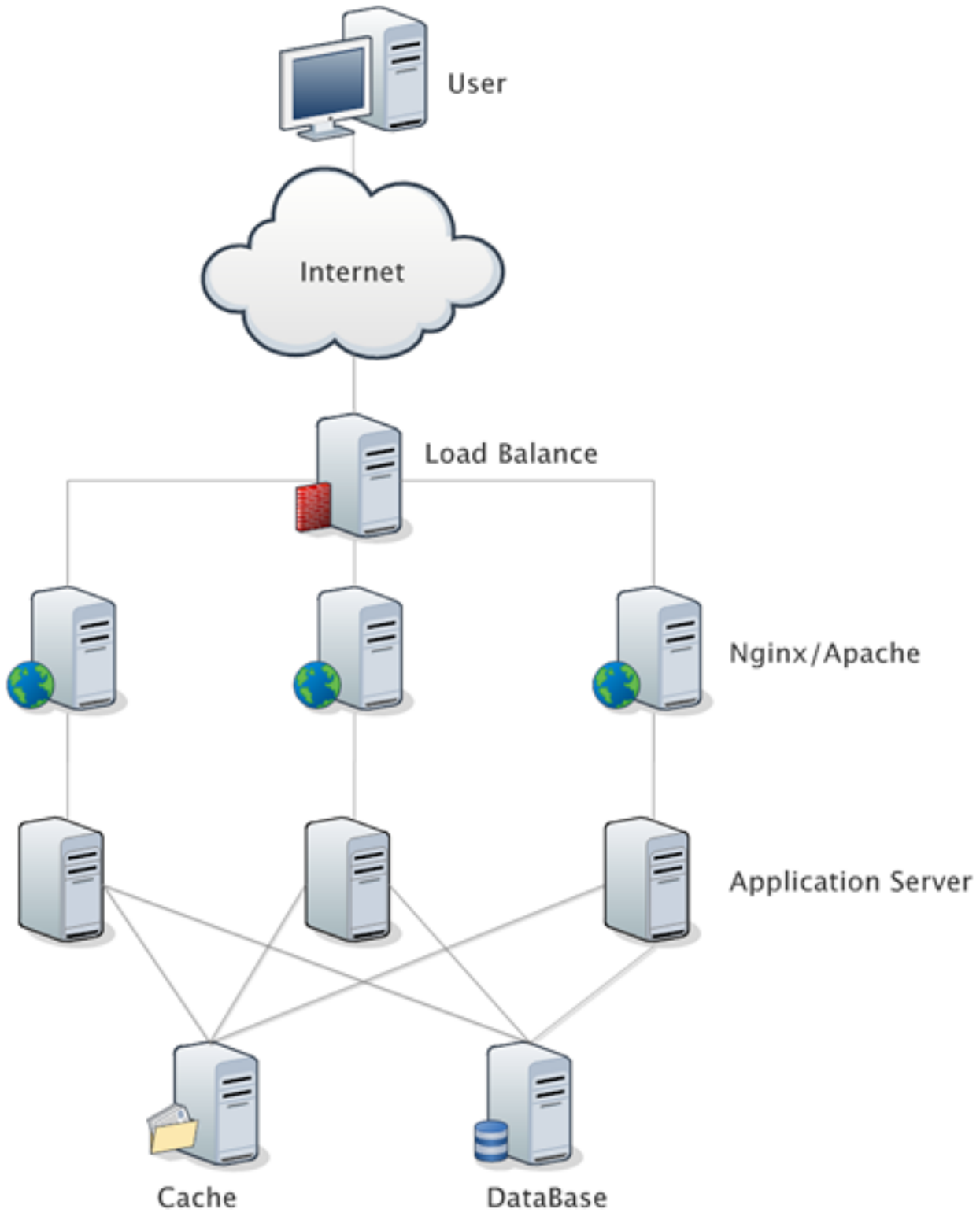
从一个开发的角度看负载均衡和LVS

发表于2013 年 02 月 10 日

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?

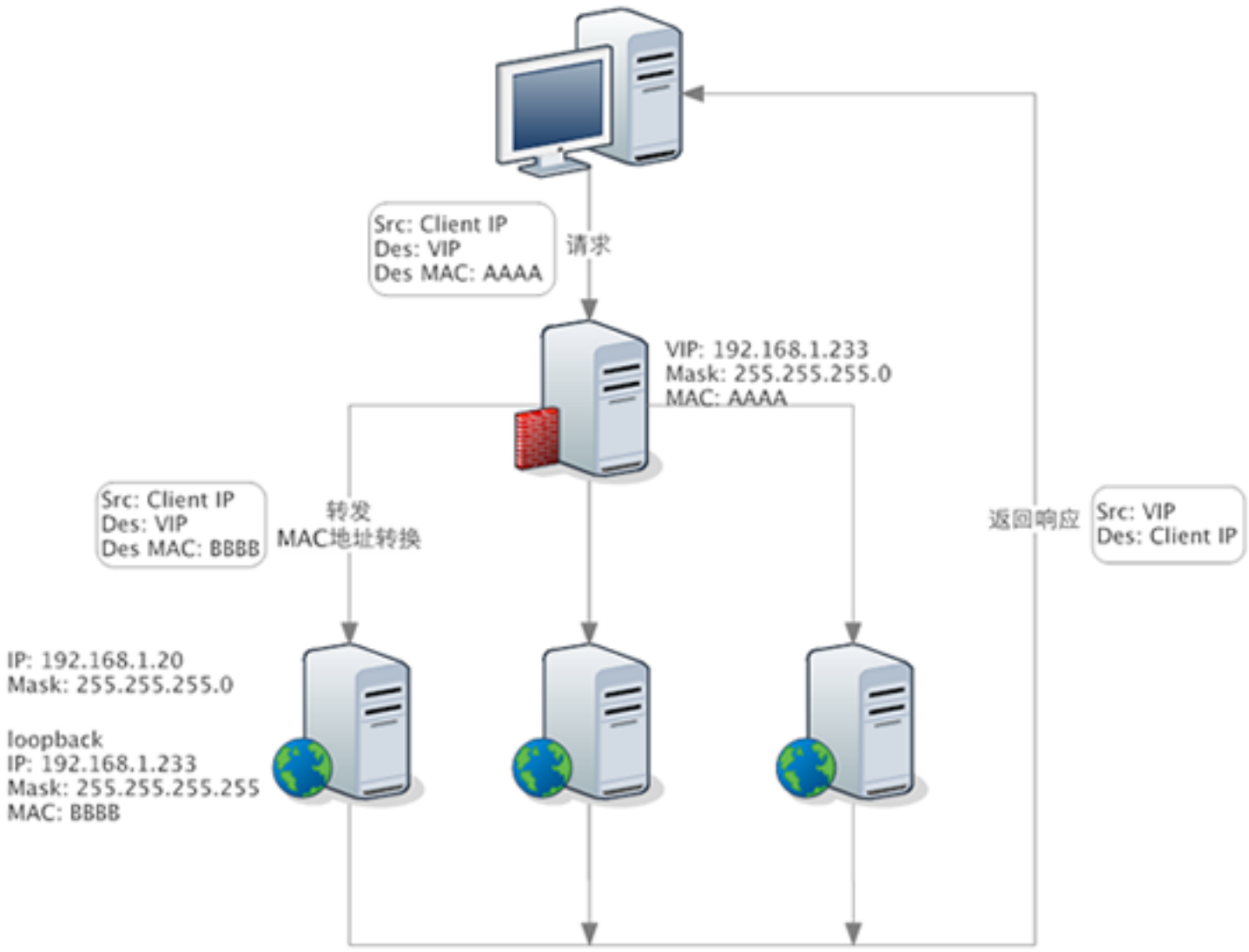
> *首先LVS不像HAProxy等七层软负载面向的是HTTP包，所以七层负载可以做的URL解析等工作，LVS无法完成。其次，某次用户访问是与服务端建立连接后交换数据包实现的，如果在第三层网络层做负载均衡，那么将失去「连接」的语义。软负载面向的对象应该是一个已经建立连接的用户，而不是一个孤零零的IP包。后面会看到，实际上LVS的机器代替真实的服务器与用户通过TCP三次握手建立了连接，所以LVS是需要关心「连接」级别的状态的。*

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。

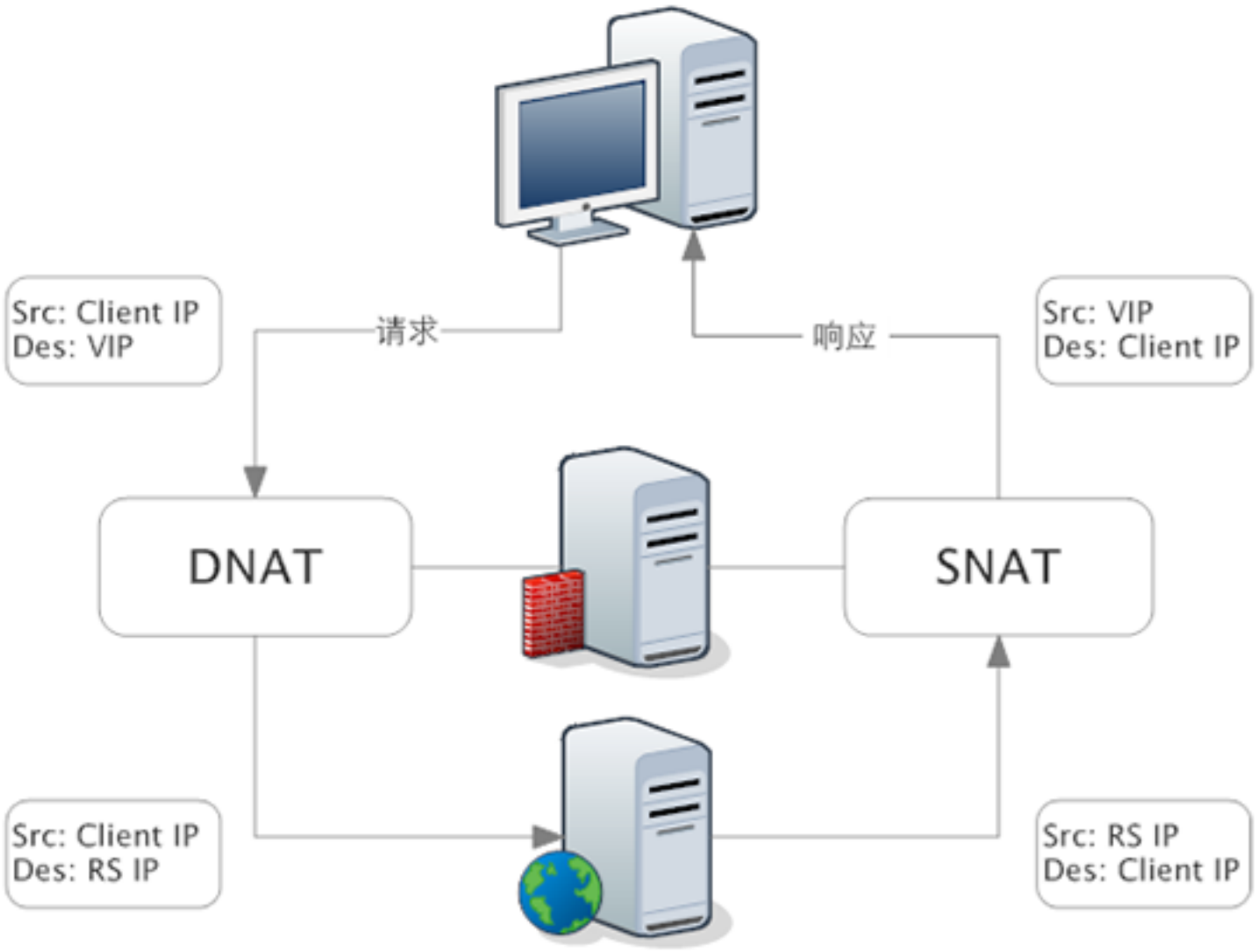
1、DR



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

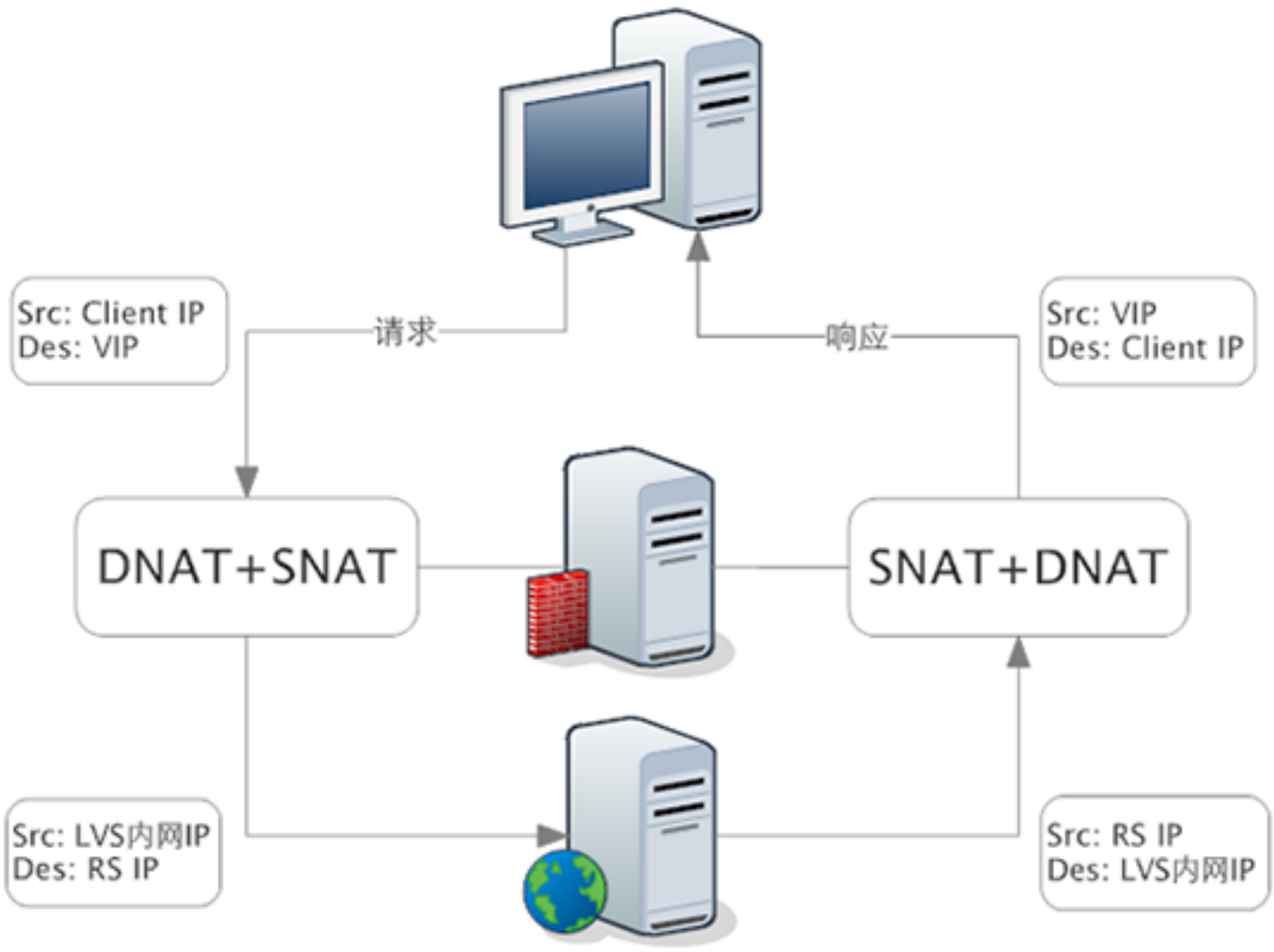
> *无论是DR还是NAT模式，不可避免的都有一个问题：LVS和RS必须在同一个VLAN下，否则LVS无法作为RS的网关。*
>
> *这引发的两个问题是：*

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

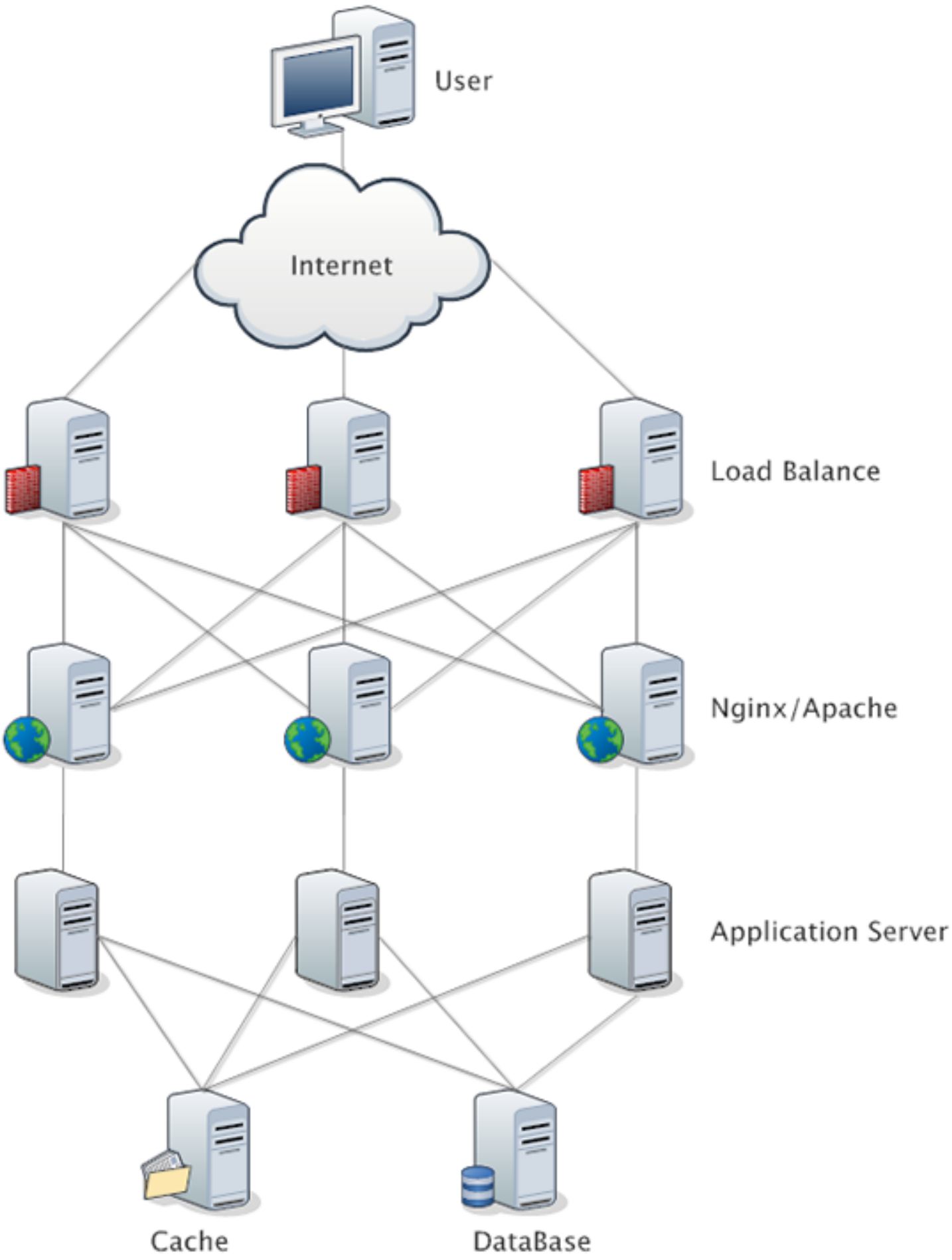
> *的目标地址从LVS内网IP改为客户端的IP。*
>
> *Full-NAT主要的思想是把网关和其下机器的通信，改为了普通的网络通信，从而解决了跨VLAN的问题。采用这种方式，LVS和RS的部署在VLAN上将不再有任何限制，大大提高了运维部署的便利性。*

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。*

更多参考资料：[吴佳明 – LVS在大规模网络环境下的应用 – O'Reilly Velocity China 2012]