**1、下载安装ipvsadm**

LVS本身是一个内核模块：ip\_vs，位于内核目录net/netfilter/ipvs/下，从2.4内核开始，linux kernel已经集成了ipvs模块；因此，当前系统(CentOS6.8)kernel都是支持LVS功能的，无需额外下载；uname -r查看内核版本并在官网：http://www.linuxvirtualserver.org/software/ipvs.html选择相对应的版本下载：

**[root@vs1 ~]#uname -r #查看内核版本**

**2.6.32-642.el6.x86\_64**

**[root@vs1 ~]#lsmod | grep ip\_vs #查看内核是否加载进了ip\_vs模块**

**[root@vs1 ~]#tar zxvf ipvsadm-1.24.tar.gz -C /usr/src**

**[root@vs1 ~]#yum install kernel-devel -y**

**[root@vs1 ~]#ln -sv /usr/src/kernel/\* /usr/src/linux #建立软连接指向Linux内核**

**[root@vs1 ~]#cd /usr/src/ipvsadm-1.24 && make && make install**

**[root@vs1 ~]#ipvsadm --help**

**##或者使用yum安装：**

**[root@vs1 ~]#yum install ipvsadm -y**

**2、LVS-NAT模型的实现：**

(1)实现前提：

1)RIP和DIP必须在同一个IP网络, 且应该使用私网地址；

2)RS的网关要指向DIP；

3)请求报文和响应报文都必须经由Director转发；

4)VS必须是linux系统， RS可以是任意系统。

(2) RS1网卡和web服务的配置(RS2与RS1相同的配置)

RS1 IP:192.168.88.66

RS2 IP:192.168.88.77

RIP和DIP在同一网段，且两台RS服务器主机网关指向DIP：192.168.88.11

**[root@rs1 ~]#route add default gw 192.168.88.11**

**[root@rs1 ~]#yum install httpd -y**

**[root@rs1 ~]#echo “<h1>RS1</h1>” > /var/www/html/index.html**

(3) VS主机：核心转发和安装ipvsadm

**启动网卡间核心转发功能：**

**[root@vs1 ~]#sysctl -w net.ipv4.ip\_forward=1 ##也可以**

**[root@vs1 ~]#echo 1 > /proc/sys/net/ipv4/ip\_forward**

**[root@vs1 ~]#sysctl -p**

(4) VS主机：定义配置lvs-nat服务(此处采用rr算法)

**##定义ipvsadm负载均衡集群规则，并查看。**

**##此处定义DIP是以-s指定为rr算法进行轮询调度，-m指定模式为lvs-nat，配置命令如下：**

**[root@vs1 ~]#ipvsadm -A -t 192.168.122.11:80 -s rr**

**[root@vs1 ~]#ipvsadm -a -t 192.168.122.11:80 -r 192.168.88.66 -m**

**[root@vs1 ~]#ipvsadm -a -t 192.168.122.11:80 -r 192.168.88.77 -m**

**[root@vs1 ~]#ipvsadm -L -n**

(5) 测试两台RS的服务是否能正常访问

**[root@test ~]#ab -c 100 -n 100 http://192.168.122.11**

**2、LVS-DR模型的实现：**

(1)实现前提：

1)保证前端路由器将目标地址为 VIP 的报文通过 ARP 解析后送往 Director。修改内核参数：在 Realserver 上修改内核参数，限制arp通告及应答级别(最常用)，修改参数arp\_ignore和arp\_announce；

2)各RIP 必须与 DIP 在同一个物理网络中;

3)RS 的 RIP 可以使用私有地址，也可以使用公网地址，Realserver 不能将网关指向 DIP。

(2) RS服务器APR内核参数解读：

**####/proc/sys/net/ipv4/conf/all/arp\_annouce定义通告级别**

**0<——>默认级别，将本地的任何接口上的配置的地址都在网络中通告**

**1<——>尽量避免向本主机上的其他网卡进行网络通信，特殊情况下其他接口也可以**

**2<——>总是使用最佳网络地址接口(仅使用定义的网卡接口在同网络通信)**

**####/proc/sys/net/ipv4/conf/all/arp\_ignore定义响应级别(0-8九个级别)，响应时忽略方式**

**0<——>都全都响应**

**1<——>只对从本接口进入的请求响应，且本接口地址是个网络地址**

**注释：一般使用arp\_annouce=2，arp\_ignore=1**

(3) RS预配置脚本

**[root@rs1 ~]#vim rs.sh**

**#!/bin/bash**

**vip=192.168.122.11**

**mask='255.255.255.255'**

**case $1 in**

**start)**

**echo 1 > /proc/sys/net/ipv4/conf/all/arp\_ignore**

**echo 1 > /proc/sys/net/ipv4/conf/lo/arp\_ignore**

**echo 2 > /proc/sys/net/ipv4/conf/all/arp\_announce**

**echo 2 > /proc/sys/net/ipv4/conf/lo/arp\_announce**

**ifconfig lo:0 $vip netmask $mask broadcast $vip up**

**route add -host $vip dev lo:0;;**

**stop)**

**ifconfig lo:0 down**

**echo 0 > /proc/sys/net/ipv4/conf/all/arp\_ignore**

**echo 0 > /proc/sys/net/ipv4/conf/lo/arp\_ignore**

**echo 0 > /proc/sys/net/ipv4/conf/all/arp\_announce**

**echo 0 > /proc/sys/net/ipv4/conf/lo/arp\_announce;;**

**\*)**

**echo "Usage $(basename $0) start|stop"**

**exit 1;;**

**esac**

(4) VS的配置脚本(DR模型：-g参数)

**[root@vs1 ~]#vim vs.sh**

**#!/bin/bash**

**vip='192.168.122.11'**

**iface='eth1'**

**mask='255.255.255.255'**

**port='80'**

**rs1='192.168.88.66'**

**rs2='192.168.88.77'**

**scheduler='wrr'**

**type='-g' ##启用DR模型**

**case $1 in**

**start)**

**ifconfig $iface $vip netmask $mask broadcast $vip up**

**iptables -F**

**ipvsadm -A -t ${vip}:${port} -s $scheduler**

**ipvsadm -a -t ${vip}:${port} -r ${rs1} $type -w 1**

**ipvsadm -a -t ${vip}:${port} -r ${rs2} $type -w 1;;**

**stop)**

**ipvsadm -C**

**ifconfig $iface down;;**

**\*)**

**echo "Usage $(basename $0) start|stop"**

**exit 1;;**

**esac**

(5) 测试两台RS的服务是否能正常访问

**[root@test ~]#ab -c 100 -n 100 http://192.168.122.11**

**3、通过防火墙标记来定义LVS**

防火墙标记可以实现多个集群服务绑定为同一个，实现统一调度；将共享一组RS的集群服务统一进行定义FWM基于iptables的mangle表实现防护墙标记功能，定义标记做策略路由

(1) FWM定义集群的方式

**1)在director上netfilter的mangle表的PREROUTING定义用于"打标"的规则**

**Usage： iptables -t mangle -A PREROUTING -d $vip -p $protocol --dport $port -j MARK**

**--set-mark**

**2)基于FWM定义集群服务：**

**Usage： ipvsadm -A -f # -s scheduler**

(2) 示例

**[root@localhost~]# iptables -t mangle -A PREROUTING -d 192.168.122.11 -p tcp --dport 80**

**-j MARK --set-mark 5**

**[root@localhost~]# ipvsadm -A -f 5 -s rr**

**[root@localhost~]# ipvsadm -a -f 5 -r 192.168.88.66 -g**

**[root@localhost~]# ipvsadm -a -f 5 -r 192.168.88.77  -g**

**4、LVS持久连接功能：LVS persistence**

(1) LVS persistence功能

无论ipvs使用何种scheduler，其都能够实现在指定时间范围内始终将来自同一个ip地址的请求发往同一个RS；实现方式和lvs调度的十种算法无关，通过lvs持久连接模板(hash表)实现，当超过自定义的可持节连接时长候再根据LVS算法本身进行调度。ipvsadm命令中”-p”选项实现，在”-p”后不指定具体数字(单位:秒)，默认为300，到时候会自动延长2分钟，对于web本身就是15秒。

(2) 防火墙标记端口持久(PFWMC)

将两个或两个以上服务通过防火墙打标绑定在一起，这些服务的请求实现同时定向与同一台RS服务器，服务绑定同一RS

(3) 示例：

**##lvs-dr模式下以rr算法绑定http和https服务,将这两个服务标记为99**

**[root@localhost~]#iptables -t mangle -A PREROUTING -d 192.168.122.11 -p tcp --dport 80 -j MARK --set-mark 99**

**[root@localhost~]#iptables -t mangle -A PREROUTING -d 192.168.122.11 -p tcp --dport 443 -j MARK --set-mark 99**

**[root@localhost~]#ipvsadm -A -f 99 -s rr -p**

**[root@localhost~]#ipvsadm -a -f 99 -r 192.168.88.66 -g**

**[root@localhost~]#ipvsadm -a -f 99 -r 192.168.88.77 -g**

(4) 保存及重载规则：

**##保存：建议保存至/etc/sysconfig/ipvsadm**

**[root@localhost~]#ipvsadm-save > /etc/sysconfig/ipvsadm #或**

**[root@localhost~]#ipvsadm -S > /etc/sysconfig/ipvsadm**

**[root@localhost~]#service ipvsadm stop #停止ipvadm服务**

**##重载：**

**[root@localhost~]#ipvsadm-restore < /etc/sysconfig/ipvsadm #或**

**[root@localhost~]#ipvsadm -R < /etc/sysconfig/ipvsadm**

**[root@localhost~]#service ipvsadm start #启动ipvsadm服务**

**5、关于LVS单点故障的思考**

1、Director不可用，整个系统将不可用：

解决方案：高可用

keepalived——或——heartbeat/corosync

2、某RS不可用时，Director依然会调度请求至此RS；

    解决方案：对各RS的健康状态做检查，失败时禁用，成功时启用；

keepalived——或——heartbeat/corosync——或——ldirectord

3、检测方式：

        (a) 网络层检测；

        (b) 传输层检测，端口探测；

        (c) 应用层检测，请求某关键资源；

        ok --> failure

        failure --> ok