MySQL主从复制配置：

mysql, binary

CHANGE MASTER TO

MASTER\_HOST = 'host\_name',

MASTER\_USER = 'user\_name',

MASTER\_PASSWORD = 'password',

MASTER\_PORT = port\_num

MASTER\_LOG\_FILE = 'master\_log\_name'

MASTER\_LOG\_POS = master\_log\_pos

1、从服务器能不能执行“写”操作？

CREATE

INSERT

如何阻止写从服务器？

my.cnf

[mysqld]

read-only = 1

不能阻止 SQL Thread

如果某用户有SUPER权限，则不被阻止；

mysql> FLUSH TABLES WITH READ LOCK;

2、一个主服务器可否多从？可以

一从是否多主？不行

3、主-->从：异步

mysql 5.5 google 补丁

半同步： semisync

半同步如果无法在指定时间完成-->自动 降到异步模式；

4、如何从服务器的mysql服务在启动时候不要自动启动从服务线程？

master.info

relay-log.info

在从服务器上：

[mysqld]

skip-slave-start=1

5、数据库复制过滤

在主服务器上实现：

binlog-do-db=testdb

binlog-do-db=mydb

binlog-ignore-db=mysql

主服务器

[mysqld]

binlog-do-db=magedu

在主服务器过滤：任何不涉及到数据库相关的写操作都不会被记录到二进制日志当中；

从服务器：

replicate\_do\_db

rpplicate\_ignore\_db

replicate\_do\_table

replicate\_ignore\_table

replicate\_wild\_do\_table

replicate\_wild\_ignore\_table

在从服务器上只复制mageedu一个数据库：

[mysqld]

replicate\_do\_db=mageedu

replicate\_do\_db=mysql

设置半同步步骤：

在Master和Slave的mysql命令行运行如下代码：

# On Master

mysql> INSTALL PLUGIN rpl\_semi\_sync\_master SONAME 'semisync\_master.so';

mysql> SET GLOBAL rpl\_semi\_sync\_master\_enabled = 1;

mysql> SET GLOBAL rpl\_semi\_sync\_master\_timeout = 1000;

# On Slave

mysql> INSTALL PLUGIN rpl\_semi\_sync\_slave SONAME 'semisync\_slave.so';

mysql> SET GLOBAL rpl\_semi\_sync\_slave\_enabled = 1;

mysql> STOP SLAVE IO\_THREAD; START SLAVE IO\_THREAD;

在Master和Slave的my.cnf中编辑：

# On Master

[mysqld]

rpl\_semi\_sync\_master\_enabled=1

rpl\_semi\_sync\_master\_timeout=1000 # 1 second

# On Slave

[mysqld]

rpl\_semi\_sync\_slave\_enabled=1

# 也可通过设置全局变量的方式来设置,如下：

set global rpl\_semi\_sync\_master\_enabled=1

# 取消加载插件

mysql> UNINSTALL PLUGIN rpl\_semi\_sync\_master;

==============================================

查看从服务器上的semi\_sync是否开启:

mysql> SHOW GLOBAL STATUS LIKE 'rpl\_semi%';

查看主服务器上的semi\_sync是否开启，注意clients 变为1 ，证明主从半同步复制连接成功:

mysql> SHOW GLOBAL STATUS LIKE 'rpl\_semi%';

6、主服务器崩溃，事务已经提交-->写入二进制日志；

在主-从架构上建议使用的配置：

主服务器：

sync\_binlog=1

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

从服务器：

skip\_slave\_start=1

read\_only=1

SSL:

REQURIED SSL

auto\_increment

1,3,5

2,4,8

设置主-主复制：

1、在两台服务器上各自建立一个具有复制权限的用户；

2、修改配置文件：

# 主服务器上

[mysqld]

server-id = 10

log-bin = mysql-bin

auto-increment-increment = 2

auto-increment-offset = 1

# 从服务器上

[mysqld]

server-id = 20

log-bin = mysql-bin

auto-increment-increment = 2

auto-increment-offset = 2

3、如果此时两台服务器均为新建立，且无其它写入操作，各服务器只需记录当前自己二进制日志文件及事件位置，以之作为另外的服务器复制起始位置即可

server1|mysql> SHOW MASTER STATUS\G

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

File: mysql-bin.000001

Position: 710

Binlog\_Do\_DB:

Binlog\_Ignore\_DB:

1 row in set (0.00 sec)

server2|mysql> SHOW MASTER STATUS\G

mysql> SHOW MASTER STATUS\G

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

File: mysql-bin.000003

Position: 811

Binlog\_Do\_DB:

Binlog\_Ignore\_DB:

1 row in set (0.00 sec)

4、各服务器接下来指定对另一台服务器为自己的主服务器即可：

server1|mysql> CHANGE MASTER TO ...,MASTER\_LOG\_FILE='mysql-bin.000003', MASTER\_LOG\_POS=811

server2|mysql> CHANGE MASTER TO ...,MASTER\_LOG\_FILE='mysql-bin.000001', MASTER\_LOG\_POS=710

Multi Master Mysql

MySQL

90.0%

数据分布

备份

MySQL:

主从，主服务器可以低

HA： NAS，corosync, mysql

drbd: