MHA 简要配置文档

——马哥 Linux 运维培训：马哥

一、MHA

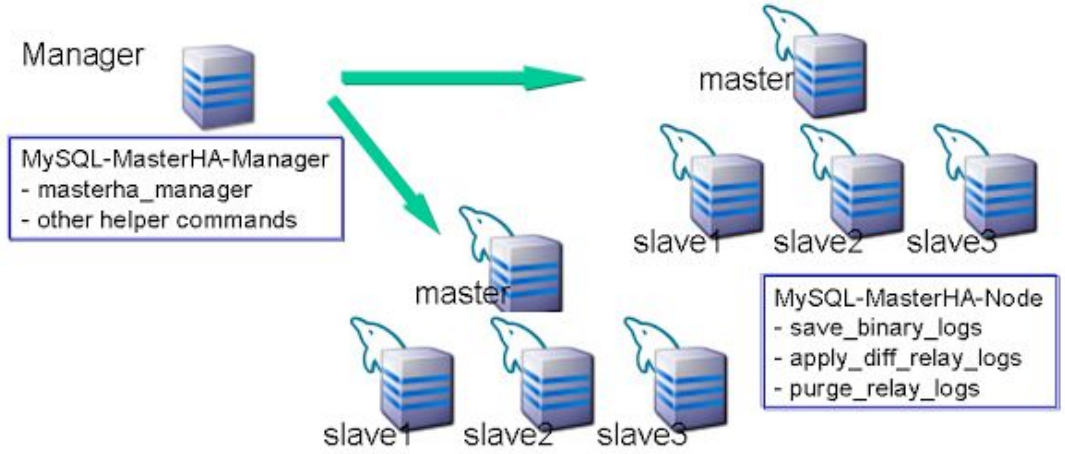
1.1 关于 MHA

MHA（Master HA）是一款开源的 MySQL 的高可用程序，它为 MySQL 主从复制架构提供了 automating master failover 功能。MHA 在监控到 master 节点故障时，会提升其中拥有最新数据的 slave 节点成为新的 master 节点，在此期间，MHA 会通过于其它从节点获取额外信息来避免一致性方面的问题。MHA 还提供了 master 节点的在线切换功能，即按需切换 master/slave 节点。

MHA 服务有两种角色，MHA Manager(管理节点)和 MHA Node(数据节点)：

MHA Manager：通常单独部署在一台独立机器上管理多个 master/slave 集群，每个 master/slave 集群称作一个 application；

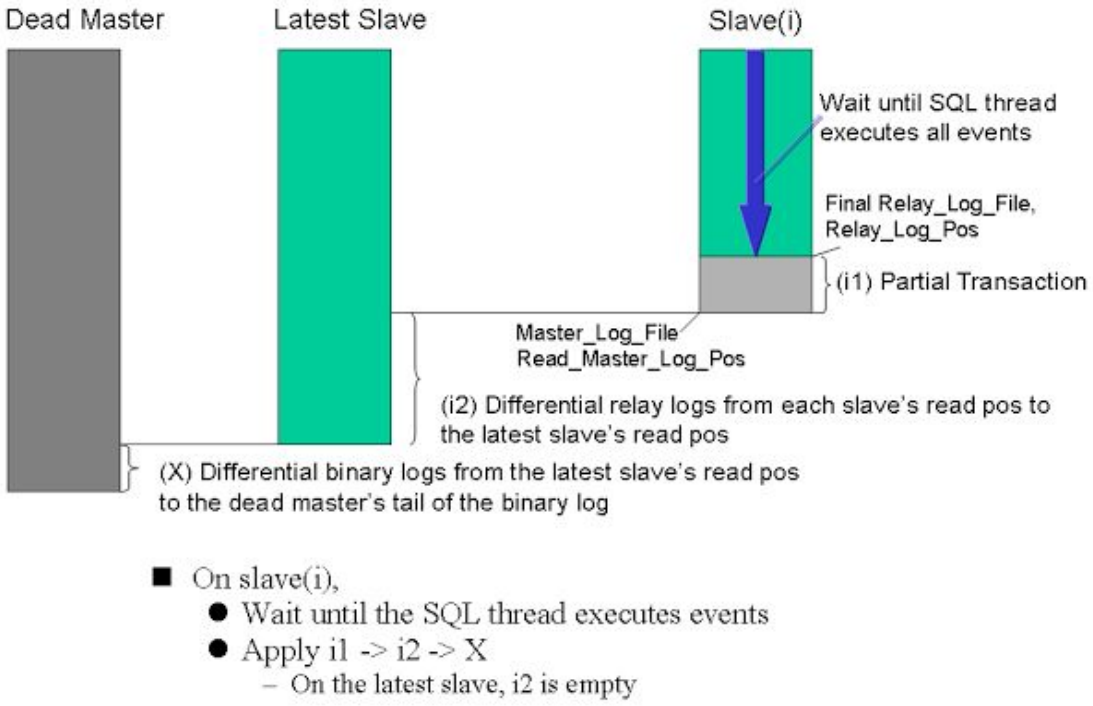
MHA node：运行在每台 MySQL 服务器上(master/slave/manager)，它通过监控具备解析和清理 logs 功能的脚本来加快故障转移。



- Manager package: Can manage multiple {master, slaves} pairs
  - masterha_manager : Automated master monitoring and failover command
  - Other helper scripts: Manual master failover, online master switch, con checking, etc
- Node package: Deploying on all MySQL servers
  - save_binary_logs: Copying master's binary logs if accessible
  - apply_diff_relay_logs: Generating differential relay logs from the latest slave, and applying all differential binlog events
  - purge_relay_logs: Deleting relay logs without stopping SQL thread

1.2 Architecture of MHA

MySQL 复制集群中的 master 故障时，MHA 按如下步骤进行故障转移。



1.3 MHA 组件

MHA 会提供诸多工具程序，其常见的如下所示。

Manager 节点:
- masterha_check_ssh：MHA 依赖的 SSH 环境检测工具；
- masterha_check_repl：MySQL 复制环境检测工具；
- masterha_manager：MHA 服务主程序；
- masterha_check_status：MHA 运行状态探测工具；
- masterha_master_monitor：MySQL master 节点可用性监测工具；
- masterha_master_switch：master 节点切换工具；
- masterha_conf_host：添加或删除配置的节点；
- masterha_stop：关闭 MHA 服务的工具；

Node 节点:
- save_binary_logs：保存和复制 master 的二进制日志：
- apply_diff_relay_logs：识别差异的中继日志事件并应用于其它 slave：
- filter_mysqlbinlog：去除不必要的 ROLLBACK 事件(MHA 已不再使用这个工具)：
- purge_relay_logs：清除中继日志(不会阻塞 SQL 线程)：

自定义扩展：
- secondary_check_script：通过多条网络路由检测 master 的可用性；

- master_ip_failover_script：更新 application 使用的 masterip；
- shutdown_script：强制关闭 master 节点；
- report_script：发送报告；
- init_conf_load_script：加载初始配置参数；
- master_ip_online_change_script：更新 master 节点 ip 地址；

二、准备 MySQL Replication 环境

MHA 对 MySQL 复制环境有特殊要求，例如各节点都要开启二进制日志及中继日志，各从节点必须显式启用其 read-only 属性，并关闭 relay_log_purge 功能等，这里先对其配置做事先说明。

本实验环境共有四个节点，其角色分配如下所示。

node1: MHA Manager
node2: MariaDB master
node3: MariaDB slave
node4: MariaDB slave

各节点的/etc/hosts 文件配置内容中添加：

```
172.16.100.67 node1.magedu.com node1
172.16.100.68 node2.magedu.com node2
172.16.100.69 node3.magedu.com node3
172.16.100.70 node4.magedu.com node4
```

初始主节点 master 配置：

```
server_id=1
relay-log=relay-bin
log-bin=master-bin
```

所有 slava 节点依赖的配置：

```
server_id=2      # 复制集群中的各节点的 id 均必须惟一；
relay-log=relay-bin
log-bin=master-bin
relay_log_purge=0
read_only=1
```

按上述要求分别配置好主从节点之后，按 MySQL 复制配置架构的配置方式将其配置完成，并启动 master 节点和各 slave 节点，以及为各 slave 节点启动其 IO 和 SQL 线程，确保主从复制运行无误。

而后，在所有 MySQL 节点授权拥有管理权限的用户可在本地网络中有其它节点上远程访问。当然，此时仅需要且只能在 master 节点运行类似如下 SQL 语句即可。

mysql> GRANT ALL ON *.* TO 'mhaadmin'@'172.16.100.%' IDENTIFIED BY 'mhapass';

三、安装配置 MHA

3.1 准备基于 ssh 互信通信环境

MHA 集群中的各节点彼此之间均需要基于 ssh 互信通信，以实现远程控制及数据管理功能。简单起见，可在 Manager 节点生成密钥对儿，并设置其可远程连接本地主机后，将私钥文件及 authorized_keys 文件复制给余下的所有节点即可。

下面的操作在 manager 节点操作即可。

```
~]# ssh-keygen -t rsa -P ''
~]# cat .ssh/id_rsa.pub >> .ssh/authorized_keys
~]# chmod go=  .ssh/authorized_keys

~]# scp -p .ssh/id_rsa .ssh/authorized_keys  root@node2:/root/.ssh/
~]# scp -p .ssh/id_rsa .ssh/authorized_keys  root@node3:/root/.ssh/
~]# scp -p .ssh/id_rsa .ssh/authorized_keys  root@node4:/root/.ssh/
```

注意：请事先确保 node2, node3 和 node4 上的/root/.ssh 目录存在。

3.2 安装 MHA

除了源码包，MHA 官方也提供了 rpm 格式的程序包，其下载地址为 https://code.google.com/p/mysql-master-ha/wiki/Downloads?tm=2。CentOS 7 系统可直接使用适用于 el6 的程序包。另外，MHA Manage 和 MHA Node 程序包的版本并不强制要求一致。

Manager 节点：

```
~]# yum install mha4mysql-manager-0.56-0.el6.noarch.rpm
```

所有节点，包括 Manager：

```
~]# yum install mha4mysql-node-0.56-0.el6.noarch.rpm
```

3.3 初始化 MHA

Manger 节点需要为每个监控的 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。全局配置文件默认为/etc/masterha_default.cnf，其为可选配置。如果仅监控一组 master/slave 集群，也可直接通过 application 的配置来提供各服务器的默认配置信息。而每个 application 的配置文件路径为自定义，例如，本示例中将使用/etc/masterha/app1.cnf，其内容如下所示。

```
[server default]
user=mhaadmin          # MySQL Administrator
password=mhapass      # MySQL Administrator's password
manager_workdir=/data/masterha/app1
manager_log=/data/masterha/app1/manager.log
remote_workdir=/data/masterha/app1
ssh_user=root
repl_user=repluser
repl_password=replpass
ping_interval=1

[server1]
hostname=172.16.100.68
#ssh_port=22022
candidate_master=1

[server2]
hostname=172.16.100.69
#ssh_port=22022
candidate_master=1

[server3]
hostname=172.16.100.70
#ssh_port=22022
#no_master=1
```

检测各节点间 ssh 互信通信配置是否 OK：

~]# masterha_check_ssh --conf=/etc/masterha/app1.cnf
输出信息最后一行类似如下信息，表示其通过检测。
[info] All SSH connection tests passed successfully.

检查管理的 MySQL 复制集群的连接配置参数是否 OK：

~]# masterha_check_repl --conf=/etc/masterha/app1.cnf
输出信息如下所示，最后一行的“Health is OK”信息表示通过检测。
Mon Nov 9 17:22:48 2015 - [info] Slaves settings check done.
Mon Nov 9 17:22:48 2015 - [info]
172.16.100.68(172.16.100.68:3306) (current master)
 +--172.16.100.69(172.16.100.69:3306)
 +--172.16.100.70(172.16.100.70:3306)

…… ……

MySQL Replication Health is OK.

启动 MHA：

~]# nohup masterha_manager --conf=/etc/masterha/app1.cnf > /data/masterha/app1/manager.log 2>&1 &

启动成功后，可通过如下命令来查看 master 节点的状态。

~]# masterha_check_status --conf=/etc/masterha/app1.cnf
app1 (pid:4978) is running(0:PING_OK), master:172.16.100.68

上面的信息中“app1 (pid:4978) is running(0:PING_OK)”表示 MHA 服务运行 OK，否则，则会显示为类似“app1 is stopped(1:NOT_RUNNING).”。

如果要停止 MHA，需要使用 masterha_stop 命令。

~]# masterha_stop --conf=/etc/masterha/app1.cnf
Stopped app1 successfully.

3.4 测试故障转移

(1) 在 master 节点关闭 mariadb 服务

~]# killall -9 mysqld mysqld_safe

(2) 在 manager 节点查看日志

/data/masterha/app1/manager.log 日志文件中出现的如下信息，表示 manager 检测到 172.16.100.68 节点故障，而后自动执行故障转移，将 172.16.100.69 提升为了主节点。

Master 172.16.100.68(172.16.100.68:3306) is down!

Check MHA Manager logs at node1.magedu.com:/data/masterha/app1/manager.log for details.

Started automated(non-interactive) failover.
The latest slave 172.16.100.69(172.16.100.69:3306) has all relay logs for recovery.
Selected 172.16.100.69(172.16.100.69:3306) as a new master.
172.16.100.69(172.16.100.69:3306): OK: Applying all logs succeeded.
172.16.100.70(172.16.100.70:3306): This host has the latest relay log events.
Generating relay diff files from the latest slave succeeded.
172.16.100.70(172.16.100.70:3306): OK: Applying all logs succeeded. Slave started, replicating from 172.16.100.69(172.16.100.69:3306)
172.16.100.69(172.16.100.69:3306): Resetting slave info succeeded.

Master failover to 172.16.100.69(172.16.100.69:3306) completed successfully.

注意，故障转移完成后，manager 将会自动停止，此时使用 masterha_check_status 命令检测将会遇到错误提示，如下所示。

```
]# masterha_check_status --conf=/etc/masterha/app1.cnf
app1 is stopped(2:NOT_RUNNING).
```

(3) 提供新的从节点以修复复制集群

原有 master 节点故障后，需要重新准备好一个新的 MySQL 节点。基于来自于 master 节点的备份恢复数据后，将其配置为新的 master 的从节点即可。注意，新加入的节点如果为新增节点，其 IP 地址要配置为原来 master 节点的 IP，否则，还需要修改 app1.cnf 中相应的 ip 地址。随后再次启动 manager，并再次检测其状态。

```
# masterha_check_status --conf=/etc/masterha/app1.cnf
app1 (pid:5508) is running(0:PING_OK), master:172.16.100.69

~]# masterha_check_repl --conf=/etc/masterha/app1.cnf
……
172.16.100.69(172.16.100.69:3306) (current master)
 +--172.16.100.68(172.16.100.68:3306)
 +--172.16.100.70(172.16.100.70:3306)
……
```

四、进一步工作

前面三个步骤已经配置了一个基本的 MHA 环境。不过，为了更多实际应用的需求，还需要进一步完成如下操作。

(1) 提供额外检测机制，以名对 master 的监控做出误判；
(2) 在 master 节点上提供虚拟 ip 地址向外提供服务，以名 master 节点转换时，客户端的请求无法正确送达；
(3) 进行故障转移时对原有 master 节点执行 STONITH 操作以避免脑裂；可通过指定 shutdon_script 实现；
(4) 必要时，进行在线 master 节点转换；