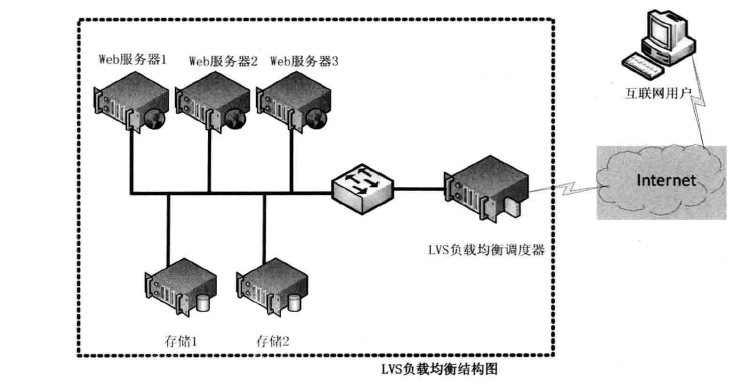
LVS服务器集群

# LVS负载均衡简介

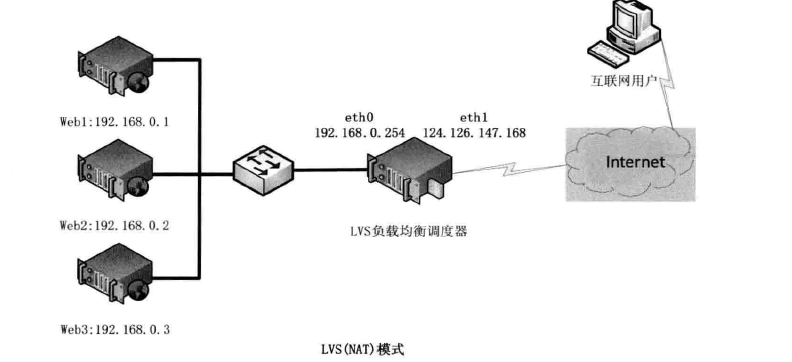
LVS是Linux Virtual Server的简写，意即Linux虚拟服务器，是一个虚拟的服务器集群系统。本项目在1998年5月由章文嵩博士成立，是中国国内最早出现的自由软件项目之一。目前LVS已经集成到linux内核中。

我们先分析实现虚拟网络服务的主要技术，指出IP负载均衡技术是在负载调度器的实现技术中效率最高的。在已有的IP负载均衡技术中，主要有通过网络地址转换（Network Address Translation）将一组服务器构成一个高性能的、高可用的虚拟服务器，我们称之为VS/NAT技术（Virtual Server via Network Address Translation）。在分析VS/NAT的缺点和网络服务的非对称性的基础上，我们提出了通过IP隧道实现虚拟服务器的方法VS/TUN （Virtual Server via IP Tunneling），和通过直接路由实现虚拟服务器的方法VS/DR（Virtual Server via Direct Routing），它们可以极大地提高系统的伸缩性。VS/NAT、VS/TUN和VS/DR技术是LVS集群中实现的三种IP负载均衡技术。LVS主要组成部分为：

* 负载调度器(load balancer/ Director)，它是整个集群对外面的前端机，负责将客户的请求发送到一组服务器上执行，而客户认为服务是来自一个IP地址(我们可称之为虚拟IP地址)上的。
* 服务器池(server pool/ Realserver)，是一组真正执行客户请求的服务器，执行的服务一般有WEB、MAIL、FTP和DNS等。
* 共享存储(shared storage)，它为服务器池提供一个共享的存储区，这样很容易使得服务器池拥有相同的内容，提供相同的服务。



# 基于NAT的LVS负载均衡

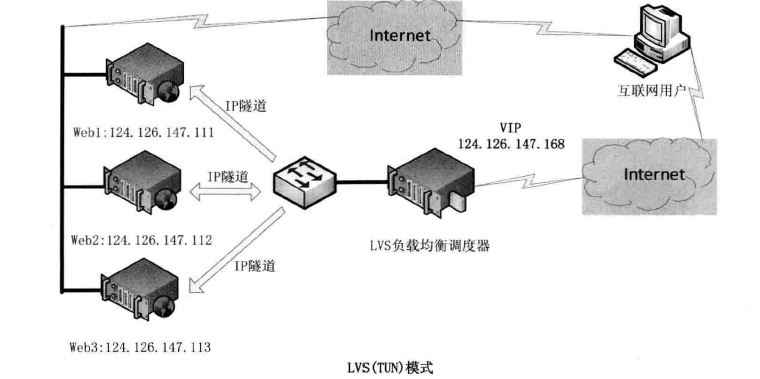


原理：就是把客户端发来的数据包的IP头的目的地址，在负载均衡器上换成其中一台RS的IP地址，并发至此RS来处理,RS处理完成后把数据交给经过负载均衡器,负载均衡器再把数据包的原IP地址改为自己的IP，将目的地址改为客户端IP地址即可｡期间,无论是进来的流量,还是出去的流量,都必须经过负载均衡器｡

优点：集群中的物理服务器可以使用任何支持TCP/IP操作系统，只有负载均衡器需要一个合法的IP地址。

缺点：扩展性有限。当服务器节点（普通PC服务器）增长过多时,负载均衡器将成为整个系统的瓶颈，因为所有的请求包和应答包的流向都经过负载均衡器。当服务器节点过多时，大量的数据包都交汇在负载均衡器那，速度就会变慢！

# 基于TUN的LVS负载均衡

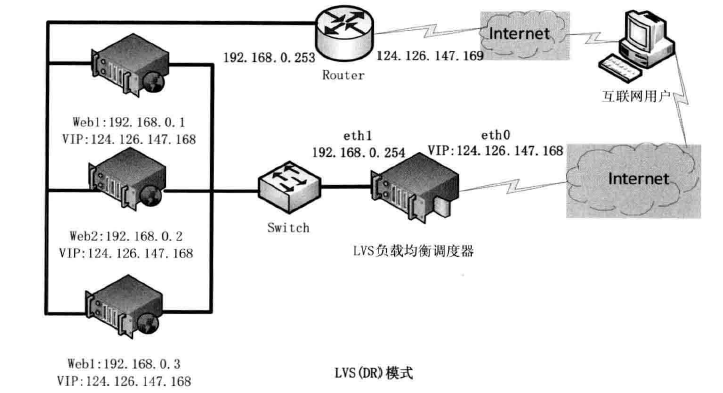


原理：互联网上的大多Internet服务的请求包很短小，而应答包通常很大。隧道模式的思路就是讲将请求和相应数据分离，让调度器处理数据请求，让后面的真实服务器将相应包直接发送给客户端。客户端发来的数据包，封装一个新的IP头标记(仅目的IP)发给RS,RS收到后,先把数据包的头解开,还原数据包,处理后,直接返回给客户端,不需要再经过负载均衡器｡注意,由于RS需要对负载均衡器发过来的数据包进行还原,所以说必须支持IPTUNNEL协议｡所以,在RS的内核中,必须编译支持IPTUNNEL这个选项

优点：负载均衡器只负责将请求包分发给后端节点服务器，而RS将应答包直接发给用户。所以减少了负载均衡器的大量数据流动，负载均衡器不再是系统的瓶颈，就能处理很巨大的请求量，这种方式，一台负载均衡器能够为很多RS进行分发。而且跑在公网上就能进行不同地域的分发。

缺点：隧道模式的RS节点需要合法IP，这种方式需要所有的服务器支持”IP Tunneling”(IP Encapsulation)协议，服务器可能只局限在部分Linux系统上。

# 基于DR的LVS负载均衡



原理：LVS仅承担数据的入站请求以及根据算法选择合理的真实服务器，最终由后端的真实服务器来相应数据包返回给客户端。和隧道模式不同的是，直接路由模式DR要求调度器和后端服务器必须在同一个局域网内。VIP地址需要在调度器和后端所有服务器之间共享。

负载均衡器和RS都使用同一个IP对外服务｡但只有DR对ARP请求进行响应,所有RS对本身这个IP的ARP请求保持静默｡也就是说,网关会把对这个服务IP的请求全部定向给DR,而DR收到数据包后根据调度算法,找出对应的RS,把目的MAC地址改为RS的MAC（因为IP一致）并将请求分发给这台RS｡这时RS收到这个数据包,处理完成之后，由于IP一致，可以直接将数据返给客户，则等于直接从客户端收到这个数据包无异,处理后直接返回给客户端｡由于负载均衡器要对二层包头进行改换,所以负载均衡器和RS之间必须在一个广播域,也就是同一局域网｡

优点：和TUN（隧道模式）一样，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。与VS-TUN相比，VS-DR这种实现方式不需要隧道结构，因此可以使用大多数操作系统做为物理服务器。

# LVS负载均衡调度算法

## 轮叫调度(Round-RobinScheduling)

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,它均等对待每一台服务器,而不管服务器上实际的连接数和系统负载｡RR

## 加权轮叫调度(WeightedRound-RobinScheduling)

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｡这样可以保证处理能力强的服务器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｡WRR

## 最小连接调度(Least-ConnectionScheduling)

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｡

## 加权最小连接调度(WeightedLeast-ConnectionScheduling)

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

## 基于局部性的最少链接(Locality-BasedLeastConnectionsScheduling)

基于局部性的最少链接"调度算法是针对目标IP地址的负载均衡,目前主要用于Cache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｡

## 带复制的基于局部性最少链接(Locality-BasedLeastConnectionswithReplicationScheduling)

带复制的基于局部性最少链接"调度算法也是针对目标IP地址的负载均衡,目前主要用于Cache集群系统｡它与LBLC算法的不同之处是它要维护从一个目标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

## 目标地址散列调度(Destination HashingScheduling)

目标地址散列"调度算法根据请求的目标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。

## 源地址散列调度(Source HashingScheduling)

源地址散列"调度算法根据请求的源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｡

# 部署LVS服务

LVS已经集成在linux内核模块中，但整个LVS环境又分为内核层和用户层，内核层负责核心算法调度；用户层需要安装ipvsadm工具，通过命令将管理员需要的工作模式和实际的算法传递给内核来实现。LVS的内核模块名称为ip\_vs。

## ipvsadm安装

|  |
| --- |
| #yum安装  yum install ipvsadm -y  #源码安装  yum -y install gcc popt-devel popt-static libnl libnl-devel  wget <http://www.linuxvirtualserver.org/software/kernel-2.6/ipvsadm-1.26.tar.gz>  tar -zxvf ipvsadm-1.26.tar.gz -C /usr/src  cd /usr/src/ipvsadm-1.26/  make && make install |

## ipvsadm命令的使用

描述：linux虚拟服务器管理工具

用法：ipvsadm 选项 服务器地址 -s 算法

或者ipvsadm 选项 服务器地址 -r 真实服务器地址 [工作模式] [权重]…

选项：

|  |
| --- |
| -A --add-service：添加虚拟服务，使用ip,端口,协议唯一定义一个虚拟服务。  -E --edit-service：编辑内核虚拟服务器表中的一条虚拟服务器记录。  -D --delete-service： 删除内核虚拟服务器表中的一条虚拟服务器记录。  -C --clear： 清除内核虚拟服务器表中的所有记录。  -R --restore： 恢复虚拟服务器规则  -S --save： 保存虚拟服务器规则，输出为-R 选项可读的格式  -a --add-server： 在一个虚拟服务器中增加一台新的真实服务器  -e --edit-server： 编辑一条虚拟服务器记录中的某条真实服务器记录  -d --delete-server： 删除一条虚拟服务器记录中的某条真实服务器记录  -L|-l --list： 显示内核虚拟服务器表  -t --tcp-service service-address： 说明虚拟服务器提供的是tcp的服务[ip:port]  -u --udp-service service-address 说明虚拟服务器提供的是udp 的服务  -s --scheduler scheduler 使用的调度算法，rr|wrr|lc|wlc|lblc|lblcr|dh|sh|sed|nq,默认的调度算法是：wlc  -r --real-server server-address ：真实的服务器[Real-Server:port]  -g --gatewaying ：指定LVS 的工作模式为直接路由模式（也是LVS 默认的模式）  -i --ipip ：指定LVS 的工作模式为TUN隧道模式  -m --masquerading ：指定LVS 的工作模式为NAT 模式  -w --weight weight ：真实服务器的权值  -c --connection ：显示LVS 目前的连接 如：ipvsadm -L -c  -n --numeric 输出IP 地址和端口的数字形式 |

## ipvsadm命令应用

|  |
| --- |
| #添加虚拟服务，设置调度  ipvsadm -A -t 172.17.128.83:80 -s rr  ipvsadm -a -t 172.17.128.83:80 -r 172.16.129.201:8080 –m  ipvsadm -a -t 172.17.128.83:80 -r 172.16.129.62:8080 -m  ipvsadm -A -t 172.17.128.83:80 -s wrr  ipvsadm -a -t 172.17.128.83:80 -r 172.16.129.201:8080 -i -w 1  ipvsadm -a -t 172.17.128.83:80 -r 172.16.129.62:8080 -i -w 2  ipvsadm –Ln #查看LVS规则表  ipvsadm –Lnc#产看当前LVS调度情况  #删除虚拟服务的某个真实服务器  ipvsadm -d -t 172.17.128.83:80 -r 172.16.129.62:8080  #虚拟服务规则的备份和还原  ipvsadm -S > /tmp/ip\_vs.bak #备份  ipvsadm –C #清空规则  ipvsadm -R < /tmp/ip\_vs.bak #恢复  #修改虚拟服务  ipvsadm -E -t 172.17.128.83:80 -s wrr |

# LVS负载均衡应用案例

## VS/NET 模式

|  |
| --- |
| #拓扑图  #配置Linux Director（前端负载调度器）IP  vim /etc/sysconfig/network-scripts/ifcfg-eth0  vim /etc/sysconfig/network-scripts/ifcfg-eth1  #创建虚拟服务  ipvsadm -C  ipvsadm -A -t 115.29.210.85:80 -s rr  ipvsadm -a -t 115.29.210.85:80 -r 10.161.212.254:80 -m  ipvsadm -a -t 115.29.210.85:80 -r 10.161.164.55:80 -m  #保持调度规则  ipvsadm-save  #开启路由转发功能  echo 1 > /proc/sys/net/ipv4/ip\_forward  #sed -i /ip\_forward/s/0/1/ /etc/sysctl.conf  sysctl -p  #关闭防火墙  iptables -F  iptables -X  service iptables save  #配置 Real Server 真实服务器,设置ip  vim /etc/sysconfig/network-scripts/ifcfg-eth0  #分别执行  yum -y install httpd  echo "115.29.211.14" > /var/www/html/index.html  echo "115.29.241.37" > /var/www/html/index.html  service httpd start  #查看调度情况  ipvsadm -Lnc |

## VS/DR直接路由模式

|  |  |
| --- | --- |
| #拓扑图    #Linux Director 负载调度器，配置网络环境  vim /etc/sysconfig/network-scripts/ifcfg-eth0  vim /etc/sysconfig/network-scripts/ifcfg-eth1  #创建虚拟服务  ipvsadm -C  ipvsadm -A -t 115.29.210.85:80 -s wrr  ipvsadm -a -t 115.29.210.85:80 -r 10.161.212.254:80 -g -w 1  ipvsadm -a -t 115.29.210.85:80 -r 10.161.164.55:80 -g -w 1  #配置真实服务器（多台）  vim /etc/sysconfig/network-scripts/ifcfg-eth0  vim /etc/sysconfig/network-scripts/ifcfg-lo:0(VIP)   |  | | --- | | DEVICE=lo:0  ONBOOT=yes  BOOTPROTO=static  IPADDR=115.29.210.85  NETMASK=255.255.255.255 |   service network restart  echo 1 > /proc/sys/net/ipv4/conf/lo/arp\_ignore  echo 2 > /proc/sys/net/ipv4/conf/lo/arp\_announce  echo 1 > /proc/sys/net/ipv4/conf/all/arp\_ignore  echo 2 > /proc/sys/net/ipv4/conf/all/arp\_announce  sysctl -p  yum -y install httpd  echo "115.29.211.14" > /var/www/html/index.html  echo "115.29.241.37" > /var/www/html/index.html  service httpd start |

编写并运行lvs.sh脚本，绑定VIP地址到LVS-MASTER上，并设定LVS工作模式等。lvs.sh脚本的代码内容如下所示：

|  |
| --- |
| #!/bin/bash  SNS\_VIP=172.16.129.212  SNS\_RIP1=172.16.130.64  SNS\_RIP2=172.16.130.66  . /etc/rc.d/init.d/functions  logger $0 called with $1  case "$1" in  start)  # set squid vip  #/sbin/ipvsadm-set 30 5 60  /sbin/ifconfig eth0:0 $SNS\_VIP broadcast $SNS\_VIP netmask 255.255.255.255 broadcast $SNS\_VIP up  /sbin/route add -host $SNS\_VIP dev eth0:0  /sbin/ipvsadm -A -t $SNS\_VIP:80 -s wlc -p 120  /sbin/ipvsadm -a -t $SNS\_VIP:80 -r $SNS\_RIP1:80 -g -w 1  /sbin/ipvsadm -a -t $SNS\_VIP:80 -r $SNS\_RIP2:80 -g -w 1  touch /var/lock/subsys/ipvsadm >/dev/null 2>&1  ;;  stop)  /sbin/ipvsadm -C  /sbin/ipvsadm -Z  ifconfig eth0:0 down  route del $SNS\_VIP  rm -rf /var/lock/subsys/ipvsadm >/dev/null 2>&1  echo "ipvsadm stoped"  ;;  status)  if [ ! -e /var/lock/subsys/ipvsadm ];then  echo "ipvsadm stoped"  exit 1  else  echo "ipvsadm OK"  fi  ;;  \*)  echo "Usage: $0 {start|stop|status}"  exit 1  esac  exit 0 |

在后端的两台Web服务器上执行realserver.sh脚本，此脚本的主要作用为：绑定VIP地址，并设定arp抑制:

|  |
| --- |
| #!/bin/bash  SNS\_VIP=172.16.130.63  . /etc/rc.d/init.d/functions  case "$1" in  start)  ifconfig lo:0 $SNS\_VIP netmask 255.255.255.255 broadcast $SNS\_VIP  /sbin/route add -host $SNS\_VIP dev lo:0  echo "1" >/proc/sys/net/ipv4/conf/lo/arp\_ignore  echo "2" >/proc/sys/net/ipv4/conf/lo/arp\_announce  echo "1" >/proc/sys/net/ipv4/conf/all/arp\_ignore  echo "2" >/proc/sys/net/ipv4/conf/all/arp\_announce  sysctl -p >/dev/null 2>&1  echo "RealServer Start OK"  ;;  stop)  ifconfig lo:0 down  route del $SNS\_VIP >/dev/null 2>&1  echo "0" >/proc/sys/net/ipv4/conf/lo/arp\_ignore  echo "0" >/proc/sys/net/ipv4/conf/lo/arp\_announce  echo "0" >/proc/sys/net/ipv4/conf/all/arp\_ignore  echo "0" >/proc/sys/net/ipv4/conf/all/arp\_announce  echo "RealServer Stoped"  ;;  \*)  echo "Usage: $0 {start|stop}"  exit 1  esac  exit 0 |