一 mysql-mha集群

１，前言

１）MHA（master hight availability）（至少3台）

由日本DeNA公司youshimaton开发

是一套优秀的实现mysql高可用的解决方案

数据库的自动故障切换操作能做到0～30秒之内

mha能确保在故障切换过程中保证数据的一致性，以达到真正意义上的高可用

２）MHA组成

MHA Manager

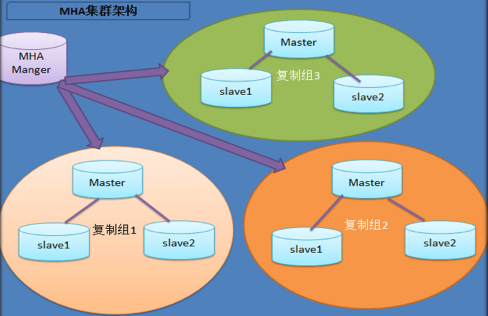
可以单独部署在一台独立的机器上，管理其他节点

也可以部署在一台slave节点上

MHA Node

运行在每台mysql服务器上

３）MHA集群构架



４）MHA工作过程

由manager定时探测集群中的master节点

当master故障时,manager自动将拥有最新数据的slave提升为新的master

关键点：

从宕机崩溃的master保存二进制日志文件事件

识别含有最新更新的slave

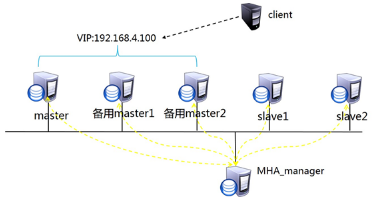
应用差异的中继日志(relay-log)到其他的slave

应用从master保存的二进制日志事件

提升一个slave为新的master

使其他的slave连接新的master进行复制

５）拓扑图



Ip规划：



２，实战

－－51主机配置

1）配置master主节点，数据库服务器配置文件

[root@mysql-51 ~]# vim /etc/my.cnf

[mysqld]

log\_bin=master51

server\_id=51

binlog\_format="mixed"

relay\_log\_purge=off　　　#不自动删除本机的中继日志文件

plugin-load="rpl\_semi\_sync\_master=semisync\_master.so;rpl\_semi\_sync\_slave=semisync\_slave.so"

rpl-semi-sync-master-enabled=1

rpl-semi-sync-slave-enabled=1

[root@mysql-51 ~]# systemctl restart mysqld

Mysql>grant replication slave on \*.\* to repluser@’%’ identified by ‘123456’;

－－52主机配置

１）备用1 master52，数据库服务器配置文件

[root@mysql-52 ~]# vim /etc/my.cnf

[mysqld]

log\_bin=master52

binlog\_format="mixed"

server\_id=52

relay\_log\_purge=off

plugin-load="rpl\_semic\_sync\_master=semicsync\_master.so;rpl\_semi\_sync\_slave=semisync\_slave.so"

rpl-semi-sync-master-enabled=1

rpl-semi-sync-slave-enabled=1

[root@mysql-52 ~]# systemctl restart mysqld

mysql> grant replication slave on \*.\* to repluser@'%' identified by '123456';

mysql> change master to \

-> master\_host='192.168.4.51',

-> master\_user='repluser',

-> master\_password='123456',

-> master\_log\_file='master51.000001',

-> master\_log\_pos=154;

mysql> start slave;

mysql> show slave status\G;

..........

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

－－53主机配置

１）备用2 master53，数据库服务器配置文件

[root@mysql-53 ~]# vim /etc/my.cnf

[mysqld]

log-bin=master53

server\_id=53

binlog-format="mixed"

relay\_log\_purge=off

plugin-load="rpl\_semi\_sync\_master=semisync\_master.so;rpl\_semi\_sync\_slave=semisync\_slave.so"

rpl-semi-sync-master-enabled=1

rpl-semi-sync-slave-enabled=1

[root@mysql-53 ~]# systemctl restart mysqld

mysql> grant replication slave on \*.\* to repluser@'%' identified by '123456';

mysql> change master to

-> master\_host='192.168.4.51'

-> ,master\_user='repluser',

-> master\_password='123456',

-> master\_log\_file='master51.000001',

-> master\_log\_pos=154;

mysql> start slave;

mysql> show slave status\G;

........

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

－－54主机配置

１）从库slave54　数据服务器配置文件

[root@mysql-54 ~]# vim /etc/my.cnf

[mysqld]

server\_id=54

relay\_log\_purge=off

plugin\_load="rpl\_semi\_sync\_slave=semisync\_slave.so"　#加载半同步插件

rpl\_semi\_sync\_slave\_enabled=1　　　#开启半同步插件

[root@mysql-54 ~]# systemctl restart mysqld

mysql> change master to

-> master\_host='192.168.4.51',

-> master\_user='repluser',

-> master\_password='123456',

-> master\_log\_file='master51.000001',

-> master\_log\_pos=154;

mysql> start slave;

mysql> show slave status\G;

......

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

－－55主机配置

１）从库slave55，数据库服务器配置文件

[root@mysql-55 ~]# vim /etc/my.cnf

[mysqld]

#log\_bin=master55

#binlog\_format="mixed"

server\_id=55

relay\_log\_purge=off

plugin\_load="rpl\_semi\_sync\_slave=semisync\_slave.so"

rpl\_semi\_sync\_slave\_enabled=1

[root@mysql-55 ~]# systemctl restart mysqld

mysql> change master to

-> master\_host='192.168.4.51',

-> master\_user='repluser',

-> master\_password='123456',

-> master\_log\_file='master51.000001',

-> master\_log\_pos=154;

mysql> start slave;

mysql> show slave status\G;

．．．．

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

３，简单测试

－－测试一

51主机：

mysql> create database db8;

mysql> create table db8.t1(id int);

mysql> insert into db8.t1 values(51);

mysql> grant select,insert on db8.\* to jack@'%' identified by '123456';

52，53，54，55主机查看

mysql> select \* from db8.t1;　　#看到都一样

+------+

| id |

+------+

| 51 |

+------+

－－测试二

主机client50

[root@mysql-50 ~]# mysql -ujack -p123456 -h192.168.4.51

mysql> insert into db8.t1 values(50);

52，53，54，55主机都看到

mysql> select \* from db8.t1;　　#看到都一样

+------+

| id |

+------+

| 51 |

| 50 |

４，配置集群

－－集群环境准备

1）51-55相互无密码连接(以51为例)

[root@mysql-51 ~]# ssh-keygen -f "/root/.ssh/id\_rsa" -N ""

[root@mysql-51 ~]# for i in 51 52 53 54 55; do ssh-copy-id 192.168.4.$i; done

2）56无密码连接51,52,53,54,55

[root@mysql-56 ~]# ssh-keygen -f "/root/.ssh/id\_rsa" -N ""

[root@mysql-56 ~]# for i in 51 52 53 54 55 56; do ssh-copy-id 192.168.4.$i; done

3）51-56安装perl软件包

yum -y install perl\*

4）安装MHA集群软件

51－56主机安装mha-node，其他perl依赖包（以51为例）

[root@mysql-51 jluo]# yum -y install mha-soft-student/perl\*.rpm　#安装依赖包

[root@mysql-51 jluo]# rpm -ivh mha-soft-student/mha4mysql-node-0.56-0.el6.noarch.rpm #安装mha-node

56主机安装mha-manage

[root@mysql-56 jluo]# tar -xf mha-soft-student/mha4mysql-manager-0.56.tar.gz

[root@mysql-56 jluo]# cd mha4mysql-manager-0.56/

[root@mysql-56 mha4mysql-manager-0.56]# perl Makefile.PL

[root@mysql-56 mha4mysql-manager-0.56]# make

[root@mysql-56 mha4mysql-manager-0.56]# make install

－－配置集群

1）查看master管理命令是否正确

[root@mysql-56 mha4mysql-manager-0.56]# ls /usr/local/bin/master\*

/usr/local/bin/masterha\_check\_repl /usr/local/bin/masterha\_master\_monitor

/usr/local/bin/masterha\_check\_ssh /usr/local/bin/masterha\_master\_switch

/usr/local/bin/masterha\_check\_status /usr/local/bin/masterha\_secondary\_check

/usr/local/bin/masterha\_conf\_host /usr/local/bin/masterha\_stop

/usr/local/bin/masterha\_manager

２）配置管理节点主机配置文件

[root@mysql-56 mha4mysql-manager-0.56]# mkdir /etc/mha　#新建目录

[root@mysql-56 mha4mysql-manager-0.56]# cp samples/conf/app1.cnf /etc/mha/　　　　　　#拷贝配置文件过去

[root@mysql-56 ~]# vim /etc/mha/app1.cnf

[server default]

manager\_workdir=/etc/mha　　#管理主机的工作目录

manager\_log=/etc/mha/manager.log #日志

Master\_ip\_failover\_script=/etc/mha/ master\_ip\_failover #故障切换脚本位置和名称（perl）

ssh\_user=root　　　　　　　#56登录51-55的用户名

ssh\_port=22　　　　　　　　#端口号

repl\_user=repluser #主从同步用户名

repl\_password=123456

user=root　　　　　　　　　　　　#数据库用户名，56监视主库用的

password=123456

[server1]

hostname=192.168.4.51 #主机ip

port=3306

candidate\_master=1 #设置为候选主库master

[server2]

hostname=192.168.4.52

port=3306

candidate\_master=1

[server3]

hostname=192.168.4.53

port=3306

candidate\_master=1

[server4]

hostname=192.168.4.54

port=3306

no\_master=1 #不竟选主库master

[server5]

hostname=192.168.4.55

port=3306

no\_master=1

３）创建vip地址切换脚本

[root@mysql-56 ~]# cp /jluo/mha4mysql-manager-0.56/samples/scripts/master\_ip\_failover /etc/mha/　　　#复制检测perl脚本

[root@mysql-56 ~]#chmod +x /etc/mha/master\_ip\_failover

[root@mysql-56 ~]# vim /etc/mha/master\_ip\_failover

my $vip = '192.168.4.100/24'; # Virtual IP

my $key = "1";

my $ssh\_start\_vip = "/sbin/ifconfig eth0:$key $vip";

my $ssh\_stop\_vip = "/sbin/ifconfig eth0:$key down";

４）51-55授权root给56

mysql> grant all on \*.\* to root@'192.168.4.%' identified by '123456';　#前面配置好了主从同步就直接在51操作自动会同步过去

mysql> select user,host from mysql.user where user='root';　#验证是否添加成功

[root@mysql-56 ~]# yum -y install mariadb　　　#安装包测试一下

[root@mysql-56 ~]# mysql -uroot -p123456 -h192.168.4.51　#测试连接成功就ok了

５）把vip地址绑定在当前主库51的eth0接口上

[root@mysql-51 jluo]# ifconfig eth0:1 192.168.4.100　　#临时绑定vip,不能设置永久的，否则切换不了

[root@mysql-51 jluo]# ifconfig eth0:1

６）启动管理服务

##在管理节点上做ssh检查

[root@mysql-56 ~]# masterha\_check\_ssh --conf=/etc/mha/app1.cnf

Mon Nov 26 16:47:08 2018 - [info] All SSH connection tests passed successfully.

##在管理节点上查看集群状态

[root@mysql-56 ~]# masterha\_check\_repl --conf=/etc/mha/app1.cnf

MySQL Replication Health is OK.

7）启动服务

masterha\_manager --conf=/etc/mha/app1.cnf　[选项]

--remove\_dead\_master\_conf #在/etc/mha/app1.cnf删除宕机主库配置

--ignore\_last\_failover #忽略xxx.headlth文件，即8个小时连续故障切换vip

命令：

[root@mysql-56 ~]# masterha\_manager --conf=/etc/mha/app1.cnf \

> --remove\_dead\_master\_conf --ignore\_last\_failover　#注意终端会挂起

[root@mysql-56 ~]# masterha\_check\_status --conf=/etc/mha/app1.cnf　#查看状态

app1 (pid:9196) is running(0:PING\_OK), master:192.168.4.51

５，测试

测试一：插入查看

50客户端:

[root@mysql-50 ~]# mysql -ujack -p123456 -h192.168.4.100

mysql> insert into db8.t1 values (88);

mysql> select \* from db8.t1;

+------+

| id |

+------+

| 51 |

| 50 |

| 88 |

测试二：关闭51主机查看主库位置

[root@mysql-51 jluo]# systemctl stop mysqld

mysql> select @@hostname;　　　　#刚才开的终端并没有断开

+------------+

| @@hostname |

+------------+

| mysql-52 |　　　　　　　　　#说明已经切换主库了

也可在54从库中查看主库信息

mysql> show slave status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.4.52　　　　　#主机ip是主机52的

在56配置文件查看是否有server1的配置

[root@mysql-56 ~]# cat /etc/mha/app1.cnf | grep server1　#无显示

[root@mysql-56 ~]# masterha\_check\_status --conf=/etc/mha/app1.cnf

app1 is stopped(2:NOT\_RUNNING).　　#主库挂了，hma-manager（管理节点）就被杀死，最后触发脚本执行主库切换操作

154　　　　　　692

测试三：恢复51数据库服务

[root@mysql-56 ~]# vim /etc/mha/app1.cnf　　#添加如下几行

[server1]

candidate\_master=1

hostname=192.168.4.51

port=3306

51主机：设置为52的从库

mysql> change master to \

-> master\_host='192.168.4.52',

-> master\_user='repluser',

-> master\_password='123456',

-> master\_log\_file='master52.000001',

-> master\_log\_pos=298;

mysql> start slave;