**Mycat 新一代Mysql分布式集群，大数据处理中间件，中国第一开源软件，欢迎志愿者参与：**

<http://code.google.com/p/opencloudb/>

本文档由Mycat志愿者团队提供，转载请注明，谢谢。

## [MySQL主从复制配置](http://www.blogjava.net/dongbule/archive/2010/08/22/329602.html)

Master-2(192.168.1.201)

)

Master-1(192.168.1.200)

)

Slave-2(192.168.1.203)

)

Slave-1(192.168.1.202)

)

Mysql的主从复制至少是需要两个Mysql的服务，当然Mysql的服务是可以分布在不同的服务器上，也可以在一台服务器上启动多个服务。

(1)首先确保主从服务器上的Mysql版本相同

(2)在主服务器上, 创建一个充许从数据库来访问的用户slave,密码为：123456

,然后使用REPLICATION SLAVE赋予权限,如:

mysql> GRANT REPLICATION SLAVE ON \*.\* TO 'slave'@'192.168.1.202' IDENTIFIED BY '123456';

Query OK, 0 rows affected (0.13 sec)

(3)修改主数据库的配置文件my.cnf(windows为my.ini),开启BINLOG，并设置server-id的值，修改之后必须重启Mysql服务

[mysqld]

server-id=1

log-bin = mysql-bin

(4)之后可以得到主服务器当前二进制日志名和偏移量，这个操作的目的是为了在从数据库启动后，从这个点开始进行数据的恢复

mysql> show master status;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

File: mysql-bin.000003 --------二进制日志名

Position: 243 -------偏移量

Binlog\_Do\_DB: --------要进行主从复制的数据库

Binlog\_Ignore\_DB: --------不进行主从复制的数据库

1 row in set (0.00 sec)

(5)好了，现在可以停止主数据的的更新操作，并生成主数据库的备份，我们可以通过mysqldump到处数据到从数据库，当然了，你也可以直接用cp命令将数据文件复制到从数据库去

注意在导出数据之前先对主数据库进行READ LOCK，以保证数据的一致性

mysql> flush tables with read lock;

Query OK, 0 rows affected (0.19 sec)

之后是mysqldump

mysqldump -h127.0.0.1 -p3306 -uroot -p 123456 > /home/mysqldump/test.sql

最好在主数据库备份完毕，恢复写操作

mysql> unlock tables;

Query OK, 0 rows affected (0.28 sec)

(6)将刚才主数据备份的test.sql复制到从数据库，进行导入

(7)接着修改从数据库的my.cnf(windows为my.ini),增加server-id参数,指定复制使用的用户,主数据库服务器的ip,端口以及开始执行复制日志的文件和位置

[mysqld]

server-id=2

log\_bin = /var/log/mysql/mysql-bin.log

master-host =192.168.1.200

master-user=slave

master-pass=123456

master-port =3306

master-connect-retry=60

replicate-do-db =test

(8)在从服务器上,启动slave进程

mysql> start slave;

(9)在从服务器进行show salve status验证

mysql> SHOW SLAVE STATUS;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: localhost

Master\_User: root

Master\_Port: 3306

Connect\_Retry: 3

Master\_Log\_File: mysql-bin.003

Read\_Master\_Log\_Pos: 79

Relay\_Log\_File: gbichot-relay-bin.003

Relay\_Log\_Pos: 548

Relay\_Master\_Log\_File: mysql-bin .003

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

(10)好了,现在可以在我们的主服务器做一些更新的操作,然后在从服务器查看是否已经更新

## [MySQL主主复制配置](http://www.blogjava.net/dongbule/archive/2010/08/22/329602.html)

(1)创建并授权用户

这一步在每一台（主）服务器上创建一个用户，并为之授权，使它们可以互相访问彼此的数据库

**在Master-1上：**

创建一个允许Master-2来访问的用户master2,密码为：123456

mysql> GRANT REPLICATION SLAVE ON \*.\*

TO 'master2'@'192.168.1.201'  IDENTIFIED BY '123456';

**在Master-2上：**

创建一个允许Master-2来访问的用户master1,密码为：123456

mysql> GRANT REPLICATION SLAVE ON \*.\*

TO 'master1'@'192.168.1.200'  IDENTIFIED BY '123456';

(2)**修改MySQL主配置文件**

**Master-1上：**

[mysqld]

server-id=1

log-bin = mysql-bin

auto-increment-increment = 2

auto-increment-offset = 1

**Master-2上：**

[mysqld]

server-id=2

log-bin = mysql-bin

auto-increment-increment = 2

auto-increment-offset = 2

注：二都只有server-id不同和 auto-increment- offset不同auto-increment-offset是用来设定数据库中自动增长的起点的，回为这两能服务器都设定了一次自动增长值2，所以它们的起点必须得不同，这样才能避免两台服务器数据同步时出现主键冲突，配置好重启MySQL读取新的配置文件

另：auto-increment-increment的值应设为整个结构中服务器的总数，本案例用到两台服务器，所以值设为2

(3)复制其中一台服务器的数据库到别外一台服务器(同上主从复制，略过)

(4)互相通告二进制日志位置(先得到主服务器当前二进制日志名和偏移量，参照主从配置第(4)步)

**在Master-1上：**

mysql>

   > CHANGE MASTER TO

   > MASTER\_HOST='192.168.1.201',

   > MASTER\_USER='master1',

   > MASTER\_PASSWORD='123456',

   > MASTER\_LOG\_FILE='mysql-bin.000001',

   > MASTER\_LOG\_POS=106;

**在Master-2上：**

mysql>

   > CHANGE MASTER TO

   > MASTER\_HOST='192.168.1.200',

   > MASTER\_USER='master2',

   > MASTER\_PASSWORD='123456',

   > MASTER\_LOG\_FILE='mysql-bin.000006',

   > MASTER\_LOG\_POS=213;

(6)、启动复制（Replication）功能

在两台主机上分别执行

mysql>

   > START SLAVE;

配置到此完成！