1、集群分类：

（1）高性能计算集群（HPC）

（2）负载均衡集群（LB）

（3）高可用集群（HA）

**LB和HA在企业中用的更多。**

2、服务器架构

《大型网站技术架构》李志慧

IBM：小型机

ORACLE：数据库

EMC存储：

去IOE运动

**一般的架构一台服务器：**

**运行时：速度问题，**

**故障时：单点故障**

**安全问题：一个泄露全部泄露**

**LVS**（Linux Virtual Server）**调度器**

客户端-----LVS---Vainish(缓存)-----LVS（集群）---web放的代码(集群)---memcache（多台）---DB（高可用）------专门的存储服务器（分布式存储集群，负责所有的图片，影视频）【距离较远**CDN：**内容分发网络】

99% 一年宕机时间不超过4天

99.9% 一年宕机时间不超过10小

99.99% 一年宕机时间不超过1小时

99.999% 一年宕机时间不超过6分钟

————————————————————

**服务器架构：**

**《大型网站技术架构》李志慧**

**---主要讲淘宝的网络架构----一切以业务为准**

————————————————————

**Nginx并发量一般在3～4W左右**

**Appache默认150并发**

**针对：大中型网站--1000～2000**

**小型网站100～1000根据业务**

**微型机：PC**

**小型机：几十万，上百万**

——————————————————————

**配置LAMP分离的DISCUZ论坛**

1、创建三台虚拟机并初始化

Node1.tedu.cn 192.168.4.1 mysql

Node2.tedu.cn 192.168.4.2 web1

Node3.tedu.cn 192.168.4.3 web2

2、配置node1为数据库服务器

[root@node1 ~]# yum install -y mariadb-server

[root@node1 ~]# systemctl start mariadb

[root@node1 ~]# systemctl enable mariadb

[root@node1 ~]# mysql

MariaDB [(none)]> grant all on \*.\* to 'admin'@'192.168.4.%' identified by 'tedu.cn';

3、配置node2节点为web服务器

[root@node2 ~]# yum install -y httpd php php-mysql

[root@node2 ~]# systemctl start httpd

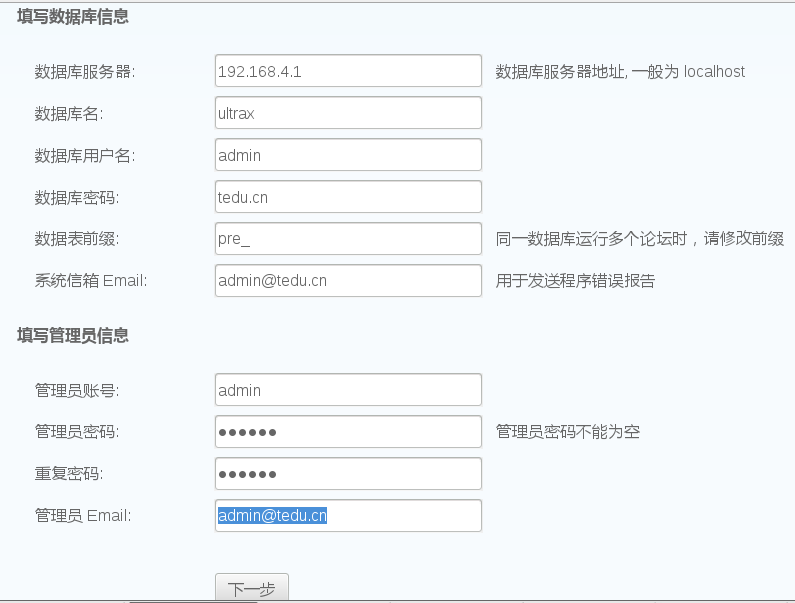
[root@node2 ~]# systemctl enable httpd

[root@node2 ~]# unzip Discuz\_X3.0\_SC\_UTF8.zip

[root@node2 ~]# cp -r upload /var/www/html/bbs

[root@node2 ~]# chown -R apache.apache /var/www/html/bbs

访问http://192.168.4.2/bbs进行安装



4、后台管理

使用admin登陆后，访问“管理中心”。在后台管理界面，可以通过“论坛”更改版面设置。

5、配置node3成为第二台web服务器

[root@node3 ~]# yum install -y httpd php php-mysql

[root@node3 ~]# systemctl start httpd

[root@node3 ~]# systemctl enable httpd

把node2的bbs打包拷贝到node3上解压到/var/www/html/下

[root@node2 ~]# cd /var/www/html/

[root@node2 html]# tar czf bbs.tar.gz bbs

[root@node2 html]# scp bbs.tar.gz 192.168.4.3:/var/www/html/

[root@node3 ~]# cd /var/www/html/

[root@node3 html]# tar xzf bbs.tar.gz

LVS: Linux虚拟服务器

|  |  |
| --- | --- |
| VIP | 虚拟IP地址，客户端访问集群的地址 |
| RIP | 集群所提供应用程序的地址（HTTP，FTP） |
| DIP | 调度器的真实地址 |

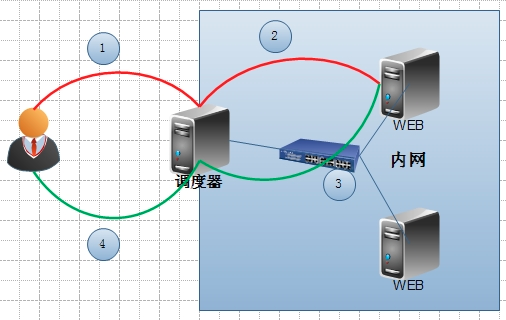
1、工作模式：

（1）tun隧道模式：很少使用

（2）nat模式：相对较少

优点：节点服务器使用私有IP地址，与负载调度器位于同一个物理网络，安全性比DR模式和TUN模式要高。

缺点：（压力大的根本原因）大规模应用场景中，调度器容易成为系统瓶颈



（3）dr模式：路由模式，应用最多

优点：负载均衡器也只是分发请求，应答包通过单独的路由方法返回给客户端，大大提高了服务器并发能力。

不足：1、LVS-RS间必须在同一个VLAN。

      2、RS上绑定VIP，风险大。

1. full nat：在很大规模环境下使用
2. NAT只修改目的IP / FULL nat需要修改目的IP 和源IP

1.1.1.1

9.9.9.9

LVS

9.9.9.9/192.168.4.10

192.168.4.10--->192.168.3.1

（1.1.1.1） (9.9.9.9)

核心路由器

9.9.9.9

192.168.3.1

192.168.2.1

Fullnat：可以进行跨VLAN通信（但需要编辑内核）

将外网的1.1.1.1 -----> 9.9.9.9改写为全内网的192.168.4.10------>192.168.3.1，在返回时，则返回到核心路由后直接转化为9.9.9.9------>1.1.1.1的改之前的源地址与目标地址。（但是在不同版本的Linux上都需要打不同的内核布丁才可以）

2、调度算法

LVS的调度方法分为两种，一种是静态方法，一种是动态方法：

静态方法：仅根据算法本身实现调度；实现起点公平，不管服务器当前处理多少请求，分配的数量一致

动态方法：根据算法及后端RS当前的负载状况实现调度；不管以前分了多少，只看分配的结果是不是公平

（1）轮询rr

（2）加权轮询wrr

（3）最少连接lc 把新的连接请求分配到当前连接数最小的服务器

（4）加权最少连接wlc

（5）基于局部的最少连接lblc

（6）带复制的基于局部的最少连接lblcr

（7）源地址散列sh 简单的说就是有将同一客户端的请求发给同一个real server

（8）目标地址散列dh 将同样的请求发送给同一个server

（9）期望的最少的延迟sed

（10）最少队列调度nq 此算法是sed改进，就是说不管你的权重多大都会被分配到请求

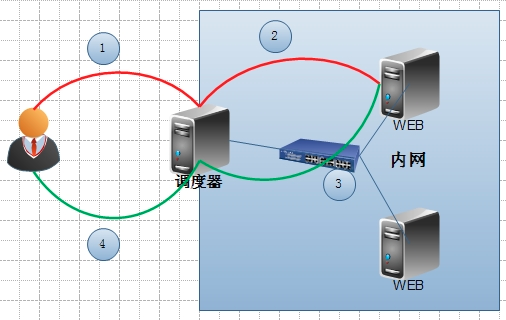
**配置LVS－NAT模式**

（1）  客户端访问集群的VIP地址，请求web服务。（请求报文：源地址为CIP，目标地址为VIP）

（2）  调度器收到客户端的请求报文，会修改请求报文中的目标地址（VIP）为RIP，并且将请求根据相应的调度算法送往后端web服务器。（请求报文：源地址CIP，目标地址为RIP）

（3）  Web服务器收到请求，检查报文是访问自己的，并且自己也提供web服务，就会响应这个请求报文；并且发送给调度器（响应报文，源地址RIP，目标地址CIP）

（4）  调度器收到web服务器的响应报文，会根据自己内部的追踪机制，判断出用户访问的是VIP，这个时候会修改源地址为VIP地址响应客户端请求。



1、修改两台web服务器，为其添加网关192.168.4.4

[root@node2 html]# ifdown eth0; ifup eth0 -----**注意：必须如此否则会改不回来**

2、创建虚拟机node4作为调度器

Node4.tedu.cn eth0:192.168.4.4 eth2:201.1.1.4

3、打开node4的路由转发功能（RHEL7默认已经打开，可以跳过）

[root@node4 ~]# sysctl -a | grep ip\_forward

[root@node4 ~]# echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

[root@node4 ~]# sysctl -p

4、配置node4作为LVS服务器

[root@node4 ~]# yum install -y ipvsadm

(1)创建虚拟服务器

[root@node4 ~]# ipvsadm -A -t 201.1.1.4:80 -s rr

（2）向虚拟服务器中加入real server

-m表示NAT模式，-w设置权重，-t是TCP

[root@node4 ~]# ipvsadm -a -t 201.1.1.4:80 -r 192.168.4.2 -m

[root@node4 ~]# ipvsadm -a -t 201.1.1.4:80 -r 192.168.4.3 -m -w 2

（3）查看

[root@node4 ~]# ipvsadm -Ln

（4）验证：访问http://201.1.1.4/bbs

在两台服务器上制作不同的主页，以便访问时可以区分。

[root@room8pc16 kvms\_ansi]# curl http://201.1.1.4

5、相关命令

从虚拟服务器中删除RIP

[root@node4 ~]# ipvsadm -d -t 201.1.1.4:80 -r 192.168.4.3

删除虚拟服务器

[root@node4 ~]# ipvsadm -D -t 201.1.1.4:80

修改调度算法为WRR

[root@node4 ~]# ipvsadm -E -t 201.1.1.4:80 -s wrr

每隔1秒钟运行ipvsadm -Ln

[root@node4 bin]# watch -n1 ipvsadm -Ln

**配置LVS－DR模式**

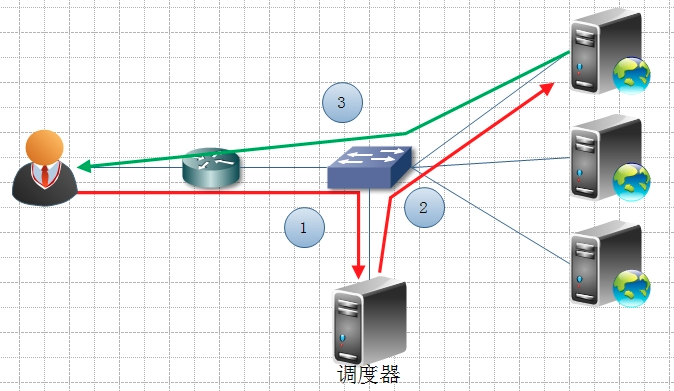
（1）首先，客户端CIP的请求发送给LVS调度器的VIP。

（2）LVS调度器收到客户端的请求包后，将数据包的MAC地址改成LVS调度器选择的某一台RS的MAC地址，并通过交换机（数据链路层）发送给RS服务器（因为MAC地址是rs服务器，所以，rs可以接收到该数据报。）注意：此时数据包的目的及源ip地址没有发生任何改变。

（3）

1、RS的数据链路层收到发送来的数据报文请求后，会从链路层往上传给IP层，此时IP层需要验证请求的目标IP地址。因为包的目标IP（即VIP）并不是像常规数据报那样为RS的本地IP，而仅仅目的MAC地址是RS的。所以，在RS上需要配置一个VIP的loopbackdevice，是因为loopback device是服务器本地使用的网络接口，对外是不可见的，不会跟LVS的ip冲突。

2、RS处理数据包完成后，将应答直接返回给客户端（源ip为VIP，目标ip为CIP）。回复数据报不在经过调度器。因此，如果对外提供LVS负载均衡服务，则RS需要连上互联网才能将应答包返回给客户端。不过，RS最好为带公网IP的服务器，这样可以不经过网关直接回应客户，如果多个RS使用了同一网关出口，网关会成为LVS架构的瓶颈，会大大降低LVS的性能



1、清除eth2的IP地址和LVS规则

[root@node4 bin]# ifdown eth2

[root@node4 bin]# ipvsadm -D -t 201.1.1.4:80

[root@node4 bin]# kill %1 ->杀掉后台1号作业

2、在调度器的物理网卡上添加vip

[root@node4 bin]# cd /etc/sysconfig/network-scripts/

[root@node4 network-scripts]# cp ifcfg-eth0{,:0}

[root@node4 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@node4 network-scripts]# ifup eth0:0

3、在real server的lo上配置vip

[root@node2 html]# cd /etc/sysconfig/network-scripts/

[root@node2 network-scripts]# cp ifcfg-lo{,:0}------**拷贝一个并扩展命名**

[root@node2 network-scripts]# vim ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.4.100

子网掩码必须是4个255，否则其他主机启动时，都是地址被占用

NETMASK=255.255.255.255

NETWORK=192.168.4.100

BROADCAST=192.168.4.100

ONBOOT=yes

NAME=lo:0

[root@node2 network-scripts]# ifup lo:0

4、在real server上修改内核参数，使得客户端发送到VIP的请求只有调度服务器响应，real server(web服务器)不要要响应。

[root@node2 ~]# sysctl -a | grep arp\_ig

[root@node2 ~]# echo "net.ipv4.conf.all.arp\_ignore = 1" >> /etc/sysctl.conf

[root@node2 ~]# echo "net.ipv4.conf.lo.arp\_ignore = 1" >> /etc/sysctl.conf

[root@node2 ~]# sysctl -a | grep arp\_ann

[root@node2 ~]# echo "net.ipv4.conf.all.arp\_announce = 2" >> /etc/sysctl.conf

[root@node2 ~]# echo "net.ipv4.conf.lo.arp\_announce = 2" >> /etc/sysctl.conf

[root@node2 ~]# sysctl -p ------**设置生效**

5、创建规则，默认LVS采用DR模式，也可以明确使用-g

[root@node4 ~]# ipvsadm -A -t 192.168.4.100:80 -s lc

[root@node4 ~]# ipvsadm -a -t 192.168.4.100:80 -r 192.168.4.2

[root@node4 ~]# ipvsadm -a -t 192.168.4.100:80 -r 192.168.4.3 -g

6、配置服务

[root@node4 ~]# touch /etc/sysconfig/ipvsadm -----**只有创建才能重启成功**

[root@node4 ~]# systemctl start ipvsadm

[root@node4 ~]# systemctl enable ipvsadm

注意：若两台LVS上已经无法满足当前的流量：

可以考虑到LVS+OSPF进行复负载均衡

更高效LVS+DPDK+OSPF

DPDK----->绕过内核直接处理数据------>懂得C语言

++++++++++++++++++++++++++

Ctrl + F5强制重新连接服务器

------------------------------------------------

History | grep ipvsadm

!+行号执行历史命令

-----------------------------------------------

脚本的编写思路：

先作最基础的：基本命令

再i加条件：什么时候加，什么时候删掉

再加一些其他内容丰富

--------------------------------------------------

Vim中

替换指令：

：2（起始），9（结束）s/要替换的内容/替换成的内容/g(全部)

脚本监控：

#!/bin/bash

VIP=201.1.1.4:80

RIP1=192.168.4.2

RIP2=192.168.4.3

while :

do

for IP in $RIP1 $RIP2

do

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

web\_stat=$?

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

web\_in\_lvs=$?

if [ $web\_stat -ne 0 -a $web\_in\_lvs -eq 0 ]; then

ipvsadm -d -t $VIP -r $IP

elif [ $web\_stat -eq 0 -a $web\_in\_lvs -ne 0 ]; then

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

fi

done

sleep 1

done