

第1章 MHA介绍

1.MHA介绍

- 1 MHA(Master High Availability)目前在MySQL高可用方面是一个相对成熟的解决方案。
- 2 它由日本DeNA公司youshimaton(现就职于Facebook公司)开发，是一套优秀的作为MySQL高可用性环境下故障切换和主从提升的高可用软件。
- 3
- 4 在MySQL故障切换过程中，MHA能做到在0~30秒之内自动完成数据库的故障切换操作，并且在进行故障切换的过程中，MHA能在最大程度上保证数据的一致性，以达到真正意义上的高可用。
- 5
- 6 MHA还提供在线主库切换的功能，能够安全地切换当前运行的主库到一个新的主库中（通过将主库提升为主库），大概0.5-2秒内即可完成。

2.MHA的优势

- 1 1.自动故障转移快
- 2 2.主库崩溃不存在数据一致性问题
- 3 3.配置不需要对当前mysql环境做重大修改
- 4 4.不需要添加额外的服务器(仅一台manager就可管理上百个replication)
- 5 5.性能优秀，可工作在半同步复制和异步复制，当监控mysql状态时，仅需要每隔N秒向master发送ping包(默认3秒)，所以对性能无影响。你可以理解为MHA的性能和简单的主从复制框架性能一样。
- 6 6.只要replication支持的存储引擎，MHA都支持，不会局限于innodb

3.MHA功能

- 1 1.监控
- 2 2.选主
- 3 3.应用透明(vip)
- 4 4.故障提醒
- 5 5.额外数据补偿
- 6 6.剔除故障节点
- 7 7.manager 程序"自杀"

4.MHA组件功能说明

- | | | |
|----|-------------------------|---------------------------|
| 1 | manager 组件 | |
| 2 | masterha_manger | 启动MHA |
| 3 | masterha_check_ssh | 检查MHA的SSH配置状况 |
| 4 | masterha_check_repl | 检查MySQL复制状况，配置信息 |
| 5 | masterha_master_monitor | 检测master是否宕机 |
| 6 | masterha_check_status | 检测当前MHA运行状态 |
| 7 | masterha_master_switch | 控制故障转移（自动或者手动） |
| 8 | masterha_conf_host | 添加或删除配置的server信息 |
| 9 | | |
| 10 | node 组件 | |
| 11 | save_binary_logs | 保存和复制master的二进制日志 |
| 12 | apply_diff_relay_logs | 识别差异的中继日志事件并将其差异的事件应用于其他的 |
| 13 | purge_relay_logs | 清除中继日志（不会阻塞SQL线程） |

第2章 基础主从复制环境搭建部署

0.部署环境说明

1 | 三个MySQL节点，GTID，主从复制

1.mysql环境

清除以前的环境:

```
1  pkill mysqld
2  rm -rf /data/
3
4  mkdir -p /data/mysql_3306/
5  mkdir -p /mysql_binlog/
6  chown -R mysql:mysql /opt/mysql*
7  chown -R mysql:mysql /data/
8  chown -R mysql:mysql /mysql_binlog/
9
10 cat > /etc/my.cnf <<EOF
11 [mysqld]
12 user=mysql
13 datadir=/data/mysql_3306
14 basedir=/opt/mysql/
15 socket=/tmp/mysql.sock
16 port=3306
17 server_id=51
18 log_bin=/mysql_binlog/mysql-bin
19 autocommit=0
20 binlog_format=row
21 gtid-mode=on
22 enforce-gtid-consistency=true
23 log-slave-updates=1
24
25 [mysql]
26 socket=/tmp/mysql.sock
27
28 [client]
29 socket=/tmp/mysql.sock
30 EOF
31
32 mysqld --initialize-insecure --user=mysql --basedir=/opt/mysql --
datadir=/data/mysql_3306/
33 systemctl stop mysqld
34 systemctl start mysqld
35 netstat -lntup|grep 3306
36 mysqladmin password '123'
```

操作命令:

```
1  mkdir -p /data/mysql_3306/
2  mkdir -p /mysql_binlog/
3
4  yum remove mariadb-libs -y
```

```
5 rm -rf /etc/my.cnf
6 yum install -y libaio-devel
7
8 tar xzf mysql-5.7.28-linux-glibc2.12-x86_64.tar.gz -C /opt/
9 mv /opt/mysql-5.7.28-linux-glibc2.12-x86_64 /opt/mysql-5.7.28
10 ln -s /opt/mysql-5.7.28 /opt/mysql
11 echo 'export PATH=$PATH:/opt/mysql/bin' >>/etc/profile
12 source /etc/profile
13 mysql -V
14
15 useradd -s /sbin/nologin -M mysql
16
17 chown -R mysql:mysql /opt/mysql*
18 chown -R mysql:mysql /data/
19 chown -R mysql:mysql /mysql_binlog/
20
21 #db-51的配置
22 cat > /etc/my.cnf <<EOF
23 [mysqld]
24 user=mysql
25 datadir=/data/mysql_3306
26 basedir=/opt/mysql/
27 socket=/tmp/mysql.sock
28 port=3306
29 server_id=51
30 log_bin=/mysql_binlog/mysql-bin
31 autocommit=0
32 binlog_format=row
33 gtid-mode=on
34 enforce-gtid-consistency=true
35 log-slave-updates=1
36
37 [mysql]
38 socket=/tmp/mysql.sock
39
40 [client]
41 socket=/tmp/mysql.sock
42 EOF
43
44 #db-52的配置
45 cat > /etc/my.cnf <<EOF
46 [mysqld]
47 user=mysql
48 datadir=/data/mysql_3306
49 basedir=/opt/mysql/
50 socket=/tmp/mysql.sock
51 port=3306
52 server_id=52
53 log_bin=/mysql_binlog/mysql-bin
54 autocommit=0
55 binlog_format=row
56 gtid-mode=on
57 enforce-gtid-consistency=true
58 log-slave-updates=1
59
60 [mysql]
61 socket=/tmp/mysql.sock
62
```

```

63 [client]
64 socket=/tmp/mysql.sock
65 EOF
66
67 #db-53的配置
68 cat > /etc/my.cnf <<EOF
69 [mysqld]
70 user=mysql
71 datadir=/data/mysql_3306
72 basedir=/opt/mysql/
73 socket=/tmp/mysql.sock
74 port=3306
75 server_id=53
76 log_bin=/mysql_binlog/mysql-bin
77 autocommit=0
78 binlog_format=row
79 gtid-mode=on
80 enforce-gtid-consistency=true
81 log-slave-updates=1
82
83 [mysql]
84 socket=/tmp/mysql.sock
85
86 [client]
87 socket=/tmp/mysql.sock
88 EOF
89
90 mysqld --initialize-insecure --user=mysql --basedir=/opt/mysql --
datadir=/data/mysql_3306/
91
92 cp /opt/mysql/support-files/mysql.server /etc/init.d/mysqld
93 chkconfig --add mysqld
94 systemctl start mysqld
95 netstat -lntup|grep 3306
96 mysqladmin password '123'

```

2.主从复制环境

```

1 #db-51创建复制用户
2 mysql -uroot -p123
3 grant replication slave on *.* to repl@'10.0.0.%' identified by '123';
4
5 #db-52和db-53配置复制信息
6 mysql -uroot -p123
7 change master to
8 master_host='10.0.0.51',
9 master_user='repl',
10 master_password='123' ,
11 MASTER_AUTO_POSITION=1;
12 start slave;
13
14 #检查复制状态
15 show slave status\G

```

第3章 MHA服务搭建部署

1.创建软连接-所有机器都操作

```
1 ln -s /opt/mysql/bin/mysqlbinlog /usr/bin/mysqlbinlog
2 ln -s /opt/mysql/bin/mysql /usr/bin/mysql
```

2.节点互信-所有机器都操作

```
1 ssh-keygen
2 ssh-copy-id 10.0.0.51
3 ssh-copy-id 10.0.0.52
4 ssh-copy-id 10.0.0.53
```

3.各节点验证-所有机器都操作

```
1 ssh 10.0.0.51 hostname
2 ssh 10.0.0.52 hostname
3 ssh 10.0.0.53 hostname
```

4.所有节点安装node软件

```
1 yum install perl-DBD-MySQL -y
2 rpm -ivh mha4mysql-node-0.58-0.el7.centos.noarch.rpm
```

5.db-53 安装Manager软件

```
1 yum install -y perl-Config-Tiny epel-release perl-Log-Dispatch perl-Parallel-
  ForkManager perl-Time-HiRes
2 yum install -y mha4mysql-manager-0.58-0.el7.centos.noarch.rpm
```

6.db-51 主库中创建mha需要的用户

```
1 grant all privileges on *.* to mha@'10.0.0.%' identified by 'mha';
2 select user,host from mysql.user;
```

7.db-53 准备Manager配置文件

```
1 #创建配置文件目录
2 mkdir -p /etc/mha
3
4 #创建日志目录
5 mkdir -p /var/log/mha/app1
6
7 #编辑mha配置文件
8 cat > /etc/mha/app1.cnf << EOF
9 [server default]
10 manager_log=/var/log/mha/app1/manager
11 manager_workdir=/var/log/mha/app1
12 master_binlog_dir=/mysql_binlog/
13 user=mha
14 password=mha
```

```

15 ping_interval=2
16 repl_password=123
17 repl_user=repl
18 ssh_user=root
19 [server1]
20 hostname=10.0.0.51
21 port=3306
22 [server2]
23 hostname=10.0.0.52
24 port=3306
25 [server3]
26 hostname=10.0.0.53
27 port=3306
28 EOF
29
30 #####配置文件解释-不要复制,k
31 #[server default]
32 #manager_log=/var/log/mha/app1/manager          # MHA的工作日志设置
33 #manager_workdir=/var/log/mha/app1              # MHA工作目录
34 #master_binlog_dir=/data/binlog                 # 主库的binlog目录
35 #user=mha                                       # 监控用户

36 #password=mha                                # 监控密码
37 #ping_interval=2                             # 心跳检测的间隔时间
38 #repl_password=123                           # 复制用户
39 #repl_user=repl                               # 复制密码
40 #ssh_user=root                                # ssh互信的用户
41 #[server1]                                    # 节点信息....

```

8.状态检查 (db03)

```

1 masterha_check_ssh --conf=/etc/mha/app1.cnf
2 masterha_check_rep --conf=/etc/mha/app1.cnf

```

9.开启MHA-manager (db03)

```

1 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 &

```

10.查看MHA状态

```

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

```

第4章 MHA工作原理

1.选主策略

- 1 1.日志量最新
- 2 2.备选主
- 3 3.不被选主

2.故障转移流程

1. 从宕机崩溃的master保存二进制日志事件(binlog events);
2. 识别含有最新更新的slave;
3. 应用差异的中继日志(relay log)到其他的slave;
4. 应用从master保存的二进制日志事件(binlog events);
5. 提升一个slave为新的master;
6. 使其他的slave连接新的master进行复制;

第5章 MHA应用透明

1.vip故障转移脚本

- 1 上传脚本文件到/usr/local/bin
- 2 \cp -a * /usr/local/bin

2.修改权限

- 1 chmod +x /usr/local/bin/*

3.替换字符

- 1 dos2unix /usr/local/bin/*

4.修改内容

- 1 vim /usr/local/bin/master_ip_failover
- 2 my \$vip = '10.0.0.55/24';
- 3 my \$key = '1';
- 4 my \$ssh_start_vip = "/sbin/ifconfig eth0:\$key \$vip";
- 5 my \$ssh_stop_vip = "/sbin/ifconfig eth0:\$key down";
- 6 my \$ssh_Bcast_arp= "/sbin/arping -I eth0 -c 3 -A 10.0.0.55";

5.修改配置文件

- 1 vim /etc/mha/app1.cnf
- 2 master_ip_failover_script=/usr/local/bin/master_ip_failover

6.重启mha

- 1 masterha_stop --conf=/etc/mha/app1.cnf
- 2 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 &

7.手工在主库添加VIP

- 1 ifconfig eth0:1 10.0.0.55/24

第6章 MHA故障提醒

1.准备脚本

```
1 vim /usr/local/bin/send_report
2 my $smtp='smtp.qq.com';
3 my $mail_from='526195417@qq.com';
4 my $mail_user='526195417';
5 my $mail_pass='njwygmkbvz1ubiji';
6 my $mail_to='526195417@qq.com';
```

2.修改配置文件

```
1 vim /etc/mha/app1.cnf
2 report_script=/usr/local/bin/send_report
```

3.重启MHA

```
1 masterha_stop --conf=/etc/mha/app1.cnf
2 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 &
```

第7章 MHA数据补偿

1.创建必要目录(db03)

```
1 mkdir -p /data/binlog_server/
2
3 chown -R mysql:mysql /data/*
4 cd /data/binlog_server/
```

2.拉取最新日志

注意:

拉取日志的起点,需要按照目前从库的已经获取到的二进制日志点为起点

```
1 mysql -e "show slave status \G"|grep "Master_Log"
2 mysqlbinlog -R --host=10.0.0.51 --user=mha --password=mha --raw --stop-never
  mysql-bin.000003 &
3 ll /data/binlog_server
```

3.MHA配置文件设置

```
1 vim /etc/mha/app1.cnf
2 [binlog1]
3 no_master=1
4 hostname=10.0.0.53
5 master_binlog_dir=/data/binlog_server/
```

4.重启MHA


```
1 masterha_stop --conf=/etc/mha/app1.cnf
2 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 &
```

第8章 MHA模拟故障演练

1. 停掉主库
2. 查看VIP是否会漂移
3. 查看从库是否升为新的主库
4. 查看从库是否切换为新的复制关系

第9章 MHA修复方案

1. 确定三个数据库节点都在线

```
1 #db-51启动数据库
2 systemctl start mysqld
```

2. db-51修复主从复制关系

```
1 change master to
2 master_host='10.0.0.52',
3 master_user='rep1',
4 master_password='123' ,
5 MASTER_AUTO_POSITION=1;
6 start slave;
7 show slave status\G
```

3. 判断VIP是否存在

```
1 #db-52确认是否存在VIP
2 ip a|grep 10.0.0.55
```

4. 检查binlogserver并修复

```
1 ps -ef|grep mysqlbinlog
2 rm -rf /data/binlog_server/*
3 cd /data/binlog_server/
4 mysql -e "show slave status \G"|grep "Master_Log"
5 mysqlbinlog -R --host=10.0.0.51 --user=mha --password=mha --raw --stop-never
  mysql-bin.000003 &
```

5. 检查配置文件确认三个节点是否存在

```
1 #添加新节点到配置文件
2 masterha_conf_host --command=add --conf=/etc/mha/app1.cnf --hostname=10.0.0.51
  --block=server1 --params="port=3306"
3 -----删除命令-----
4 masterha_conf_host --command=delete --conf=/etc/mha/app1.cnf --block=server1
```

6.检查ssh互信和repl

```
1 masterha_check_ssh --conf=/etc/mha/app1.cnf
2 masterha_check_rep1 --conf=/etc/mha/app1.cnf
```

7.启动MHA

```
1 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 &
```

8.检查MHA

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

第10章 MHA维护

1.master_ip_online_change_script功能

```
1 在线切换时，自动锁原主库，VIP自动切换
```

2.准备切换脚本

```
1 vim /usr/local/bin/master_ip_online_change
2
3 my $vip = "10.0.0.55/24";
4 my $key = "1";
5 my $ssh_start_vip = "/sbin/ifconfig eth0:$key $vip";
6 my $ssh_stop_vip = "/sbin/ifconfig eth0:$key $vip down";
7 my $ssh_Bcast_arp= "/sbin/arping -I eth0 -c 3 -A 10.0.0.55";
```

3.修改MHA配置文件

```
1 vim /etc/mha/app1.cnf
2 master_ip_online_change_script=/usr/local/bin/master_ip_online_change
```

4.停 MHA

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

5.检查repl

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

6.在线切换

```
1 masterha_master_switch --conf=/etc/mha/app1.cnf --master_state=alive --
  new_master_host=10.0.0.51 --orig_master_is_new_slave --
  running_updates_limit=10000
```

7.重构binlogserver

```
1 [root@db-53 ~]# ps -ef |grep [m]ysqlbinlog
2 root      33698  23489  0 21:18 pts/0    00:00:00 mysqlbinlog -R --
  host=10.0.0.52 --user=mha --password=x x --raw --stop-never mysql-bin.000003
3 [root@db-53 ~]# kill 33698
4 [root@db-53 ~]# cd /data/binlog_server/
5 [root@db-53 binlog_server]# rm -rf *
6 [root@db-53 binlog_server]# mysqlbinlog -R --host=10.0.0.51 --user=mha --
  password=mha --raw --stop-never mysql-bin.000003 &
7 [1] 36893
```

8.启动MHA

```
1 [root@db-53 ~]# 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 &
2 [2] 36915
3
4 [root@db-53 ~]# masterha_check_status --conf=/etc/mha/app1.cnf
5 app1 (pid:36915) is running(0:PING_OK), master:10.0.0.51
```