龙果学院《基于 Dubbo 的分布式系统架构视频教程》

高可用架构篇 第 17 节

MyCat 高可用负载均衡集群的实现（HAProxy + Keepalived + MyCat）

一、本节课程的依赖课程

《高可用架构篇--第 13 节--MySQL 源码编译安装（CentOS-6.6+MySQL-5.6）》

《高可用架构篇--第 14 节--MySQL 主从复制的配置（CentOS-6.6+MySQL-5.6）》

《高可用架构篇--第 15 节--MyCat 在 MySQL 主从复制基础上实现读写分离》

《高可用架构篇--第 16 节--MyCat 集群部署（HAProxy+MyCat）》

本节课程要解决的问题：

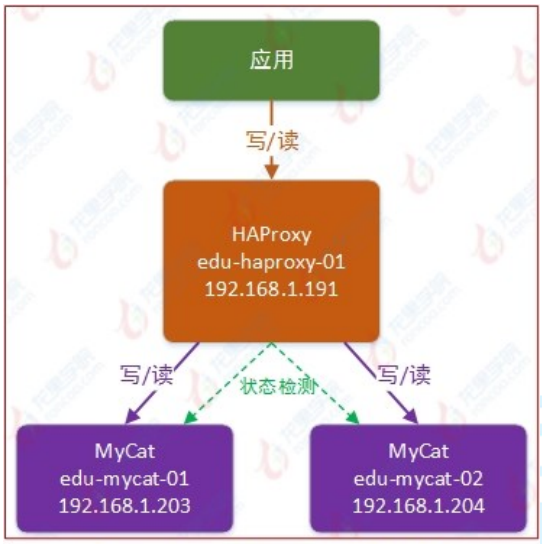


图1：HAProxy节点，还存在单点隐患

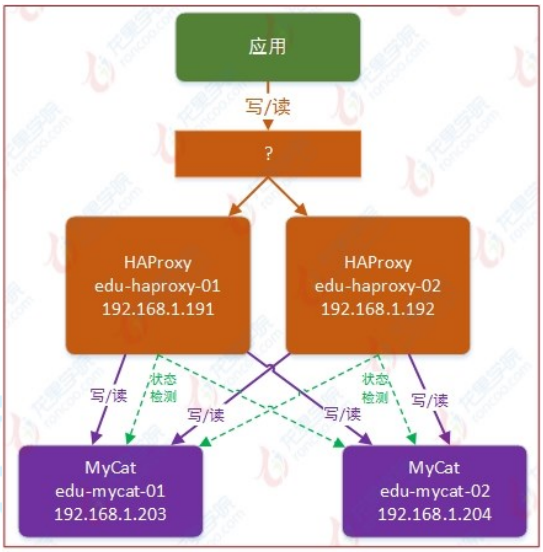


图2：如何实现HAProxy双节点的高可用?

二、软件版本

操作系统：CentOS-6.6-x86_64

JDK 版本：jdk1.7.0_72

Keepalived 版本：keepalived-1.2.18.tar.gz

HAProxy 版本：haproxy-1.5.16.tar.gz

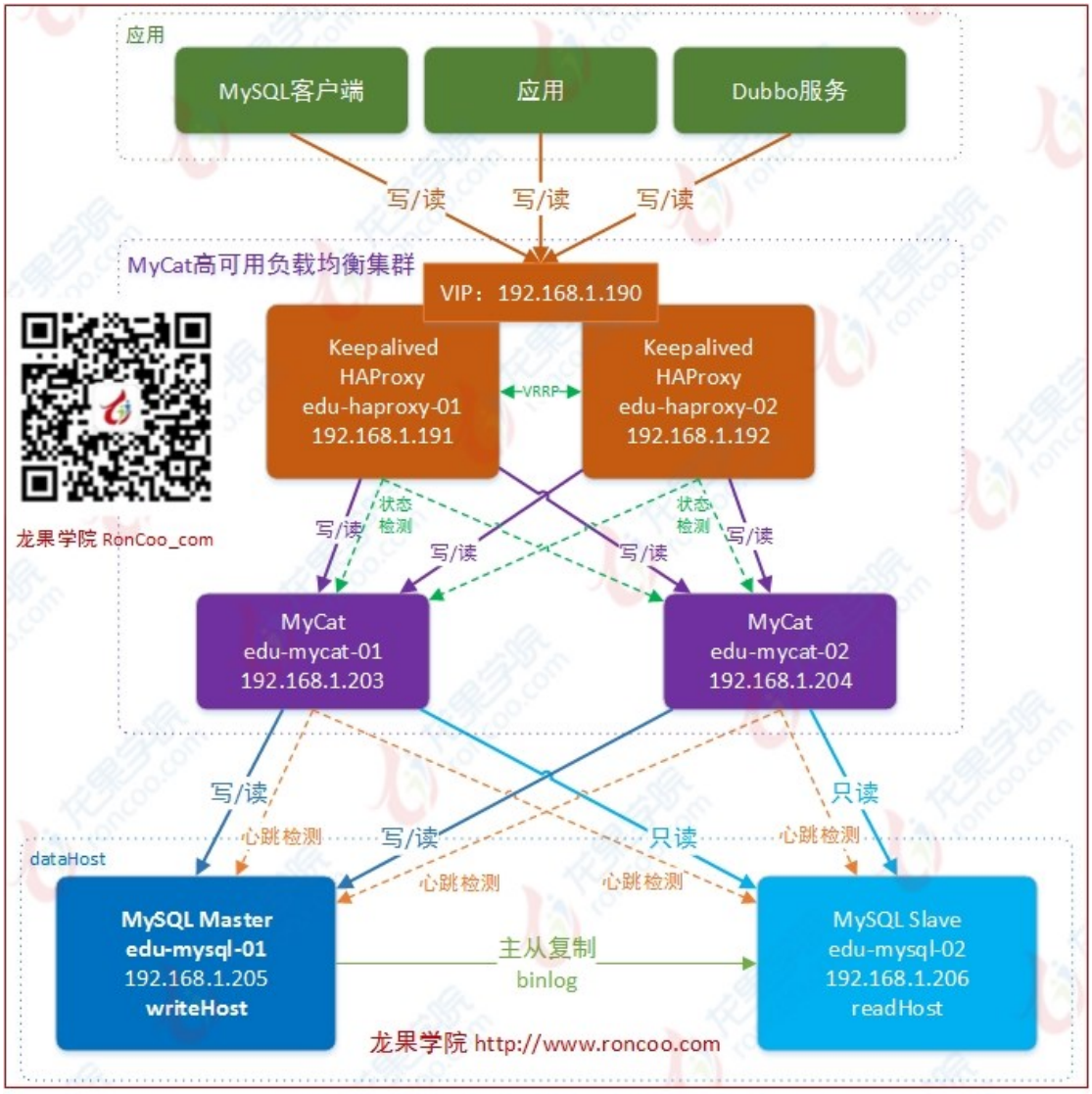
MyCat 版本：Mycat-server-1.4-release-20151019230038-linux.tar.gz

MySQL 版本：mysql-5.6.26.tar.gz

三、部署环境规划

名称	IP		主机名	配置
HAProxy 主机 1	VIP:192.168.1.190	192.168.1.191	edu-haproxy-01（默认 Master）	2 核、2G
HAProxy 主机 2		192.168.1.192	edu-haproxy-02（默认 Backup）	2 核、2G
MyCat 主机 1	192.168.1.203		edu-mycat-01	4 核、4G
MyCat 主机 2	192.168.1.204		edu-mycat-02	4 核、4G
MySQL 主节点	192.168.1.205		edu-mysql-01	4 核、4G
MySQL 从节点	192.168.1.206		edu-mysql-02	4 核、4G

四、MyCat 高可用负载均衡集群部署架构图如下：



MySQL高可用读写分离集群架构【吴水成】

图解说明：

(1) HAProxy 实现了 MyCat 多节点的集群高可用和负载均衡，而 HAProxy 自身的高可用则可以通过 Keepalived 来实现。因此，HAProxy 主机上要同时安装 HAProxy 和 Keepalived，Keepalived 负责为该服务器抢占 vip（虚拟 ip，图中的 192.168.1.190），抢占到 vip 后，对该主机的访问可以通过原来的 ip（192.168.1.191）访问，也可以直接通过 vip（192.168.1.190）访问。

(2) Keepalived 抢占 vip 有优先级，在 keepalived.conf 配置中的 priority 属性决定。但是一般哪台主机上的 Keepalived 服务先启动就会抢占到 vip，即使是 slave，只要先启动也能抢到(要注意避免 Keepalived 的资源抢占问题)。

(3) HAProxy 负责将对 vip 的请求分发到 MyCat 集群节点上，起到负载均衡的作用。同时 HAProxy 也能检测到 MyCat 是否存活，HAProxy 只会将请求转发到存活的 MyCat 上。

(4) 如果 Keepalived+HAProxy 高可用集群中的一台服务器宕机，集群中另外一台服务器上的 Keepalived 会立刻抢占 vip 并接管服务，此时抢占了 vip 的 HAProxy 节点可以继续提供服务。

(5) 如果一台 MyCat 服务器宕机，HAPorxy 转发请求时不会转发到宕机的 MyCat 上，所以 MyCat 依然可用。

综上：MyCat 的高可用及负载均衡由 HAProxy 来实现，而 HAProxy 的高可用，由 Keepalived 来实现。

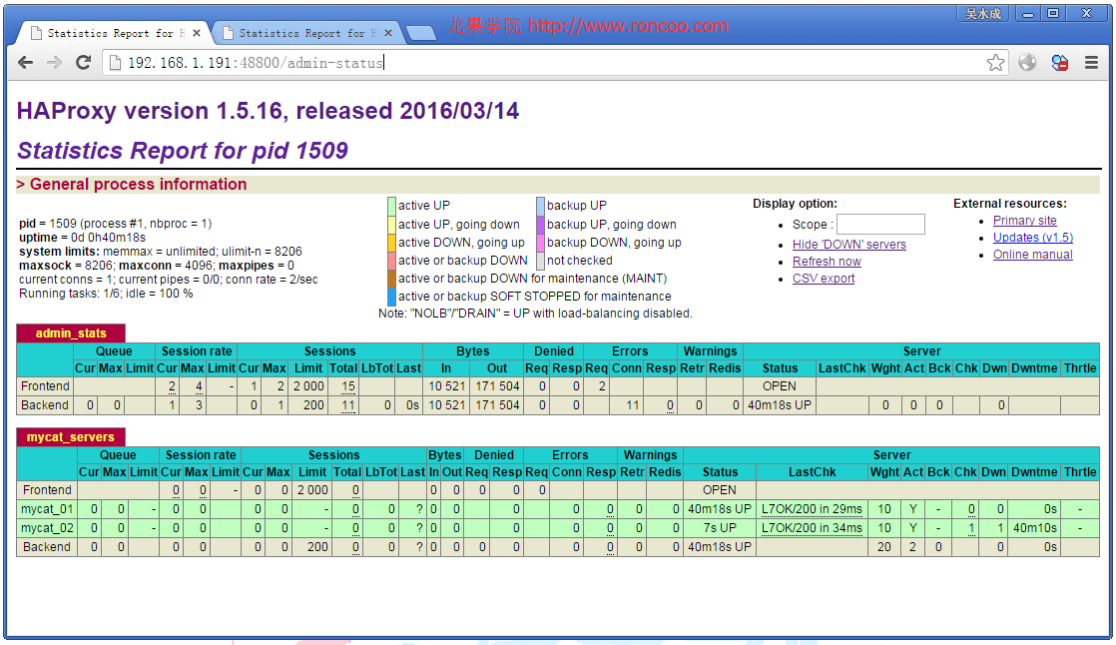
五、HAProxy 节点 2 的部署

HAProxy 主机 2（edu-haproxy-02，192.168.1.192）请参考上一节课程《高可用架构篇--第 16 节--MyCat 集群部署（HAProxy+MyCat）》对进行对等部署和做相应配置。

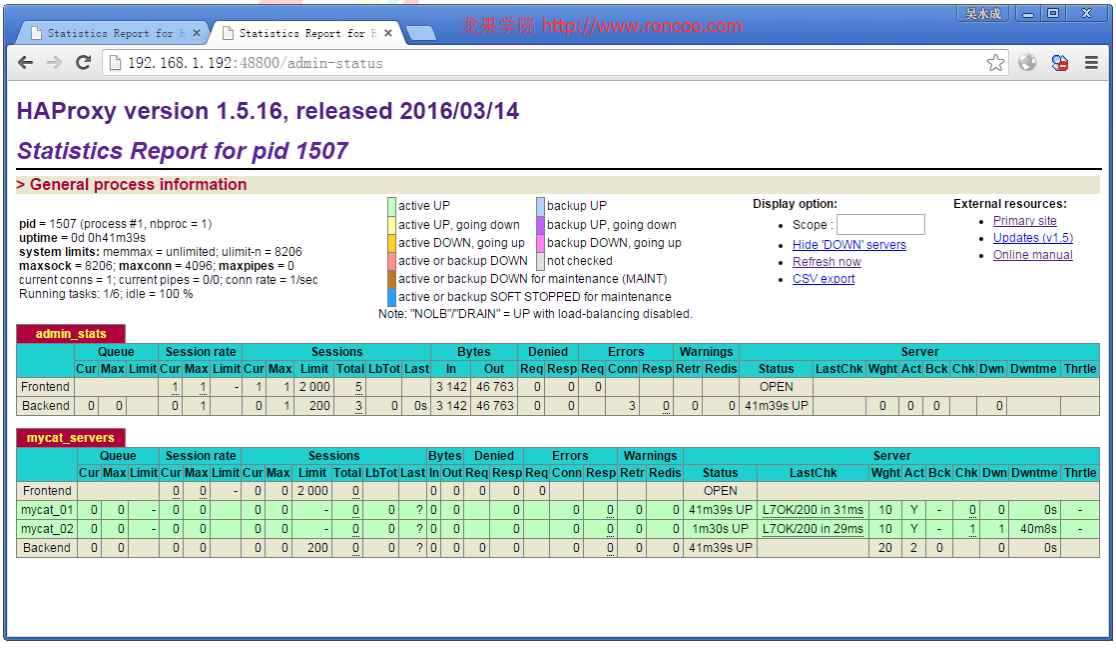
注意配置文件的调整：

多节点部署时 haproxy.cfg 配置文件中的 node 、 description 配置的值要做相应调整。

HAProxy 节点 1 的状态信息页：http://192.168.1.191:48800/admin-status



HAProxy 节点 2 的状态信息页：http://192.168.1.192:48800/admin-status



六、Keepalived 介绍 （官网：http://www.haproxy.org/ ）

Keepalived 是一种高性能的服务器高可用或热备解决方案，Keepalived 可以用来防止服务器单点故障的发生，通过配合 Haproxy 可以实现 web 前端服务的高可用。

Keepalived 以 VRRP 协议为实现基础，用 VRRP 协议来实现高可用性(HA)。VRRP(Virtual Router Redundancy Protocol)协议是用于实现路由器冗余的协议，VRRP 协议将两台或多台路由器设备虚拟成一个设备，对外提供虚拟路由器 IP(一个或多个)，而在路由器组内部，如果实际拥有这个对外 IP 的路由器如果工作正常的话就是 MASTER，或者是通过算法选举产生。MASTER 实现针对虚拟路由器 IP 的各种网络功能，如 ARP 请求，ICMP，以及数据的转发等；其他设备不拥有该虚拟 IP，状态是 BACKUP，除了接收 MASTER 的 VRRP 状态通告信息外，不执行对外的网络功能。当主机失效时，BACKUP 将接管原先 MASTER 的网络功能。

VRRP 协议使用多播数据来传输 VRRP 数据，VRRP 数据使用特殊的虚拟源 MAC 地址发送数据而不是自身网卡的 MAC 地址，VRRP 运行时只有 MASTER 路由器定时发送 VRRP 通告信息，表示 MASTER 工作正常以及虚拟路由器 IP(组)，BACKUP 只接收 VRRP 数据，不发送数据，如果一定时间内没有接收到 MASTER 的通告信息，各 BACKUP 将宣告自己成为 MASTER，发送通告信息，重新进行 MASTER 选举状态。

七、Keepalived 的安装（192.168.1.191、192.168.1.192）

Keepalived （ http://www.keepalived.org/download.html ）

1、上传或下载 keepalived（keepalived-1.2.18.tar.gz）到 /usr/local/src 目录

2、解压安装

安装 keepalived 需要用到 openssl

```
# yum install gcc gcc-c++ openssl openssl-devel
# cd /usr/local/src
# tar -zxvf keepalived-1.2.18.tar.gz
# cd keepalived-1.2.18
# ./configure --prefix=/usr/local/keepalived
# make && make install
```

3、将 keepalived 安装成 Linux 系统服务：

因为没有使用 keepalived 的默认路径安装（默认是/usr/local），安装完成之后，需要做一些工作

复制默认配置文件到默认路径

```
# mkdir /etc/keepalived
# cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/
```

复制 keepalived 服务脚本到默认的地址

```
# cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/
# cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/
# ln -s /usr/local/keepalived/sbin/keepalived /usr/sbin/
# ln -s /usr/local/keepalived/sbin/keepalived /sbin/
```

设置 keepalived 服务开机启动

```
# chkconfig keepalived on
```

4、修改 Keepalived 配置文件

(1) MASTER 节点配置文件（192.168.1.191）

```
# vi /etc/keepalived/keepalived.conf
! Configuration File for keepalived
global_defs {
   ## keepalived 自带的邮件提醒需要开启 sendmail 服务。建议用独立的监控或第三方 SMTP
   router_id edu-haproxy-01  ## 标识本节点的字条串，通常为 hostname
}
## keepalived 会定时执行脚本并对脚本执行的结果进行分析，动态调整 vrrp_instance 的优先级。
## 如果脚本执行结果为 0，并且 weight 配置的值大于 0，则优先级相应的增加。
## 如果脚本执行结果非 0，并且 weight 配置的值小于 0，则优先级相应的减少。
## 其他情况，维持原本配置的优先级，即配置文件中 priority 对应的值。
vrrp_script chk_haproxy {
    script "/etc/keepalived/haproxy_check.sh"  ## 检测 haproxy 状态的脚本路径
    interval 2  ## 检测时间间隔
    weight 2  ## 如果条件成立，权重+2
}
## 定义虚拟路由，VI_1 为虚拟路由的标示符，自己定义名称
vrrp_instance VI_1 {
    state BACKUP  ## 默认主设备（priority 值大的）和备用设备（priority 值小的）都设置为 BACKUP，
            ## 由 priority 来控制同时启动情况下的默认主备，否则先启动的为主设备
    interface eth1  ## 绑定虚拟 IP 的网络接口，与本机 IP 地址所在的网络接口相同，我的是 eth1
    virtual_router_id 91  ## 虚拟路由的 ID 号，两个节点设置必须一样，可选 IP 最后一段使用，
                ## 相同的 VRID 为一个组，他将决定多播的 MAC 地址
    priority 120  ## 节点优先级，值范围 0-254，MASTER 要比 BACKUP 高
    nopreempt  ## 主设备（priority 值大的）配置一定要加上 nopreempt，否则非抢占也不起作用
    advert_int 1  ## 组播信息发送间隔，两个节点设置必须一样，默认 1s

    ## 设置验证信息，两个节点必须一致
    authentication {
        auth_type PASS
        auth_pass 1111  ## 真实生产，按需求对应该过来
    }
    ## 将 track_script 块加入 instance 配置块
    track_script {
        chk_haproxy  ## 检查 HAProxy 服务是否存活
    }

    ## 虚拟 IP 池，两个节点设置必须一样
    virtual_ipaddress {
        192.168.1.190  ## 虚拟 ip，可以定义多个，每行一个
    }
}
```

(2)BACKUP 节点配置文件（192.168.1.192）：

```
# vi /etc/keepalived/keepalived.conf
! Configuration File for keepalived
global_defs {
   router_id edu-haproxy-02
}

vrrp_script chk_haproxy {
    script "/etc/keepalived/haproxy_check.sh"
    interval 2
    weight 2
}

vrrp_instance VI_1 {
    state BACKUP
    interface eth1
    virtual_router_id 91
    priority 110
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    track_script {
        chk_haproxy
    }
    virtual_ipaddress {
        192.168.1.190
    }
}
```

特别注意：如果非抢占模式不生效，在 Keepalived 的故障节点恢复后会再次导抢占 vip，从而因 vip 切换而闪断带来的风险（视频解说）。按以上配置，配置了 Keepalived 非抢占模式，配置及注意点如下：

(1) 主设备、从设备中的 state 都设置为 BACKUP

(2) 主设备、从设备中都不要配置 mcast_src_ip （本机 IP 地址）

(3) 默认主设备（priority 值大的 Keepalived 节点）配置一定要加上 nopreempt，否则非抢占不起作用

(4) 防火墙配置允许组播（主、备两台设备上都需要配置，keepalived 使用 224.0.0.18 作为 Master 和 Backup 健康检查的通信 IP）

```
# iptables -I INPUT -i eth1 -d 224.0.0.0/8 -p vrrp -j ACCEPT
# iptables -I OUTPUT -o eth1 -d 224.0.0.0/8 -p vrrp -j ACCEPT
```

（eth1 为主机的网卡设备名称，生产环境服务器可以用独立网卡来处理组播和心跳检测等）

```
# service iptables save
```

重启防火墙：

```
# service iptables restart
```

5、编写 Haproxy 状态检测脚本 /etc/keepalived/haproxy_check.sh（已在 keepalived.conf 中配置）

脚本要求：如果 haproxy 停止运行，尝试启动，如果无法启动则杀死本机的 keepalived 进程，keepalied 将虚拟 ip 绑定到 BACKUP 机器上。内容如下：

```
# mkdir -p /usr/local/keepalived/log
# vi /etc/keepalived/haproxy_check.sh
#!/bin/bash
START_HAPROXY="/etc/rc.d/init.d/haproxy start"
STOP_HAPROXY="/etc/rc.d/init.d/haproxy stop"
LOG_FILE="/usr/local/keepalived/log/haproxy-check.log"
HAPS=`ps -C haproxy --no-header |wc -l`
date "+%Y-%m-%d %H:%M:%S" >> $LOG_FILE
echo "check haproxy status" >> $LOG_FILE
if [ $HAPS -eq 0 ];then
    echo $START_HAPROXY >> $LOG_FILE
    $START_HAPROXY >> $LOG_FILE 2>&1
    sleep 3
    if [ `ps -C haproxy --no-header |wc -l` -eq 0 ];then
        echo "start haproxy failed, killall keepalived" >> $LOG_FILE
        killall keepalived
    fi
fi
```

保存后，给脚本赋执行权限：

```
# chmod +x /etc/keepalived/haproxy_check.sh
```

6、启动 Keepalived

```
# service keepalived start
Starting keepalived: [  OK  ]
```

Keepalived 服务管理命令：

停止：service keepalived stop

启动：service keepalived start

重启：service keepalived restart

查看状态：service keepalived status

八、Keepalived + Haproxy 的高可用测试

1、关闭 192.168.1.191 中的 Haproxy，Keepalived 会将它重新启动

```
# service haproxy stop
```

```
[root@edu-haproxy-01 keepalived]#
[root@edu-haproxy-01 keepalived]# service haproxy stop
Shutting down haproxy: [  OK  ]
[root@edu-haproxy-01 keepalived]# ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 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: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 00:50:56:a1:4a:fd brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.191/24 brd 192.168.1.255 scope global eth1
    inet 192.168.1.190/32 scope global eth1
    inet6 fe80::250:56ff:fea1:4afd/64 scope link
       valid_lft forever preferred_lft forever
[root@edu-haproxy-01 keepalived]#
```


2、关闭 192.168.1.191 中的 Keepalived，VIP（192.168.1.190）会被 192.168.1.192 抢占

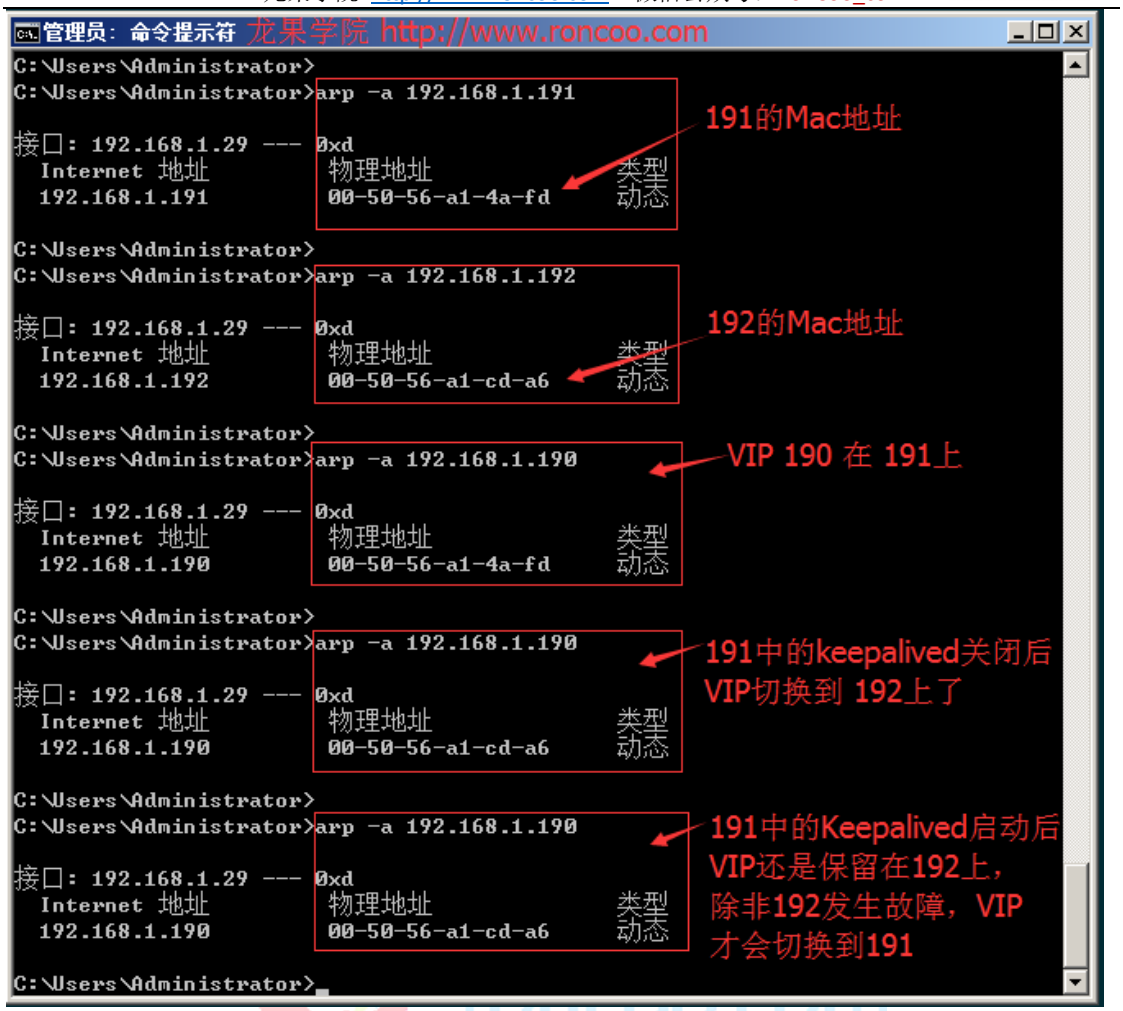
```
# service keepalived stop
```

```
[root@edu-haproxy-01 keepalived]# service keepalived stop
Stopping keepalived: [  OK  ]
[root@edu-haproxy-01 keepalived]# ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 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: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 00:50:56:a1:4a:fd brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.191/24 brd 192.168.1.255 scope global eth1
    inet6 fe80::250:56ff:fea1:4afd/64 scope link
       valid_lft forever preferred_lft forever
[root@edu-haproxy-01 keepalived]#
```


Keepalived 停止后，192.168.1.191 节点的网络接口中的 VIP（192.168.1.190）将消失

```
[root@edu-haproxy-02 keepalived]# ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 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: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 00:50:56:a1:cd:a6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.192/24 brd 192.168.1.255 scope global eth1
    inet 192.168.1.190/32 scope global eth1
    inet6 fe80::250:56ff:fea1:cda6/64 scope link
       valid_lft forever preferred_lft forever
[root@edu-haproxy-02 keepalived]#
```

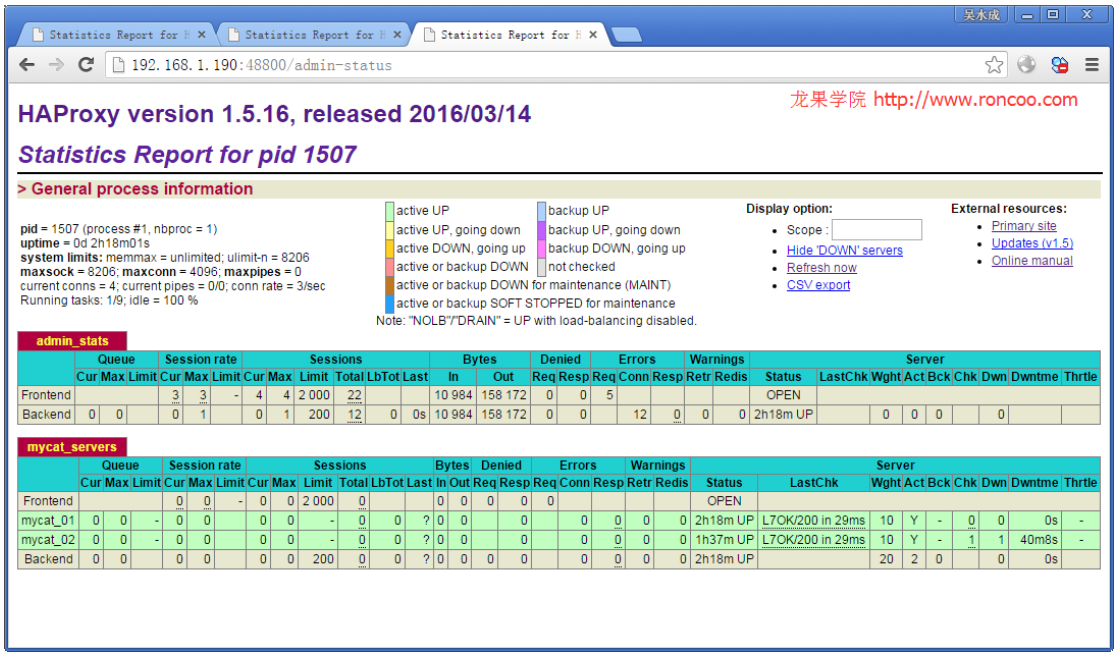
查看此时 VIP 对应的 MAC，Windows 下使用 CMD 命令查看：



说明此时 VIP 已经漂移到物理主机 192.168.1.192 上了

再通过 VIP 来访问 Haproxy 集群，访问到的也是 192.168.1.192

http://192.168.1.190:48800/admin-status



3、重新启动 192.168.1.191 中的 Keepalived，vip（192.168.1.190）保留在 192.168.1.192 主机上，不会出现 191 启动抢占 vip 的情况。

```
# service keepalived start
```

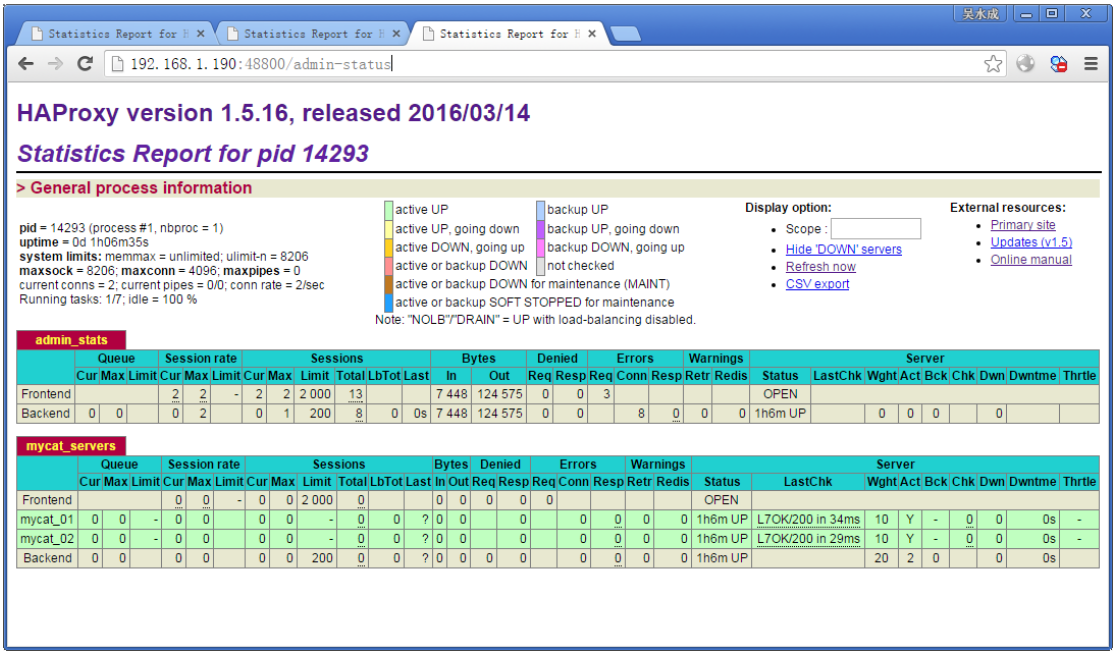
```
[root@edu-haproxy-01 log]# service keepalived start
Starting keepalived: [  OK  ]
[root@edu-haproxy-01 log]# ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 00:50:56:a1:4a:fd brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.191/24 brd 192.168.1.255 scope global eth1
    inet6 fe80::250:56ff:fea1:4afd/64 scope link
       valid_lft forever preferred_lft forever
[root@edu-haproxy-01 log]#
```


4、模拟抢占了 vip 的节点（192.168.1.192）中的 HAProxy 故障或启动失败。

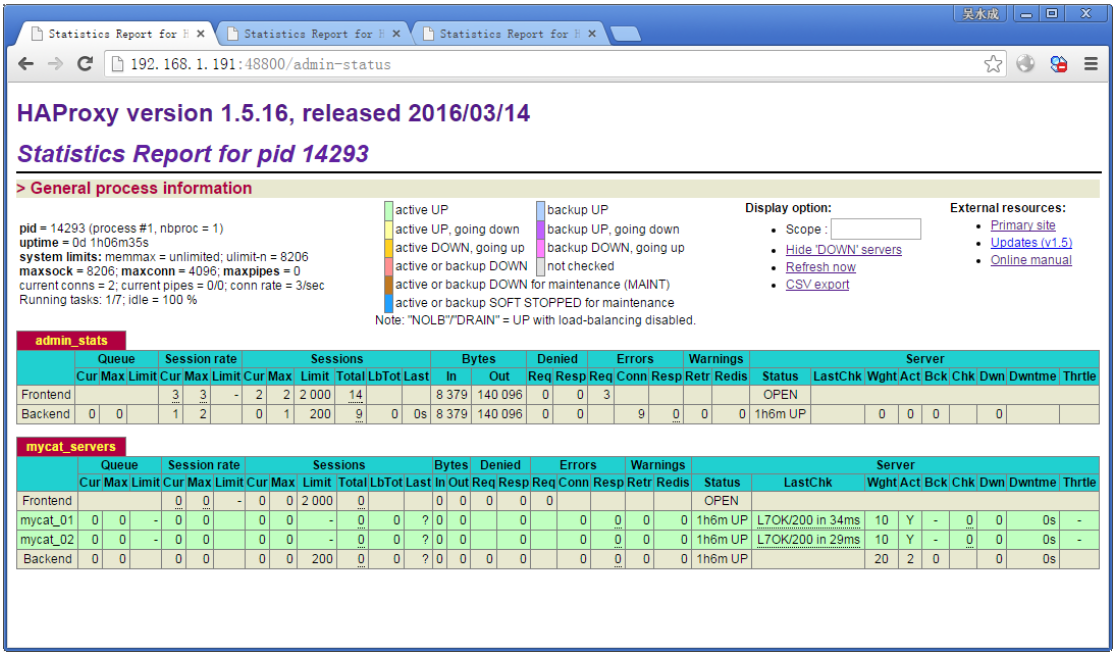
方式：把 192 节点中的 haproxy.cfg 文件重命名为 haproxy.cfg_bak，并把 haproxy 服务进行 kill 掉，此时 keepalived 会尝试去启动 haproxy，会由于找不到配置文件而启动失败，此时就会进行 haproxy_check.sh 脚本中的 killall keepalived 命令，结束 keepalived 进行。随后就是 192.168.1.191 节点重新抢占 vip

```
[root@edu-haproxy-01 log]# ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 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: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 00:50:56:a1:4a:fd brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.191/24 brd 192.168.1.255 scope global eth1
    inet 192.168.1.190/32 scope global eth1
    inet6 fe80::250:56ff:fea1:4afd/64 scope link
       valid_lft forever preferred_lft forever
[root@edu-haproxy-01 log]#
```


http://192.168.1.190:48800/admin-status



http://192.168.1.191:48800/admin-status



九、通过 vip 访问数据库、验证 vip 切换后的数据库访问

更多课程信息，请关注 龙果学院 官方网站 http://www.roncoo.com/

或关注 龙果 微信公众号 RonCoo_com