

3.数据校验与修复工具调研

1.PingCAP的sync-diff-inspector工具

官方文档: <https://docs.pingcap.com/zh/tidb/stable/sync-diff-inspector-overview>

快速使用

下载页面: <https://docs.pingcap.com/zh/tidb/stable/download-ecosystem-tools>

这里在10.2.3.167环境已经下载并解压到tidb-community-toolkit目录下:

```
1 [root@localhost tools]# pwd
2 /usr/local/tools
3 [root@localhost tools]# ll
4 total 1163956
5 drwxr-xr-x 6 root root      4096 Jan 16 13:37 tidb-community-toolkit
6 -rw-r--r-- 1 root root 1191886466 Jan 12 18:27 tidb-community-toolkit-v7.5.0-
  linux-amd64.tar.gz
```

为了快速使用,我们这里仅先将同一个表,既作为源,又作为目标,进行校验。

1.配置

在tidb-community-toolkit目录下新增配置文件:

config.toml

```
1 ##### Global config #####
2 check-thread-count = 4           # 检查数据的线程数量
3 export-fix-sql = true             # 如果开启,若表数据存在不一致,则输出用于修复的
  SQL语句
4 check-struct-only = false        # 只对比表结构而不对比数据
5 ##### Datasource config #####
6 [data-sources]
7 [data-sources.mysql1]           # 上游MySQL数据库配置(源端)
8     host = "10.2.3.167"
```

```

9      port = 3306
10     user = "user"
11     password = "Lachesis-mh_1024"
12 [data-sources.mysql2] # 下游MySQL数据库配置（目标端）
13     host = "10.2.3.167"
14     port = 3306
15     user = "user"
16     password = "Lachesis-mh_1024"
17
18 ##### Task config #####
19 [task]
20     output-dir = "./output_1"
21     source-instances = ["mysql1"] # 上游数据库，内容是data-sources声明的唯一标识id，分库分表场景下支持多个上游数据库，如：["mysql10", "mysql20"]
22     target-instance = "mysql2" # 下游数据库，内容是data-sources声明的唯一标识id
23     target-check-tables = ["windranger_emr.pat_inhos_order_group_bak1"] # 需要比对的下游数据库的表，每个表需要包含数据库名和表名，两者由.隔开

```

2.执行

```

1 # ./sync_diff_inspector --config=./config.toml
2 A total of 1 tables need to be compared
3
4 Progress [>-----] 0%
5 Comparing the table structure of
6 ``windranger_emr`.`pat_inhos_order_group_bak1`` ... equivalent
7 Comparing the table data of ``windranger_emr`.`pat_inhos_order_group_bak1``
8 ... equivalent
9
10 Progress [=====] 100%
11 A total of 1 table have been compared and all are equal.
12 You can view the comparison details through './output_1/sync_diff.log'

```

3.查看结果

输出文件：

```
1 # tree ./output_1
2 ./output_1
3 |─ checkpoint
4 |   └─ 8cf88e35534ead7225e205a6a8ae2911df262ef67ba58083e9fe15f61f1411a8
5 |─ fix-on-mysql2
6 |─ summary.txt
7 └─ sync_diff.log
```

具体说明参见：<https://docs.pingcap.com/zh/tidb/stable/sync-diff-inspector-overview#%E8%BE%93%E5%87%BA%E6%96%87%E4%BB%B6>

查看校验与修复的汇总输出文件：

```
[root@localhost tidb-community-toolkit]# cat output_1/summary.txt
Summary
```

Source Database

```
host = "10.2.3.167"
port = 3306
user = "user"
```

Target Databases

```
host = "10.2.3.167"
port = 3306
user = "user"
```

Comparison Result

The table structure and data in following tables are equivalent.

TABLE	UPCOUNT	DOWNCOUNT
`windranger_emr`.`pat_inhos_order_group_bak1`	28735078	28735078

Time Cost: 1m54.330992337s
Average Speed: 36.621741MB/s

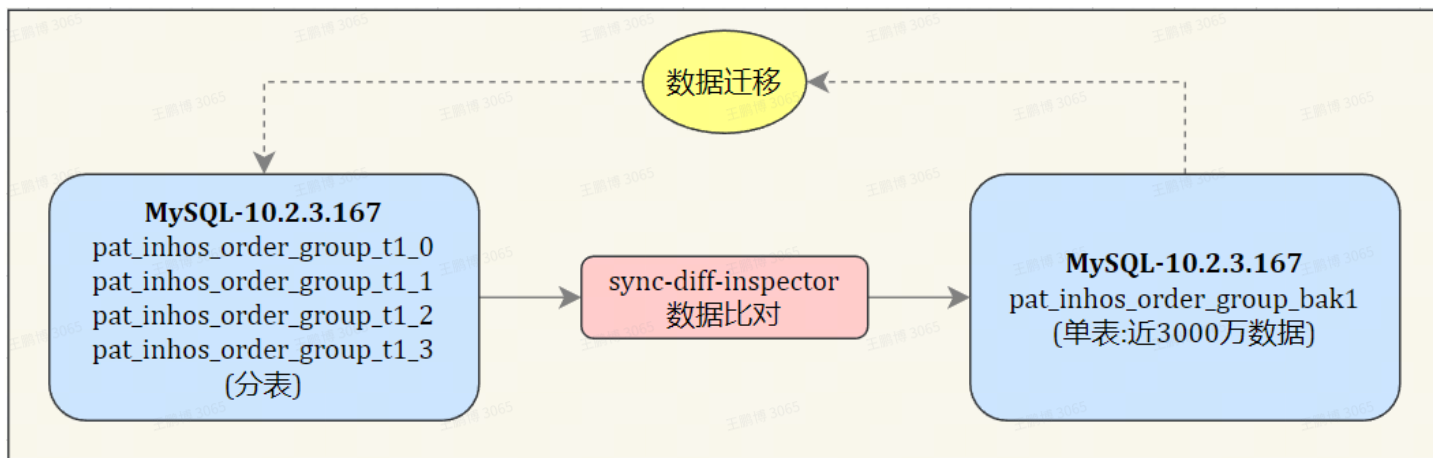
校验耗时

可以看到表结构和数据是相同的。

是否符合预期：是！

分库分表校验_场景1

前提：准备单表和分库分表，并将单表中的数据已经迁移至分库分表。



1.配置

在tidb-community-toolkit目录下新增配置文件：

config_order_group.toml

```
1  ##### Global config #####
2  check-thread-count = 4          # 检查数据的线程数量
3  export-fix-sql = true           # 如果开启，若表数据存在不一致，则输出用于修复的
    SQL语句
4  check-struct-only = false      # 只对比表结构而不对比数据
5
6  ##### Datasource config #####
7  [data-sources.mysql1]          # 上游MySQL数据库配置（源端）
8      host = "10.2.3.167"
9      port = 3306
10     user = "user"
11     password = "Lachesis-mh_1024"
12     route-rules = ["rule1"]
13  [data-sources.mysql2]          # 下游MySQL数据库配置（目标端）
14     host = "10.2.3.167"
15     port = 3306
16     user = "user"
17     password = "Lachesis-mh_1024"
18
19  ##### Routes #####
20  # 如果需要对比大量的不同库名或者表名的表的数据，或者用于校验上游多个分表与下游总表的数据，
    可以通过table-rule来设置映射关系
21  [routes.rule1]
22     schema-pattern = "windranger_emr" # 匹配数据源的库名，支持通配
        符"*"和"?"
23     table-pattern = "pat_inhos_order_group_t1_[0-3]" # 匹配数据源的表名，支持通配
        符"*"和"?"
24     target-schema = "windranger_emr" # 目标库名
```

```

25 target-table = "pat_inhos_order_group_bak1" # 目标表名
26
27 ##### Task config #####
28 [task]
29     output-dir = "./output_order_group_sharding"
30     source-instances = ["mysql1"] # 上游数据库，内容是data-sources声明的唯一标识id，分库分表场景下支持多个上游数据库，如：["mysql10", "mysql20"]
31     target-instance = "mysql2" # 下游数据库，内容是data-sources声明的唯一标识id
32     target-check-tables = ["windranger_emr.pat_inhos_order_group_bak1"] # 需要比对的下游数据库的表，每个表需要包含数据库名和表名，两者由.隔开

```

2.执行

```

1 # ./sync_diff_inspector --config=./config.toml
2 A total of 1 tables need to be compared
3
4 Progress [>-----] 0%
5 Comparing the table structure of
6 ``windranger_emr``.``pat_inhos_order_group_bak1`` ... failure
7 Comparing the table data of ``windranger_emr``.``pat_inhos_order_group_bak1``
8 ... equivalent
9
10 Progress [=====] 100%
11 The structure of ``windranger_emr``.``pat_inhos_order_group_bak1`` is not equal
12 The rest of tables are all equal.
13 A total of 1 tables have been compared, 0 tables finished, 1 tables failed, 0
14 tables skipped.
15 The patch file has been generated in
16 'output_order_group_sharding/fix-on-mysql2/'
17 You can view the comparision details through
18 './output_order_group_sharding/sync_diff.log'

```

从上面日志看到，校验结果是：

- 表结构不相同：因为本来对比表名称就不同，所以表结构是不同的。

- 表数据相同。

3.查看结果

查看校验与修复的汇总输出文件：

```
[root@localhost tidb-community-toolkit]# cat ./output_order_group_sharding/summary.txt
Summary

Source Database

host = "10.2.3.167"
port = 3306
user = "user"

Target Databases

host = "10.2.3.167"
port = 3306
user = "user"

Comparison Result

The table structure and data in following tables are equivalent
The following tables contains inconsistent data

+-----+-----+-----+-----+-----+-----+
| TABLE | RESULT | STRUCTURE EQUALITY | DATA DIFF ROWS | UPCOUNT | DOWNCOUNT |
+-----+-----+-----+-----+-----+-----+
| `windranger_emr`.`pat_inhos_order_group_bak1` | succeed | false | +0/-0 | 28735078 | 28735078 |
+-----+-----+-----+-----+-----+-----+

Time Cost: 4m2.743084715s
Average Speed: 17.248689MB/s
```

注意：并不是像图中红字那样理解的！而是说以下展示的相同的表结构和表数据信息，可以看到并没有。接下来，展示的是表中不一致数据信息。

是否符合预期：是！

4.扩展

这里假设我改动了单表的数据，看看能否根据工具生成最终的数据修复SQL？

```
1 UPDATE windranger_emr.pat_inhos_order_group_bak1 t SET t.create_person =
  'system_001' WHERE t.seq_id = 2165679
```

重新执行，查看结果：

```
1 # ./sync_diff_inspector --config=./config_order_group.toml
```

```

2 A total of 1 tables need to be compared
3
4 Progress [>-----] 0%
0/0
5 Comparing the table structure of
  ``windranger_emr`.`pat_inhos_order_group_bak1`` ... failure
6 Comparing the table data of ``windranger_emr`.`pat_inhos_order_group_bak1``
  ... failure
7
8 Progress [=====>] 100%
0/0
9 The structure of `windranger_emr`.`pat_inhos_order_group_bak1` is not equal
10 The data of `windranger_emr`.`pat_inhos_order_group_bak1` is not equal
11
12 The rest of tables are all equal.
13
14 A total of 1 tables have been compared, 0 tables finished, 1 tables failed, 0
  tables skipped.
15 The patch file has been generated in
16     'output_order_group_sharding/fix-on-mysql2/'
17 You can view the comparision details through
  './output_order_group_sharding/sync_diff.log'

```

从上面日志看到，校验结果是：

- 表结构不相同：因为本来对比表名称就不同，所以表结构是不同的。
- 表数据不相同：因为我们修改了一条记录。那具体的变动信息是哪些呢？有没有修复SQL呢？

查看校验与修复的汇总输出文件：


```
[root@localhost tidb-community-toolkit]# cat ./output_order_group_sharding/summary.txt
Summary

Source Database

host = "10.2.3.167"
port = 3306
user = "user"

Target Databases

host = "10.2.3.167"
port = 3306
user = "user"

Comparison Result

The table structure and data in following tables are equivalent

The following tables contains inconsistent data 表包含不一致的数据信息
+-----+-----+-----+-----+-----+-----+
| TABLE | RESULT | STRUCTURE EQUALITY | DATA DIFF ROWS | UPCOUNT | DOWNCOUNT |
+-----+-----+-----+-----+-----+-----+
| `windranger_emr`.`pat_inhos_order_group_bak1` | succeed | false | +1/-1 | 28735078 | 28735078 |
+-----+-----+-----+-----+-----+-----+

Time Cost: 3m59.433643854s
Average Speed: 17.487100MB/s
```

此时，查看数据修复SQL：

```
[root@localhost tidb-community-toolkit]# tree output_order_group_sharding/
output_order_group_sharding/
├── checkpoint
│   └── 544fd16ddfe48de09fc3594bb712788a3c0c2cc2984e0013bd3f4fdcfb9cd155
├── fix-on-mysql2
│   └── windranger_emr:pat_inhos_order_group_bak1:0:0-0:0.sql
├── summary.txt
└── sync_diff.log
```

查看具体的SQL：

```
[root@localhost tidb-community-toolkit]# cat output_order_group_sharding/fix-on-mysql2/windranger_emr\pat_inhos_order_group_bak1\0:0-0:0.sql
-- table: windranger_emr.pat_inhos_order_group_bak1
-- range in sequence: (seq_id) <= (2262080)
/*
  DIFF COLUMNS | `CREATE PERSON`
  source data   | 'system'
  target data    | 'system_001'
*/
REPLACE INTO `windranger_emr`.`pat_inhos_order_group_bak1`(`seq_id`,`group_unique_code`,`order_group_no`,`inhos_code`,`orderbar`,`package_bar`,`plan_time`,`order_sort_no`,`source_type`,`isprint`,`execute_status`,`execute_date`,`print_date`,`execute_person`,`print_person`,`apply_time`,`is dispensed`,`remark`,`reason`,`execute_type`,`start_execute_user`,`start_execute_date`,`start_check_user`,`end_execute_user`,`end_execute_date`,`create_time`,`create_person`,`update_time`,`update_person`) VALUES (2165679,'107844532202160207815846237','815846237','815846237','10784453','blrxy@107844532202160207815846237',NULL,'2022-02-16 02:07:12',1,NULL,0,0,NULL,NULL,NULL,NULL,0,NULL,NULL,0,NULL,NULL,NULL,NULL,'2022-02-16 18:00:05','system',NULL,NULL);
```

可以看到，是我们先前更新的数据信息，该SQL的作用是将目标表数据更新为与源表一致。

可以看到这里是以上游表为参照，所以最终的目的是让下游表与上游表数据保持一致，通过官方文档证实！





一个 SQL 文件会包含该 chunk 的所属表以及表示的范围信息。对每个修复 SQL 语句，有三种情况：











- 下游数据库缺失行，则是 REPLACE 语句
- 下游数据库冗余行，则是 DELETE 语句
- 下游数据库行部分数据不一致，则是 REPLACE 语句，但会在 SQL 文件中通过注释的方法标明不同的列

```
-- table: sbtest.sbtest99
-- range in sequence: (3690708) < (id) <= (3720581)
/*
DIFF COLUMNS | `k` | `c` | `PAD`
source data | 2501808 | 'hello' | 'world'
target data | 5003616 | '0709824117-9809973320-4456050422' | '1714066100-7057807621-1425865505'
*/
REPLACE INTO `sbtest`.`sbtest99`(`id`,`k`,`c`,`pad`) VALUES (3700000,2501808,'hello','world');
```

是否符合预期：是！

Original Confluence page attachments

Name	Size	Created by	Created on	Labels	Comments
 image-2024-1-16_15-8-... 44.92KB	44.92 KB	王鹏博	2024-01-16T15:07:38.000+08:00		
 image-2024-1-16_15-6-... 41.68KB	41.68 KB	王鹏博	2024-01-16T15:05:24.000+08:00		
 image-2024-1-16_14-51-... 105.90KB	105.90 KB	王鹏博	2024-01-16T14:50:13.000+08:00		
 image-2024-1-16_14-45-... 93.99KB	93.99 KB	王鹏博	2024-01-16T14:44:25.000+08:00		

<div><div>image-2024-1-16_14-43-... 32.70KB</div></div>	32.70 KB	王鹏博	2024-01-16T14:42:49.000+08:00		
<div><div>image-2024-1-16_14-42-... 105.70KB</div></div>	105.70 KB	王鹏博	2024-01-16T14:41:00.000+08:00		
<div><div>image-2024-1-16_14-25-... 106.99KB</div></div>	106.99 KB	王鹏博	2024-01-16T14:24:28.000+08:00		
<div><div>image-2024-1-16_14-24-... 103.06KB</div></div>	103.06 KB	王鹏博	2024-01-16T14:23:02.000+08:00		
<div><div>image-2024-1-16_14-0-... 78.58KB</div></div>	78.58 KB	王鹏博	2024-01-16T13:59:20.000+08:00		