常见的负载均衡技术及其应用

摘要

本文首先介绍了负载均衡的概念，其次对目前主流的负载均衡技术进行简单的介绍，着重分析比较了四层负载均衡技术和七层负载均衡技术，对它们各自的应用场景 Lvs和Nginx和用到的算法进行简单介绍。

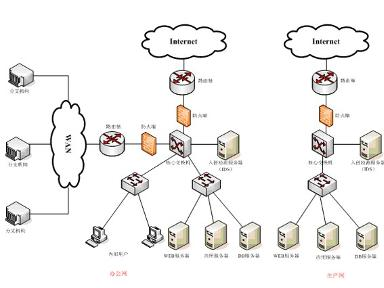
**关键字**：负载均衡、四层负载均衡技术、七层负载均衡技术、 Lvs、Nginx

# 引言

由于现有网络的各个核心部分随着业务量的提高，访问量和数据流量的快速增长，其处理能力和计算强度也相应地增大，使得单一的[服务器设备](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=168454890&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)根本无法承担。在此情况下，如果扔掉现有设备去做大量的硬件升级，这样将造成现有资源的浪费，而且如果再面临下一次业务量的提升时，这又将导致再一次硬件升级的高额成本投入，甚至性能再卓越的设备也不能满足当前业务量增长的需求。为解决此类问题，负载均衡的概念被提出。

# 负载均衡含义

负载均衡（又称为负载分担），英文名称为Load Balance，其意思就是将负载（工作任务）进行平衡、分摊到多个操作单元上进行执行，例如[Web服务器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=267249&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)、[FTP服务器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=449752&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)、企业关键[应用服务器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=4341240&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)和其它关键任务服务器等，从而共同完成工作任务。

[](https://baike.sogou.com/PicBooklet.v?relateImageGroupIds=%26lemmaId=1346976%26now=https:/pic.baike.soso.com/p/20140128/20140128100957-613478386.jpg%26type=1)

图一 负载均衡

需要说明的是：负载均衡设备不是基础网络设备，而是一种性能优化设备。对于网络应用而言，并不是一开始就需要负载均衡，当网络应用的访问量不断增长，单个处理单元无法满足负载需求时，网络应用流量将要出现瓶颈时，负载均衡才会起到作用。

负载均衡有两方面的含义：其一是将大量的并发处理转发给后端多个节点处理，减少工作响应时间；其二是将单个繁重的工作转发给后端多个节点处理，处理完再返回给负载均衡中心，再返回给用户。

负载均衡的目的是创建一个系统：将来自实际运行服务的物理服务器中的“服务”（用户访问站点）进行虚拟化处理，在大量物理服务器之间实现负载均衡，并使这些服务器对外界看起来犹如一个大服务器，为用户提供高可用性、高扩展性、可预测的应用服务。

# 负载均衡发展过程

负载均衡技术经历了一个迅速而又曲折的发展过程：

在萌芽期，专用的负载均衡设备尚未实现商用，许多人尝试利用当时已有技术实现高扩展性和高可用性的目标。DNS轮询（Domain Name System：域名系统）是当时最常用的技术。从可扩展性的角度来看，该方法具有优良的可扩展性：新增服务器后只需在DNS的域名列表中新增一个IP即可。但是从可用性角度来看，该方法几乎没有可用性的任何保证。DNS轮询方法由于不具有对服务器工作状态的解析能力，只能提供一种不可控制的负载分配，无法实现真正意义上的负载均衡。

在发展期：基于应用的负载均衡解决方案在小规模范围表现良好。这类解决方案是在应用服务器的应用软件或操作系统上直接部署负载均衡能力。大部分解决方案设计到基本的网络欺骗，其中典型的方法是让集群中的所有服务器监听（接收）除自身物理地址外集群IP。基于应用的负载均衡解决方案扩展性强，只需构建一台服务器，并将其加到集群中，即可进行扩容。然而集群成员之间需要互相保持联系以获得其它服务器的连接信息，新服务器的加入，使得集群成员服务器间的网络流量呈指数型增长。因此该方法只适用于较小规模的集群。

在成熟期，基于网络的负载均衡方案采用一些具有强大功能的基于网络的设备，且这些设备与应用无关，位于应用服务器的外部，因此能够采用更直接的网络技术实现负载均衡。从本质上讲，这些设备会向外部世界展现一个虚拟服务器地址，而当用户试图连接时，它会将连接通过双向网络地址转换（NAT）转移到最适合的真正的服务器上。这是真正意义上的负载均衡技术。

# 负载均衡技术分类

目前负载均衡技术大多数是用于提高诸如在Web服务器、FTP服务器和其它关键任务服务器上的Internet服务器程序的可用性和可伸缩性。现在的负载均衡技术多种多样。

4.1 根据所采用的设备对象可将负载均衡分为：软件负载均衡和硬件负载均衡。

软件负载均衡解决方案：是指在一台或多台服务器相应的操作系统上安装一个或多个附加软件来实现负载均衡，如DNS Load Balance，CheckPoint Firewall-1 ConnectControl，Keepalive+ipvs等，它的优点是基于特定环境，配置简单，使用灵活，成本低廉，可以满足一般的负载均衡需求。软件解决方案缺点也较多，因为每台服务器上安装额外的软件运行会消耗系统不定量的资源，越是功能强大的模块，消耗得越多，所以当连接请求特别大的时候，软件本身会成为服务器工作成败的一个关键；软件可扩展性并不是很好，受到操作系统的限制；由于操作系统本身的Bug，往往会引起安全问题。

硬件负载均衡解决方案：是直接在服务器和外部网络间安装负载均衡设备，这种设备通常是一个独立于系统的硬件，我们称之为负载均衡器。由于专门的设备完成专门的任务，独立于操作系统，整体性能得到大量提高，加上多样化的负载均衡策略，智能化的流量管理，可达到最佳的负载均衡需求。负载均衡器有多种多样的形式，除了作为独立意义上的负载均衡器外，有些负载均衡器集成在交换设备中，置于服务器与Internet链接之间，有些则以两块网络适配器将这一功能集成到PC中，一块连接到Internet上，一块连接到后端服务器群的内部网络上。

## 4.2根据应用的地理结构可将负载均衡分为：本地负载均衡和全局负载均衡。

本地负载均衡是指对本地的服务器群做负载均衡，它能有效地解决数据流量过大、网络负荷过重的问题，并且不需花费昂贵开支购置性能卓越的服务器，充分利用现有设备，避免服务器单点故障造成数据流量的损失。其有灵活多样的均衡策略把数据流量合理地分配给服务器群内的服务器共同负担。即使是再给现有服务器扩充升级，也只是简单地增加一个新的服务器到服务群中，而不需改变现有网络结构、停止现有的服务。

全局负载均衡是指对分别放置在不同的地理位置、有不同网络结构的服务器群间作负载均衡。它主要用于在一个多区域拥有自己服务器的站点，为了使全球用户只以一个IP地址或域名就能访问到离自己最近的服务器，从而获得最快的访问速度，也可用于子公司分散站点分布广的大公司通过Intranet（企业内部互联网）来达到资源统一合理分配的目的。

## 4.3根据OSI模型可将负载均衡分为：二层负载均衡、三层负载均衡、四层负载均衡、七层负载均衡。

二层负载均衡是基于数据链路层的负债均衡，即让负债均衡服务器和业务服务器绑定同一个虚拟IP（即VIP），客户端直接通过这个VIP进行请求。这种方式负载方式虽然控制粒度比较粗，但是优点是负载均衡服务器的压力会比较小，负载均衡服务器只负责请求的进入，不负责请求的响应（响应是有后端业务服务器直接响应给客户端），吞吐量会比较高。

三层负载均衡是基于网络层的负载均衡，通俗的说就是按照不同机器不同IP地址进行转发请求到不同的机器上。

这种方式虽然比二层负载多了一层，但从控制的颗粒度上看，并没有比二层负载均衡更有优势，并且，由于请求的进出都要经过负载均衡服务器，会对其造成比较大的压力，性能也比二层负载均衡要差。

四层负载均衡是基于IP+端口的负载均衡：在三层负载均衡的基础上，通过发布三层的IP地址（VIP），然后加四层的端口号，来决定哪些流量需要做负载均衡，对需要处理的流量进行NAT处理，转发至后台服务器，并记录下这个TCP或者UDP的流量是由哪台服务器处理的，后续这个连接的所有流量都同样转发到同一台服务器处理。

七层负载均衡是基于虚拟的URL或主机IP的负载均衡：在四层负载均衡的基础上（没有四层是绝对不可能有七层的），再考虑应用层的特征，比如同一个Web服务器的负载均衡，除了根据VIP加80端口辨别是否需要处理的流量，还可根据七层的URL、浏览器类别、语言来决定是否要进行负载均衡。

# 常用技术的应用场景

OSI是一个[开放性](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%80%A7/3129237" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E4%B8%83%E5%B1%82%E5%8D%8F%E8%AE%AE/_blank)的通信系统互连参考模型，它是一个定义得非常好的协议规范。OSI模型有7层结构，每层都可以有几个子层。 OSI的7层从上到下分别有7[应用层](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E5%B1%82/4329788" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E4%B8%83%E5%B1%82%E5%8D%8F%E8%AE%AE/_blank)、6[表示层](https://baike.baidu.com/item/%E8%A1%A8%E7%A4%BA%E5%B1%82/4329716" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E4%B8%83%E5%B1%82%E5%8D%8F%E8%AE%AE/_blank)、5会话层、4[传输层](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E5%B1%82/4329536" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E4%B8%83%E5%B1%82%E5%8D%8F%E8%AE%AE/_blank)、3网络层、2[数据链路层](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF%E5%B1%82/4329290" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E4%B8%83%E5%B1%82%E5%8D%8F%E8%AE%AE/_blank)、1[物理层](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%B1%82/4329158" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E4%B8%83%E5%B1%82%E5%8D%8F%E8%AE%AE/_blank)。其中高层（即7、6、5、4层）定义了应用程序的功能，下面3层（即3、2、1层）主要面向通过网络的[端到端](https://baike.baidu.com/item/%E7%AB%AF%E5%88%B0%E7%AB%AF/8851783" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E4%B8%83%E5%B1%82%E5%8D%8F%E8%AE%AE/_blank)，点到点的[数据流](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%B5%81/3002243" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E4%B8%83%E5%B1%82%E5%8D%8F%E8%AE%AE/_blank)。

针对网络上负载过重的不同瓶颈所在，从网络的不同层次入手，我们可以采用相应的负载均衡技术来解决现有问题。 在上述的负载均衡技术中，四层负载均衡和七层负载均衡技术是很常用的两种，对其进行研究有助于新的负载均衡技术的提出与改进。

## **5.1**四层/七层负载均衡含义

四层负载均衡工作在OSI模型中的四层，即传输层。四层负载均衡只能根据报文中目标地址和源地址对请求进行转发，而无法修改或判断所请求资源的具体类型，然后经过负载均衡内部的调度算法转发至要处理请求的服务器。四层负载均衡单纯的提供了终端到终端的可靠连接，并将请求转发至后端，连接至始至终都是同一个。LVS就是很典型的四层负载均衡。

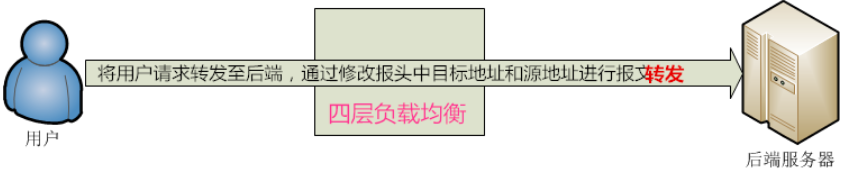


图2 四层负载均衡

七层负载均衡工作在OSI模型的第七层应用层，所以七层负载均衡可以基于请求的应用层信息进行负载均衡，例如根据请求的资源类型分配到后端服务器，而不再是根据IP和端口选择。七层负载均衡的功能更丰富更灵活，也能使整个网络更智能。如上图所示，在七层负载均衡两端(面向用户端和服务器端)的连接都是独立的。Nginx是很典型的四层负载均衡。

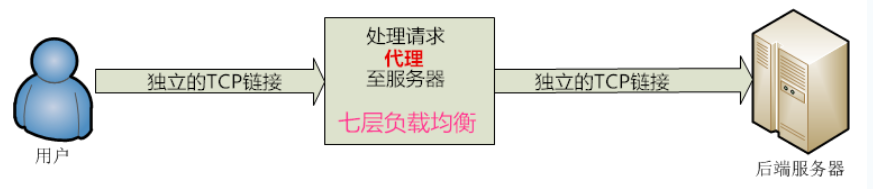


图3 七层负载均衡

## **5.2**应用场景

七层应用负载的好处，是使得整个网络更"智能化"。例如访问一个网站的用户流量，可以通过七层的方式，将对图片类的请求转发到特定的图片服务器并可以使用缓存技术；将对文字类的请求可以转发到特定的文字服务器并可以使用压缩技术。

当然这只是七层应用的一个小案例，从技术原理上，这种方式可以对客户端的请求和服务器的响应进行任意意义上的修改，极大的提升了应用系统在网络层的灵活性。很多在后台，例如Nginx或者Apache上部署的功能可以前移到负载均衡设备上，例如客户请求中的Header重写，服务器响应中的关键字过滤或者内容插入等功能。

另外一个常常被提到功能就是安全性。网络中最常见的SYN Flood攻击，即黑客控制众多源客户端，使用虚假IP地址对同一目标发送SYN攻击，通常这种攻击会大量发送SYN报文，耗尽服务器上的相关资源，以达到Denial of Service(DoS)的目的。

从技术原理上也可以看出，四层模式下这些SYN攻击都会被转发到后端的服务器上；而七层模式下这些SYN攻击自然在负载均衡设备上就截止，不会影响后台服务器的正常运营。另外负载均衡设备可以在七层层面设定多种策略，过滤特定报文，例如SQL Injection等应用层面的特定攻击手段，从应用层面进一步提高系统整体安全。

现在的7层负载均衡，主要还是着重于应用HTTP协议，所以其应用范围主要是众多的网站或者内部信息平台等基于B/S开发的系统。 4层负载均衡则对应其他TCP应用，例如IM即时通讯、实时消息推送等socket长连接系统。

**5.3** Nginx

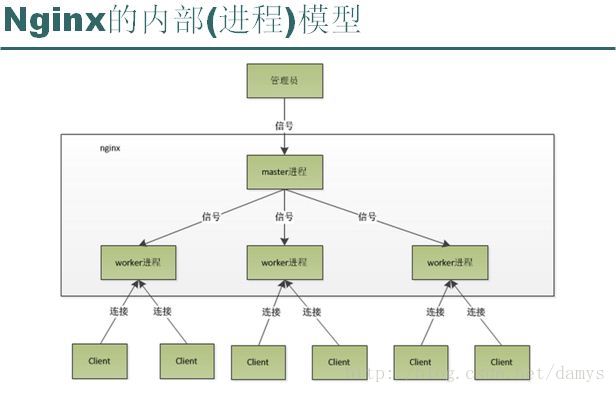
Nginx是一款轻量级的高性能web服务器，同时也是一款非常优秀的负载均衡器和反向代理服务器。由于支持强大的正则匹配规则、动静分离、URLrewrite功能及安装配置简单且对网络稳定性依赖非常小等优点，所以常用来做为七层负载均衡使用。在硬件不差的情况下，通常可以稳定支持几万的并发连接，在硬件性能足够好，且对系统内核参数及Nginx配置进行优化甚至可以达到10万以上的并发。

图4 Nginx的内部模型

  当nginx接到 一个http请求之后，会找通过查找配置文件，并在配置文件中找到相应的地址映射，该地址也叫location block，而location中配置的文件会启动相应的block完成地址分发工作；   而一个location中的指令会涉及到一个handle模块和多个filter模块；handler负责完成请求和响应的结果处理，而filter负责对响应的内容进行处理。

**5.4 Lvs**

LVS是Linux Virtual Server的简写，意即Linux虚拟服务器，是一个虚拟的服务器集群系统。常用来做为四层负载均衡使用。是中国国内最早出现的自由软件项目之一。

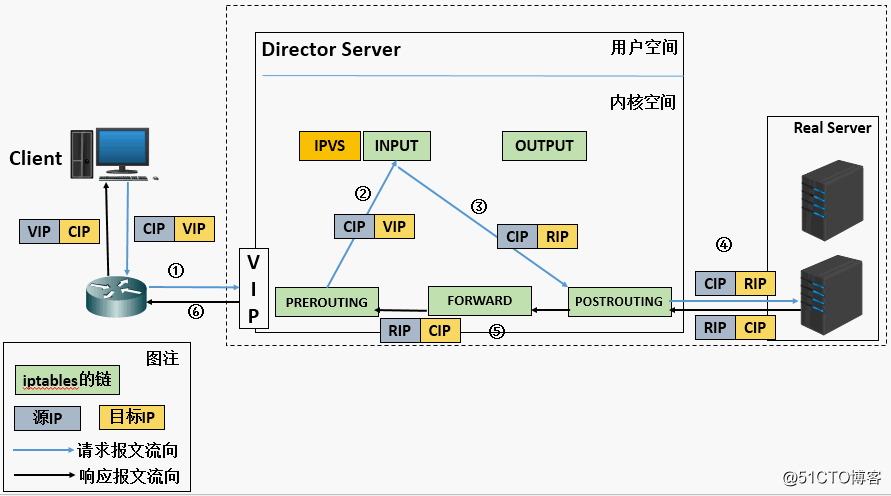


图5 lvs的原理

lvs工作原理：  
1、当用户向负载均衡调度器（Director Server）发起请求，调度器将请求发往至内核空间

2、PREROUTING链首先会接收到用户请求，判断目标IP确定是本机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已定义好的集群服务进行比对，如果用户请求的就是定义的集群服务，那么此时IPVS会强行修改数据包里的目标IP地址及端口，并将新的数据包发往POSTROUTING链

4、POSTROUTING链接收数据包后发现目标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

# 常用算法

6.1以下是Nginx作为七层负载均衡常用的几种调度算法和适用的业务场景

6.1.1 轮询（默认调度算法）

特点：每个请求按时间顺序逐一分配到不同的后端服务器处理。  
适用业务场景：后端服务器硬件性能配置完全一致，业务无特殊要求时使用。

upstream backendserver{

server 192.168.0.14：80

max\_fails=2 fail\_timeout=10s;

server 192.168.0.15：80

max\_fails=2 fail\_timeout=10s; }

6.1.2加权轮询

特点：指定轮询几率，weight值(权重)和访问比例成正比，用户请求按权重比例分配。  
适用业务场景：用于后端服务器硬件性处理能力不平均的情形。

upstream backendserver {

server 192.168.0.14:80

weight=5

max\_fails=2

fail\_timeout=10s;

server 192.168.0.15:80 weight=10

max\_fails=2

fail\_timeout=10s;

}

6.1.3、ip\_hash

特点：每个请求按访问ip的hash结果分配，这样每个访客固定访问一个后端服务器，可以解决session会话保持问题。  
 适用业务场景：适用于需要账号登录的系统，会话连接保持的业务。

upstream backendserver {

ip\_hash;

server 192.168.0.14:80

max\_fails=2 fail\_timeout=10s;

server 192.168.0.15:80

max\_fails=2

fail\_timeout=10s;

}

6.1.4 最少连接数 least\_conn

特点：按nginx反向代理与后端服务器之间的连接数，连接数最少的优先分配。适用业务场景：适用于客户端与后端服务器需要保持长连接的业务。

upstream backendserver {

least\_conn;

server 192.168.0.14:80

max\_fails=2

fail\_timeout=10s;

server 192.168.0.15:80

max\_fails=2

fail\_timeout=10s;

}

6.2以下是Lvs作为四层负载均衡常用的几种调度算法和适用的业务场景

6.2.1 轮叫调度（Round Robin Scheduling）

该算法是以轮叫的方式依次将请求调度不同的服务器，即每次调度执行i = (i + 1) mod n，并选出第i台服务器。算法的优点是其简洁性，它无需记录当前所有连接的状态，所以它是一种无状态调度。

对比分析j = i;

do {

j = (j + 1) mod n;

if (W(Sj) > 0) {

i = j;

return Si;

}

} while (j != i);

return NULL;

6.2.2加权轮叫调度

加权轮叫调度（Weighted Round-Robin Scheduling）算法可以解决服务器间性能不一的情况，它用相应的权值表示服务器的处理性能，服务器的缺省权值为1。假设服务器A的权值为1，B的 权值为2，则表示服务器B的处理性能是A的两倍。加权轮叫调度算法是按权值的高低和轮叫方式分配请求到各服务器。权值高的服务器先收到的连接，权值高的服 务器比权值低的服务器处理更多的连接，相同权值的服务器处理相同数目的连接数。

while (true) {

i = (i + 1) mod n;

if (i == 0) {

cw = cw - gcd(S);

if (cw <= 0) {

cw = max(S);

if (cw == 0)

return NULL;

}

}

if (W(Si) >= cw)

return Si;

}

6.2.3 最小连接调度

最小连接调度（Least-Connection Scheduling）算法是把新的连接请求分配到当前连接数最小的服务器。最小连接调度是一种动态调度算法，它通过服务器当前所活跃的连接数来估计服务 器的负载情况。调度器需要记录各个服务器已建立连接的数目，当一个请求被调度到某台服务器，其连接数加1；当连接中止或超时，其连接数减一。

for (m = 0; m < n; m++) {

if (W(Sm) > 0) {

for (i = m+1; i < n; i++) {

if (W(Si) <= 0)

continue;

if (C(Si) < C(Sm))

m = i;

}

return Sm;

}

}

return NULL;

6.2.4加权最小连接调度

加权最小连接调度（Weighted Least-Connection Scheduling）算法是最小连接调度的超集，各个服务器用相应的权值表示其处理性能。服务器的缺省权值为1，系统管理员可以动态地设置服务器的权 值。加权最小连接调度在调度新连接时尽可能使服务器的已建立连接数和其权值成比例。

for (m = 0; m < n; m++) {

if (W(Sm) > 0) {

for (i = m+1; i < n; i++) {

if (C(Sm)\*W(Si) > C(Si)\*W(Sm))

m = i;

}

return Sm;

}

}

return NULL;

# 对比分析

7.1在技术原理上

所谓四层负载均衡，也就是主要通过报文中的目标地址和端口，再加上负载均衡设备设置的服务器选择方式，决定最终选择的内部服务器。

以常见的TCP为例，负载均衡设备在接收到第一个来自客户端的SYN 请求时，即通过上述方式选择一个最佳的服务器，并对报文中目标IP地址进行修改(改为后端服务器IP），直接转发给该服务器。TCP的连接建立，即三次握手是客户端和服务器直接建立的，负载均衡设备只是起到一个类似路由器的转发动作。在某些部署情况下，为保证服务器回包可以正确返回给负载均衡设备，在转发报文的同时可能还会对报文原来的源地址进行修改。

所谓七层负载均衡，也称为“内容交换”，也就是主要通过报文中的真正有意义的应用层内容，再加上负载均衡设备设置的服务器选择方式，决定最终选择的内部服务器。

以常见的TCP为例，负载均衡设备如果要根据真正的应用层内容再选择服务器，只能先代理最终的服务器和客户端建立连接(三次握手)后，才可能接受到客户端发送的真正应用层内容的报文，然后再根据该报文中的特定字段，再加上负载均衡设备设置的服务器选择方式，决定最终选择的内部服务器。负载均衡设备在这种情况下，更类似于一个代理服务器。负载均衡和前端的客户端以及后端的服务器会分别建立TCP连接。所以从这个技术原理上来看，七层负载均衡明显的对负载均衡设备的要求更高，处理七层的能力也必然会低于四层模式的部署方式。

7.2 四层负载均衡和七层负载均衡技术的总体对比：

智能性：七层负载均衡由于具备OIS七层的所有功能，所以在处理用户需求上能更加灵活，从理论上讲，七层模型能对用户的所有跟服务端的请求进行修改。例如对文件header添加信息，根据不同的文件类型进行分类转发。四层模型仅支持基于网络层的需求转发，不能修改用户请求的内容。

安全性：七层负载均衡由于具有OSI模型的全部功能，能更容易抵御来自网络的攻击；四层模型从原理上讲，会直接将用户的请求转发给后端节点，无法直接抵御网络攻击。

复杂度：四层模型一般比较简单的架构，容易管理，容易定位问题；七层模型架构比较复杂，通常也需要考虑结合四层模型的混用情况，出现问题定位比较复杂。

效率比：四层模型基于更底层的设置，通常效率更高，但应用范围有限；七层模型需要更多的资源损耗，在理论上讲比四层模型有更强的功能，现在的实现更多是基于http应用。

# 参考文献

1. 何东泽,杨桂芹. [基于SDN的数据中心多路径负载均衡算法](https://kns.cnki.net/kcms/detail/detail.aspx?filename=LZTX201904007&dbcode=CJFQ&dbname=CJFD2019&v=gEo%mmd2F8%mmd2F6XAXPEnSDhAa%mmd2FKG%mmd2BkaEOX862ExRFO064nd02vh49OdHW46kwmn%mmd2FgQgquPP" \t "https://kns.cnki.net/kcms/detail/frame/kcmstarget)[J].  兰州交通大学学报. 2019(04)
2. 许红亮,杨桂芹,蒋占军.基于软件定义网络的数据中心自适应多路径负载均衡算法[J/OL].计算机应用:1-6[2021-01-09]
3. 陈淑平,周慧霖,何王全,漆锋滨.用于超大Infiniband网络的负载均衡多播路由[J/OL].计算机工程与应用:1-12[2021-01-09]
4. 叶晓涵,李德识.基于强化学习的水下传感网负载均衡路由算法[J].计算机工程与设计,2020,41(12):3301-3307.
5. 闫建新,仲桂芳.基于云平台虚拟化资源负载均衡的研究与实现[J].信息通信,2020(11):220-223.
6. 李凡,李涤,李伟岸.边缘覆盖网络存储负载均衡分配法研究与设计[J].计算机仿真,2020,37(11):384-387+464
7. 孙永源. 分布式雾计算中任务卸载与负载均衡方案研究[D].南京邮电大学,2020..