此文章已于 11:12:36 2016/5/3 发布到 MADHEX

负载均衡的几种方式

**摘要：**

负载均衡(Load Balance)是集群技术（Cluster）的一种应用。负载均衡可以将工作任务分摊到多个处理单元，从而提高并发处理能力。目前最常见的负载均衡应用是Web负载均衡。根据实现的原理不同，**常见的web负载均衡技术包括：DNS轮询、IP负载均衡和CDN。**其中IP负载均衡可以使用硬件设备或软件方式来实现。

**什么是web负载均衡**

服务器集群(Cluster)使得多个服务器节点能够协同工作，根据目的的不同，服务器集群可以分为：

* 高性能集群：将单个重负载的请求分散到多个节点进行处理，最后再将处理结果进行汇总
* 高可用集群：提高冗余单元，避免单点故障
* 负载均衡集群：**将大量的并发请求分担到多个处理节点。由于单个处理节点的故障不影响整个服务，负载均衡集群同时也实现了高可用性。**

一般提到的负载均衡(Load Balance)，是指实现负载均衡集群。负载均衡实现了横向扩展（Scale Out），避免纵向的升级（Scale Up）换代。

本文中的web负载均衡，特指能够分担web请求（http，https等）的负载均衡技术。

**基本原理**

任何的负载均衡技术都要想办法建立某种一对多的映射机制：一个请求的入口映射到多个处理请求的节点，从而实现分而治之（Divide and Conquer）。

这种映射机制使得多个物理存在对外体现为一个虚拟的整体，对服务的请求者屏蔽了内部的结构。

采用不同的机制建立映射关系，可以形成不同的负载均衡技术，常见的包括：

* DNS轮询
* CDN
* IP负载均衡

**DNS**

DNS轮询是最简单的负载均衡方式。以域名作为访问入口，通过配置多条DNS A记录使得请求可以分配到不同的服务器。

DNS轮询没有快速的健康检查机制，而且只支持WRR的调度策略导致负载很难“均衡”，通常用于要求不高的场景。并且DNS轮询方式直接将服务器的真实地址暴露给用户，不利于服务器安全。

**CDN**

CDN（Content Delivery Network，内容分发网络）。通过发布机制将内容同步到大量的缓存节点，并在DNS服务器上进行扩展，找到里用户最近的缓存节点作为服务提供节点。

因为很难自建大量的缓存节点，所以通常使用CDN运营商的服务。目前国内的服务商很少，而且按流量计费，价格也比较昂贵。

**IP负载均衡**

IP负载均衡是基于特定的TCP/IP技术实现的负载均衡。比如NAT、DR、Turning等。是最经常使用的方式。关于其原理，可以参考下面另一篇文章：[lvs中的负载均衡方式](http://thinkinside.tk/2013/06/02/lvs_lb_strategy.html)。

IP负载均衡可以使用硬件设备，也可以使用软件实现。硬件设备的主要产品是F5-BIG-IP-GTM（简称F5)，**软件产品主要有LVS、HAProxy、NginX。其中LVS、HAProxy可以工作在4-7层，NginX工作在7层。**关于三者的简单对比，可以参考[这里](http://localhost:4000/2012/10/16/weblayer_nginx_keepalived.html#2.3%20%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1%E5%99%A8%E9%80%89%E5%9E%8B-ref)。

硬件负载均衡设备可以将核心部分做成芯片，性能和稳定性更好，而且商用产品的可管理性、文档和服务都比较好。唯一的问题就是价格。

软件负载均衡通常是开源软件。自由度较高，但学习成本和管理成本会比较大。

**F5**

F5的全称是F5-BIG-IP-GTM，是最流行的硬件负载均衡设备，其并发能力达到百万级。F5的主要特性包括：

1. **多链路的负载均衡和冗余**

可以接入多条ISP链路，在链路之间实现负载均衡和高可用。

1. **防火墙负载均衡**

F5具有异构防火墙的负载均衡与故障自动排除能力。

1. **服务器负载均衡**

这是F5最主要的功能，F5可以配置针对所有的对外提供服务的服务器配置Virtual Server实现负载均衡、健康检查、回话保持等。

1. **高可用**

F5设备自身的冗余设计能够保证99.999%的正常运行时间，双机F5的故障切换时间为毫秒级。

使用F5可以配置整个集群的链路冗余和服务器冗余，提高可靠的健康检查机制，以保证高可用。

1. **安全性**

与防火墙类似，F5采用缺省拒绝策略，可以为任何站点增加额外的安全保护，防御普通网络攻击，包括DDoS、IP欺骗、SYN攻击、teartop和land攻击、ICMP攻击等。

1. **易于管理**

F5提供HTTPS、SSH、Telnet、SNMP等多种管理方式，包含详尽的实时报告和历史纪录报告。同时还提供二次开发包(i-Control)。

1. 其他

F5还提供了SSL加速、软件升级、IP地址过滤、带宽控制等辅助功能。

**LVS**

LVS(Linux Virtual Server, Linux虚拟服务器），是[章文嵩博士](http://zh.linuxvirtualserver.org/)开发的开放软件，目前已经集成到Linux内核中。

基于不同的网络技术，**LVS支持多种负载均衡机制。包括：VS/NAT（基于网络地址转换技术）、VS/TUN（基于IP隧道技术）和VS/DR（基于直接路由技术）。**

此外，为了适应不同的需要，淘宝开发了[VS/FULLNAT](http://kb.linuxvirtualserver.org/wiki/IPVS_FULLNAT_and_SYNPROXY)，从本质上来说也是基于网络地址转换技术。最近还有一个基于VS/FULLNAT的[DNAT](https://github.com/xiaomi-sa/dsnat)模块。

不管使用哪种机制，LVS都不直接处理请求，而是将请求转发到后面真正的服务器(Real Server)。不同的机制，决定了响应包如何返回到客户端。

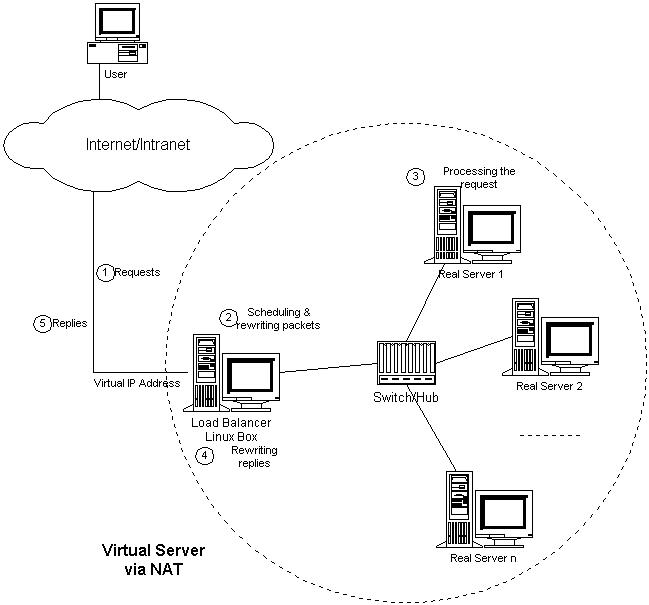
**VS/NAT**

[NAT（Network Address Translation，网络地址转换）](http://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E5%9C%B0%E5%9D%80%E8%BD%AC%E6%8D%A2)也叫做网络掩蔽或者IP掩蔽，是将IP 数据包头中的IP 地址转换为另一个IP 地址的过程。

NAT能够将私有（保留）地址转化为合法IP地址，通常用于一个公共IP地址和多个内部私有IP地址直接的映射，广泛应用于各种类型Internet接入方式和各种类型的网络中。

通过使用NAT将目的地址转换到多个服务器的方式，可以实现负载均衡，同时能够隐藏并保护内部服务器，避免来自网络外部的攻击。商用负载均衡设备如Cisco的LocalDirector、F5的Big/IP和Alteon的ACEDirector都是基于NAT方法。

VS/NAT(Virtual Server via Network Address Translation)是基于NAT技术实现负载均衡的方法。其[**架构**](http://lib.csdn.net/base/16)如下图所示：



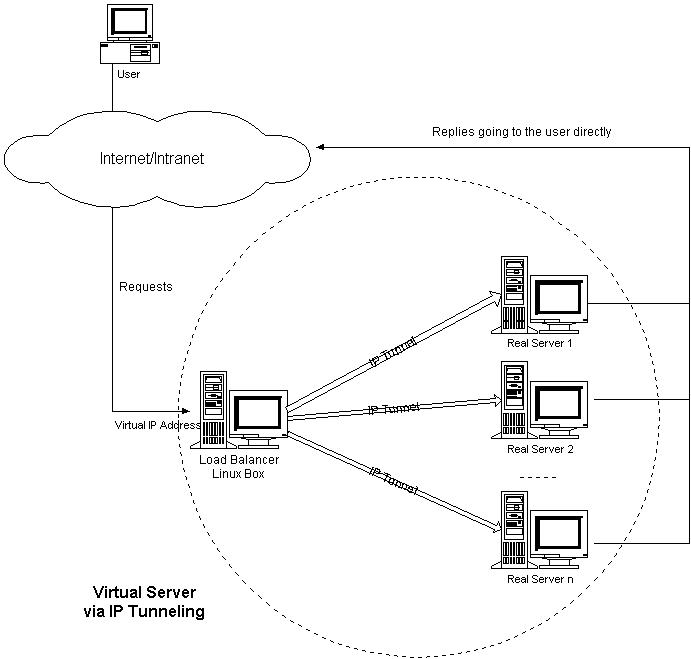
1. 客户通过Virtual IP Address（虚拟服务的IP地址）访问网络服务时，请求报文到达调度器
2. 调度器根据连接调度算法从一组真实服务器中选出一台服务器，将报文的目标地址Virtual IP Address改写成选定服务器的地址，报文的目标端口改写成选定服务器的相应端口，最后将修改后的报文发送给选出的服务器。
3. 真实的服务器处理请求，并将响应报文发到调度器。
4. 调度器将报文的源地址和源端口改为Virtual IP Address和相应的端口
5. 调度器将修改过的报文发给用户

**在VS/NAT的集群系统中，请求和响应的数据报文都需要通过负载调度器，当真实服务器的数目在10台和20台之间时，负载调度器将成为整个集群系统的新瓶颈。大多数Internet服务都有这样的特点：请求报文较短而响应报文往往包含大量的数据。如果能将请求和响应分开处理，即在负载调度器中只负责调度请求而响应直接返回给客户，将极大地提高整个集群系统的吞吐量。**比如IP隧道技术。

**VS/TUN**

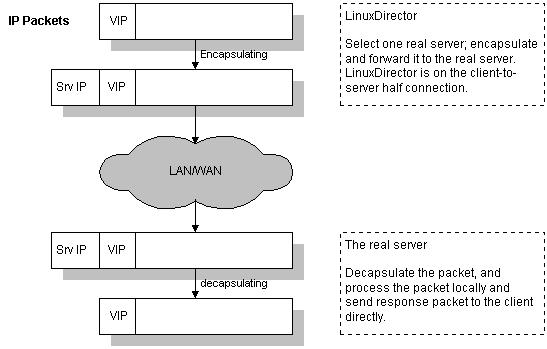
[IP Tunneling(IP隧道)技术](http://baike.baidu.cn/view/467497.htm)，又称为IP封装技术(IP encapsulation)，是一种在网络之间传递数据的方式。可以将一个IP报文封装到另一个IP报文（可能是不同的协议）中，并转发到另一个IP地址。IP隧道主要用于移动主机和虚拟私有网络（Virtual Private Network），在其中隧道都是静态建立的，隧道一端有一个IP地址，另一端也有唯一的IP地址。

VS/TUN（Virtual Server via IP Tunneling）是基于隧道技术实现负载均衡的方法。其架构如下图所示：



VS/TUN与VS/NAT的工作机制大体上相同，区别在于：

1. 调度器转发报文的时候进行了协议的二次封装，真实的服务器接收到请求后先进行解包。过程如下图所示：



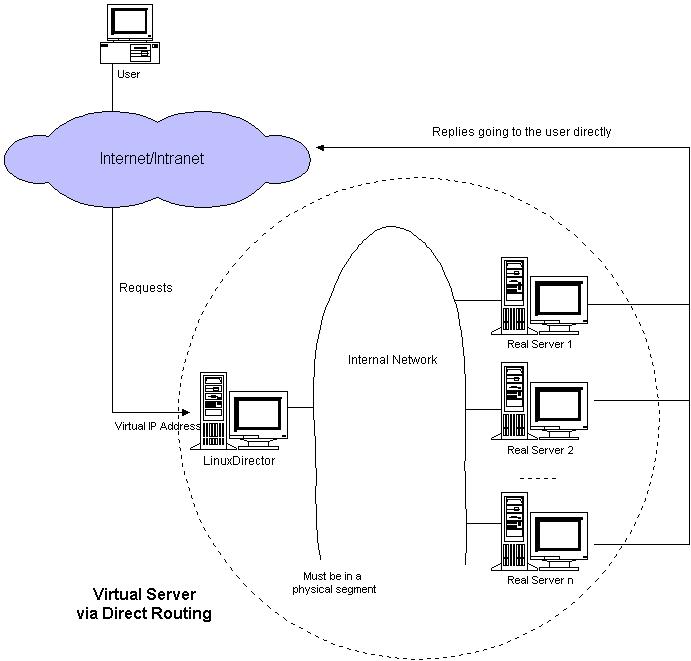
1. 响应报文从后端服务器直接返回给客户，不需要经过调度器。

**VS/DR**

[DR(Direct Routing, 直接路由)](http://baike.baidu.cn/view/3089936.htm), 路由器学习路由的方法之一。路由器对于自己的网络接口所直连的网络之间的通信，可以自动维护路由表，而且不需要进行路由计算。

直接路由通常用在一个三层交换机连接几个VLAN的情况，只要设置直接路由VLAN之间就可以通信，不需要设置其他的路由方式。

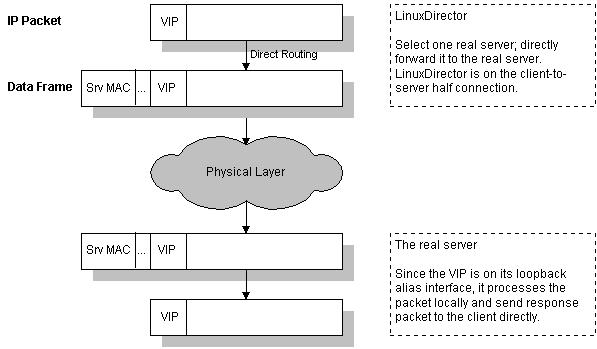
VS/DR(Virtual Server via Direct Routing)是基于直接路由实现负载均衡的方法。其架构如下图所示：



跟VS/TUN方法相同，VS/DR利用大多数Internet服务的非对称特点，负载调度器中只负责调度请求，而服务器直接将响应返回给客户，可以极大地提高整个集群系统的吞吐量。

VS/DR要求调度器和服务器组都必须在物理上有一个网卡通过不分段的局域网相连，即通过交换机或者高速的HUB相连，中间没有隔有路由器。VIP地址为调度器和服务器组共享，调度器配置的VIP地址是对外可见的，用于接收虚拟服务的请求报文；所有的服务器把VIP地址配置在各自的Non-ARP网络设备上，它对外面是不可见的，只是用于处理目标地址为VIP的网络请求。

**VS/DR的整个过程与VS/TUN非常类似，不同之处在于调度器不对请求包进行二次封装，只是将目标MAC地址更改为经过调度算法选出的目标服务器的MAC地址。**如下图：



**三种方法的优缺点比较**

**VS/NAT**

* 优点
  + 对后端服务器的操作系统无要求
  + 只需要一个IP地址配置在调度器上，服务器组可以用私有的IP地址。
  + 支持端口映射
* 缺点
  + 请求和响应报文都需要通过调度器，伸缩能力有限（10+)
  + 要求服务器和调度器在同一个VLAN
  + 需要将服务器的默认网关指向调度器
  + 对于那些将IP地址或者端口号在报文数据中传送的网络服务，需要编写相应的应用模块来转换报文数据中的IP地址或者端口号

**VS/TUN**

* 优点
  + 不需要调度应答报文，性能高
  + 服务器和调度器可以不在同一个VLAN
  + 支持广域负载均衡
* 缺点
  + 所有的服务器必须支持“IP Tunneling”协议，要安装内核模块（比如IPIP等），配置复杂
  + 有建立IP隧道的开销
  + 服务器上直接绑定虚拟IP(Virtaul IP)，风险很大
  + 服务器需要联通外网
  + 不支持端口映射

**VS/DR**

* 优点
  + 与VS/TUN相比，没有IP隧道的开销，性能最好
* 缺点
  + 要求调度器与服务器都有一块网卡连在同一物理网段（同一个VLAN）上
  + 要求服务器网络设备（或者设备别名）不作ARP响应，或者能将报文重定向（Redirect）到本地的Socket端口上
  + 服务器上直接绑定虚拟IP(Virtaul IP)，风险很大
  + 不支持端口映射

**VS/FULLNAT**

如上节所述，前面三种传统的负载均衡机制各自存在一些不足。

[VS/FULLNAT](http://kb.linuxvirtualserver.org/wiki/IPVS_FULLNAT_and_SYNPROXY)是为了解决这些不足而新开发的一种转发模式。VS/FULLNAT的特点是：

1. 调度器和服务器可以跨VLAN通信，不需要配置在同一个网段
2. 请求和应答报文都经过调度器，服务器不需要绑定虚拟IP

VS/FULLNAT这两个特点可以简化网络拓扑，降低运维成本和风险。

**如何选择**

1. 如果人少钱多，不在乎性能的损耗愿意多买服务器，同时希望最大程度较少运维的工作量，可以选择FULLNAT
2. 很大众的方式是用DR，没有太多的优点但也没有太多的缺点
3. 如果要搞广域网负载均衡，那就用TUN吧
4. 个人感觉NAT不是为了互联网用的。小并发的实验性应用或者用在非web场合，比如mysql集群等。当然，如果需要端口映射，必须使用NAT方式

本文转自：

<http://thinkinside.tk/2013/06/03/lb_solutions_list.html>

<http://thinkinside.tk/2013/06/02/lvs_lb_strategy.html>