

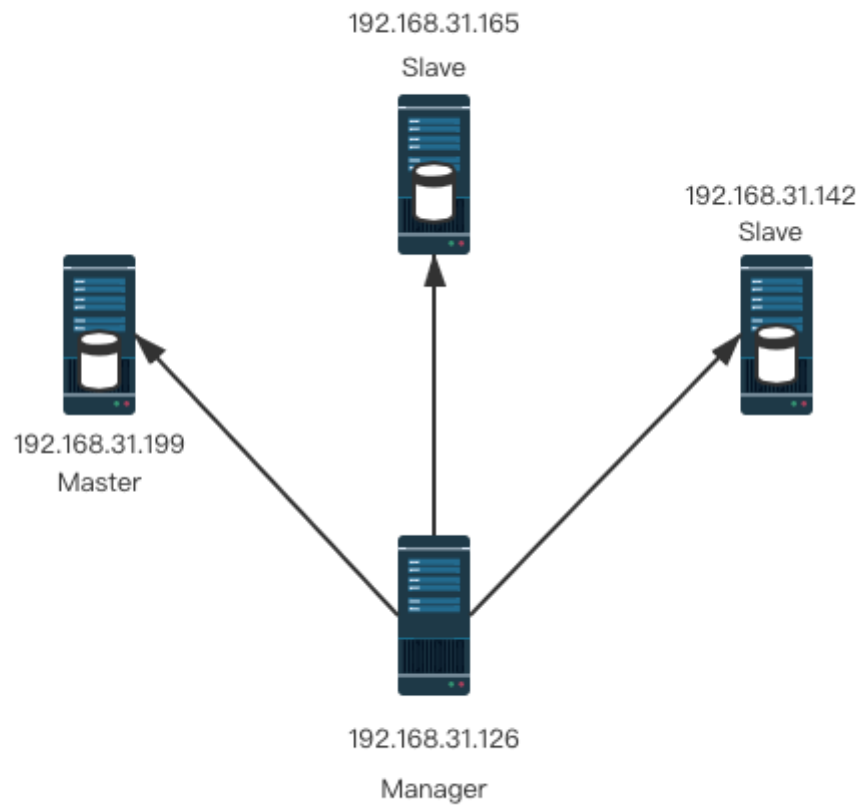
## 1. 环境软件版本

环境&软件	版本
虚拟机&VMware Workstation	15 Pro
服务器&CentOS	7.8
数据库&Mysql	5.7.28
远程连接&Xshell	6
远程文件传输&WinSCP、Xftp	

## 2. 环境架构介绍

架构如图所示，4台机器的IP和角色如下表：

机器名称	IP	角色	权限
Mysql_Master	192.168.31.199	数据库Master	可读写、主库
Mysql_Slave1	192.168.31.165	数据库Slave	只读、从库
Mysql_Slave2	192.168.31.142	数据库Slave	只读、从库
Mysql_MHA	192.168.31.126	MHA Manager	高可用监控



## 3. MySQL主从搭建

### 3.1 MySQL安装 ( 3台 )

#### 下载

```
wget https://cdn.mysql.com/archives/mysql-5.7/mysql-5.7.28-1.el7.x86_64.rpm-bundle.tar
```

#### 解压

```
tar xvf mysql-5.7.28-1.el7.x86_64.rpm-bundle.tar
mysql-community-embedded-5.7.28-1.el7.x86_64.rpm
mysql-community-libs-compat-5.7.28-1.el7.x86_64.rpm
mysql-community-devel-5.7.28-1.el7.x86_64.rpm
mysql-community-embedded-compat-5.7.28-1.el7.x86_64.rpm
mysql-community-libs-5.7.28-1.el7.x86_64.rpm
mysql-community-test-5.7.28-1.el7.x86_64.rpm
mysql-community-common-5.7.28-1.el7.x86_64.rpm
mysql-community-embedded-devel-5.7.28-1.el7.x86_64.rpm
mysql-community-client-5.7.28-1.el7.x86_64.rpm
mysql-community-server-5.7.28-1.el7.x86_64.rpm
```

#### 安装

要移除CentOS自带的mariadb-libs，否则会提示冲突

```
rpm -qa|grep mariadb  
rpm -e mariadb-libs-5.5.41-2.el7_0.x86_64 --nodeps
```

由于MySQL的server服务依赖了common、libs、client，所以需要按照以下顺序依次安装。

RPM是Red Hat公司随Redhat Linux推出的一个软件包管理器，通过它能够更加方便地实现软件的安装。rpm常用的命令有以下几个：

```
-i, --install 安装软件包  
-v, --verbose 可视化，提供更多的详细信息的输出  
-h, --hash 显示安装进度  
-U, --upgrade=<packagefile>+ 升级软件包  
-e, --erase=<package>+ 卸载软件包  
--nodeps 不验证软件包的依赖
```

组合可得到几个常用命令：

```
安装软件: rpm -ivh rpm包名  
升级软件: rpm -Uvh rpm包名  
卸载软件: rpm -e rpm包名  
查看某个包是否被安装 rpm -qa | grep 软件名称
```

下面就利用安装命令来安装mysql：

```
rpm -ivh mysql-community-common-5.7.28-1.el7.x86_64.rpm  
rpm -ivh mysql-community-libs-5.7.28-1.el7.x86_64.rpm  
rpm -ivh mysql-community-libs-compat-5.7.28-1.el7.x86_64.rpm  
rpm -ivh mysql-community-client-5.7.28-1.el7.x86_64.rpm  
rpm -ivh mysql-community-server-5.7.28-1.el7.x86_64.rpm  
rpm -ivh mysql-community-devel-5.7.28-1.el7.x86_64.rpm
```

## 启动

初始化用户

```
mysqld --initialize --user=mysql
```

查看初始密码

```
cat /var/log/mysqld.log | grep password
```

启动mysql服务

```
systemctl start mysqld
```

配置为开机启动

```
systemctl enable mysqld
```

接下来修改默认密码。

```
mysql> SET PASSWORD = PASSWORD('123456');  
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

## 3.2 关闭防火墙

不同的MySQL直接要互相访问，需要关闭Linux的防火墙，否则就要在配置/etc/sysconfig/iptables中增加规则。配置防火墙不是本次作业的重点，所以四台服务器均关闭防火墙。

```
systemctl stop firewalld
```

## 3.3 MySQL主从配置

### Master节点

#### 修改Master配置文件

```
#bin_log配置  
log_bin=mysql-bin  
server-id=1  
sync-binlog=1  
binlog-ignore-db=information_schema  
binlog-ignore-db=mysql  
binlog-ignore-db=performance_schema  
binlog-ignore-db=sys  
#relay_log配置  
relay_log=mysql-relay-bin  
log_slave_updates=1  
relay_log_purge=0  
#gtid配置  
#开启gtid  
#gtid_mode=on  
#enforce_gtid_consistency=1
```

#### 重启服务

```
systemctl restart mysqld
```

#### 主库给从库授权

在MySQL命令行执行如下命令：

```
grant replication slave on *.* to root@'%' identified by 'root';  
grant all privileges on *.* to root@'%' identified by 'root';  
  
flush privileges;  
show master status; //查看主库master_log_file='mysql-  
bin.000007',master_log_pos=154
```

### Slave节点

修改Slave配置文件，两台Slave的server-id分别设置为2和3

```
#bin_log配置  
log_bin=mysql-bin  
#服务器ID,从库1是2,从库2是3
```

```
server-id=2
sync-binlog=1
binlog-ignore-db=information_schema
binlog-ignore-db=mysql
binlog-ignore-db=performance_schema
binlog-ignore-db=sys
#relay_log配置
relay_log=mysql-relay-bin
log_slave_updates=1
relay_log_purge=0
read_only=1
#gtid配置
#开启gtid
#gtid_mode=on
#enforce_gtid_consistency=1
```

## 重启服务

```
systemctl restart mysqld
```

## 开启同步

在Slave节点的MySQL命令行执行如下命令：

```
change master to
master_host='192.168.31.199',master_port=3306,master_user='root',master_password
='123456',master_log_file='mysql-bin.000007',master_log_pos=154;

start slave; // 开启同步
```

## 3.4 配置半同步复制

### Master节点

#### 安装插件

```
install plugin rpl_semi_sync_master soname 'semisync_master.so';
show variables like '%semi%'
```

#### 修改配置文件

```
# 自动开启半同步复制
rpl_semi_sync_master_enabled=ON
rpl_semi_sync_master_timeout=1000
```

## 重启服务

```
systemctl restart mysqld
```

## Slave节点

两台Slave节点都执行以下步骤。

#### 安装插件

```
install plugin rpl_semi_sync_slave soname 'semisync_slave.so';
```

### 修改配置文件

```
# 自动开启半同步复制  
rpl_semi_sync_slave_enabled=ON
```

### 重启服务

```
systemctl restart mysqld
```

### 测试半同步状态

首先通过MySQL命令行检查参数的方式，查看半同步是否开启。

```
show variables like '%semi%'
```

然后通过MySQL日志再次确认。

```
cat /var/log/mysqld.log
```

可以看到日志中已经启动半同步：

```
Start semi-sync binlog_dump to slave (server_id: 2), pos(mysql-bin.000005, 154)
```

## 4. MHA高可用搭建

### 四台机器ssh互通

在四台服务器上分别执行下面命令，生成公钥和私钥，换行回车采用默认值

```
ssh-keygen -t rsa
```

在三台MySQL服务器分别执行下面命令，将公钥拷到MHA Manager服务器上

```
ssh-copy-id 192.168.31.126
```

之后可以在MHA Manager服务器上检查下，看看.ssh/authorized\_keys文件是否包含3个公钥

```
cat /root/.ssh/authorized_keys
```

从MHA Manager服务器执行下面命令，向其他三台MySQL服务器分发公钥信息。

```
scp authorized_keys hadoop25-02:$PWD
```

```
scp /root/.ssh/authorized_keys 192.168.31.199:$PWD  
scp /root/.ssh/authorized_keys 192.168.31.165:$PWD  
scp /root/.ssh/authorized_keys 192.168.31.142:$PWD
```

可以MHA Manager执行下面命令，检测下是否实现ssh互通。

```
ssh hadoop01;
```

## MHA下载安装

### MHA下载

MySQL5.7对应的MHA版本是0.5.8，所以在GitHub上找到对应的rpm包进行下载，MHA manager和node的安装包需要分别下载：

```
https://github.com/yoshinorim/mha4mysql-manager/releases/tag/v0.58
https://github.com/yoshinorim/mha4mysql-node/releases/tag/v0.58
```

下载后，将Manager和Node的安装包分别上传到对应的服务器。（可使用WinSCP等工具）

- 三台MySQL服务器需要安装node
- MHA Manager服务器需要安装manager和node

提示：也可以使用wget命令在linux系统直接下载获取，例如

```
wget https://github.com/yoshinorim/mha4mysql-
manager/releases/download/v0.58/mha4mysql-manager-0.58-
0.e17.centos.noarch.rpm
```

### MHA node安装

在四台服务器上安装mha4mysql-node。

MHA的Node依赖于perl-DBD-MySQL，所以要先安装perl-DBD-MySQL。

```
yum install perl-DBD-MySQL -y
rpm -ivh mha4mysql-node-0.58-0.e17.centos.noarch.rpm
```

### MHA manager安装

在MHA Manager服务器安装mha4mysql-node和mha4mysql-manager。

MHA的manager又依赖了perl-Config-Tiny、perl-Log-Dispatch、perl-Parallel-ForkManager，也分别进行安装。

```
wget http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
rpm -ivh epel-release-latest-7.noarch.rpm
yum install perl-DBD-MySQL perl-Config-Tiny perl-Log-Dispatch perl-Parallel-
ForkManager -y

wget https://github.com/yoshinorim/mha4mysql-
node/releases/download/v0.58/mha4mysql-node-0.58-0.e17.centos.noarch.rpm
rpm -ivh mha4mysql-node-0.58-0.e17.centos.noarch.rpm
wget https://github.com/yoshinorim/mha4mysql-
manager/releases/download/v0.58/mha4mysql-manager-0.58-0.e17.centos.noarch.rpm
rpm -ivh mha4mysql-manager-0.58-0.e17.centos.noarch.rpm
```

提示：由于perl-Log-Dispatch和perl-Parallel-ForkManager这两个被依赖包在yum仓库找不到，因此安装epel-release-latest-7.noarch.rpm。在使用时，可能会出现下面异常：**Cannot retrieve metalink for repository: epel/x86\_64**。可以尝试使用/etc/yum.repos.d/epel.repo，然后注释掉mirrorlist，取消注释baseurl。

## MHA 配置文件

MHA Manager服务器需要为每个监控的 Master/Slave 集群提供一个专用的配置文件，而所有的 Master/Slave 集群也可共享全局配置。

### 初始化配置目录

```
#目录说明
#/var/log                (CentOS目录)
#      /mha              (MHA监控根目录)
#      /app1             (MHA监控实例根目录)
#      /manager.log      (MHA监控实例日志文件)
mkdir -p /var/log/mha/app1
touch /var/log/mha/app1/manager.log
```

### 配置监控全局配置文件

```
vim /etc/masterha_default.cnf
```

```
[server default]
#用户名
user=root
#密码
password=root
#ssh登录账号
ssh_user=root
#主从复制账号
repl_user=root
#主从复制密码
repl_password=root
#ping次数
ping_interval=1
#二次检查的主机
secondary_check_script=masterha_secondary_check -s 192.168.31.199 -s
192.168.31.165 -s 192.168.31.142
```

### 配置监控实例配置文件

```
vim /etc/mha/app1.cnf
```

```
[server default]
#MHA监控实例根目录
manager_workdir=/var/log/mha/app1
#MHA监控实例日志文件
manager_log=/var/log/mha/app1/manager.log

#[serverx]          服务器编号
#hostname            主机名
#candidate_master    可以做主库
#master_binlog_dir   binlog日志文件目录

[server1]
hostname=192.168.31.199
candidate_master=1
master_binlog_dir="/var/lib/mysql"

[server2]
```



```
hostname=192.168.31.165
candidate_master=1
master_binlog_dir="/var/lib/mysql"

[server3]
hostname=192.168.31.142
candidate_master=1
master_binlog_dir="/var/lib/mysql"
```

## MHA 配置检测

### 执行ssh通信检测

在MHA Manager服务器上执行：

```
masterha_check_ssh --conf=/etc/mha/app1.cnf
```

### 检测MySQL主从复制

在MHA Manager服务器上执行：

```
masterha_check_rep1 --conf=/etc/mha/app1.cnf
```

出现“MySQL Replication Health is OK.”证明MySQL复制集群没有问题。

## MHA Manager启动

在MHA Manager服务器上执行：

```
nohup masterha_manager --conf=/etc/mha/app1.cnf --remove_dead_master_conf --
ignore_last_failover < /dev/null > /var/log/mha/app1/manager.log 2>&1 &
```

查看监控状态命令如下：

```
masterha_check_status --conf=/etc/mha/app1.cnf
```

查看监控日志命令如下：

```
tail -f /var/log/mha/app1/manager.log
```

## 测试MHA故障转移

### 模拟主节点崩溃

在MHA Manager服务器执行打开日志命令：

```
tail -200f /var/log/masterha/app1/app1.log
```

关闭Master MySQL服务器服务，模拟主节点崩溃

```
systemctl stop mysqld
```

查看MHA日志，可以看到哪台slave切换成了master

```
show master status;
```

## 将原主启动切换回主

启动MySQL服务

```
systemctl start mysqld
```

挂到新主做从库

```
change master to
master_host='192.168.31.199',master_port=3306,master_user='root',master_password
='root',master_log_file='xxx',master_log_pos=xxx;
start slave; // 开启同步
```

提示：master\_log\_file和master\_log\_pos两个参数需要去新主库查看，show master status \G;

使用MHA在线切换命令将原主切换回来

```
masterha_master_switch --conf=/etc/mha/app1.cnf --master_state=alive --
new_master_host=192.168.31.199 --new_master_port=3306 --orig_master_is_new_slave
--running_updates_limit=10000
```

## 测试SQL脚本

```
create TABLE position (
id int(20),
name varchar(50),
salary varchar(20),
city varchar(50)
) ENGINE=innodb charset=utf8;
```

```
insert into position values(1, 'Java', 13000, 'shanghai');
insert into position values(2, 'DBA', 20000, 'beijing');
```

```
create TABLE position_detail (
id int(20),
pid int(20),
description text
) ENGINE=innodb charset=utf8;
```

```
insert into position_detail values(1, 1, 'Java Developer');

insert into position_detail values(2, 2, 'Database Administrator');
```

## 5. 遇到的问题

### 主从节点配置

配置主从后，检查从节点状态，发现报错：

```
Last_IO_Errno: 1593
Last_IO_Error: Fatal error: The slave I/O thread stops because master and slave
have equal MySQL server UUIDs; these UUIDs must be different for replication to
work.
```

然后去Master和Slave节点检查UUIDs。

```
cat /var/lib/mysql/auto.cnf
```

发现从节点和主节点的UUIDs果然是一样的，此处是因为我偷了个懒，把199服务器装好之后直接复制虚拟机磁盘，导入Parallels生成了另外两台从节点。

解决办法如下：

```
systemctl stop mysqld.service
rm -rf /var/lib/mysql/auto.cnf
systemctl start mysqld.service
```

之后MySQL会重新生成Server UUIDs。

此时通过show slave status查看Slave状态，又报了另外一个错误：

```
Last_IO_Errno: 1236
Last_IO_Error: Got fatal error 1236 from master when reading data from binary
log: 'Could not find first log file name in binary log index file'
```

这是因为一开始搭建主从同步之后并没有成功，我在Master执行了一些命令，没有同步到Slave，导致Slave的日志读取位置发生不一致。

此处有两种做法：第一种是从新去Master节点查看最新的pos，然后重新设置为最新pos，但是这种做法会丢失数据。第二种是正确的做法：

```
stop slave;
reset slave;
start slave;
```

先停止Slave，然后重置Slave，它的作用是“删除SLAVE数据库的relaylog日志文件，并重新启用新的relaylog文件”，然后再重启Slave。

此处文件改动比较小，采用这种做法不会丢失数据，但是如果数据量比较大了，这种做法同步起来就会比较耗时了。

## MHA安装异常

安装perl-DBD-MySQL的过程中遇到了一个报错，提示缺少libmysqlclient.so.18。

```
Error: Package: perl-DBD-MySQL-4.023-6.el7.x86_64 (base)
Requires: libmysqlclient.so.18()(64bit)
Error: Package: perl-DBD-MySQL-4.023-6.el7.x86_64 (base)
Requires: libmysqlclient.so.18(libmysqlclient_18)(64bit)
```

这是由于之前安装MySQL的时候没有把所有的包都安装，解决办法，重新安装libs-compat包。

```
rpm -ivh mysql-community-libs-compat-5.7.28-1.el7.x86_64.rpm
```

## MHA配置检测失败

执行命令`masterha_check_repl -conf=/etc/app1.cnf`后报错：

```
[error][/usr/share/perl5/vendor_perl/MHA/MasterMonitor.pm, ln364] None of slaves  
can be master. Check failover configuration file or log-bin settings in my.cnf
```

发现是两个从节点没有配置log-bin日志，所以需要在从节点开启log-bin日志。在从节点的配置文件增加“log\_bin=mysql-bin”；

之后执行又报了错误：

```
[error][/usr/share/perl5/vendor_perl/MHA/ServerManager.pm, ln443] Binlog  
filtering check failed on 192.168.31.165(192.168.31.165:3306)! All log-bin  
enabled servers must have same binlog filtering rules (same binlog-do-db and  
binlog-ignore-db). Check SHOW MASTER STATUS output and set my.cnf correctly.
```

于是在各Slave节点增加以下配置：

```
sync-binlog=1 #每次写入都同步到binlog  
binlog_format=ROW  
binlog-ignore-db=performance_schema #忽略不同步  
binlog-ignore-db=information_schema  
binlog-ignore-db=sys
```

不小心在主库上启动了salve，在执行`masterha_check_repl -conf=/etc/app1.cnf`时就报错：

```
Fri May 29 00:29:14 2020 - [info] Slaves settings check done.  
Fri May 29 00:29:14 2020 - [info]  
192.168.31.199(192.168.31.199:3306) (current master)  
+--192.168.31.199(192.168.31.199:3306)  
+--192.168.31.165(192.168.31.165:3306)  
+--192.168.31.142(192.168.31.142:3306)  
  
Fri May 29 00:29:14 2020 - [info] Checking replication health on  
192.168.31.199..  
Fri May 29 00:29:14 2020 - [error][/usr/share/perl5/vendor_perl/MHA/Server.pm,  
ln490] Slave IO thread is not running on 192.168.31.199(192.168.31.199:3306)  
Fri May 29 00:29:14 2020 - [error]  
[/usr/share/perl5/vendor_perl/MHA/ServerManager.pm, ln1526] failed!  
Fri May 29 00:29:14 2020 - [error]  
[/usr/share/perl5/vendor_perl/MHA/MasterMonitor.pm, ln427] Error happened on  
checking configurations. at /usr/share/perl5/vendor_perl/MHA/MasterMonitor.pm  
line 420.
```

在主库上停止salve，然后执行reset slave all。