# mysql binlog Flashback

# ****Flashback在mysql5.6的应用****

## ****工具简介****

针对row格式的binlog可以解析为回滚操作语句，如delete反向生成insert，update生成反向的update,insert反向生成delete；在线上误操作的时候我们可以快速的将数据恢复到修改前的状态。

## ****安装****

1、下载地址

~~<http://pan.baidu.com/s/1nvGOOIl>~~ 该地址中下载的mysqlbinlog20160408文件 -v参数存在问题，使用下面的地址下载。

[https://pan.baidu.com/s/1c1Ub1x2](https://pan.baidu.com/s/1c1Ub1x2" \o "mysqlbinlog)

~~[mysqlbinlog\_56.zip](/download/attachments/5931198/mysqlbinlog_56.zip?version=1&modificationDate=1493199623000&api=v2)~~

[mysqlbinlog\_57.zip](/download/attachments/5931198/mysqlbinlog_57.zip?version=1&modificationDate=1493199524000&api=v2)

[libstdc++.so.6.0.20](/download/attachments/5931198/libstdc++.so.6.0.20?version=1&modificationDate=1493199529000&api=v2)

创建新服务binlogflashback， 并修改权限

mv /opt/mysqlbinlog /use/bin/binlogflashback

chmod 755 /use/bin/binlogflashback

## ****参数说明****

 -B, --flashback Flashback data to start\_postition or start\_datetime.

COM\_BINLOG\_DUMP or COM\_BINLOG\_DUMP\_GTID commands by

setting the option to either BINLOG-DUMP-NON-GTIDS or BINLOG-DUMP-GTIDS, respectively.

If --read-from-remote-master=BINLOG-DUMP-GTIDS is combined

--start-datetime **##从哪个时间点的binlog开始解析，格式必须为datetime，如'2016-11-11 11:11:11'。可选。默认不过滤。**  
--stop-datetime **##到哪个时间点的binlog停止解析，格式必须为datetime，如'2016-11-11 11:11:11'。可选。默认不过滤。**

--start-position **##事务开始的position点**

--stop-position **##事务结束的position点**

**使用范例**

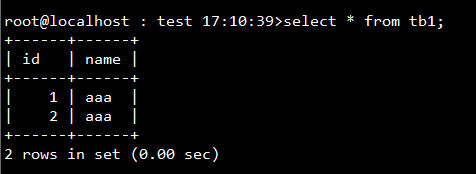
1、数据准备

在数据库中创建如下表

Create Table: CREATE TABLE `tb1` (  
`id` int(11) DEFAULT NULL,  
`name` varchar(20) DEFAULT NULL  
) ENGINE=InnoDB;

执行增、删、改动作

insert into tb1 values(1,'aaa'),(2,'aaa');



delete from tb1 where id=2;

update tb1 set name='bbb' where id=1;

2、反转

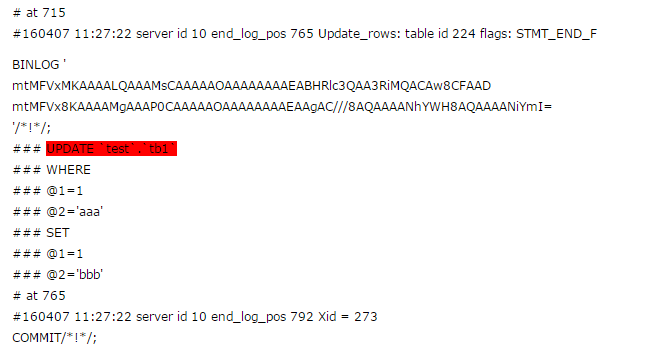
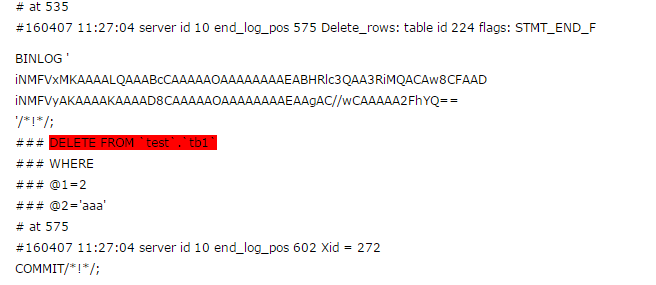
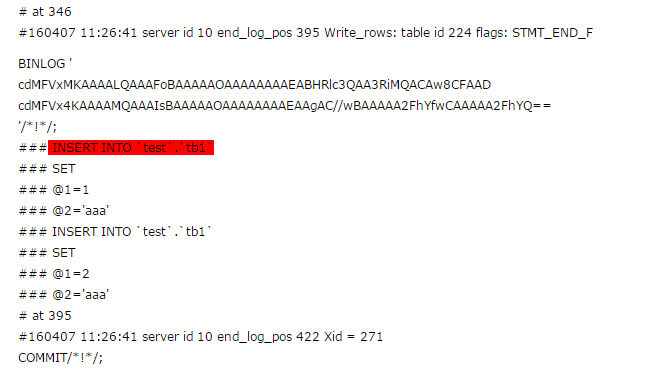
2.1、解析binlog

根据时间点解析binlog

binlogflashback -v --start-datetime='2017-4-24 14:14:00' --stop-datetime='2017-4-24 14:16:08' /data/mysqldata/mysql-bin.000001 > /tmp/tmp.sql

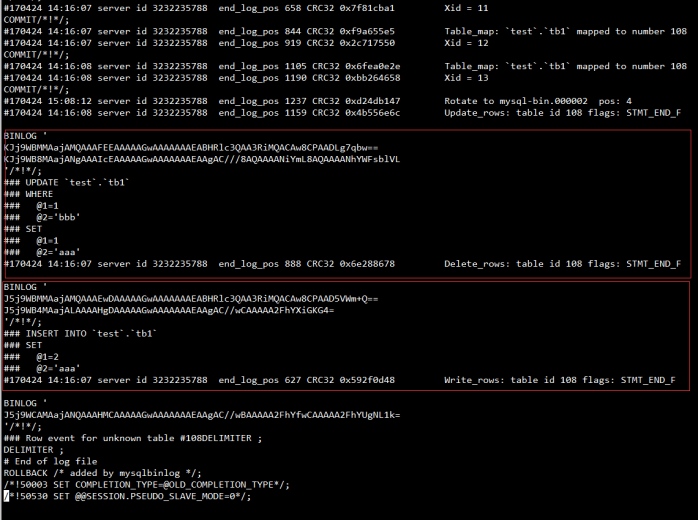
确定目标语句是否存在该范围内，并记录对应的pos点（at 346）

vim /tmp/tmp.sql



2.2、反转

binlogflashback -v -B --start-position=574 --stop-position=1109 /data/mysqldata/mysql-bin.000001 > /tmp/tmp.sql

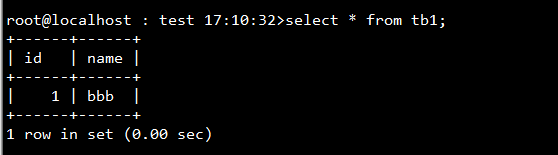


第三步

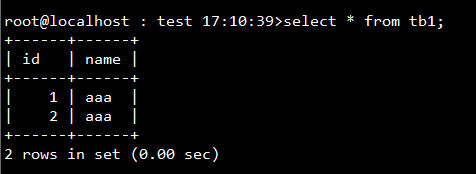
确定好反转sql直接执行，也可以提取需要的sql单独执行。

mysql -uqihang.li -p123 < /tmp/tmp.sql

执行前



执行后



# ****Flashback在mysql5.7的应用****

使用方法和上述内容一致，只是文件不同而已（mysqlbinlog\_57.zip），下面是使用期间出现的一些问题

binlogbackup -v /data/mysqldata/mysql-bin.000001



GLIBCXX\_3.4.15是关于C系统基础运行库GLIBC

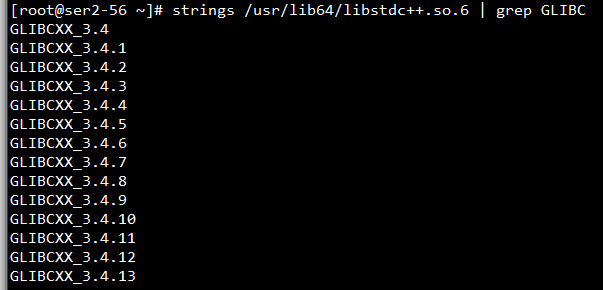
我们着手解决GLIBCXX的问题，C++库，首先查看错误信息提示中的lib文件的信息。

ll /usr/bin64/libstadc++.so.6



打印 libstdc++.so.6的输出信息中限定（grep） GLIBC库的信息

strings /usr/lib64/libstdc++.so.6 | grep GLIBC



发现/usr/lib64/libstdc++.so.6 /usr/lib64/libstdc++.so.6 -> /usr/lib64/libstdc++.so.6.0.13也就是系统以前有一个6.0.13版本，确实比较旧了，那么我们有新版本么？

没有可以下载文档前面的附件

我们现在本机找下

find / -name libstdc++.so.6\*

发现我们有一个/usr/lib64/libstdc++.so.6.0.20的高版本文件



那么我们手动来更新软连接(也可以字节指向上述目录下的文件，不用拷贝）

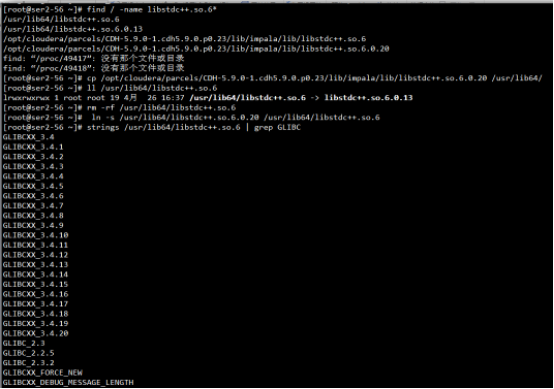
cp /opt/cloudera/parcels/CDH-5.9.0-1.cdh5.9.0.p0.23/lib/impala/lib/libstdc++.so.6.0.20 /usr/lib64/

ll /usr/lib64/libstdc++.so.6

rm -rf /usr/lib64/libstdc++.so.6

ln -s /usr/lib64/libstdc++.so.6.0.20 /usr/lib64/libstdc++.so.6

strings /usr/lib64/libstdc++.so.6 | grep GLIBC



发现GLIBC的版本已经提高到3.4.20，这已经可以满足我们要求了

binlogbackup -v /data/mysqldata/mysql-bin.000001



## ****限制****

binlog\_format=row

只支持insert、update、delete

不支持drop 、truncate、alter等ddl语句