LVS负载均衡集群

一、前期理论

1.背景需求

2.集群技术的概述

3.集群的类型

1.负载均衡集群(Load Balance Cluster) ---LB

2.高可用集群(High Availability Cluster) ---HA

3.SLA服务水平协定

4.负载均衡集群的分层介绍

➢第一层 负载调度器

➢第二层 服务器池

➢第三层 共享存储

二、LVS虚拟服务器

LVS负载均衡的工作模式

➢地址转换(Network Address Translation) :简称NAT模式

➢IP隧道(IP Tunnel) :简称TUN模式

➢直接路由(Direct Routing) :简称DR 模式

三、LVS 的负载均衡调度算法

1、轮叫调度(Round Robin) (rr)

2、加权轮叫(Weighted Round Robin) (wrr)

3、最少链接(Least Connections) (lc)

4、加权最少链接(Weighted Least Connections) (wlc)

5、基于局部性的最少链接(Locality-Based Least Connections) (lblc)

6、带复制的基于局部性最少链接(Locality-Based Least Connections with Replication) (lblcr)

7、目标地址散列(Destination Hashing) (dh)

8、源地址散列(Source Hashing) (sh)

9、最短的期望的延迟(Shortest Expected Delay Scheduling SED) (sed)

10、最少队列调度(Never Queue Scheduling NQ) (nq)

四、使用ipvsadm管理工具

1)创建虚拟服务器

2)添加服务器节点

五、实验--构建LVS负载均衡集群

1、地址转换模式(LVS-NAT)

1.关闭防火墙：

2.配置192.168.200.107的网卡IP

3.配置windows7（客户机）的IP：模拟外网

4.指定192.168.200.108及192.168.200.109的网关：

5.开启路由转发

6.配置负载分配策略

7.配置192.168.200.108和192.168.200.109的网络服务：

8.测试：172.16.1.1

9.保存负载分配策略

10.删除负载分配策略

2、直接路由模式(LVS-DR)

1) 所有主机关闭防火墙和selinux:

LVS负载均衡集群

真正想了解集群，LVS是正统

前期的理论特别重要--->面试　集群理解　故障排查

一、前期理论

1.背景需求

在互联网应用中，

随着站点对硬件性能，响应速度，服务稳定性，数据可靠性等需求越来越高

用户的体验对于网站很重要

单台服务器将难以承担所有的访问

除了使用**价格昂贵的大型机**

企业还有另一种选择来解决难题

通过**整合多台廉价的普通服务器**来构建大型集群环境（前几年很多公司追求的事）

以同一个地址对外提供相同的服务（对于用户来说有一个统一的入口）

集群（windows里面叫群集）的称呼来自英文单词Cluster、 表示一群、一串的意思

用在服务器领域则表示大量服务器的集合体，以区分于单个服务器

2.集群技术的概述

根据实际企业环境的不同，群集所提供的功能也各不相同，采用的技术细节也不同

先了解一些群集的共性特征，才能在构建和维护群集的工作中做到心中有数，

避免操作上的盲目性

3.集群的类型

无论是哪种群集，都至少包括2+节点服务器，

而对外表现为一个整体，只提供一个访问入口(域名或IP地址)，

相当于一台大型计算机能力。

根据群集所针对的目标不同，可以分为以下三种类型：

很重要，要理解，还需知道它们的实现目标，及典型的产品

1.负载均衡集群(Load Balance Cluster) ---LB

目标：

- 提高应用系统的响应能力
- 尽可能处理更多的访问请求
- 减少延迟
- 获得高并发
- 高负载(LB)的整体性能
- 例如：食堂打饭的窗口

例如：DNS 轮询，应用层交换，反向代理等，

负载分配依赖于主节点(调度器 | 负载均衡器 比如nginx)的分流算法，

将来自客户端的访问请求分担给多个服务器节点，从而缓解整个系统的负载压力，

实现集群的软件：Linux Virtual Server(LVS)、Haproxy、 Nginx

这三个软件在软件实现里面特别火

实现集群的硬件：

F5（主要做集群硬件设备）主要产品是 BIG-IP、在硬件行业中绝对的龙头老大

F5收购了nginx

Ctrix Netscaler、A10、Array、 绿盟、梭子鱼

阿里云基于Nginx包装一个产品：SLB（基于nginx做出来的一个产品）

淘宝的 Tengine

2.高可用集群(High Availability Cluster) ---HA

目标：

- 提高应用系统的可靠性
- 尽可能减少中断时间
- 确保服务的连续性
- 达到高可用(HA)的容错效果

例如:故障切换，双机热备，多机热备等。

工作模式包括双工，主从两种模式。

双工即节点同时在线;主从则只有主节点在线，当出现故障时从节点自动切换为主节点。

这类集群中比较著名的有: Heartbeat、 Keepalived

3.SLA服务水平协定

面试&聊天：你们公司高可用可达到几个9?

Unavailability Durations Based on Availability Percentages

Availability	Unavailability Per Year
98%	7.3 days
99%	3.65 days
99.8%	17 hrs 31 min
99.9%	8 hrs 45 min
99.99%	52.5 min
99.999%	5.25 min
99.9999%	31.5 sec

整年整个系统的高可用

整个集群对外提供服务的时候，终中断的时间，加起来

面试的时候不要吹牛逼吹大了，人家问你几个9，不要说5个或6个9

银行之类的，各种灾备，到达5个9，然后过度到6个

12306或游戏公告升级，用户无法上线，银行发公告，几点到几点不能转账业务

对外提供业务的公司，3个9，就很好了

政府网站的高可用可low了

4.高性能运算集群(High Performance Computer Cluster) --HPC

目标：

- 提高应用系统的CPU运算速度
- 扩展硬件资源和分析能力
- 获得相当于大型，超级计算机的高性能运算(HPC)能力

例如:云计算，网格计算，

HPC主要依赖于分布式运算，并行计算

通过专用硬件和软件将多个服务器的CPU，内存等资源整合在一起，

实现大型，超级计算机才具有的计算能力。

软件：Hadoop（大数据）

4.负载均衡集群的分层介绍

主要基于静态来讨论，动态的是session问题的相关讨论

在典型的负载均衡集群中,主要包含三个层级。

1.LVS通过前端至少一个负载调度器(LoadBalancer，也叫负载均衡器)（通常是两台）

无缝地将网络请求调度到真实服务器上，

从而使得服务器集群的结构对客户是透明的，

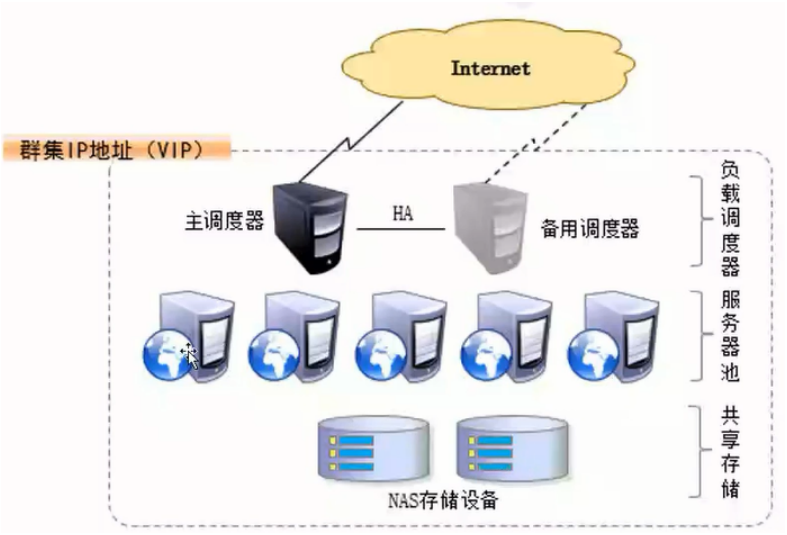
2.客户访问集群系统提供的网络服务就像访问一台高性能、高可用的服务器一样。

客户程序不受服务器集群的影响不需作任何修改。

系统的伸缩性通过在服务器机群中透明地加入或删除一个节点来达到,

通过检测节点或服务进程故障和正确地重置使系统达到高可用性。

3.为了保持服务的一致性，所有节点使用共享存储设备。



➢第一层 负载调度器

是访问整个群集系统的唯一入口，

对外使用所有服务器共有的VIP（VirtualIP， 虚拟IP）地址，也称为群集IP地址

通常会配置主、备两台调度器实现热备份

当主调度器失效以后平滑替换至备用调度器，确保高可用性

➢第二层 服务器池

是整个集群中真正提供业务环境的（如：静动态网站）

群集所提供的应用服务(如，HTTP FTP)由服务器池承担

其中的每个节点具有独立的RIP（RealIP，真实IP)地址

只处理调度器分发过来的客户机请求

当某个节点暂失效时，负载调度器的容错机制会将其隔离

等待错误排除以后再重新纳入服务器池

➢第三层　共享存储

为服务器池中的所有节点提供稳定、一致的文件存取服务

确保整个群集的统一性

在Linux/UNIX 环境中，共享存储可以使用NAS设备

或者提供NFS (Network File System,网络文件系统)共享服务的专用服务器

二、LVS虚拟服务器

Linux Virtual Server是针对Linux内核开发的一个负载均衡项目，

由我国的章文嵩博士在1998年5月创建 ----国人开发

阿里就职首席架构师

官方站点位于http://www .linuxvirtualserver.org/

LVS实际上相当于基于IP地址的虚拟化应用

为基于IP地址和内容请求分发的负载均衡提出了一种高效的解决方法

LVS现在已成为Linux内核的一部分，默认编译为ip_vs 模块，必要时能够自动调用

在CentOS7系列系统中，以下操作可以手动加载ip_vs模块

并查看当前系统中ip_vs模块的版本信息

```
[root@localhost ~]# yum -y install ipvsadm              //安装ipvsadm管理工具
[root@localhost ~]# modprobe ip_ vs                     //加载ip_vs模块
[root@localhost ~]# IsmodI | grep ^ip_vs                //查看ip_vs 模块信息。
ip_vs           141432           0
[root@localhost ~]# cat /proc/net/ip_vs                 //查看ip_vs 版本信息
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
-> RemoteAddress:Port Forward Weight ActiveConn InActConn
```

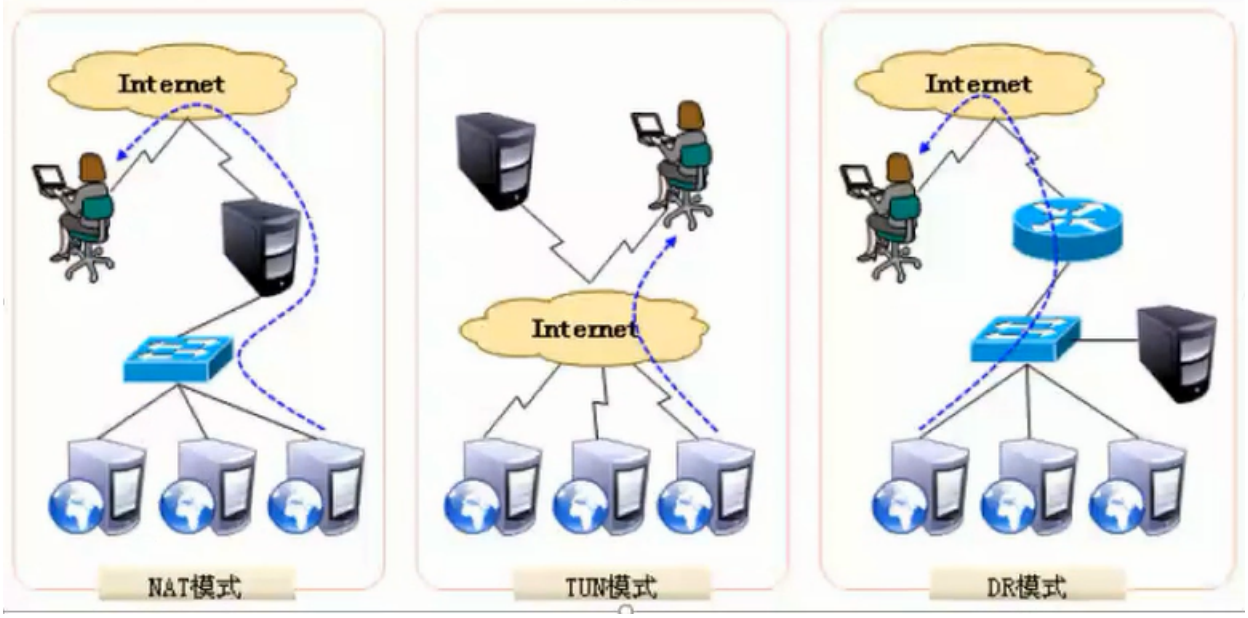
LVS负载均衡的工作模式

关于群集的负载调度技术，可以基于IP、端口、内容等进行分发，

其中基于IP的负载调度是效率最高的

基于IP的负载均衡模式中，

工作模式:地址转换(NAT) 、IP隧道(TUNNEL) 、直接路由(DR) 、FULLNAT



➢**地址转换(Network Address Translation) :简称NAT模式**

类似于防火墙的私有网络结构

负载调度器作为所有服务器节点的网关

即作为客户机的访问入口，也是各节点回应客户机的访问出口

服务器节点使用私有IP地址，与负载调度器位于同一个物理网络

安全性要优于其他两种方式

缺点是整个架构的负载不高(调度器压力大)

用户访问数据的进出都在调度服务器身上，是个瓶颈

➢**IP隧道(IP Tunnel) :简称TUN模式**

性能和并发在所有模式中最好，但成本也是最高的，基本上很少用

采用开放式的网络结构

负载调度器仅作为客户机的访问入口

各节点通过各自的Internet连接直接回应客户机

而不再经过负载调度器

服务器节点分散在互联网中的不同位置

每个机器都具有独立的公网IP地址

通过专用IP隧道（IPIP协议）与负载调度器相互通信

➢**直接路由(Direct Routing) :简称DR 模式**

采用半开放式的网络结构，与TUN模式的结构类似，
但各节点并不是分散在各地，而是与调度器位于同一个物理网络
负载调度器与各节点服务器通过本地网络连接，不需要建立专用的IP隧道

以上三种工作模式中，
NAT方式只需要一个公网IP地址
从而成为最易用的一种负载均衡模式，安全性也比较好
许多硬件负载均衡设备就是采用这种方式
相比较而言，DR模式和TUN模式的负载能力更加强大、适用范围更广
但节点的安全性能要稍差一些

下面将介绍LVS所支持的主要负载调度算法
以及在LVS负载均衡调度器上如何使
用ipvsadm管理工具

三、LVS 的负载均衡调度算法

关注每一个算法的简称（小写），很重要，在配置中，主要写它的简写
前4个必须记住
LVS官方推出10种算法，
有很多地方说是8种算法，但是大部分人是不能全部说不出来8种的，原因：
使用的不多；

1、轮叫调度(Round Robin) (rr)

调度器通过“轮叫”调度算法
将外部请求按顺序轮流分配到集群中的真实服务器上（你一个；我一个；它一个）
它均等地对待每一台服务器，
而不管服务器上实际的连接数和系统负载

2、加权轮叫(Weighted Round Robin) (wrr)

调度器通过“加权轮叫”调度算法
根据真实服务器的不同处理能力来调度访问请求
（比如权值为1:2；你处理一个，我处理两个）
这样可以保证处理能力强的服务器能处理更多的访问流量
调度器可以自动问询真实服务器的负载情况，并动态地调整其权值

3、最少链接(Least Connections) (lc)

调度器通过“最少连接”调度算法

动态地将网络请求调度到已建立的链接数最少的服务器上

如果集群系统的真实服务器具有相近的系统性能

采用“最小连接”调度算法可以较好地均衡负载

4、加权最少链接(Weighted Least Connections) (wlc)

很多企业用wcl比较多

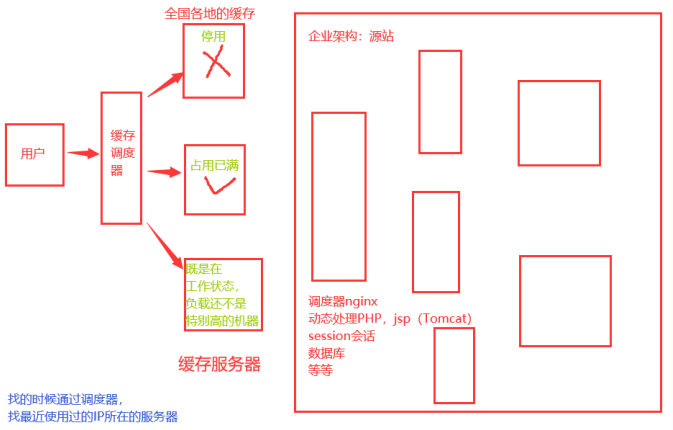
在集群系统中的服务器性能差异较大的情况下

调度器采用“加权最少链接”调度算法优化负载均衡性能

具有较高权值的服务器将承受较大比例的活动连接负载

调度器可以自动问询真实服务器的负载情况，并动态地调整其权值

5、基于局部性的最少链接(Locality-Based Least Connections) (lblc)



是针对目标IP地址的负载均衡

目前主要用于Cache（缓存）集群系统

该算法根据请求的目标IP地址找出该目标IP地址最近使用的服务器

若该服务器是可用的且没有超载

将请求发送到该服务器;

若服务器不存在，或者该服务器超载且有服务器处于一半的工作负载

则用“最少链接” 的原则选出一个可用的服务器，将请求发送到该服务器

6、带复制的基于局部性最少链接(Locality-Based Least Connections with Replication) (lblcr)

是针对目标IP地址的负载均衡，

目前主要用于Cache（缓存）集群系统。

lblcr与LBLC算法的不同之处是

lblcr要维护从一个目标IP地址到一组服务器的映射，

而LBLC算法维护从一个目标IP地址到一台服务器的映射。

该算法根据请求的目标IP地址找出该目标IP地址对应的服务器组，

按“最小连接”原则从服务器组中选出一台服务器，
若服务器没有超载，将请求发送到该服务器;
若服务器超载，则按“最小连接”原则从这个集群中选出一台服务器，
将该服务器加入到服务器组中,将请求发送到该服务器。
同时，当该服务器组有一段时间没有被修改，
将最忙的服务器从服务器组中删除，以降低复制的程度

7、目标地址散列(Destination Hashing) (dh)

用在缓存比较多
根据请求的目标IP地址，
作为散列键(HashKey)
从静态分配的散列表找出对应的服务器，
若该服务器是可用的且未超载，将请求发送到该服务器，
否则返回空（目标服务器出问题了）

8、源地址散列(Source Hashing) (sh)

类似于ip hash
根据请求的源IP地址,
作为散列键(HashKey)从静态分配的散列表找出对应的服务器，
若该服务器是可用的且未超载，将请求发送到该服务器，否则返回空。

9、最短的期望的延迟(Shortest Expected Delay Scheduling SED) (sed)

基于wlc算法。
举例:
ABC三台机器分别权重123 ,连接数也分别是123
那么如果使用WLC算法的话一个新请求进入时它可能会分给ABC中的任意一个。
使用sed算法后会进行这样一个运算: （作为了解）
A(1+1)/1
B(1+2)/2
C(1+3)/3
根据运算结果，把连接交给C。

10、最少队列调度(Never Queue Scheduling NQ) (nq)

无需队列。如果有台realserver 的连接数=0就直接分配过去，不需要在进行sed运算

四、使用ipvsadm管理工具

ipvsadm是在负载调度器上使用的LVS群集管理工具，

通过调用ip_vs模块来添加、删除服务器节点，以及查看群集的运行

```
[root@localhost ~]# rpm -q ipvsadm     # 查看软件包是否安装
ipvsadm-1.27-7.el7.x86_ 64
[root@localhost ~]# ipvsadm -V     #查看版本号
ipvsadm v1.27 2008/5/15 (compiled with popt and IPVS v1.2.1)
```

LVS群集的管理工作主要包括:

- 创建虚拟服务器
- 添加服务器节点
- 查看群集节点状态
- 删除服务器节点
- 保存负载分配策略

下面分别展示使用ipvsadm命令的操作方法

1)创建虚拟服务器

若群集的VIP地址为192.168.200.254，

针对TCP 80端口提供负载分流服务，使用的调度算法为轮询，

则对应的ipvsadm命令操作如下所示。

对于负载均衡调度器来说，VIP必须是本机实际已启用的IP地址

```
[root@localhost ~]# ipvsadm -A -t 192.168.200.254:80 -s rr
     #这条命令主要在调度器上敲，包括后面的都是在调度器上敲的，只是命令的含义不同
```

上述操作中，

选项-A表示添加虚拟服务器，

-t 用来指定VIP地址及TCP端口，

-s 用来指定负载调度算法——轮询 (rr)、加权轮询(wrr) 、最少连接(lc)、加权最少连(wlc)等

2)添加服务器节点

为虚拟服务器192.168.200.254添加4个服务器节点，IP地址依次为192.168.200.112、192.168.200.113、192.168.200.114、192.168.200.115，

对应的ipvsadm命令操作如下所示。

若希望使用保持连接，还应添加“-p 60”选项，其中60为保持时间(秒)

```
[root@localhost ~]# ipvsadm -a -t 192.168.200.254:80 -r 192.168.200.112:80 -m -w 1
[root@localhost ~]# ipvsadm -a -t 192.168.200.254:80 -r 192.168.200.113:80 -m -w 1
[root@localhost ~]# ipvsadm -a -t 192.168.200.254:80 -r 192.168.200.114:80 -m -w 1
[root@localhost ~]# ipvsadm -a -t 192.168.200.254:80 -r 192.168.200.115:80 -m -w 1
```

上述操作中，

选项

-a　　表示添加服务器

-t　　用来指定VIP地址及TCP端口

-r　　用来指定RIP地址及TCP端口

-m　　表示使用NAT群集模式(-g DR模式、-i TUN模式)

-w　　用来设置权后重(权重为0时表示暂停节点)

3)查看群集节点状态

结合选项 -L可以列表查看LVS虚拟服务器

可以指定只查看某一个VIP地址(默认为查看所有)

结合选项 -n 将以数字形式显示地址、端口等信息（这样速度比较快）

-c 表示链接

上述输出结果中，

Forward 列下的Masq对应的Masquerade (地址伪装)，表示采用的群集模式为NAT

如果是Route,则表示采用的群集模式为DR

4)删除服务器节点

需要从服务器池中删除某一个节点时，

使用选项-d （把-a 换成 -d 就可以）

执行删除操作必须指定目标对象，包括节点地址、虚拟IP地址。

例如，以下操作将会删除LVS群集192.168.200.254 中的节点192.168.200.115

```
[root@localhost ~]# ipvsadm -d   -t 192.168.200.254:80 -r 192.168.200.115:80
```

需要删除整个虚拟服务器时，

使用选项-D并指定虚拟IP地址即可，无需指定节点。

例如若执行“ipvsadm -D -t 192. 168.200.254.80”则删除此虚拟服务器

5)保存负载分配策略

使用导出/导入工具ipvsadm-save/ipvsadm-restore可以保存、恢复LVS策略。

当然也可以快速清除、重建负载分配策略

```
[root@localhost ~]# ipvsadm-save > /etc/sysconfig/ipvsadm //保存策略
[root@localhost ~]# cat /etc/sysconfig/ipvsadm //确定保存结果
-A -t 192.168.200.254:http -s rr
-a -t 192.168.200.254:http -r minion-one:http -m -w 1
-a -t 192.168.200.254:http -r 192.168.200.113:http -m -w 1
-a -t 192.168.200.254:http -r 192.168.200.114:http -m -w 1
[root@localhost ~]# systemctl stop ipvsadm //停止服务（清除策略)
[root@localhost ~]# systemctl start ipvsadm //起动服务（重建规则）
```

五、实验--构建LVS负载均衡集群

环境准备：

克隆1（192.168.200.107）：添加网卡（VMware2）

windows：网络适配器：VMware2

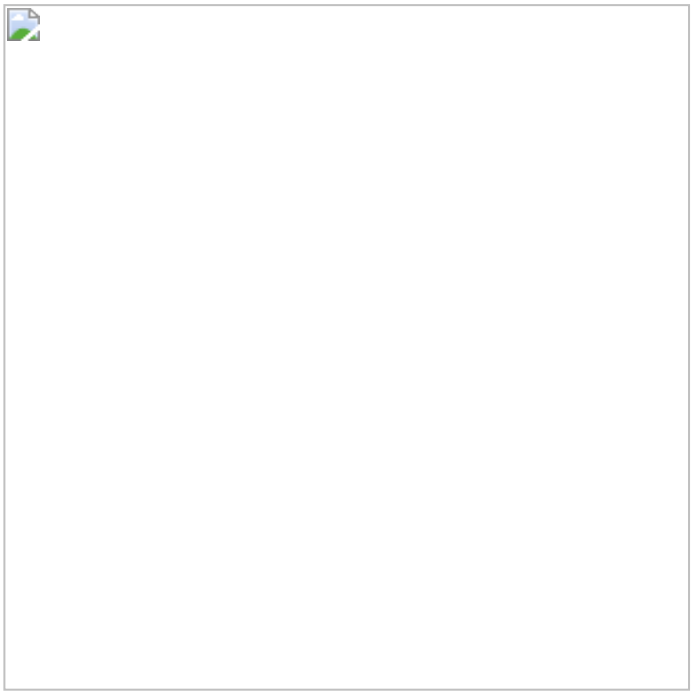
1、地址转换模式(LVS-NAT)

在NAT模式的群集中，

LVS负载调度器是所有节点服务器访问Internet的网关服务器，

其外网地址172.16.1.1 同时也作为整个群集的VIP地址，

LVS 调度器具有两块网卡，分别连接内外网



1.关闭防火墙：

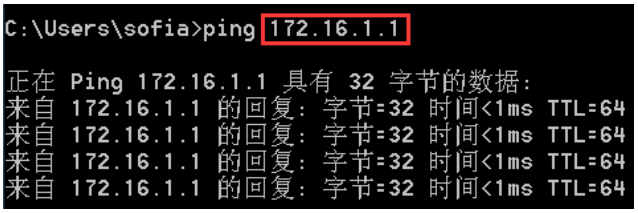
```
[root@localhost ~]# iptables -F
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# setenforce 0
setenforce: SELinux is disabled
```

2.配置192.168.200.107的网卡IP

```
[root@localhost ~]# ls /etc/sysconfig/network-scripts/
[root@localhost ~]# ifconfig
[root@localhost ~]# cd /etc/sysconfig/network-scripts/
[root@localhost network-scripts]# ls
[root@localhost network-scripts]# cp ifcfg-ens32 ifcfg-ens34
[root@localhost network-scripts]# vim ifcfg-ens34
NAME=ens34
DEVICE=ens34
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.16.1.1
NETMASK=255.255.255.0
[root@localhost network-scripts]# ifdown ens34
成功断开设备 'ens34'。
[root@localhost network-scripts]# ifup ens34
连接已成功激活（D-Bus 活动路
径：/org/freedesktop/NetworkManager/ActiveConnection/17）
[root@localhost ~]# ifconfig
ens34:       inet 172.16.1.1
```

3.配置windows7（客户机）的IP：模拟外网

测试连通性：



4.指定192.168.200.108及192.168.200.109的网关：

```
[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens32
GATEWAY=192.168.200.107
```

5.开启路由转发

对于LVS负载调度器来说，需开启路由转发规则

以便节点服务器能够访问Internet

所有的节点服务器，共享存储均位于私有网络结构

其默认网关设为LVS负载调度器的内网地址(192.168.200.111)

```
[root@localhost ~]# vim /etc/sysctl.conf
net.ipv4.ip_forward=1       # 末尾加入
[root@localhost ~]# sysctl -p     # 生效
net.ipv4.ip_forward = 1
```

6.配置负载分配策略

```
[root@localhost ~]# ipvsadm -C
[root@localhost ~]# rpm -q ipvsadm
未安装软件包 ipvsadm
[root@localhost ~]# rpm -ivh /media/cdrom/Packages/ipvsadm-1.27-7.el7.x86_64.rpm
[root@localhost ~]# modprobe ip_vs    # 加载模块
[root@localhost ~]# ipvsadm -A -t 172.16.1.1:80 -s rr
[root@localhost ~]# ipvsadm -a -t 172.16.1.1:80 -r 192.168.200.108:80 -m -w 1
[root@localhost ~]# ipvsadm -a -t 172.16.1.1:80 -r 192.168.200.109:80 -m -w 1
```

到这集群已经做完了~

调度器做到这，实验已经成了

```
[root@localhost ~]# ipvsadm -Ln          # 查看
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  172.16.1.1:80 rr
  -> 192.168.200.108:80           Masq    1      0          0
  -> 192.168.200.109:80           Masq    1      0          0
```

7.配置192.168.200.108和192.168.200.109的网络服务：

```
[root@localhost ~]# rpm -q httpd
[root@localhost ~]# yum -y install httpd
[root@localhost ~]# systemctl start httpd
[root@localhost ~]# echo "server1" > /var/www/html/index.html
# 192.168.200.108定义网页
[root@localhost ~]# echo "server2" > /var/www/html/index.html
# 192.168.200.109定义网页
```

8.测试：172.16.1.1

调度器：192.168.200.107

```
[root@localhost ~]# ipvsadm -Lnc
IPVS connection entries
pro expire state       source             virtual            destination
TCP 14:58  ESTABLISHED 172.16.1.2:49164   172.16.1.1:80      192.168.200.108:80
```

9.保存负载分配策略

使用导出/导入工具ipvsadm- save/ipysadm-restore可以保存、恢复LVS策略

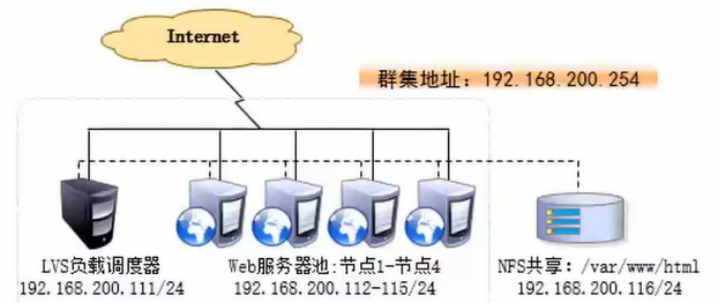
当然也可以快.速清除、重建负载分配策略

```
[root@localhost ~]# ipvsadm-save -n > /etc/sysconfig/ipvsadm
# 有时候不好使的时候，用ipvsadm-save -n （不加-n 不会以数字的形式显示IP地址）
[root@localhost ~]# cat /etc/sysconfig/ipvsadm
-A -t localhost:http -s rr
-a -t localhost:http -r 192.168.200.108:http -m -w 1
-a -t localhost:http -r 192.168.200.109:http -m -w 1
```

10.删除负载分配策略

```
[root@localhost ~]# ipvsadm -C
[root@localhost ~]# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
[root@localhost ~]# systemctl start ipvsadm    （它是写到文档里变永久了吗）
[root@localhost ~]# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  127.0.0.1:80 rr
  -> 192.168.200.108:80           Masq    1      0          0
  -> 192.168.200.109:80           Masq    1      0          0
```

2、直接路由模式(LVS-DR)



在DR模式的群集中，LVS负载调度器作为群集的访问入口，但不作为网关使用;
服务器池中的所有节点都各自接入Internet,
节点服务器发送给客户机的Web响应数据包不需要经过LVS负载调度器

这种方式入站、出站访问数据被分别处理，
因此LVS负载调度器和所有的节点服务器都需要配置有VIP地址，
以便响应对整个群集的访问。　　----问题一，二
考虑到数据存储的安全性，共享存储设备会放在内部的专用网络中

实验：　一台调度器，两台web服务器

还原环境：

```
[root@localhost ~]# ipvsadm -C
[root@localhost ~]# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
[root@localhost ~]# ipvsadm-save -n > /etc/sysconfig/ipvsadm
[root@localhost ~]# vim /etc/sysctl.conf
#net.ipv4.ip_forward=1 （注释）
[root@localhost ~]# sysctl -p    # 使生效
192.168.200.10恢复指定的默认网关
[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens32
GATEWAY=192.168.200.1
[root@localhost ~]# systemctl restart network
[root@localhost ~]# route -n            # 192.1638.200.108（9）两台服务器都做
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.200.1   0.0.0.0         UG    100    0        0 ens32
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
192.168.200.0   0.0.0.0         255.255.255.0   U     100    0        0 ens32
[root@localhost ~]# systemctl start httpd
[root@localhost ~]# systemctl restart httpd
```

正式开始实验：

1) 所有主机关闭防火墙和selinux:

```
[root@localhost ~]# systemctl stop firewallde
```

```
[root@localhost ~]# iptables -F
[root@localhost ~]# setenforce 0
```

2)配置负载调度器192.168.200.107

配置虚拟IP地址(VIP) 采用虚拟接口的方式(ens32:0)

为网卡ens32绑定VIP地址，以便响应群集访问

```
[root@localhost ~]# yum -y install ipvsadm
[root@localhost ~]# ifconfig ens32:0 192.168.200.254 netmask 255.255.255.0
```

注意在生产环境中要把它做成一个永久（192.168.200.254）

就是再复制一个ifcfg-ens32:0的一个文件，在里面配置一个IP，做成一个永久的就可以了

这是和内部通信的DIP

```
[root@localhost ~]# ifconfig ens32
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.200.107  netmask 255.255.255.0  broadcast
192.168.200.255
```

这是和外部通信的VIP

```
[root@localhost ~]# ifconfig ens32:0
ens32:0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.200.254  netmask 255.255.255.0  broadcast
192.168.200.255
```

3)配置负载调度的策略

```
[root@localhost ~]# ipvsadm -A -t 192.168.200.254:80 -s rr
[root@localhost ~]# ipvsadm -a -t 192.168.200.254:80 -r 192.168.200.108:80 -g -w
1
[root@localhost ~]# ipvsadm -a -t 192.168.200.254:80 -r 192.168.200.109:80 -g -w
1
[root@localhost ~]# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.200.254:80 rr
  -> 192.168.200.108:80           Route   1      0          0
  -> 192.168.200.109:80           Route   1      0          0
```

到这集群就好了~

4)配置节点服务器192.168.200.108 192.168.200.109

使用DR模式时，节点服务器也需要配置VIP地址，

并调整内核的ARP响应参数以阻止更新VIP的MAC地址，避免发生冲突。

除此之外，Web服务的配置与NAT方式类似

在每个节点服务器，同样需要有VIP地址192.168.200.254，

但此地址仅用作发送Web响应数据包的源地址，

并不需要监听客户机的访问请求(改由调度器监听并分发)。

因此使用虚拟接口lo:0来承载VIP地址，并为本机添加一条路由记录，

将访问VIP的数据限制在本地以避免通信紊乱

```
[root@localhost ~]# ifconfig lo:0 192.168.200.254 netmask 255.255.255.255
[root@localhost ~]# ifconfig lo:0
lo:0: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 192.168.200.254  netmask 255.255.255.255
        loop  txqueuelen 1000  (Local Loopback)
[root@localhost ~]# route add -host 192.168.200.254 dev lo:0
```

5)安装httpd，写网页，重启服务

```
[root@localhost ~]# yum -y install httpd
[root@localhost ~]# systemctl start httpd
[root@localhost ~]# echo "server1" > /var/www/html/index.html
[root@localhost ~]# systemctl restart httpd
```

6)调整内核ARP响应参数 (在108和109上配置)

```
[root@localhost  ~]#  vim  /etc/sysctl.conf
net.ipv4.conf.all.arp_ignore  =  1
net.ipv4.conf.all.arp_announce  =  2
net.ipv4.conf.default.arp_ignore  =  1
net.ipv4.conf.default.arp_announce  =  2
net.ipv4.conf.lo.arp_ignore  =  1
net.ipv4.conf.lo.arp_announce  =  2
[root@localhost  ~]#  sysctl  -p
```

➢arp ignore=1

系统只回答目的IP为是本地IP的包,也就是对广播包不做响应

➢arp_ announce=2

系统忽略IP包的源地址(source address)，而根据目标主机(target host)，选择本地地址

7)测试LVS群集

安排多台测试机，从Internet中 直接访问http://192.168.200.254/

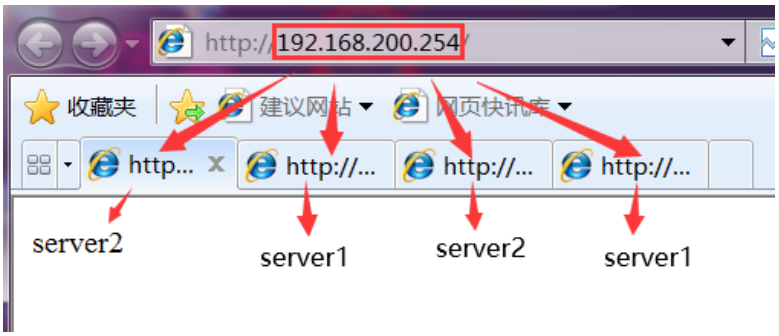
将能够看到由真服务器提供的网页内容

如果各节点的网页不同，则不同客户机看到的网页可能也不一样(可以多刷新几次)

在LVS负载调度器中，通过查看节点状态可以观察当前的负载分配情况，

对于轮询算法来说，每个节点所获得的连接负载应大致相当

注意：将win7改成VMware8（即默认NAT模式）



六、NFS共享存储服务

NFS (Network File System)是一种基于TCP/IP传输的网络文件系统协议

最初由SUN公司开发

通过使用NFS协议，客户机可以像访问本地目录一样访问远程服务器中的共享资源

对于大多数负载均衡群集来说，使用NFS协议来共享数据存储是比较常见的做法

NFS也是NAS存储设备必然支持的一种协议

1、使用NFS发布共享资源

NFS服务的实现依赖于RPC (Remote Process Call,远程过程调用)机制，

（将别的远程服务器的数据映射到本地的一个过程）

以完成远程到本地的映射过程。

在CentOS7系统中， 需要安装nfs-utils、rpcbind 软件包来提供NFS共享服务，

nfs-utils 用于NFS共享发布和访问，rpcbind 用于RPC支持

1)安装nfs-utils rpcbind软件包

提供RPC支持的服务为rpcbind，

提供NFS共享的服务为nfs,完成安装以后建议调整

这两个服务的自启动状态，

以便每次开机后自动启用。手动加载NFS共享服务时，

应该先启动rpcbind，然后再启动nfs（尤其是在老的系统里面）

查看服务：

```
[root@NFS ~]# rpm -q nfs-utils rpcbind
nfs-utils-1.3.0-0.54.el7.x86_64
rpcbind-0.2.0-44.el7.x86_64
```

启动服务：

```
[root@NFS ~]# systemctl enable rpcbind
[root@NFS ~]# systemctl enable nfs
```

2)设置共享目录

NFS的配置文件为/etc/exports，文件内容默认为空(无任何共享)----都需要自己去写

在服务器192. 168. 200. 110中的exports文件中设置共享资源时

记录格式为　　目录位置　　客户机地址　　(权限选项)

目录位置 ：你要共享的目录的位置 ，具体你要把什么目录对外做共享

客户机地址：授权哪些客户机地址可以访问

(权限选项)：这些客户有哪些权限

例如，若要将文件夹/opt/wwwroot共享给m2.168.200.0/24网段使用，

允许读写操作，配置如下所示

```
[root@NFS ~]# mkdir /opt/wwwroot
[root@NFS ~]# vim /etc/exports
/opt/wwwroot 192.168.200.0/24(rw,sync,no_root_squash)
```

其中客户机地址可以是主机名，IP地址、网段地址，允许使用*、?通配符

权限选项中的rw表示允许读写(ro 为只读)，sync 表示同步写入

no_root_squash表示当客户机以root身份访问时赋予本地root权限

(默认是root_squash将作为nfsnobody用户降权对待)

当需要将同一个目录共享给不同的客户机，且分配不同的权限时

只要以空格分隔指定多个“客户机(权限选项)”即可

例如，以下操作将/var/ftp/public目录共享给两个客户机

并分别给予只读、读写权限

3)启动NFS服务程序

```
[root@NFS ~]# systemctl restart rpcbind
[root@NFS ~]# systemctl restart nfs
```

4)查看本机发布的NFS共享目录

```
[root@NFS ~]# showmount -e
Export list for NFS:
/opt/wwwroot 192.168.200.0/24
```

对于客户机来说（192.168.200.109）：也可以看到列表

```
[root@localhost ~]# showmount -e 192.168.200.110
Export list for 192.168.200.110:
/opt/wwwroot 192.168.200.0/24
```

2、在WEB Server中访问NFS共享资源

NFS协议的目标是提供一种网络文件系统，

因此对NFS共享的访问也使用mount命令来进行挂载，对应的文件系统类型为nfs，

即可以手动挂载，也可以加入fstab配置文件，来实现开机自动挂载，

考虑到群集系统中的网络稳定性，NFS服务器与客户机之间最好使，用专有网络进行连接。

在192.168.200.108和192.168.200.109上挂载：

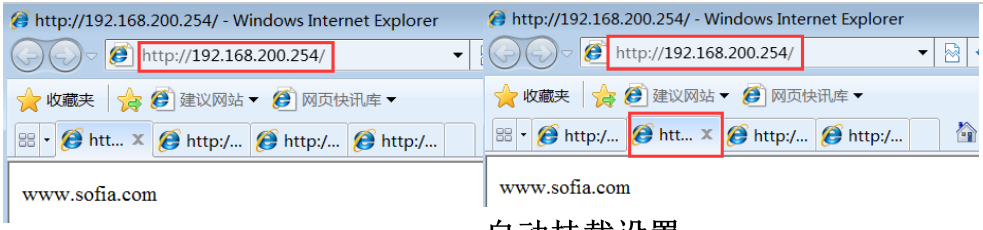
```
[root@localhost ~]# mount 192.168.200.110:/opt/wwwroot /var/www/html/
```

在NFS机器上配置：

```
[root@NFS ~]# echo "www.sofia.com" > /opt/wwwroot/index.html
[root@NFS ~]# cat /opt/wwwroot/index.html
```

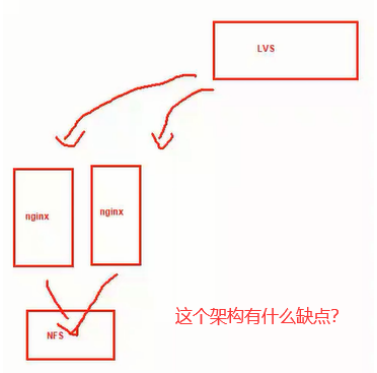
在192.168.200.108和192.168.200.109上查看：

```
[root@localhost ~]# cat /var/www/html/index.html
www.sofia.com
```



自动挂载设置：

```
[root@localhost ~]# vim /etc/fstab
192.168.200.110:/opt/wwwroot        /var/www/html    nfs    defaults,_netdev    0 0
```



1、nfs单点（nfs+rsync、nfs+drbd+heartbeat）

drdb拿两个机器的各自的一个硬盘给它们做了一个同步

heartbeat 对外提供了一个VIP

NFS故障解决在这

2、NFS没有用户校验、所以只能用内网中（不要暴露在公网上）

就写了一个/etc/exports文件

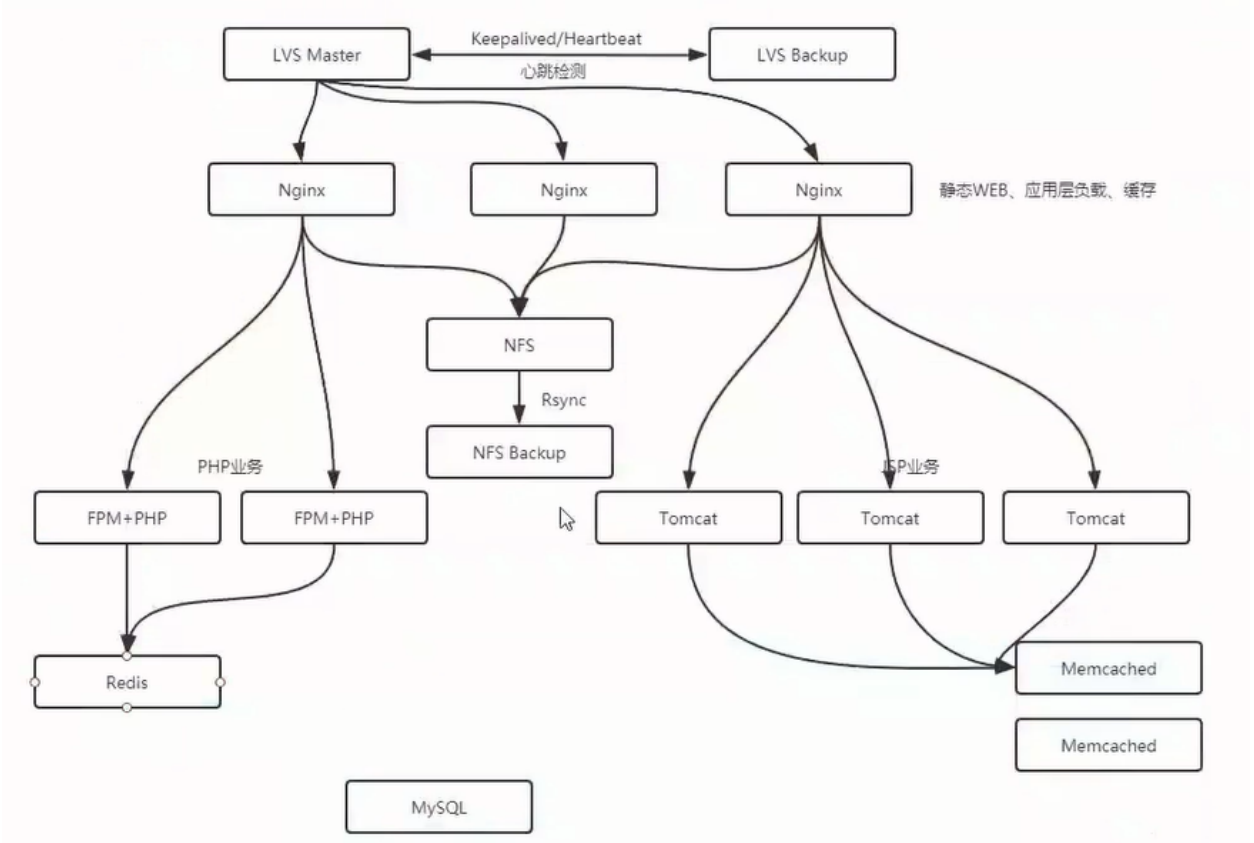
它没有用户验证，也不需要用户登录

只要是200网段的主机，它就有授权

只要别人能配上200网段，来模拟这个网段的主机，它就有授权权限

所以最好的办法就是别把它暴露在公网上

面试---流程图



keepalived/heartbeat （心跳检测） ：高可用

nginx ： 静态web ，调度器（7层负载） 网站缓存

FPM-PHP ： 处理PHP的业务

Tomcat :处理 jsp 网页

NFS :备份 rsync 数据同步

Redis memcache :写入

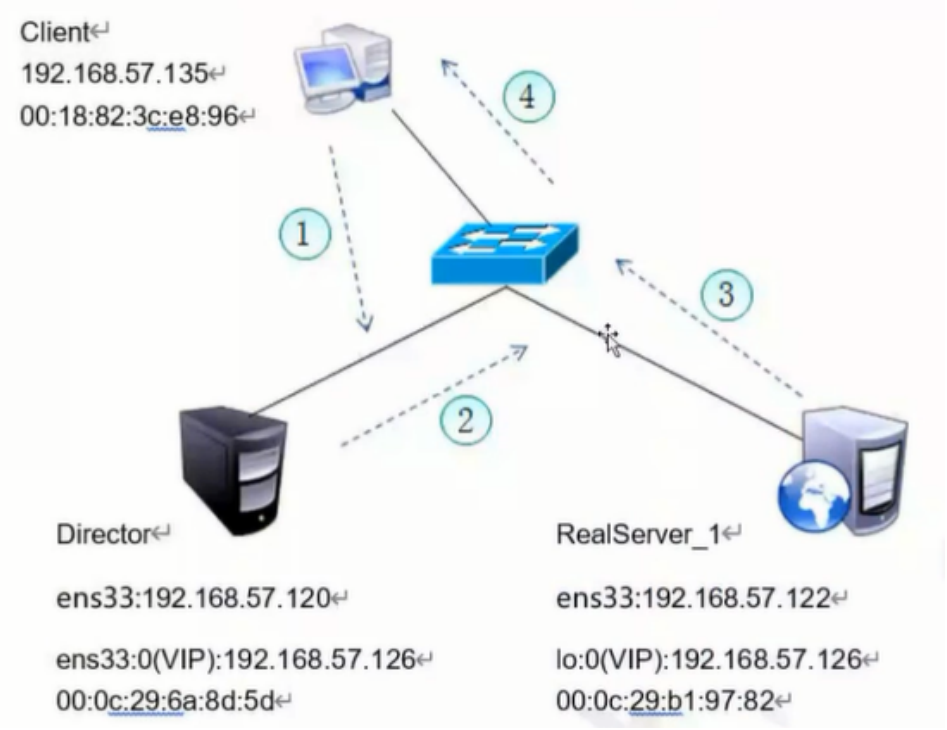
Mysql : 数据库

七、LVS DR模式数据包流向分析

为方便进行原理分析，

将Client与群集系统的所有机器放在同一网络中，

数据包流经的路线为1-2-3-4

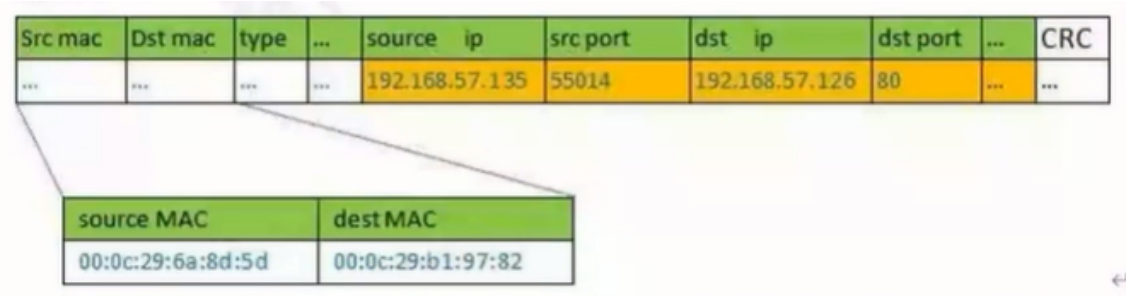


1、Client 向目标VIP发出请求，Director (负载均衡器)接收。此时IP包头及数据帧头信息为：

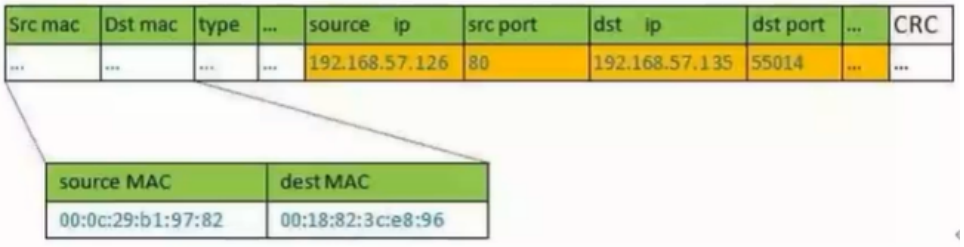
Src mac	Dst mac	type	...	source ip	src port	dst ip	dst port	...	CRC
...	...	...	...	192.168.57.135	55014	192.168.57.126	80	...	...

source MAC	dest MAC
00:18:82:3c:e8:96	00:0c:29:6a:8d:5d

2、Director 根据负载均衡算法选择RealServer_ 1,不修改也不封装IP报文，而是将数据帧的MAC地址改为RealServer_ 1的MAC地址，然后在局域网上发送。IP 包头及数据帧头信息如下:



3.、RealServer_1收到这个帧，解封装后发现目标IP与本机匹配(RealServer事先绑定了VIP)， 于是处理这个报文。随后重新封装报文，发送到局域网。此时IP包头及数据帧头信息为:



4、Client将收到回复报文。Client认为得到正常的服务，而不会知道是哪一台服务器处理的

注意：如果跨网段，那么报文通过路由器经由Internet返回给用户。

八、LVS-DR中的ARP问题分析

ARP地址解析协议，将已知的IP地址解析为mac地址

问题：

在LVS-DR负载均衡集群中，

负载均衡器与节点服务器都要配置相同的VIP地址，

但是在局域网（ARP）中具有相同的IP地址，势必会造成各服务器ARP广播通信的紊乱

当一个ARP广播发送到LVS-DR集群时

因为负载均衡器和节点服务器都是连接到相同的网络上的

它们都会接收到ARP广播，此时应该只有前端的负载均衡器进行响应

而其他节点服务器不应该响应ARP广播

解决方法:

第一个问题：

对节点服务器进行处理，使其不响应针对VIP的ARP广播请求，

使用虚接口lo:0承载VIP地址，

设置内核参数arp ignore=1

系统只响应目的IP为本地RP的ARP请求

RealServer返回报文(源IP是VIP)经路由器转发

在重新封装报文时，需要先获取路由器的MAC地址

发送ARP请求时，Linux默认使用IP包的源IP地址(即VIPI作为ARP请求包中的源IP地址

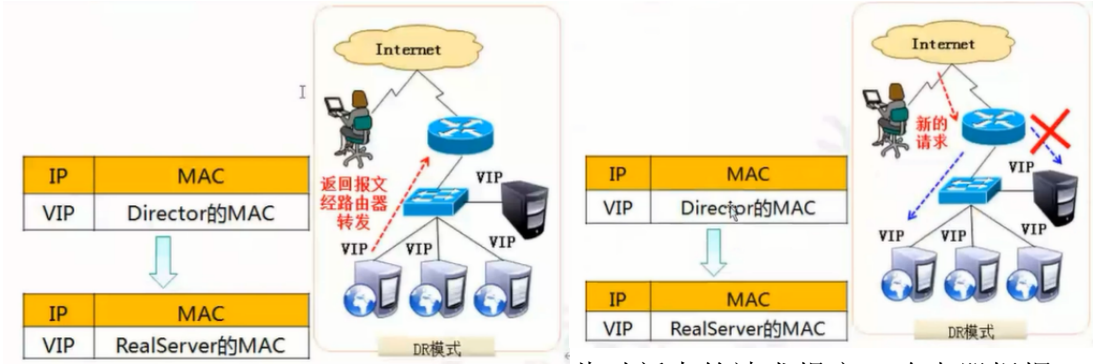
而不使用发送接口(例如ens33)的IP地址

源IP	VIP
源MAC	RealServer的MAC
目的IP	路由器的IP
目的MAC	?

问题：

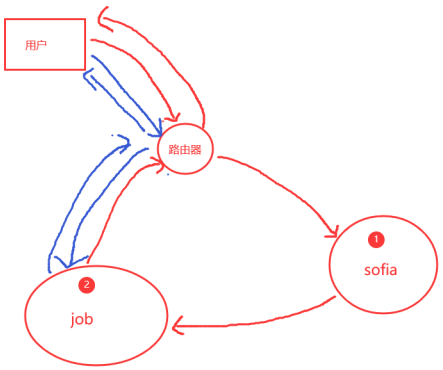
路由器收到ARP请求后，将更新ARP表内容，

原有的VIP对应Director的MAC地址会被更新为VIP对应RealServer的MAC地址



此时新来的请求报文，路由器根据ARP表项，

会将该报文转发给RealServer,从而导致Director的VIP失效



解决办法：

对节点服务器进行处理,设置内核参数arp_announce=2:

系统不使用IP包的源地址(VIP)

来作为ARP请求的源IP地址，而选择发送接口的IP (RIP) 地址

源IP：RIP

源MAC：RIP的MAC

目的IP：路由IP

目的MAC：路由MAC

解决ARP的两个问题的方法

```
[root@localhost ~]# vim /etc/sysctl.conf
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.default.arp_ignore = 1
net.ipv4.conf.default.arp_announce = 2
net.ipv4.conf.lo.arp_ignore = 1
net.ipv4.conf.lo.arp_announce = 2
[root@localhost ~]# sysctl -p
```

总结：

1.节点服务器配置到lo口上，并设置arp_ignore=1来解决ARP广播问题

2.节点服务器arp_announce = 2来解决节点服务器向路由器发送arp请求，

将使用自己的RIP地址，而不是VIP地址，防止路由器更新ARP缓存表时，出现问题

面试题

---　中高级运维，对集群的理解很重要

---　注意：将详细的MAC地址转换过程，画出来