MySQL备份恢复

1. Binlog日志深入分析

1.1. Binlog记录模式及参数配置

- DDL：全部记录，定义语言
- DML：除select以外都会记录

```
log_bin=mysql-bin
binlog_format=statement|row
```

mysql-bin是basename，mysql-bin-000001.log

Binlog有三种记录模式

- statement：SBR：delete from mytable 、update（基于一个简单的回放）
  - create table
  - insert
  - update
  - delete
  - insert
  - update table set a=1

先查看了一下日志

```
mysql> show binlog events in 'mysql-bin.000001';
```

```
mysqlbinlog --start-position=100 --stop-position=120 --database=mydb mysql-bin.000001 > mysql.sql
```

statement里面面只有操作语句是非注释的，其他的说明都是#注释的

看我们的binlog日志大小

```
show variables like '%binlog_size%';  #如果一个事务超过binlon大小不会写入下一个
max_binlog_size=1024m  #每个binlog日志文件大小
expire_logs_days=7  #binlog的过期时间
binlog_cache_size=32768     DML操作不频繁 <=1m，DML频繁且事务大 2-4m
max_binlog_cache_size     32位4G，64位16P
```

```
mysql> flush logs; #生成一个新binlog
mysql> show binary logs; #查看系统binlog数
```

如果是mysqldump

1、找到这个表最初的记录表结构，和当时的数据，把这个数据insert全部拿出来

2、insert into，备份的时间点和出事的那个阶段咋办?

- row：RBR：update、delete=10，展示出10条更改前和更改后的语句

这个时候使用show binlog来查看已经看不到语句

```
mysql> show binlog events in 'mysql-bin.000003'; #?是否还有用
```

```
mysqlbinlog --base64-output=decode-rows -v mysql-bin.000003
```

```
### UPDATE `mydb`.`ad_user`
### WHERE
###   @1=1
###   @2='arry'
###   @3='777888'
### SET
###   @1=1
###   @2='arry'
###   @3='123456'
### UPDATE `mydb`.`ad_user`
### WHERE
###   @1=2
###   @2='gavin'
###   @3='777888'
### SET
###   @1=2
###   @2='gavin'
###   @3='123456'
### UPDATE `mydb`.`ad_user`
### WHERE
###   @1=3
###   @2='coding'
###   @3='777888'
### SET
###   @1=3
###   @2='coding'
###   @3='123456'
```

```
mysqlbinlog --base64-output=decode-rows -vv mysql-bin.000003  #增加数据类型了
```

```
### DELETE FROM `mydb`.`ad_user`
### WHERE
###   @1=1
###   @2='arry'
###   @3='123456'
### DELETE FROM `mydb`.`ad_user`
### WHERE
###   @1=2
###   @2='gavin'
###   @3='123456'
### DELETE FROM `mydb`.`ad_user`
### WHERE
###   @1=3
###   @2='coding'
###   @3='123456'
```

- mixed：MBR：90%都是statement的模式

> 90%的语句都是以statement模式进行的

1.2. Binlog日志的正确打开方式

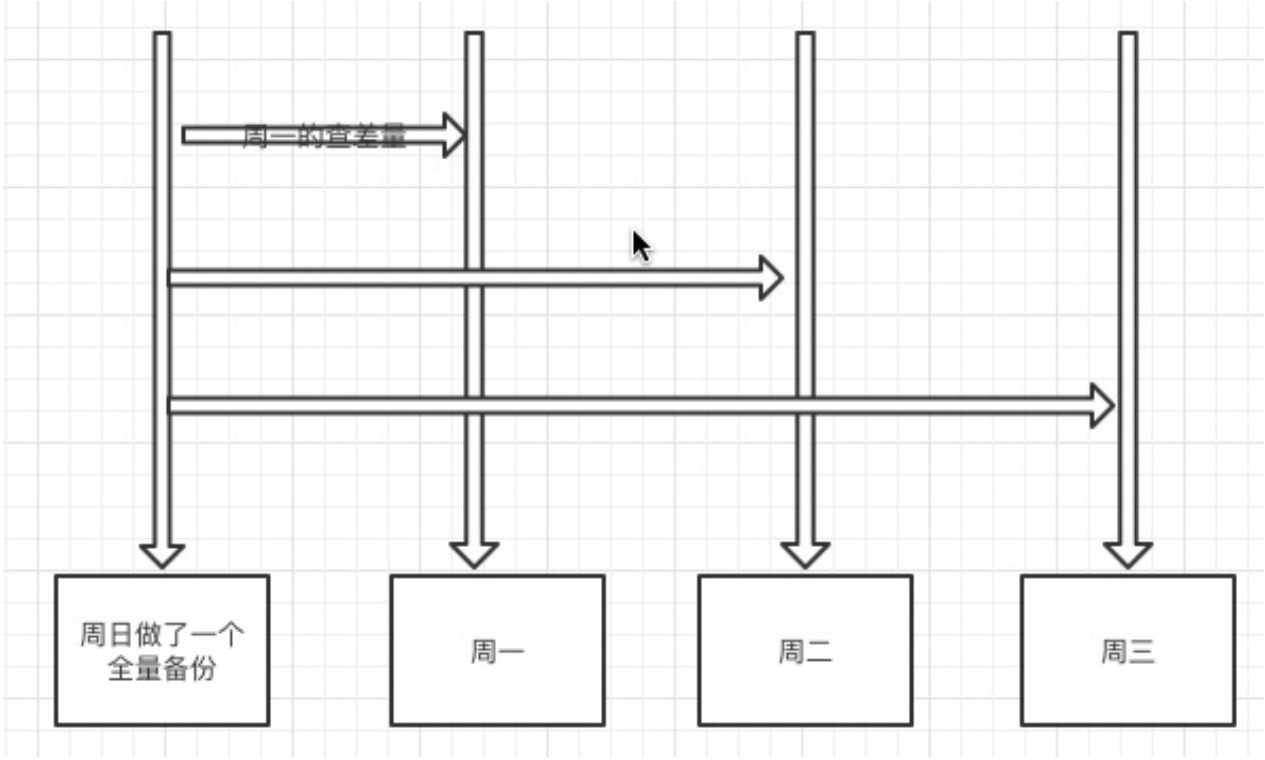
```
mysql> show binlog events in 'mysql-bin.000002'
mysql> show binlog events [IN 'log_name'][FROM pos][LIMIT [offset,] row_count];
```

```
mysqlbinlog --no-defaults --database=mydb --base64-output=decode-rows -v --start-position=123 --stop-position=456 mysql-bin.000002
mysqlbinlog --no-defaults --database=mydb --base64-output=decode-rows -v --start-datetime='2019-12-11 16:30:00' --stop-datetime='2019-12-11 16:31:00' mysql-bin.000003
```

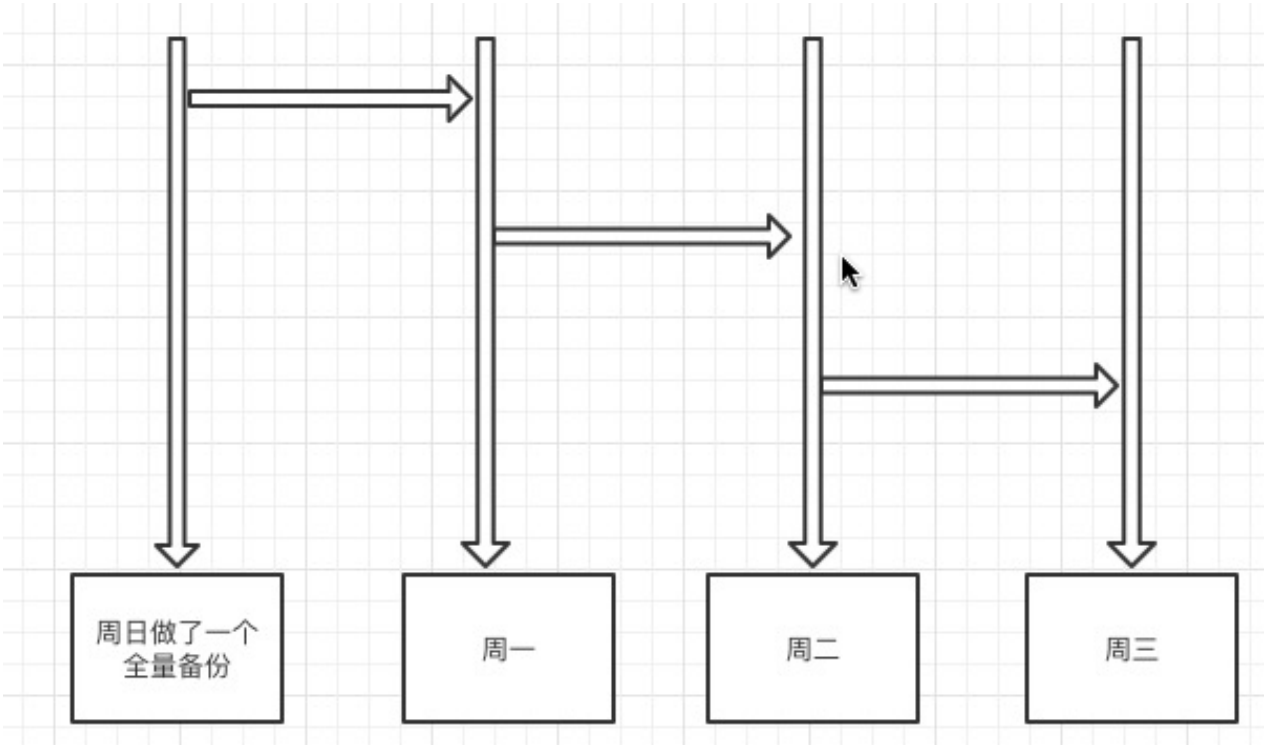
Log_name	Pos	Event_type	Server_id	End_log_pos	Info
mysql-bin.000003	4	Format_desc	1	120	Server ver: 5.6.46-log, Binlog ver: 4
mysql-bin.000003	120	Query	1	192	BEGIN
mysql-bin.000003	192	Table_map	1	248	table_id: 71 (mydb.ad_user)
mysql-bin.000003	248	Update_rows	1	392	table_id: 71 flags: STMT_END_F
mysql-bin.000003	392	Xid	1	423	COMMIT /* xid=14 */
mysql-bin.000003	423	Query	1	495	BEGIN
mysql-bin.000003	495	Table_map	1	551	table_id: 71 (mydb.ad_user)
mysql-bin.000003	551	Delete_rows	1	640	table_id: 71 flags: STMT_END_F
mysql-bin.000003	640	Xid	1	671	COMMIT /* xid=19 */
mysql-bin.000003	671	Stop	1	694	

2. 对备份的正确认识

- 全量备份：对应时间的数据是全量的一个备份
- 差异备份：周日做了一次



- 增量备份



- 时间点恢复

上面三个备份节点都是一个定时的数据补偿，在定时备份完成后至任意备份时间节点前，这段时间出现问题需要Binlog能做的事情了

- 热备

数据库的读写操作均可正常进行，是要通过备份工具，myisam引擎不支持热备，innoDB支持热备

- 温备

数据库只能进行读操作，不能进行写操作

- 冷备

要让数据库停机

- 物理备份

直接copy数据文件

- 逻辑备份

将数据库里的数据导出进行备份的方式就是逻辑备份

2.1. MySQL常用的备份工具

- mysqldump
  - mysql自带的备份工具，是逻辑备份
  - innodb可以使用mysqldump进行热备
  - myisam可以使用mysqldump进行温备
  - 如果数据量较小可以使用
- xtrabackup
  - Percona提供
  - 是一种物理备份工具
  - 支持完全备份，差异备份，增量备份
- select语句直接备份
  - select * from a into outfile '/usr/local/a.bak'
- cp命令
  - 只能进行冷备

2.2 数据一致性的理解

- 数据一致性

  热备：数据库还依旧可以读写

  4:00 进行定时备份，假设你的数据非常多，需要备份10-20分钟

  小刘账户余额在4点有200元，4点10分的时候，他转出了50元

  假如在4点10分前还没有备份到余额表，4点11分开始备份余额表
- 在备份场景下如何保证数据一致性
  - 第一种方式，在备份的时候给所有表加锁，只能读不能写
    - 如果锁表的时候可以把写入的数据先放入MQ或缓存，待备份完成补偿进数据库，还有就是要考虑及时读的问题
  - 第二种方式：在备份开始的时候就对数据库的所有数据进行一个“快照”，快照记录了开始备份的那一刻的数据状态

2.3 使用mysqldump备份

- 缺点：当数据位浮点型，会出现精度丢失
- 如果要进行并行备份可以使用mydumper/myloader

```
mysqldump -uroot -p123456 --databases mydb > mydb.sql #导出带数据库的备份脚本
mysqldump -uroot -p123456 --databases mydb ad_user > mydb.sql #导出数据库指定表
mysqldump -uroot -p123456 --all-databases > mydb.sql #导出所有数据库
mysqldump -uroot -p123456 -d mydb > mydb.sql #导出数据库的所有表结构
mysqldump -uroot -p123456 -d mydb ad_user > mydb.sql #导出数据库的某个表结构
```

- --master-data

  某个时间全量备份：每天晚上4点-中间12点数据挂了-明天晚上4点之间这段时间就需要时间点恢复

  前天晚上4点全量+4点-12点的binlog（如果知道4点备份的那个position）

  能够在我们导出数据的时候在我们的脚本里带上全量结束的position

  --master-data=0|1|2

  1：如果主库被删除了，从库也会被删除，拿着备份文件去从库告知从库执行完从什么位置开始同步

  2：只记录备份的position，可以用这个位置快速导出binlog的语句

- --flush-logs

  在备份的那个时点新建一个binlog

- 其他常用选项
  - --routines： 存储过程
  - --triggers： 触发器
  - --events： 事件

2.4 不同存储引擎下如何进行备份

- innodb

  热备：需要在mysqldump里加入一个参数：--single-transaction

  会基于备份生成一个独立的事务，专门进行对应时点快照数据处理的

```
mysqldump -uroot -p123456 --master-data=2 --single-transaction --routines --triggers --events --databases mydb > mydb.sql
```

- myisam

  温备：因为这个引擎不能支持事务，要保证数据一致性要锁表：--lock-tables

```
mysqldump -uroot -p123456 --master-data=2 --lock-tables --routines --triggers --events --databases mydb > mydb.sql
```

--lock-all-tables #配置导出所有数据库 --all-databases