,16374502
,17985767
,16381494
,18048017
,18051349
,18045645
,18031829
,18027579
,18026189
,18022247
,18011675
,17988865
,17988747
,17987687
,17987231
,17985179
,17981389
,17980745
,17977629
,17976025
,17975955
,17975009
,17969976
,17954114
,17156500
,17156484
,16384950
,16383386
,16382962
```

```
 71 ,16382818
 72 ,16382730
 73 ,16382610
 74 ,16382510
 75 ,16382410
 76 ,16381954
 77 ,16381770
 78 ,16381566
 79 ,16381562
 80 ,16381554
 81 ,16380946
 82 ,16380750
 83 ,16380694
 84 ,16380514
 85 ,16380458
 86 ,16380150
 87 ,16380094
 88 ,16380022
 89 ,16379874
 90 ,16379302
 91 ,16378838
 92 ,16378490
 93 ,16378074
 94 ,16377362
 95 ,16377062
 96 ,16377006
 97 ,16376118
 98 ,16375982
 99 ,16375774
100 ,16375598
101 ,16375058
102 ,16374806
103 ,16374650
104 ,16374252
105 ,16373968
106 ,16373892
107 ,16373838
108 ,16373830
109 ,16373810
110 ,16373750
111 ,16373738
112 ,16373730
113 ,16373718
114 ,16373706
115 ,16373666
116 ,16373642
117 ,16373638
```

```
,16373630
,16373626
,16373618
,16373570
,16373546
,16373526
,16373770
,16380710
,17156488
,16381778
,17978185
,18051565
,17977763
,17980695
,16374152
,18050309
,17969982
,16382154
,16383698
,16373510
,17983577
,16380734
  )
  group by table_name;

-- region-merge相关的3个参数是:
max-merge-region-keys
max-merge-region-size
split-merge-interval
分别代表:
小于指定大小的region和分裂时间
超过split-merge-interval的才能merge

  show config where Name like '%max-merge-region%' or Name like '%split-merge-
interval';

```

准实时从库上查询:

Type	Instance	Name	Val
pd	10.3.8.232:2379	schedule.max-merge-region-keys	
pd	10.3.8.232:2379	schedule.max-merge-region-size	
pd	10.3.8.232:2379	schedule.split-merge-interval	1h0
pd	10.3.8.231:2379	schedule.max-merge-region-keys	
pd	10.3.8.231:2379	schedule.max-merge-region-size	
pd	10.3.8.231:2379	schedule.split-merge-interval	1h0
pd	10.3.8.230:2379	schedule.max-merge-region-keys	
pd	10.3.8.230:2379	schedule.max-merge-region-size	
pd	10.3.8.230:2379	schedule.split-merge-interval	1h0

主库集群查询：

Type	Instance	Name	Value
pd	10.3.8.244:2379	schedule.max-merge-region-keys	2000
pd	10.3.8.244:2379	schedule.max-merge-region-size	20
pd	10.3.8.244:2379	schedule.split-merge-interval	1h0m
pd	10.3.8.246:2379	schedule.max-merge-region-keys	2000
pd	10.3.8.246:2379	schedule.max-merge-region-size	20
pd	10.3.8.246:2379	schedule.split-merge-interval	1h0m
pd	10.3.8.245:2379	schedule.max-merge-region-keys	2000
pd	10.3.8.245:2379	schedule.max-merge-region-size	20
pd	10.3.8.245:2379	schedule.split-merge-interval	1h0m

使用pd-ctl命令行，修改实时生效：

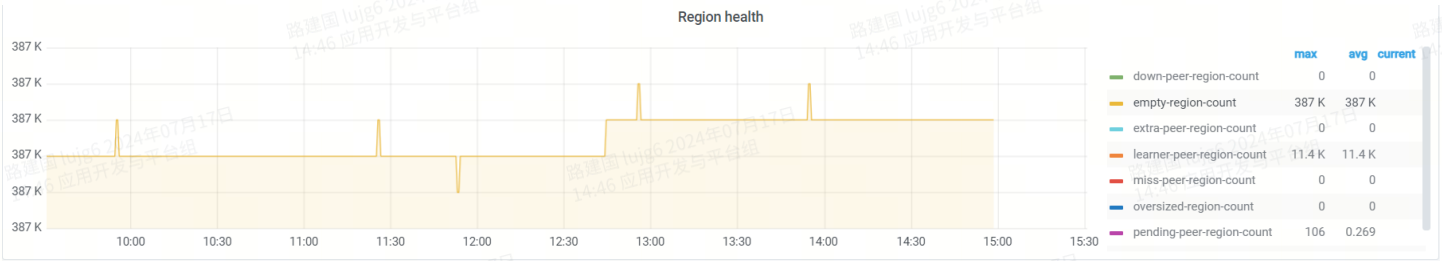
```
[root@wbjdnchzttdb01 ~]# tiup ctl:v6.5.0 pd -u http://10.3.8.231:2379 -i
Starting component `ctl`: /root/.tiup/components/ctl/v6.5.0/ctl pd -u http://10.3.8.231:2379 -i
» config show
{
  "replication": {
    "enable-placement-rules": "true",
    "enable-placement-rules-cache": "false",
    "isolation-level": "",
    "location-labels": "host",
    "max-replicas": 3,
```

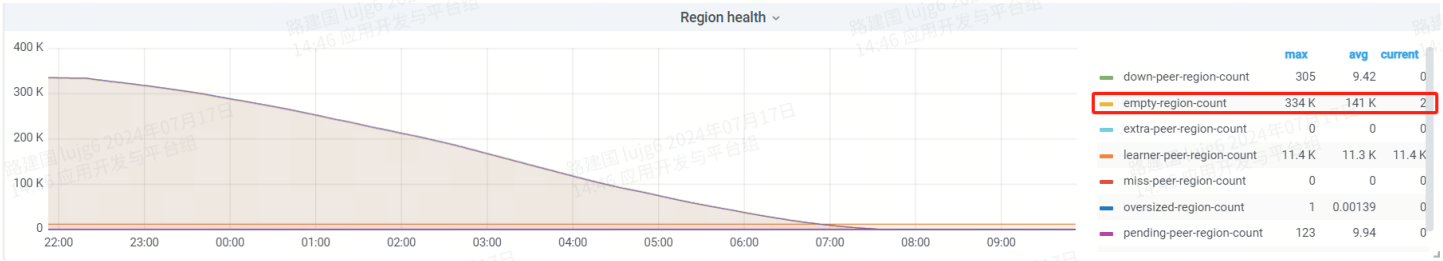
```
    "strictly-match-label": "false"
  },
  "schedule": {
    "enable-cross-table-merge": "true",
    "enable-diagnostic": "false",
    "enable-joint-consensus": "true",
    "enable-tikv-split-region": "true",
    "enable-witness": "false",
    "high-space-ratio": 0.7,
    "hot-region-cache-hits-threshold": 3,
    "hot-region-schedule-limit": 4,
    "hot-regions-reserved-days": 7,
    "hot-regions-write-interval": "10m0s",
    "leader-schedule-limit": 40,
    "leader-schedule-policy": "count",
    "low-space-ratio": 0.8,
    "max-merge-region-keys": 0,
    "max-merge-region-size": 0,
    "max-pending-peer-count": 2147483647,
    "max-snapshot-count": 40,
    "max-store-down-time": "30m0s",
    "max-store-preparing-time": "48h0m0s",
    "merge-schedule-limit": 8,
    "patrol-region-interval": "10ms",
    "region-schedule-limit": 40,
    "region-score-formula-version": "v2",
    "replica-schedule-limit": 64,
    "split-merge-interval": "1h0m0s",
    "swtich-witness-interval": "1h0m0s",
    "tolerant-size-ratio": 0
  }
}

» config set max-merge-region-keys 200000
Success!
» config set max-merge-region-size 20
Success!
» exit

-- 查看是否生效
mysql>   show config where Name like '%max-merge-region%' or Name like '%split-
merge-interval';
+------+-----------------+--------------------------------+--------+
| Type | Instance        | Name                           | Value  |
+------+-----------------+--------------------------------+--------+
| pd   | 10.3.8.232:2379 | schedule.max-merge-region-keys | 200000 |
| pd   | 10.3.8.232:2379 | schedule.max-merge-region-size | 20     |
```

```
| pd   | 10.3.8.232:2379 | schedule.split-merge-interval   | 1h0m0s |
| pd   | 10.3.8.231:2379 | schedule.max-merge-region-keys | 200000 |
| pd   | 10.3.8.231:2379 | schedule.max-merge-region-size | 20     |
| pd   | 10.3.8.231:2379 | schedule.split-merge-interval   | 1h0m0s |
| pd   | 10.3.8.230:2379 | schedule.max-merge-region-keys | 200000 |
| pd   | 10.3.8.230:2379 | schedule.max-merge-region-size | 20     |
| pd   | 10.3.8.230:2379 | schedule.split-merge-interval   | 1h0m0s |
+------+-----------------+---------------------------------+--------+
9 rows in set (0.03 sec)
```





配置前后，empty-region有了很明显的合并效果。

26、tidb集群中node_exporter和blackbox_exporter不跟服务组件启动问题：

```
-- 查看tidb集群中各个节点的node_exporter和blackbox_exporter服务是否开机启动:
# tiup cluster exec bj_sjzt_tidb --sudo  --command "systemctl list-unit-files|grep -E 'node_exporter|blackbox_exporter'"


Outputs of systemctl list-unit-files|grep -E 'node_exporter|blackbox_exporter' on 10.3.8.252:22:
stdout:
blackbox_exporter-9125.service                enabled
node_exporter-9110.service                    enabled
node_exporter.service                         disabled

Outputs of systemctl list-unit-files|grep -E 'node_exporter|blackbox_exporter' on 10.3.8.230:22:
stdout:
blackbox_exporter-9125.service                enabled
node_exporter-9110.service                    enabled
node_exporter.service                         disabled
```

```
16
17 Outputs of systemctl list-unit-files|grep -E 'node_exporter|blackbox_exporter'
   on 10.3.8.231:22:
18 stdout:
19 blackbox_exporter-9125.service                enabled
20 node_exporter-9110.service                    enabled
21 node_exporter.service                         disabled
22
23 Outputs of systemctl list-unit-files|grep -E 'node_exporter|blackbox_exporter'
   on 10.3.8.232:22:
24 stdout:
25 blackbox_exporter-9125.service                enabled
26 node_exporter-9110.service                    enabled
27 node_exporter.service                         disabled
28
29 Outputs of systemctl list-unit-files|grep -E 'node_exporter|blackbox_exporter'
   on 10.3.8.233:22:
30 stdout:
31 blackbox_exporter-9125.service                enabled
32 node_exporter-9110.service                    enabled
33 node_exporter.service                         disabled
34
35 Outputs of systemctl list-unit-files|grep -E 'node_exporter|blackbox_exporter'
   on 10.3.8.234:22:
36 stdout:
37 blackbox_exporter-9125.service                enabled
38 node_exporter-9110.service                    enabled
39 node_exporter.service                         disabled
40
41 Outputs of systemctl list-unit-files|grep -E 'node_exporter|blackbox_exporter'
   on 10.3.8.247:22:
42 stdout:
43 blackbox_exporter-9125.service                enabled
44 node_exporter-9110.service                    enabled
45 node_exporter.service                         disabled
46
47 Outputs of systemctl list-unit-files|grep -E 'node_exporter|blackbox_exporter'
   on 10.3.8.253:22:
48 stdout:
49 blackbox_exporter-9125.service                enabled
50 node_exporter-9110.service                    enabled
51 node_exporter.service                         disabled
52
53
54 # tiup cluster exec bj_sjzt_tidb --sudo  --command "systemctl stop
   node_exporter.service;systemctl disable node_exporter.service;systemctl
```

```
restart node_exporter-9110.service;systemctl restart blackbox_exporter-
9125.service"

-- 重启所有集群里的节点的node_exporter和blackbox服务：
# tiup cluster exec bj_sjzt_tidb --sudo  --command "systemctl restart
node_exporter-9110.service;systemctl restart blackbox_exporter-9125.service"

-- 再次检查服务及端口是否正常启动：
# tiup cluster exec bj_sjzt_tidb --sudo  --command "netstat -atnpu|grep -i -E
'node_expo|blackbox'|grep LISTEN"
Outputs of netstat -atnpu|grep -i -E 'node_expo|blackbox'|grep LISTEN on
10.3.8.230:22:
stdout:
tcp6       0      0 :::9110                 :::*                    LISTEN
 21739/bin/node_expo
tcp6       0      0 :::9125                 :::*                    LISTEN
 21741/bin/blackbox_

Outputs of netstat -atnpu|grep -i -E 'node_expo|blackbox'|grep LISTEN on
10.3.8.231:22:
stdout:
tcp6       0      0 :::9110                 :::*                    LISTEN
 15575/bin/node_expo
tcp6       0      0 :::9125                 :::*                    LISTEN
 15584/bin/blackbox_

Outputs of netstat -atnpu|grep -i -E 'node_expo|blackbox'|grep LISTEN on
10.3.8.232:22:
stdout:
tcp6       0      0 :::9125                 :::*                    LISTEN
 9335/bin/blackbox_e
tcp6       0      0 :::9110                 :::*                    LISTEN
 9331/bin/node_expor

Outputs of netstat -atnpu|grep -i -E 'node_expo|blackbox'|grep LISTEN on
10.3.8.233:22:
stdout:
tcp6       0      0 :::9125                 :::*                    LISTEN
 7437/bin/blackbox_e
tcp6       0      0 :::9110                 :::*                    LISTEN
 7435/bin/node_expor

Outputs of netstat -atnpu|grep -i -E 'node_expo|blackbox'|grep LISTEN on
10.3.8.234:22:
stdout:
tcp6       0      0 :::9110                 :::*                    LISTEN
 25248/bin/node_expo
```

```
tcp6       0      0 :::9125                 :::*                    LISTEN
 25250/bin/blackbox_

Outputs of netstat -atnpu|grep -i -E 'node_expo|blackbox'|grep LISTEN on
10.3.8.247:22:
stdout:
tcp6       0      0 :::9110                 :::*                    LISTEN
 30453/bin/node_expo
tcp6       0      0 :::9125                 :::*                    LISTEN
 30455/bin/blackbox_

Outputs of netstat -atnpu|grep -i -E 'node_expo|blackbox'|grep LISTEN on
10.3.8.253:22:
stdout:
tcp6       0      0 :::9125                 :::*                    LISTEN
 16319/bin/blackbox_
tcp6       0      0 :::9110                 :::*                    LISTEN
 16317/bin/node_expo

Outputs of netstat -atnpu|grep -i -E 'node_expo|blackbox'|grep LISTEN on
10.3.8.252:22:
stdout:
tcp6       0      0 :::9110                 :::*                    LISTEN
 37765/bin/node_expo
tcp6       0      0 :::9125                 :::*                    LISTEN
 37768/bin/blackbox_


-- 北京主库集群:
# tiup cluster exec pro-sunac-tidb-newfee --sudo  --command "systemctl stop
node_exporter.service;systemctl disable node_exporter.service;systemctl daemon-
reload;systemctl restart node_exporter-9100.service;systemctl restart
blackbox_exporter-9115.service"

-- 重启所有集群里的节点的node_exporter和blackbox服务:
# tiup cluster exec pro-sunac-tidb-newfee --sudo  --command "systemctl restart
node_exporter-9100.service;systemctl restart blackbox_exporter-9115.service"
# tiup cluster exec pro-sunac-tidb-newfee --sudo  --command "netstat -
atnpu|grep -i -E 'node_expo|blackbox'|grep LISTEN"
```

27、包含特殊符号的复杂密码进行cdc配置时，需要使用如下工具进行转义：

https://meyerweb.com/eric/tools/dencoder/

URL Decoder/Encoder

URL Decoder/Encoder Input a string of text and encode or decode it as you like. Handy for turning encoded JavaScript URLs from complete gibberish into readable gibberish. If you'd like to have the URL

```
1
```

28、代价模型选择问题

TiDB v6.2.0 引入了新的代价模型 Cost Model Version 2。

Cost Model Version 2 对代价公式进行了更精确的回归校准，调整了部分代价公式，比此前版本的代价公式更加准确。

你可以通过设置变量 `tidb_cost_model_version` 来控制代价模型的版本。

```
-- 集群是从v5.4.0版本升级到v6.5.0版本，默认这个参数还使用的是1，需要修改为2
mysql> show variables like '%tidb_cost_model_version%';
+--------------------------+-------+
| Variable_name            | Value |
+--------------------------+-------+
| tidb_cost_model_version | 1     |
+--------------------------+-------+
1 row in set (0.00 sec)

set global tidb_cost_model_version =2;

-- 修改后，断开当前会话连接，再重新连接，查询此参数是否生效

mysql> show variables like '%tidb_cost_model_version%';
+--------------------------+-------+
| Variable_name            | Value |
+--------------------------+-------+
| tidb_cost_model_version | 2     |
+--------------------------+-------+
1 row in set (0.01 sec)
```

29、查询死锁或事务等待

关于这些系统表的详细说明，请参考以下文档：

- `TIDB_TRX` 与 `CLUSTER_TIDB_TRX`：提供当前 TiDB 节点上或整个集群上所有运行中的事务的信息，包括事务是否处于等锁状态、等锁时间和事务曾经执行过的语句的 Digest 等信息。
- `DATA_LOCK_WAITS`：提供关于 TiKV 内的悲观锁等锁信息，包括阻塞和被阻塞的事务的 `start_ts`、被阻塞的 SQL 语句的 Digest 和发生等待的 key。
- `DEADLOCKS` 与 `CLUSTER_DEADLOCKS`：提供当前 TiDB 节点上或整个集群上最近发生过的若干次死锁的相关信息，包括死锁环中事务之间的等待关系、事务当前正在执行的语句的 Digest 和发生等待的 key。

```
-- 1.查看当前实例下有哪些正在运行的事务
select
  *
from
  INFORMATION_SCHEMA.tidb_trx;

-- 2.查看当前整个集群上所有实例下有哪些正在运行的事务信息
  select
  *
from
  INFORMATION_SCHEMA.cluster_tidb_trx;

-- 3.查看当前实例下有哪些死锁的事务
SELECT * FROM INFORMATION_SCHEMA.DEADLOCKS;

-- 4.查看当前整个集群所有实例下有哪些死锁事务
select * from INFORMATION_SCHEMA.CLUSTER_DEADLOCKS;

```

30、执行analyze table报错：

```
mysql> analyze table es_charge_owner_fee_print_reminder;
ERROR 1317 (70100): Query execution was interrupted

```

`max_execution_time`

- 作用域：SESSION | GLOBAL
- 是否持久化到集群：是
- 默认值：`0`

- 范围：`[0, 2147483647]`
- 单位：毫秒
- 语句最长执行时间。默认值 (0) 表示无限制。

注意

- `max_execution_time` 目前只用于控制只读语句的最大执行时长，实际精度在 100ms 级别，而非更准确的毫秒级别。
- 对于使用了 `MAX_EXECUTION_TIME` Hint 的 SQL 语句，这些语句的最长执行时间将不受该变量限制，而是由该 Hint 进行限制。你也可以使用该 Hint 来创建 SQL 绑定，详情请参考 SQL 操作常见问题。

问题原因：设置了会话的最大执行超时时间，只需要在会话级别临时修改为0。

官方说明：`max_execution_time` 目前对所有类型的语句生效，并非只对 `SELECT` 语句生效，与 MySQL 不同（只对 `SELECT` 语句生效）。实际精度在 100ms 级别，而非更准确的毫秒级别。

```
mysql> show variables like 'max_execution_time';
+--------------------+--------+
| Variable_name      | Value  |
+--------------------+--------+
| max_execution_time | 600000 |
+--------------------+--------+
1 row in set (0.00 sec)

mysql> set max_execution_time=0;
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'max_execution_time';
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| max_execution_time | 0     |
+--------------------+-------+
1 row in set (0.00 sec)

mysql> analyze table es_charge_owner_fee_print_reminder;
```

31、研发的一些定时任务运行时，报错：

```
报错:
Cause: java.sql.SQLException: other error for mpp stream: DB::Exception: Exchange receiver meet error : DB::Exception: Region epoch not match after retries: Region {159494,7,10032} not in region cache. ; uncategorized SQLException; SQL state [HY000]; error code [1105]; other error for mpp stream: DB::Exception: Exchange receiver meet error : DB::Exception: Region epoch not match after retries: Region {159494,7,10032} not in region cache.; nested exception is java.sql.SQLException: other error for mpp stream: DB::Exception: Exchange receiver meet error : DB::Exception: Region epoch not match after retries: Region {159494,7,10032} not in region cache. at
```

使用pd-ctl命令查看region:

```
[root@wtj7vpnchzctdb01 tidb-community-server-v7.5.0-linux-amd64]#  ./pd-ctl region 159494
null

[root@wtj7vpnchzctdb01 tidb-community-server-v7.5.0-linux-amd64]#  ./pd-ctl region 7
null

[root@wtj7vpnchzctdb01 tidb-community-server-v7.5.0-linux-amd64]#  #  ./pd-ctl -u http://10.3.9.9:2379 region 10032
{
  "id": 10032,
  "start_key": "7480000000000001FF1C5F698000000000FF0000020400009373FF4E97DEFF010044000FF3500450037FF0038FF004400360033FF00FF42003600340032FFFF00310045003700300FFFF00300039003100FF41FF003600350045FF0031FF0034003100FF420046FF00390036FF00390038FF000000FF0000000000F70380FF000000422B557000FE",
  "end_key": "7480000000000001FF1C5F698000000000FF0000030400009373FF0480EA5F01003200FF4600410034FF0043FF004300360046FF00FF2D003200310031FFFF0031002D00310031FFFF00450043002D00FF41FF003400350035FF002DFF0041003400FF420042FF00360044FF00350038FF004100FF3200360044FF0000FF000000000000F703FF800000000347AA74FF0000000000000000F7",
  "epoch": {
    "conf_ver": 5,
    "version": 297
  },
  "peers": [
    {
      "role_name": "Voter",
      "id": 10033,
      "store_id": 1
    },
```

```
    {
      "role_name": "Voter",
      "id": 10034,
      "store_id": 4
    },
    {
      "role_name": "Voter",
      "id": 10035,
      "store_id": 5
    }
  ],
  "leader": {
    "role_name": "Voter",
    "id": 10034,
    "store_id": 4
  },
  "cpu_usage": 0,
  "written_bytes": 0,
  "read_bytes": 0,
  "written_keys": 0,
  "read_keys": 0,
  "approximate_size": 308,
  "approximate_keys": 1597993
}

```

二、Mysql数据库相关问题

1. Archery审核平台日常工单问题

常规运维任务：

```
查询新收费每个月度运维上线工单数据，按照天分组统计
SELECT
t1.group_name,
substr(t1.create_time,1,10) cr_time,
COUNT(*) g_num
FROM `sql_workflow` t1
WHERE t1.`create_time`>='2023-08-01 00:00:01'
AND  t1.`create_time`<='2023-08-31 00:00:00'
AND t1.`group_name` NOT IN
(
'企业管理组',
```

```
'生活服务组',
'智慧社区组',
'基础平台组',
'住宅组',
'dba数据组',
'项目内测组'
)
and t1.group_name='数科-企业管理组-生产环境-新收费'
GROUP BY t1.`group_name`,cr_time
ORDER BY 2;

2.其他日常任务处理
SELECT
-- t1.group_name,
substr(t1.create_time,1,10) cr_time,
COUNT(*) g_num
FROM `sql_workflow` t1
WHERE t1.`create_time`>='2023-08-01 00:00:01'
AND  t1.`create_time`<='2023-08-31 00:00:00'
AND t1.`group_name` NOT IN
(
'企业管理组',
'生活服务组',
'智慧社区组',
'基础平台组',
'住宅组',
'dba数据组',
'项目内测组'
)
and t1.group_name!='数科-企业管理组-生产环境-新收费'
GROUP BY
-- t1.`group_name`
cr_time
ORDER BY 2;
```

2. Archery审核平台慢日志TOP10统计

```
-- 统计已经接入到archery审核平台的实例慢日志top10，按照fingerprint分组，取最近1周7天内，
-- 查询次数大于等于5，平均耗时大于等于2s,对平均耗时进行排名从1到10，取出每个分组前10条记录。
SELECT
CASE  t.`hostname_max`
WHEN '10.3.0.175:3306' THEN '物业-缴费'
WHEN '10.3.8.138:3306' THEN '物业-主数据'
```

```
WHEN '10.3.8.240:3306' THEN '物业-工单'
WHEN '10.3.8.65:3306' THEN  '物业-电商'
WHEN '10.3.8.112:3306' THEN  '物业-计划运营'
WHEN '10.3.8.113:3306' THEN  '物业-计划运营'
WHEN '10.3.8.114:3306' THEN  '物业-计划运营'
WHEN '10.3.8.185:3306' THEN  '物业-臻心'
WHEN '10.3.8.186:3306' THEN  '物业-臻心'
WHEN '10.3.8.187:3306' THEN  '物业-臻心'
WHEN '172.17.44.44:3306' THEN  '物业-中台报表'
WHEN '10.3.8.160:3306' THEN  '物业-集采电商'

ELSE t.`hostname_max`
END AS '业务名称',
t.`hostname_max` '源库ip地址',
t.user_max '业务账号',
CASE
WHEN t.`db_max` IS NULL AND t.user_max ='xcx_server' THEN 'sunac_xcx'
WHEN t.`db_max` IS NULL AND t.user_max ='master_data' THEN 'sunacwy_mdm'
WHEN t.`db_max` IS NULL AND t.user_max ='fanr_reader' THEN 'plan_exec'
WHEN t.`db_max` IS NULL AND t.user_max ='plan' THEN 'plan_exec'
WHEN t.`db_max` IS NULL AND t.user_max ='zx_task' THEN 'zx_task'

-- WHEN 'xcx_server' THEN 'sunac_xcx'
ELSE t.`db_max`
END '数据库名称',
t.`sample` '样本SQL',
t.ts_time '统计时间',
t.ts_cnt 'SQL数量',
t.Query_time_sum '最大耗时',
t.`Query_time_max` '最大耗时',
t.`Query_time_median` '平均耗时',
t.Top10
FROM
  (SELECT
    t1.`hostname_max`,
    t1.user_max,
    t1.`db_max`,
    t1.`sample`,
    -- t2.`fingerprint`,
     SUBSTRING(t1.`ts_min`, 1, 10) 'ts_time',
    t1.ts_cnt,
    t1.Query_time_sum,
    t1.`Query_time_max`,
    t1.`Query_time_median`,
    ROW_NUMBER () OVER (
      PARTITION BY hostname_max
  ORDER BY t1.`Query_time_median` DESC
```

```
  ) AS top10
  FROM
    `mysql_slow_query_review_history` t1
    INNER JOIN `mysql_slow_query_review` t2
      ON t1.`checksum` = t2.`checksum`
  WHERE t1.`user_max` NOT IN ('dba_manager')
    AND t1.`ts_min` >= '2023-03-27 00:00:01'
    AND t1.`ts_min` <= '2023-03-31 00:00:01'
    AND t1.ts_cnt >= 2
    AND t1.`Query_time_median` >= 2
   GROUP BY t2.`fingerprint`
  -- ORDER BY t1.`ts_cnt` DESC,t1.`Query_time_median` DESC
  -- order by t1.`Query_time_median` DESC
  ) t
WHERE t.top10 <= 10;


-- 按照查询总耗时排序
SELECT
CASE  t.`hostname_max`
WHEN '10.3.0.175:3306' THEN '物业-缴费'
WHEN '10.3.8.138:3306' THEN '物业-主数据'
WHEN '10.3.8.240:3306' THEN '物业-工单'
WHEN '10.3.8.65:3306' THEN  '物业-电商'
WHEN '10.3.8.112:3306' THEN  '物业-计划运营'
WHEN '10.3.8.113:3306' THEN  '物业-计划运营'
WHEN '10.3.8.114:3306' THEN  '物业-计划运营'
WHEN '10.3.8.185:3306' THEN  '物业-臻心'
WHEN '10.3.8.186:3306' THEN  '物业-臻心'
WHEN '10.3.8.187:3306' THEN  '物业-臻心'
WHEN '172.17.44.44:3306' THEN  '物业-中台报表'
WHEN '10.3.8.160:3306' THEN  '物业-集采电商'
ELSE t.`hostname_max`
END AS '业务名称',
t.`hostname_max` '源库ip地址',
t.user_max '业务账号',
CASE
WHEN t.`db_max` IS NULL AND t.user_max ='xcx_server' THEN 'sunac_xcx'
WHEN t.`db_max` IS NULL AND t.user_max ='master_data' THEN 'sunacwy_mdm'
WHEN t.`db_max` IS NULL AND t.user_max ='fanr_reader' THEN 'plan_exec'
WHEN t.`db_max` IS NULL AND t.user_max ='plan' THEN 'plan_exec'
WHEN t.`db_max` IS NULL AND t.user_max ='zx_task' THEN 'zx_task'

-- WHEN 'xcx_server' THEN 'sunac_xcx'
ELSE t.`db_max`
END '数据库名称',
t.`sample` '样本SQL',
```

```
t.ts_time '统计时间',
t.ts_cnt 'SQL数量',
t.Query_time_sum 'sum最大耗时',
t.`Query_time_median` '平均耗时',
t.Top10
FROM
  (SELECT
    t1.`hostname_max`,
    t1.user_max,
    t1.`db_max`,
    t1.`sample`,
    -- t2.`fingerprint`,
     SUBSTRING(t1.`ts_min`, 1, 10) 'ts_time',
    t1.ts_cnt,
    t1.Query_time_sum,
    t1.`Query_time_max`,
    t1.`Query_time_median`,
    ROW_NUMBER () OVER (
      PARTITION BY hostname_max
  -- order by t1.`Query_time_median` DESC
  ORDER BY t1.`Query_time_sum` DESC
  ) AS top10
  FROM
    `mysql_slow_query_review_history` t1
    INNER JOIN `mysql_slow_query_review` t2
      ON t1.`checksum` = t2.`checksum`
  WHERE 1=1
    AND t1.`user_max` NOT IN ('dba_manager')
    AND t1.`ts_min` >= '2023-03-27 00:00:01'
    AND t1.`ts_min` <= '2023-03-31 00:00:01'
    AND t1.ts_cnt >= 2
    AND t1.`Query_time_median` >= 2
   GROUP BY t2.`fingerprint`
  -- ORDER BY t1.`ts_cnt` DESC,t1.`Query_time_median` DESC
  -- order by t1.`Query_time_median` DESC
  ) t
WHERE t.top10 <= 10;
```

3.mysql慢日志时间与系统时间相差8小时问题的解决

```
这是由于log_timestamps这个参数设置造成的，查询当前设置
mysql> show variables like '%log_time%';
+----------------+-------+
| Variable_name  | Value |
+----------------+-------+
```

```
| log_timestamps | UTC    |
+----------------+-------+
1 row in set (0.01 sec)
UTC是世界统一时间，而现在的系统为北京时间是东八区，比UTC早了8个小时，所以这里设置为SYSTEM

mysql> set global log_timestamps=SYSTEM;
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like '%log_time%';
+----------------+--------+
| Variable_name  | Value  |
+----------------+--------+
| log_timestamps | SYSTEM |
+----------------+--------+
1 row in set (0.01 sec)

设置这个参数后，archery审核平台接入慢日志也可以正常读取和展示了。
```

4.工单业务程序端日志报错：

```
2023-06-12 12:29:56 routing INFO [7f2f447a5700] [routing:balancing] fd=76 Pre-auth socket failure 10.4.8.38:57220: client auth timed out
2023-06-12 12:29:56 routing INFO [7f2f447a5700] [routing:balancing] 1 connection errors for 10.4.8.38:57220 (max 100)
2023-06-12 12:30:47 routing INFO [7f2f447a5700] [routing:balancing] resetting connection error counter for 10.4.8.38:57358 from 1 back to 0
2023-06-14 16:00:58 routing INFO [7f2ea45c5700] [routing:balancing] fd=196 Pre-auth socket failure 10.4.8.37:33392: client auth timed out
2023-06-14 16:00:58 routing INFO [7f2ea45c5700] [routing:balancing] 1 connection errors for 10.4.8.37:33392 (max 100)
2023-06-14 16:01:48 routing INFO [7f2ea45c5700] [routing:balancing] fd=196 Pre-auth socket failure 10.4.8.37:33838: client auth timed out
```

处理办法：

```
在/etc/my.cnf配置文件中，添加如下:
max_connect_errors = 10000
```

5. insert数据时，报错如下：

```
ERROR 3100 (HY000): Error on observer while running replication hook 'before_commit'.
```

解决方法：

```
该错误出现原因：
事务大小超过数据库默认大小。
show global variables like '%group_replication_transaction_size_limit%';
一般大小为141M(150000000)，修改为2倍
set global group_replication_transaction_size_limit=300000000;
```

6.主从同步报错：

```
      Seconds_Behind_Master: NULL
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 1317
               Last_SQL_Error: Could not execute Delete_rows event on table order_admin.adm_node_operate_t; Query execution was interrupted, Error_code: 1317; Got error 168 - 'Unknown (generic) error from engine' from storage engine, Error_code: 1030; handler error HA_ERR_GENERIC; the event's master log mysql-bin.000990, end_log_pos 1135271749
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 1
                  Master_UUID: 56deef9f-ca40-11ea-a5ed-005056b55198
             Master_Info_File: mysql.slave_master_info
```

解决办法：

1.根据报错,找到主库上对应binlog日志文件

2.使用mysqlbinlog和split命令对日志文件进行解析和分割

3.使用grep过滤定位pos位的事务id

4.设置跳过

6、myloader将tidb或mysql的数据导入到mysql：

```
# myloader -h 172.17.44.67 -u root -p 'xxxx' -P 3307 -d /acdata/backup/tj_maindb_03_tabs_2023-09-19_bk/ -t 8 -v 3
```

7、mysqlbinlog使用和split切分文件且保持完整性

```
# mysqlbinlog -vv --base64-output=decode-rows mysql-bin.000804 > ./mysql-bin.000804.row.log
[root@wyvpdsppmysql01 dba_tools]# split -C 512m -d -a 3 ./mysql-bin.000804.row.log  mysql-bin.000804.row.log_split_
[root@wyvpdsppmysql01 dba_tools]# ll mysql-bin.000804.row.log*
-rw-r--r--. 1 root root 2617129755 Mar 26 10:10 mysql-bin.000804.row.log
-rw-r--r--. 1 root root  536870890 Mar 26 10:16 mysql-bin.000804.row.log_split_000
-rw-r--r--. 1 root root  536870876 Mar 26 10:16 mysql-bin.000804.row.log_split_001
-rw-r--r--. 1 root root  536870894 Mar 26 10:16 mysql-bin.000804.row.log_split_002
-rw-r--r--. 1 root root  536870836 Mar 26 10:16 mysql-bin.000804.row.log_split_003
-rw-r--r--. 1 root root  469646259 Mar 26 10:16 mysql-bin.000804.row.log_split_004
```

8、maridb同时导出多个schema下的所有表结构

```
# mysqldump -h 10.3.8.161 -u dba_manager -p'123456' -P 3306  --no-data --databases jicai_backend jicai_distribution jicai_goods jicai_member jicai_sss jicai_system jicai_trade   > ./jicai_7schemas_struct_1107.bk.sql

--no-data：这个选项告诉mysqldump只导出表结构，不导出数据。
--databases：指定要导出的数据库schema的名称，用空格分隔，例如 schema1 和 schema2。替换为你实际要导出的schema名称。
```

9、mysql同时导出多个schema下的所有表结构

```
# mysqldump -h rm-2zea9q835h87a503m.mysql.rds.aliyuncs.com -u dba_manager -
```

```
p'123456' -P 3306  -B -d finance_base finance_business finance_inventory
finance_invoice finance_payment finance_settlement --set-gtid-purged=OFF >
./finance_5schemas_struct_1107.bk.sql
2
3
```

10、阿里云RDS使用mysqldump导出数据报错：

```
mysqldump: Couldn't execute 'SELECT COLUMN_NAME,
JSON_EXTRACT(HISTOGRAM, '$."number-of-buckets-specified"')                FROM
information_schema.COLUMN_STATISTICS                WHERE SCHEMA_NAME =
'sunac_smart_statistics' AND TABLE_NAME = 'eba_cockpit_attr';': Unknown table
'column_statistics' in information_schema (1109)
```

解决方法：（添加参数--column-statistics=0）

```
# mysqldump -h rm-2zep742ur502gnplm.mysql.rds.aliyuncs.com -u dba_manager -
p'123456' -P 3306 --no-data --databases sunac_smart_statistics sunac_eba_msj
sunac_nacos --set-gtid-purged=OFF --column-statistics=0 >
./zhsq_eba_3schemas_struct_1109.bk.sql
```

```
使用mysqldump导出数据时，要求不锁表：(添加参数：--single-transaction)
# mysqldump  -h rm-2zec1ou9m3urxzt14.mysql.rds.aliyuncs.com -u dba_manager -
p'xxxxx' -P 3306 --set-gtid-purged=OFF --single-transaction framex_lsp
order_form finance_receivepayment  > ./framex_lsp_14_tabs_data_`date
+%F`_bk.sql
```

11.archery平台连接腾讯云RDS进行在线DDL操作时，报错：

```
Execute: Unexpected database port reported: 20508.
```

解决办法：

```
在ghost配置中，添加参数：
ghost_aliyun_rds = true
```

12.表有损坏，报错如下：

```
Table xxx  is marked as crashed and should be repaired
```

解决办法：

```
首先进入mysql命令台：
mysql -u root -p 回车  输入密码
查询所有的库
mysql> show databases;
进入数据库"sunacwy_mdm"是库名
mysql> use sunacwy_mdm;
check table table_name（table_name:是出现错误的表）用来检查出现问题的表的状态，出现错误就正常
mysql> check table tb_report_room_coverage_details;
+-----------------------------------------------+-------+----------+-------------------------------------------+
| Table                                         | Op    | Msg_type | Msg_text                                  |
+-----------------------------------------------+-------+----------+-------------------------------------------+
| sunacwy_mdm.tb_report_room_coverage_details | check | warning  | Table is marked as crashed                |
| sunacwy_mdm.tb_report_room_coverage_details | check | warning  | 2 clients are using or haven't closed the table properly |
| sunacwy_mdm.tb_report_room_coverage_details | check | error    | Record at pos: 602288480 is not remove-marked |
| sunacwy_mdm.tb_report_room_coverage_details | check | error    | record delete-link-chain corrupted        |
| sunacwy_mdm.tb_report_room_coverage_details | check | error    | Corrupt                                   |
+-----------------------------------------------+-------+----------+-------------------------------------------+
5 rows in set (0.02 sec)
然后用repair table table_name;
mysql> repair table tb_report_room_coverage_details;
```

```
+-----------------------------------------------+--------+----------+------------
-----------------------------+
| Table                                         | Op     | Msg_type | Msg_text
                             |
+-----------------------------------------------+--------+----------+------------
-----------------------------+
| sunacwy_mdm.tb_report_room_coverage_details | repair | warning  | Number of
rows changed from 0 to 464093 |
| sunacwy_mdm.tb_report_room_coverage_details | repair | status   | OK
                             |
+-----------------------------------------------+--------+----------+------------
-----------------------------+
2 rows in set (30.05 sec)

再用check table table_name,检查一下就ok
mysql> check table tb_report_room_coverage_details;
+-----------------------------------------------+-------+----------+----------+
| Table                                         | Op    | Msg_type | Msg_text |
+-----------------------------------------------+-------+----------+----------+
| sunacwy_mdm.tb_report_room_coverage_details | check | status   | OK       |
+-----------------------------------------------+-------+----------+----------+
1 row in set (2.70 sec)

```

13、查询archery审核平台回退备份数据：

```
SELECT
id,
rollback_statement,
opid_time,
SUBSTRING(opid_time,1,10),
FROM_UNIXTIME(SUBSTRING(opid_time,1,10))
FROM `rm_2zec1ou9m3urxzt14_mysql_rds_aliyuncs_com_3306_shop`.`goods`
WHERE id<=1075810 AND id>=1074810;
```

14.mysql的慢日志查询分析：

```
用法：
直接分析慢查询文件：
pt-query-digest slow.log > slow20170803.log
```

```
分析最近1小时内的查询：
pt-query-digest --since=1h slow.log > 1slow.log

分析指定时间范围内的查询：
# pt-query-digest /acdata/mysql/c376c43e4f24-slow.log --since '2024-03-17 16:30:00' --until '2024-03-19 04:00:01' >  ./10.3.8.160_0318_0000_0319_1200.log

分析指含有select语句的慢查询：
pt-query-digest --filter '$event->{fingerprint} =~ m/^select/i' slow.log > slow4.log

针对某个用户的慢查询：
pt-query-digest --filter '($event->{user} || "") =~ m/^admin/i' slow.log > slow5.log

查询所有所有的全表扫描或full join的慢查询：
pt-query-digest --filter '(($event->{Full_scan} || "") eq "yes") ||(($event->{Full_join} || "") eq "yes")' slow.log> slow6.log

把查询保存到query_review表：
pt-query-digest --user=root -password=abc123 --review h=localhost,D=test,t=query_review --create-review-table slow.log

把查询保存到query_history表：
pt-query-digest --user=root -password=abc123 --review h=localhost,D=test,t=query_ history --create-review-table slow.log
```

15、Dbdoctor部署agent报错：

```
[root@wtj1vpfrmysql01 agent]# ./agent --install -h 172.17.44.40 -k 172.17.44.70:9092 -s 172.17.44.70:13000
2024-06-14 11:58:22.582296439 [ERROR] [agent check_time:65]: time check failed,time check failed,time out of sync
```

问题原因，agent上的时间和server端的时间不一致。

解决办法：调整系统时间和server端时间保持一致即可。

```
# timedatectl set-ntp yes
[root@wtj1vpfrmysql01 agent]# timedatectl status
      Local time: Fri 2024-06-14 12:04:12 CST
```

```
  Universal time: Fri 2024-06-14 04:04:12 UTC
        RTC time: Fri 2024-06-14 04:04:12
       Time zone: Asia/Shanghai (CST, +0800)
     NTP enabled: yes
NTP synchronized: yes
 RTC in local TZ: no
      DST active: n/a
```

三、Redis相关问题

```
异常信息是:
Redis Client On Error: ReplyError: ERR illegal address: 2.0.1.114:58924 Config right?
```

解决办法:

在阿里云redis中添加白名单配置即可。

四、Sqlserver数据库相关问题

针对新收费全量备份数据，每隔一段时间，服务器本地清理失败的问题。采用新的处理策略，在共享存储服务器上，创建清理脚本，每天定时自动调用执行清理14天以前的数据，只保留14天以内的数据。

```
@echo 删除指定路径下指定N天前的文件
@echo off
@echo.----------------------------------------------------------------------
@echo.**********************************************************************
@echo.    温馨提醒:
@echo.       1.以管理员身份运行
@echo.       2.以文件的最后修改日期为准
@echo.       3.需要系统自带的forfiles命令的支持
@echo.       4.使用时请先测试，删除的数据不可恢复
@echo.       5.如果测试结果无误，把del前面的echo去掉，即可实现真正删除
@echo.----------------------------------------------------------------------
```

```
@echo.

rem 指定待删除文件的存放路径
set SrcDir=D:\dbbaka4\prod_tidb
rem 指定几天前的文件
set DaysAgo=14
@echo.----------------------------------------------------------------------------
@echo 正在扫描 %SrcDir%路径下%DaysAgo%天前的备份文件......

rem 下面命令为真正删除符合条件的文件日志记录
forfiles /p %SrcDir% /s /m *.* /d -%DaysAgo% /c "cmd /c echo  正在删除 del /f /q /a  @path"  >>./%DATE:~0,4%%DATE:~5,2%%DATE:~8,2%%time:~0,2%%time:~3,2%.log

@echo.
@echo 正在删除文件中......

rem 下面命令为真正删除符合条件的文件且不可恢复
forfiles /p %SrcDir% /s /m *.* /d -%DaysAgo% /c "cmd /c  del /f /q /a  @path"
@echo 删除已完成，若想查看具体删除了哪些文件请看本地删除日志
@echo.
pause exit
```

我们可以在机器上创建一个定时计划任务，每天自行执行脚本判断清理过期的备份数据。

1. 开始-->所有程序-->附件-->系统工具-->任务计划程序
2. 任务计划程序（本地）-->任务接话程序库-->创建基本任务-->
3. 输入 名称： 描述：点击下一步
4. 触发器：设置周期
5. 设置详细时间
6. 操作选择 启动程序
7. 选择要执行的bat脚本就是上面所写好的脚本
8. 点击完成

注：验证是否能够成功执行计划任务，在计划任何的执行界面中配置的启动时间提前当前时间2分钟，待显示上次成功执行（0X0返回码表示成功执行）以后确认定时任务可以成功执行。

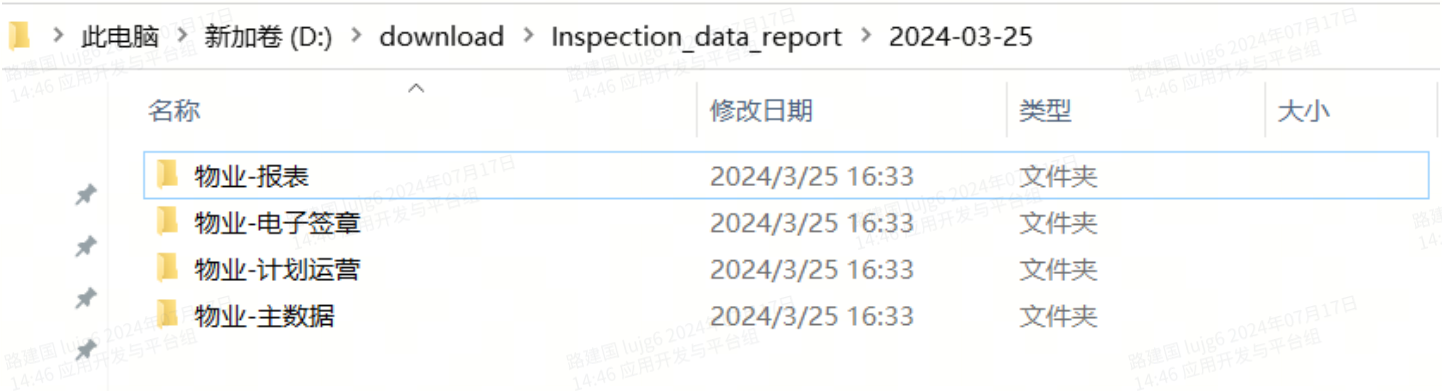
五、linux相关问题

1、编码问题：

在下载文件时，Linux 下的文件名如果包含非Windows系统默认编码的字符（如UTF-8编码的中文字符），在 Windows 系统中可能会显示为乱码。解决这个问题的一种方法是在压缩文件夹之前，使用 `convmv` 或 `iconv` 命令在Linux系统中将文件名转换为Windows支持的编码格式（如GBK）

解决办法：使用如下命令会将Inspection_data_report目录下的所有文件名或文件夹名从UTF-8转换为GBK编码。

```
1 # convmv -f utf-8 -t gbk -r --notest --replace Inspection_data_report
```



反向将gbk转换为utf-8编码：

```
1 # convmv -f gbk -t utf-8 -r --notest --replace Inspection_data_report
```

```
[root@wtj1vpdbmgnt01 check_mysql_report]# ll Inspection_data_report/2024-03-25/
total 16
drwxr-xr-x 2 root root 4096 Mar 25 16:42 物业-主数据
drwxr-xr-x 2 root root 4096 Mar 25 16:42 物业-报表
drwxr-xr-x 2 root root 4096 Mar 25 16:42 物业-电子签章
drwxr-xr-x 2 root root 4096 Mar 25 16:42 物业-计划运营
```

2、umount无法卸载问题:

完整的过程可能如下：

1. 找到占用进程：

```
1 lsof +D /sunac_data/nfs
```

1. 或

```
fuser -m /sunac_data/nfs
```

1. 强制终止进程：

```
fuser -km /sunac_data/nfs
```

1. 尝试卸载：

```
umount /sunac_data/nfs
```

1. 如果失败，尝试强制卸载：

```
umount -l /sunac_data/nfs
```

1. 或

```
umount -f /sunac_data/nfs
```

这样可以有效解决目标挂载点忙的问题并成功卸载。请谨慎操作，特别是在强制终止进程时，确保不会影响系统的关键功能。

```
# rm -rf /sunac_data/nfs
# mkdir -p /sunac_data/nfs
# mount -v -t cifs //10.3.8.211/dbbaka4$ /sunac_data/nfs -o domain=localhost,user=databak,password='Weydt2204#%jg',dir_mode=0777,file_mode=0777
mount.cifs kernel mount options: ip=10.3.8.211,unc=\\10.3.8.211\dbbaka4$,dir_mode=0777,file_mode=0777,user=databak,domain=localhost,pass=********
```

3、linux下cp拷贝文件保持文件时间等属于不变：

如果你只需要修改时间和访问时间，可以使用 `touch`：

```
1
2 cp file.txt /backup/file.txt
3 touch -r file.txt /backup/file.txt
```