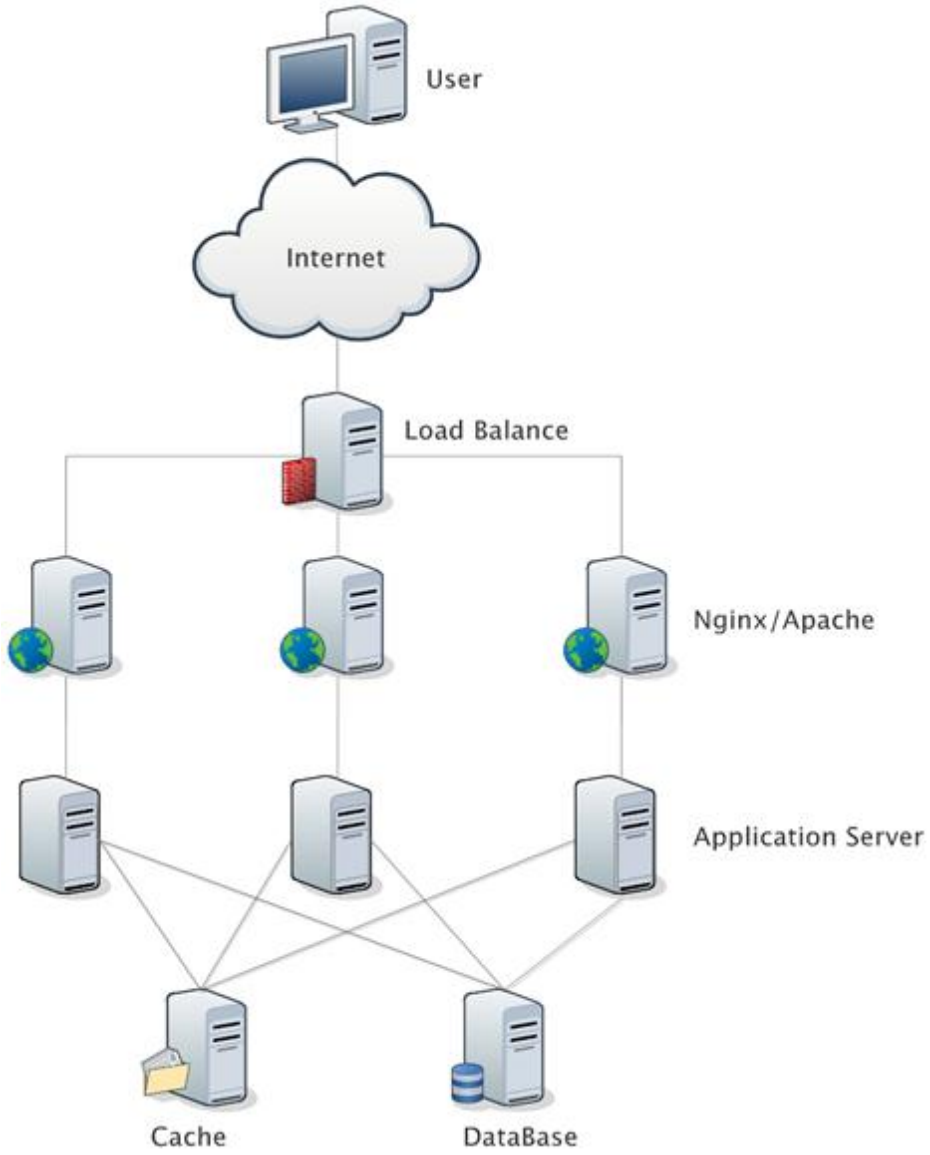
题目：Web 基础架构：负载均衡和 LVS

在大规模互联网应用中，负载均衡设备是必不可少的一个节点，源于互联网应用的高并发和大流量的冲击压力，我们通常会在服务端部署多个无状态的应用服务器和若干有状态的存储服务器（数据库、缓存等等）。

一、负载均衡的作用

负载均衡设备的任务就是作为应用服务器流量的入口，挑选最合适的一台服务器，将客户端的请求转发给它处理，实现客户端到真实服务端的透明转发。最近几年很火的「云计算」以及分布式架构，本质上也是将后端服务器作为计算资源、存储资源，由某台管理服务器封装成一个服务对外提供，客户端不需要关心真正提供服务的是哪台机器，在它看来，就好像它面对的是一台拥有近乎无限能力的服务器，而本质上，真正提供服务的，是后端的集群。

一个典型的互联网应用的拓扑结构是这样的：



二、负载均衡的类型

负载均衡可以采用硬件设备 ,也可以采用软件负载。商用硬件负载设备成本通常较高(一台几十万上百万很正常），所以在条件允许的情况下我们会采用软负载，软负载解决的两个核心问题是：选谁、转发，其中最著名的是 LVS（Linux Virtual Server）。

三、软负载——LVS

LVS 是四层负载均衡，也就是说建立在 OSI 模型的第四层——传输层之上，传输层上有我们熟悉的 TCP/UDP，LVS 支持 TCP/UDP 的负载均衡。

LVS 的转发主要通过修改 IP 地址（NAT 模式，分为源地址修改 SNAT 和目标地址修改 DNAT）、修改目标 MAC（DR 模式）来实现。

那么为什么 LVS 是在第四层做负载均衡？

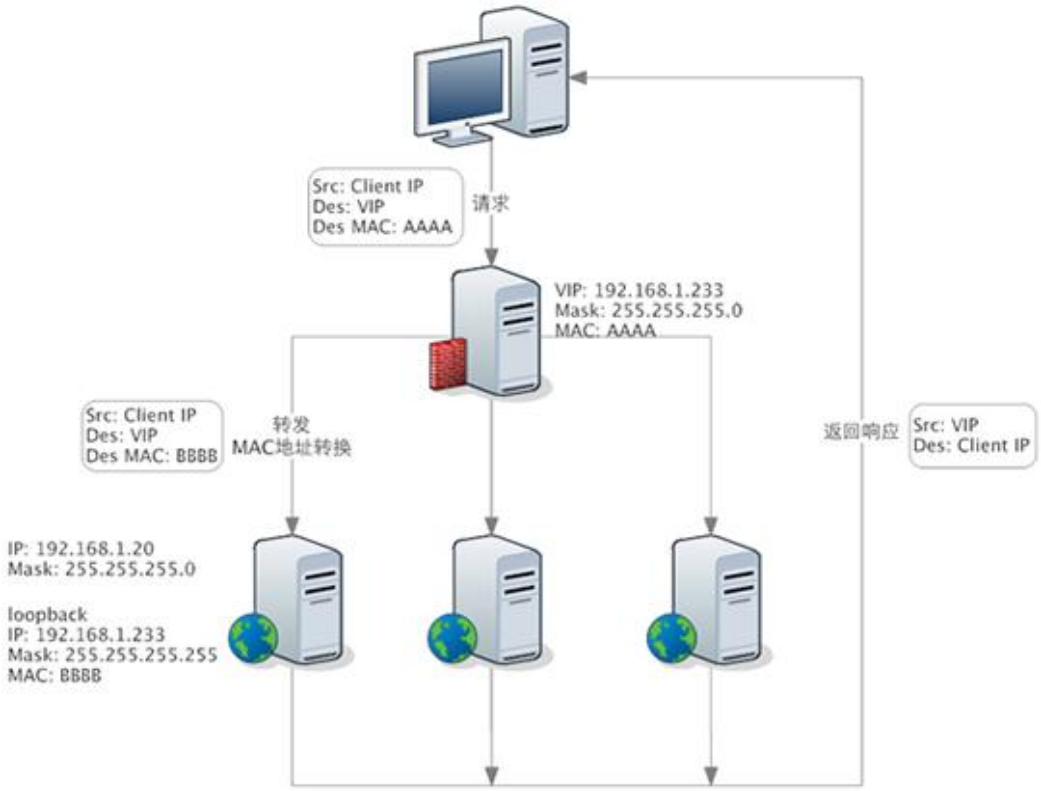
首先 LVS 不像 HAProxy 等七层软负载面向的是 HTTP 包，所以七层负载可以做的 URL 解析等工作，LVS 无法完成。其次，某次用户访问是与服务端建立连接后交换数据包实现的，如果在第三层网络层做负载均衡，那么将失去「连接」的语义。软负载面向的对象应该是一个已经建立连接的用户，而不是一个孤零零的 IP 包。后面会看到，实际上 LVS 的机器代替真实的服务器与用户通过 TCP 三次握手建立了连接，所以 LVS 是需要关心「连接」级别的状态的。

LVS 的工作模式主要有 4 种：

DR
NAT
TUNNEL
Full-NAT

这里挑选常用的 DR、NAT、Full-NAT 来简单介绍一下。

1、DR



请求由 LVS 接受，由真实提供服务的服务器（RealServer, RS）直接返回给用户，返回的时候不经过 LVS。

DR 模式下需要 LVS 和绑定同一个 VIP（RS 通过将 VIP 绑定在 loopback 实现）。

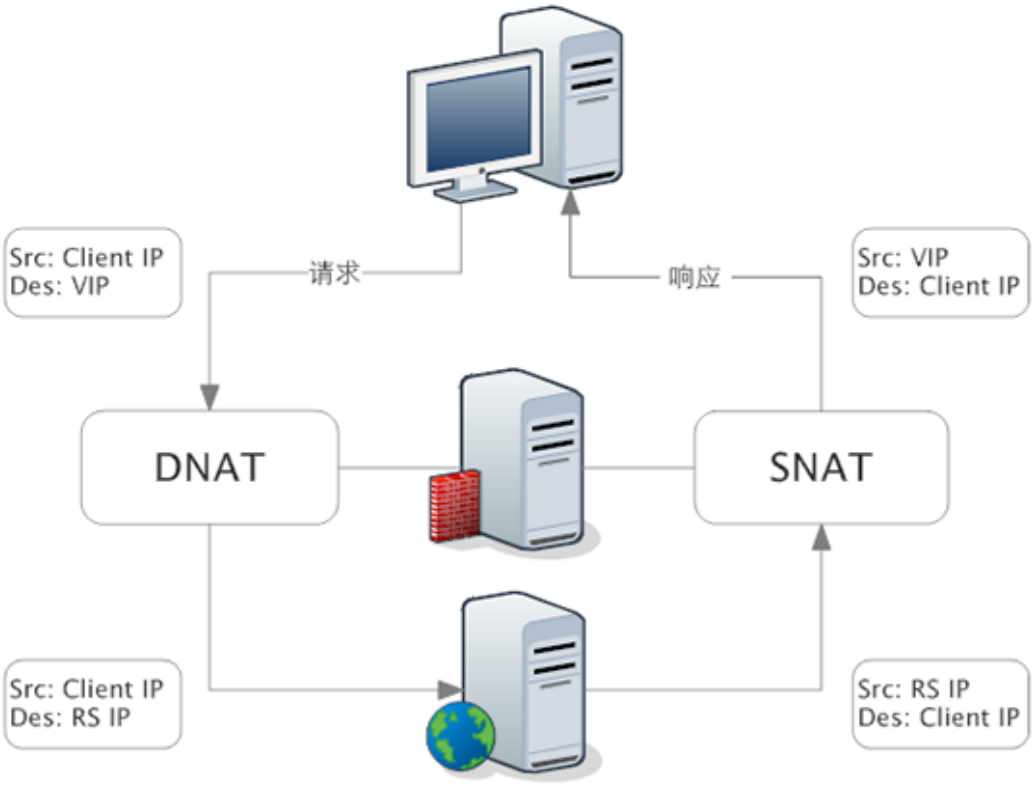
一个请求过来时，LVS 只需要将网络帧的 MAC 地址修改为某一台 RS 的 MAC，该包就会被转发到相应的 RS 处理，注意此时的源 IP 和目标 IP 都没变，LVS 只是做了一下移花接木。

RS 收到 LVS 转发来的包，链路层发现 MAC 是自己的，到上面的网络层，发现 IP 也是自己的，于是这个包被合法地接受，RS 感知不到前面有 LVS 的存在。

而当 RS 返回响应时，只要直接向源 IP（即用户的 IP）返回即可，不再经过 LVS。

DR 模式是性能最好的一种模式。

2、NAT



NAT（Network Address Translation）是一种外网和内网地址映射的技术。

NAT 模式下，网络报的进出都要经过 LVS 的处理。LVS 需要作为 RS 的网关。

当包到达 LVS 时，LVS 做目标地址转换（DNAT），将目标 IP 改为 RS 的 IP。RS 接收到包以后，仿佛是客户端直接发给它的一样。

RS 处理完，返回响应时，源 IP 是 RS IP，目标 IP 是客户端的 IP。

这时 RS 的包通过网关（LVS）中转，LVS 会做源地址转换（SNAT），将包的源地址改为 VIP，这样，这个包对客户端看起来就仿佛是 LVS 直接返回给它的。客户端无法感知到后端 RS 的存在。

3、Full-NAT

无论是 DR 还是 NAT 模式，不可避免的都有一个问题：LVS 和 RS 必须在同一个 VLAN 下，否则 LVS 无法作为 RS 的网关。

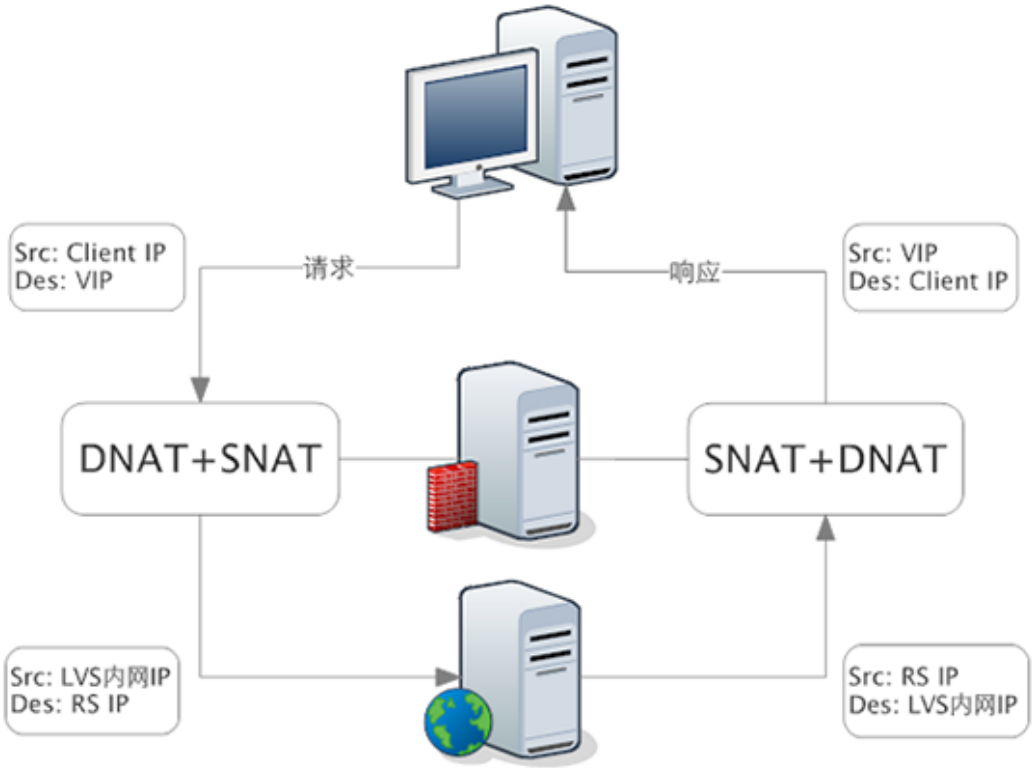
这引发的两个问题是：

1、同一个 VLAN 的限制导致运维不方便，跨 VLAN 的 RS 无法接入。

2、LVS 的水平扩展受到制约。当 RS 水平扩容时，总有一天其上的单点 LVS 会成为瓶颈。

Full-NAT 由此而生，解决的是 LVS 和 RS 跨 VLAN 的问题，而跨 VLAN 问题解决后，LVS 和 RS 不再存在 VLAN 上的从属关系，可以做到多个 LVS 对应多个 RS，解决水平扩容的问题。

Full-NAT 相比 NAT 的主要改进是，在 SNAT/DNAT 的基础上，加上另一种转换，转换过程如下：



在包从 LVS 转到 RS 的过程中，源地址从客户端 IP 被替换成了 LVS 的内网 IP。

内网 IP 之间可以通过多个交换机跨 VLAN 通信。

当 RS 处理完接受到的包，返回时，会将这个包返回给 LVS 的内网 IP，这一步也不受限于 VLAN。

LVS 收到包后，在 NAT 模式修改源地址的基础上，再把 RS 发来的包中的目标地址从 LVS 内网 IP 改为客户端的 IP。

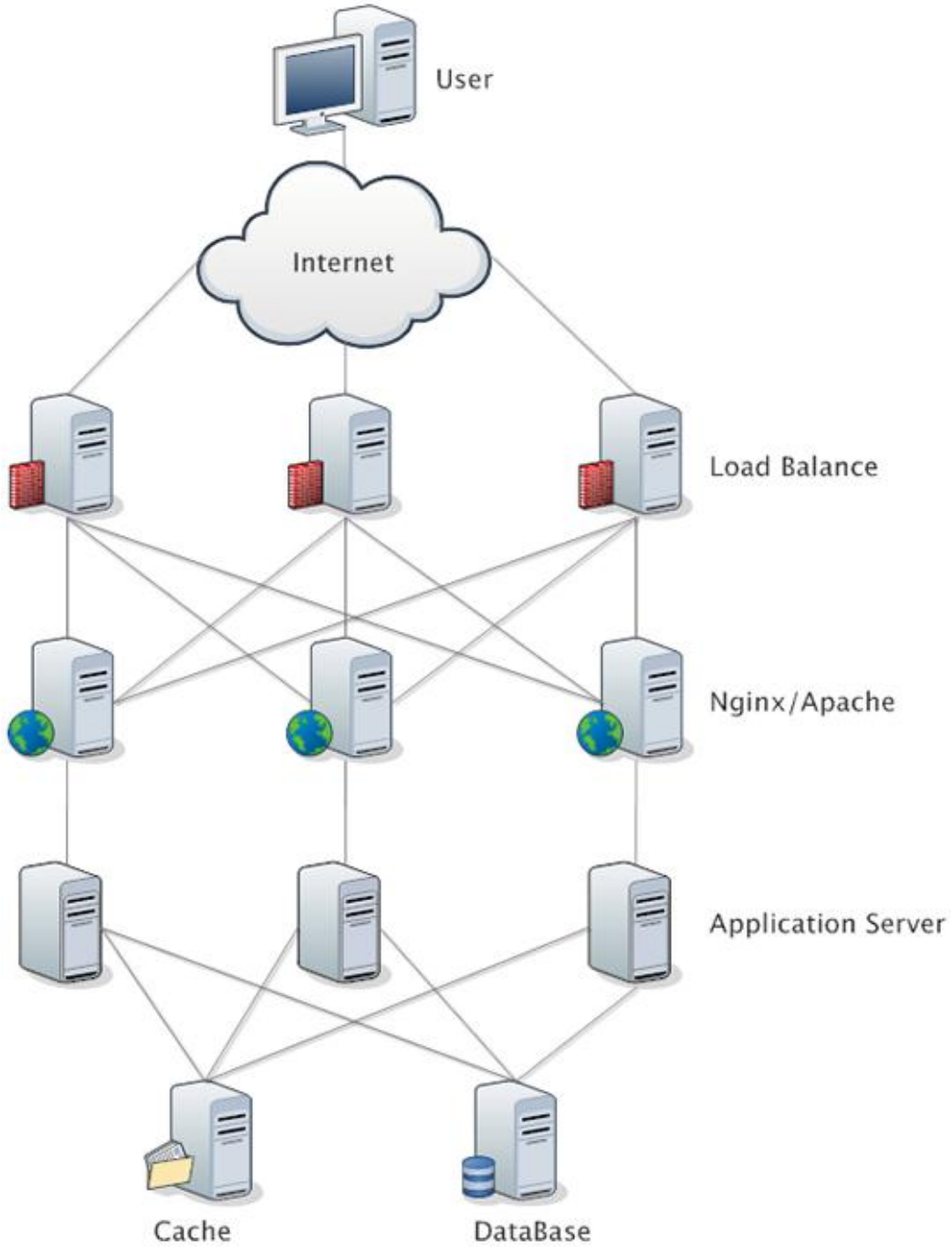
Full-NAT 主要的思想是把网关和其下机器的通信，改为了普通的网络通信，从而解决了跨 VLAN 的问题。采用这种方式，LVS 和 RS 的部署在 VLAN 上将不再有任何限制，大大提高了运维部署的便利性。

4、Session

客户端与服务端的通信，一次请求可能包含多个 TCP 包，LVS 必须保证同一连接的 TCP 包，必须被转发到同一台 RS，否则就乱套了。为了确保这一点，LVS 内部维护着一个 Session 的 Hash 表，通过客户端的某些信息可以找到应该转发到哪一台 RS 上。

5、LVS 集群化

采用 Full-NAT 模式后，可以搭建 LVS 的集群，拓扑结构如下图：



6、容灾

容灾分为 RS 的容灾和 LVS 的容灾。

RS 的容灾可以通过 LVS 定期健康检测实现，如果某台 RS 失去心跳，则认为其已经下线，不会在转发到该 RS 上。

LVS 的容灾可以通过主备+心跳的方式实现。主 LVS 失去心跳后，备 LVS 可以作为热备立即替换。

容灾主要是靠 KeepAlived 来做的。