



链滴

Mysql 数据库之主从分离

作者: [shangjing105](#)

原文链接: <https://ld246.com/article/1490580473182>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

介绍

MySQL数据库设置读写分离，可以使对数据库的写操作和读操作在不同服务器上执行，提高并发量相应速度。

现在的网站一般大点的，都采用有数据库主从分离、读写分离，即起到备份作用也可以减轻数据库写的压力，一直听说过这些，但是自己从没有自己动手亲手实践过，今天有时间实践一下，记录下过。

实验环境

我准备了两台服务器，一个是本机电脑，一个是远程vps，分别在两台机子上装的有数据库。

MySQL安装我就不介绍了，这里需要注意的是：MySQL安装的版本最好一致，如果不一致，低版本高版本读的时候可能有问题，最好保持一致。

主库master

45.78.57.4 centos 7 Linux系统, mysql版本 5.1.73

从库slave

本机127.0.0.1 macOS系统, mysql版本 5.1.73

配置

创建用户

在主库创建一个用户，用于从库读取主库的执行日志。

需要在mysql命令行里执行，需要先登录命令行

```
GRANT REPLICATION SLAVE ON *.* TO 'test'@'45.78.57.4' IDENTIFIED BY 'test';
```

修改my.cnf

linux系统在 /etc/my.cnf ,mac系统在安装的MySQL的目录，windows也一样。

在my.cnf文件里增加一下代码

```
server-id = 1 //数据库ID号  
log-bin=master-bin //启用二进制日志  
log-bin-index=master-bin.index //二进制日志名称
```

这里注意不要放在文件的末尾，要放在前面，即[mysqld]后，这里放上我的my.cnf内容

```
[mysqld]  
server-id=1  
log-bin=master-bin  
log-bin-index=master-bin.index  
  
datadir=/var/lib/mysql  
socket=/var/lib/mysql/mysql.sock  
user=mysql
```

```
symbolic-links=0
max_allowed_packet=100M
```

```
[mysqld_safe]
log-error=/var/log/mysql/mysql.log
pid-file=/var/run/mysql/mysql.pid
```

查看状态

登陆mysql命令行后，输入show master status，如果出现下面信息代表主库配置完成

```
mysql> show master status;
+-----+-----+-----+-----+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| master-bin.000001 | 672675 |          |          |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

记录下File和Position两个内容，从库配置的时候会用到这个。

从库配置

在本机电脑（从库）上找到my.cnf文件，然后添加以下内容，这个配置和主库的配置意思是一样的

```
server-id=2
relay-log=slave-relay-bin
relay-log-index=slave-relay-bin.index
```

注意确定和主库的位置一样，我就因为位置放置在末尾导致一直关联不上。

关联主从库

最后一步很重要，登录从库的MySQL命令行，执行以下代码，主要是关联主库的一些信息。

```
change master to master_host='45.78.57.4', #Master 服务器Ip
master_port=3306,
master_user='test',
master_password='test',
master_log_file='master-bin.000001', #Master日志文件名
master_log_pos=672675; #Master日志同步开始位置
```

注意是否执行成功，如果执行失败就好好检查下代码，看看哪里写错了。

如果执行正常，就启动从库slave，并查看下连接状态。

```
//需要再mysql命令行执行
start slave;
show slave status\G; //查看slave连接状态
```

状态信息

```
Slave_IO_State: Waiting for master to send event
```

```
Master_Host: 45.78.57.4
Master_User: test
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: master-bin.000001
Read_Master_Log_Pos: 672913
Relay_Log_File: slave-relay-bin.000044
Relay_Log_Pos: 504
Relay_Master_Log_File: master-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

注意!

这两个状态必须为Yes才算成功，如果不是，则检查上面步骤那一步配置错误。

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

测试

现在你在主库上添加一条数据，看看从库上是否有一个相同的数据，如果有则配置正常，功能使用正。

主从分离的原理主要是：开启主库的执行日志功能，然后从库读取主库的日志信息，然后将主库执行的SQL语句在从库上面执行一遍就做到主从分离，主从数据保持一直，备份数据的功能。