一、初步了解binlog

1、MySQL的二进制日志binlog可以说是MySQL最重要的日志，它记录了所有的DDL和DML语句（除了数据查询语句select）,以事件形式记录，还包含语句所执行的消耗的时间，MySQL的二进制日志是事务安全型的。

a、DDL

----Data Definition Language 数据库定义语言

主要的命令有create、alter、drop等，ddl主要是用在定义或改变表(table)的结构,数据类型，表之间的连接和约束等初始工作上，他们大多在建表时候使用。

b、DML

----Data Manipulation Language 数据操纵语言

主要命令是slect,update,insert,delete,就像它的名字一样，这4条命令是用来对数据库里的数据进行操作的语言

2、mysqlbinlog常见的选项有一下几个：

a、--start-
datetime：从二进制日志中读取指定等于时间戳或者晚于本地计算机的时间

b、--stop-
datetime：从二进制日志中读取指定小于时间戳或者等于本地计算机的时间
取值和上述一样

c、--start-position：从二进制日志中读取指定position 事件位置作为开始。

d、--stop-position：从二进制日志中读取指定position 事件位置作为事件截至

3、一般来说开启binlog日志大概会有1%的性能损耗。

4、binlog日志有两个最重要的使用场景。

a、mysql主从复制：mysql replication在master端开启binlog,master把它的二进制日志传递给slaves来达到master-slave数据一致的目的。

b、数据恢复：通过mysqlbinlog工具来恢复数据。

binlog日志包括两类文件：

1)、二进制日志索引文件(文件名后缀为.index)用于记录所有的二进制文件。

2)、二进制日志文件(文件名后缀为.00000*)记录数据库所有的DDL和DML(除了数据查询语句select)语句事件。

二、开启binlog日志

1、编辑打开mysql配置文件/application/mysql3307/my.cnf在

[mysqld]区块添加

log-bin=mysql-bin(也可指定二进制日志生成的路径，如：log-bin=/opt/Data/mysql-bin)

server-id=1

binlog_format=MIXED(加入此参数才能记录到insert语句)

2、重启mysqld服务

/application/mysql3307/bin/mysqladmin -uroot -S /application/mysql3307/logs/mysql.sock -p shutdown

nohup /application/mysql3307/bin/mysqld_safe --defaults-file=/application/mysql3307/my.cnf --user=mysql &

3、查看binlog日志是否开启

mysql> show variables like 'log_%';

```
mysql> show master logs;
+------------------+-----------+
| Log_name | File_size |
+------------------+-----------+
| mysql-bin.000001 | 149 |
| mysql-bin.000002 | 4102 |
```

2、查看master状态，即最后（最新）一个binlog日志的编号名称，及其最后一个操作事件pos结束点(Position)值。

show master status;

```
[root@vm-002 mysql]# mysqlbinlog mysql-bin.000002
.....
# at 624
#160925 21:29:53 server id 1 end_log_pos 796 Query thread_id=3 exec_time=0 error_code=0
SET TIMESTAMP=1474810193/*!*/;
insert into member(`name`,`sex`,`age`,`classid`) values('wangshibo','m',27,'cls1'),
('guohuihui','w',27,'cls2')       #执行的sql语句
/*!*/;
# at 796
```

解释：

server id 1:数据库主机的服务号

end_log_pos 796 :sql结束时的pos节点

thread_id=11:线程号

e、也可根据时间点查看

/home/software/mysql-5.1.72-linux-x86_64-glibc23/bin/mysqlbinlog --no-defaults mysql-bin.000720 --start-datetime="2018-09-12 18:45:00" --stop-datetime="2018-09-12:18:47:00"

2、上面这种办法读取出binlog日志的全文内容比较多，不容易分辨查看到pos点信息，下面介绍一种更为方便的查询命令：

mysql> show binlog events [IN 'log_name'] [FROM pos] [LIMIT [offset,] row_count];

参数解释：

a、IN 'log_name':指定要查询的binlog文件名（不指定就是第一个binlog文件）

b、FROM
pos:指定从哪个pos起始点开始查起（不指定就是从整个文件首个pos点开始算）

c、LIMIT【offset】：偏移量(不指定就是0)

d、row_count :查询总条数（不指定就是所有行）

```
mysql> show master status;
+------------------+----------+--------------+------------------+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000003 | 106 | | |
```

也就是说，mysql-bin.000003是用来记录4:00之后对数据库的所有'增删改操作'。

3、早上九点上班了，由于业务的需求会对数据库进行各种'增删改'操作。

比如：在ops库下和ops1库下member表内插入、修改了数据等等：

先是早上进行插入数据：

```
insert into ops.member(`name`,`sex`,`age`,`classid`) values('yiyi','w',20,'cls1'),('xiaoer','m',22,'cls3'),('zhangsan','w',21,'cls5'),('lisi','m',20,'cls4'),('wangwu','w',26,'cls6');

insert into ops1.member(`name`,`sex`,`age`,`classid`) values('yiyi','w',20,'cls1'),('xiaoer','m',22,'cls3'),('zhangsan','w',21,'cls5'),('lisi','m',20,'cls4'),('wangwu','w',26,'cls6');
```

```
mysql> select * from member;
+----+-----------+-----+-----+---------+
| id | name | sex | age | classid |
+----+-----------+-----+-----+---------+
| 1 | wangshibo | m | 27 | cls1 |
| 2 | 小二 | w | 27 | cls2 |
| 3 | yiyi | w | 20 | cls1 |
| 4 | 李四 | m | 22 | cls3 |
| 5 | zhangsan | w | 21 | cls5 |
| 6 | lisi | m | 20 | cls4 |
```

5、在下午18:00的时候，悲剧莫名其妙的出现了！

手贱执行了drop语句，直接删除了ops1库！吓尿！

drop database ops;

drop database ops1;

再手残又创建了一个数据库ops2并插入数据

create database ops2;

use ops2;

CREATE TABLE IF NOT EXISTS `member` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT,`name` varchar(16) NOT NULL,`sex` enum('m','w') NOT NULL DEFAULT 'm',`age` tinyint(3) unsigned NOT NULL,`classid` char(6) DEFAULT NULL,PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;

insert into ops2.member(`name`,`sex`,`age`,`classid`) values('yiyi','w',20,'cls1'),('xiaoer','m',22,'cls3'),('zhangsan','w',21,'cls5'),('lisi','m',20,'cls4'),('wangwu','w',26,'cls6');

6、这种时候，一定不要慌张!!!

先仔细查看最后一个binlog日志，并记录下关键的pos点，到底是哪个pos点的操作导致了数据库的破坏（通常在最后几步）；

a、先备份一下最后一个binlog日志文件：

cd /application/mysql3306/mysql_data

cp -v mysql-bin.000004 /application/data/backup/

ls /application/data/backup/

```
mysql> show master status;
+------------------+----------+--------------+------------------+-------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------------+----------+--------------+------------------+-------------------+
| mysql-bin.000004 |     1262 |              |                  |                   |
+------------------+----------+--------------+------------------+-------------------+
1 row in set (0.00 sec)

mysql> flush logs;
Query OK, 0 rows affected (0.03 sec)

mysql> show master status;
+------------------+----------+--------------+------------------+-------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------------+----------+--------------+------------------+-------------------+
```

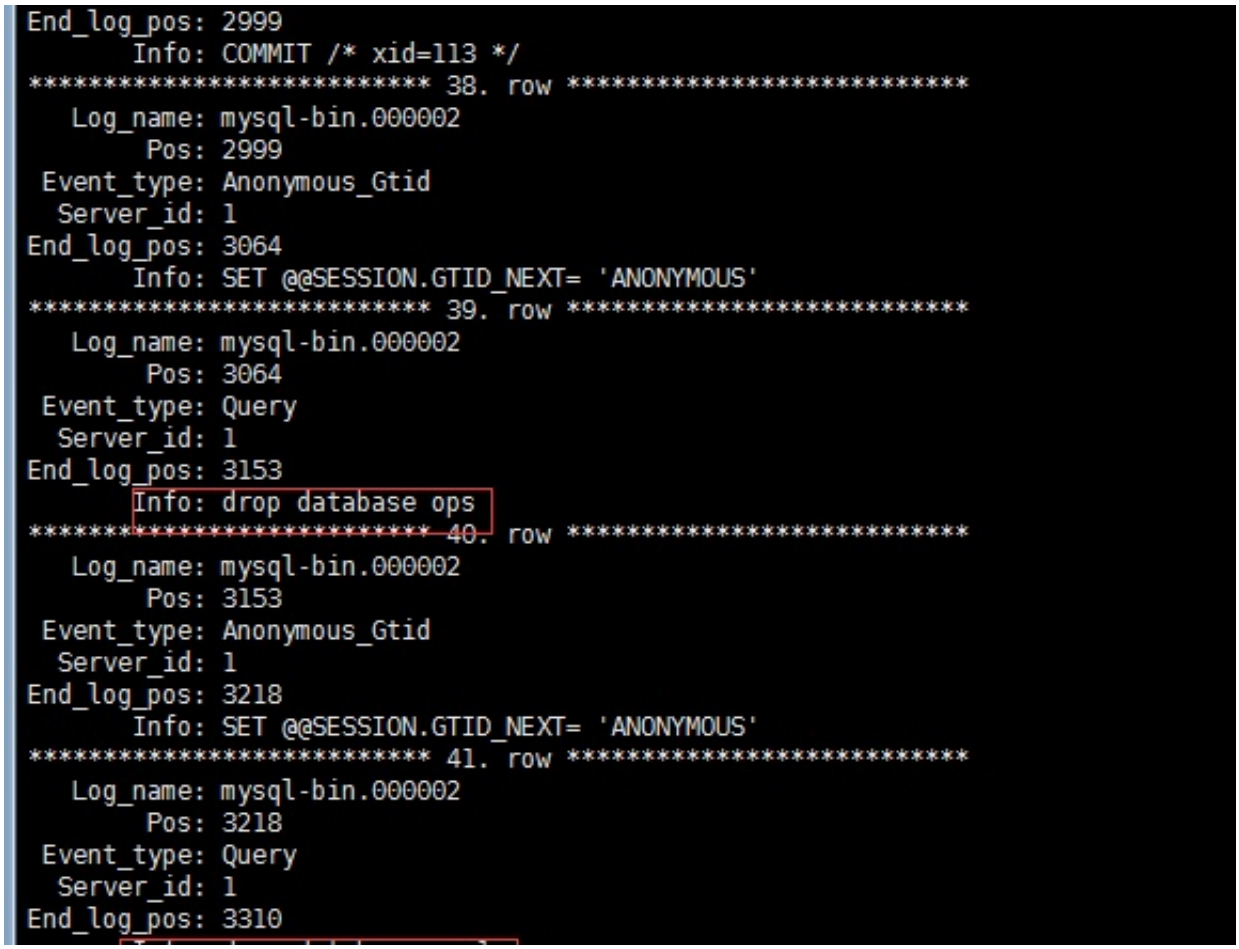
7、读取binlog日志的方法上面已经说到。

a、方法一：使用mysqlbinlog读取binlog日志：

/application/mysql3306/bin/mysqlbinlog
/application/mysql3306/mysql_data/mysql-bin.000004

b、登录服务器，并查看（推荐此种方法）

show binlog events in 'mysql-bin.000003';



通过分析，造成库ops数据破坏的pos点区间是介于3064-3153之间（这是按照日志区间的pos节点算的），造成库ops1库破坏的pos区间是介于3218-3310之间，只要恢复到相应pos点之前就可以了。

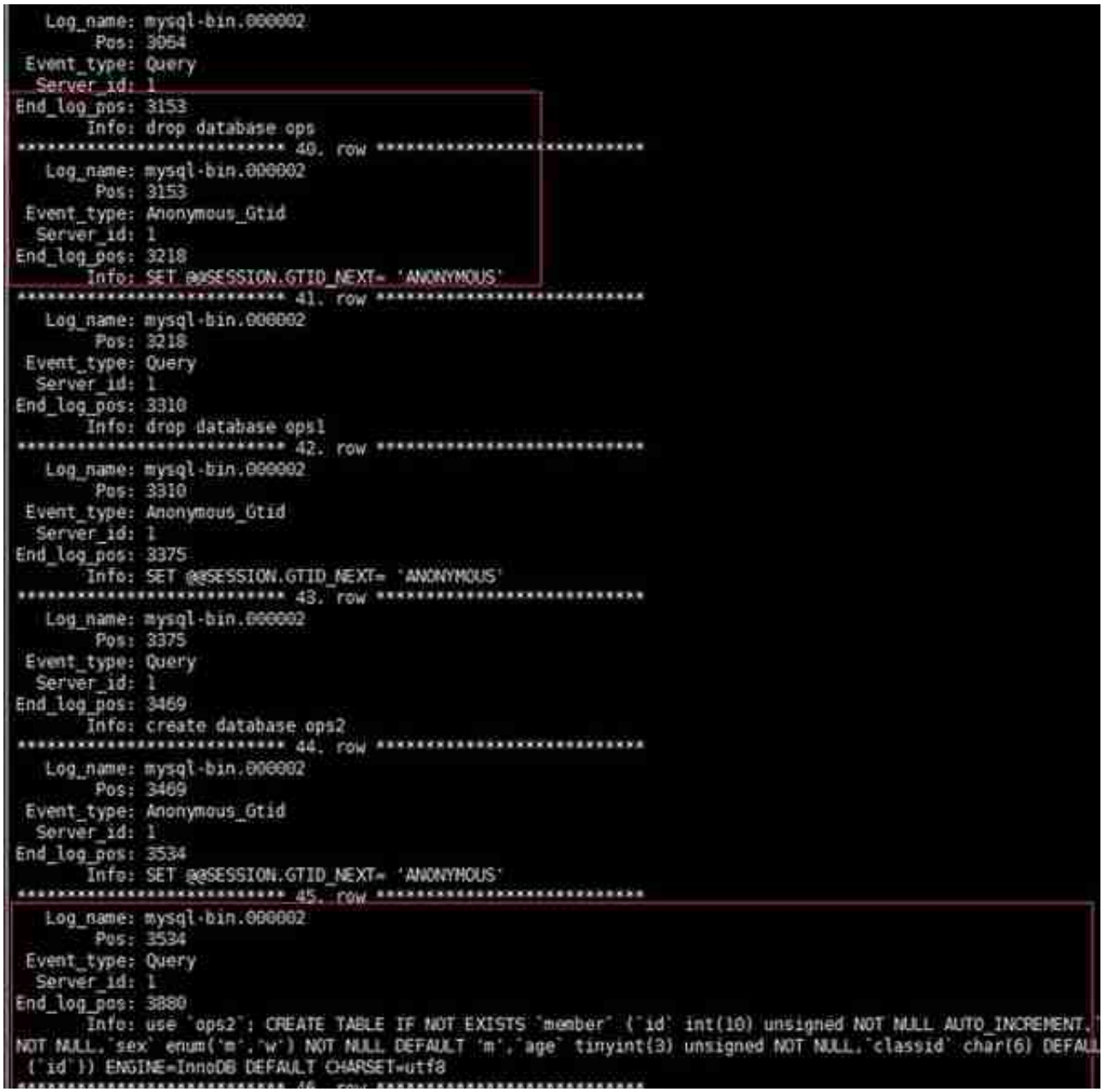
8、先把凌晨4点全备的数据恢复（建议另起一个库，等恢复成功后再替换掉当前库即可）

cd /application/data/backup/

gzip -d ops_2018-09-11.sql.gz

/application/mysql3307/bin/mysql -uroot -S /application/mysql3307/logs/mysql.sock -p123456 <ops_2018-09-11.sql

这样就恢复了截止凌晨4:00前的备份数据了



/application/mysql3306/bin/mysqlbinlog --start-position=3153 --stop-position=3880 /application/mysql3306/mysql_data/mysql-bin.000002 | /application/mysql3307/bin/mysql -uroot -S /application/mysql3307/logs/mysql.sock -p123456 -v

h、此时后面创建的表member恢复回来了但是库ops1被删除了，因为在这中间有删除ops1库的操作，若想继续恢复后面表中插入的数据只需要以建表后的pos点为开始点即可恢复除删库之外的所有数据。

/application/mysql3306/bin/mysqlbinlog --start-position=3880 /application/mysql3306/mysql_data/mysql-bin.000002 |

/application/mysql3307/bin/mysql -uroot -S /application/mysql3307/logs/mysql.sock -p123456 -v

10、另外：也可指定时间节点区间恢复（部分恢复）：按时间恢复需要mysqlbinlog命令读binlog日志内容，找时间节点。

a、/application/mysql3306/bin/mysqlbinlog /application/mysql3306/mysql_data/mysql-bin.000002

/application/mysql3306/bin/mysqlbinlog --start-datetime="2018-09-12 10:38:45" /application/data/backup/mysql-bin.000002 | /application/mysql3307/bin/mysql -uroot -S /application/mysql3307/logs/mysql.sock -p123456 -v

c、基本原理和通过pos点恢复差不多。

六、总结：

所谓恢复，就是让mysql将保存在binlog日志中指定段落区间的sql语句逐个重新执行一次而已。