# 1.    引言

“MySQL主从复制”技术在互联网行业常见高可用[架构](http://lib.csdn.net/base/architecture" \o "大型网站架构知识库" \t "_blank)中应用非常广泛，例如常见的一主一从复制架构、keepalived＋MySQL双主（主从）复制架构、MHA＋一主两从复制架构等等都应用了MySQL主从复制技术。但因主从复制是基于binlog的逻辑复制，难免出现复制数据不一致的风险，这个风险不但会引起用户数据访问前后不一致的风险，而且会导致后续复制出现1032、1062错误进而引起复制架构停滞的隐患，为了及时发现并解决这个问题，我们需要定期或不定期地开展主从复制数据一致性的校验和修复工作，那么如何实现这项工作呢？又如何实现这项工作的自动化呢？我们来探讨这些问题。

# 2.    数据一致性校验和修复方法

为了实现主从复制数据一致性校验和修复，我们首先推荐两个热门工具，分别是percona公司的 pt-table-checksum和pt-table-sync，前者用来实现主从复制数据一致性的校验，后者实现数据修复，将数据修复到一致。

## 2.1        工作原理

**pt-table-checksum**通过SQL在主库执行数据块的校验，再将相同的语句传送到从库，并在从库上计算数据块的校验，最后将主从库相同块的校验值进行对比，辨别主从数据是否不一致。

**pt-table-sync**用来修复主从复制数据的不一致，使得它们修复到最终一致，也可以实现多个实例或者是应用双写或多写的多个不相关的[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "_blank)实例修复到一致。同时它还内部集成了pt-table-checksum的校验功能，可以一边校验一边修复，也可以基于pt-table-checksum的计算结果来进行修复。

再次提示:只需要在MySQL主库上安装和使用pt工具

## 2.2        下载及方法

     这两个工具均包含在percona-toolkit里，线上下载地址：  https://www.percona.com/downloads/percona-toolkit/

     在MySQL主数据库上下载：

wget https://www.percona.com/downloads/percona-toolkit/2.2.20/RPM/percona-toolkit-2.2.20-1.noarch.rpm

依赖包安装：

yum install perl-DBI.x86\_64 -y

yum install perl-Time-HiRes.x86\_64 -y

yum install perl-Class-DBI-mysql.noarch -y

yum install perl-IO-Socket-SSL.noarch -y

yum install perl-TermReadKey.x86\_64 -y

percona-toolkit工具安装：

rpm -ivh percona-toolkit-2.2.20-1.noarch.rpm

## 2.3        校验和修复方法

### 2.3.1在主库创建校验账号(可跳过)

GRANT UPDATE,INSERT,DELETE,SELECT, PROCESS, SUPER, REPLICATION SLAVE ON \*.\* TO 'ptcheck'@'MasterIP'identified by 'PASSWORD';

GRANT ALL ON test.\* TO ' ptcheck '@'MasterIP' IDENTIFIED BY 'PASSWORD';

(我一般直接用root账号校验和修复，没有执行这个步骤)

### 2.3.2在MySQL主库创建test库，并创建校验信息表(可跳过)

CREATE DATABASE IF NOT EXISTS test DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

CREATETABLE IF NOT EXISTS checksums (

db char(64)NOT NULL,

tblchar(64) NOT NULL,

chunk intNOT NULL,

chunk\_timefloat NULL,

chunk\_indexvarchar(200) NULL,

lower\_boundarytext NULL,

upper\_boundarytext NULL,

this\_crcchar(40) NOT NULL,

this\_cntint NOT NULL,

master\_crcchar(40) NULL,

master\_cntint NULL,

tstimestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY(db, tbl, chunk),

INDEXts\_db\_tbl (ts, db, tbl)

)ENGINE=InnoDB;

(注意了，这一步骤我也从来没有执行过，因为执行pt-table-checksum时，会自动创建库和表)

### 2.3.3判断主键

若无主键做校验和修复对性能影响非常重，数据校验和修复最重要的约束便是主健，无主键或唯一索引，将导致修复不成功。

执行pt-table-checksum校验时，没有“主键或唯一索引”的表都会报错，我们从日志里可以知道，你此时可以去添加“主键或唯一索引”再重新pt-table-checksum这张表

### 2.3.4主从数据校验

./pt-table-checksum --nocheck-binlog-format --nocheck-plan --nocheck-replication-filters --replicate=test.checksums  --databases=db1 --tables=tb1 -h 192.168.XXX.XX -P 3306 -u'hangxing' -p'PASSOWRD' --recursion-method="processlist"

解析：

--no-check-binlog-format 不检查复制的binlog模式。

       --nocheck-replication-filters 不检查复制过滤器，建议启用。

--replicate=test.checksums 检查结果写入test库的checksums表里。

--databases=db1  --tables=tb1 校验db1库里的tb1表，若无参数则校验全库全表。

-h 192.168.XXX.XX -P 3306 主库IP地址和3306端口。

-u'hangxing' -p'PASSOWRD' 校验账号密码。

--recursion-method="processlist" 用processlist的方法来发现从库。

我常用的命令举例

全库校验

pt-table-checksum u='root',p='jishu8cc',P=3306 --nocheck-replication-filters --nocheck-binlog-format --replicate=test.checksums 2>&1 | tee /1.log

指定库校验

pt-table-checksum u='root',p='jishu8cc',P=3306 --databases=smarttmw\_cn,scratch --nocheck-replication-filters --nocheck-binlog-format --replicate=test.checksums 2>&1 | tee /1.log

指定表校验

pt-table-checksum u='root',p='jishu8cc',P=3306 --nocheck-replication-filters --tables=ecloud1\_7\_1r2m1.tch\_teacher\_csv --nocheck-binlog-format --replicate=test.checksums 2>&1 | tee /1.log

执行后的输出结果：

TS   ERRORS      DIFFS      ROWS  CHUNKS  SKIPPED   TIME  TABLE

03-23T15:29:17    0     1    30000       1       0  1.270 testhx1.testhx1

解析：

TS            ：完成检查的时间。

ERRORS     ：检查时候发生错误和警告的数量。

DIFFS        ：0表示一致，大于0表示不一致。主要看这一列有无不一致数据。

ROWS       ：表的行数。

CHUNKS    ：被划分到表中的块的数目。

SKIPPED    ：由于错误或警告或过大，则跳过块的数目。

TIME         ：执行的时间。

TABLE       ：被检查的表名。

    上述输出关键看DIFFS列，结果为0说明数据一致，无需进行数据修复，如果不为0则需要继续开展数据一致性修复工作。上述语句执行后也会将详细的内容会写入test库的checksums表中，可以查看这个库表得到详细的数据校验信息，此表中信息内容格式举例如下：

主库的test.checksums中输出this\_crc和master\_crc，无不一致。

mysql> select \* fromtest.checksums;

+---------+---------+-------+------------+-------------+----------------+----------------+----------+----------+------------+------------+---------------------+

| db      | tbl    | chunk | chunk\_time | chunk\_index | lower\_boundary | upper\_boundary | this\_crc | this\_cnt | master\_crc| master\_cnt | ts                  |

+---------+---------+-------+------------+-------------+----------------+----------------+----------+----------+------------+------------+---------------------+

| testhx1 | testhx1 |     1 |  0.003661 | NULL        | NULL           | NULL           | cac6c46f|        4 | cac6c46f   |         4 | 2016-03-23 15:29:16 |

+---------+---------+-------+------------+-------------+----------------+----------------+----------+----------+------------+------------+---------------------+

1 row in set (0.00 sec)

从库的test.checksums中输出this\_crc和master\_crc，不一致。

mysql>select \* from checksums;

+---------+---------+-------+------------+-------------+----------------+----------------+----------+----------+------------+------------+---------------------+

|db      | tbl     | chunk | chunk\_time | chunk\_index |lower\_boundary | upper\_boundary | this\_crc |this\_cnt | master\_crc | master\_cnt | ts                  |

+---------+---------+-------+------------+-------------+----------------+----------------+----------+----------+------------+------------+---------------------+

|testhx1 | testhx1 |     1 |   0.003661 | NULL        | NULL           | NULL           | 7c2e5f75|        5 | cac6c46f  |          4 | 2016-03-23 15:29:16 |

+---------+---------+-------+------------+-------------+----------------+----------------+----------+----------+------------+------------+---------------------+

1row in set (0.00 sec)

点评:

我一般在一主一从架构时，是不会去看这个表的，因为在上面pt-table-checksum生成的log里，我已可以从DIFFS列知道tables是否存在数据不一致。但是大家可以思考一下，如果您的公司不是使用一注意从架构，而是一主多从架构呢？那么DIFFS列只说明有不一致，那么到底是哪一个从库不一致？

这就需要到每个从库里查看test.checksums的内容了，对吧。

### 2.3.5主从数据修复

pt-table-sync --execute --replicate test.checksums --sync-to-master h=10.1.135.10,P=3306,u=root,p=jishu8cc

再次提醒:

主库上执行，IP填你需要修复的从库的IP地址。

他会根据pt-table-checksum的结果test.checksums表，来修复数据。

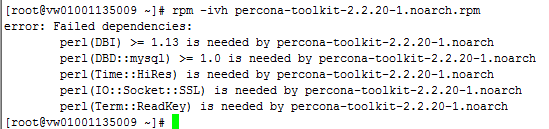
### 2.3.6再次校验

参考2.3.4

# 3.Troubleshoot

1.安装遇到依赖问题

安装时发现需要安装以下5个依赖包，如何找到依赖包的包名？



参考:

yum search perl | grep IO | grep Socket | grep SSL



# 4.总结

pt-table-checksum是校验主从数据不一致的最好工具。由于MySQL复制自身的缺陷，或主从切换不严谨，或备份软件bug等原因，都可能导致主从数据的不一致。不管你管不管，不一致都在那里，就看数据对你重不重要，重要的话，就定期做下检查并修复吧。

参考文章:

<http://blog.csdn.net/hangxing_2015/article/details/52585855>

http://www.cnblogs.com/huminxxl/p/3978559.html