Mysql 二进制日志管理

MySQL二进制日志是以事务安全的方式包含更新日志中可用的所有信息。它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。

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。

### 打开mysql二进制日志

编辑mysql配置文件：/etc/my.cnf（windows下是my.ini），将“bin-log”一行前面的注释去掉，默认是：#log-bin=mysql-bin，（没有的话，自己手动添加一行），等号后面的mysql-bin是生成日志的路径和文件名，这里表示在mysql的data目录下生成，你也可以自己改路径，如log-bin=/data/mysql/log/mysql-bin。

还可以设置以下选项：

binlog-do-db= *db\_name* #将*db\_name*更新记录到二进制日志中

binlog-ignore-db=*db\_name #*不将*db\_name*更新保存到二进制日志中

max\_binlog\_size= 104857600 #每个日志文件最大100M

保存，重启mysql。

重启后看到有生成类似mysql-bin.000001、mysql-bin.000002的文件就说明已经成功打开了二进制日志。

### 管理二进制日志文件

1、查看当前已有的二进制日志文件，执行sql“SHOW BINLOG EVENTS \G”或“SHOW MASTER LOGS”：

mysql> SHOW BINLOG EVENTS \G;

mysql> **SHOW MASTER LOGS；**

2、清除二进制日志文件：  
mysql> PURGE { MASTER|BINARY } LOGS TO 'log\_name';mysql> PURGE { MASTER|BINARY } LOGS BEFORE 'datetime';例如：   
mysql> PURGE MASTER LOGS TO 'mysql-bin.000001';mysql> PURGE MASTER LOGS BEFORE '2014-08-12 23:00:00';

另一种清除二进制日志方法：  
mysql > flush logs;mysql > reset master;

### 从二进制日志文件恢复数据

mysqlbinlog：用于处理二进制日志文件的实用工具；可以用它来恢复从指定的时间点开始 (例如，从你最后一次备份)直到现在或另一个指定的时间点的数据。

使用方法：mysqlbinlog [options] log-files。

mysqlbinlog有很多选项，具体可以使用mysqlbinlog –help查看:

如：bin>mysqlbinlog --help

*（如果没有将mysqlbinlog设置成系统命令，linux下可以用find / -name "mysqlbinlog"搜索出mysqlbinlog的位置，windows下在mysql安装目录的bin文件夹下）*

*如果执行mysqlbinlog时出现类似unknown variable 'default-character-set=utf8'错误提示，是因为在my.cnf（或my.ini）中的client选项组中添加了default-character-set=utf8，可以有两种办法解决这个bug：*

*第一种，执行时加上--no-defaults选项；如：mysqlbinlog --no-defaults --help。*

*第二种，在使用mysqlbinlog工具时把my.cnf（或my.ini）中client选项组中的default-character-set=utf8注释掉。注意要记得在不在使用mysqlbinlog工具时取消这个注释。*

**部分常用选项说明：**

**--start-datetime=datetime**　#从二进制日志中第1个日期时间等于或晚于datetime参量的事件开始读取, 格式应符合DATETIME或TIMESTAMP数据类型。datetime值是相对于运行**mysqlbinlog**的机器上的本地时区时间。

**--stop-datetime=datetime**　#从二进制日志中第1个日期时间等于或晚于datetime参量的事件停止读取，描述同start-datetime。

**--start-position=N** #从二进制日志中第1个位置等于N参量时的事件开始读。

·       **--stop-position=N**　 #从二进制日志中第1个位置等于N参量时的事件停止读。

示例：

1. 从mysql-bin.000001中恢复从2010-01-01 01:00:00开始之后的数据：

bin> mysqlbinlog --start-datetime="2010-01-01 01:00:00" mysql-bin.000001 | mysql -uroot -proot

1. 从mysql-bin.000001中恢复从2010-01-01 01:00:00到2010-01-02 01:00:00之间的数据：

bin> mysqlbinlog --start-datetime="2010-01-01 01:00:00" --stop-datetime="2010-01-02 01:00:00" mysql-bin.000001 | mysql -uroot -proot

1. 从多个日志文件中恢复数据：

bin> mysqlbinlog mysql-bin.000001 mysql-bin.000002 | mysql -uroot -proot

或

bin> ls -t -r -1 mysql-bin.\* | xargs mysqlbinlog | mysql -uroot -proot

1. 其他方式恢复：

bin> mysqlbinlog mysql-bin.000001 > /temp/binlog.sql #符号>表示创建新的文件，如果原来已经存在，则会覆盖掉原文件。

bin> mysqlbinlog --stop-datetime="2010-01-02 01:00:00" mysql-bin.000002 >> /temp/binlog.sql #符号>>表示追加到内容后面。

bin> mysql -uroot -proot -e “source /temp/binlog.sql”　#这步必须执行

# [Mysql清理二进制日志的技巧](http://blog.csdn.net/cdefg198/article/details/7063524)

分类： [mysql](http://blog.csdn.net/cdefg198/article/category/864410)2011-12-12 14:40 1480人阅读 [评论](http://blog.csdn.net/cdefg198/article/details/7063524#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/cdefg198/article/details/7063524#report)

1：二进制日志

二进制日志记录了所有的DDL（数据定义语言）语句和DML（数据操作语言）语句，但是不记录包括数据查询的语句。语句以“事件”的形式保存，它描述了数据的更改过程，此日志对于灾难时的数据恢复起着极其重要的作用

2：日志的位置和格式

当用—log-bin[=file\_name]选项启动时，mysqld将包含所有更新数据的SQL命令写入日志文件。如果没有给出file\_name值，默认名为主机名后面跟\_bin，如果给出了文件名，但没有包含路劲，则文件默认被写入参数DATADIR（数据目录）指定的目录

3：日志的读取

由于日志以二进制的方式存储，不能直接读取，需要用mysqlbinlog工具来查看，语法如下：

#mysqlbinlog log\_file

4：日志的删除

对于比较繁忙的OLTP系统，由于每天生产日志量大，这些日志如果长时间不清理，将会对磁盘空间带来很大的浪费，因此，定期删除日志是DBA维护Mysql数据库的一个重要工作内容，下面将介绍几种删除日志的常见方法

（1）：

执行“reset master;”命令，该命令将删除所有二进制日志，新日志的编号从“000001” 开始，命令如下

Mysql>reset master;

（2）：

执行“Purge master logs to ‘mysql-bin.\*\*\*\*\*’” 命令，该命令将删除“\*\*\*\*\*” 编号之前的所有日志，下列中删除了“mysql-bin.000001”之前编号的所有日志

Mysql>purge master logs to ‘mysql-bin.000015;

从结果中发现，编号000015之前的所有日志都已经删除

（3）：

执行“purge master logs before ‘yyyy-mm-dd hh24:min:ss’”命令，该命令将删除日期为“yyyy-mm-dd hh24:mi:ss”之前产生的所有日志，下列中删除了日期在“2010-05-22 01:00:00”之前的所有日志

Mysql>purge master logs before ‘2010-05-22 01:00:00’’;

（4）：

设置参数—expire\_logs\_days=#(days)，此参数的含义是设置日志的过期天数，过来指定的天数后日志将会被自动删除，这样将有利于减少DBA管理日志的工作量。

#vi /etc/my.cnf

[mysqld]

expire\_logs\_days=15

这样，15天前的日志都会被删除，系统自动删除