Linux-负载均衡LVS

LVS负载均衡

负载均衡集群是Load Balance 集群的缩写，翻译成中文就是负载均衡集群。常用的负载均衡开源软件有Nginx、LVS、Haproxy，商业的硬件负载均衡设备有F5、Netscale等。

负载均衡LVS基本介绍

LB集群的架构和原理很简单，就是当用户的请求过来时，会直接分发到Director Server上，然后它把用户的请求根据设置好的调度算法，智能均衡的分发后端真正服务器（real server）上。为了避免不同机器上用户请求的数据不一样，需要用到了共享存储，这样保证所有用户请求的数据是一样的。

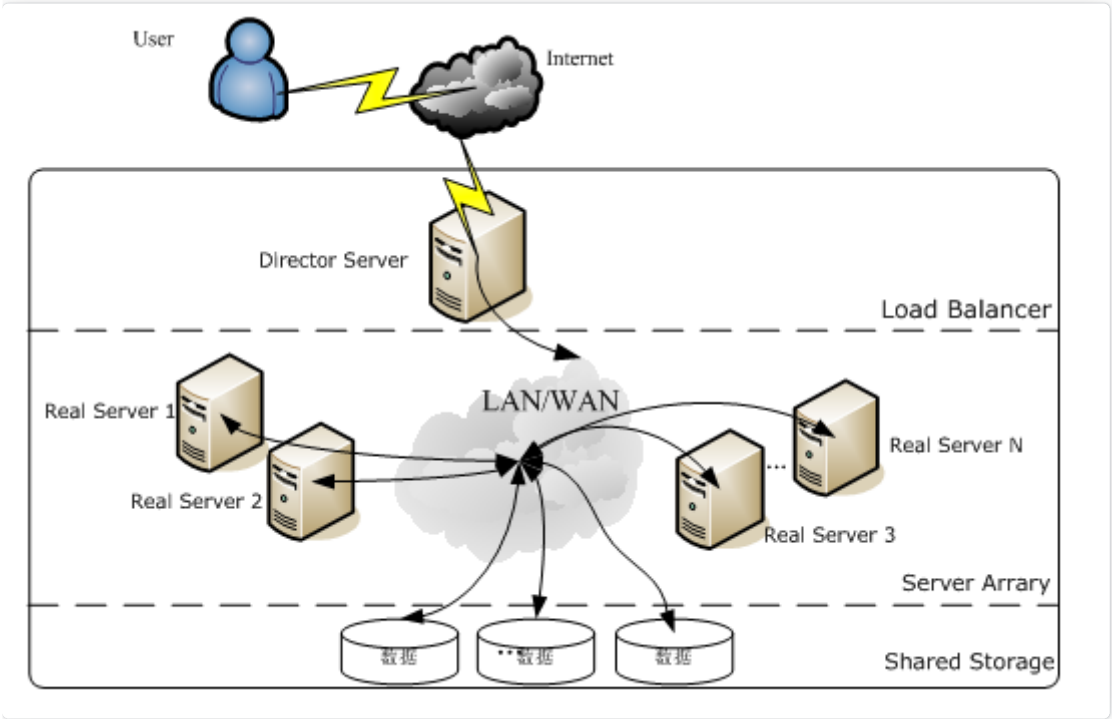
LVS是Linux Virtual Server 的简称，也就是linux虚拟服务器。这是由章文嵩博士发起的一个开源项目，官网：http://www.linuxvirtualserver.org 现在LVS已经是 Linux 内核标准的一部分。使用 LVS 可以达到的技术目标是：通过 LVS 达到的负载均衡技术和 Linux 操作系统实现一个高性能高可用的 Linux 服务集群，它具有良好的可靠性、可扩展性和可操作性。从而以廉价的成本实现最优的性能。 LVS 是一个实现负载均衡集群的开源软件项目，LVS架构从逻辑上可分为调度层、Server集群层和共享存储。

LVS集群采用IP负载均衡技术和基于内容请求分发技术。调度器具有很好的吞吐率，将请求均衡地转移到不同的服务器上执行，且调度器自动屏蔽掉服 务器的故障，从而将一组服务器构成一个高性能的、高可用的虚拟服务器。整个服务器集群的结构对客户是透明的，而且无需修改客户端和服务器端的程序。

LVS的体系架构

负载均衡的原理很简单，就是当客户端发起请求时，请求直接发给Director Server（调度器），这时会根据设定的调度算法，将请求按照算法的规定智能的分发到真正的后台服务器。以达到将压力均摊。但是我们知道，http的连接时无状态的，假设这样一个场景，我登录某宝买东西，当我看上某款商品时，我将它加入购物车，但是我刷新了一下页面，这时由于负载均衡的原因，调度器又选了新的一台服务器为我提供服务，我刚才的购物车内容全都不见了，这样就会有十分差的用户体验。所以就还需要一个存储共享，这样就保证了用户请求的数据是一样的。所以LVS负载均衡分为三层架构(也就是LVS负载均衡主要组成部分)：

如图：



LVS的各个层次的详细介绍：

- Load Balancer层：

  位于整个集群系统的最前端，有一台或者多台负载调度器（Director Server）组成，LVS模块就是安装在Director Server上，而Director的主要作用类似于一个路由器，它含有完成LVS功能所设定的路由表，通过这些路由表把用户的请求分发给Server Array层的

应用服务器（Real Server）上。同时，在Director Server上还要安装队Real Server服务的监控模块Ldirectord，此模块用于检测各个Real Server服务的健康状况。在Real Server不可用时把它从 LVS 路由表中剔除，恢复时重新加入。

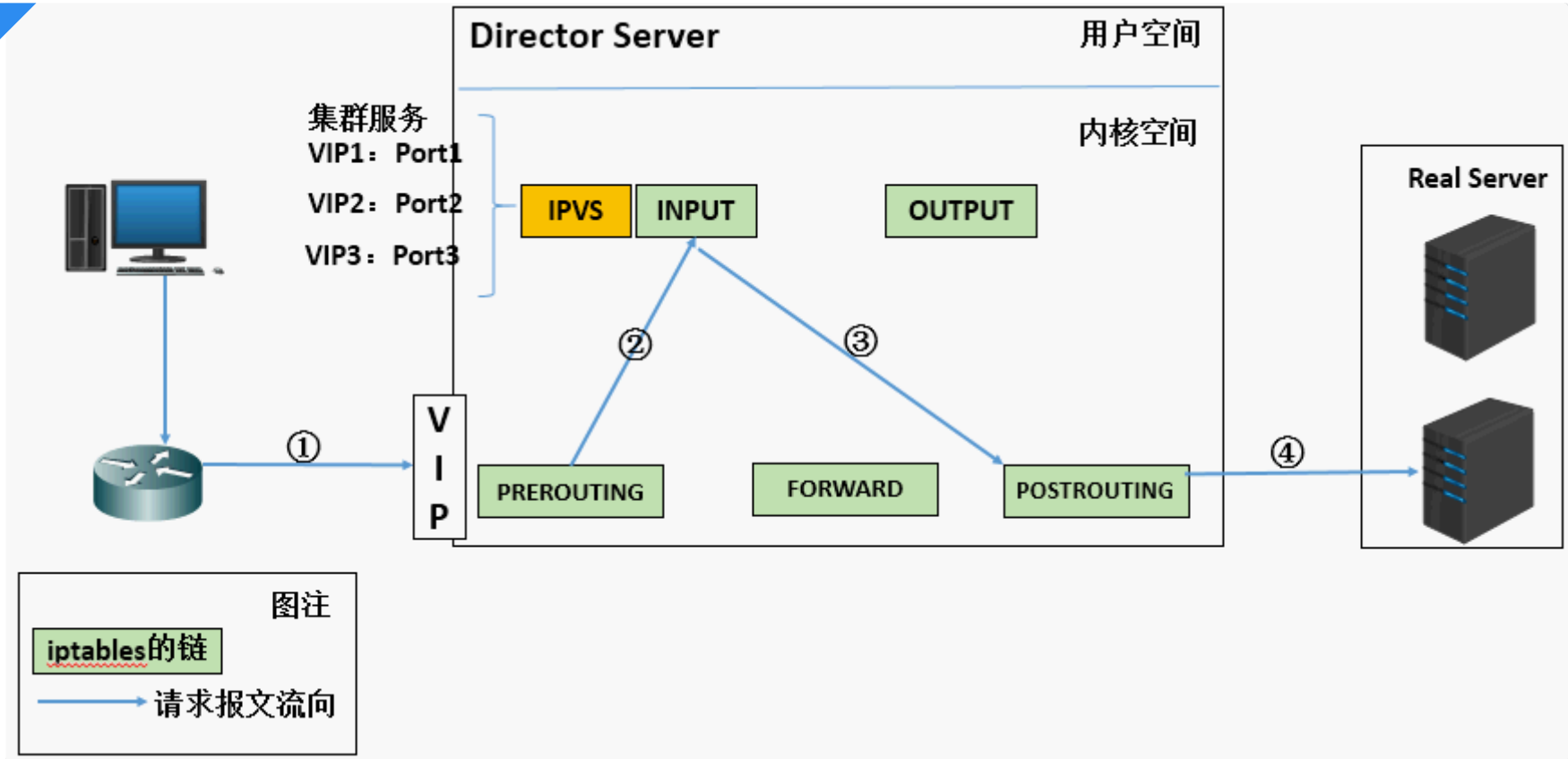
- Server Arrary层：

  由一组实际运行应用服务的机器组成，Real Server可以是WEB 服务器、MALL服务器、FTP服务器、DNS服务器、等等，每个Real Server 之间通过高速的LAN或分布在各地的WAN相连接，在实际的应用中，Director Server也可以同时兼任Real Server的角色。

- Shared Storage层：

  是为所有Real Server提供共享存储空间和内容一致性的存储区域，在物理上，一般有磁盘阵列设备组成，为了提供内容的一致性，一般可以通过NFS网络文件系统共享数据，但是NFS在繁忙的业务系统中，性能并不是很好，此时可以采用集群文件系统，列如Red hat的GFS文件系统等等。一个公司得有一个后台账目吧，这才能协调。不然客户把钱付给了A，而换B接待客户，因为没有相同的账目。B说客户没付钱，那这样就不是客户体验度的问题了。

LVS的实现原理



（1）当用户负载均衡调度器（Director Server）发起请求，调度器将请求发往至内核空间

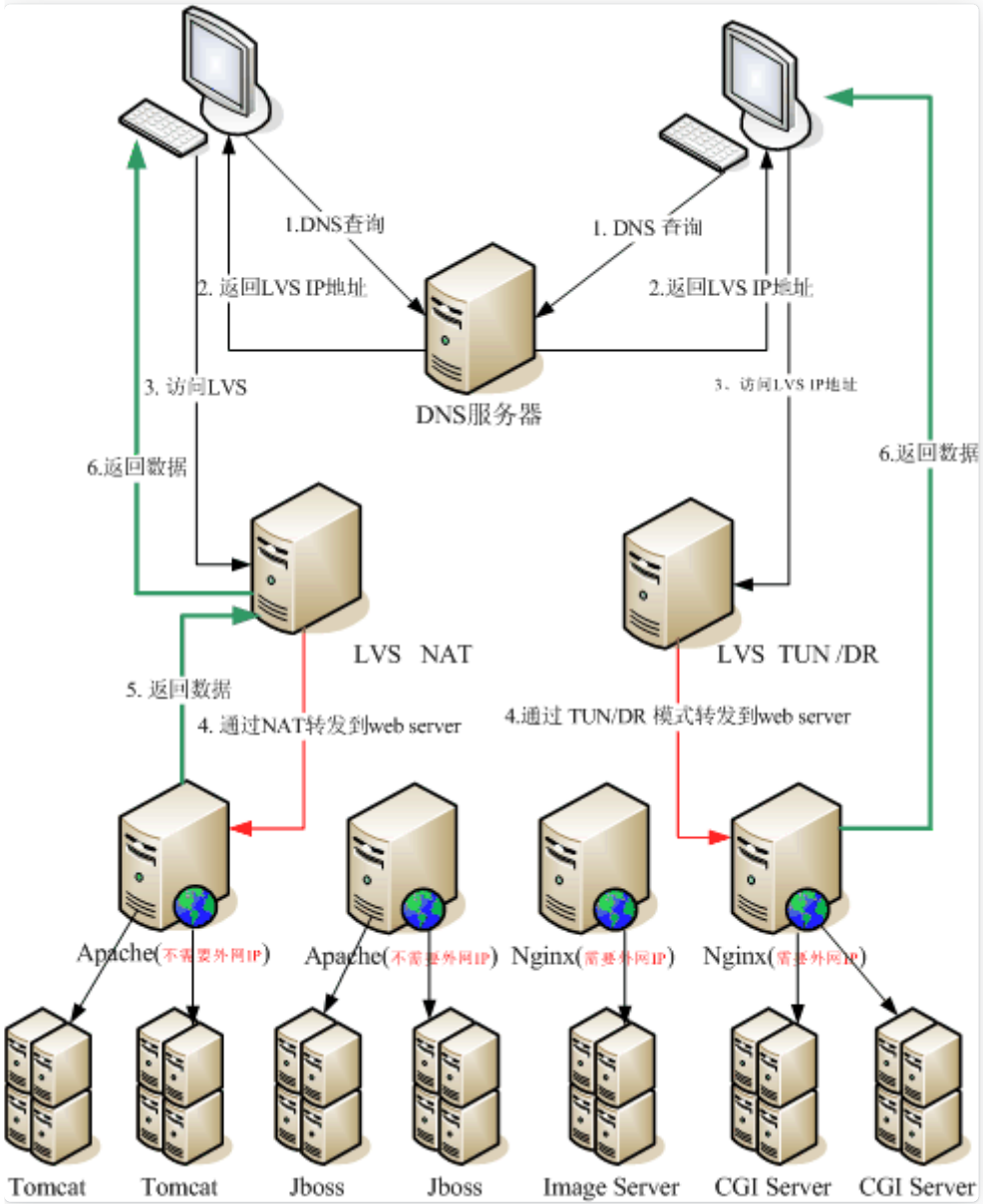
（2）PREROUTING 链首先会接受到用户请求，判断目标IP确实是本地IP，将数据包发往 INPUT 链

（3）IPVS 是工作在 INPUT 链上的，当用户请求到达INPUT时，IPVS 会将用户请求和自己定义好的集群服务进行比对，如果用户请求的就是集群服务，那么此时 IPVS 会强行修改数据包里的目标IP地址和端口，并将新的数据包发往 POSTROUTING 链

（4）POSTROUTING 链将收到数据包后发现目标IP地址刚好是自己的后端服务器，那么此时通过选路，将数据包最终发送给后端的服务器

LVS的工作原理

LVS 的工作模式分为4中分别是 NAT，DR，TUN，FULL-NAT。其中做个比较，由于工作原理的关系的，NAT的配置最为简单，但是NAT对调度器的压力太大了，导致其效率最低，DR和TUN的工作原理差不多，但是DR中，所有主机必须处于同一个物理环境中，而在TUN中，所有主机可以分布在不同的位置，服务器一个在纽约，一个在深圳。最多应用的是FULL-NAT。



LVS相关术语

（1）DS：Director Server　　指的是前端负载均衡器节点。

（2）RS：Real Server　　后端真实的工作服务器。

（3）VIP：向外部直接面向用户请求，作为用户请求的目标的ip地址。

（4）DIP：Director Server IP　　主要用于和内部服务器通讯的ip地址。

（5）RIP：Real Server IP　　后端服务器的ip地址。

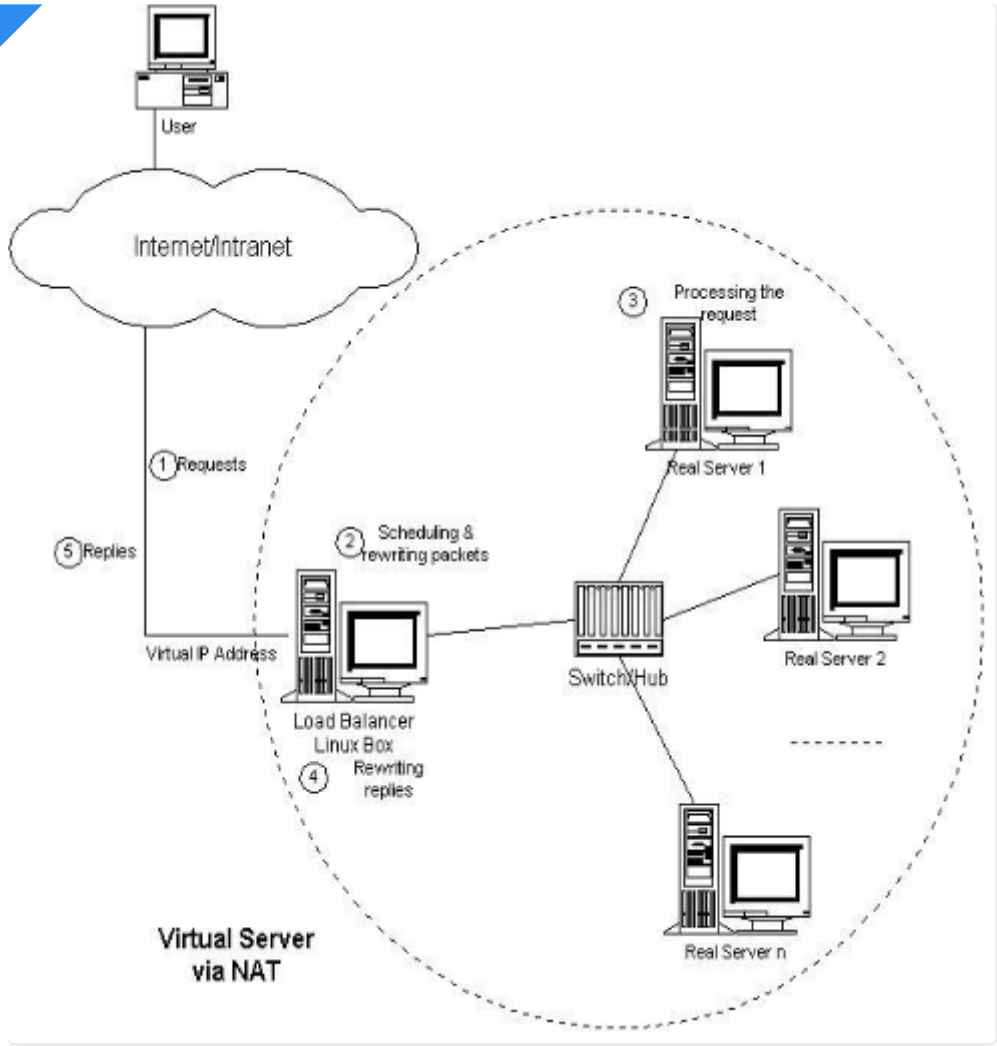
（6）CIP：Client IP　　访问客户端的IP地址。

LVS工作模式和原理

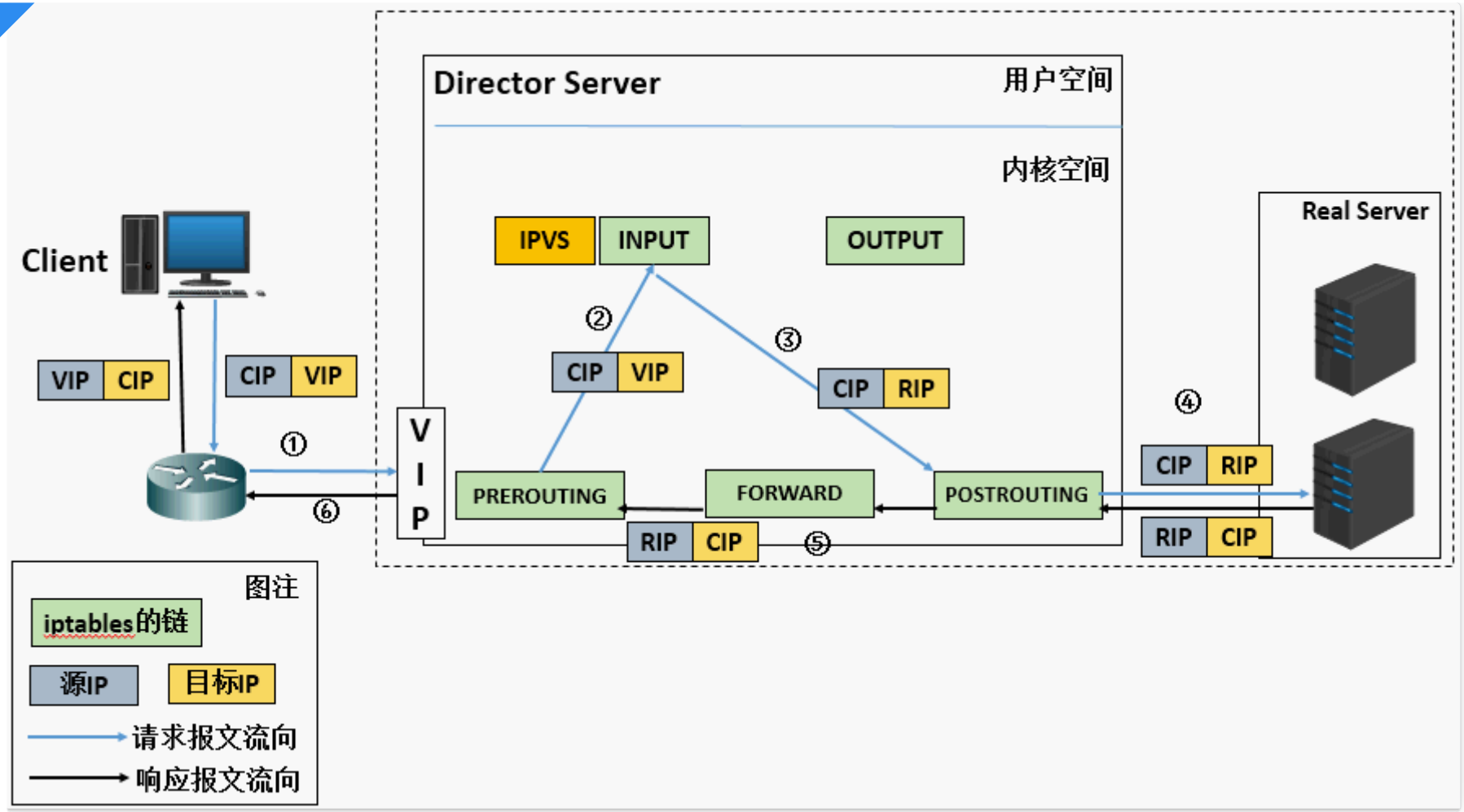
NAT 模式-网络地址转换

这个是通过网络地址转换的方法来实现调度的。首先调度器(LB)接收到客户的请求数据包时（请求的目的IP为VIP），根据调度算法决定将请求发送给哪个后端的真实服务器（RS）。然后调度就把客户端发送的请求数据包的目标IP地址及端口改成后端真实服务器的IP地址（RIP）,这样真实服务器（RS）就能够接收到客户的请求数据包了。真实服务器响应完请求后，查看默认路由（NAT模式下我们需要把RS的默认路由设置为LB服务器。）把响应后的数据包发送给LB,LB再接收到响应包后，把包的源地址改成虚拟地址（VIP）然后发送回给客户端。

VS/NAT是一种最简单的方式，所有的RealServer只需要将自己的网关指向Director即可。客户端可以是任意操作系统，但此方式下，一个Director能够带动的RealServer比较有限。在VS/NAT的方式下，Director也可以兼为一台RealServer。VS/NAT的体系结构如图所示。


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
Switch/Hub
Real Server 2
Real Server n
Virtual Server via NAT

NAT 模式工作原理：



（1）当用户请求到达Director Server，此时的请求数据报文会先到内核空间的PREROUTING链。此时报文的源IP为 CIP，目标IP为 VIP。

（2）PREROUTING检查发现数据包的目标IP 是本机，将数据包发送至INPUT链。

（3）IPVS比对数据包请求的服务是否为集群服务，若是，修改数据包的目标IP地址为后端服务器IP，然后将数据包发送至POSTROUTING链。此时报文的源IP为 CIP，目标IP为 RIP。

（4）POSTROUTING链通过选路，将数据包发送给Real Server。

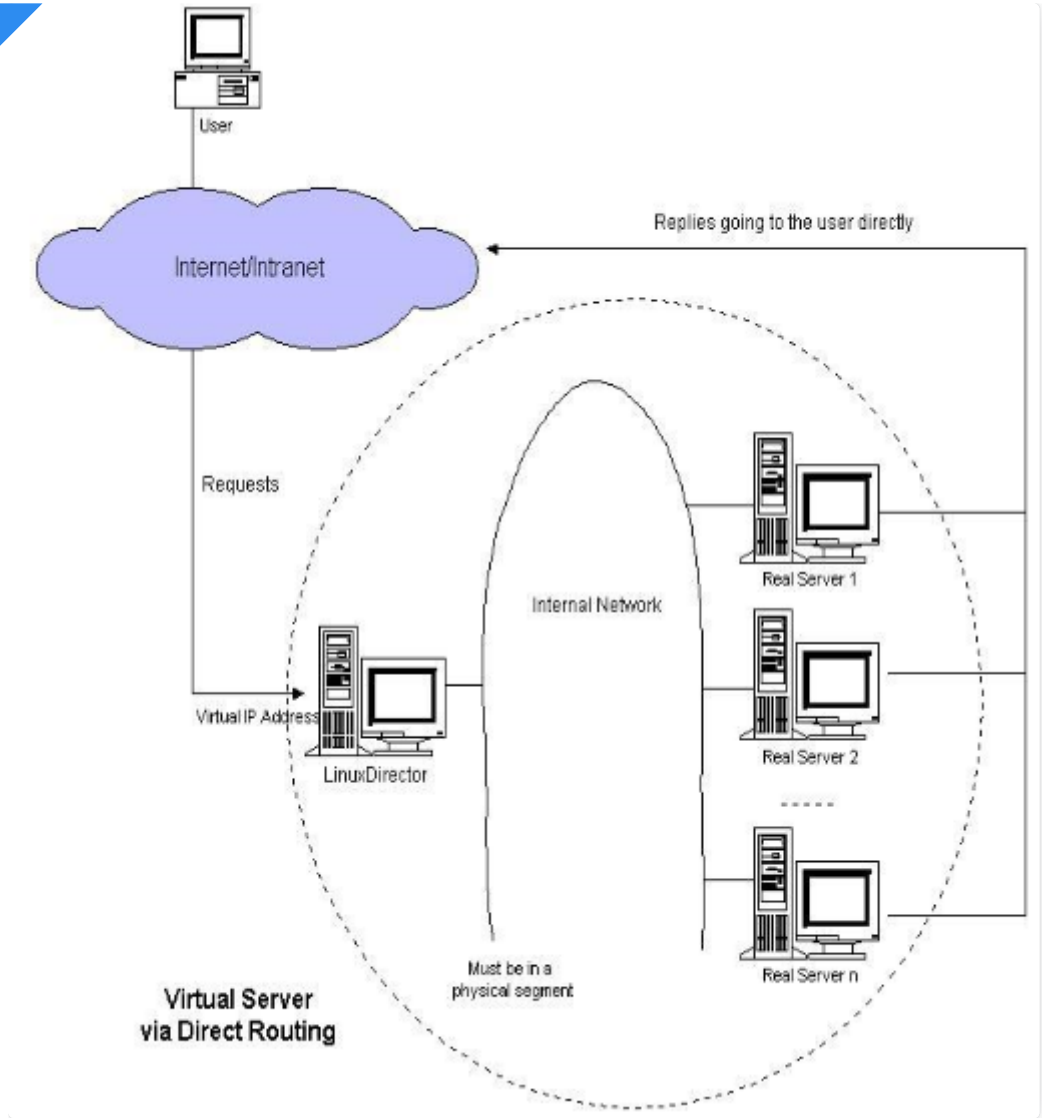
（5）Real Server对比发现目标为自己的IP，开始构建响应报文发回给Director Server。此时报文的源IP为 RIP，目标IP为 CIP。

（6）Director Server在响应客户端前，此时会将源IP地址修改为自己的VIP地址，然后响应给客户端。此时报文的源IP为 VIP，目标IP为 CIP。

DR 模式-直接路由模式

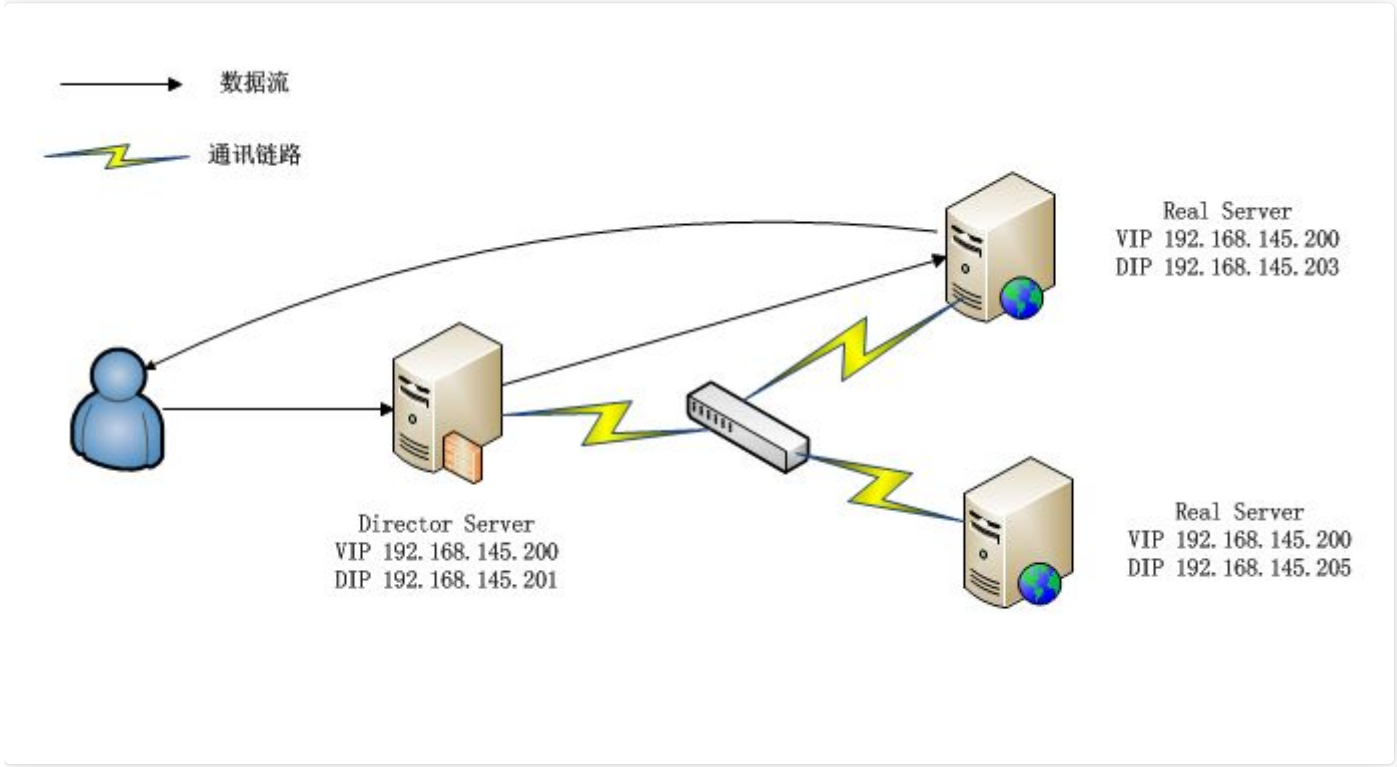
DR模式也就是用直接路由技术实现虚拟服务器。它的连接调度和管理与VS/NAT和VS/TUN中的一样，但它的报文转发方法又有不同，VS/DR通过改写请求报文的MAC地址，将请求发送到Real Server，而Real Server将响应直接返回给客户，免去了VS/TUN中的IP隧道开销。这种方式是三种负载调度机制中性能最高最好的，但是必须要求Director Server与Real Server都有一块网卡连在同一物理网段上。

Director和RealServer必需在物理上有一个网卡通过不间断的局域网相连。 RealServer上绑定的VIP配置在各自Non-ARP的网络设备上(如lo或tunl),Director的VIP地址对外可见，而RealServer的VIP对外是不可见的。RealServer的地址即可以是内部地址，也可以是真实地址。



DR模式是通过改写请求报文的目标MAC地址，将请求发给真实服务器的，而真实服务器响应后的处理结果直接返回给客户端用户。同TUN模式一样，DR模式可以极大的提高集群系统的伸缩性。而且DR模式没有IP隧道的开销，对集群中的真实服务器也没有必要必须支持IP隧道协议的要求。但是要求调度器LB与真实服务器RS都有一块网卡连接到同一物理网段上，必须在同一个局域网环境。

DR 模式工作原理图：



（1）首先用户用CIP请求VIP。

（2）根据上图可以看到，不管是Director Server 还是Real Server 上都需要配置相同的VIP，那么当用户请求到达我们的集群网络的前端路由器的时候，请求数据包的源地址为CIP，目标地址为VIP；此时路由器还会发广播问谁是VIP，那么我们集群中所有的节点都配置有VIP，此时谁先响应路由器那么路由器就会将用户请求发给谁，这样一来我们的集群系统是不是没有意义了，那我们可以在网关路由器上配置静态路由指定VIP就是Director Server，或者使用一种机制不让Real Server 接受来自网络中的ARP 地址解析请求，这样一来用户的请求包都会经过Director Server。

（3）当用户请求到达Director Server，此时请求的数据报文会先到内核空间的PREROUTING链，此时报文的源IP为CIP，目标IP为VIP。

（4）PREROUTING检查发现数据包的目标IP为本机，将数据包发送至INPUT链。

（5）IPVS对比数据包请求的服务是否为集群服务，若是，将请求报文中的源MAC地址修改DIP的MAC地址，将目标MAC地址修改为RIP的MAC地址，然后将数据包发至POSTROUTING链，此时的源IP和目标IP均未修改，仅修改了源MAC地址为DIP的MAC地址，目标MAC地址为RIP的MAC地址。

（6）由于DS和RS在同一个网络中，所以是通过二层来传输，POSTROUTING链检查目标MAC地址为RIP的MAC地址，那么此时数据包将会发至Real Server。

（7）RS发现请求报文的MAC地址是自己的MAC地址，就接收报文。处理完成之后，将相应报文通过lo接口传送给eth0网卡然后向外发出。此时的源IP地址为VIP，目标IP为CIP。

（8）响应报文最终送达至客户端。

配置DR的三种方式：

- **第一种：** 在路由器上明显说明vip对应的地址一定是Director上的MAC，只要绑定，以后再跟vip通信也不用再请求了，这个绑定是静态的，所以它也不会失效，也不会再次发起请求，但是有个前提，我们的路由设备必须有操作权限才能够绑定MAC地址，万一这个路由器是运营商操作的，我们没法操作怎么办？第一种方式固然很简单，但未必可行。
- **第二种：** 在个别主机上（列如：红帽）它们引进的有一种程序arptables，它有点类似iptables，它肯定是基于arp或者MAC做访问控制的，很显然我们只需要在每一个Real Server上定义arptables规则，如果用户arp广播请求的目标地址是本机的vip则不予响应，或者说响应的报文不让出去，很显然（gateway）是接收不到的，也就是director响应的报文才能到达gateway，这个也行。第二种方式我们可以基于arptables。
- **第三种：** 在相对较新的版本中新增了两个内核参数（kernelparameter），第一个是arp_ignore定义接受到ARP请求时的响应级别；第二个是arp_announce定义将自己地址向外通告时的通告级别。[提示：很显然我们现在的系统一般在内核中都是支持这些参数的，我们用参数的方式进行调整更具有朴实性，它还不依赖额外的条件，像arptables，也不依赖外在路由配置的设置，反而通常我们使用的是第三种配置方式]

arp_ignore:定义接收到ARP请求时的响应级别
0：只要本地设置的有相应的地址，就给予响应。（默认）
1：仅回应目标IP地址是本地的入网地址的arp请求。
2：仅回应目标IP地址是本地的入网地址，而且源IP和目标IP在同一个子网的arp请求。
3：不回应网络界面的arp请求，而只对设置的唯一和连接地址做出回应。
4-7：保留未使用。
8：不回应所有的arp请求。

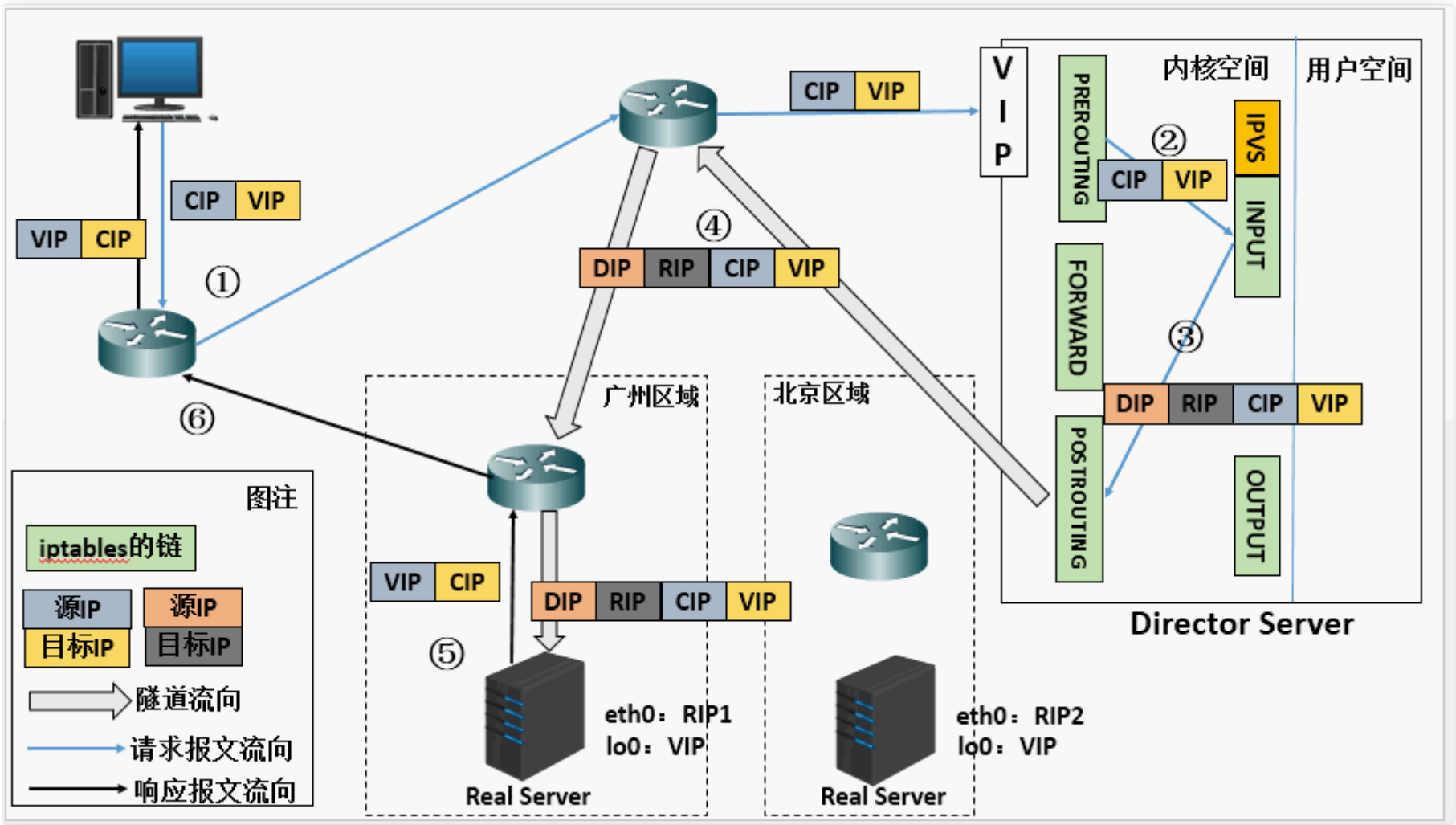
arp_announce：定义将自己地址向外通告的通告级别：
0：将本地任何接口上的任何地址向外通告。
1：视图仅向目标网络通告与其网络匹配的地址。
2：仅向与本地接口上地址匹配的网络进行通告。

DR模式的特性

- 保证前端路由将目标地址为VIP报文统统发给Director Server，而不是RS。
- Director和RS的VIP为同一个VIP。
- RS可以使用私有地址，也可以是公网地址，如果使用公网地址，此时可以通过互联网对RIP进行直接访问。
- RS跟Director Server必须在同一个物理网络中。
- 所有的请求报文经由Director Server，但响应报文必须不能经过Director Server。
- 不支持地址转换，也不支持端口转换。
- RS 可以是大多数常见的操作系统。
- RS 的网关绝不允许指向DIP（因为我们不允许它经过Director）
- RS上的lo接口配置VIP的ip地址
- DR模式是市面上用得最广的。
- 缺陷：RS和DS必须在同一机房。

Tunnel 模式

Tunnel 模式工作原理：



（1）当用户请求到达Director Server，此时请求的数据报文会先拿到内核空间的PREROUTING链，此时报文的源IP为CIP，目标IP为VIP。

（2）PREROUTING检查发现数据包的目标IP是本机，将数据包发送至INPUT链。

（3）IPVS对比数据包请求的服务是否为集群服务，若是，在请求报文的首部再次封装一层IP报文，封装源IP为DIP，目标IP为RIP。然后发至POSTROUTING链，此时源IP为DIP，目标IP为RIP。

（4）POSTROUTING链根据最新封装的IP报文，将数据包发送至RS（因为在外层多封装了一层IP首部，所以可以理解为 此时通过隧道传输）。此时源IP为DIP，目标IP为RIP。

（5）RS接收到报文后发现是自己的IP地址，就将报文接收下来，拆除掉最外层的IP后，会发现里面还有一层IP首部，而且目标是自己的lo接口VIP，那么此时RS开始处理请求，处理完成之后，通过lo接口发送给eth0网卡，然后向外传递。此时源IP为VIP，目标IP为CIP。

（6）响应报文最终送达至客户端。

Tunnel模式的特性

RIP、VIP、DIP全是公网地址。

RS的网关不会也不可能指向DIP。

所有的请求报文经由Director Server，但响应报文必须不能经过Director Server。

不支持端口映射。

RS的系统必须支持隧道。

LVS 的调度算法

固定调度算法：rr，wrr，dh，sh

动态调度算法：wlc，lc，lblc，lblcr

固定调度算法：即调度器不会去判断后端服务器的繁忙与否，一如既往得将请求派发下去。

动态调度算法：调度器会去判断后端服务器的繁忙程度，然后依据调度算法动态得派发请求。

rr：轮询（round robin）

这种算法是最简单的，就是按依次循环的方式将请求调度到不同的服务器上，该算法最大的特点就是简单。轮询算法假设所有的服务器处理请求的能力都是一样的，调度器会将所有的请求平均分配给每个真实服务器，不管后端 RS 配置和处理能力，非常均衡地分发下去。这个调度的缺点是，不管后端服务器的繁忙程度是怎样的，调度器都会讲请求依次发下去。如果A服务器上的请求很快请求完了，而B服务器的请求一直持续着，将会导致B服务器一直很忙，而A很闲，这样便没起到均衡的左右。

wrr：加权轮询（weight round robin）

这种算法比 rr 的算法多了一个权重的概念，可以给 RS 设置权重，权重越高，那么分发的请求数越多，权重的取值范围 0 – 100。主要是对rr算法的一种优化和补充， LVS 会考虑每台服务器的性能，并给每台服务器添加要给权值，如果服务器A的权值为1，服务器B的权值为2，则调度到服务器B的请求会是服务器A的2倍。权值越高的服务器，处理的请求越多。

dh：目标地址散列调度算法 （destination hash）

简单的说，即将同一类型的请求分配给同一个后端服务器，例如将以 .jgp、.png等结尾的请求转发到同一个节点。这种算法其实不是为了真正意义的负载均衡，而是为了资源的分类管理。这种调度算法主要应用在使用了缓存节点的系统中，提高缓存的命中率。

sh：源地址散列调度算法（source hash）

即将来自同一个ip的请求发给后端的同一个服务器，如果后端服务器工作正常没有超负荷的话。这可以解决session共享的问题，但是这里有个问题，很多企业、社区、学校都是共用的一个IP，这将导致请求分配的不均衡。

lc：最少连接数（least-connection）

这个算法会根据后端 RS 的连接数来决定把请求分发给谁，比如 RS1 连接数比 RS2 连接数少，那么请求就优先发给 RS1。这里问题是无法做到会话保持，即session共享。

wlc：加权最少连接数（weight least-connection）

这个比最少连接数多了一个加权的概念，即在最少连接数的基础上加一个权重值，当连接数相近，权重值越大，越优先被分派请求。

lblc：基于局部性的最少连接调度算法（locality-based least-connection）

将来自同一目的地址的请求分配给同一台RS如果这台服务器尚未满负荷，否则分配给连接数最小的RS，并以它为下一次分配的首先考虑。

lblcr：基于地址的带重复最小连接数调度 (Locality-Based Least-Connection with Replication)

这个用得少，可以略过。

NAT 模式实现参考 ->点我

DR 模式实现参考 ->点我