**7\_Cluster01LVS简介LVS-NAT集群LVS-DR集群**

**一 集群**

**1.1 定义:**

一组通过高速网络互联的计算组,并以单一系统的模式加以管理

将很多服务器集中在一起,提供同一种服务,在客户端看来就像只有一个服务器

可以在付出较低成本的情况下获得在性能\可靠性\灵活性方面的相对较高的收益

**任务调度**是集群系统中的**核心技术**

**1.2 目的**

**提高性能**:如计算密集型应用,如:天气预报\核实验模拟

**降低成本**:相对百元美元级的超级计算机,价格便宜

**提高可扩展性**:只要增加集群节点即可

**增强可靠性**:多个节点完成相同功能,避免单点失败

**1.3 集群分类**

**高性能计算集群(HPC)**:通过以集群开发的并行应用程序,解决复杂的科学问题

**负载均衡(LB)集群**:客户端负载在计算机集群中尽可能平均分摊

**高可用(HA)集群**:避免单点故障,当一个系统发生故障时,可以快速迁移

**二 LVS**

**2.1 Linux Virtual Server**(Linux默认安装并启动,在内核中)

实现高可用的\可伸缩的web mail cache 和media 等网络服务.

最终目标是利用Linux和LVS集群软件实现一个高可用\高性能\低成本的服务器应用集群.

**2.2 LVS集群组层**

**前端**:负载均衡层 由一台或多台负载调度器构成

**中间**:服务器群组层 由一组实际运行应用的服务器组成

**底端**:数据共享存储层 提供共享存储空间的存储区域

**2.3 LVS术语**

**director server**:调度服务器 将负载分发到real server的服务器

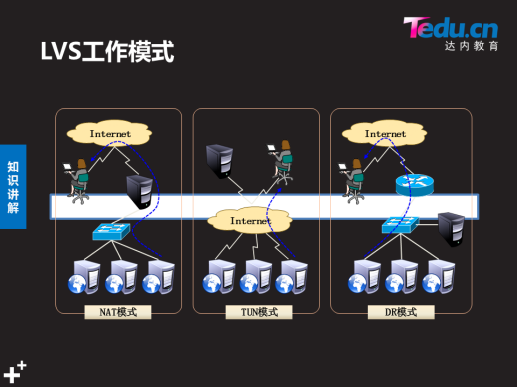
**real server**:真实服务器 真正提供应用服务的服务器

**vip**: virtual ip ds或rs公布给用户访问的虚拟IP地址

**rip**: real ip 集群节点上使用的IP地址(rs使用的IP)

**dip**: director ip 调度器(ds)连接节点服务器(rs)的IP地址

**2.4 LVS工作模式(用路由器的原理来理解LVS)** 电脑主机+LVS=软路由



2.4.1 LVS/NAT

通过网络地址转换实现的虚拟服务器

数据包流向 clinet proxy (NAT转换) web1

数据包 s:4.10 s:4.10 s:4.10 s:4.10

d:4.5 d:4.5 d:2.100 d:2.100

数据包流向 clinet proxy (NAT转换) web1

数据包 s:4.5 s:4.5 s:2.100 s:2.100

d:4.10 d:4.10 d:4.10 d:4.10

**LVS/NAT模式下web服务器必须配置网关**

大并发访问时,调度器的性能成为瓶颈

2.4.2 VS/DR

直接使用路由技术实现虚拟服务器

节点服务器需要配置vip,注意mac地址广播

2.4.3 VS/TUN

通过隧道方式实现虚拟服务器

**2.5 负载均衡调度算法**

**2.5.1 轮询(round robin)[rr]**:将客户端请求**平均分发**到real server

**2.5.2 加权轮训(weighted round robin)[wrr]**:根据real server**权重值进行轮询**调度

**2.5.3 最少连接(least connections)[lc]**:选择**连接数最少**的服务器

**2.5.4 加权最少连接[wlc]**:根据real server权重值,选择连接数最少的服务器

2.5.5源地址散列(source hashing)[sh]:将请求的目标地址作为散列键(hash key)从静态分配的散列表中找出对应的服务器,效果同nginx的ip hash

2.5.6 其他调度算法:

基于局部性的最少链接 最短的期望延迟

带复制的基于局部性的最少链接 最少队列调度

目标地址散列(destination hashing)

**三 LVS-NAT集群**

**3.1 安装前准备**

LVS的IP负载均衡技术是通过IPVS模块实现的

IPVS模块已经成为Linux组成部分

**3.2 安装ipvsadm**

rpm方式 或 yum方式

**3.3 ipvsadm用法**

3.3.1 创建虚拟服务器

**-A** 添加虚拟群集服务器

**-t** 设置集群地址(vip)

**-s** 指定负载调度算法

3.3.2 添加\删除服务器节点

**-a** 添加真实服务器

**-d** 删除真实服务器

**-r** 指定真实服务器(real server)的地址

**-m** 使用NAT模式;-g -i分别对应DR TUN模式(默认为-g)

**-w** 为节点服务器设置权重,默认为1

3.3.3 查看IPVS

**ipvsadm -Ln**

**四 案例：ipvsadm命令用法**

**4.1 问题**

准备一台Linux服务器，安装ipvsadm软件包，练习使用ipvsadm命令，实现如下功能：

使用命令添加基于TCP一些的集群服务

在集群中添加若干台后端真实服务器

实现同一客户端访问，调度器分配固定服务器

会使用ipvsadm实现规则的增、删、改

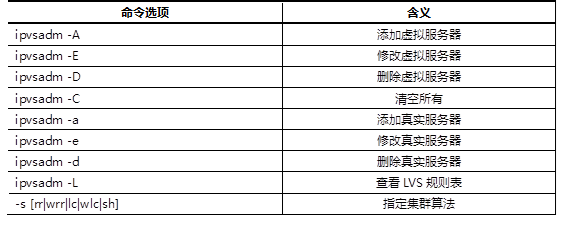
保存ipvsadm规则

**4.2 方案**

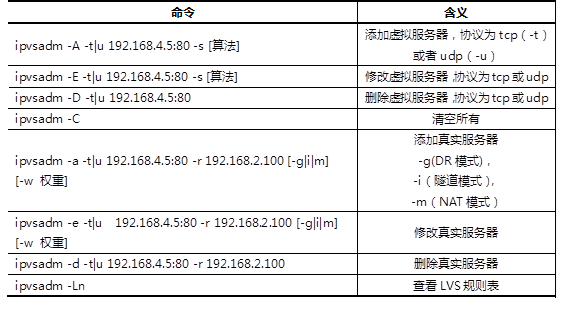
安装ipvsadm软件包，关于ipvsadm的用法可以参考man ipvsadm资料。

常用ipvsadm命令语法格式如表-1及表-2所示。

表－1 ipvsadm命令选项



表－2 ipvsadm语法案例



**4.3 步骤**

**步骤一：使用命令增、删、改LVS集群规则**

**1）创建LVS虚拟集群服务器（算法为加权轮询：wrr）**

proxy ~]# yum -y install ipvsadm #ipvsadm用于沟通内核中的LVS

proxy ~]# ipvsadm **-A** -t 192.168.4.5:80 **-s** **wrr** #-t表示TCP协议

proxy ~]# ipvsadm **-Ln** #查看IPVS

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.5:80 wrr

**2）为集群添加若干real server**

proxy ~]# ipvsadm **-a** -t 192.168.4.5:80 **-r** 192.168.2.100 -w 1

# -r 指定real server,-w 指定权重

proxy ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.5:80 wrr

-> 192.168.2.100:80 router 1 0 0

proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.200 -w 2

proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.201 -m -w 3

proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.202 -m -w 4

**3）修改集群服务器设置(修改调度器算法，将加权轮询修改为轮询)**

proxy ~]# ipvsadm **-E** -t 192.168.4.5:80 **-s rr**

proxy ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.5:80 rr

-> 192.168.2.100:80 router 1 0 0

-> 192.168.2.200:80 masq 2 0 0

-> 192.168.2.201:80 masq 2 0 0

-> 192.168.2.202:80 masq 1 0 0

**4）修改read server（使用-g选项，将模式改为DR模式）**

proxy ~]# ipvsadm **-e** -t 192.168.4.5:80 -r 192.168.2.202 **-g**

**5）查看LVS状态**

proxy ~]# ipvsadm -Ln

**6）创建另一个集群**（算法为最少连接算法；使用-m选项，设置工作模式为NAT模式）

proxy ~]# ipvsadm -A -t 192.168.4.5:3306 **-s lc**

proxy ~]# ipvsadm -a -t 192.168.4.5:3306 -r 192.168.2.100 **-m**

proxy ~]# ipvsadm -a -t 192.168.4.5:3306 -r 192.168.2.200 -m

**7）永久保存所有规则**(-n以数字方式显示ip和端口)

proxy ~]# **ipvsadm-save -n > /etc/sysconfig/ipvsadm**

**8）清空所有规则**

proxy ~]# ipvsadm -C #清空后所有都没有了

**五 案例：部署LVS-NAT集群**

**5.1 问题**

使用LVS实现NAT模式的集群调度服务器，为用户提供Web服务：

集群对外公网IP地址为192.168.4.5

调度器内网IP地址为192.168.2.5

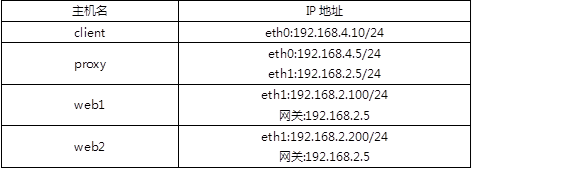
真实Web服务器地址分别为192.168.2.100、192.168.2.200

使用加权轮询调度算法，真实服务器权重分别为1和2

**5.2 方案**

实验拓扑结构主机配置细节如表-3所示。

表-3



使用4台虚拟机，1台作为Director调度器、2台作为Real Server、1台客户端，拓扑结构如图-1所示，**注意：web1和web2必须配置网关地址。**



图-1

**route -n 查看网关**

**5.3 步骤**

**步骤一：配置基础环境**

1）设置Web服务器（以web1为例）

web1 ~]# yum -y install httpd

web1 ~]# echo "192.168.2.100" > /var/www/html/index.html

2）启动Web服务器软件

web1 ~]# systemctl restart httpd

3)关闭防火墙与SELinux

web1 ~]# systmctl stop firewalld

web1 ~]# setenforce 0

**步骤二：部署LVS-NAT模式调度器**

1)确认**调度器的路由转发功能**(如果已经开启，可以忽略)

proxy ~]# **echo 1 > /proc/sys/net/ipv4/ip\_forward #临时设置**

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

1

proxy ~]# **echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf**

#修改配置文件，**永久设置**调度器的路由转发功能,注意空格

2）创建集群服务器

proxy ~]# yum -y install ipvsadm

proxy ~]# ipvsadm -A -t 192.168.4.5:80 -s wrr

3）添加真实服务器

proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.100 -w 1 -m

proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.200 -w 1 -m

4）查看规则列表，并保存规则

proxy ~]# ipvsadm -Ln

proxy ~]# ipvsadm-save -n > /etc/sysconfig/ipvsadm

**步骤三：客户端测试**

客户端使用curl命令反复连接http://192.168.4.5，查看访问的页面是否会轮询到不同的后端真实服务器。

**LVS-NAT不带健康检查,若某台服务器down了,调度器仍然会调度该服务器**

**六 案例：部署LVS-DR集群**

**6.1 问题**

使用LVS实现DR模式的集群调度服务器，为用户提供Web服务：

客户端IP地址为192.168.4.10

LVS调度器VIP地址为192.168.4.15

LVS调度器DIP地址设置为192.168.4.5

真实Web服务器地址分别为192.168.4.100、192.168.4.200

使用加权轮询调度算法，web1的权重为1，web2的权重为2

说明： CIP是客户端的IP地址；

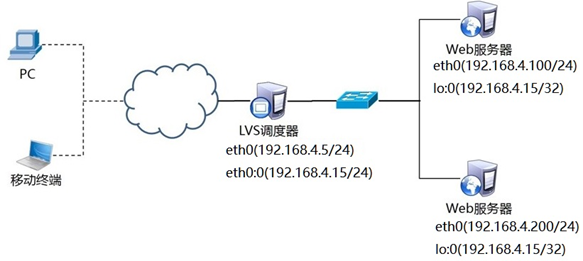
VIP是对客户端提供服务的IP地址；

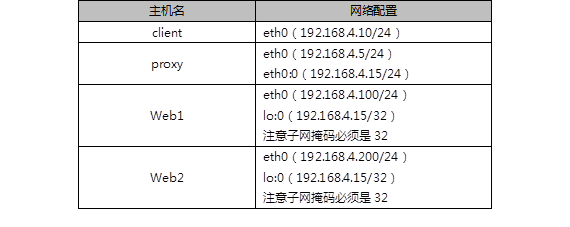
RIP是后端服务器的真实IP地址；

DIP是调度器与后端服务器通信的IP地址（VIP必须配置在虚拟接口）。

**6.2 方案**

使用4台虚拟机，1台作为客户端、1台作为Director调度器、2台作为Real Server，拓扑结构如图所示。实验拓扑结构主机配置细节如表所示。





**6.3 步骤**

实现此案例需要按照如下步骤进行。

说明： CIP是客户端的IP地址； VIP是对客户端提供服务的IP地址；

RIP是后端服务器的真实IP地址；

DIP是调度器与后端服务器通信的IP地址（VIP必须配置在虚拟接口）。

**步骤一：配置实验网络环境**

**1）设置Proxy代理服务器的VIP和DIP**

**注意：为了防止冲突，VIP必须要配置在网卡的虚拟接口！！！**

proxy ~]# cd /etc/sysconfig/network-scripts/

proxy ~]# cp ifcfg-eth0{,:0}

#复制ifcfg-eth0文件,命名为ifcfg-eth0:0,ifcfg-eth0:0为虚拟网卡

proxy ~]# vim ifcfg-eth0 #设置DIP

TYPE=Ethernet

BOOTPROTO=none

NAME=eth0

DEVICE=eth0

ONBOOT=yes

IPADDR=192.168.4.5

PREFIX=24

proxy ~]# vim ifcfg-eth0:0 #设置VIP

**TYPE=Ethernet**

BOOTPROTO=none #不自动获取IP

DEFROUTE=yes

**NAME=eth0:0**

**DEVICE=eth0:0**

ONBOOT=yes #开机是否激活

**IPADDR=192.168.4.15**

**PREFIX=24** #子掩长度

proxy ~]# systemctl restart network

2）设置Web1服务器网络参数RIP

web1 ~]# nmcli connection modify eth0 ipv4.method manual \

ipv4.addresses 192.168.4.100/24 connection.autoconnect yes

web1 ~]# nmcli connection up eth0

接下来给web1配置VIP地址。

注意：这里的子网掩码必须是32（也就是全255），网络地址与IP地址一样，广播地址与IP地址也一样。

web1 ~]# cd /etc/sysconfig/network-scripts/

web1 ~]# cp ifcfg-lo{,:0}

web1 ~]# vim ifcfg-lo:0

**DEVICE=lo:0**

**IPADDR=192.168.4.15**

**NETMASK=255.255.255.255**

**NETWORK=192.168.4.15 #网络位,在192.168.4.15网段**

**BROADCAST=192.168.4.15 #用于向client(192.168.4.10)传输数据时广播**

ONBOOT=yes

**NAME=lo:0**

防止地址冲突的问题：

这里因为web1也配置与代理一样的VIP地址，默认肯定会出现地址冲突；

sysctl.conf文件写入这下面四行的主要目的就是访问192.168.4.15的数据包，只有调度器会响应，其他主机都不做任何响应，这样防止地址冲突的问题。

web1 ~]# vim /etc/sysctl.conf

#手动写入如下4行内容

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

**#回答广播或对外广播时,默认0有啥说啥;1选择性说;2啥都不说;注意空格**

#当有arp广播问谁是192.168.4.15时，本机忽略该ARP广播，不做任何回应

#本机不要向外宣告自己的lo回环地址是192.168.4.15

web1 ~]# sysctl -p #刷新以上设置

重启网络服务，设置防火墙与SELinux

web1 ~]# systemctl restart network

web1 ~]# ifconfig

**常见错误：如果重启网络后未正确配置lo:0，有可能是NetworkManager和network服务有冲突，关闭NetworkManager后重启network即可。**

**web1 ~]# systemctl stop NetworkManager**

**web1 ~]# systemctl restart network**

3）设置Web2服务器网络参数RIP

[root@web2 ~]# nmcli connection modify eth0 ipv4.method manual \

ipv4.addresses 192.168.4.200/24 connection.autoconnect yes

[root@web2 ~]# nmcli connection up eth0

接下来给web2配置VIP地址

注意：这里的子网掩码必须是32（也就是全255），网络地址与IP地址一样，广播地址与IP地址也一样。

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

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

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

DEVICE=lo:0

IPADDR=192.168.4.15

NETMASK=255.255.255.255

NETWORK=192.168.4.15

BROADCAST=192.168.4.15

ONBOOT=yes

NAME=lo:0

防止地址冲突的问题：

这里因为web1也配置与代理一样的VIP地址，默认肯定会出现地址冲突；

sysctl.conf文件写入这下面四行的主要目的就是访问192.168.4.15的数据包，只有调度器会响应，其他主机都不做任何响应，这样防止地址冲突的问题。

[root@web2 ~]# vim /etc/sysctl.conf

#手动写入如下4行内容

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

#当有arp广播问谁是192.168.4.15时，本机忽略该ARP广播，不做任何回应

#本机不要向外宣告自己的lo回环地址是192.168.4.15

[root@web2 ~]# sysctl -p

重启网络服务，设置防火墙与SELinux

[root@web2 ~]# systemctl restart network

[root@web2 ~]# ifconfig

常见错误：如果重启网络后未正确配置lo:0，有可能是NetworkManager和network服务有冲突，关闭NetworkManager后重启network即可。（非必须的操作）

web1 ~]# systemctl stop NetworkManager

web1 ~]# systemctl restart network

**步骤二：proxy调度器安装软件并部署LVS-DR模式调度器**

1）安装软件（如果已经安装，此步骤可以忽略）

proxy ~]# yum -y install ipvsadm

2）清理之前实验的规则，创建新的集群服务器规则

proxy ~]# ipvsadm -C #清空所有规则

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

3）添加真实服务器(-g参数设置LVS工作模式为DR模式，-w设置权重)

proxy ~]# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.100 -g -w 1

proxy ~]# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.200 -g -w 1

4）查看规则列表，并保存规则

proxy ~]# ipvsadm -Ln

TCP 192.168.4.15:80 wrr

-> 192.168.4.100:80 Route 1 0 0

-> 192.168.4.200:80 Route 2 0 0

**步骤三：客户端测试**

客户端使用curl命令反复连接http://192.168.4.15，查看访问的页面是否会轮询到不同的后端真实服务器。

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)

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 [ $? -eq 0 ];then #$IP的主机的web服务正常提供服务

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

#若$IP的主机**是**(不是)LVS虚拟集群服务器中的真实主机,则**不将**(将)该主机添加到LVS虚拟主机集群中.

else #$IP的主机的web服务不能正常提供服务

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

#若$IP的主机**是**(不是)LVS虚拟集群服务器中的真实主机,则**将**(不将)该主机从LVS虚拟集群服务器中删除.

fi

done

sleep 1

done

可使用cron计划任务或加&号放入后台执行.