MHA 一主一从构建

吴炳锡

MHA 介绍

MHA 是自动化完成 MySQL Replcation 结构的故障转移。 可以快速完成将从服务器晋级为主服务器（通常 10-30s）,不会造成同步中数据不一致，而且不需要购买更多的服务器，也不会有性能损耗，没有复杂的布署(非常容易安装), 最为利害的也不用更改现有的布署环境。

MHA 当然也提供在线的有计划的维护主从切换： 可以更改一个正在运行的 Master 到一个新的 Master 也是很安全的，整个过程只需要 0.5-2 秒对于写的阻塞。

安装基本环境介绍

一主一从：

Mha manager & slave	Node20	192.168.11.20
Master 机器	Node21	192.168.11.21
vip		192.168.11.100

思考：MHA manager 为什么要放置到 slave 上。

安装步骤

两台物理机器信任机制建立

在 192.168.11.20 使用 ssh-keygen 生成 key

#ssh-keygen 一路回车就行了。最后会在~/.ssh 下面产生：id_rsa id_rsa.pub 两个文件。

生成信任文件：

```
#cd ~/.ssh/
#cat id_rsa.pub >authorized_keys
#chmod 600 *
```

保留.ssh 下面只有 id_rsa, id_rsa.pub 其它的文件可以删或是备份移走。

```
#cd ~
#scp –r .ssh    192.168.11.21:~/
```

注意目标机器上文件权限为: 600

MySQL 安装

MySQL 建议使用 MySQL-5.5.X 后版本。 自行安装，完成一主三从的结构。（主从配置不在描述）

如果使用 mysql-5.6 建议禁用 GTID。

大致的步骤：

MySQL 安装完毕后初始化帐号相关：

```
Delete from mysql.user where user!='root' or host!='localhost';
Truncate table mysql.db;
Drop database test;
Flush privileges;
grant all privileges on *.* to 'wubx'@'%' identified by 'wubxwubx';
grant replication slave on *.* to 'repl' identified by 'repl4slave';
```

构建: node20-> node21 的复制

在主节点上执行 sh /etc/masterha/init_vip.sh 绑定 VIP。

MHA 相关安装

Mha 的安装分为 mha manager 节点和 mha node 节点。需文件不太一样。安装文件可以从 https://code.google.com/p/mysql-master-ha/wiki/Downloads 下载。参考打包里的 rpm 包。

在一主一从的结构中建议两个节点都要安装 manger, node 包。

机器上安装

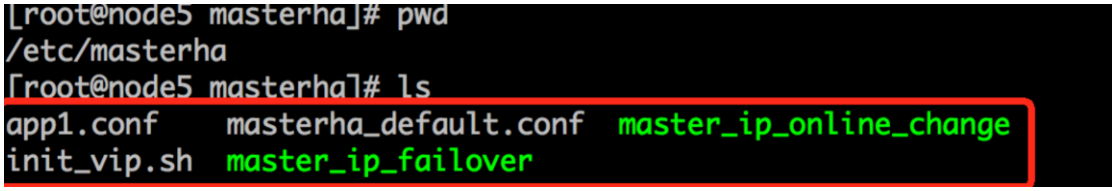
```
# yum install perl-DBD-MySQL
# yum install perl-Config-Tiny
# yum install perl-Log-Dispatch
# yum install perl-Parallel-ForkManager
```

```
[root@wubx public]# rpm -ivh mha4mysql-node-0.56-0.el6.noarch.rpm
Preparing...                ########################################### [100%]
   1:mha4mysql-node         ########################################### [100%]
[root@wubx public]# rpm -ivh mha4mysql-manager-0.56-0.el6.noarch.rpm
Preparing...                ########################################### [100%]
   1:mha4mysql-manager      ########################################### [100%]
```

确认安装没有错误。

配置

MHA 的配置相关简单，只用在 manager 节点上进行配置限可。配置文件最少一个，一般可以分成两个这样为了减少一个 manager 节点管理多个集群时可以少写一点配置。 但我们这里一主一从结构，建议配置文件配置完毕后，放置到两台机器上都是一样的。
配置文件如下：



全局级配置文件：**/etc/masterha/masterha_default.conf**

```
[server default]
#MySQL 的用户和密码
user=wubx
password=wubxwubx

#系统 ssh 用户
ssh_user=root

#复制用户
repl_user=repl
repl_password= repl4mha


#监控
ping_interval=1
shutdown_script=""

#切换调用的脚本
master_ip_failover_script= /etc/masterha/master_ip_failover
master_ip_online_change_script= /etc/masterha/master_ip_online_change
```

说明 shutdown_script 主要用来设置在 master 进行切换时，要执行的脚本动作，这个动作，可以设置把机器关了来防止脑列，也可以用做一些其它动作（前提那台机器还活着）。

[root@wubx .ssh]# cat /etc/masterha/app1.cnf

```
[server default]
```

```
#mha manager 工作目录
manager_workdir = /var/log/masterha/app1
manager_log = /var/log/masterha/app1/app1.log
remote_workdir = /var/log/masterha/app1

[server1]
hostname=192.168.1.20
master_binlog_dir = /data/mysql/mysql3306/logs
candidate_master = 1
check_repl_delay = 0        #用防止 master 故障时，切换时 slave 有延迟，卡在那里切不过来。

[server2]
hostname=192.168.1.21
master_binlog_dir=/data/mysql/mysql3306/logs
candidate_master=1
check_repl_delay=0
```

里面的 IP 根据实际情况更改。

配置文件测试

测试 Ssh ok

 # masterha_check_ssh --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf

确认可以看到所有的服务器上 ssh 测试通过。

查看是不是具备跑 masterha_manger, 主从结构是不是 OK 之类。

#masterha_check_repl --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf

最终可以看到现有 master/slave 结构。

MHA 启动及关闭

```
# masterha_manager --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf > /tmp/mha_manager.log 2>&1  &
```

master 去执行：

#sh /etc/masterha/init_vip.sh

确认 VIP 绑定成功，如果业务按 VIP 配置的访问 DB，应该已经可以正常访问。

注意：

第一次起动，主库上的 VIP 不会自动绑定，需要手功调用 init_vip.sh 去绑定，主库发生故障切换会进行 vip 的漂移。

检查是否启动：

```
#masterha_check_status --global_conf=/etc/masterha/masterha_default.conf
--conf=/etc/masterha/app1.conf
```

停止 mha

```
#masterha_stop --global_conf=/etc/masterha/masterha_default.conf
--conf=/etc/masterha/app1.conf
Stopped app1 successfully.
[1]+ Exit 1 nohup masterha_manager
--global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf >
/tmp/mha_manager.log 2>&1
```

MHA 日常维护命令集

1.查看 ssh 登陆是否成功

masterha_check_ssh **--global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf**

2.查看复制是否建立好

masterha_check_repl --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf

3.启动 mha

nohup masterha_manager --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf > /tmp/mha_manager.log < /dev/null 2>&1 &

当有 slave 节点宕掉的情况是启动不了的，加上--ignore_fail_on_start 即使有节点宕掉也能启动 mha

nohup masterha_manager --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf --ignore_fail_on_start > /tmp/mha_manager.log < /dev/null 2>&1 &
需要在配置文件中设置 ignore_fail=1

4.检查启动的状态
masterha_check_status--global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf

5.停止 mha
masterha_stop --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf

6.failover 后下次重启
每次 failover 切换后会在管理目录生成文件 app1.failover.complete ，下次在切换的时候会发现有这个文件导致切换不成功，需要手动清理掉。
rm -rf /masterha/app1/app1.failover.complete
也可以加上参数--ignore_last_failover

7.手工 failover
手工 failover 场景，master 死掉，但是 masterha_manager 没有开启，可以通过手工 failover：
masterha_master_switch --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf --dead_master_host=old_ip --master_state=dead --new_master_host=new_ip --ignore_last_failover

8.masterha_manager 是一种监视和故障转移的程序。另一方面,masterha_master_switch 程序不监控主库。 masterha_master_switch 可以用于主库故障转移,也可用于在线总开关。

9.手动在线切换
masterha_master_switch --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf --master_state=alive --new_master_host=192.168.199.78 --orig_master_is_new_slave
或者
masterha_master_switch --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf --master_state=alive --new_master_host=192.168.199.78 -orig_master_is_new_slave --running_updates_limit=10000

--orig_master_is_new_slave 切换时加上此参数是将原 master 变为 slave 节点，如果不加此参数，原来的 master 将不启动

--running_updates_limit=10000 切换时候选 master 如果有延迟的话，mha 切换不能成功，加上此参数表示延迟在此时间范围内都可切换（单位为 s），但是切换的时间长短是由 recover 时 relay 日志的大小决定

手动在线切换 mha，切换时需要将在运行的 mha 停掉后才能切换。
在备库先执行 DDL，一般先 stop slave，一般不记录 mysql 日志，可以通过 set SQL_LOG_BIN = 0 实现。然后进行一次主备切换操作，再在原来的主库上执行 DDL。这种方法适用于增减索引，如果是增加字段就需要额外注意。

Online master switch 开始只有当所有下列条件得到满足。
1. IO threads on all slaves are running // 在所有 slave 上 IO 线程运行。
2. SQL threads on all slaves are running //SQL 线程在所有的 slave 上正常运行。
3. Seconds_Behind_Master on all slaves are less or equal than --running_updates_limit seconds // 在所有的 slaves 上 Seconds_Behind_Master 要小于等于 running_updates_limit seconds
4. On master, none of update queries take more than --running_updates_limit seconds in the show processlist output // 在主上，没有更新查询操作多于 running_updates_limit seconds 在 show processlist 输出结果上。

可以通过如下命令停止 mha
masterha_stop --global_conf=/etc/masterha/masterha_default.conf --conf=/etc/masterha/app1.conf

思考：

如果从库进行了一次 failover 操作后，下一次 masterha_manager 需要启动那台机器上。