Python开发之运维架构

讲师：王晓春

高性能集群Linux Virtual Server

本章内容

- 集群概念
- LVS介绍
- LVS实现
- LVS高可用性

Cluster概念

- 系统扩展方式：
  - Scale UP：向上扩展,增强
  - Scale Out：向外扩展,增加设备，调度分配问题，Cluster
- Cluster：集群,为解决某个特定问题将多台计算机组合起来形成的单个系统
- Linux Cluster类型：
  - LB：Load Balancing，负载均衡
  - HA：High Availiablity，高可用，SPOF（single Point Of failure）
    - MTBF:Mean Time Between Failure 平均无故障时间
    - MTTR:Mean Time To Restoration（ repair）平均恢复前时间
    - A=MTBF/（MTBF+MTTR）
    - (0,1)：99%, 99.5%, 99.9%, 99.99%, 99.999%, 99.9999%
  - HPC：High-performance computing，高性能 www.top500.org
- 分布式系统：
  - 分布式存储：云盘
  - 分布式计算：hadoop，Spark

Cluster分类

- LB Cluster的实现
- 硬件
  - F5 Big-IP
  - Citrix Netscaler
  - A10 A10
- 软件
  - lvs：Linux Virtual Server
  - nginx：支持四层调度
  - haproxy：支持四层调度
  - ats：apache traffic server，yahoo捐助
  - perlbal：Perl 编写
  - pound

Cluster分类

- 基于工作的协议层次划分：
- 传输层（通用）：DPORT

  LVS：

  nginx：stream

  haproxy：mode tcp
- 应用层（专用）：针对特定协议，自定义的请求模型分类

  proxy server：

  http：nginx, httpd, haproxy(mode http), ...

  fastcgi：nginx, httpd, ...

  mysql：mysql-proxy, ...

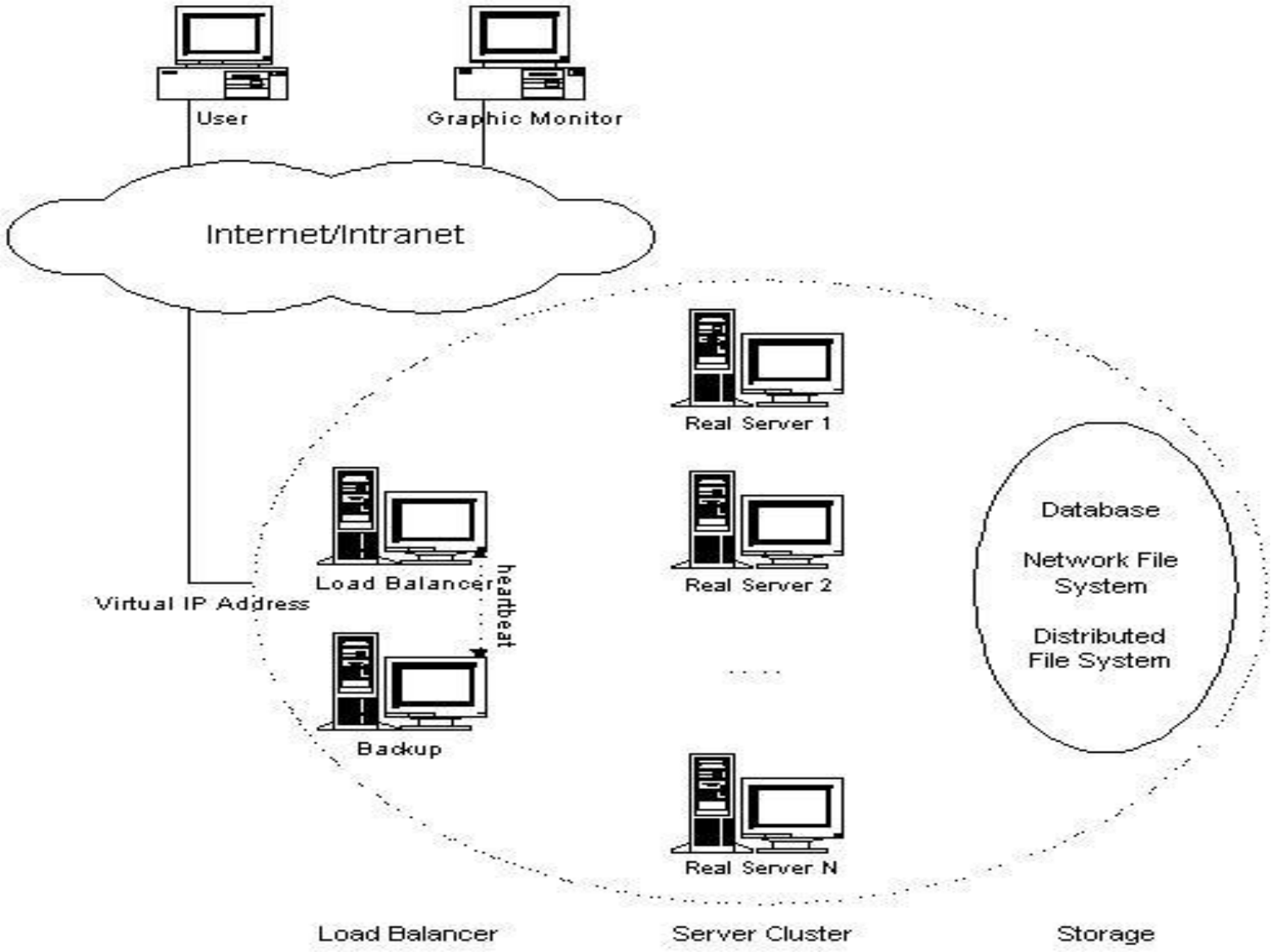
Cluster相关

- 会话保持：负载均衡
  - (1) session sticky：同一用户调度固定服务器
    - Source IP：LVS sh算法（对某一特定服务而言）
    - Cookie
  - (2) session replication：每台服务器拥有全部session
    - session multicast cluster
  - (3) session server：专门的session服务器
    - Memcached，Redis
- HA集群实现方案
  - keepalived:vrrp协议
  - ais:应用接口规范
    - heartbeat
    - cman+rgmanager(RHCS)
    - coresync_pacemaker

LVS介绍

- LVS：Linux Virtual Server，负载调度器，集成内核 章文嵩 阿里
  官网：http://www.linuxvirtualserver.org/
  VS: Virtual Server，负责调度
  RS: Real Server，负责真正提供服务
  L4：四层路由器或交换机
- 工作原理：VS根据请求报文的目标IP和目标协议及端口将其调度转发至某RS，根据调度算法来挑选RS
- iptables/netfilter：
  iptables：用户空间的管理工具
  netfilter：内核空间上的框架
  流入：PREROUTING --> INPUT
  流出：OUTPUT --> POSTROUTING
  转发：PREROUTING --> FORWARD --> POSTROUTING DNAT：目标地址转换； PREROUTING

LVS集群体系结构



LVS概念

- lvs集群类型中的术语：
  - VS：Virtual Server，Director Server(DS)
    Dispatcher(调度器)，Load Balancer
  - RS：Real Server(lvs), upstream server(nginx)
    backend server(haproxy)
  - CIP：Client IP
  - VIP: Virtual serve IP　　VS外网的IP
  - DIP: Director IP　　VS内网的IP
  - RIP: Real server IP

  - 访问流程：CIP <--> VIP == DIP <--> RIP

lvs集群的类型

- lvs: ipvsadm/ipvs
  - ipvsadm：用户空间的命令行工具，规则管理器
    用于管理集群服务及RealServer
  - ipvs：工作于内核空间netfilter的INPUT钩子上的框架
- lvs集群的类型：
  - lvs-nat：修改请求报文的目标IP,多目标IP的DNAT
  - lvs-dr：操纵封装新的MAC地址
  - lvs-tun：在原请求IP报文之外新加一个IP首部
  - lvs-fullnat：修改请求报文的源和目标IP

lvs-nat模式

- lvs-nat：

本质是多目标IP的DNAT，通过将请求报文中的目标地址和目标端口修改为某挑出的RS的RIP和PORT实现转发

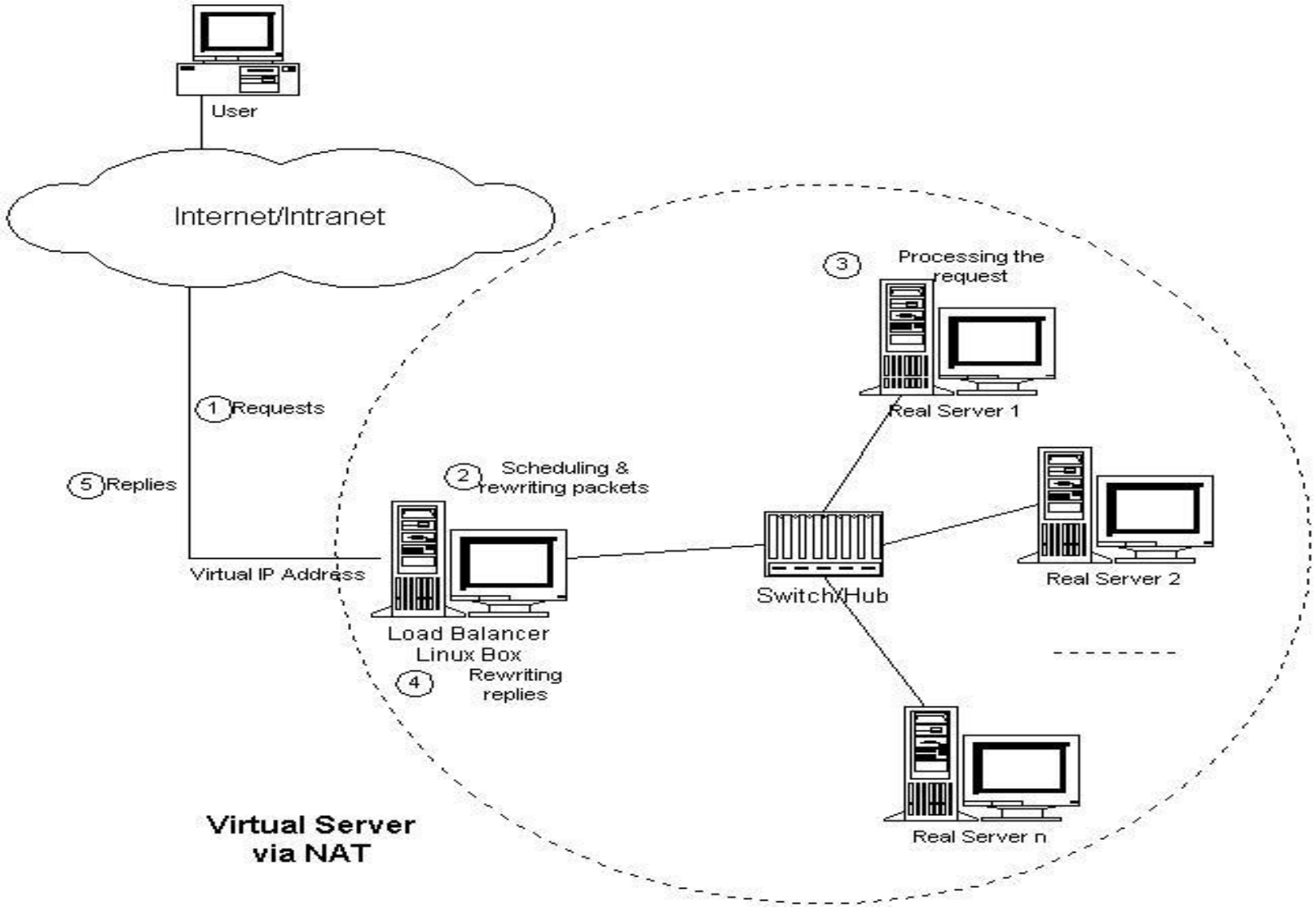
（1）RIP和DIP必须在同一个IP网络，且应该使用私网地址；RS的网关要指向DIP

（2）请求报文和响应报文都必须经由Director转发，Director易于成为系统瓶颈

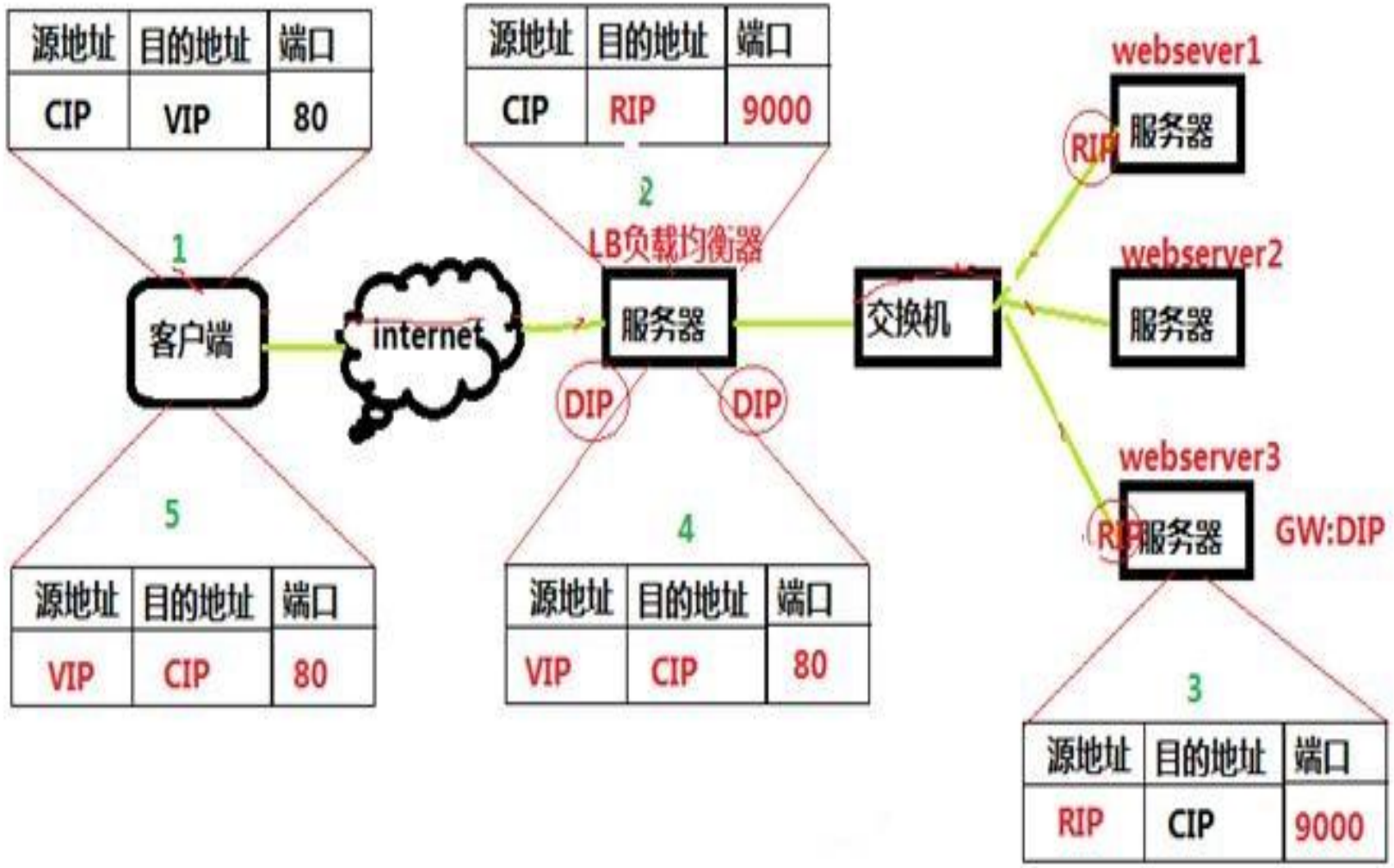
（3）支持端口映射，可修改请求报文的目标PORT

（4）VS必须是Linux系统，RS可以是任意OS系统

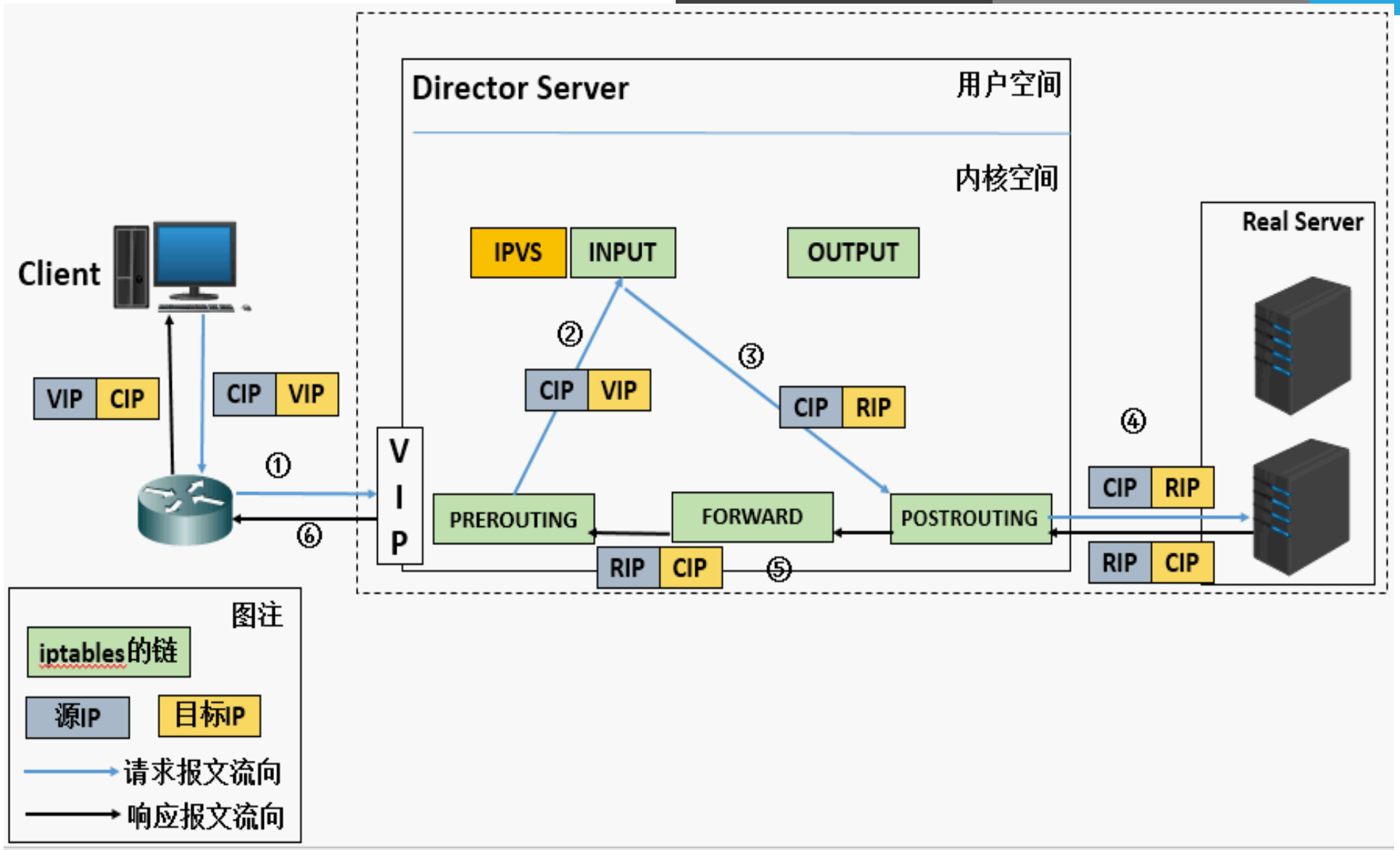
VS/NAT的体系结构


User
Internet/Intranet
1 Requests
5 Replies
Virtual IP Address
2 Scheduling & rewriting packets
Load Balancer Linux Box
4 Rewriting replies
3 Processing the request
Real Server 1
Real Server 2
Real Server n
Switch/Hub
Virtual Server via NAT

NAT模式IP包调度过程



NAT模式


Director Server
用户空间
内核空间
Client
IPVS
INPUT
OUTPUT
Real Server
VIP
CIP
CIP
VIP
CIP
VIP
CIP
RIP
①
②
③
④
⑤
⑥
V
I
P
PREROUTING
FORWARD
POSTROUTING
RIP
CIP
CIP
RIP
RIP
CIP
图注
iptables的链
源IP
目标IP
请求报文流向
响应报文流向

LVS-DR模式

- LVS-DR：Direct Routing，直接路由，LVS默认模式,应用最广泛,通过为请求报文重新封装一个MAC首部进行转发，源MAC是DIP所在的接口的MAC，目标MAC是某挑选出的RS的RIP所在接口的MAC地址；源IP/PORT，以及目标IP/PORT均保持不变
- Director和各RS都配置有VIP

(1) 确保前端路由器将目标IP为VIP的请求报文发往Director

- 在前端网关做静态绑定VIP和Director的MAC地址
- 在RS上使用arptables工具

```
arptables -A IN -d $VIP -j DROP
arptables -A OUT -s $VIP -j mangle --mangle-ip-s $RIP
```

- 在RS上修改内核参数以限制arp通告及应答级别

```
arp_announce
arp_ignore
```

LVS-DR模式

(2) RS的RIP可以使用私网地址，也可以是公网地址；RIP与DIP在同一IP网络；RIP的网关不能指向DIP，以确保响应报文不会经由Director

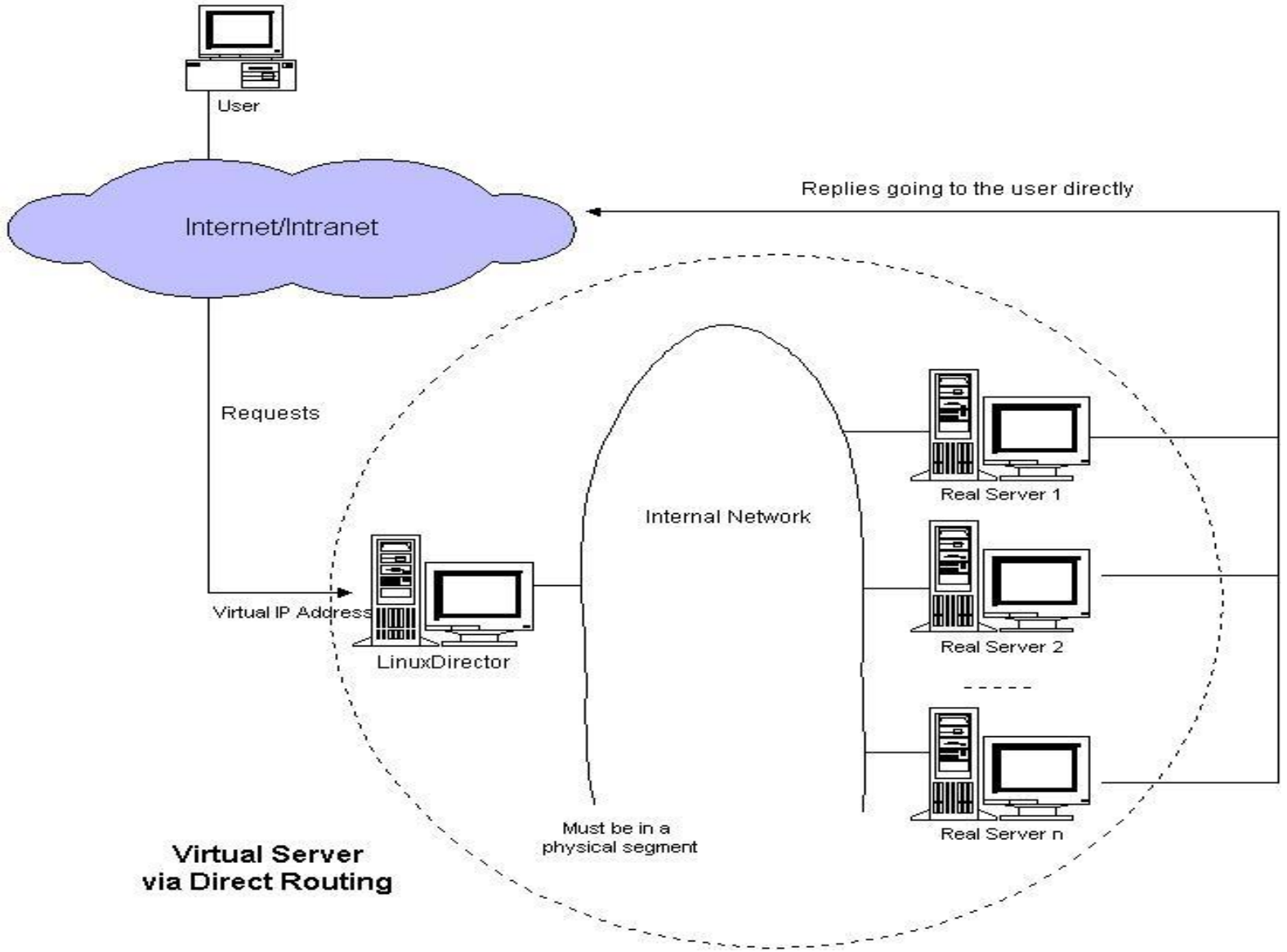
(3) RS和Director要在同一个物理网络

(4) 请求报文要经由Director，但响应报文不经由Director，而由RS直接发往Client

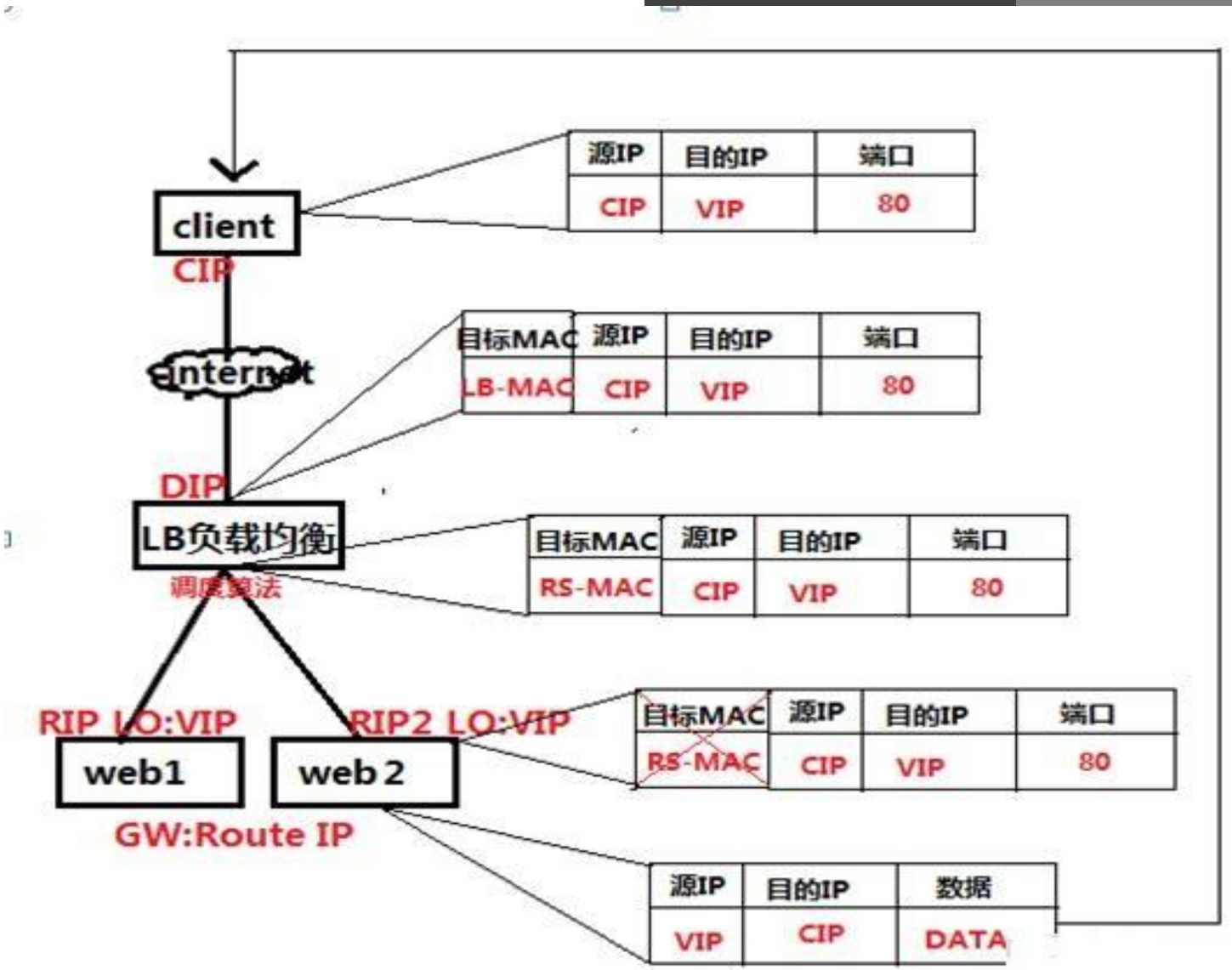
(5) 不支持端口映射（端口不能修败）

(6) RS可使用大多数OS系统

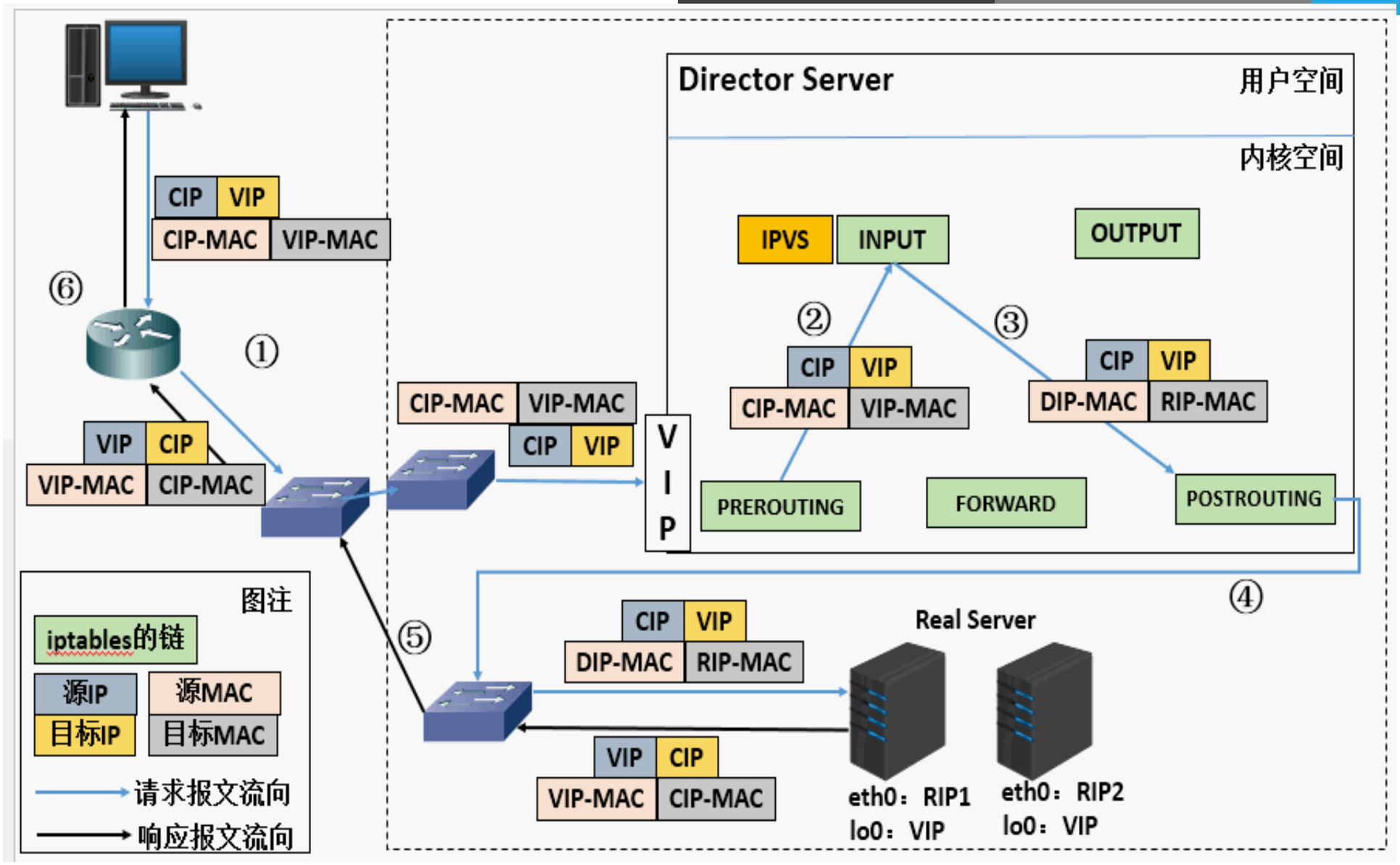
VS/DR体系结构


User
Internet/Intranet
Replies going to the user directly
Requests
Virtual IP Address
LinuxDirector
Internal Network
Real Server 1
Real Server 2
Real Server n
Must be in a
physical segment
Virtual Server
via Direct Routing

DR模式IP包调度过程



DR模式


Director Server
用户空间
内核空间
IPVS
INPUT
OUTPUT
PREROUTING
FORWARD
POSTROUTING
VIP
CIP
VIP-MAC
CIP-MAC
DIP-MAC
RIP-MAC
①
②
③
④
⑤
⑥
Real Server
eth0：RIP1
lo0：VIP
eth0：RIP2
lo0：VIP
图注
iptables的链
源IP
目标IP
源MAC
目标MAC
请求报文流向
响应报文流向

lvs-tun模式

- lvs-tun：

转发方式：不修改请求报文的IP首部（源IP为CIP，目标IP为VIP），而在原IP报文之外再封装一个IP首部（源IP是DIP，目标IP是RIP），将报文发往挑选出的目标RS；RS直接响应给客户端（源IP是VIP，目标IP是CIP）

(1) DIP, VIP, RIP都应该是公网地址

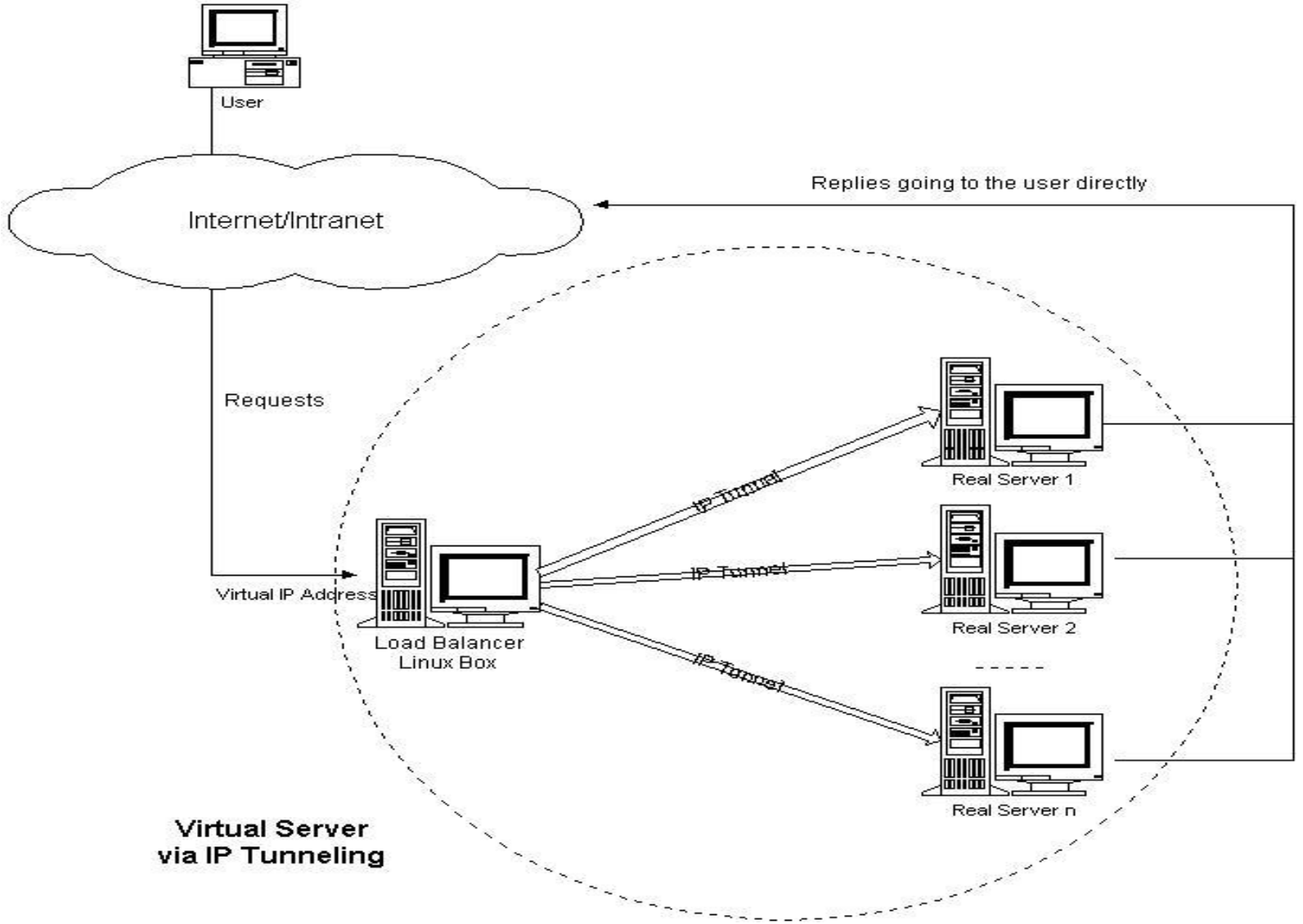
(2) RS的网关不能，也不可能指向DIP

(3) 请求报文要经由Director，但响应不能经由Director

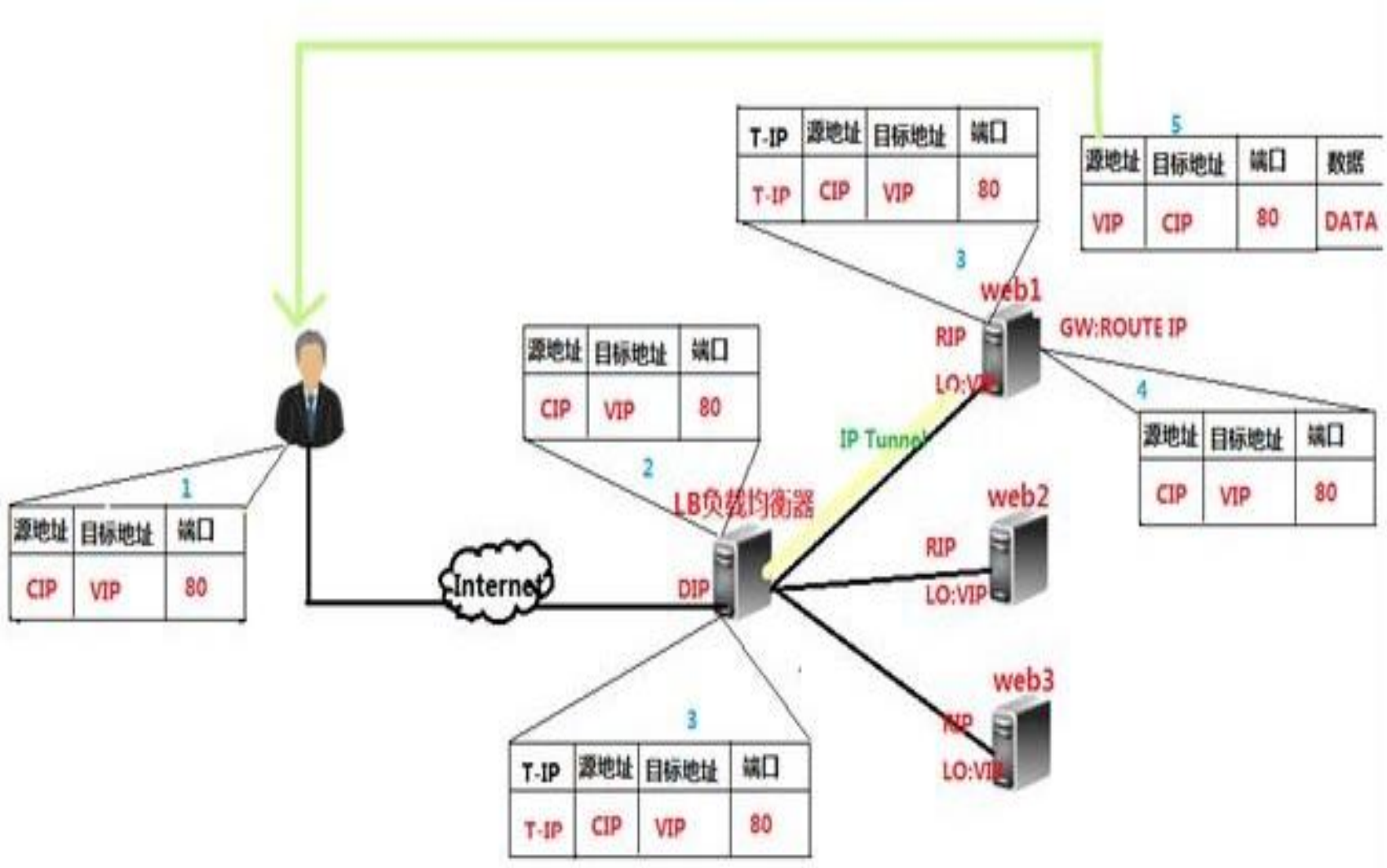
(4) 不支持端口映射

(5) RS的OS须支持隧道功能

VS/TUN体系结构



TUN模式IP包调度过程



lvs-fullnat模式

- lvs-fullnat：通过同时修改请求报文的源IP地址和目标IP地址进行转发

  CIP --> DIP

  VIP --> RIP

(1) VIP是公网地址，RIP和DIP是私网地址，且通常不在同一IP网络；因此，RIP的网关一般不会指向DIP

(2) RS收到的请求报文源地址是DIP，因此，只需响应给DIP；但Director还要将其发往Client

(3) 请求和响应报文都经由Director

(4) 支持端口映射；

注意：此类型kernel默认不支持

LVS工作模式总结

	VS/NAT	VS/TUN	VS/DR
Server	any	Tunneling	Non-arp device
server network	private	LAN/WAN	LAN
server number	low (10~20)	High (100)	High (100)
server gateway	load balancer	own router	Own router

- lvs-nat与lvs-fullnat：请求和响应报文都经由Director
  lvs-nat：RIP的网关要指向DIP
  lvs-fullnat：RIP和DIP未必在同一IP网络，但要能通信
- lvs-dr与lvs-tun：请求报文要经由Director，但响应报文由RS直接发往Client
  lvs-dr：通过封装新的MAC首部实现，通过MAC网络转发
  lvs-tun：通过在原IP报文外封装新IP头实现转发，支持远距离通信

ipvs scheduler

- ipvs scheduler：
- 根据其调度时是否考虑各RS当前的负载状态

  两种：静态方法和动态方法
- 静态方法：仅根据算法本身进行调度

1、RR：roundrobin，轮询

2、WRR：Weighted RR，加权轮询

3、SH：Source Hashing，实现session sticky，源IP地址hash；将来自于同一个IP地址的请求始终发往第一次挑中的RS，从而实现会话绑定

4、DH：Destination Hashing；目标地址哈希，将发往同一个目标地址的请求始终转发至第一次挑中的RS，典型使用场景是正向代理缓存场景中的负载均衡，如：宽带运营商

ipvs scheduler

- 动态方法：主要根据每RS当前的负载状态及调度算法进行调度
  Overhead=value 较小的RS将被调度
  1、LC：least connections 适用于长连接应用
  Overhead=activeconns*256+inactiveconns
  2、WLC：Weighted LC，默认调度方法
  Overhead=(activeconns*256+inactiveconns)/weight
  3、SED：Shortest Expection Delay,初始连接高权重优先
  Overhead=(activeconns+1)*256/weight
  4、NQ：Never Queue，第一轮均匀分配，后续SED
  5、LBLC：Locality-Based LC，动态的DH算法，使用场景：根据负载状态实现正向代理
  6、LBLCR：LBLC with Replication，带复制功能的LBLC
  解决LBLC负载不均衡问题，从负载重的复制到负载轻的RS

ipvs

- ipvsadm/ipvs：
- ipvs：

  grep -i -C 10 "ipvs" /boot/config-VERSION-RELEASE.x86_64

  支持的协议：TCP， UDP， AH， ESP， AH_ESP, SCTP
- ipvs集群：

  管理集群服务

  管理服务上的RS

ipvsadm包构成

- ipvsadm：
- 程序包：ipvsadm
  - Unit File: ipvsadm.service
  - 主程序：/usr/sbin/ipvsadm
  - 规则保存工具：/usr/sbin/ipvsadm-save
  - 规则重载工具：/usr/sbin/ipvsadm-restore
  - 配置文件：/etc/sysconfig/ipvsadm-config

ipvsadm命令

- ipvsadm命令：
- 核心功能：
  - 集群服务管理：增、删、改
  - 集群服务的RS管理：增、删、改
  - 查看

ipvsadm -A|E -t|u|f service-address [-s scheduler] [-p [timeout]] [-M netmask] [--pe persistence_engine] [-b sched-flags]

ipvsadm -D -t|u|f service-address 删除

ipvsadm –C 清空

ipvsadm –R 重载

ipvsadm -S [-n] 保存

ipvsadm -a|e -t|u|f service-address -r server-address [options]

ipvsadm -d -t|u|f service-address -r server-address

ipvsadm -L|l [options]

ipvsadm -Z [-t|u|f service-address]

ipvsadm命令

- 管理集群服务：增、改、删
- 增、改：

ipvsadm -A|E -t|u|f service-address [-s scheduler] [-p [timeout]]

- 删除：

ipvsadm -D -t|u|f service-address

- service-address：

-t|u|f：

-t: TCP协议的端口，VIP:TCP_PORT

-u: UDP协议的端口，VIP:UDP_PORT

-f：firewall MARK，标记，一个数字

- [-s scheduler]：指定集群的调度算法，默认为wlc

ipvsadm命令

- 管理集群上的RS：增、改、删
- 增、改：ipvsadm -a|e -t|u|f service-address -r server-address [-g|i|m] [-w weight]
- 删：ipvsadm -d -t|u|f service-address -r server-address
- server-address：

  rip[:port] 如省略port，不作端口映射
- 选项：

  lvs类型：

  -g: gateway, dr类型，默认

  -i: ipip, tun类型

  -m: masquerade, nat类型

  -w weight：权重

ipvsadm命令

- 清空定义的所有内容：ipvsadm –C
- 清空计数器：ipvsadm -Z [-t|u|f service-address]
- 查看：ipvsadm -L|l [options]
  - --numeric, -n：以数字形式输出地址和端口号
  - --exact：扩展信息，精确值
  - --connection，-c：当前IPVS连接输出
  - --stats：统计信息
  - --rate ：输出速率信息
- ipvs规则： /proc/net/ip_vs
- ipvs连接：/proc/net/ip_vs_conn

保存及重载规则

- 保存：建议保存至/etc/sysconfig/ipvsadm

  ipvsadm-save > /PATH/TO/IPVSADM_FILE

  ipvsadm -S > /PATH/TO/IPVSADM_FILE

  systemctl stop ipvsadm.service

- 重载：

  ipvsadm-restore < /PATH/FROM/IPVSADM_FILE

  ipvsadm -R < /PATH/FROM/IPVSADM_FILE

  systemctl restart ipvsadm.service

LVS

- 负载均衡集群设计时要注意的问题

  (1) 是否需要会话保持

  (2) 是否需要共享存储

  共享存储：NAS， SAN， DS（分布式存储）

  数据同步：

- lvs-nat：

  设计要点：

  (1) RIP与DIP在同一IP网络, RIP的网关要指向DIP

  (2) 支持端口映射

  (3) Director要打开核心转发功能

作业

- NAT模型实现http负载均衡集群
- NAT模型实现https负载均衡集群

  注意：RS: 都要提供同一个私钥和同一个证书

LVS-DR

- DR模型中各主机上均需要配置VIP，解决地址冲突的方式有三种：
  (1) 在前端网关做静态绑定
  (2) 在各RS使用arptables
  (3) 在各RS修改内核参数，来限制arp响应和通告的级别
- 限制响应级别：arp_ignore
  0：默认值，表示可使用本地任意接口上配置的任意地址进行响应
  1: 仅在请求的目标IP配置在本地主机的接收到请求报文的接口上时，才给予响应
- 限制通告级别：arp_announce
  0：默认值，把本机所有接口的所有信息向每个接口的网络进行通告
  1：尽量避免将接口信息向非直接连接网络进行通告
  2：必须避免将接口信息向非本网络进行通告

RS的预配置脚本


```
#!/bin/bash
vip=192.168.0.100
mask='255.255.255.255‘
dev=lo:1
case $1 in
start)
        echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
        echo 1 > /proc/sys/net/ipv4/conf/lo/arp_ignore
        echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
        echo 2 > /proc/sys/net/ipv4/conf/lo/arp_announce
        ifconfig $dev $vip netmask $mask broadcast $vip up
        route add -host $vip dev $dev
        ;;
stop)
        ifconfig $dev down
        echo 0 > /proc/sys/net/ipv4/conf/all/arp_ignore
        echo 0 > /proc/sys/net/ipv4/conf/lo/arp_ignore
        echo 0 > /proc/sys/net/ipv4/conf/all/arp_announce
        echo 0 > /proc/sys/net/ipv4/conf/lo/arp_announce
        ;;
*)
        echo "Usage: $(basename $0) start|stop"
        exit 1
        ;;
esac
```

VS的配置脚本


```
#!/bin/bash
vip='192.168.0.100'
iface='eth0:1'
mask='255.255.255.255'
port='80'
rs1='192.168.0.101'
rs2='192.168.0.102'
scheduler='wrr'
type='-g'
case $1 in
start)
        ifconfig $iface $vip netmask $mask broadcast $vip up
        iptables -F
        ipvsadm -A -t ${vip}:${port} -s $scheduler
        ipvsadm -a -t ${vip}:${port} -r ${rs1} $type -w 1
        ipvsadm -a -t ${vip}:${port} -r ${rs2} $type -w 1
        ;;
stop)
        ipvsadm -C
        ifconfig $iface down
        ;;
*)
        echo "Usage $(basename $0) start|stop  “;exit 1
        ;;
esac
```

作业

- 实现NAT模式
- 网络拓扑要求：VIP与DIP/RIP不在同一网络
- DR模型实现http负载均衡集群
- DR模型实现https负载均衡集群

  注意：RS: 都要提供同一个私钥和同一个证书
- DR模型实现mysql负载均衡集群

FireWall Mark

- FWM：FireWall Mark
- MARK target 可用于给特定的报文打标记

  --set-mark value

  其中：value 为十六进制数字
- 借助于防火墙标记来分类报文，而后基于标记定义集群服务；可将多个不同的应用使用同一个集群服务进行调度
- 实现方法：
  - 在Director主机打标记：

    iptables -t mangle -A PREROUTING -d $vip -p $proto –m multiport --dports $port1,$port2,… -j MARK --set-mark NUMBER
  - 在Director主机基于标记定义集群服务：

    ipvsadm -A -f NUMBER [options]

持久连接

- session 绑定：对共享同一组RS的多个集群服务，需要统一进行绑定，lvs sh算法无法实现
- 持久连接（ lvs persistence ）模板：实现无论使用任何调度算法，在一段时间内（默认360s ），能够实现将来自同一个地址的请求始终发往同一个RS

  ipvsadm -A|E -t|u|f service-address [-s scheduler] [-p [timeout]]
- 持久连接实现方式：
  - 每端口持久（PPC）：每个端口对应定义为一个集群服务，每集群服务单独调度
  - 每防火墙标记持久（PFWMC）：基于防火墙标记定义集群服务；可实现将多个端口上的应用统一调度，即所谓的port Affinity
  - 每客户端持久（PCC）：基于0端口（表示所有服务）定义集群服务，即将客户端对所有应用的请求都调度至后端主机，必须定义为持久模式

LVS高可用性

- 1 Director不可用，整个系统将不可用；SPoF Single Point of Failure

  解决方案：高可用

  keepalived heartbeat/corosync

- 2 某RS不可用时，Director依然会调度请求至此RS

  解决方案： 由Director对各RS健康状态进行检查，失败时禁用，成功时启用

  keepalived heartbeat/corosync ldirectord

  检测方式：

  (a) 网络层检测，icmp

  (b) 传输层检测，端口探测

  (c) 应用层检测，请求某关键资源

  RS全不用时：backup server, sorry server

ldirectord

- ldirectord：监控和控制LVS守护进程，可管理LVS规则
- 包名：ldirectord-3.9.6-0rc1.1.1.x86_64.rpm
- 文件：

  /etc/ha.d/ldirectord.cf　　主配置文件

  /usr/share/doc/ldirectord-3.9.6/ldirectord.cf 配置模版

  /usr/lib/systemd/system/ldirectord.service 服务

  /usr/sbin/ldirectord　　主程序

  /var/log/ldirectord.log 日志

  /var/run/ldirectord.ldirectord.pid　pid文件

Ldirectord配置文件示例


```
checktimeout=3
checkinterval=1
autoreload=yes
logfile=“/var/log/ldirectord.log“      #日志文件
quiescent=no                           #down时yes权重为0，no为删除
virtual=5                              #指定VS的FWM或IP：port
        real=172.16.0.7:80 gate 2
        real=172.16.0.8:80 gate 1
        fallback=127.0.0.1:80 gate     #sorry server
        service=http
        scheduler=wrr
checktype=negotiate
checkport=80
request="index.html"
receive=“Test Ldirectord"
```

关于马哥教育

马哥教育
IT人的高薪职业学院

- 博客：http://mageedu.blog.51cto.com
- 主页：http://www.magedu.com
- QQ：1661815153, 113228115
- QQ群：203585050, 279599283

祝大家学业有成

谢　谢