**LVS**

集群：通过高速的网络把一组计算机组合在一起,提供同一种特定的服务.

给用户的感觉是只有一台服务器.

任务调度是集群的核心技术.

集群分类:

高性能集群、高可用集群、负载均衡集群

LVS:章文嵩博士开发的软件,现已被内置到Linux内核中.

|web1服务器

client-----调度器--|web2服务器 共享存储

|web3服务器

前端调度器

中间应用服务器

后端共享存储

参考案例视频:LVS与nginx

LVS工作模式:

1)NAT模式

(数据从哪进，就从哪里出)数据量大时返回数据到调度器会成为瓶颈

2)DR直连路由模式

数据从调度器进入，进行负载均衡，web服务器直接返回数据给客户

3)Tunnel隧道模式

LVS调度算法:

轮询（Round Robin）

加权轮询（Weighted Round Robin）

最少连接（Least Connections）

加权最少连接（ Weighted Least Connections ）

源地址哈希（Source Hash）

基于局部性的最少链接

带复制的基于局部性最少链接

目标地址散列（Destination Hashing）

最短的期望的延迟

最少队列调度

【实验一：LVS-NAT实验】

【准备实验环境】

|-- web1

client------proxy------

|-- web2

client(IP地址192.168.4.10)

proxy主机作LVS调度器(IP地址:192.168.4.5+192.168.2.5)

web1做web服务器(IP地址:192.168.2.100)

web2做web服务器(IP地址:192.168.2.200)

web1和web2启动web服务(可以是nginx或者httpd或者tomcat)

#把web1和web2的4网段的网卡临时关闭.

[web1]

[root@web1 ~]# yum -y install httpd

[root@web1 ~]# echo "2.100" > /var/www/html/index.html

[root@web1 ~]# systemctl restart httpd

[root@web1 ~]# firewall-cmd --set-default-zone=trusted

[root@web1 ~]# setenforce 0

[web2]

[root@web2 ~]# yum -y install httpd

[root@web2 ~]# echo "2.200" > /var/www/html/index.html

[root@web2 ~]# systemctl restart httpd

[root@web2 ~]# firewall-cmd --set-default-zone=trusted

[root@web2 ~]# setenforce 0

[重要的步骤:给web1和web2配置网关]

[root@web1 ~]# nmcli connection modify ens33 \

ipv4.method manual ipv4.gateway 192.168.2.5

[root@web1 ~]# nmcli connection up ens33

[root@web1 ~]# ip route show #查看默认网关

#备注:网卡名称不能照抄,需要自己查看下2.100的网卡名称

[root@web2 ~]# nmcli connection modify ens33 \

ipv4.method manual ipv4.gateway 192.168.2.5

[root@web2 ~]# nmcli connection up ens33

[root@web2 ~]# ip route show #查看默认网关

#备注:网卡名称不能照抄,需要自己查看下2.200的网卡名称

【LVS调度器配置】

备注:LVS已经内置到Linux内核了,但是我们无法直接操控内核

所以需要安装一个软件ipvsadm,帮助我们间接去管理内核的LVS

人打命令写规则告诉ipvsadm,由ipvsadm再传递给内核的LVS

最终实际工作,承担调度任务的是LVS.

[root@proxy ~]# yum -y install ipvsadm

[root@proxy ~]# ipvsadm -A -t 192.168.4.5:80 -s rr

#备注:-A(add)创建虚拟服务器,

#VIP[集群服务器的IP]是192.168.4.5,给用户访问的IP

#-t是TCP协议,-u是UDP协议

#-s(scheduler)调度算法:rr|wrr|lc|wlc|sh|dh|lblc|lblcr|sed|nq

[root@proxy ~]# ipvsadm -Ln #查看集群配置

[root@proxy ~]# ipvsadm -a \

-t 192.168.4.5:80 -r 192.168.2.100 -m -w 1

[root@proxy ~]# ipvsadm -a \

-t 192.168.4.5:80 -r 192.168.2.200 -m

#备注:往虚拟服务器中添加真正的后端web服务器(real server)

-a(add)添加真实服务器,

-t 192.168.4.5:80代表往192.168.4.5这个集群中添加real server

-r(real)真正后端服务器的IP是192.168.2.100和192.168.2.200

-m代表LVS工作模式为NAT模式(-g是DR模式,-i是Tun隧道模式)

-w(weight)设置服务器的权重

[root@proxy ~]# ipvsadm -Ln #查看集群配置

【重要:LVS代理服务器必须开启路由转发功能】

[root@proxy ~]# cat /proc/sys/net/ipv4/ip\_forward

#查看路由功能是否开启0,代表关闭,1代表开启

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

#临时打开(重启后失效)

[root@proxy ~]# vim /etc/sysctl.conf

手动加入如下内容,可以永久开启路由功能

net.ipv4.ip\_forward = 1

保存退出后,使用sysctl -p刷新

[root@proxy ~]# sysctl -p

#备注:LVS工作在Linux内核,只要Linux启动了LVS就启动了!

[root@proxy ~]# firewall-cmd --set-default-zone=trusted

[root@proxy ~]# setenforce 0

使用真实主机测试集群效果:

firefox http://192.168.4.5

[Ctrl+F5]刷新

常见错误:

1.proxy: curl http://192.168.2.100/200

2.proxy: cat /proc/sys/net/ipv4/ip\_forward

3.proxy: ipvsadm -Ln

检查内容, 连接数量

4.web1和web2

:ip route show

5.所有主机的防火墙和SELinux

6.检查web1和web2不能有网卡配置了4.0网段的IP,如果有需要down掉

【ipvsadm常用其他命令】:

[root@proxy ~]# ipvsadm -e -t 192.168.4.5:80 \

-r 192.168.2.100 -g -w 2

#-e(edit)修改后端服务器配置

#-t 192.168.4.5:80(需要修改的哪个集群里面的真实服务器)

#-r 192.168.2.100(需要修改的是4.5集群下192.168.2.100的配置)

#-g把工作模式改成DR模式(-m是NAT模式,-g是DR模式,-i是隧道模式)

#-w修改权重为2.(如果调度算法是RR则这里的权重无效)

[root@proxy ~]# ipvsadm -d -t 192.168.4.5:80 -r 192.168.2.100

#删除192.168.4.5集群中的某一台后端服务器

#这里是删除192.168.2.100服务器.

#d(delete)

[root@proxy ~]# ipvsadm -A -t 192.168.4.5:3306 -s wrr

#-A创建虚拟服务器集群

[root@proxy ~]# ipvsadm -E -t 192.168.4.5:80 -s wlc

#-E(edit)修改虚拟机服务器集群属性

#-t 192.168.4.5:80(需要修改哪个集群的属性)

#-s wlc修改算法为加权最小连接算法

[root@proxy ~]# ipvsadm -D -t 192.168.4.5:80

#删除某一个集群

[root@proxy ~]# ipvsadm -C

#-C(clear)清空所有

【实验二：LVS-DR实验】

【环境准备】

proxy主机IP地址:192.168.4.5,(#192.168.2.5可有可无)

web1主机的IP:192.168.4.100,(#192.168.2.100可有可无)

web2主机的IP:192.168.4.200,(#192.168.2.200可有可无)

概念:

VIP:调度器给用户访问的那个IP(集群IP)

DIP:调度器连接后端真实服务器的IP

RIP:后端真实服务器的IP

CIP:客户端IP

4.100 VIP(4.15)

4.10 4.15 4.5 | web1(RIP) 伪装VIP

客户端----->VIP(调度器)DIP----

| web2(RIP) 伪装VIP

4.200 VIP(4.15)

【重点】:VIP必须配置在网卡的虚拟接口(辅助接口)上

DIP(director IP)必须配置在网卡的主接口上

【给LVS主机配置VIP和DIP】

1)配置DIP(网卡名称不能照抄)

[root@proxy ~]# nmcli conn modify ens34 ipv4.method manual ipv4.addr 192.168.4.5/24 autoconnect yes

[root@proxy ~]# nmcli conn up ens34

2)配置VIP(给有4.5这个IP的网卡配置VIP,看清自己的网卡名称)

#注意:千万不能照抄网卡名称!!!(拷贝有4.5这个IP的网卡)

[root@proxy ~]# cd /etc/sysconfig/network-scripts/

[root@proxy network-scripts]# cp ifcfg-ens34{,:0}

[root@proxy network-scripts]# vim ifcfg-ens34:0

修改文件,最终全文内容如下:(网卡名称不能照抄)

TYPE=Ethernet #类型为以太网

BOOTPROTO=none #如何配置IP[none手动]或[dhcp]

NAME=ens34:0 #网卡名称(con-name)

DEVICE=ens34:0 #设备名称(ifname)

ONBOOT=yes #开机是否自动激活

IPADDR=192.168.4.15 #网卡IP地址

PREFIX=24 #子网掩码

[root@proxy network-scripts]# systemctl restart network

[root@proxy network-scripts]# yum -y install net-tools

[root@proxy network-scripts]# ifconfig

如果配置正确,看不到新配置的IP可以关闭NetworkManager后重启

[root@proxy network-scripts]# systemctl stop NetworkManager

[root@proxy network-scripts]# systemctl restart network

【配置web1】

1)配置正常RIP#不能照抄网卡名称

[root@web1 ~]# nmcli connection modify ens33 ipv4.method manual ipv4.addresses 192.168.4.100/24 autoconnect yes

[root@web1 ~]# nmcli connection up ens33

2)给web1伪装VIP(必须伪装到lo网卡)

[root@web1 ~]# cd /etc/sysconfig/network-scripts/

[root@web1 network-scripts]# cp ifcfg-lo{,:0}

[root@web1 network-scripts]# vim ifcfg-lo:0

修改内容如下:

DEVICE=lo:0 #设备名称

IPADDR=192.168.4.15 #IP地址

NETMASK=255.255.255.255 #子网掩码

NETWORK=192.168.4.15 #网络地址(网络位)

BROADCAST=192.168.4.15 #广播地址

ONBOOT=yes #开机是否自动激活

NAME=lo:0 #网卡名称

[root@web1 network-scripts]# vim /etc/sysctl.conf

手动添加如下内容(防止地址冲突):

net.ipv4.conf.all.arp\_ignore = 1

net.ipv4.conf.lo.arp\_ignore = 1

net.ipv4.conf.lo.arp\_announce = 2

net.ipv4.conf.all.arp\_announce = 2

[root@web1 network-scripts]# sysctl -p

[root@web1 network-scripts]# systemctl restart network

[root@web1 network-scripts]# ifconfig

如果配置正确,看不到新配置的IP可以关闭NetworkManager后重启

[root@we1 network-scripts]# systemctl stop NetworkManager

[root@web1 network-scripts]# systemctl restart network

备注:

arp\_ignore是其他主机发送ARP广播时web1和web2不回应自己是4.15.

arp\_announce是本机与其他主机通信时,不主动广播自己是4.15.

【配置web2】

1)配置正常RIP#不能照抄网卡名称

[root@web2 ~]# nmcli connection modify ens33 ipv4.method manual ipv4.addresses 192.168.4.200/24 autoconnect yes

[root@web2 ~]# nmcli connection up ens33

2)给web1伪装VIP(必须伪装到lo网卡)

[root@web2 ~]# cd /etc/sysconfig/network-scripts/

[root@web2 network-scripts]# cp ifcfg-lo{,:0}

[root@web2 network-scripts]# vim ifcfg-lo:0

修改内容如下:

DEVICE=lo:0 #设备名称

IPADDR=192.168.4.15 #IP地址

NETMASK=255.255.255.255 #子网掩码

NETWORK=192.168.4.15 #网络地址(网络位)

BROADCAST=192.168.4.15 #广播地址

ONBOOT=yes #开机是否自动激活

NAME=lo:0 #网卡名称

[root@web2 network-scripts]# vim /etc/sysctl.conf

手动添加如下内容(防止地址冲突):

net.ipv4.conf.all.arp\_ignore = 1

net.ipv4.conf.lo.arp\_ignore = 1

net.ipv4.conf.lo.arp\_announce = 2

net.ipv4.conf.all.arp\_announce = 2

[root@web2 network-scripts]# sysctl -p #刷新参数

[root@web2 network-scripts]# systemctl restart network

[root@web2 network-scripts]# ifconfig

如果配置正确,看不到新配置的IP可以关闭NetworkManager后重启

[root@web2 network-scripts]# systemctl stop NetworkManager

[root@web2 network-scripts]# systemctl restart network

备注:

arp\_ignore是其他主机发送ARP广播时web1和web2不回应自己是4.15.

arp\_announce是本机与其他主机通信时,不主动广播自己是4.15.

[web1和web2配置web服务]

[root@web1 ~]# systemctl start httpd

[root@web1 ~]# echo "192.168.4.100" > /var/www/html/index.html

[root@web2 ~]# systemctl start httpd

[root@web2 ~]# echo "192.168.4.200" > /var/www/html/index.html

【给proxy配置LVS规则】

[root@proxy ~]# ipvsadm -C

[root@proxy ~]# ipvsadm -A -t 192.168.4.15:80 -s wrr

[root@proxy ~]# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.100:80

[root@proxy ~]# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.200:80

[root@proxy ~]# ipvsadm -Ln

客户端使用浏览器访问http://192.168.4.15

Ctrl+F5

扩展知识：默认LVS不带健康检查功能，需要自己手动编写动态检测脚本，实现该功能：(参考脚本如下，仅供参考)

[root@proxy ~]# vim check.sh

#!/bin/bash

VIP=192.168.4.15:80

RIP1=192.168.4.100

RIP2=192.168.4.200

while :

do

for IP in $RIP1 $RIP2

do

curl -s http://$IP &>/dev/vnull

if [ $? -eq 0 ];then

ipvsadm -Ln |grep -q $IP || ipvsadm -a -t $VIP -r $IP

else

ipvsadm -Ln |grep -q $IP && ipvsadm -d -t $VIP -r $IP

fi

done

sleep 1

done