[集群简介 1](#_Toc1685054)

[LVS 2](#_Toc1685055)

[LVS工作模式 4](#_Toc1685056)

[负载均衡调度算法： 4](#_Toc1685057)

[搭建NAT模式的LVS集群 5](#_Toc1685058)

[准备LAMP分离环境论坛服务 5](#_Toc1685059)

[ipvsadm用法： 7](#_Toc1685060)

[配置LVS-NAT模式 8](#_Toc1685061)

[LVS-DR模式： 10](#_Toc1685062)

[ipvsadm服务 13](#_Toc1685063)

[HAProxy服务器 17](#_Toc1685064)

[HAProxy工作模式： 18](#_Toc1685065)

[HTTP协议解析 18](#_Toc1685066)

[hapoxy配置（负载均衡集群） 21](#_Toc1685067)

# 集群简介

什么是集群：

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

将很多服务器集中起来一起,提供同一种服务,在客户端看来就象是只有一个服务器

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

**集群目的：**

提高性能：如计算密集型应用,如:天气预报、核试验模拟

降低成本：相对百万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

重点：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

集群分类： \*

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

LB：负载均衡集群，客户端负载在计算机集群中尽可能平均分摊 \*

HA：高可用集群，避免单点故障,当一个系统发生故障时,可以快速迁移 \*

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

负载均衡集群调度器：nginx/lvs/haporxy/F5

# LVS

LVS项目介绍：

Linux 虚拟服务器(LVS)是章文嵩在国防科技大学就读博士期间创建的

LVS可以实现高可用的、可伸缩的Web、Mail、Cache和Media等网络服务

最终目标是利用Linux操作系统和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

LVS集群组成

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服务的服务器组成

底端：数据共享存储层

提供共享存储空间的存储区域

LVS术语：

Director Server:调度服务器 【lvs】

将负载分发到Real Server的服务器

Real Server:真实服务器 【web】

真正提供应用服务的服务器

VIP:虚拟IP地址

公布给用户访问的虚拟IP地址

RIP:真实IP地址

集群节点上使用的IP地址

DIP:调度器连接节点服务器的IP地址

## LVS工作模式

NAT：

通过网络地址转换实现的虚拟服务器

大并发访问时,调度器的性能成为瓶颈

DR： 常用

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

TUN：

通过隧道方式实现虚拟服务器

Full nat:大规模部署

## 负载均衡调度算法：

LVS目前实现了10种调度算法

常用调度算法有4种

1、轮询(Round Robin)：将客户端请求平均分发到Real Server

2、加权轮询(Weighted Round Robin)：根据Real Server权重值进行轮询调度

3、最少连接(Least Connections)：选择连接数最少的服务器

4、加权最少连接( Weighted Least Connections )：根据Real Server权重值,选择连接数最少的服务器

5、源地址散列(Source Hashing)：根据请求的目标IP地址,作为散列键(Hash Key)从静态分配的散列表找出对应的服务器

其他调度算法：6、基于局部性的最少链接lblc

7、带复制的基于局部性最少链接lblcr

8、目标地址散列(Destination Hashing)

9、最短的期望的延迟sed

10、最少队列调度nq

## 搭建NAT模式的LVS集群

### 准备LAMP分离环境论坛服务

服务器分类：

（1）应用服务器：web服务器。存程序源码，如php、python、jsp

（2）数据库服务器：mysql

（3）文件服务器：存静态文件

一，准备四台虚拟机，全新的,免密登录

vh05.tedu.cn 192.168.4.5 mysql

vh06.tedu.cn 192.168.4.6 web

vh07.tedu.cn 192.168.4.7 web

vh08.tedu.cn 192.168.4.8 lvs

二，安装web服务器

1.在vh06/vh07上安装软件包

[root@vh06 ~]# yum -y install httpd php php-mysql

[root@vh06 ~]# systemctl restart httpd

[root@vh06 ~]# systemctl enable httpd

三，配置数据库服务器，在vh05上

[root@vh05 ~]# yum -y install mariadb-server

[root@vh05 ~]# systemctl restart mariadb

[root@vh05 ~]# systemctl enable mariadb

[root@vh05 ~]# mysql

MariaDB [(none)]> grant all on \*.\* to 'admin'@"%" identified by "tedu.cn";

四，在vh06上安装Discuz

[root@room9pc01 ~]# scp -r 888888/项目/Discuz\_X3.3\_SC\_UTF8.zip 192.168.4.6:/root

[root@vh06 ~]# unzip Discuz\_X3.3\_SC\_UTF8.zip

[root@vh06 ~]# cp -r upload/ /var/www/html/bbs

[root@vh06 ~]# chown -R apache.apache /var/www/html/bbs

[root@room9pc01 ~]# firefox http://192.168.4.6/bbs &

同意---->下一步----> 数据库服务器:192.168.4.5---->数据库用户名: admin----->数据库密码:tedu.cn

管理员账号:admin--->密码随便（论坛帐号）--->下一步

登录时，输入的是论坛的密码

管理中心：输入的是授权的密码

模块管理中心：输入的是论坛帐号密码---->论坛

[root@vh05 ~]# mysql

MariaDB [(none)]> use ultrax

MariaDB [ultrax]> show tables;

MariaDB [ultrax]> desc pre\_common\_member;

MariaDB [ultrax]> select username,email from pre\_common\_member;

+----------+------------------+

| username | email |

+----------+------------------+

| admin | admin@admin.com |

| haha | 123456789@qq.com |

+----------+------------------+

MariaDB [ultrax]> desc pre\_forum\_post;

MariaDB [ultrax]> select author,subject,message from pre\_forum\_post;

+--------+----------------+--------------------+

| author | subject | message |

+--------+----------------+--------------------+

| admin | hfdbvgewtkhtrj | wml/hahah/zuimei

|

+--------+----------------+--------------------+

五、在vh07上配置

把06上的bbs打包，解压缩到07的/var/www/html/

访问192.168.4.7/bbs

可以看到和vh06同样的界面

因为都是用的同一个数据库

### ipvsadm用法：

-Ln 查看命令

-A 添加虚拟服务器

-t tcp协议，设置群集地址

-s 指定负载调度算法

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

-w 为节点服务器设置权重,默认为1

### 配置LVS-NAT模式

1、新建虚拟机vh08.tedu.cn 192.168.4.8

2、主机角色：

vh05.tedu.cn 192.168.4.5 mysql

vh06.tedu.cn 192.168.4.6 web

vh07.tedu.cn 192.168.4.7 web

vh08.tedu.cn 192.168.4.8 lvs调度器-nat模式

物理主机 客户端

3、基于上午的环境，做一些改进

（1）在vh06和vh07 web服务器上把网关指向192.168.4.8

# nmtui 添加网关

# ifdown eth0; ifup eth0

【[root@vh06 html]# nmcli connection modify eth0 ipv4.method manual ipv4.gateway 192.168.4.8 connection.autoconnect yes】

（2）在vh08的eth2上配置额外的外网地址201.1.1.8/24

[root@vh08 ~]# nmtui

[root@vh08 ~]# systemctl restart NetworkManager

【[root@vh08 ~]# nmcli connection add con-name eth2 ifname eth2 type ethernet

[root@vh08 ~]# nmcli connection modify eth2 ipv4.method manual ipv4.addresses 201.1.1.8/24 connection.autoconnect yes

[root@vh08 ~]# nmcli connection up eth2

（3）打开vh08的路由转发功能(RHEL7默认已打开，以前版本需要做以下配置)

[root@vh08 ~]# sysctl -a 查看所有的内核参数

[root@vh08 ~]# sysctl -a | grep ip\_forward

[root@vh08 ~]# vim /etc/sysctl.conf

net.ipv4.ip\_forward = 1

[root@vh08 ~]# sysctl -p

net.ipv4.ip\_forward = 1 让内核参数生效

4、配置LVS

（1）在vh08上安装

[root@vh08 ~]# yum -y install ipvsadm

（2）创建虚拟服务器

[root@vh08 ~]# ipvsadm -A -t 201.1.1.8:80 -s rr //不加-s选项默认是wlc -A add 添加

（3）将real server添加到虚拟服务器中（向VIP中加入RIP）

创建虚拟服务器，使用tcp的80端口，调度算法是rr

**[root@vh08 ~]# ipvsadm -a -t 201.1.1.8:80 -r 192.168.4.6 -m -w 2**

**[root@vh08 ~]# ipvsadm -a -t 201.1.1.8:80 -r 192.168.4.7 -m**

（4）查看规则

[root@vh08 ~]# ipvsadm -Ln

（5）验证。给两台web服务器配置不同的页面

在vh06和vh07上做一个网页

[root@vh06 html]# vim index.html // vim中ctrl+n 补全上一次输入的同样开头的单词

<html><h1><<font color=red>我想和你一起吃火锅</font></h1></html>

[root@vh06 html]# systemctl restart httpd

[root@vh07 html]# vim index.html

<html><h1><font color=yellow>你是屎吗？黄色的！</font></h1></html>

[root@vh07 html]# systemctl restart httpd

[root@room8pc16 ~]# firefox http://201.1.1.8/

网页上ctrl+F5 刷新，刷新页面时，按ctrl+f5重新连接服务器

!!!这个轮询是普通轮询，只有加权轮询才会检查 -w的权重。

（6）修改调度算法为WRR

**[root@vh08 ~]# ipvsadm -E -t 201.1.1.8:80 -s wrr**

**[root@vh08 ~]# ipvsadm -Ln**

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 201.1.1.8:80 wrr

-> 192.168.4.6:80 Masq 2 0 1

-> 192.168.4.7:80 Masq 1 0 1

从客户端访问两个网址时，以2:1权重刷新

（7）删除

从规则中移除real server

[root@vh08 ~]# ipvsadm -d -t 201.1.1.8:80 -r 192.168.4.6

[root@vh08 ~]# ipvsadm -D -t 201.1.1.8:80 //删除整个调度器

## LVS-DR模式：

一、拓扑：LVS调度器只有一个ip地址，它和real server在同一网络

二、客户机要把数据发给VIP。VIP需要出现在每台服务器上（调度器、web服务器）

三、为了地址不冲突，需要把VIP配置在调度器eth0上，把VIP配置在web服务器的lo上

四、每台主机都有VIP，客户端发来的请求，只有LVS回应、为了实现这一点，需要改web服务器的内核参数

五、实施

1、清理LVS－NAT的配置

删除lvs规则

[root@vh08 bin]# ipvsadm -D -t 201.1.1.8:80

删除eth2的地址

[root@vh08 bin]# nmtui 删除地址，将地址获取改为自动

[root@vh08 bin]# ifdown eth2

【[root@vh08 ~]# nmcli connection delete eth2】

2、在vh08上配置VIP

[root@vh08 bin]# cd /etc/sysconfig/network-scripts/

[root@vh08 network-scripts]# cp ifcfg-eth0{,:0}

[root@vh08 network-scripts]# vim ifcfg-eth0:0

TYPE=Ethernet

BOOTPROTO=none

NAME=eth0:0

DEVICE=eth0:0

ONBOOT=yes

IPADDR=192.168.4.100

PREFIX=24

[root@vh08 network-scripts]# ifup eth0:0

3、把VIP配置在vh06和vh07两台real server的lo

[root@vh06 html]# cd /etc/sysconfig/network-scripts/

[root@vh06 network-scripts]# cp ifcfg-lo{,:0}

[root@vh06 network-scripts]# vim ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.4.100

NETMASK=255.255.255.255 //如果写成了255.255.255.0将影响该网段其他主机,不管配置成192.168.4.0/24网段的哪个地址，都会出现地址冲突

NETWORK=192.168.4.100

BROADCAST=192.168.4.100

ONBOOT=yes

NAME=lo:0

[root@vh06 network-scripts]# ifup lo:0

4、在WEB服务器上修改内核参数

三台主机都有VIP192.168.4.100。当请求发送过来时，应该是调度器响应请求， real server响应。为了实现这个功能，需要修改real server的内核参数。

[root@vh06 ~]# sysctl -a | grep arp\_ig

[root@vh06 ~]# echo "net.ipv4.conf.all.arp\_ignore = 1" >> /etc/sysctl.conf

[root@vh06 ~]# echo "net.ipv4.conf.lo.arp\_ignore = 1" >> /etc/sysctl.conf

[root@vh06 ~]# sysctl -a | grep arp\_ann

[root@vh06 ~]# echo "net.ipv4.conf.all.arp\_announce = 2" >> /etc/sysctl.conf

[root@vh06 ~]# echo "net.ipv4.conf.lo.arp\_announce = 2" >> /etc/sysctl.conf

[root@vh06 ~]# sysctl -p 应用内核参数

5、配置LVS DR规则

[root@vh08 network-scripts]# ipvsadm -A -t 192.168.4.100:80 -s lc

[root@vh08 network-scripts]# ipvsadm -a -t 192.168.4.100:80 -r 192.168.4.6 -g //-g路由模式

[root@vh08 network-scripts]# ipvsadm -a -t 192.168.4.100:80 -r 192.168.4.7

[root@vh08 network-scripts]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.100:80 lc

-> 192.168.4.2:80 Route 1 0 0

-> 192.168.4.3:80 Route 1 0 0

6、验证：

[root@room9pc01 ~]# firefox 192.168.4.100

[root@room9pc01 ~]# curl http://192.168.4.100

<html><h1><font color=red>我想和你一起吃火锅</font></h1></html>

[root@room9pc01 ~]# curl http://192.168.4.100

<html><h1><font color=yellow>你是屎吗？黄色的！</font></h1></html>

[root@room9pc01 ~]# curl http://192.168.4.100

<html><h1><font color=red>我想和你一起吃火锅</font></h1></html>

[root@vh08 network-scripts]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.100:80 wlc

-> 192.168.4.6:80 Route 1 0 10

-> 192.168.4.7:80 Route 1 0 11

## ipvsadm服务

lvs起动状态管理

安装完ipvsadm后，在/usr/lib/systemd/system 下会出现ipvsadm.service。

如果第一次启动ipvsadm服务时出错，可以执行ipvsadm-save -n > /etc/sysconfig/ipvsadm 再去执行启动服务

[root@vh08 network-scripts]# systemctl status ipvsadm.service

● ipvsadm.service - Initialise the Linux Virtual Server

Loaded: loaded (/usr/lib/systemd/system/ipvsadm.service; disabled; vendor preset: disabled)

Active: inactive (dead)

root@vh08 network-scripts]# journalctl -xe

6月 15 15:54:56 vh04.tedu.cn bash[2260]: /bin/bash: /etc/sysconfig/ipvsadm: 没有那个文件或目录

[root@vh08 network-scripts]# vim /usr/lib/systemd/system/ipvsadm.service

[Service]

Type=oneshot

ExecStart=/bin/bash -c "exec /sbin/ipvsadm-restore < /etc/sysconfig/ipvsadm"

ExecStop=/bin/bash -c "exec /sbin/ipvsadm-save -n > /etc/sysconfig/ipvsadm"

ExecStop=/sbin/ipvsadm -C

[root@vh08 ]# ipvsadm-save -n > /etc/sysconfig/ipvsadm

然后再正常控制服务状态

[root@vh08 -]# systemctl start ipvsadm

[root@vh08 -]# systemctl enable ipvsadm

###################################################################################

在vh08里面编写脚本，自动删除宕机的节点，自动添加恢复的节点

[root@vh08 ~]# vim 1.sh

#!/bin/bash

RIP1=192.168.4.2

RIP2=192.168.4.3

VIP=201.1.1.4:80

while :

do

for IP in $RIP1 $RIP2

do

curl -I http://${IP} &> /dev/null

web\_state=$?

ipvsadm -Ln | grep ${IP} &> /dev/null

ip\_in\_lvs=$?

if

[ $web\_state -ne 0 -a $ip\_in\_lvs -eq 0 ]

then

ipvsadm -d -t $VIP -r $IP

elif

[ $web\_state -eq 0 -a $ip\_in\_lvs -ne 0 ]

then

ipvsadm -a -t $VIP -r $IP -m

fi

done

sleep 3

done

[root@vh08 ~]# sh 1.sh &

[root@vh06 html]# systemctl stop httpd

[root@vh08 ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 201.1.1.4:80 rr

-> 192.168.4.3:80 Masq 1 0 0

[root@vh06 html]# systemctl restart httpd

[root@vh08 ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 201.1.1.4:80 rr

-> 192.168.4.2:80 Masq 1 0 0

-> 192.168.4.3:80 Masq 1 0 0

[root@vh08 ~]# vim ./monnitor.sh

VIP=192.168.4.100:80

RIP1=192.168.4.2

RIP2=192.168.4.3

while :

do

for i in $RIP1 $RIP2

do

curl http://$i &> /dev/null

webstate=$?

ipvsadm -Ln | grep $i >/dev/null

web\_in\_lvs=$?

[[ $webstate -ne 0 && $web\_in\_lvs -eq 0 ]] && ipvsadm -d -t $VIP -r $i

[[ $webstate -eq 0 && $web\_in\_lvs -ne 0 ]] && ipvsadm -a -t $VIP -r $i

done

sleep 3

done

[root@vh08 ~]# watch -n1 ipvsadm -Ln 每隔一秒执行后面的命令，并置顶

//vh07和vh06 stop和start httpd 服务查看状态

vim使用技巧：

:13,20s/^/ / 在13到20行开头添加一个table空格

56kbps = 7 kB/s

1byte = 8bit

# HAProxy服务器

HAProxy概述

HAProxy简介：工作在代理的模式下

它是免费、快速并且可靠的一种解决方案

适用于那些负载特大的web站点,这些站点通常又需要会话保持或七层处理

提供高可用性、负载均衡以及基于TCP和HTTP应用的代理

负载均衡器:nginx/haproxy/lvs/f5(硬件级别，很贵)

衡量负责均衡器性能的因素：

Session rate 会话率：每秒钟产生的会话数

Session concurrency 并发会话数：服务器处理会话的时间越长,并发会话数越多（同时有多少访问）

Data rate 数据速率：

- 以MB/s或Mbps衡量

- 大的对象导致并发会话数增加

- 高会话数、高数据速率要求更多的内存

## HAProxy工作模式：

mode http：客户端请求被深度分析后再发往服务器

mode tcp：客户端与服务器之间建立会话,不检查第七层信息

mode health：仅做健康状态检查,已经不建议使用

## HTTP协议解析

HTTP解析：

当HAProxy运行在HTTP模式下,HTTP请求(Request)和响应(Response)均被完全分析和索引,这样便于创建恰当的匹配规则

理解HTTP请求和响应,对于更好的创建匹配规则至关重要

HTTP事务模型：

HTTP协议是事务驱动的

每个请求(Request)仅能对应一个响应(Response)

常见模型：

HTTP close：客户端向服务器建立一个TCP连接

客户端发送请求给服务器

服务器响应客户端请求后即断开连接

如果客户端到服务器的请求不只一个,那么就要不断的去建立连接

TCP三次握手消耗相对较大的系统资源,同时延迟较大

Keep-alive：一次连接可以传输多个请求

客户端需要知道传输内容的长度,以避免无限期的等待传输结束

降低两个HTTP事务间的延迟

需要相对较少的服务器资源

Pipelining：仍然使用Keep-alive

在发送后续请求前,不用等前面的请求已经得到回应

适用于有大量图片的页面

降低了多次请求之间的网络延迟

HTTP头部信息

请求头部信息：

方法：GET

URI：/serv/login.php?lang=en&profile=2

版本：HTTP/1.1

请求头包含许多有关的客户端环境和请求正文的有用信息,如浏览器所使用的语言、请求正文的长度等

响应头部信息：

版本：HTTP/1.1

状态码：200 【有的状态码是300 Movred Permanently 永久重定向，，5？？拒绝访问】

原因：OK

HAProxy配置实例

配置文件说明

HAProxy配置参数来源：

命令行：总是具有最高优先级

global部分：全局设置进程级别参数

代理声明部分：来自于default、listen、frontend和backend

配置文件可由如下部分构成:

default：为后续的其他部分设置缺省参数

缺省参数可以被后续部分重置

frontend：描述接收客户端侦听套接字(socket)集

backend：描述转发链接的服务器集

listen：把frontend和backend结合到一起的完整声明

代理：

正向代理：帮助客户端缓存服务器数据

反向代理：帮助缓存服务器数据

## hapoxy配置（负载均衡集群）

1.vh04上安装haproxy

[root@vh04 ~]# yum -y install haproxy

2.修改配置文件

[root@vh04 ~]# vim /etc/haproxy/haproxy.cfg

26 log 127.0.0.1 local2 [err warning info debug]

28 chroot /var/lib/haproxy haproxy的pid存放路径

29 pidfile /var/run/haproxy.pid 最大连接数,默认4000

30 maxconn 4000

31 user haproxy

32 group haproxy

33 daemon 创建1个进程进入deamon模式运行

defaults

mode http 默认的模式mode {tcp|http|health} log global采用全局定义的日志

option dontlognull 不记录健康检查的日志信息

option httpclose 每次请求完毕后主动关闭hEp通道

option httplog 日志类别hEp日志格式

option forwardfor 后端服务器可以从HEp Header中获得客户端ip

option redispatch serverid服务器挂掉后强制定向到其他健康服务器

timeout connect 10000 如果backend没有指定,默认为10s

timeout client 300000 客户端连接超时

timeout server 300000 服务器连接超时

maxconn 60000 最大连接数

retries 3 3次连接失败就认为服务不可用,也可以通过后面设置

将这个以下部分全部删除，

60 #---------------------------------------------------------------------

61 # main frontend which proxys to the backends

62 #---------------------------------------------------------------------

在最下面添加一下内容

定义监控界面

listen stats

bind 0.0.0.0:1080 端口号随便写

stats refresh 30s 每30s刷新一次

stats uri /stats 查看刷新文件的页面url

stats realm haprox manager 最后两个随便写

stats auth admin:admin 统计页面的用户名和密码，前面是用户名

listen web\_backend 0.0.0.0:80 后端服务器，监听在80端口

cookie SERVERID rewrite

balance roundrobin

server web1 192.168.4.2:80 cookie a1i1 check inter 2000 rise 2 fall 5 检查这台服务器，两千毫秒检查一次，检查超过5次是不成功的

server web2 192.168.4.3:80 cookie a1i2 check inter 2000 rise 2 fall 5

3.启动haproxy服务

[root@vh04 ~]# systemctl stop httpd //调度器需要关闭httpd服务

[root@vh04 ~]# systemctl restart haproxy

访问验证：

[root@vh01 ~]# firefox 192.168.4.4:1080/stats

用户名和密码都是admin

stop掉vh02和vh03任何一台，刷新页面，查看页面颜色的变化

配置调度器本身也是日志服务器，可以接受网络发来的日志

[root@vh04 ~]# netstat -nutlp |grep :514

[root@vh04 ~]# vim /etc/rsyslog.conf

去掉日志注释

15 $ModLoad imudp

16 $UDPServerRun 514

19 $ModLoad imtcp

20 $InputTCPServerRun 514

[root@vh04 ~]# systemctl restart rsyslog

[root@vh04 ~]# netstat -nutlp |grep :514

tcp 0 0 0.0.0.0:514 0.0.0.0:\* LISTEN 7897/rsyslogd

tcp6 0 0 :::514 :::\* LISTEN 7897/rsyslogd

udp 0 0 0.0.0.0:514 0.0.0.0:\* 7897/rsyslogd

udp6 0 0 :::514 :::\* 7897/rsyslogd

客户端访问：

[root@room9pc01 ~]# firefox 192.168.4.4:1080/stats

[root@room9pc01 ~]# firefox 192.168.4.4

这两个访问哪个都可以

查看haproxy的日志：

[root@vh04 ~]# tail -f /var/log/messages

客户端访问负载均衡，将有日志产生

# Keepalived热备 ： 高可用主备机制

## Keepalived概述

调度器出现单点故障,如何解决?

Keepalived实现了高可用集群

Keepalived最初是为LVS设计的,专门监控各服务器节点的状态

Keepalived后来加入了VRRP功能,防止单点故障

Keepalived运行原理：

Keepalived检测每个服务器节点状态

服务器节点异常或工作出现故障, Keepalived将故障节点从集群系统中剔除

故障节点恢复后,Keepalived再将其加入到集群系统中

所有工作自动完成,无需人工干预

HSRP：热备份路由协议，思科私有

VRRP：虚拟冗余路由协议 ，IETF共有标准相

HA：高可用

心跳：HearBeat，相当于路由器设备上的hell消息

## 双机热备：keepalived

一、配置高可用web集群

1、拓扑：两台web服务器，一台数据库服务器

2、在web服务器上安装keepalived

[root@vh02 ~]# yum -y install keepalived

3、修改配置

[root@vh02 ~]# vim /etc/keepalived/keepalived.conf

将全局的 vrrp\_strict 这一行注释掉

global\_defs {

notification\_email { 管理员email地址

root@localhost

}

notification\_email\_from admin@tedu.cn 谁发

smtp\_server 127.0.0.1 用哪台服务器发

smtp\_connect\_timeout 30

router\_id LVS\_DEVEL 设置路由的ID号

vrrp\_skip\_check\_adv\_addr

vrrp\_garp\_interval 0

vrrp\_gna\_interval 0

}

vrrp\_instance VI\_1 { VI\_1 随便起的名，实例名

state MASTER 主服务器，从属服务器改为BACKUP

interface eth0 用哪一个网卡

virtual\_router\_id 51 产生虚拟路由器的ID号，ID号要求一样

priority 150 优先级

advert\_int 1 通告间隔，每隔一秒发一次，心跳1秒

authentication { 认证，用密码做认证的，是共享密码，要求一样

auth\_type PASS

auth\_pass 1111 主辅服务器密码必须一致

}

virtual\_ipaddress { 虚拟地址，两个节点必须一样

192.168.4.200

}

}

剩下的全部删除

4、启动服务

[root@vh02 ~]# systemctl start keepalived

5、查看虚拟ip

[root@vh02 ~]# ip a s eth0 //用ifconfig看不到 一个网卡的第二个ip

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 52:54:00:ec:bb:c1 brd ff:ff:ff:ff:ff:ff

inet 192.168.4.2/24 brd 192.168.4.255 scope global eth0

valid\_lft forever preferred\_lft forever

inet 192.168.4.200/32 scope global eth0

valid\_lft forever preferred\_lft forever

[root@vh02 ~]# scp /etc/keepalived/keepalived.conf 192.168.4.3:/etc/keepalived/keepalived.conf

[root@vh03 ~]# vim /etc/keepalived/keepalived.conf

global\_defs {

notification\_email {

root@localhost

}

notification\_email\_from admin@tedu.cn

smtp\_server 127.0.0.1

smtp\_connect\_timeout 30

router\_id LVS\_DEVEL

vrrp\_skip\_check\_adv\_addr

vrrp\_garp\_interval 0

vrrp\_gna\_interval 0

}

vrrp\_instance VI\_1 {

state BACKUP 从属的用BACKUP

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.4.200

}

}

6.清空防火墙规则

[root@vh02 ~]# iptables -F

[root@room9pc01 ~]# ping 192.168.4.200

[root@room9pc01 ~]# firefox 192.168.4.200/bbs

在vh02上关闭keepalived，再查看eth0的ip，再用客户端访问

7、验证

[root@vh02 ~]# systemctl stop keepalived

[root@vh03 ~]# ip a s eth0

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 52:54:00:df:6b:7c brd ff:ff:ff:ff:ff:ff

inet 192.168.4.3/24 brd 192.168.4.255 scope global eth0

valid\_lft forever preferred\_lft forever

inet 192.168.4.200/32 scope global eth0

valid\_lft forever preferred\_lft forever

[root@room9pc01 ~]# firefox 192.168.4.200/bbs 依然可以访问

然后再把vh02重启服务，ip a s eth0 可以看到ip，在vh03上就没有了

## LVS+keepalived

在keepalived高可用配置文件里，

双主结构，把vrrp\_instance VI\_1 {再复制一份，03设为master，02设为backup，再用一个新的vip就可以了，实现双主结构

LVS+keepalived，实现高可用，负载均衡的web集群

一，web服务器配置

添加node5

1、创建额外的调度器 vh05.tedu.cn 192.168.4.5/24

2、清除vh04上的lvs规则，因为规则由keepalived配置

[root@vh04 ~]# ipvsadm -D -t 192.168.4.100:80

3、在vh05上安装lvs

[root@vh05 ~]# yum -y install ipvsadm

4、web服务器需要在lo上配置vip，需要修改内核参数

5、不要在调度器上手工配置vip，因为vip由keepalived决定出现在哪台调度器上。所以把vh04的vip清除

[root@vh04 ~]# rm -rf /etc/sysconfig/network-scripts/ifcfg-eth0:0

[root@vh04 ~]# ifdown eth0:0

6、在调度器上安装keepalived

[root@vh05 ~]# yum -y install keepalived

[root@vh04 ~]# yum -y install keepalived

7、修改调度器配置

[root@vh04 ~]# vim /etc/keepalived/keepalived.conf

5 root@localhost

7 notification\_email\_from admin@tedu.cn

8 smtp\_server 127.0.0.1

10 router\_id vh04 路由器ID号

12 # vrrp\_strict

21 priority 150

28 192.168.4.100 与vip调度器地址一样

给lvs配置规则

32 virtual\_server 192.168.4.100 80 {

33 delay\_loop 6

34 lb\_algo rr

35 lb\_kind DR

36 persistence\_timeout 50 // 50秒内客户端访问我，让他访问相同的调度器

39 real\_server 192.168.4.2 80 {

41 TCP\_CHECK {

删除41行下面的8行

42 connect\_timeout 3 超时时间3秒

43 nb\_get\_retry 3 失败一共检查三次

44 delay\_before\_retry 3

45 }

46 }

复制上面的8行，剩下的全部删除

47 real\_server 192.168.4.3 80 {

48 weight 1

49 TCP\_CHECK {

50 connect\_timeout 3

51 nb\_get\_retry 3

52 delay\_before\_retry 3

53 }

54 }

55 }

删除后续全部内容、

8、启动服务

[root@vh04 ~]# systemctl restart keepalived

[root@vh04 ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.100:80 rr persistent 50

-> 192.168.4.2:80 Route 1 0 0

-> 192.168.4.3:80 Route 1 0 0

[root@vh04 ~]# ip a s eth0

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 52:54:00:e3:2e:83 brd ff:ff:ff:ff:ff:ff

inet 192.168.4.4/24 brd 192.168.4.255 scope global eth0

valid\_lft forever preferred\_lft forever

inet 192.168.4.100/32 scope global eth0

valid\_lft forever preferred\_lft forever

[root@room9pc01 ~]# firefox 192.168.4.100

访问的是vh03上的内容

9、备份调度器，同上，注意state为BACKUP

[root@vh04 ~]# scp /etc/keepalived/keepalived.conf 192.168.4.9:/etc/keepalived/keepalived.conf

state BACKUP

priority 100

[root@vh09 ~]# systemctl restart keepalived

[root@vh09 ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.100:80 rr persistent 50

-> 192.168.4.2:80 Route 1 0 0

-> 192.168.4.3:80 Route 1 0 0

9、备份lvs调度器启动侯，没有vip，可以把vh04关机，查看vh05的情况

[root@vh04 ~]# systemctl stop keepalived

[root@vh04 ~]# ip a s eth0

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 52:54:00:e3:2e:83 brd ff:ff:ff:ff:ff:ff

inet 192.168.4.4/24 brd 192.168.4.255 scope global eth0

valid\_lft forever preferred\_lft forever

[root@vh05 ~]# ip a s eth0

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 52:54:00:1d:71:cf brd ff:ff:ff:ff:ff:ff

inet 192.168.4.5/24 brd 192.168.4.255 scope global eth0

valid\_lft forever preferred\_lft forever

inet 192.168.4.100/32 scope global eth0

valid\_lft forever preferred\_lft forever

[root@vh04 ~]# mail 可以查到邮件

[root@vh05 ~]# mail

## keepalived高可用 + mysql主从 配置

实现当一台数据库挂掉时，可以自动转换到另一台数据库上工作，需要三台虚拟机，mysql1做主，给用户授权，允许任何主机登录，mysql2做从，

mysql1的主配置文件

server\_id=51

log-bin=db51

binlog-format="mixed"

主库授权：mysql> grant replication slave on \*.\* to yaya@"%" identified by "123456";

mysql> show master status;

配置虚拟vip lo:0 192.168.4.100

mysql2的主配置文件

server\_id=52

配置从库：

mysql> change master to

-> master\_host="192.168.4.51",

-> master\_user="yaya",

-> master\_password="123456",

-> master\_log\_file="db51.000001",

-> master\_log\_pos=154;

启动slave进程

mysql> start slave;

配置虚拟vip lo:0 192.168.4.100

keepalived高可用

配置虚拟vip eth0:0 192.168.4.100

修改配置文件

real\_server mysql1的ip

real\_server mysql2的ip

启动服务

访问mysql1

[root@vh04 ~]# mysql -h192.168.4.1 -uadmin -p123456

当mysql1当掉时，访问

[root@vh04 ~]# mysql -h192.168.4.2 -uadmin -p123456

依然可以访问，因为设的固定ip是一样的，都是通过虚拟ip 4.100访问的

# 集群调度软件对比

Nginx分析

• 优点

– 工作在7层,可以针对http做分流策略

– 正则表达式比HAProxy强大

– 安装、配置、测试简单,通过日志可以解决多数问题

– 并发量可以达到几万次

– Nginx还可以作为Web服务器使用

• 缺点

– 仅支持http、https、mail协议,应用面小

– 监控检查仅通过端口,无法使用url检查

LVS分析

• 优点

– 负载能力强,工作在4层,对内存、CPU消耗低

– 配置性低,没有太多可配置性,减少人为错误

– 应用面广,几乎可以为所有应用提供负载均衡

• 缺点

– 不支持正则表达式,不能实现动静分离

– 如果网站架构庞大,LVS-DR配置比较繁琐

HAProxy分析

• 优点

– 支持session、cookie功能

– 可以通过url进行健康检查

– 效率、负载均衡速度,高于Nginx,低于LVS

– HAProxy支持TCP,可以对MySQL进行负载均衡

– 调度算法丰富

• 缺点

– 正则弱于Nginx

– 日志依赖于syslogd,不支持apache日志

集群和存储总结汇总：

集群分类：HPC/LB/HA

HPC在企业中比较少，科研用的多

HA：高可用。负载不高，防止单点故障

LB：负载均衡。负载非常高，单一服务器无法承受很大的压力。

负载均衡设备：

硬件：F5/Netscaler/Radware

软件：LVS/NGINX/HAPROXY

硬件优点：快 。缺点：贵

软件优点：便宜。缺点：慢

lvs的效率可以达到F5的60%，nginx/haproxy效率达到F5的10%

nginx 只能用于web和mail的调度，支持丰富的正则表达式。可以检查后端服务器状态。配置文件需要更多的改动。三款软件中，效率最低。

haproxy 可以支持4层功能（mode tcp）。正则弱于nginx。可以检查后端服务器状态。代替客户访问real server，real server返回数据给haproxy，所以流量很大

lvs 只支持到第4层。因为只能识别4层信息，也就无所谓正则了，不支持正则表达式。本身没有后端服务器的健康检查机制。配置性较低

lvs dr模式，real server直接回复客户请求，不经过调度器，所以基本没有流量

开始网络规模小，没有非常的负载，日PV只有几百万，考虑nginx/haproxy就够了。

规模扩大，日PV达到千万以上级别，就要考虑lvs