

后盾网 人人做后盾

www.houdunwang.com

Mysql主从复制

后盾网 2011-2017

在实际企业应用环境当中，单台mysql数据库是不足以满足日后业务需求的。譬如服务器发生故障，没有备份服务器来提供服务的话，业务就得停止。鉴于这种情况，我们来学习一下mysql主从复制。

使用mysql主从复制的好处有：

- 采用主从服务器这种架构，稳定性得以提升。如果主服务器发生故障，我们可以使用从服务器来提供服务。
- 在主从服务器上分开处理用户的请求，可以提升数据处理效率。
- 将主服务器上的数据复制到从服务器上，保护数据免受意外的损失。

主从复制的特点

1. 主服务器验证连接。
2. 主服务器为从服务器开启一个线程。
3. 从服务器将主服务器日志的偏移位告诉主服务器。
4. 主服务器检查该值是否小于当前二进制日志偏移位。
5. 如果小于，则通知从服务器来取数据。
6. 从服务器持续从主服务器取数据，直至取完，这时，从服务器线程进入睡眠，主服务器线程同时进入睡眠。
7. 当主服务器有更新时，主服务器线程被激活，并将二进制日志推送给从服务器，并通知从服务器线程进入工作状态。
8. 从服务器SQL线程执行二进制日志，随后进入睡眠状态。

主从同步原理

下载地址:

<http://mirrors.sohu.com/mysql/MySQL-5.5/MySQL-server-5.5.55-1.rhel5.i386.rpm>

<http://mirrors.sohu.com/mysql/MySQL-5.5/MySQL-client-5.5.55-1.rhel5.i386.rpm>

安装:

rpm -ivh 服务端安装包名称

rpm -ivh 客户端安装名称

错误处理:

缺少perl包:

- error: Failed dependencies:/usr/bin/perl is needed by MySQL-server-5.5.36-1.rhel5.i386
- 执行: **yum install -y perl**

与原先版本冲突:

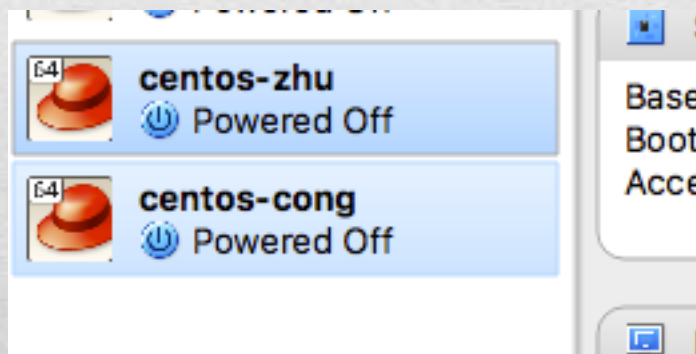
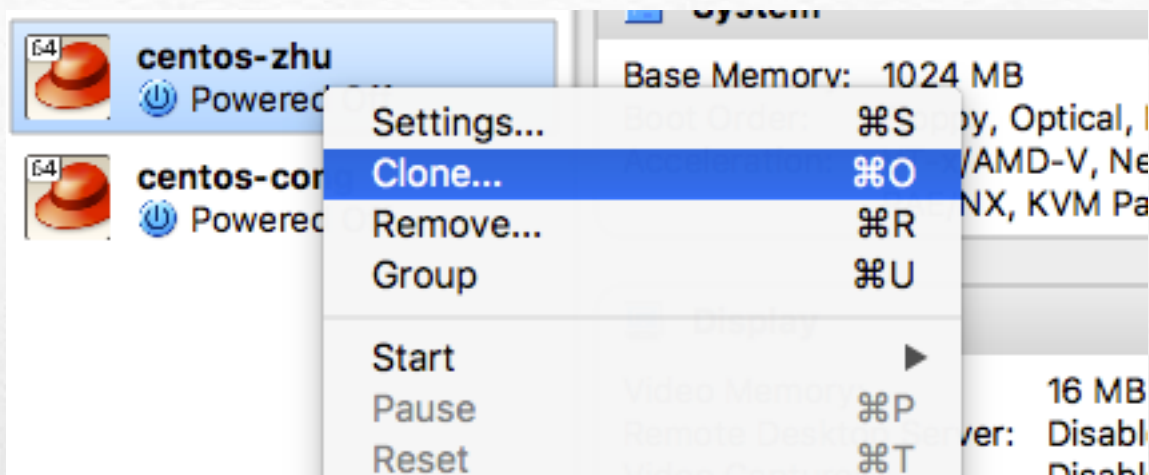
- 强制删除原版本mysql软件: **yum erase mysql-libs-5.1.73-7.el6.x86_64**

复制配置文件: **cp /usr/share/mysql/my-large.cnf /etc/my.cnf**

启动: **service mysql start**

安装实验环境(主从相同安装方式)

复制一台centos，必须有一台主服务器，一台从服务器。



克隆虚拟机

修改/etc/my.cnf配置文件如下：

1. log-bin=mysql-bin 启动二进制文件（已有，什么也不需要做）
2. server-id=1 服务器ID（已有,什么也不需要做）

binlog-do-db（随便找一个地方，新增加）

- 设置binlog记录的数据库，多个写多条
- binlog-do-db = houdunwang#第一个

binlog-ignore-db（不用设置，看看就行了）

- 设置binlog不记录的数据库，多个写多条
- binlog-ignore-db = mysql#第一个

设置binlog保存的天数（不用设置，看看就行了）

- expire-logs-days=10 #binlog 保存10天(默认为30天)

设置主服务器

先进入到Mysql命令行

GRANT创建用户并授权

1. `grant replication slave on *.* to 'slave'@'192.168.2.201' identified by 'admin888';`
2. `FLUSH PRIVILEGES;` (刷新授权表)

详细说明

`replication slave` 授权为从服务器复制操作

`On *.*` 所有库中的所有表

`192.168.2.201`是从服务器的IP

`'slave'@' 192.168.2.201'`通过192.168.2.201主机登录的slave帐号

`IDENTIFIED BY` 设置密码

查看用户

```
select user,host from mysql.user;
```

添加登录主服务器的帐号

```
set names utf8;  
  
create database houdunwang charset utf8;  
  
use houdunwang;  
  
create table user(  
  
uid int primary key auto_increment,  
  
uname char(20)  
  
);  
  
insert into user set uname='hdw';
```

在主服务器创建数据库

锁定主服务器所有表

- mysql命令: `FLUSH TABLES WITH READ LOCK;`

备份所有数据库数据

- 非mysql命令: `mysqldump -uroot -p houdunwang>houdunwang.sql`

启动slave服务器, 并且设置好slave服务器的IP

复制sql数据到slave

- 非mysql命令: `scp houdunwang.sql root@192.168.2.201:/tmp`

Master设置(主服务器)

修改从服务器my.cnf配置文件

- log-bin=mysql-bin 启动二进制文件（已有）
- server_id = 2(不能与主服务器相同,改成2)

重起Mysql服务器

- service mysql restart

创建houdunwang数据库

- mysql命令:
- set names utf8;
- create database houdunwang charset utf8;

导入从主服务器复制的sql文件

- 非mysql命令: mysql -uroot -p houdunwang</tmp/houdunwang.sql

注:

- 如果出现错误: ERROR! MySQL server PID file could not be found!
- 执行 rm -rf /var/lib/mysql/mysql-bin.index

设置从服务器

主服务器状态

```
hdxj — root@houdunwang:~ — ssh — 69x7
root@houdunwang:~ root@houdunwang:/usr/share/mysql
mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000003 |      729 | houdunwang   |                   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

从服务器列表

```
hdxj — root@houdunwang:~ — ssh — 69x7
root@houdunwang:~ root@houdunwang:/usr/share/mysql
mysql> SHOW SLAVE HOSTS;
+-----+-----+-----+-----+
| Server_id | Host | Port | Master_id |
+-----+-----+-----+-----+
|          2 |     | 3306 |          1 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

查看状态--主服务器执行

同步binlog日志

- `mysql> change master to
master_host='192.168.31.212',master_user='slave',master_passwo
rd='admin888',master_log_file='mysql-bin.
000004',master_log_pos=120;`

开启从服务器(slave)同步

- `mysql> start slave;`

设置从服务器

解除主服务器 (Master)表锁定

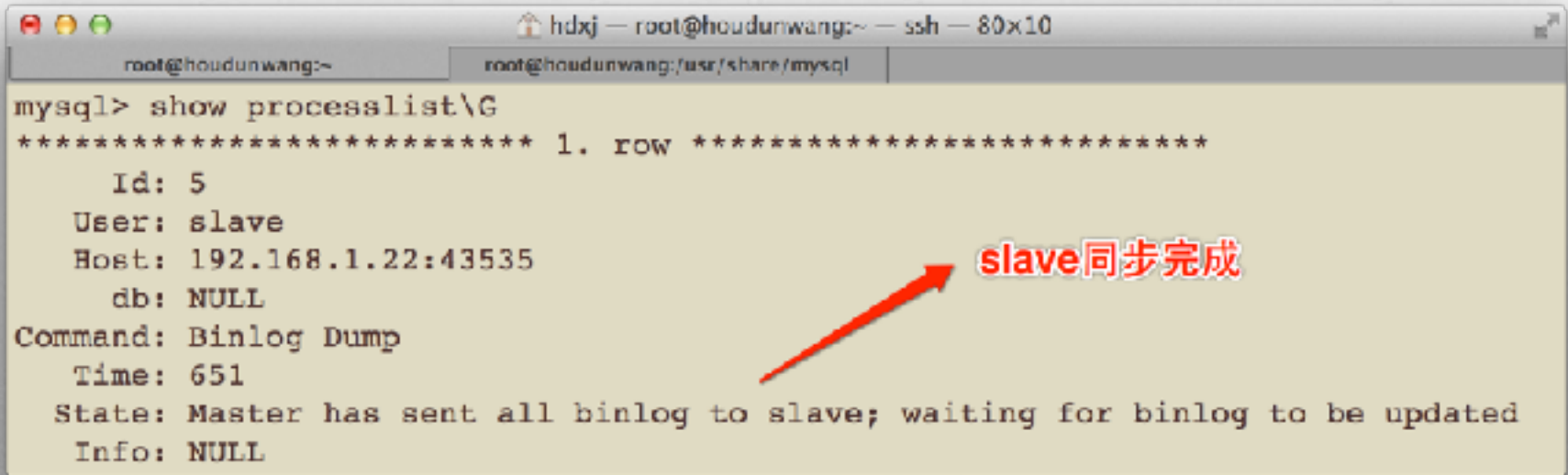
- `mysql> unlock tables;`

同步测试

- 在主服务器上插入数据，观察从服务器是否已经同步

设置主服务器

show processlist

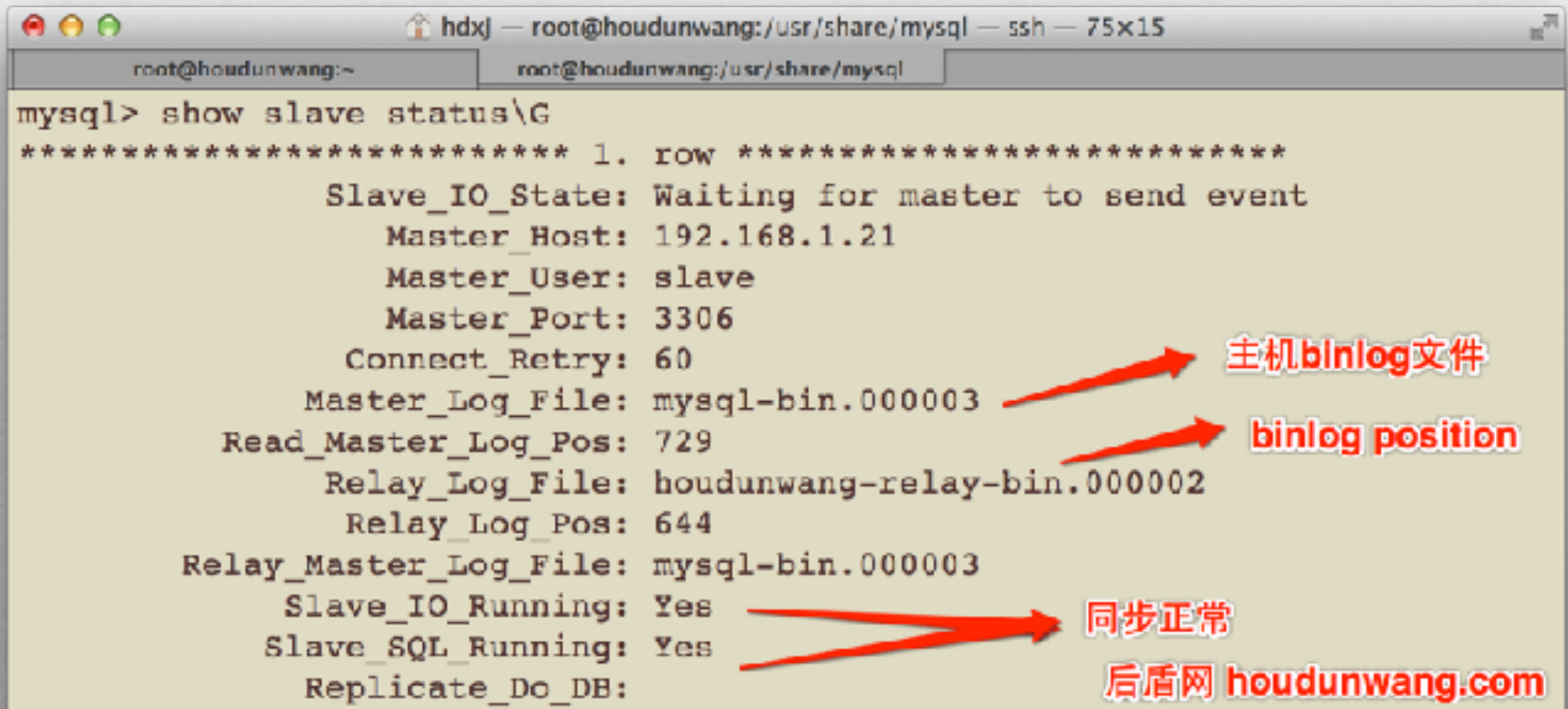


A terminal window titled 'hdxj — root@houdunwang:~ — ssh — 80x10' shows the output of the 'show processlist' command. The output lists a single process with ID 5, user 'slave', host '192.168.1.22:43535', and command 'Binlog Dump'. The state is 'Master has sent all binlog to slave; waiting for binlog to be updated'. A red arrow points from the text 'slave同步完成' to the 'State' field.

```
mysql> show processlist\G
***** 1. row *****
  Id: 5
  User: slave
  Host: 192.168.1.22:43535
  db: NULL
Command: Binlog Dump
  Time: 651
State: Master has sent all binlog to slave; waiting for binlog to be updated
Info: NULL
```

查看状态—主服务器执行

show slave status\G



The screenshot shows a terminal window with the command `mysql> show slave status\G` and its output. Red arrows point from specific output lines to Chinese annotations on the right side of the terminal.

```
mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 192.168.1.21
      Master_User: slave
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000003
      Read_Master_Log_Pos: 729
      Relay_Log_File: houdunwang-relay-bin.000002
      Relay_Log_Pos: 644
      Relay_Master_Log_File: mysql-bin.000003
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
```

主机binlog文件
binlog position
同步正常
后盾网 houdunwang.com

查看状态—从服务器执行