[WindowsHYPERLINK "http://www.cnblogs.com/taven/archive/2012/07/16/2593983.html"下HYPERLINK "http://www.cnblogs.com/taven/archive/2012/07/16/2593983.html"MySQLHYPERLINK "http://www.cnblogs.com/taven/archive/2012/07/16/2593983.html"的主从热备（自动同步）配置](http://www.cnblogs.com/taven/archive/2012/07/16/2593983.html)

本配置方法适用于5.1之后的版本，个人在5.5上配置成功（5.1之前版本请参考另外的配置说明）

 环境说明：

Master：192.168.1.200

Slave：192.168.1.210

MySQL 的 Master 配置：

配置my.ini：

[mysqld]

# The TCP/IP Port the MySQL Server will listen on

port=3306

server-id=200

# replicate-do-db = NearbyPeople

log-bin=mysql-bin

relay-log=relay-bin

binlog-bin-db= NearbyPeople

relay-log-index=relay-bin-index

replicate\_wild\_do\_table= NearbyPeople.%

# replicate\_wild\_do\_table= NearbyPeople.%

# replicate\_wild\_do\_table=lmis.%

\*配置好后，重启Master 的 MySQL服务！！！

service mysqld restart

授权 ：

 flush privileges;   #刷新系统授权表

grant replication slave  on NearbyPeople.\* to ' root '@'1.13.239.14' identified by ' root' with grant option;  #授权用户xkhz只能从192.168.21.129这个IP访问主服务器192.168.21.128上面的数据库，并且只具有数据库备份的权限

//--------------------------------------------------

MySQL 的 Slave 配置：

配置my.ini：

[mysqld]

# The TCP/IP Port the MySQL Server will listen on

port=3306

server-id = 210

replicate-do-db = NearbyPeople

slave-skip-errors = 1062

#slave-skip-errors=1062,1053,1146 #跳过指定error no类型的错误  
#slave-skip-errors=all #跳过所有错误

\*配置好后，重启Slave 的 MySQL服务！！！

//--------------------------------------------------

登录Slave，然后执行下面命令：

stop slave;

change master to master\_host='101.200.233.241',master\_user='xkhz',master\_password='Xkhz%^#', master\_port=17215, master\_log\_file='mysql-bin.000001',master\_log\_pos=57361018;

start slave;

show slave status;

查看这两项是否为YES，yes为正常。

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

slave stop;  
set GLOBAL SQL\_SLAVE\_SKIP\_COUNTER=1;  
slave start;

两台MYSQL互相备份同步

MySQL支持单向、异步复制，复制过程中一个服务器充当主服务器，而一个或多个其它服务器充当从服务器。主服务器将更新写入二进制日志文件，并维护日志文件的一个索引以跟踪日志循环。

服务器在日志中读取的最后一次成功更新的位置。从服务器接收从那时起发生的任何更新，然后封锁并等待主服务器通知下一次更新。

在实际项目中，两台分布于异地的主机上安装有MySQL数据库，两台服务器互为主备，客户要求当其中一台机器出现故障时，另外一台能够接管服务器上的应用，这就需要两台数据库的数据要实时保持一致，在这里使用MySQL的同步功能实现双机的同步复制。

以下是操作实例：

1、数据库同步设置

主机操作系统：RedHat Enterprise Linux 5

数据库版本：MySQL Ver 14.12 Distrib 5.0.22

前提：MySQL数据库正常启动

假设两台主机地址分别为：

ServA：172.18.3.3

ServB：172.18.3.4

1.1 配置同步账号

在ServA上增加一个ServB可以登录的帐号：

MySQL>GRANT all privileges ON \*.\* TO tongbu@'172.18.3.4' IDENTIFIED BY '123456'; (注意：tongbu是用于同步的帐号，123456是tongbu的密码)

在ServB上增加一个ServA可以登录的帐号：

MySQL>GRANT all privileges ON \*.\* TO tongbu@'172.18.3.3' IDENTIFIED BY '123456';

1.2 配置数据库参数

1、以root用户登录ServA，修改ServA的my.cnf文件

vi /etc/my.cnf

在[MySQLd]的配置项中增加如下配置：

1 default-character-set=utf8

2

3 log-bin=MySQL-bin

4

5 relay-log=relay-bin

6

7 relay-log-index=relay-bin-index

8

9 server-id=1

10

11 master-host=172.18.3.4 #同步服务器的IP

12

13 master-user=tongbu #用于同步的帐号

14

15 master-password=123456 #用于同步帐号的密码

16

17 master-port=3306 #同步服务器的端口

18

19 master-connect-retry=30 #同步的时间间隔，单位是秒

20

21 binlog-do-db=umsdb #需要同步的库

22

23 replicate-do-db=umsdb #需要同步的库

24

25 replicate-ignore-table=umsdb.boco\_tb\_menu

26

27 replicate-ignore-table=umsdb.boco\_tb\_connect\_log

28

29 replicate-ignore-table=umsdb.boco\_tb\_data\_stat

30

31 replicate-ignore-table=umsdb.boco\_tb\_log\_record

32

33 replicate-ignore-table=umsdb.boco\_tb\_workorder\_record

2、以root用户登录ServB，修改ServB的my.cnf文件

vi /etc/my.cnf

在[MySQLd]的配置项中增加如下配置：

1 default-character-set=utf8

2

3 log-bin=MySQL-bin

4

5 relay-log=relay-bin

6

7 relay-log-index=relay-bin-index

8

9 server-id=2

master-host=218.60.65.17

master-user= xkhz

master-password= Xkhz%^#

master-port= 17215

master-connect-retry=30

change master to master\_host='218.60.65.17',master\_user='xkhz',master\_password='Xkhz%^#';

20

21 binlog-do-db=umsdb

22

23 replicate-do-db=umsdb

24

25 replicate-ignore-table=umsdb.boco\_tb\_menu

26

27 replicate-ignore-table=umsdb.boco\_tb\_connect\_log

28

29 replicate-ignore-table=umsdb.boco\_tb\_data\_stat

30

31 replicate-ignore-table=umsdb.boco\_tb\_log\_record

32

33 replicate-ignore-table=umsdb.boco\_tb\_workorder\_record

1.3 手工执行数据库同步

假设以ServA为主服务器，在ServB上重启MySQL：

service MySQLd restart

在ServB上用root用户登录MySQL，执行：

MySQL> stop slave;

MySQL> load data from master;

MySQL> start slave;

在ServA上重启MySQL：

service MySQLd restart

1.4 查看数据库同步状态

在MySQL命令提示符下执行：

MySQL> show slave status\G

将显示同步进程的状态，如下所示，两行蓝色字体为slave进程状态，如果都为yes表示正常;红色字体表示同步错误指示，如果有问题会有错误提示：

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

2

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

4

5 Master\_Host: 10.21.2.90

6

7 Master\_User: tongbu

8

9 Master\_Port: 3306

10

11 Connect\_Retry: 30

12

13 Master\_Log\_File: localhost-bin.000005

14

15 Read\_Master\_Log\_Pos: 39753882

16

17 Relay\_Log\_File: localhost-relay-bin.000062

18

19 Relay\_Log\_Pos: 9826663

20

21 Relay\_Master\_Log\_File: localhost-bin.000005

22

23 Slave\_IO\_Running: Yes

24

25 Slave\_SQL\_Running: Yes

26

27 Replicate\_Do\_DB: bak,umsdb

28

29 Replicate\_Ignore\_DB:

30

31 Replicate\_Do\_Table:

32

33 Replicate\_Ignore\_Table: umsdb.boco\_tb\_connect\_log,umsdb.boco\_tb\_menu,umsdb.boco\_tb\_workorder\_record,

umsdb.boco\_tb\_data\_stat,umsdb.boco\_tb\_log\_record

34

35 Replicate\_Wild\_Do\_Table:

36

37 Replicate\_Wild\_Ignore\_Table:

38

39 Last\_Errno: 0

40

41 Last\_Error:

42

43 Skip\_Counter: 0

44

45 Exec\_Master\_Log\_Pos: 39753882

46

47 Relay\_Log\_Space: 9826663

48

49 Until\_Condition: None

50

51 Until\_Log\_File:

52

53 Until\_Log\_Pos: 0

54

55 Master\_SSL\_Allowed: No

56

57 Master\_SSL\_CA\_File:

58

59 Master\_SSL\_CA\_Path:

60

61 Master\_SSL\_Cert:

62

63 Master\_SSL\_Cipher:

64

65 Master\_SSL\_Key:

66

67 Seconds\_Behind\_Master:

3、数据库同步测试

配置完数据库后进行测试，首先在网络正常情况下测试，在ServA上进行数据库操作，和在ServB上进行数据库操作，数据都能够同步过去。

拔掉ServB主机上的网线，然后在ServA上做一些数据库操作，之后再恢复ServB的网络环境，但是在ServB上却看不到同步的数据，通过命令show slave status\G查看发现Slave\_IO\_Running的状态是No，这种状态持续很长一段时间，数据才能同步到ServB上去。这是什么问题呢?同步延迟不会这么大吧。后来通过网上查找相关资料，找到一个同步延迟相关的参数：

--slave-net-timeout=seconds

参数含义：当slave从主数据库读取log数据失败后，等待多久重新建立连接并获取数据。

于是在配置文件中增加该参数，设置为60秒

slave-net-timeout=60

重启MySQL数据库后测试，该问题解决。

4、 数据库同步失效的解决

当数据同步进程失效后，首先手工检查slave主机当前备份的数据库日志文件在master主机上是否存在，在slave主机上运行：

MySQL> show slave status\G

一般获得如下的信息：

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

2

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

4

5 Master\_Host: 172.18.3.3

6

7 Master\_User: tongbu

8

9 Master\_Port: 3306

10

11 Connect\_Retry: 30

12

13 Master\_Log\_File: MySQL-bin.000001

14

15 Read\_Master\_Log\_Pos: 360

16

17 Relay\_Log\_File: localhost-relay-bin.000003

18

19 Relay\_Log\_Pos: 497

20

21 Relay\_Master\_Log\_File: MySQL-bin.000001

22

23 Slave\_IO\_Running: Yes

24

25 Slave\_SQL\_Running: Yes

26

27 Replicate\_Do\_DB: bak

28

29 Replicate\_Ignore\_DB:

30

31 Replicate\_Do\_Table:

32

33 Replicate\_Ignore\_Table:

34

35 Replicate\_Wild\_Do\_Table:

36

37 Replicate\_Wild\_Ignore\_Table:

38

39 Last\_Errno: 0

40

41 Last\_Error:

42

43 Skip\_Counter: 0

44

45 Exec\_Master\_Log\_Pos: 360

46

47 Relay\_Log\_Space: 497

48

49 Until\_Condition: None

50

51 Until\_Log\_File:

52

53 Until\_Log\_Pos: 0

54

55 Master\_SSL\_Allowed: No

56

57 Master\_SSL\_CA\_File:

58

59 Master\_SSL\_CA\_Path:

60

61 Master\_SSL\_Cert:

62

63 Master\_SSL\_Cipher:

64

65 Master\_SSL\_Key:

66

67 Seconds\_Behind\_Master: 0其中Master\_Log\_File描述的是master主机上的日志文件。

在master上检查当前的数据库列表：

MySQL> show master logs;

得到的日志列表如下：

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

| Log\_name | File\_size |

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

| localhost-bin.000001 | 495 |

| localhost-bin.000002 | 3394 |

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

如果slave主机上使用的的Master\_Log\_File对应的文件在master的日志列表中存在，在slave主机上开启从属服务器线程后可以自动同步：

MySQL> start slave;

如果master主机上的日志文件已经不存在，则需要首先从master主机上恢复全部数据，再开启同步机制。

在slave主机上运行：

MySQL> stop slave;

在master主机上运行：

MySQL> stop slave;

在slave主机上运行：

MySQL> load data from master;

MySQL> reset master;

MySQL> start slave;

在master主机上运行：

MySQL> reset slave;

MySQL>start slave;

注意：LOAD DATA FROM MASTER目前只在所有表使用MyISAM存储引擎的数据库上有效。

**故障现象：**

进入slave服务器，运行：

mysql> show slave status\G

         .......  
             Relay\_Log\_File: localhost-relay-bin.000535  
              Relay\_Log\_Pos: 21795072  
      Relay\_Master\_Log\_File: localhost-bin.000094  
          Slave\_IO\_Running: Yes  
          Slave\_SQL\_Running: No  
            Replicate\_Do\_DB:   
        Replicate\_Ignore\_DB:   
       ......

**解决办法一、**

Slave\_SQL\_Running: No  
1.程序可能在slave上进行了写操作

2.也可能是slave机器重起后，事务回滚造成的.  
  
一般是事务回滚造成的：  
解决办法：  
mysql> slave stop;  
mysql> set GLOBAL SQL\_SLAVE\_SKIP\_COUNTER=1;  
mysql> slave start;

**解决办法二、**

首先停掉Slave服务：slave stop  
到主服务器上查看主机状态：  
记录File和Position对应的值

进入master

mysql> show master status;  
+----------------------+----------+--------------+------------------+  
| File                 | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |  
+----------------------+----------+--------------+------------------+  
| localhost-bin.000094 | 33622483 |              |                  |   
+----------------------+----------+--------------+------------------+  
1 row in set (0.00 sec)

然后到slave服务器上执行手动同步：  
  
mysql> change master to   
> master\_host='master\_ip',  
> master\_user='user',   
> master\_password='pwd',   
> master\_port=3306,   
> master\_log\_file=localhost-bin.000094',   
> master\_log\_pos=33622483 ;  
1 row in set (0.00 sec)  
mysql> slave start;  
1 row in set (0.00 sec)

mysql> show slave status\G  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
........  
            Master\_Log\_File: localhost-bin.000094  
        Read\_Master\_Log\_Pos: 33768775  
             Relay\_Log\_File: localhost-relay-bin.000537  
              Relay\_Log\_Pos: 1094034  
      Relay\_Master\_Log\_File: localhost-bin.000094  
           Slave\_IO\_Running: Yes  
          Slave\_SQL\_Running: Yes  
            Replicate\_Do\_DB:

手动同步需要停止master的写操作！

**"Slave\_IO\_Running: No"MYSQL不同步**

|  |  |
| --- | --- |
|  |  |

"Slave\_IO\_Running: No"MYSQL不同步  
  
今天发现Mysql不同步，show slave status查看同步状态如下：  
Slave\_IO\_Running: No  
Slave\_SQL\_Running: Yes   
使用start slave也不能同步。后来，先stop slave 再reset slave再start slave就正常了。