1. 概述:

二进制日志以一种更有效的格式，并且是事务安全的方式包含更新日志中可用的所有信息。

二进制日志包含了所有更新了数据或者已经潜在更新了数据（例如，没有匹配任何行的一个DELETE）的所有语句。语句以“事件”的形式保存，它描述数据更改。

二进制日志还包含关于每个更新数据库的语句的执行时间信息。它不包含没有修改任何数据的语句。如果你想要记录所有语句（例如，为了识别有问题的查询），你应使用一般查询日志。

二进制日志的主要目的是在恢复使能够最大可能地更新数据库，因为二进制日志包含备份后进行的所有更新。

二进制日志还用于在主复制服务器上记录所有将发送给从服务器的语句。

运行服务器时若启用二进制日志则性能大约慢1%。但是，二进制日志的好处，即用于恢复并允许设置复制超过了这个小小的性能损失。

mysqld在每个二进制日志名后面添加一个数字扩展名。每次你启动服务器或刷新日志时该数字则增加。如果当前的日志大小达到max\_binlog\_size，还会自动创建新的二进制日志。如果你正使用大的事务，二进制日志还会超过max\_binlog\_size：事务全写入一个二进制日志中，绝对不要写入不同的二进制日志中。

<http://dev.mysql.com/doc/refman/5.1/zh/database-administration.html#binary-log>

<http://chinaxxren.iteye.com/blog/796739>

1. MySQL binlog 格式（Mixed，Statement，Row Level）

Row 模式：日志中会记录成每一行数据被修改的形式，然后在slave端再对相同的数据进行修改。

Statement 模式:每一条会修改数据的sql都会记录到 master的bin-log中。slave在复制的时候sql进程会解析成和原来master端执行过的相同的sql来再次执行。

在Mixed模式下，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。新版本中的Statment level还是和以前一样，仅仅记录执行的语句。而新版本的MySQL中队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。

-- 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)。

3.bin-log配置：

vi /etc/my.cnf

#开启二进制日志

log-bin=mysql-bin

binlog\_format=mixed

#二进制日志有效时间

expire\_logs\_days = 30

也可以在运行时动态修改binlog的格式。例如

mysql> SET SESSION binlog\_format = 'STATEMENT';

mysql> SET SESSION binlog\_format = 'ROW';

mysql> SET SESSION binlog\_format = 'MIXED';

<http://shanchao7932297.blog.163.com/blog/static/1363624201011133443867/>

mysql> show binary logs;

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

| Log\_name | File\_size |

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

| mysql-bin.000001 | 1932 |

| mysql-bin.000002 | 126 |

| mysql-bin.000003 | 107 |

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

3 rows in set (0.00 sec)

<http://www.361way.com/mysqlbinlog-type/934.html>

mysql> show variables like '%binlog%';

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

| Variable\_name | Value |

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

| binlog\_cache\_size | 32768 |

| binlog\_direct\_non\_transactional\_updates | OFF |

| binlog\_format | MIXED |

| binlog\_stmt\_cache\_size | 32768 |

| innodb\_locks\_unsafe\_for\_binlog | OFF |

| max\_binlog\_cache\_size | 18446744073709547520 |

| max\_binlog\_size | 1073741824 |

| max\_binlog\_stmt\_cache\_size | 18446744073709547520 |

| sync\_binlog | 0 |

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

9 rows in set (0.00 sec)

mysql> purge binary logs to 'mysql-bin.000001';

Query OK, 0 rows affected (0.10 sec)