架构图： **4**

前　　言 **5**

摘　　要 *5*

一．Mha 优点 *5*

1.M aster crash 时可以快速的进行故障切换。 5

2.M aster crash 时可以最大化的减少数据丢失 5

3.Semi-Synchronous Replication 可以最大化的减少数据的丢失。 6

4.原来应用系统整体性能不会降低太多 6

5.MHA 适合任何存储引擎 6

二．安　装 MYSQL 主从 *7*

1.首先在三台机器上装编译安装 MYSQL-5.6.10 7

2.配置 HOSTS 环境 9

3.安装 MYSQL 主从半同步 10

4.每一台机器配置互相无交互 12

5.配置主从 13

三．安装配置 MHA *14*

1.安装 14

2.配置 16

3.测试 ssh 18

4.注意 20

5.测试 mysql 22

四．测试重构 *26*

1.测试 26

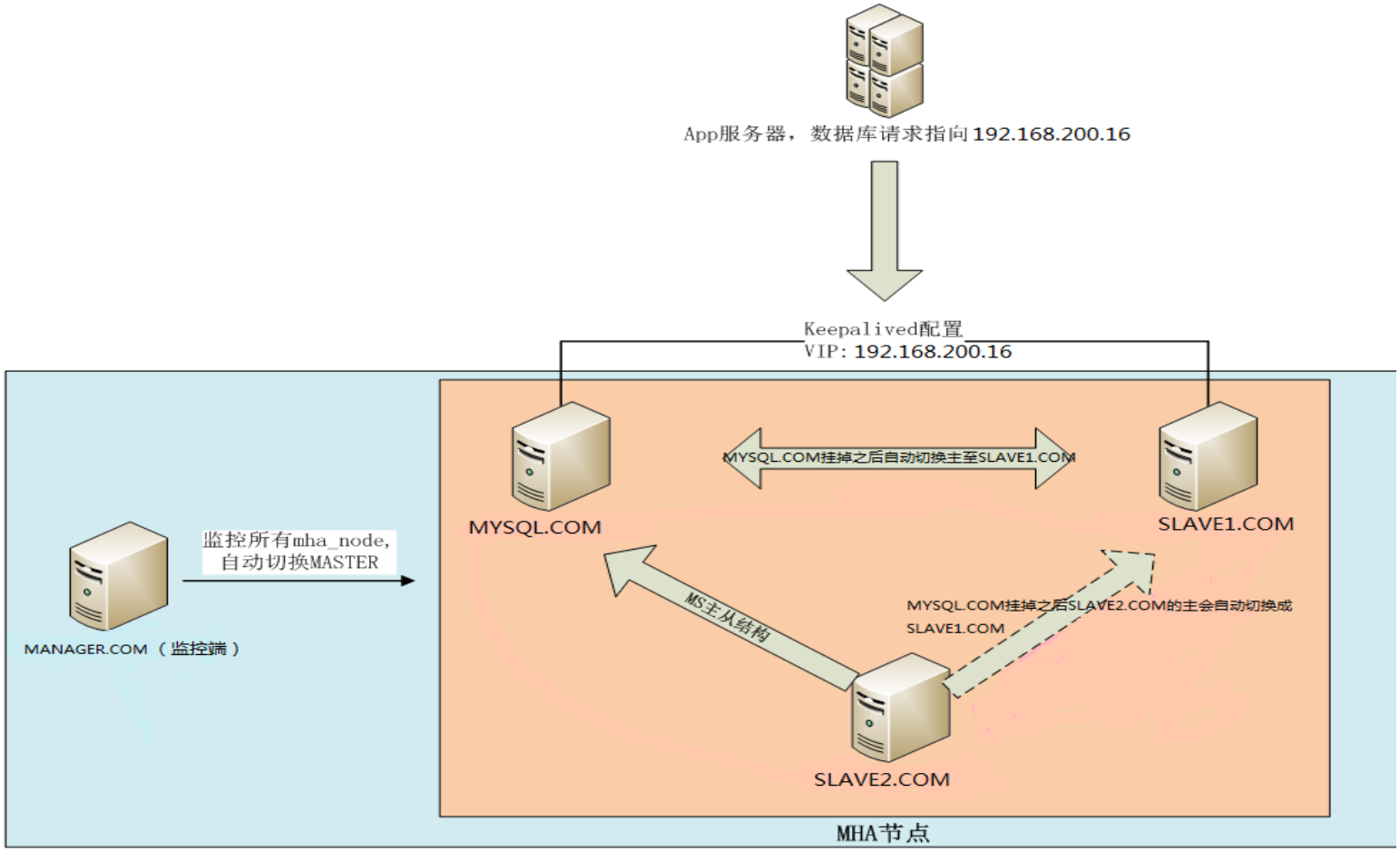
2.重构 35
五．扩展 Keepalived *37*
1.安装 keepalived 37
2.配置 keepalived 38
3.检测 mysql 服务脚本 39
4.启动测试 41
六.mha 日常维护命令 *43*

MHA 安装维护手册

[√] 初稿	[√] 工作手册	完成日期	2014-07-02
[ ] 修改	[ ] 部门报告	编写人	左越宗
[ ] 审核	[ ] 对外公告	审核人	
[ ] 发布	[ ] 制度规范	发布部门	技术部
[√] 保密	[ ] 交流反馈	发布公司	

读此文档者我已默认你会安装配置主从

架构图：


App服务器，数据库请求指向192.168.200.16
Keepalived配置
VIP：192.168.200.16
MYSQL.COM
MYSQL.COM挂掉之后自动切换主至SLAVE1.COM
SLAVE1.COM
监控所有mha_node，
自动切换MASTER
MANAGER.COM（监控端）
MS主从结构
MYSQL.COM挂掉之后SLAVE2.COM的主会自动切换成
SLAVE1.COM
SLAVE2.COM
MHA节点

前　　言

摘　要

MHA 是当 master 出现故障，挑选一个 slave 作为新的 master 并构建成新的主从架构的管理工具。从 master 出现故障到构建成新的主从架构时间是 10-30 秒。在 master 出现故障时可能会出现 slave 同步的数据不一致的现象，此工具可以自动应用差异的中继日志到其他 slave 上保证数据的一致性。

一. Mha 优点

1.M aster crash 时可以快速的进行故障切换。

9-12 秒内可以检测到 master 故障， 7-10 秒内可以关闭 master 机器避免脑裂，在几秒内可以应用差异日志，并构建新的主从架构，整个过程大约在 10-30 秒内可以完成，最大化的减少故障修复时间。

2.M aster crash 时可以最大化的减少数据丢失

当 master crash 时 MHA 自动检测选择数据同步最全的 slave，并把差异日志应用到其他 slave 上， 以保障数据的一致性。结合使用 mysql

3.Semi-Synchronous Replication 可以最大化的减少数据的丢失。

MHA 的更改升级配置等不影响线上正在运行的数据库
使用 mha 不需要增加太多的服务器
MHA 由 MHA Manager 和 MHA Node 组成。 MHA Node 运行在 MYSQL 服务器上，所以不会因为 MHA node 增加新的服务器。MHA Manager 通常需要独立运行在一台服务器上，所以你需要增加一台服务器用于监控管理运行 MHA Manager，但是一台服务器上的 MHA Manager 可以同时监控管理多达百台 master，所以总的来说服务器增加不会太多。MHA Manger 也可以运行在一台 slave 上，这样总的服务器数也不会增加。

4.原来应用系统整体性能不会降低太多

MHA 工作在异步或半同步的主从架构上。当监控 master 时，MHA 会每隔几
Mha 部署测试文档
5
秒 （默认 3 秒） 向 master 发出 ping 包并且不需要大的 sql 语句用于监控 master 的健康状况。Slave 需要开启 binlog，整体性能不会有太大的降低。

5.MHA 适合任何存储引擎

只要能主从复制的存储引擎它都支持，不限于支持事物的 innodb 引擎。
Mha 部署测试文档

二. 安　装

共有 4 台服务器： 一个管理服务器，一个 master 服务器，两个 slave 服务器。
操作系统 Centos 6.4 64 bit 。
192.168.186.141 MYSQL.COM
192.168.186.142 SLAVE1.COM
192.168.186.146 SLAVE2.COM
192.168.186.144 MANAGER.COM
数据库版本 mysql-5.6.10

1.首先在三台机器上装编译安装 MYSQL-5.6.10

关闭 selinux iptables 服务以便后期主从同步不出错
[root@MYSQL ~]# cd /usr/local/src/
[root@MYSQL src]# ls
installmysql5.sh　mysql-5.6.10　mysql-5.6.10.tar.gz
[root@MYSQL src]# sh installmysql5.sh
please enter you mysql version (eg:/mysql-5.5.34):mysql-5.6.10
please enter you mysql datadir (eg:/data/mysql/data):/home/mysql/data


installmysql5.sh

mysql-5.6.10.tar.gz

mha4mysql-node-0.53.tar.gz

mha4mysql-manager-0.53.tar.gz

2.配置 HOSTS 环境

[root@MANAGER ~]# vi /etc/hosts

[root@MYSQL etc]# vi /etc/hosts

```
127.0.0.1      localhost localhost.localdomain localhost4 localhost4.localdomain4
::1            localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.186.141 MYSQL.COM
192.168.186.142 SLAVE1.COM
192.168.186.146 SLAVE2.COM
192.168.186.144 MANAGER.COM
```

```
[root@MYSQL etc]# for i in    142 146 144;do scp /etc/hosts 192.168.186.$i:/etc/;done
root@192.168.186.142's password:
Permission denied, please try again.
root@192.168.186.142's password:
Permission denied, please try again.
root@192.168.186.142's password:
hosts                                                                  100%  266         0.3KB/s   00:00
root@192.168.186.146's password:
hosts                                                                  100%  266         0.3KB/s   00:00
root@192.168.186.144's password:
```

3.安装 MYSQL 主从半同步

```
# 所有 mysql 数据库服务器，安装半同步插件(semisync_master.so,semisync_slave.so)

mysql> install plugin rpl_semi_sync_master soname 'semisync_master.so';

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

```
[root@MYSQL etc]vi /etc/my.cnf
[mysqld]
rpl_semi_sync_master_enabled=1
rpl_semi_sync_master_timeout=1000
rpl_semi_sync_slave_enabled=1
relay_log_purge=0
skip-name-resolve
#socket=/usr/mysql.sock
#auto_increment_offset = 2
#auto_increment_increment = 2
server-id = 1
log-bin=mysql-bin
read_only=1
slave-skip-errors=1396
```

三台机器可以全部开启 仅仅 server-id 不同

```
mysql> show variables like '%sync%';

# 查看半同步状态：

mysql> show status like '%sync%';

# 有几个状态参数值得关注的：

rpl_semi_sync_master_status：显示主服务是异步复制模式还是半同步复制模式

rpl_semi_sync_master_clients：显示有多少个从服务器配置为半同步复制模式

rpl_semi_sync_master_yes_tx：显示从服务器确认成功提交的数量

rpl_semi_sync_master_no_tx：显示从服务器确认不成功提交的数量

rpl_semi_sync_master_tx_avg_wait_time：事务因开启 semi_sync，平均需要额外等待的时间

rpl_semi_sync_master_net_avg_wait_time：事务进入等待队列后，到网络平均等待时间
```

[root@MYSQL src]# service mysqld restart 每台机器重启

4.每一台机器配置互相无交互

[root@MYSQL src]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.186.141 MYSQL.COM
192.168.186.142 SLAVE1.COM
192.168.186.146 SLAVE2.COM
192.168.186.144 MANAGER.COM

[root@MYSQL src]# ssh-keygen
[root@MYSQL src]# ssh-copy-id 192.168.186.142
[root@MYSQL src]# ssh-copy-id 192.168.186.144
[root@MYSQL src]# ssh-copy-id 192.168.186.146

其他台配置方法与以上一致 保证每台互相无交互
配置完成后每一个都登录一次
[root@MYSQL ~]# ssh MANAGER.COM
[root@MYSQL ~]# ssh SLAVE1.COM
[root@MYSQL ~]# ssh SALVE2.COM
作用首次连接需要输入一次 YES 在 know-hosts 问价记录，达成无交互

5.配置主从

执行主从脚本
建议自己做 用这个脚本要是后期带来各种不便 不要怪我


mslave.sh

[root@MYSQL src]# sh mslave.sh
please enter you mysql SLAVEIP (eg:192.168.152.138):192.168.186.142
please enter you master mysql password (eg:yunwei123):123
please enter you slave mysql password (eg:yunwei123):123
please enter you master mysql binlog (eg:mysql-bin.000001):mysql-bin.000001

[root@MYSQL src]# sh mslave.sh
please enter you mysql SLAVEIP (eg:192.168.152.138):192.168.186.146
please enter you master mysql password (eg:yunwei123):123 这个是你的主 MYSQL 登录密码
please enter you slave mysql password (eg:yunwei123):123 这个是你的从的 MYSQL 登录密码
please enter you master mysql binlog (eg:mysql-bin.000001):mysql-bin.000001

至此 MYSQL 安装主从半同步配置完成

三．安装配置 MHA

1.安装

每台机器做如下操作

[root@SLAVE2 data]#rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm

[root@MANAGER src]# yum clean all
Loaded plugins: fastestmirror, refresh-packagekit, security
Cleaning repos: epel name
Cleaning up Everything
Cleaning up list of fastest mirrors

[root@MANAGER src]# yum makecache

[root@MANAGER src]# rpm --import /etc/pki/rpm-gpg/*

[root@SLAVE2 data]# yum -y install perl-DBD-MySQL perl-Config-Tiny perl-Log-Dispatch perl-Parallel-ForkManager perl-Config-IniFiles ncftp perl-Params-Validate perl-CPAN perl-Test-Mock-LWP.noarch perl-LWP-Authen-Negotiate.noarch perl-devel

[root@SLAVE2 data]#**yum install perl-ExtUtils-CBuilder perl-ExtUtils-MakeMaker** 这个百度文档没有但是显然是要装的

以下操作管理节点需要两个都安装，在3台数据库节点只要安装 MHA 的 node 节点：

```
# 如果安装下面包，报依赖关系错，请先安装 mysql-share-compat 包
# 先安装下面的 perl-dbd-mysql 包
# 在下面执行 perl 时，如果出现报错，需要安装如下这几个 perl 包： perl-devel perl-CPAN
```

[root@MANAGER src]# tar -xf mha4mysql-node-0.53.tar.gz
[root@MANAGER src]# cd mha4mysql-node-0.53
[root@MANAGER mha4mysql-node-0.53]# perl Makefile.PL
[root@MANAGER mha4mysql-node-0.53]# make && make install　　黄色字体这部分是每一台节点都要做的

[root@MANAGER src]# tar -xf　mha4mysql-manager-0.53.tar.gz
[root@MANAGER src]# cd mha4mysql-manager-0.53
[root@MANAGER mha4mysql-manager-0.53]# perl Makefile.PL
[root@MANAGER mha4mysql-manager-0.53]# make && make install
根据提示输入 如果中间有卡顿现象 直接 crtl+c 然后继续会出现下载的进度条，说明是正常的
[root@MANAGER src]# mkdir /etc/masterha

[root@MANAGER mha]# mkdir -p /master/app1
[root@MANAGERmha]# mkdir -p /scripts

[root@MANAGER mha]# cp samples/conf/* /etc/masterha/

[root@MANAGERmha]# cp samples/scripts/*　/scripts

[root@MANAGER mha4mysql-manager-0.53]# cp samples/conf/* /etc/masterha/

2.配置

[root@MANAGER masterha]# vi app1.cnf

内容如下;

[server default]

manager_workdir=/masterha/app1

manager_log=/masterha/app1/manager.log

user=mha_mon

password=123

ssh_user=root

repl_user=slave 做主从的用户 这个也是每一台都要授权的

repl_password=yunwei123 做主从的密码

ping_interval=1

```
shutdown_script=""

master_ip_online_change_script=""

report_script=""


[server1]

hostname=192.168.186.141

master_binlog_dir=/data/mysql/data

candidate_master=1


[server2]

hostname=192.168.186.142

master_binlog_dir=/data/mysql/data

candidate_master=1
```

[server3]

hostname=192.168.186.146

master_binlog_dir=/data/mysql/data
no_master=1

保存退出！
[root@MANAGER masterha]# >masterha_default.cnf

3.测试 ssh

[root@MANAGER masterha]# masterha_check_ssh --global_conf=/etc/masterha/masterha_default.cnf --conf=/etc/masterha/app1.cnf
Wed Jul 9 02:26:57 2014 - [info] Reading default configuratoins from /etc/masterha/masterha_default.cnf..
Wed Jul 9 02:26:57 2014 - [info] Reading application default configurations from /etc/masterha/app1.cnf..
Wed Jul 9 02:26:57 2014 - [info] Reading server configurations from /etc/masterha/app1.cnf..
Wed Jul 9 02:26:57 2014 - [info] Starting SSH connection tests..
Wed Jul 9 02:26:58 2014 - [debug]
Wed Jul 9 02:26:57 2014 - [debug] Connecting via SSH from root@192.168.186.141(192.168.186.141:22) to root@192.168.186.142(192.168.186.142:22)..
Wed Jul 9 02:26:57 2014 - [debug] ok.
Wed Jul 9 02:26:57 2014 - [debug] Connecting via SSH from root@192.168.186.141(192.168.186.141:22) to

root@192.168.186.146(192.168.186.146:22)..
Wed Jul 9 02:26:57 2014 - [debug] ok.
Wed Jul 9 02:26:58 2014 - [debug]
Wed Jul 9 02:26:57 2014 - [debug] Connecting via SSH from root@192.168.186.142(192.168.186.142:22) to root@192.168.186.141(192.168.186.141:22)..
Wed Jul 9 02:26:57 2014 - [debug] ok.
Wed Jul 9 02:26:57 2014 - [debug] Connecting via SSH from root@192.168.186.142(192.168.186.142:22) to root@192.168.186.146(192.168.186.146:22)..

Wed Jul 9 02:26:58 2014 - [debug] ok.
Wed Jul 9 02:26:58 2014 - [debug]
Wed Jul 9 02:26:58 2014 - [debug] Connecting via SSH from root@192.168.186.146(192.168.186.146:22) to root@192.168.186.141(192.168.186.141:22)..
Wed Jul 9 02:26:58 2014 - [debug] ok.
Wed Jul 9 02:26:58 2014 - [debug] Connecting via SSH from root@192.168.186.146(192.168.186.146:22) to root@192.168.186.142(192.168.186.142:22)..
Wed Jul 9 02:26:58 2014 - [debug] ok.
Wed Jul 9 02:26:58 2014 - [info] All SSH connection tests passed successfully.

登入每台数据库
mysql> grant all privileges on *.* to mha_mon@'%' identified by '123';
Query OK, 0 rows affected (1.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

4.注意

[root@SLAVE1 ~]# ln -s /usr/local/mysql/bin/* /usr/bin 在每台 MYSQL 服务器上做这件事情 极度重要哦

mysql>set global read_only=1; set global relay_log_purge=0; 在从上执行 或者干脆写到 my.cnf 文件里面最好

[root@SLAVE1 ~]# vi /etc/my.cnf

read_only=1
slave-skip-errors=1396
为什么要跳过这个错误呢 因为啊在主里面删除用户的时候 从会报错说没有这个用户所以跳过这个错误吧

如果数据库存在空的用户 域名的用户 一定要删除否则 MHA 连接 MYSQL 会报错连不上 一般只要在从上面删除 如果直接没删除也 OK 那就 OK 吧
如果报错登录不了就删除掉吧 或者跳过域名解析，或者你授权的时候记得也授权域名等等方法多种 我的方式是跳过域名解析的 skip-name-resolve
mysql> select user,host from mysql.user;

```
+---------+---------------+
| user    | host          |
+---------+---------------+
| root    | 127.0.0.1     |
| mha_mon | 192.168.186.% |
| repl    | 192.168.186.% |
| slave   | 192.168.186.% |
| root    | ::1           |
|         | SLAVE2.COM    |
| root    | SLAVE2.COM    |
| root    | localhost     |
+---------+---------------+
8 rows in set (0.00 sec)
mysql> drop user 'root'@SLAVE2.COM;
Query OK, 0 rows affected (0.00 sec)
```

5.测试 mysql

```
[root@MANAGER masterha]#  masterha_check_repl  --conf=/etc/masterha/app1.cnf
Wed Jul  9 04:23:16 2014 - [warning] Global configuration file /etc/masterha_default.cnf not found. Skipping.
Wed Jul  9 04:23:16 2014 - [info] Reading application default configurations from /etc/masterha/app1.cnf..
Wed Jul  9 04:23:16 2014 - [info] Reading server configurations from /etc/masterha/app1.cnf..
Wed Jul  9 04:23:16 2014 - [info] MHA::MasterMonitor version 0.53.
Wed Jul  9 04:23:17 2014 - [info] Dead Servers:
Wed Jul  9 04:23:17 2014 - [info] Alive Servers:
Wed Jul  9 04:23:17 2014 - [info]   192.168.186.141(192.168.186.141:3306)
Wed Jul  9 04:23:17 2014 - [info]   192.168.186.142(192.168.186.142:3306)
Wed Jul  9 04:23:17 2014 - [info]   SLAVE2.COM(192.168.186.146:3306)
Wed Jul  9 04:23:17 2014 - [info] Alive Slaves:
Wed Jul  9 04:23:17 2014 - [info]   192.168.186.142(192.168.186.142:3306)  Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul  9 04:23:17 2014 - [info]     Replicating from 192.168.186.141(192.168.186.141:3306)
Wed Jul  9 04:23:17 2014 - [info]     Primary candidate for the new Master (candidate_master is set)
Wed Jul  9 04:23:17 2014 - [info]   SLAVE2.COM(192.168.186.146:3306)  Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul  9 04:23:17 2014 - [info]     Replicating from 192.168.186.141(192.168.186.141:3306)
Wed Jul  9 04:23:17 2014 - [info]     Not candidate for the new Master (no_master is set)
Wed Jul  9 04:23:17 2014 - [info] Current Alive Master: 192.168.186.141(192.168.186.141:3306)
Wed Jul  9 04:23:17 2014 - [info] Checking slave configurations..
Wed Jul  9 04:23:17 2014 - [info] Checking replication filtering settings..
```

```
Wed Jul  9 04:23:17 2014 - [info]  binlog_do_db= , binlog_ignore_db=
Wed Jul  9 04:23:17 2014 - [info]  Replication filtering check ok.
Wed Jul  9 04:23:17 2014 - [info] Starting SSH connection tests..
Wed Jul  9 04:23:18 2014 - [info] All SSH connection tests passed successfully.
Wed Jul  9 04:23:18 2014 - [info] Checking MHA Node version..
Wed Jul  9 04:23:19 2014 - [info]  Version check ok.
Wed Jul  9 04:23:19 2014 - [info] Checking SSH publickey authentication settings on the current master..
Wed Jul  9 04:23:19 2014 - [info] HealthCheck: SSH to 192.168.186.141 is reachable.
Wed Jul  9 04:23:19 2014 - [info] Master MHA Node version is 0.53.
Wed Jul  9 04:23:19 2014 - [info] Checking recovery script configurations on the current master..
Wed Jul  9 04:23:19 2014 - [info]  Executing command: save_binary_logs --command=test --start_pos=4 --binlog_dir=/data/mysql/data
--output_file=/var/tmp/save_binary_logs_test --manager_version=0.53 --start_file=mysql-bin.000001
Wed Jul  9 04:23:19 2014 - [info]  Connecting to root@192.168.186.141(192.168.186.141)..
  Creating /var/tmp if not exists..    ok.
  Checking output directory is accessible or not..
   ok.
  Binlog found at /data/mysql/data, up to mysql-bin.000001
Wed Jul  9 04:23:20 2014 - [info] Master setting check done.
Wed Jul  9 04:23:20 2014 - [info] Checking SSH publickey authentication and checking recovery script configurations on all alive slave servers..
Wed Jul  9 04:23:20 2014 - [info]  Executing command : apply_diff_relay_logs --command=test --slave_user=mha_mon
--slave_host=192.168.186.142 --slave_ip=192.168.186.142 --slave_port=3306 --workdir=/var/tmp --target_version=5.6.10-log --manager_version=0.53
--relay_log_info=/data/mysql/data/relay-log.info  --relay_dir=/data/mysql/data/  --slave_pass=xxx
Wed Jul  9 04:23:20 2014 - [info]  Connecting to root@192.168.186.142(192.168.186.142:22)..
  Checking slave recovery environment settings..
    Opening /data/mysql/data/relay-log.info ... ok.
    Relay log found at /data/mysql/data, up to SLAVE1-relay-bin.000002
```

```
    Temporary relay log file is /data/mysql/data/SLAVE1-relay-bin.000002
    Testing mysql connection and privileges..Warning: Using a password on the command line interface can be insecure.
 done.
    Testing mysqlbinlog output.. done.
    Cleaning up test file(s).. done.
Wed Jul  9 04:23:20 2014 - [info]  Executing command : apply_diff_relay_logs --command=test --slave_user=mha_mon --slave_host=SLAVE2.COM
--slave_ip=192.168.186.146 --slave_port=3306 --workdir=/var/tmp --target_version=5.6.10-log --manager_version=0.53
--relay_log_info=/data/mysql/data/relay-log.info  --relay_dir=/data/mysql/data/  --slave_pass=xxx
Wed Jul  9 04:23:20 2014 - [info]  Connecting to root@192.168.186.146(SLAVE2.COM:22)..
  Checking slave recovery environment settings..
    Opening /data/mysql/data/relay-log.info ... ok.
    Relay log found at /data/mysql/data, up to slave2-relay-bin.000002
    Temporary relay log file is /data/mysql/data/slave2-relay-bin.000002
    Testing mysql connection and privileges..Warning: Using a password on the command line interface can be insecure.
 done.
    Testing mysqlbinlog output.. done.
    Cleaning up test file(s).. done.
Wed Jul  9 04:23:21 2014 - [info] Slaves settings check done.
Wed Jul  9 04:23:21 2014 - [info]
192.168.186.141 (current master)
 +--192.168.186.142
 +--SLAVE2.COM

Wed Jul  9 04:23:21 2014 - [info] Checking replication health on 192.168.186.142..
Wed Jul  9 04:23:21 2014 - [info]  ok.
Wed Jul  9 04:23:21 2014 - [info] Checking replication health on SLAVE2.COM..
```

Wed Jul 9 04:23:21 2014 - [info] ok.
Wed Jul 9 04:23:21 2014 - [warning] master_ip_failover_script is not defined.
Wed Jul 9 04:23:21 2014 - [warning] shutdown_script is not defined.
Wed Jul 9 04:23:21 2014 - [info] Got exit code 0 (Not master dead).

MySQL Replication Health is OK.

至此说明你的 MHA 已经配置好了

6.启动

[root@MANAGER ~]# nohup masterha_manager --conf=/etc/mastermha/app1.cnf > /tmp/mha_manager.log </dev/null 2>&1 & 启动 MHA

四．测试重构

1.测试

测试 将 MYSQL.COM 机器上的 MYSQL 服务关闭 ，注意观察 manager.log 日志会发现 切换到了 SLAVE1.COM 并且 SLAVE1.COM 变成了主 而 SLAVE2.COM 则变成了 SLAVE1.COM 的从

[root@MANAGER app1]# tail -f manager.log 这是启动后还没关闭主数据库的日志内容
192.168.186.141 (current master)
 +--192.168.186.142
 +--SLAVE2.COM

Wed Jul 9 18:52:32 2014 - [warning] master_ip_failover_script is not defined.
Wed Jul 9 18:52:32 2014 - [warning] shutdown_script is not defined.
Wed Jul 9 18:52:32 2014 - [info] Set master ping interval 1 seconds.
Wed Jul 9 18:52:32 2014 - [warning] secondary_check_script is not defined. It is highly recommended setting it to check master reachability from two or more routes.
Wed Jul 9 18:52:32 2014 - [info] Starting ping health check on 192.168.186.141(192.168.186.141:3306)..
Wed Jul 9 18:52:32 2014 - [info] Ping(SELECT) succeeded, waiting until MySQL doesn't respond..

[root@MYSQL ~]# service mysqld stop
Shutting down MySQL..... SUCCESS!

[root@MANAGER app1]# tail -f manager.log　　最要看最后几行 就知道有没有切换成功
192.168.186.141 (current master)
 +--192.168.186.142
 +--SLAVE2.COM
Wed Jul　9 18:56:47 2014 - [info] Dead Servers:
Wed Jul　9 18:56:47 2014 - [info]　　192.168.186.141(192.168.186.141:3306)
Wed Jul　9 18:56:47 2014 - [info] Alive Servers:
Wed Jul　9 18:56:47 2014 - [info]　　192.168.186.142(192.168.186.142:3306)
Wed Jul　9 18:56:47 2014 - [info]　　SLAVE2.COM(192.168.186.146:3306)
Wed Jul　9 18:56:47 2014 - [info] Alive Slaves:
Wed Jul　9 18:56:47 2014 - [info]　　192.168.186.142(192.168.186.142:3306)　Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul　9 18:56:47 2014 - [info]　　Replicating from 192.168.186.141(192.168.186.141:3306)
Wed Jul　9 18:56:47 2014 - [info]　　Primary candidate for the new Master (candidate_master is set)
Wed Jul　9 18:56:47 2014 - [info]　　SLAVE2.COM(192.168.186.146:3306)　Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul　9 18:56:47 2014 - [info]　　Replicating from 192.168.186.141(192.168.186.141:3306)
Wed Jul　9 18:56:47 2014 - [info]　　Not candidate for the new Master (no_master is set)
Wed Jul　9 18:56:47 2014 - [info] Checking slave configurations..
Wed Jul　9 18:56:47 2014 - [info] Checking replication filtering settings..
Wed Jul　9 18:56:47 2014 - [info]　Replication filtering check ok.
Wed Jul　9 18:56:47 2014 - [info] Master is down!

Wed Jul 9 18:56:47 2014 - [info] Terminating monitoring script.
Wed Jul 9 18:56:47 2014 - [info] Got exit code 20 (Master dead).
Wed Jul 9 18:56:47 2014 - [info] MHA::MasterFailover version 0.53.
Wed Jul 9 18:56:47 2014 - [info] Starting master failover.
Wed Jul 9 18:56:47 2014 - [info] * Phase 2: Dead Master Shutdown Phase completed.
Wed Jul 9 18:56:47 2014 - [info] 192.168.186.142(192.168.186.142:3306) Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul 9 18:56:47 2014 - [info] Replicating from 192.168.186.141(192.168.186.141:3306)
Wed Jul 9 18:56:47 2014 - [info] Primary candidate for the new Master (candidate_master is set)
Wed Jul 9 18:56:47 2014 - [info] SLAVE2.COM(192.168.186.146:3306) Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul 9 18:56:47 2014 - [info] Replicating from 192.168.186.141(192.168.186.141:3306)
Wed Jul 9 18:56:47 2014 - [info] Not candidate for the new Master (no_master is set)
Wed Jul 9 18:56:47 2014 - [info] The oldest binary log file/position on all slaves is mysql-bin.000001:214
Wed Jul 9 18:56:47 2014 - [info] Oldest slaves:
Wed Jul 9 18:56:47 2014 - [info] 192.168.186.142(192.168.186.142:3306) Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul 9 18:56:47 2014 - [info] Replicating from 192.168.186.141(192.168.186.141:3306)
Wed Jul 9 18:56:47 2014 - [info] Primary candidate for the new Master (candidate_master is set)
Wed Jul 9 18:56:47 2014 - [info] SLAVE2.COM(192.168.186.146:3306) Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul 9 18:56:47 2014 - [info] Replicating from 192.168.186.141(192.168.186.141:3306)
Wed Jul 9 18:56:47 2014 - [info] Not candidate for the new Master (no_master is set)
Wed Jul 9 18:56:47 2014 - [info]
Wed Jul 9 18:56:47 2014 - [info] * Phase 3.2: Saving Dead Master's Binlog Phase..
Wed Jul 9 18:56:47 2014 - [info]

Wed Jul 9 18:56:48 2014 - [info] Fetching dead master's binary logs..
Wed Jul 9 18:56:48 2014 - [info] Executing command on the dead master 192.168.186.141(192.168.186.141:3306): save_binary_logs --command=save --start_file=mysql-bin.000001 --start_pos=214 --binlog_dir=/data/mysql/data --output_file=/var/tmp/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog --handle_raw_binlog=1 --disable_log_bin=0 --manager_version=0.53
Creating /var/tmp if not exists.. ok.
Concat binary/relay logs from mysql-bin.000001 pos 214 to mysql-bin.000001 EOF into /var/tmp/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog ..
Dumping binlog format description event, from position 0 to 120.. ok.
Dumping effective binlog data from /data/mysql/data/mysql-bin.000001 position 214 to tail(237).. ok.
Concat succeeded.
Wed Jul 9 18:56:48 2014 - [info] scp from root@192.168.186.141:/var/tmp/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog to local:/masterha/app1/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog succeeded.
Wed Jul 9 18:56:49 2014 - [info] HealthCheck: SSH to 192.168.186.142 is reachable.
Wed Jul 9 18:56:49 2014 - [info] HealthCheck: SSH to SLAVE2.COM is reachable.
Wed Jul 9 18:56:49 2014 - [info]
Wed Jul 9 18:56:49 2014 - [info] * Phase 3.3: Determining New Master Phase..
Wed Jul 9 18:56:49 2014 - [info]
Wed Jul 9 18:56:49 2014 - [info] Finding the latest slave that has all relay logs for recovering other slaves..
Wed Jul 9 18:56:49 2014 - [info] All slaves received relay logs to the same position. No need to resync each other.
Wed Jul 9 18:56:49 2014 - [info] Searching new master from slaves..
Wed Jul 9 18:56:49 2014 - [info] Candidate masters from the configuration file:
Wed Jul 9 18:56:49 2014 - [info] 192.168.186.142(192.168.186.142:3306) Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul 9 18:56:49 2014 - [info] Replicating from 192.168.186.141(192.168.186.141:3306)

Wed Jul 9 18:56:49 2014 - [info] Primary candidate for the new Master (candidate_master is set)
Wed Jul 9 18:56:49 2014 - [info] Non-candidate masters:
Wed Jul 9 18:56:49 2014 - [info] SLAVE2.COM(192.168.186.146:3306) Version=5.6.10-log (oldest major version between slaves) log-bin:enabled
Wed Jul 9 18:56:49 2014 - [info] Replicating from 192.168.186.141(192.168.186.141:3306)
Wed Jul 9 18:56:49 2014 - [info] Not candidate for the new Master (no_master is set)
Wed Jul 9 18:56:49 2014 - [info] Searching from candidate_master slaves which have received the latest relay log events..
Wed Jul 9 18:56:49 2014 - [info] New master is 192.168.186.142(192.168.186.142:3306)
Wed Jul 9 18:56:49 2014 - [info] Starting master failover..
Wed Jul 9 18:56:49 2014 - [info]
From:
192.168.186.141 (current master)
+--192.168.186.142
+--SLAVE2.COM

To:
192.168.186.142 (new master)
+--SLAVE2.COM
Wed Jul 9 18:56:49 2014 - [info]
Wed Jul 9 18:56:49 2014 - [info] * Phase 3.3: New Master Diff Log Generation Phase..
Wed Jul 9 18:56:49 2014 - [info]
Wed Jul 9 18:56:49 2014 - [info] This server has all relay logs. No need to generate diff files from the latest slave.
Wed Jul 9 18:56:49 2014 - [info] Sending binlog..
Wed Jul 9 18:56:50 2014 - [info] scp from local:/masterha/app1/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog to root@192.168.186.142:/var/tmp/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog succeeded.
Wed Jul 9 18:56:50 2014 - [info]

Wed Jul 9 18:56:50 2014 - [info] * Phase 3.4: Master Log Apply Phase..
Wed Jul 9 18:56:50 2014 - [info]
Wed Jul 9 18:56:50 2014 - [info] *NOTICE: If any error happens from this phase, manual recovery is needed.
Wed Jul 9 18:56:50 2014 - [info] Starting recovery on 192.168.186.142(192.168.186.142:3306)..
Wed Jul 9 18:56:50 2014 - [info] Generating diffs succeeded.
Wed Jul 9 18:56:50 2014 - [info] Waiting until all relay logs are applied.
Wed Jul 9 18:56:50 2014 - [info] done.
Wed Jul 9 18:56:50 2014 - [info] Getting slave status..
Wed Jul 9 18:56:50 2014 - [info] This slave(192.168.186.142)'s Exec_Master_Log_Pos equals to Read_Master_Log_Pos(mysql-bin.000001:214). No need to recover from Exec_Master_Log_Pos.
Wed Jul 9 18:56:50 2014 - [info] Connecting to the target slave host 192.168.186.142, running recover script..
Wed Jul 9 18:56:50 2014 - [info] Executing command: apply_diff_relay_logs --command=apply --slave_user=mha_mon --slave_host=192.168.186.142 --slave_ip=192.168.186.142 --slave_port=3306 --apply_files=/var/tmp/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog --workdir=/var/tmp --target_version=5.6.10-log --timestamp=20140709185647 --handle_raw_binlog=1 --disable_log_bin=0 --manager_version=0.53 --slave_pass=xxx
Wed Jul 9 18:56:50 2014 - [info]
MySQL client version is 5.6.10. Using --binary-mode.
Applying differential binary/relay log files /var/tmp/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog on 192.168.186.142:3306. This may take long time...
Applying log files succeeded.
Wed Jul 9 18:56:50 2014 - [info] All relay logs were successfully applied.
Wed Jul 9 18:56:50 2014 - [info] Getting new master's binlog name and position..
Wed Jul 9 18:56:50 2014 - [info] mysql-bin.000007:504
Wed Jul 9 18:56:50 2014 - [info] All other slaves should start replication from here. Statement should be: CHANGE MASTER TO MASTER_HOST='192.168.186.142', MASTER_PORT=3306, MASTER_LOG_FILE='mysql-bin.000007', MASTER_LOG_POS=504, MASTER_USER='repl', MASTER_PASSWORD='xxx';

Wed Jul 9 18:56:50 2014 - [warning] master_ip_failover_script is not set. Skipping taking over new master ip address.
Wed Jul 9 18:56:50 2014 - [info] Setting read_only=0 on 192.168.186.142(192.168.186.142:3306)..
Wed Jul 9 18:56:50 2014 - [info] ok.
Wed Jul 9 18:56:50 2014 - [info] ** Finished master recovery successfully.
Wed Jul 9 18:56:50 2014 - [info] * Phase 3: Master Recovery Phase completed.
Wed Jul 9 18:56:50 2014 - [info]
Wed Jul 9 18:56:50 2014 - [info] * Phase 4: Slaves Recovery Phase..
Wed Jul 9 18:56:50 2014 - [info]
Wed Jul 9 18:56:50 2014 - [info] * Phase 4.1: Starting Parallel Slave Diff Log Generation Phase..
Wed Jul 9 18:56:50 2014 - [info]
Wed Jul 9 18:56:50 2014 - [info] -- Slave diff file generation on host SLAVE2.COM(192.168.186.146:3306) started, pid: 3135. Check tmp log /masterha/app1/SLAVE2.COM_3306_20140709185647.log if it takes time..
Wed Jul 9 18:56:50 2014 - [info]
Wed Jul 9 18:56:50 2014 - [info] Log messages from SLAVE2.COM ...
Wed Jul 9 18:56:50 2014 - [info]
Wed Jul 9 18:56:50 2014 - [info] This server has all relay logs. No need to generate diff files from the latest slave.
Wed Jul 9 18:56:50 2014 - [info] End of log messages from SLAVE2.COM.
Wed Jul 9 18:56:50 2014 - [info] -- SLAVE2.COM(192.168.186.146:3306) has the latest relay log events.
Wed Jul 9 18:56:50 2014 - [info] Generating relay diff files from the latest slave succeeded.
Wed Jul 9 18:56:50 2014 - [info]
Wed Jul 9 18:56:50 2014 - [info] * Phase 4.2: Starting Parallel Slave Log Apply Phase..
Wed Jul 9 18:56:50 2014 - [info]
Wed Jul 9 18:56:50 2014 - [info] -- Slave recovery on host SLAVE2.COM(192.168.186.146:3306) started, pid: 3137. Check tmp log /masterha/app1/SLAVE2.COM_3306_20140709185647.log if it takes time..
Wed Jul 9 18:56:51 2014 - [info]
Wed Jul 9 18:56:51 2014 - [info] Log messages from SLAVE2.COM ...

Wed Jul 9 18:56:51 2014 - [info]
Wed Jul 9 18:56:50 2014 - [info] Sending binlog..
Wed Jul 9 18:56:51 2014 - [info] scp from local:/masterha/app1/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog to root@SLAVE2.COM:/var/tmp/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog succeeded.
Wed Jul 9 18:56:51 2014 - [info] Starting recovery on SLAVE2.COM(192.168.186.146:3306)..
Wed Jul 9 18:56:51 2014 - [info] Generating diffs succeeded.
Wed Jul 9 18:56:51 2014 - [info] Waiting until all relay logs are applied.
Wed Jul 9 18:56:51 2014 - [info] done.
Wed Jul 9 18:56:51 2014 - [info] Getting slave status..
Wed Jul 9 18:56:51 2014 - [info] This slave(SLAVE2.COM)'s Exec_Master_Log_Pos equals to Read_Master_Log_Pos(mysql-bin.000001:214). No need to recover from Exec_Master_Log_Pos.
Wed Jul 9 18:56:51 2014 - [info] Connecting to the target slave host SLAVE2.COM, running recover script..
Wed Jul 9 18:56:51 2014 - [info] Executing command: apply_diff_relay_logs --command=apply --slave_user=mha_mon --slave_host=SLAVE2.COM --slave_ip=192.168.186.146 --slave_port=3306 --apply_files=/var/tmp/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog --workdir=/var/tmp --target_version=5.6.10-log --timestamp=20140709185647 --handle_raw_binlog=1 --disable_log_bin=0 --manager_version=0.53 --slave_pass=xxx
Wed Jul 9 18:56:51 2014 - [info]
MySQL client version is 5.6.10. Using --binary-mode.
Applying differential binary/relay log files /var/tmp/saved_master_binlog_from_192.168.186.141_3306_20140709185647.binlog on SLAVE2.COM:3306. This may take long time...
Applying log files succeeded.
Wed Jul 9 18:56:51 2014 - [info] All relay logs were successfully applied.
Wed Jul 9 18:56:51 2014 - [info] Resetting slave SLAVE2.COM(192.168.186.146:3306) and starting replication from the new master 192.168.186.142(192.168.186.142:3306)..
Wed Jul 9 18:56:51 2014 - [info] Executed CHANGE MASTER.
Wed Jul 9 18:56:51 2014 - [info] Slave started.

Wed Jul 9 18:56:51 2014 - [info] End of log messages from SLAVE2.COM.
Wed Jul 9 18:56:51 2014 - [info] -- Slave recovery on host SLAVE2.COM(192.168.186.146:3306) succeeded.
Wed Jul 9 18:56:51 2014 - [info] All new slave servers recovered successfully.
Wed Jul 9 18:56:51 2014 - [info]
Wed Jul 9 18:56:51 2014 - [info] * Phase 5: New master cleanup phease..
Wed Jul 9 18:56:51 2014 - [info]
Wed Jul 9 18:56:51 2014 - [info] Resetting slave info on the new master..
Wed Jul 9 18:56:51 2014 - [info] 192.168.186.142: Resetting slave info succeeded.
Wed Jul 9 18:56:51 2014 - [info] Master failover to 192.168.186.142(192.168.186.142:3306) completed successfully.
Wed Jul 9 18:56:51 2014 - [info]

----- Failover Report -----

app1: MySQL Master failover 192.168.186.141 to 192.168.186.142 succeeded

Master 192.168.186.141 is down!

Check MHA Manager logs at MANAGER.COM:/masterha/app1/manager.log for details.

Started automated(non-interactive) failover.
The latest slave 192.168.186.142(192.168.186.142:3306) has all relay logs for recovery.
Selected 192.168.186.142 as a new master.
192.168.186.142: OK: Applying all logs succeeded.
SLAVE2.COM: This host has the latest relay log events.
Generating relay diff files from the latest slave succeeded.
SLAVE2.COM: OK: Applying all logs succeeded. Slave started, replicating from 192.168.186.142.

192.168.186.142: Resetting slave info succeeded.
Master failover to 192.168.186.142(192.168.186.142:3306) completed successfully. 说明切换成功了

[root@slave2 ~]# mysql -e "show slave status\G"
*************************** 1. row ***************************
Slave_IO_State: Waiting for master to send event
Master_Host: 192.168.186.142
Master_User: repl
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000007
Read_Master_Log_Pos: 504
Relay_Log_File: slave2-relay-bin.000002
Relay_Log_Pos: 283
Relay_Master_Log_File: mysql-bin.000007
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
看到已经切换到和 142 同步了 本来是和 141 同步的 此时的 SALVE1.COM 已经变成主了 说明已然生效

2.重构

重构我想就不要我做了吧，就是这时候等于你的主挂了 切换在 SLAVE1.COM 上变成了主 因此重构我提供一种方案（方案多种），拿出一台新的服务器重新加入以 142 为主做成从 再 app1.conf 配置文件中加入

[root@MANAGER app1]# rm -rf app1.failover.complete 删除该文件后 再次启动 manager 端即可

五．扩展 Keepalived

1.安装 keepalived

[root@MYSQL src]# wget http://www.keepalived.org/software/keepalived-1.2.12.tar.gz
[root@MYSQL src]# tar -xf keepalived-1.2.12.tar.gz
[root@MYSQL src]# cd keepalived-1.2.12
[root@MYSQL src]# yum -y install gcc gcc-c++ gcc-g77 ncurses-devel bison libaio-devel cmake libnl* libpopt* popt-static openssl-devel
[root@MYSQL keepalived-1.2.12]# ./configure
[root@MYSQL keepalived-1.2.12]# make && make install
[root@MYSQL src]#mkdir /etc/keepalived/

[root@MYSQL src]# cp /usr/local/etc/keepalived/keepalived.conf /etc/keepalived/
[root@MYSQL src]# cp /usr/local/etc/rc.d/init.d/keepalived /etc/init.d/
[root@MYSQL src]# cp /usr/local/etc/sysconfig/keepalived /etc/sysconfig/
[root@MYSQL src]#cp /usr/local/sbin/keepalived /usr/sbin/

2.配置 keepalived

```
[root@MYSQL keepalived]# vi keepalived.conf
! Configuration File for keepalived

global_defs {
   notification_email {
     acassen@firewall.loc
     failover@firewall.loc
     sysadmin@firewall.loc
   }
   notification_email_from Alexandre.Cassen@firewall.loc
   smtp_server 192.168.200.1
   smtp_connect_timeout 30
   router_id LVS_DEVEL              ##配置是为了标识当前节点，两个节点的此项设置可相同，也可不相同
}

vrrp_instance VI_1 {
    state MASTER                    #指定 A 节点为主节点 备用节点上设置为 BACKUP 即可
    interface eth0
    virtual_router_id 51              #VRRP 组名，两个节点的设置必须一样，以指明各个节点属于同一 VRRP
          priority 100        #主节点的优先级（1-254之间），备用节点必须比主节点优先级低
       advert_int 1
          authentication {            #设置验证信息，两个节点必须一致
       auth_type PASS
```

```
        auth_pass 1111
    }
    virtual_ipaddress {
    192.168.200.16
        }
    }
```

3.检测 mysql 服务脚本

```
[root@MYSQL keepalived]# vi /root/check_mysql.sh
#!/bin/bash
MYSQL=/usr/local/mysql/bin/mysql
MYSQL_HOST=127.0.0.1
MYSQL_USER=root
MYSQL_PASSWORD=123
CHECK_TIME=3
#mysql   is working MYSQL_OK is 1 , mysql down MYSQL_OK is 0
MYSQL_OK=1
function check_mysql_helth (){
$MYSQL -h $MYSQL_HOST -u $MYSQL_USER -e "show status;" >/dev/null 2>&1
if [ $? = 0 ] ;then
      MYSQL_OK=1
```

```
else
        MYSQL_OK=0
fi
        return $MYSQL_OK
}
while [ $CHECK_TIME -ne 0 ]
do
        let "CHECK_TIME -= 1"
        check_mysql_helth
if [ $MYSQL_OK = 1 ] ; then
        CHECK_TIME=0
        exit 0
fi
if [ $MYSQL_OK -eq 0 ] &&   [ $CHECK_TIME -eq 0 ]
then
        pkill keepalived
exit 1
fi
sleep 1
Done
```

该脚本做一个计划任务每分钟做一次检查 MYSQL 服务如果挂掉了的话 VIP 就会在 SALVE1.COM 上生效 这样的话 client 端连接的 VIP 就会从 MYSQL.COM 上变到了 SLAVE1.COM 上 此时 SLAVE1.COM 由于 MHA 的生效 已经变为主对外提供服务了 VIP 也在 SALVE1.COM 上 所以从头到尾 客户端只需要连接 VIP 就可以了

```
[root@MYSQL keepalived]# yum -y install cronie
[root@MYSQL ~]# crontab   -l
*/1 * * * * bash /root/checkmysql.sh
```

以上步骤再在下一次切换的 SLAVE 上做一次也就是 SLAVE1.COM 上做一次 keepalived.conf 配置文件不同的地方已做注释

4.启动测试

两边启动 keepalived

```
[root@MYSQL keepalived]# keepalived -f /etc/keepalived/keepalived.conf
[root@MYSQL keepalived]# ps -ef |grep keep
root      3230     1  0 23:27 ?        00:00:00 keepalived -f /etc/keepalived/keepalived.conf
root      3231  3230  0 23:27 ?        00:00:00 keepalived -f /etc/keepalived/keepalived.conf
root      3232  3230  0 23:27 ?        00:00:00 keepalived -f /etc/keepalived/keepalived.conf
root      3234  2538  0 23:27 pts/0    00:00:00 grep keep

[root@MYSQL keepalived]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
```

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:c9:85:ba brd ff:ff:ff:ff:ff:ff
    inet 192.168.186.141/24 brd 192.168.186.255 scope global eth0
    inet 192.168.200.16/32 scope global eth0
    inet6 fe80::20c:29ff:fec9:85ba/64 scope link
       valid_lft forever preferred_lft forever

测试关闭该主机的 MYSQL 服务看看 VIP 是否飘到配好 keepalive 的 SLAVE1.COM 上 MHA 中的 MYSQL 主是不是也变成了 SLAVE1.COM 如是 至此全部完成。。 谢谢!!

keepalived.conf checkmysql.sh

mha 日常维护命令

1.查看 ssh 登陆是否成功

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

2.查看复制是否建立好

```
masterha_check_repl --conf=/etc/masterha/app1.cnf
```

3.启动 mha

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

4.检查启动的状态

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

5.停止 mha

```
masterha_stop  --conf=/etc/masterha/app1.cnf
```

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 --conf=/etc/masterha/app1.cnf --dead_master_host=10.50.2.10 --master_state=dead --new_master_host=10.50.2.12 --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 --conf=/etc/app1.cnf --master_state=alive --new_master_host=192.168.119.74 --orig_master_is_new_slave
或者
masterha_master_switch --conf=/etc/app1.cnf --master_state=alive --new_master_host=192.168.119.74 --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。

可以通过如下命令停止 mha

```
masterha_stop --conf=/etc/app1.cnf
```