# 概述

Binlog它记录了所有的DDL和DML(除了数据查询语句)语句，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。

一般来说开启二进制日志大概会有1%的性能损耗(参见MySQL官方中文手册 5.1.24版)。

二进制有两个最重要的使用场景:

1、MySQL Replication在Master端开启binlog，Mster把它的二进制日志传递给slaves来达到master-slave数据一致的目的。

2、自然就是数据恢复了，通过使用mysqlbinlog工具来使恢复数据。

二进制日志包括两类文件：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，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)语句事件。

# 日志格式

binlog有三种格式：Statement, Row和Mixed.

基于SQL语句的复制（statement-based replication, SBR）

基于行的复制（row-based replication, RBR）

混合模式复制（mixed-based replication, MBR）

## Statement

每一条会修改数据的sql都会记录在binlog中。

优点：不需要记录每一行的变化，减少了binlog日志量，节约了IO, 提高了性能。

缺点：由于记录的只是执行语句，为了这些语句能在slave上正确运行，因此还必须记录每条语句在执行的时候的一些相关信息，以保证所有语句能在slave得到和在master端执行的时候相同的结果。另外mysql的复制，像一些特定函数的功能，slave可与master上要保持一致会有很多相关问题。

相比row能节约多少性能与日志量，这个取决于应用的SQL情况，正常同一条记录修改或者插入row格式所产生的日志量还小鱼statement产生的日志量，但是考虑到如果带条件的update操作，以及整表删除，alter表等操作，row格式会产生大量日志，因此在考虑是否使用row格式日志时应该根据应用的实际情况，其所产生的日志量会增加多少，以及带来的IO性能问题。

## Row

5.1.5版本的MySQL才开始支持row level的复制,它不记录sql语句上下文相关信息，仅保存哪条记录被修改。

优点： binlog中可以不记录执行的sql语句的上下文相关的信息，仅需要记录那一条记录被修改成什么了。所以row的日志内容会非常清楚的记录下每一行数据修改的细节。而且不会出现某些特定情况下的存储过程，或function，以及trigger的调用和触发无法被正确复制的问题.

缺点:所有的执行的语句当记录到日志中的时候，都将以每行记录的修改来记录，这样可能会产生大量的日志内容。

新版本的MySQL中对row level模式也被做了优化，并不是所有的修改都会以row level来记录，像遇到表结构变更的时候就会以statement模式来记录，如果sql语句确实就是update或者delete等修改数据的语句，那么还是会记录所有行的变更。

## Mixed

从5.1.8版本开始，MySQL提供了Mixed格式，实际上就是Statement与Row的结合。

在Mixed模式下，一般的语句修改使用statment格式保存binlog，如一些函数，statement无法完成主从复制的操作，则采用row格式保存binlog，MySQL会根据执行的每一条具体的sql语句来区分对待记录的日志形式，也就是在Statement和Row之间选择一种。

# 操作命令

## 开启

开启binlog日志：

1、vi编辑打开mysql配置文件

# vi /usr/local/mysql/etc/my.cnf

在[mysqld] 区块

设置/添加 log-bin=mysql-bin 确认是打开状态(值 mysql-bin 是日志的基本名或前缀名)；

重启mysqld服务使配置生效

# pkill mysqld

# /usr/local/mysql/bin/mysqld\_safe --user=mysql &

2、也可登录mysql服务器，通过mysql的变量配置表，查看二进制日志是否已开启 单词：variable[ˈvɛriəbəl] 变量

登录服务器

# /usr/local/mysql/bin/mysql -uroot -p123456

mysql> show variables like 'log\_%';

## 查看所有binlog日志列表

mysql> show master logs;

## 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

mysql> show master status;

## 刷新log日志

自此刻开始产生一个新编号的binlog日志文件

mysql> flush logs;

注：每当mysqld服务重启时，会自动执行此命令，刷新binlog日志；在mysqldump备份数据时加 -F 选项也会刷新binlog日志；

## 重置(清空)所有binlog日志

mysql> reset master;

# 解析工具