

一、什么是集群？一组计算机完成相同的工作。

二、为什么要使用集群？处理高并发访问。

三、集群的分类？

LB 负载均衡集群（多台主机共同分担同一项服务）

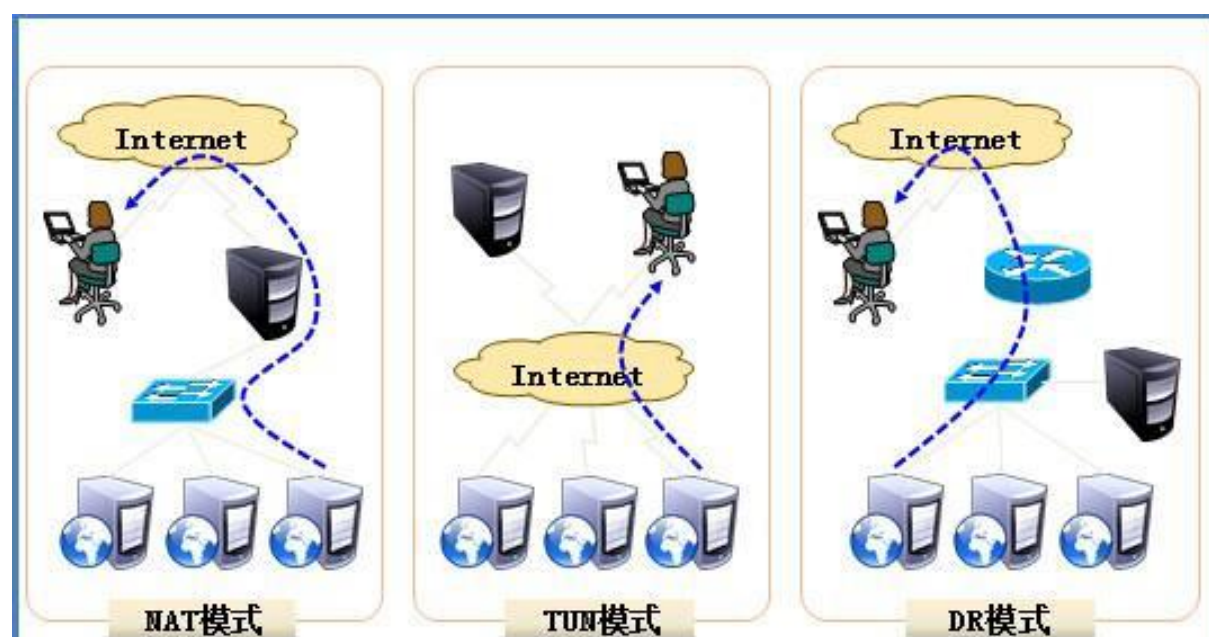
HA 高可用集群（一主一备 当主不能提供服务时 备用服务器接替主提供服务）

HPC 高性能计算集群（主要用来作运算 应用在一些专业领域：气象云图、地质勘探、航天、航空）

四、配置 LB 集群

4.1、LVS：linux 虚拟服务，由中国人章文嵩博士开发，现已被编入 linux 内核。

4.2、LVS 工作模式：VS/nat 模式、VS/DR 模式、VS/TUN 模式



五、LVS 中主机的角色

客户端：

分发器(director)：把客户端的请求分发给提供服务的服务器

Realserver：提供具体服务的服务器

六、LVS 术语

cip 客户端主机的 ip 地址

vip 分发器公网接口 ip 地址 用来对外提供服务

dip 分发器私网接口 ip 地址 又叫直连 ip

rip 真实服务器的 ip 地址

七、VS/NAT 模式实验拓扑



cip 2.2.2.2/8
vip 2.2.2.1/8
dip 192.168.1.1/24
rip 192.168.1.100/24 192.168.1.200/24

VS/NAT 配置过程:

Client: ifconfig eth0 2.2.2.2/8
route add default gw 2.2.2.1/8

web_A: ifconfig eth0 192.168.1.100/24
route add default gw 192.168.1.1/24
yum -y install httpd
service httpd start
echo 192.168.1.100 > /var/www/html/index.html

web_B: ifconfig eth0 192.168.1.200/24
route add default gw 192.168.1.1/24
yum -y install httpd
service httpd start
echo 192.168.1.200 > /var/www/html/index.html

LVS: ifconfig eth0 2.2.2.1/8
ifconfig eth1 192.168.1.1/24

//开启内核的路由转发功能

方法 1

echo 1 > /proc/sys/net/ipv4/ip_forward
echo “echo 1 > /proc/sys/net/ipv4/ip_forward” >> /etc/rc.local

方法 2

```
vim /etc/sysctl.conf
```

```
net.ipv4.ip_forward = 1
```

```
:wq
```

```
[root@localhost ~]# sysctl -p
```

//安装软件包 LB 集群功能的软件包 (此包无依赖, 也可直接使用 rpm -ivh 安装)

```
[root@localhost /]# mount /dev/cdrom1 /mnt
```

```
[root@localhost /]# cat /etc/yum.repos.d/a.repo
```

```
[rhel-LoadBalancer]
```

```
name=Red Hat Enterprise Linux $releasever - $basearch - Source
```

```
baseurl=file:///mnt/LoadBalancer
```

```
enabled=1
```

```
gpgcheck=0
```

```
:wq
```

```
[root@localhost yum.repos.d]#yum clean all
```

```
[root@localhost yum.repos.d]#yum -y install ipvsadm
```

```
[root@localhost ~]# rpm -q ipvsadm
```

```
ipvsadm-1.25-10.el6.x86_64
```

```
[root@localhost ~]#
```

//编写策略

```
[root@localhost ~]# ipvsadm -A -t 2.2.2.1:80 -s rr
```

```
[root@localhost ~]# ipvsadm -a -t 2.2.2.1:80 -r 192.168.1.100 -m
```

```
[root@localhost ~]# ipvsadm -a -t 2.2.2.1:80 -r 192.168.1.200 -m
```

```
[root@localhost ~]# service ipvsadm save #保存策略, 不保存只当前有效
```

```
ipvsadm: Saving IPVS table to /etc/sysconfig/ipvsadm: [确定]
```

```
[root@localhost ~]# chkconfig ipvsadm on #设置开机启动
```

[root@localhost ~]# service ipvsadm status #查看状态

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port	Forward	Weight	ActiveConn	InActConn
TCP 2.2.2.1:80 rr				
-> 192.168.1.100:80	Masq	1	0	0
-> 192.168.1.200:80	Masq	1	0	0

[root@localhost ~]#

ipvsadm 命令选项说明:

-A 添加虚拟服务 (ip 地址是分发器 vip 的地址)

-t tcp 传输协议

-s 指定算法

rr 轮询算法 (你一次我一次)

ipvsadm -A -t 2.2.2.2:80 -s rr

-a 向虚拟服务里添加 realserver

-r 指定真正提供服务的服务器的 ip 地址

-m 集群的工作模式是 nat 模式(m 伪装的意思)

-g 集群的工作模式是 DR 模式

-w 指定权重值 不指定时, 默认值是 1

[root@localhost ~]# ipvsadm -a -t 2.2.2.1:80 -r 192.168.1.100:80 -m

[root@localhost ~]# ipvsadm -a -t 2.2.2.1:80 -r 192.168.1.100:80 -m -w 2

-d 把 realserver 从虚拟服务里删除。 ipvsadm -d -t 2.2.2.1:80 -r 192.168.1.100:80

-D 删除虚拟服务。 ipvsadm -D -t 2.2.2.1:80

-E 修改调度算法。 ipvsadm -E -t 2.2.2.1:80 -s wrr

-e 修改 realserver 的权重值。

ipvsadm -e -t 2.2.2.1:80 -r 192.168.1.100:80 -m -w 1

-C 清空策略。 ipvsadm -C
-Ln 查看策略信息。 ipvsadm -Ln
--stats 查看进程包 和 进出 字节数的信息。 ipvsadm -Ln --stats

八、客户端测试

```
[root@localhost ~]# ifconfig | head -2  
eth0        Link encap:Ethernet   HWaddr 00:0C:29:AF:B9:F1  
            inet addr:2.2.2.2   Bcast:2.255.255.255   Mask:255.0.0.0
```

```
[root@localhost ~]#
```

```
[root@localhost ~]# route -n | tail -1  
0.0.0.0        2.2.2.1        0.0.0.0        UG    0    0    0 eth2
```

```
[root@localhost ~]#
```

```
[root@localhost ~]# elinks --dump http://2.2.2.1  
192.168.1.200
```

```
[root@localhost ~]# elinks --dump http://2.2.2.1  
192.168.1.100
```

```
[root@localhost ~]# elinks --dump http://2.2.2.1  
192.168.1.200
```

```
[root@localhost ~]# elinks --dump http://2.2.2.1  
192.168.1.100
```

```
[root@localhost ~]#
```

九、查看分发器的状态信息

```
[root@localhost ~]# ipvsadm -Ln --stats
```

IP Virtual Server version 1.2.1 (size=4096)

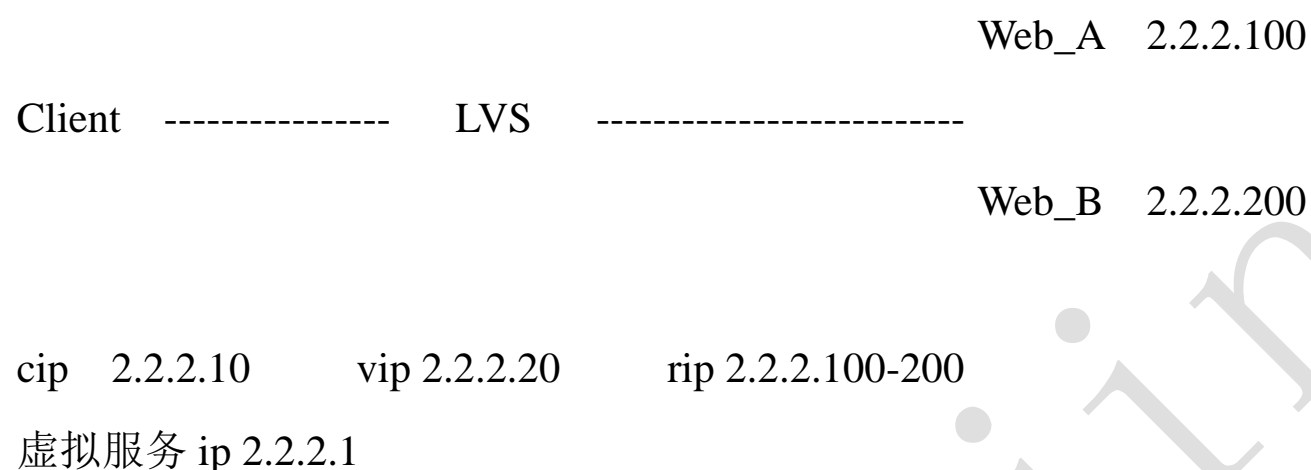
Prot	LocalAddress:Port	Conns	InPkts	OutPkts	InBytes	OutBytes
	-> RemoteAddress:Port					
TCP	2.2.2.1:80	4	20	20	1652	2196
	-> 192.168.1.100:80	2	10	10	826	1098
	-> 192.168.1.200:80	2	10	10	826	1098

```
[root@localhost ~]#
```

会发现：出去的字节数远远大于进入的字节数，当客户端访问的数据量大时，分发器就成了数据传输的瓶颈!!! 若让分发器只负责分发请求，realserver 直接把数据回复给客户端的话，这个问题就解决了^_^。 VS/DR 模式就是让 realserver 直接把数据回复给客户端。

十、配置 VS/DR 模式 LB 集群

* 配置 VS/DR 模式时，vip 和 rip 必须在同一网段内。生产环境中都是公网 ip 地址。



分析：

client 访问的目标地址是 vip；所以只有 vip 给 client 回包,client 才会收；其他地址给 client 回的包 client 是不会收的。那怎样才能让 client 收 realserver 的包呢？就是让 realserver 拥有 vip 地址；但让 realserver 拥有 vip 地址后，在 DR 这个模式里，此时就有 3 台主机都有 vip 地址了；这时，客户端发请求时，就有 3 台主机给客户端回包，但是真正能给客户端提供分发的主机只有分发器；若是 realserver 用 vip 地址给 client 回包的话,这时候 client 端的请求就不能正确的被分发了。所以当 realserver 拥有了 vip 地址后，要让 realserver 不响应 client 主机发送的找 vip 地址的 arp 广播包就 ok 了。假设此时 realserver 没有回应 client 端访问分发器 vip 地址的请求，只有分发器接收到了客户端的请求，此时分发器就要发送找 realserver 的 arp 广播包，此时分发器有 2 个地址 vip 和 dip,此时要保证，分发器找 realserver 的 ARP 广播包，要从分发器自己的 dip 接口发送出去。因为，若分发器把找 realserver 的 arp 广播包从自己的 vip 发出去，2 个 realserver 也有 VIP 地址，当 realserver 回包时，会发现源 ip 是 rip；目的 ip 是 vip,因 realserver 自己也有 vip，这时 realserver 就会认为这个包是给自己的；自己就收下了这个包。这样就会导致分发器找不到 realserver 主机，这样的话请求就分发不出去，客户端也就得不到服务了，但若分发器把找 realserver 的请求从自己的 dip 口发出去的话,realserver 回包时，包中源地址是 rip，目的地址是 dip 这样分发器就知道把请求分发给那个 realserver 了。

10.1、VS/DR 模式要解决的几个问题？

- 1、让 realserver 拥有 vip 地址
- 2、让 realserver 不响应客户端访问分发器 vip 地址的 arp 广播包。
- 3、让分发器把找 realserver 的 ARP 广播包从自己的 dip 接口发送出去。

10.2、让 realservr 拥有 vip 地址

Web_A 配置:

```
[root@localhost ~]# ifconfig | head -2
```

```
eth1      Link encap:Ethernet  HWaddr 00:0C:29:6D:FB:03  
          inet addr:2.2.2.100  Bcast:192.168.1.255  Mask:255.255.255.0
```

```
[root@localhost ~]# ifconfig  lo:1  2.2.2.1/32
```

```
[root@localhost ~]#
```

```
[root@localhost ~]# echo 1 > /proc/sys/net/ipv4/conf/lo/arp_ignore
```

```
[root@localhost ~]# echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
```

```
[root@localhost ~]# echo 2 > /proc/sys/net/ipv4/conf/lo/arp_announce
```

```
[root@localhost ~]# echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
```

Web_B 配置:

```
[root@localhost ~]# ifconfig  | head -2
```

```
eth1      Link encap:Ethernet  HWaddr 00:0C:29:93:93:39  
          inet addr: 2.2.2.200  Bcast:192.168.1.255  Mask:255.255.255.0
```

```
[root@localhost ~]#
```

```
[root@localhost ~]# ifconfig  lo:1  2.2.2.1/32
```

```
[root@localhost ~]#
```

```
[root@localhost ~]# echo 1 > /proc/sys/net/ipv4/conf/lo/arp_ignore
```

```
[root@localhost ~]# echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
```

```
[root@localhost ~]# echo 2 > /proc/sys/net/ipv4/conf/lo/arp_announce
```

```
[root@localhost ~]# echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
```

1 只响应找自己的 arp 广播包

2 用自己 MAC 地址帮兄弟接收和回应 arp 广播包，找自己的也接收

lo eth1 或 eth1 eth2 这些网络接口 都在一层上 被称为兄弟接口

//内核参数说明:

arp_ignore :

0(默认值): 回应任何网络接口上对任何本地 IP 地址的 arp 查询请求

-
- 1 只回答目标 IP 地址是来访网络接口本地地址的 ARP 查询请求
 - 2 只回答目标 IP 地址是来访网络接口本地地址的 ARP 查询请求,且来访 IP 必须在该网络接口的子网段内
 - 3 不回答该网络接口的 arp 请求, 而只对设置的唯一和连接地址做出回应
 - 4-7 保留未使用
 - 8 不回应所有（本地地址）的 arp 查询

arp_announce:

0 (默认): 在任意网络接口上的任何本地地址

1 尽量避免不在该网络接口子网段的本地地址做出 arp 回应。当发起 ARP 请求的源 IP 地址是被设置应该经由路由达到此网络接口的时候很

有用。此时会检查来访 IP 是否为所有接口上的子网段内 ip 之一。如果改来访 IP 不属于各个网络接口上的子网段内, 那么将采用级别 2 的方式来进行处理

2 对查询目标使用最适当的本地地址。在此模式下将忽略这个 IP 数据包的源地址并尝试选择与能与该地址通信的本地地址。首要是选择所

有的网络接口的子网中外出访问子网中包含该目标 IP 地址的本地地址。如果没有合适的地址被发现, 将选择当前的发送网络接口或其他的可能接受到该 ARP 回应的网络接口来进行发送。

10.2、分发器配置

```
[root@localhost ~]#
```

```
[root@localhost ~]# ifconfig eth0 | head -2
```

```
eth0      Link encap:Ethernet  HWaddr 00:0C:29:3A:C8:B8
          inet addr: 2.2.2.20  Bcast:2.2.2.255  Mask:255.0.0.0
```

```
[root@localhost ~]#
```

```
[root@localhost ~]# ifconfig eth0:1 2.2.2.1/32
```

```
[root@localhost ~]# ipvsadm -C
```

```
[root@localhost ~]# ipvsadm -A -t 2.2.2.1:80 -s rr
```

```
[root@localhost ~]# ipvsadm -a -t 2.2.2.1:80 -r 2.2.2.100:80 -g
```

```
[root@localhost ~]# ipvsadm -a -t 2.2.2.1:80 -r 2.2.2.200:80 -g
```

```
[root@localhost ~]#
```



```
[root@localhost ~]# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096):
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward  Weight ActiveConn InActConn
TCP  2.2.2.1:80 rr
  -> 2.2.2.100:80                 Route    1      0          0
  -> 2.2.2.200:80                 Route    1      0          0
[root@localhost ~]#
```

```
[root@localhost ~]# ipvsadm -Ln --stats
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port          Conns    InPkts  OutPkts  InBytes  OutBytes
  -> RemoteAddress:Port
TCP  2.2.2.1:80                  0         0         0         0         0
  -> 2.2.2.200:80                0         0         0         0         0
  -> 2.2.2.100:80                0         0         0         0         0
[root@localhost ~]#
```

十一、客户端访问

```
[root@localhost ~]# ifconfig eth0 2.2.2.10
elinks --dump http://2.2.2.1 //交替出现 2 个 realserver 的 web 页面就 ok 了
```

十二、在分发器上查看状态

```
ipvsadm -Ln --stats
//查看数据包流量 只有从分发器进入的数据包 没有从分发器出去的数据包

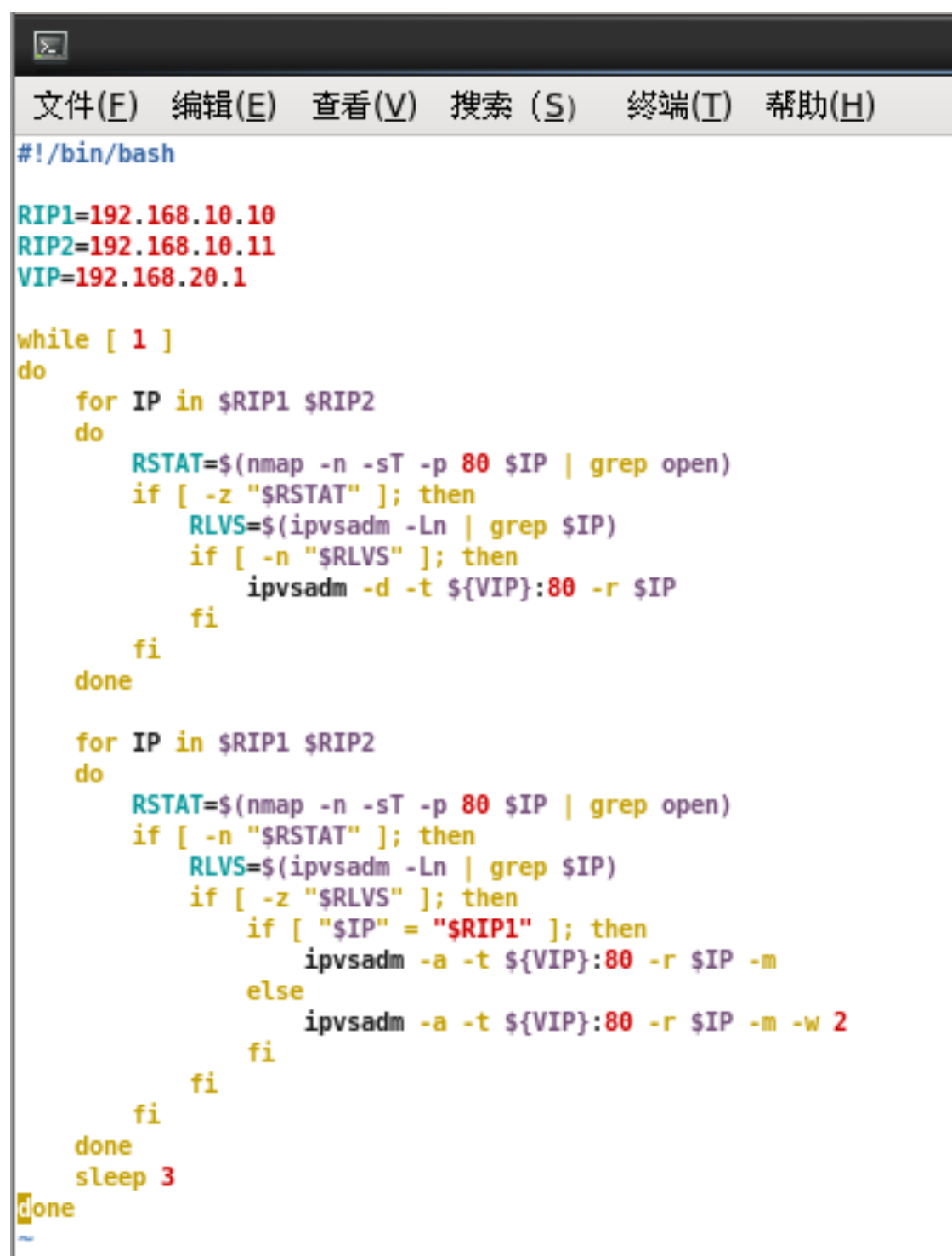
ipvsadm -Ln -c
//在分发器上，查看访问自己的客户端地址
```

十三、LVS 没有对 realserver 做健康性检查的功能,要自己写检查脚本。

*当把某个 realserver 上的 web 服务停止后，分发器仍然会把客户端的请求分发给此 realserver, 并且也不会把其从策略中删除。

```
[root@localhost ~]# yum -y nmap
```

```
[root@localhost ~]# vim /root/check_lvs.sh
```



```
#!/bin/bash

RIP1=192.168.10.10
RIP2=192.168.10.11
VIP=192.168.20.1

while [ 1 ]
do
    for IP in $RIP1 $RIP2
    do
        RSTAT=$(nmap -n -sT -p 80 $IP | grep open)
        if [ -z "$RSTAT" ]; then
            RLVS=$(ipvsadm -Ln | grep $IP)
            if [ -n "$RLVS" ]; then
                ipvsadm -d -t ${VIP}:80 -r $IP
            fi
        fi
    done

    for IP in $RIP1 $RIP2
    do
        RSTAT=$(nmap -n -sT -p 80 $IP | grep open)
        if [ -n "$RSTAT" ]; then
            RLVS=$(ipvsadm -Ln | grep $IP)
            if [ -z "$RLVS" ]; then
                if [ "$IP" = "$RIP1" ]; then
                    ipvsadm -a -t ${VIP}:80 -r $IP -m
                else
                    ipvsadm -a -t ${VIP}:80 -r $IP -m -w 2
                fi
            fi
        fi
    done
    sleep 3
done
```

在分发器上，置入后台运行 `sh /root/check_lvs.sh &` 然后手动停止某个 web 服务器的网站服务 `service httpd stop`

在分发器上查看策略 `ipvsadm -Ln` 会看到 web 服务停止服务的条目在策略中没有了，再手动把服务启动，又自动添加上了。