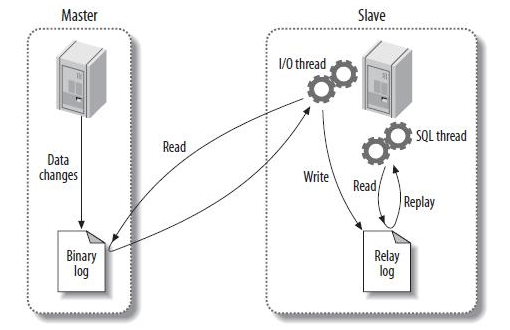
**[MySQL复制的概述、安装、故障、技巧、工具](http://blog.netoearth.com/html/201104/mysql%e5%a4%8d%e5%88%b6%e7%9a%84%e6%a6%82%e8%bf%b0%e3%80%81%e5%ae%89%e8%a3%85%e3%80%81%e6%95%85%e9%9a%9c%e3%80%81%e6%8a%80%e5%b7%a7%e3%80%81%e5%b7%a5%e5%85%b7.htm" \o "Permanent Link to MySQL复制的概述、安装、故障、技巧、工具)**

*On 2011年04月5日, in* [*database*](http://blog.netoearth.com/html/category/database)*, by netoearth*

同[MongoDB](http://www.mongodb.org/)，[Redis](http://redis.io/)这样的NoSQL数据库的复制相比，[MySQL复制](http://dev.mysql.com/doc/refman/5.5/en/replication.html)显得相当复杂！

**概述**

首先主服务器把数据变化记录到主日志，然后从服务器通过I/O线程读取主服务器上的主日志，并且把它写入到从服务器的中继日志中，接着SQL线程读取中继日志，并且在从服务器上重放，从而实现MySQL复制。具体如下图所示：

[](http://huoding.com/wp-content/uploads/2011/03/replication.png)MySQL复制

整个过程反映到从服务器上，对应三套日志信息，可在从服务器上用如下命令查看：

mysql> SHOW SLAVE STATUS;

* Master\_Log\_File & Read\_Master\_Log\_Pos：下一个传输的主日志信息。
* Relay\_Master\_Log\_File & Exec\_Master\_Log\_Pos：下一个执行的主日志信息。
* Relay\_Log\_File & Relay\_Log\_Pos：下一个执行的中继日志信息。

理解这些信息对于解决故障至关重要，后文会详细阐述。

**安装**

先在主服务器上创建复制账号：

mysql> GRANT REPLICATION SLAVE ON \*.\*

TO '<SLAVE\_USER>'@'<SLAVE\_HOST>'

IDENTIFIED BY '<SLAVE\_PASSWORD>';

注：出于安全性和灵活性的考虑，不要把root等具有[SUPER](http://dev.mysql.com/doc/refman/5.5/en/privileges-provided.html#priv_super)权限用户作为复制账号。

然后设置主服务器配置文件（缺省：/etc/my.cnf）：

[mysqld]

server\_id = 100

log\_bin = mysql-bin

log\_bin\_index = mysql-bin.index

sync\_binlog = 1

innodb\_flush\_log\_at\_trx\_commit = 1

innodb\_support\_xa = 1

注：一定要保证主从服务器各自的server\_id唯一，避免冲突。

注：如果没有指定[log\_bin](http://dev.mysql.com/doc/refman/5.5/en/server-system-variables.html#sysvar_log_bin)的话，缺省会使用主机名作为名字，如此一来一旦主机名发生改变，就会出问题，所以推荐指定log\_bin（从服务器的relay\_log存在一样的问题）。

注：[sync\_binlog](http://dev.mysql.com/doc/refman/5.5/en/replication-options-binary-log.html#sysvar_sync_binlog)，[innodb\_flush\_log\_at\_trx\_commit](http://dev.mysql.com/doc/refman/5.5/en/innodb-parameters.html#sysvar_innodb_flush_log_at_trx_commit)，[innodb\_support\_xa](http://dev.mysql.com/doc/refman/5.5/en/innodb-parameters.html#sysvar_innodb_support_xa)三个选项都是出于安全目的设置的，不是复制的必须选项。

接着设置从服务器配置文件（缺省：/etc/my.cnf）：

[mysqld]

server\_id = 200

log\_bin = mysql-bin

log\_bin\_index = mysql-bin.index

relay\_log = mysql-relay-bin

relay\_log\_index = mysql-relay-bin.index

read\_only = 1

skip\_slave\_start = 1

log\_slave\_updates = 1

注：如果用户有SUPER权限，则[read\_only](http://dev.mysql.com/doc/refman/5.5/en/server-system-variables.html#sysvar_read_only)无效。

注：有了[skip\_slave\_start](http://dev.mysql.com/doc/refman/5.5/en/replication-options-slave.html#option_mysqld_skip-slave-start)，除非使用[START SLAVE](http://dev.mysql.com/doc/refman/5.5/en/start-slave.html)命令，否则从服务器不会开始复制。

注：设置[log\_slave\_updates](http://dev.mysql.com/doc/refman/5.5/en/server-system-variables.html#sysvar_log_slave_updates)，让从服务器记录日志，有助于在必要时把从切换成主。

下面最重要的步骤是如何克隆一份主服务器的数据：

如果数据库使用的是MyISAM表类型的话，可按如下方式操作：

shell> mysqldump --all-databases --master-data=1 > data.sql

注：[master-data](http://dev.mysql.com/doc/refman/5.5/en/mysqldump.html#option_mysqldump_master-data)选项缺省会打开[lock-all-tables](http://dev.mysql.com/doc/refman/5.5/en/mysqldump.html#option_mysqldump_lock-all-tables)，并写入[CHANGE MASTER TO](http://dev.mysql.com/doc/refman/5.5/en/change-master-to.html)语句。

如果数据库使用的是InnoDB表类型的话，则应该使用[single-transcation](http://dev.mysql.com/doc/refman/5.5/en/mysqldump.html#option_mysqldump_single-transaction)：

shell> mysqldump --all-databases --single-transaction --master-data=1 > data.sql

有了数据文件，传输到从服务器上并导入：

shell> mysql < data.sql

如果数据量很大的话，mysqldump会非常慢，此时直接拷贝数据文件能节省不少时间：

在拷贝之前要先锁定数据，然后再获得相关的日志信息（FILE & POSITION）：

mysql> FLUSH TABLES WITH READ LOCK;

mysql> SHOW MASTER STATUS;

接下来拷贝数据文件时，如果是MyISAM表类型的话，直接拷贝即可；如果是InnoDB表类型的话，一定要先停止MySQL服务再拷贝，否则拷贝文件可能无法使用。把拷贝的数据文件直接复制到从服务器的数据目录。

最后还需要再指定等一下日志信息：

mysql> CHANGE MASTER TO

MASTER\_HOST='<MASTER\_HOST>',

MASTER\_USER='<SLAVE\_USER>',

MASTER\_PASSWORD='<SLAVE\_PASSWORD>',

MASTER\_LOG\_FILE='<FILE>',

MASTER\_LOG\_POS=<POSITION>;

注：不要在my.cnf配置文件里设置MASTER\_USER和MASTER\_PASSWORD，因为最终生效的是CHANGE MASTER TO生成的master.info文件里的信息。

记得在从服务器上启动复制，并检查工作是否正常：

mysql> START SLAVE;

mysql> SHOW SLAVE STATUS;

如果IO线程和SQL线程都显示Yes，就可以感谢上帝了：

* Slave\_IO\_Running 对应：Master\_Log\_File & Read\_Master\_Log\_Pos
* Slave\_SQL\_Running 对应：Relay\_Master\_Log\_File & Exec\_Master\_Log\_Pos

如果显示No，则说明前面某些配置步骤出错，或者对应的日志文件有问题。

**故障**

问题：主从复制不止何故停止了，我该怎么办？

答案：复制错误多半是因为日志错误引起的，所以首先要搞清楚是主日志错误还是中继日志错误，从错误信息里一般就能判断，如果不能可以使用类似下面的mysqlbinlog命令：

shell> mysqlbinlog <MASTER\_BINLOG\_FILE> > /dev/null

shell> mysqlbinlog <SLAVE\_BINLOG\_FILE> > /dev/null

如果没有错误，则不会有任何输出，反之如果有错误，则会显示出来。

如果是主日志错误，则需要在从服务器使用[SET GLOBAL sql\_slave\_skip\_counter](http://dev.mysql.com/doc/refman/5.5/en/set-global-sql-slave-skip-counter.html)，如下：

mysql> SET GLOBAL sql\_slave\_skip\_counter = 1;

mysql> START SLAVE;

注：如果有多个错误，可能需要执行多次（提醒：主从服务器数据可能因此不一致）。

如果是中继日志错误，只要在从服务器使用SHOW SLAVE STATUS结果中的日志信息重新CHANGE MASTER TO即可，系统会抛弃当前的中继日志，重新下载：

mysql> CHANGE MASTER TO

MASTER\_LOG\_FILE='<Relay\_Master\_Log\_File>',

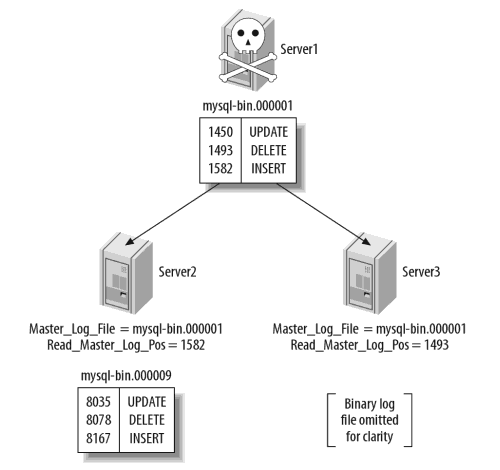
MASTER\_LOG\_POS=<Exec\_Master\_Log\_Pos>;

mysql> START SLAVE;

至于为什么使用的是Relay\_Master\_Log\_File & Exec\_Master\_Log\_Pos，参见概述。

问题：主服务器宕机了，如何把从服务器提升会主服务器？

答案：在一主多从的环境总，需选择数据最新的从服务器做新的主服务器。如下图所示：

[](http://huoding.com/wp-content/uploads/2011/04/promotion.png)提升从服务器为主服务器

在一主（Server1）两从（Server2,、Server3）的环境中，Server1宕机后，等到Server1和Server2把宕机前 同步到的日志都执行完，比较Master\_Log\_File和Read\_Master\_Log\_Pos就可以判断出谁快谁慢，因为Server2从 Server1同步的数据（1582）比Server3从Server1同步的数据（1493）新，所以应该提升Server2为新的主服务器，那么 Server3在CHANGE MASTER TO到Server2的时候应该使用什么样的参数呢？1582-1493=89，而Server2的最后的二进制日志位置是8167，所以答案是 8167-89=8078。

**技巧**

主从服务器中的表可以使用不同的表类型。比如主服务器可以使用InnoDB表类型，提供事务，行锁等高级特性，从服务器可以使用MyISAM表类 型，内存消耗少，易备份等优点。还有一个例子，一台主服务器如果同时带很多个从服务器的话，势必会影响其性能，此时可以拿出一台服务器作为从服务器代理， 使用BLACKHOLE表类型，只记录日志，不写数据，由它带多台从服务器，从而提升性能。

主从服务器中的表可以使用不同的键类型。比如主服务器用InnoDB，键用VARCHAR的话节省空间，从服务器使用MyISAM，键用CHAR提高速度，因为MyISAM有静态表一说。

主从服务器中的表可以使用不同的索引。主服务器主要用来应付写操作，所以除了主键和唯一索引等保证数据关系的索引一般都可以不加，从服务器一般用来应付读操作，所以可以针对查询特征设置索引，再进一步，不同的从服务器可以针对不同的查询设置不同的索引。

**工具**

有一些优秀的工具可以让你的复制工作得到事半功倍的效果，详细内容请参考各自文档：

* [Multi-Master Replication Manager for MySQL](http://mysql-mmm.org/)
* [Percona XtraBackup](http://www.percona.com/software/percona-xtrabackup/)
* [Maatkit](http://www.maatkit.org/)
* [Tungsten-replicator](http://code.google.com/p/tungsten-replicator/)

此外，Google Project Hosting里还有很多有趣的项目，可用[mysql+replication](http://code.google.com/hosting/search?q=label:mysql+replication)标签搜索。

说明：本文参考了下面列出的书籍中相关的内容：

[High Performance MySQL: Optimization, Backups, Replication, and More](http://www.amazon.com/High-Performance-MySQL-Optimization-Replication/dp/0596101716/)

[MySQL High Availability: Tools for Building Robust Data Centers](http://www.amazon.com/MySQL-High-Availability-Building-Centers/dp/0596807309/)