```
################################################
# File name   : Heartbeat+Mon Mysql Master-Slave 成功配置.txt
# Description :
# Requirement :
#
# Copyright(C), fkoo, 2008, All Rights Reserved.
#
# Author: Far Young Chen / fkoo (fkoo.com@gmail.com)
# URL: http://www.fkoo.com
#
#################################################

# 安装 mysql5.1.30 二进制包
nohup /mnt/hgfs/share/LAMP--mysql5.1.30-only.sh &

# 给 roo t用户添加 mysql 命令环境变量 /usr/local/mysql/bin
# 使环境变量生效
[root@S31 ~]# vi /root/.bash_profile
PATH=$PATH:$HOME/bin:/usr/local/mysql/bin
[root@S31 ~]# su -

-------------------------------------------------------------------------------------------------------------------------
--------------------
# 建立测试数据库和表单
# 说明: 表单必须建立,否则后面配置的 mon 监测不到数据库表单而触发误动作
[root@S31 ~]# mysql -p
show databases;
create database fkoodb;
use fkoodb
CREATE TABLE mytable (name VARCHAR(20), sex CHAR(1), \
birth DATE, birthaddr VARCHAR(20));
show tables;
+------------------+
| Tables_in_fkoodb |
+------------------+
|  mytable          |
+------------------+

DESCRIBE mytable;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| name       | varchar(20) | YES  |     | NULL    |       |
| sex        | char(1)     | YES  |     | NULL    |       |
| birth      | date        | YES  |     | NULL    |       |
| birthaddr  | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+

insert into mytable values ('fkoo','m','2008-11-26','jinan1');
insert into mytable values ('fkoo','m','2008-11-26','jinan2');
insert into mytable values ('fkoo','m','2008-11-26','jinan3');
insert into mytable values ('fkoo','m','2008-11-26','jinan4');
insert into mytable values ('fkoo','m','2008-11-26','jinan5');
insert into mytable values ('fkoo','m','2008-11-26','jinan6');

delete from mytable  where birthaddr = 'jinan1';
delete from mytable  where birthaddr = 'jinan2';
delete from mytable  where birthaddr = 'jinan3';
delete from mytable  where birthaddr = 'jinan4';
delete from mytable  where birthaddr = 'jinan5';
delete from mytable  where birthaddr = 'jinan6';

use fkoodb
select * from mytable;
+----------+------+------------+-----------+
| name     | sex  | birth      | birthaddr |
+----------+------+------------+-----------+
| fkooname | m    | 2008-11-26 | jinan     |
+----------+------+------------+-----------+

# drop table mytable;

# 修改 mysql 服务的 root 口令
# 说明: 因安装脚本已经将 root 口令设置为 rvdg9lip , 这略过此步骤
use mysql
update user set password=password('rvdg9lip') where user='root';

# 给
use mysql
grant replication slave on *.* to 'fkoocopy'@'172.%' identified by 'fkoopasswd';
grant select on *.* to 'fkoo_monitor'@'172.%' identified by 'FkooMonitor';

delete from user where user='';
delete from user where user='root' and host='%';
delete from user where user='root' and host='127.0.0.1';
delete from user where user='root’ and host='T254’;


# delete from mysql.user where user='fkoo_monitor’;

use mysql
flush privileges;
select * from user;
quit;

#grant all privileges on *.* to 'root’@’%’ identified by 'rvdgi,jl’;
#grant all privileges on *.* to 'fkoocopy’@’10.0.%’ identified by 'fkoopasswd’;
#delete from mysql.user where user='fkoocopy’ and host='10.0.%’;

#grant all privileges on *.* to 'fkoocopy’@’10.0.1.%’ identified by 'fkoopasswd’;
#delete from mysql.user where user='fkoocopy’ and host='10.0.1.%’;

#grant replication slave on *.* to 'fkoocopy’@’%’ identified by 'fkoopasswd’;
# delete from mysql.user where user='fkoocopy’ and host='%’;

#grant all privileges on *.* to 'fkoo_monitor’@’S221’ identified by 'FkooMonitor’;
# delete from mysql.user where user='fkoo_monitor’ and host='S221’;

#grant all privileges on *.* to 'fkoo_monitor’@’localhost’ identified by 'FkooMonitor’;
# delete from mysql.user where user='fkoo_monitor’ and host='localhost’;

#grant all privileges on *.* to 'fkoo_monitor’@’%’ identified by 'FkooMonitor’;
# delete from mysql.user where user='fkoo_monitor’ and host='%’;

[root@S221 ~]#
# cp /usr/local/mysql/support-files/my-huge.cnf /etc/my.cnf
cp /etc/my.cnf /etc/my.cnf.bak
[root@S221 ~]# vi /etc/my.cnf
# [mysqld_safe]
# log-error=/var/log/mysqld.log
# pid-file=/var/run/mysqld/mysqld.pid

log-bin=mysql-bin
server-id       = 1
#binlog-do-db=fkoodb
binlog-ignore-db = mysql,information_schema,test
auto_increment_increment = 10
auto_increment_offset = 1
```

```
master-host=172.31.0.32
master-user=fkoocopy
master-password=fkoopasswd
master-port=3306
master-connect-retry=10
report-host=S31
#replicate-do-db=fkoodb
log-slave-updates
log-warnings

[root@S221 ~]# service mysql restart
[root@S221 ~]# mysql -p
show master status;
+------------------+----------+--------------+--------------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB         |
+------------------+----------+--------------+--------------------------+
| mysql-bin.000001 |       98 | fkoodb       | mysql,information_schema |
+------------------+----------+--------------+--------------------------+

show processlist\G
*************************** 1. row ***************************
     Id: 407
   User: fkoocopy
   Host: 10.0.2.223:4228
     db: NULL
Command: Binlog Dump
   Time: 225
  State: Has sent all binlog to slave; waiting for binlog to be updated
   Info: NULL
*************************** 2. row ***************************
     Id: 437
   User: root
   Host: localhost
     db: NULL
Command: Query
   Time: 0
  State: NULL
   Info: show processlist
2 rows in set (0.00 sec)

[root@S221 ~]#
mysqldump -uroot -p fkoodb > fkoodb.sql
scp fkoodb.sql root@S222:/tmp
scp fkoodb.sql root@S223:/tmp

mysqldump -uroot -p fkoodb > /mnt/hgfs/share/fkoodb.sql


[root@S221 ~]# 安装Mon
rpm -ivh /mnt/hgfs/share/perl-Time-Period-1.20-2.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Net-SNPP-1.17-1.2.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Math-TrulyRandom-1.0-1.2.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-Convert-BER-1.3101-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Mon-0.11-2.2.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-AOL-TOC-0.340-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Authen-PAM-0.16-1.2.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-UNIVERSAL-can-1.12-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-UNIVERSAL-isa-0.06-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Test-MockObject-1.08-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Test-Mock-LWP-0.05-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-HTML-Tagset-3.20-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-HTML-Parser-3.56-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/libghttp-1.0.9-10.99_2.0.el5.i386.rpm
rpm -ivh /mnt/hgfs/share/libghttp-devel-1.0.9-10.99_2.0.el5.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-HTTP-GHTTP-1.07-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-libwww-perl-5.803-2_6.0.el5.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Net-Daemon-0.43-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-PlRPC-0.2020-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-DBI-1.602-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/mysqlclient15-5.0.45-1.el5.remi.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-DBD-mysql-4.006-1.el5.rf.i386.rpm

rpm -i /mnt/hgfs/share/perl-Time-HiRes-1.9712-1.rf.src.rpm
cd /usr/src/redhat/SPECS
rpmbuild -bp perl-Time-HiRes.spec
cd /usr/src/redhat/BUILD/Time-HiRes-1.9712/
perl Makefile.PL
make
make install
cd ../..
rm -rf BUILD/Time-HiRes-1.9712*
rm -rf SOURCES/Time-HiRes-1.9712.tar.gz
rm -rf SPECS/perl-Time-HiRes.spec

rpm -ivh /mnt/hgfs/share/mon-1.2.0-1.el5.rf.i386.rpm
cp /etc/mon/mon.cf /etc/mon/mon.cf.bak
```

```
# hostgroup 与 watch 之间必须空一行
[root@S221 ~]# vi /etc/mon/mon.cf
### group definitions (hostnames or IP addresses)
hostgroup MasterDB 172.16.0.32

watch MasterDB
    service mysql
        interval 5s
        monitor msql-mysql.monitor --mode mysql --username=fkoo_monitor \
        --password=FkooMonitor --database=fkoodb
        period wd {Mon-Sun}
        alert test.alert
                #alert mail.alert fkoo.com@gmail.com
                #upalert mail.alert fkoo.com@gmail.com
                alertevery 600s
                alertafter 3

----------------------------------------------------
hostgroup MasterDB 172.16.0.31


[root@S221 ~]#
chmod 755 /usr/lib/mon/alert.d/test.alert
# echo "service heartbeat stop" >> /usr/lib/mon/alert.d/test.alert
# echo "/usr/lib/heartbeat/hb_standby" >> /usr/lib/mon/alert.d/test.alert
echo "service mysql restart" >> /usr/lib/mon/alert.d/test.alert
echo "/usr/lib/heartbeat/hb_takeover" >> /usr/lib/mon/alert.d/test.alert

tail  /usr/lib/mon/alert.d/test.alert

cp /mnt/hgfs/share/msql-mysql.monitor /usr/lib/mon/mon.d/
chmod 755 /usr/lib/mon/mon.d/msql-mysql.monitor

service mon restart
[root@S221 ~]# monshow --full
  GROUP           SERVICE      STATUS      LAST       NEXT       ALERTS SUMMARY
R MasterDB        mysql        -           1s         3s         none

  GROUP           SERVICE      STATUS      LAST       NEXT       ALERTS SUMMARY
R MasterDB        mysql        FAIL        0s         0s         1      172.16.0.31

chkconfig mon on
chkconfig --list |grep mon
# service mon stop

chkconfig mysql on
service mysql start

-------------------------------------------------------------------------------------------------------------------------
--------------------

[root@S221&S222 ~]#
groupadd haclient
useradd -g haclient hacluster

[root@S221&S222 heartbeat]#

rpm -ivh /mnt/hgfs/share/perl-TimeDate-1.16-5.el5.noarch.rpm
rpm -ivh /mnt/hgfs/share/heartbeat-pils-2.1.4-2.1.i386.rpm
rpm -ivh /mnt/hgfs/share/heartbeat-stonith-2.1.4-2.1.i386.rpm
rpm -ivh /mnt/hgfs/share/heartbeat-2.1.4-2.1.i386.rpm
 rpm -ivh /mnt/hgfs/share/libnet-1.1.2.1-2.1.i386.rpm

cp /usr/share/doc/packages/heartbeat/ha.cf  /etc/ha.d/
cp /usr/share/doc/packages/heartbeat/authkeys /etc/ha.d/
cp /usr/share/doc/packages/heartbeat/haresources /etc/ha.d/

chkconfig --add heartbeat
chkconfig  heartbeat on
chkconfig --list |grep heartbeat

[root@S221&S222 heartbeat]#
echo "auth 1" >> /etc/ha.d/authkeys
echo "1 crc" >> /etc/ha.d/authkeys
tail /etc/ha.d/authkeys
chmod 600 /etc/ha.d/authkeys

-------------------------------------------------------------------------------------------------------------------------
--------------------
#             DB Master: S221 做以下相同的配置                #
-------------------------------------------------------------------------------------------------------------------------
--------------------
[root@S221 heartbeat]# vi /etc/ha.d/ha.cf
debugfile /var/log/ha-debug
logfile        /var/log/ha-log
```

```
keepalive 2
deadtime 30
warntime 10
initdead 120
udpport        694
bcast   eth1
ucast eth0 172.16.0.32
auto_failback off
node    S31
node    S32

[root@S221 heartbeat]#
# echo "S221 10.0.1.220 mysql mon" >> /etc/ha.d/haresources
echo "S31 172.16.0.30 " >> /etc/ha.d/haresources
tail /etc/ha.d/haresources

service heartbeat start

------------------------------------------------------------------------------------------------------------------------
--------------------
#             DB Slaver-ha: S222 / DB Slaver: S223 都做以下相同的配置                #
------------------------------------------------------------------------------------------------------------------------
--------------------
[root@S222&S223 ~]# mysql -p
create database fkoodb;
use fkoodb
show tables;
quit;
[root@S222&S223 ~]#
mysql -u root -prvdgi,jl fkoodb < /mnt/hgfs/share/fkoodb.sql

drop database fkoodb;

------------------------------------------------------------------------------------------------------------------------
--------------------
#             DB Slaver-ha: S222 做以下相同的配置                #
------------------------------------------------------------------------------------------------------------------------
--------------------
[root@S222 ~]# mysql -p

use mysql
grant replication slave on *.* to 'fkoocopy'@'%' identified by 'fkoopasswd';
flush privileges;
```

```
select * from user;
quit;

# use mysql
# update user set password=password(’rvdg9lip’) where user=’root’;

[root@S222 ~]#
cp /usr/local/mysql/support-files/my-huge.cnf /etc/my.cnf
cp /etc/my.cnf /etc/my.cnf.bak
[root@S222 ~]# vi /etc/my.cnf
log-bin=mysql-bin
# server-id       = 1
server-id       = 2
#binlog-do-db=fkoodb
binlog-ignore-db = mysql,information_schema
auto_increment_increment = 10
auto_increment_offset = 2
master-host=172.31.0.31
master-user=fkoocopy
master-password=fkoopasswd
master-port=3306
master-connect-retry=10
report-host=S32
#replicate-do-db=fkoodb
log-slave-updates


[root@S222 ~]# service mysql restart

 cat /dev/null > /usr/local/mysql/data/master.info

 stop slave;
 reset slave;

 start slave;

show master status;
show slave status\G;

service mysql stop
rm -rf /usr/local/mysql/data/fkoodb/
rm -rf /usr/local/mysql/data/mysql-bin.*
rm -rf /usr/local/mysql/data/S32
rm -rf /usr/local/mysql/data/S32-relay-bin.*
rm -rf /usr/local/mysql/data/relay-log.info
rm -rf /usr/local/mysql/data/master.info
service mysql start

service mysql stop
rm -rf /usr/local/mysql/data/fkoodb/
rm -rf /usr/local/mysql/data/mysql-bin.*
rm -rf /usr/local/mysql/data/S31
rm -rf /usr/local/mysql/data/S31-relay-bin.*
rm -rf /usr/local/mysql/data/
rm -rf /usr/local/mysql/data/relay-log.info
rm -rf /usr/local/mysql/data/master.info
service mysql start


[root@S32 ~]# rm -rf /usr/local/mysql/data/
ibdata1      ib_logfile0  ib_logfile1  mysql/       S31.err      S32.err      T254.err     test/
[root@S32 ~]# rm -rf /usr/local/mysql/data/

show slave status\G
*************************** 1. row ***************************
             Slave_IO_State: Waiting for master to send event
                Master_Host: 10.0.2.220
                Master_User: fkoocopy
                Master_Port: 3306
              Connect_Retry: 10
            Master_Log_File: mysql-bin.000003
        Read_Master_Log_Pos: 98
             Relay_Log_File: S222-relay-bin.000004
              Relay_Log_Pos: 235
      Relay_Master_Log_File: mysql-bin.000003
           Slave_IO_Running: Yes
          Slave_SQL_Running: Yes
            Replicate_Do_DB: fkoodb
        Replicate_Ignore_DB:
         Replicate_Do_Table:
     Replicate_Ignore_Table:
    Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
                 Last_Errno: 0
                 Last_Error:
               Skip_Counter: 0
        Exec_Master_Log_Pos: 98
            Relay_Log_Space: 235
            Until_Condition: None
             Until_Log_File: 
              Until_Log_Pos: 0
         Master_SSL_Allowed: No
         Master_SSL_CA_File: 
         Master_SSL_CA_Path: 
            Master_SSL_Cert: 
          Master_SSL_Cipher: 
             Master_SSL_Key: 
      Seconds_Behind_Master: 0
```

```
slave stop;
CHANGE MASTER TO MASTER_HOST='10.0.1.221';
slave stop;
reset slave;
slave start;
show slave status\G
```

```
vi /usr/local/mysql/data/master.info
```

```
-------------------------------------------------------------------------------------------------------------------------
--------------------
#             DB Slaver: S223 做以下相同的配置                #
-------------------------------------------------------------------------------------------------------------------------
--------------------
[root@S223 ~]# vi /etc/sysconfig/network
HOSTNAME=S223
[root@S223 ~]# hostname S223

[root@S223 ~]#  vi /etc/my.cnf
#log-bin=mysql-bin
server-id       = 3
master-host=10.0.1.220
#auto_increment_increment = 10
#auto_increment_offset = 3
master-user=fkoocopy
master-password=fkoopasswd
master-port=3306
master-connect-retry=10
report-host=S223
replicate-do-db=fkoodb
```

```
read-only
#log-slave-updates
log-warnings

[root@S223 ~]# service mysql restart
mysql> show slave status\G
*************************** 1. row ***************************
             Slave_IO_State: Waiting for master to send event
                Master_Host: 10.0.2.220
                Master_User: fkoocopy
                Master_Port: 3306
              Connect_Retry: 60
            Master_Log_File: mysql-bin.000004
        Read_Master_Log_Pos: 98
             Relay_Log_File: S223-relay-bin.000007
              Relay_Log_Pos: 235
      Relay_Master_Log_File: mysql-bin.000004
           Slave_IO_Running: Yes
          Slave_SQL_Running: Yes
            Replicate_Do_DB: fkoodb
        Replicate_Ignore_DB:
         Replicate_Do_Table:
     Replicate_Ignore_Table:
    Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
                 Last_Errno: 0
                 Last_Error:
               Skip_Counter: 0
        Exec_Master_Log_Pos: 98
            Relay_Log_Space: 235
            Until_Condition: None
             Until_Log_File:
              Until_Log_Pos: 0
         Master_SSL_Allowed: No
         Master_SSL_CA_File:
         Master_SSL_CA_Path:
            Master_SSL_Cert:
          Master_SSL_Cipher:
             Master_SSL_Key:
      Seconds_Behind_Master: 0
```

```
#             DB Master: S221 / DB Slaver-ha: S222 都做以下相同的配置                #
------------------------------------------------------------------------------------------------------------------------
--------------------
[root@S221&S222 ~]#
groupadd haclient
useradd -g haclient hacluster

[root@S221&S222 heartbeat]#

rpm -ivh /mnt/hgfs/share/perl-TimeDate-1.16-5.el5.noarch.rpm
rpm -ivh /mnt/hgfs/share/heartbeat-pils-2.1.4-2.1.i386.rpm
rpm -ivh /mnt/hgfs/share/heartbeat-stonith-2.1.4-2.1.i386.rpm
rpm -ivh /mnt/hgfs/share/heartbeat-2.1.4-2.1.i386.rpm
 rpm -ivh /mnt/hgfs/share/libnet-1.1.2.1-2.1.i386.rpm

cp /usr/share/doc/packages/heartbeat/ha.cf  /etc/ha.d/
cp /usr/share/doc/packages/heartbeat/authkeys /etc/ha.d/
cp /usr/share/doc/packages/heartbeat/haresources /etc/ha.d/

chkconfig --add heartbeat
chkconfig  heartbeat on
chkconfig --list |grep heartbeat

[root@S221&S222 heartbeat]#
echo "auth 1" >> /etc/ha.d/authkeys
echo "1 crc" >> /etc/ha.d/authkeys
tail /etc/ha.d/authkeys
chmod 600 /etc/ha.d/authkeys

------------------------------------------------------------------------------------------------------------------------
--------------------
#             DB Master: S221 做以下相同的配置                #
------------------------------------------------------------------------------------------------------------------------
--------------------
[root@S221 heartbeat]# vi /etc/ha.d/ha.cf
#debugfile /var/log/ha-debug
debugfile /var/log/ha-debug
#logfile        /var/log/ha-log
logfile        /var/log/ha-log
#keepalive 2
keepalive 2
#deadtime 30
deadtime 30
```

```
#warntime 10
warntime 10
#initdead 120
initdead 120
#udpport        694
udpport        694
bcast   eth1
ucast eth0 10.0.1.222
#auto_failback on
auto_failback off
node    S221
node    S222

[root@S221 heartbeat]#
echo "S221 10.0.1.220 " >> /etc/ha.d/haresources
# echo "S221 10.0.1.220 mysql mon" >> /etc/ha.d/haresources
tail /etc/ha.d/haresources

-------------------------------------------------------------------------------------------------------------------------
--------------------
#             DB Slaver-ha: S222 做以下相同的配置                #
-------------------------------------------------------------------------------------------------------------------------
--------------------
[root@S222 heartbeat]# vi /etc/ha.d/ha.cf
#debugfile /var/log/ha-debug
debugfile /var/log/ha-debug
#logfile        /var/log/ha-log
logfile        /var/log/ha-log
#keepalive 2
keepalive 2
#deadtime 30
deadtime 30
#warntime 10
warntime 10
#initdead 120
initdead 120
#udpport        694
udpport        694
bcast   eth1
ucast eth0 10.0.1.221
#auto_failback on
auto_failback off
node    S221
node    S222

[root@S222 heartbeat]#
echo "S221 10.0.1.220" >> /etc/ha.d/haresources
tail /etc/ha.d/haresources

------------------------------------------------------------------------------------------------------------------------
--------------------
#             DB Master: S221 / DB Slaver-ha: S222 都做以下相同的配置                #
------------------------------------------------------------------------------------------------------------------------
--------------------
service heartbeat start

------------------------------------------------------------------------------------------------------------------------
--------------------
#             调试命令                #
------------------------------------------------------------------------------------------------------------------------
--------------------
service heartbeat status
service mysql status
service mon status

ifconfig
monshow --full

service heartbeat start
service mysql start
service mon start

service heartbeat stop
service mysql stop
service mon stop

service heartbeat restart
service mysql restart
service mon restart

tail /var/log/ha-log
tail /var/log/messages
tail /tmp/test.alert.log

cat /dev/null > /tmp/test.alert.log
cat /dev/null > /var/log/ha-log
```

```
# 如果活动节点需进行维修时，可先将其转移成备援执行
/usr/lib/heartbeat/hb_standby

# 如果备援节点要接掌回来的话执行
/usr/lib/heartbeat/hb_takeover


-------------------------------------------------------------------------------------------------------------------------
--------------------
#           参考资料                #
-------------------------------------------------------------------------------------------------------------------------
--------------------


use fkoodb
select * from mytable;

slave stop;
CHANGE MASTER TO
MASTER_LOG_FILE='mysql-bin.000002',
MASTER_LOG_POS=737;
slave start;
show slave status\G
```

PS :
1，Slave机器的权限问题，不但要给slave机器File权限，还要给它REPLICATION SLAVE的权限。
2. 在修改完Slave机器/etc/my.cnf之后，slave机器的mysql服务启动之前，记得要删除掉master.info
3，在show master status 或着show slave status 不正常时，看看.err是怎样说的。
4，Slave上Mysql的Replication工作有两个线程, I/O thread和SQL thread 。I/O 的作用是从master 3306端口上把它的binlog取过来(master在被修改了任何内容之后,就会把修改了什么写到自己的binlog等待slave更新),然后写到本地的relay-log,而SQL thread则是去读本地的relay-log,再把它转换成本Mysql所能理解的语句，于是同步就这样一步一步的完成.决定I/O thread的是/var/lib/mysql/master.info,而决定SQL thread的是/var/lib/mysql/relay-log.info.

*******相关命令:

```
stop slave   #停止同步
start slave   #开始同步，从日志终止的位置开始更新。
SET SQL_LOG_BIN=0|1 #主机端运行，需要super权限，用来开停日志，随意开停，会造成主机从机数据不一致，造成错误
SET GLOBAL SQL_SLAVE_SKIP_COUNTER=n # 客户端运行，用来跳过几个事件，只有当同步进程出现错误而停止的时候才可以执行。
RESET MASTER #主机端运行,清除所有的日志，这条命令就是原来的FLUSH MASTER
RESET SLAVE   #从机运行，清除日志同步位置标志，并重新生成master.info
```

虽然重新生成了master.info,但是并不起用，最好，将从机的mysql进程重启一下，

```
LOAD TABLE tblname FROM MASTER #从机运行，从主机端重读指定的表的数据，每次只能读取一个，受timeout时间限制，需要调整
timeout时间。执行这个命令需要同步账号有reload和super权限。以及对相应的库有select权限。如果表比较大，要增加
net_read_timeout 和 net_write_timeout的值
LOAD DATA FROM MASTER #从机执行，从主机端重新读入所有的数据。执行这个命令需要同步账号有reload和super权限。以及对相应的库
有select权限。如果表比较大，要增加net_read_timeout 和 net_write_timeout的值
CHANGE MASTER TO master_def_list #在线改变一些主机设置，多个用逗号间隔,比如
CHANGE MASTER TO
MASTER_HOST='master2.mycompany.com',
MASTER_USER='replication',
MASTER_PASSWORD='bigs3cret'
FLUSH MASTER;
FLUSH SLAVE;
MASTER_POS_WAIT() #从机运行
SHOW MASTER STATUS #主机运行，看日志导出信息
SHOW SLAVE HOSTS #主机运行，看连入的从机的情况。
SHOW SLAVE STATUS (slave)
show status slave\G;
SHOW MASTER LOGS (master)
SHOW BINLOG EVENTS [ IN 'logname' ] [ FROM pos ] [ LIMIT [offset,] rows ]
PURGE [MASTER] LOGS TO 'logname' ; PURGE [MASTER] LOGS BEFORE 'date'

PURGE MASTER LOGS TO 'mysql-bin.010';
PURGE MASTER LOGS BEFORE '2003-04-02 22:46:26';
```

```
load data from master;
flush slave;
show slave status;
slave stop;
reset slave;
reset master;
slave start;
SHOW MASTER STATUS;
FLUSH MASTER;
FLUSH SLAVE;
Change master to master_host=’192.168.0.2′,master_user=’slave’,
master_password=’test’, master_log_file=’mysql-bin.000001′,master_log_pos=0;
```

一些错误信息的处理，主从服务器上的命令，及状态信息。
在从服务器上使用show slave status\G
Slave_IO_Running,为No,
则说明IO_THREAD没有启动，请执行start slave io_thread
Slave_SQL_Running为No

则复制出错,查看Last_error字段排除错误后执行start slave sql_thread
查看Slave_IO_State字段空 //复制没有启动
Connecting to master//没有连接上master
Waiting for master to send event//已经连上

主服务器上的相关命令：

```
show master status
show slave hosts
show logs
show binlog events
purge logs to ’log_name’
purge logs before ’date’
reset master(老版本flush master)
set sql_log_bin=
```

从服务器上的相关命令:

```
slave start
slave stop
SLAVE STOP IO_THREAD //此线程把master段的日志写到本地
SLAVE start IO_THREAD
SLAVE STOP SQL_THREAD //此线程把写到本地的日志应用于数据库
SLAVE start SQL_THREAD
reset slave
SET GLOBAL SQL_SLAVE_SKIP_COUNTER
load data from master
show slave status(SUPER,REPLICATION CLIENT)
CHANGE MASTER TO MASTER_HOST=, MASTER_PORT=,MASTER_USER=, MASTER_PASSWORD= //动态改变master信息
PURGE MASTER [before ’date’] 删除master端已同步过的日志
```

6.3.1 Master 同步线程状态

以下列出了master的 Binlog Dump 线程 State 字段中最常见的几种状态。如果在master上没有 Binlog Dump 线程，那么同步就没有在运行。
也就是说，没有slave连接上来。

Sending binlog event to slave
事件是由二进制日志构成，一个事件通常由更新语句加上其他信息。线程读取到一个事件并正发送到slave上。

Finished reading one binlog; switching to next binlog
读取完了一个二进制日志，正切换到下一个。

Has sent all binlog to slave; waiting for binlog to be updated
已经读取完全部未完成更新日志，并且全部都发送到slave了。它处于空闲状态，正等待在master上执行新的更新操作以在二进制日志中产生新
的事件，然后读取它们。

Waiting to finalize termination
当前线程停止了，这个时间很短。

6.3.2 Slave的I/O线程状态

以下列出了slave的I/O线程 State 字段中最常见的几种状态。从MySQL 4.1.1开始，这个状态在执行 SHOW SLAVE STATUS 语句结果的 Slave_IO_State 字段也会出现。这意味着可以只执行 SHOW SLAVE STATUS 语句就能了解到更多的信息。

Connecting to master
该线程证尝试连接到master上。

Checking master version
确定连接到master后出现的一个短暂的状态。

Registering slave on master
确定连接到master后出现的一个短暂的状态。

Requesting binlog dump
确定连接到master后出现的一个短暂的状态。该线程向master发送一个请求，告诉它要请求的二进制文件以及开始位置。

Waiting to reconnect after a failed binlog dump request
如果二进制日志转储(binary log dump)请求失败了(由于连接断开)，该线程在休眠时进入这个状态，并定期重连。重连的时间间隔由 --master-connect-retry 选项来指定。

Reconnecting after a failed binlog dump request
该线程正尝试重连到master。

Waiting for master to send event
已经连接到master，正等待它发送二进制日志。如果master闲置时，这个状态可能会持续较长时间，如果它等待超过 slave_read_timeout 秒
，就会发生超时。这时，它就会考虑断开连接，然后尝试重连。

Queueing master event to the relay log
已经读取到一个事件，正把它拷贝到中继日志中以备SQL线程处理。

Waiting to reconnect after a failed master event read
读日志时发生错误(由于连接断开)。该线程在重连之前休眠 master-connect-retry 秒。

Reconnecting after a failed master event read
正尝试重连到master。当连接确定后，状态就变成 Waiting for master to send event。

Waiting for the slave SQL thread to free enough relay log space
relay_log_space_limit 的值非零，中继日志的大小总和超过这个值了。I/O线程等待SQL线程先处理中继日志然后删除它们以释放足够的空间
。

Waiting for slave mutex on exit
当前线程停止了，这个时间很短。

6.3.3 Slave的SQL线程状态

以下列出了slave的SQL线程 State 字段中最常见的几种状态：

Reading event from the relay log
从中继日志里读到一个事件以备执行。

Has read all relay log; waiting for the slave I/O thread to update it
已经处理完中继日志中的全部事件了，正等待I/O线程写入更新的日志。

Waiting for slave mutex on exit
当前线程停止了，这个时间很短。

楼主可以尝试在主从机器都运行的情况下,在从机输入:

```
MYSQL>SLAVE START;
```

强制同步一下看看行不?如果不行,那么:
1.停止主,从机器的MYSQL服务.
2.在服务全停的情况下,将主,从端的所有 bin.00* -relay-bin.00* master.info relay-log.info都删除
3.复制主机的/var/lib/mysql(或者说是你的数据库文件)到从机同样的位置(这一步很重要,我参考过官方网站,这一步其实就是一个数据库的SNAPSHOT数据快照,它要求用MYSQLDUMP生成个快照,但复制不是更省事吗:),相当于强制同步了一下)
4.启动MASTER,SLAVE
5.在从端MYSQL>SLAVE START;
应该就可以了.

> 尝试一下这么做：
> 1. 停止slave，记住binlog_file,binlog_pos
> 2. 删除所有 relay log
> 3. 用change master 重新指定包括binlog在内的相关参数

已经试过这种方法了，但不行。

原来，要先stop slave，然后再rest slave，再change master到正确的状态，最后start slave就可以了。

```
rm -rf /usr/local/mysql/data/mysql-bin.*
rm -rf /usr/local/mysql/data/master.info
rm -rf /usr/local/mysql/data/*-relay-bin.*
ls /usr/local/mysql/data/

rm -rf /usr/local/mysql/data/fkoodb/*
ls /usr/local/mysql/data/fkoodb/

scp -pr /usr/local/mysql/data/fkoodb/ root@S222:/usr/local/mysql/data/
scp -pr /usr/local/mysql/data/fkoodb/ root@S223:/usr/local/mysql/data/


cat /dev/null > /usr/local/mysql/data/master.info
cat /usr/local/mysql/data/master.info

vi /usr/local/mysql/data/mysql-bin.000002
vi /usr/local/mysql/data/S223-relay-bin.000010
```