# 负载均衡技术及其运用

大数据1801 项岩

## 一、负载均衡

### 1.概述

负载均衡，英文名称为Load Balance，其含义就是指将负载（工作任务）进行平衡、分摊到多个操作单元上进行运行，例如FTP服务器、Web服务器、企业核心应用服务器和其它主要任务服务器等，从而协同完成工作任务。

负载均衡构建在原有网络结构之上，它提供了一种透明且廉价有效的方法扩展服务器和网络设备的带宽、加强网络数据处理能力、增加吞吐量、提高网络的可用性和灵活性。

### 2.分类

#### a、软/硬件负载均衡

[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6)负载均衡解决方案是指在一台或多台[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)相应的[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F)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](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)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](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F)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](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)和外部网络间安装负载均衡设备，这种设备通常称之为[负载均衡器](https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1%E5%99%A8)，由于专门的设备完成专门的任务，独立于[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F)，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](https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1%E5%99%A8)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](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)与Internet链接之间，有些则以两块[网络适配器](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E9%80%82%E9%85%8D%E5%99%A8" \t "_blank)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。一般而言，硬件负载均衡在功能、性能上优于[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6" \t "_blank)方式，不过成本昂贵。

#### b、本地/全局负载均衡

负载均衡从其应用的地理结构上分为本地负载均衡(Local Load Balance)和全局负载均衡(Global Load Balance，也叫地域负载均衡)，本地负载均衡针对本地范围的服务器群做负载均衡，全局负载均衡针对不同地理位置、不同网络结构的服务器群做负载均衡。

本地负载均衡不需要花费高额成本购置高性能服务器，只需利用现有设备资源,就可有效避免服务器单点故障造成数据流量的损失，通常用来解决数据流量过大、网络负荷过重的问题。同时它拥有形式多样的均衡策略把数据流量合理均衡的分配到各台服务器。如果需要在现在服务器上升级扩充，不需改变现有网络结构、停止现有服务，仅需要在服务群中简单地添加一台新服务器。

全局负载均衡主要解决全球用户只需一个域名或IP地址就能访问到离自己距离最近的服务器获得最快的访问速度，它在多区域都拥有自己的服务器站点，同时也适用于那些子公司站点分布广的大型公司通过企业内部网(Intranet)达到资源合理分配的需求。

全局负载均衡具备的特点：1、提高服务器响应速度，解决网络拥塞问题，达到高质量的网络访问效果。2、能够远距离为用户提供完全的透明服务,真正实现与地理位置无关性3、能够避免各种单点失效，既包括数据中心、服务器等的单点失效，也包括专线故障引起的单点失效。

### 3.部署方式

负载均衡有三种部署方式：路由模式、桥接模式、服务直接返回模式。路由模式部署灵活，约60%的用户采用这种方式部署；桥接模式不改变现有的网络架构；服务直接返回（DSR）比较适合吞吐量大特别是内容分发的网络应用。约30%的用户采用这种模式。

#### a、路由模式（推荐）

路由模式的部署方式，[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)的网关必须设置成负载均衡机的LAN口地址，且与WAN口分署不同的[逻辑网络](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E7%BD%91%E7%BB%9C" \t "_blank)。因此所有返回的流量也都经过负载均衡。这种方式对网络的改动小，能均衡任何下行流量。

#### b、桥接模式

桥接模式配置简单，不改变现有网络。负载均衡的WAN口和LAN口分别连接上行设备和下行[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "_blank)。LAN口不需要配置IP（WAN口与LAN口是桥连接），所有的[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "_blank)与负载均衡均在同一[逻辑网络](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E7%BD%91%E7%BB%9C)中。

由于这种安装方式[容错性](https://baike.baidu.com/item/%E5%AE%B9%E9%94%99%E6%80%A7" \t "_blank)差，网络架构缺乏弹性，对[广播风暴](https://baike.baidu.com/item/%E5%B9%BF%E6%92%AD%E9%A3%8E%E6%9A%B4)及其他[生成树协议](https://baike.baidu.com/item/%E7%94%9F%E6%88%90%E6%A0%91%E5%8D%8F%E8%AE%AE)循环相关联的错误敏感，因此一般不推荐这种安装架构。

#### c、服务直接返回模式

这种安装方式负载均衡的LAN口不使用，WAN口与[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "_blank)在同一个网络中，互联网的客户端访问负载均衡的虚IP（VIP），虚IP对应负载均衡机的WAN口，负载均衡根据策略将流量分发到服务器上，服务器直接响应客户端的请求。因此对于[客户端](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF)而言，响应他的IP不是负载均衡机的虚IP（VIP），而是[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "_blank)自身的IP地址。也就是说返回的流量是不经过负载均衡的。因此这种方式适用大流量高带宽要求的服务。

## 二、六种常用的web负载均衡技术

### 1. http重定向

当http代理（比如浏览器）向web服务器请求某个URL后，web服务器可以通过http响应头信息中的Location标记来返回一个新的URL。这意味着HTTP代理需要继续请求这个新的URL，完成自动跳转。

性能缺陷：1、吞吐率限制：主站点服务器的吞吐率平均分配到了被转移的服务器。现假设使用RR（Round Robin）调度策略，子服务器的最大吞吐率为1000reqs/s，那么主服务器的吞吐率要达到3000reqs/s才能完全发挥三台子服务器的作用，那么如果有100台子服务器，那么主服务器的吞吐率可想而知得有大？相反，如果主服务的最大吞吐率为6000reqs/s，那么平均分配到子服务器的吞吐率为2000reqs/s，而现子服务器的最大吞吐率为1000reqs/s，因此就得增加子服务器的数量，增加到6个才能满足。2、重定向访问深度不同：有的重定向一个静态页面，有的重定向相比复杂的动态页面，那么实际服务器的负载差异是不可预料的，而主站服务器却一无所知。因此整站使用重定向方法做负载均衡不太好。

我们需要权衡转移请求的开销和处理实际请求的开销，前者相对于后者越小，那么重定向的意义就越大，例如下载。你可以去很多镜像下载网站试下，会发现基本下载都使用了Location做了重定向。

### 2. DNS负载均衡

DNS 负责提供域名解析服务，当访问某个站点时，实际上首先需要通过该站点域名的DNS服务器来获取域名指向的IP地址，在这一过程中，DNS服务器完成了域名到IP地址的映射，同样，这样映射也可以是一对多的，这时候，DNS服务器便充当了负载均衡调度器，它就像http重定向转换策略一样，将用户的请求分散到多台服务器上，但是它的实现机制完全不同。

相比http重定向，基于DNS的负载均衡完全节省了所谓的主站点，或者说DNS服务器已经充当了主站点的职能。但不同的是，作为调度器，DNS服务器本身的性能几乎不用担心。因为DNS记录可以被用户浏览器或者互联网接入服务商的各级DNS服务器缓存，只有当缓存过期后才会重新向域名的DNS服务器请求解析。也说是DNS不存在http的吞吐率限制，理论上可以无限增加实际服务器的数量。

特性:1、可以根据用户IP来进行智能解析。DNS服务器可以在所有可用的A记录中寻找离用记最近的一台服务器。2、动态DNS：在每次IP地址变更时，及时更新DNS服务器。当然，因为缓存，一定的延迟不可避免。

### 3. 反向代理负载均衡

几乎所有主流的Web服务器都热衷于支持基于反向代理的负载均衡。它的核心工作就是转发HTTP请求。

相比前面的HTTP重定向和DNS解析，反向代理的调度器扮演的是用户和实际服务器中间人的角色：1、任何对于实际服务器的HTTP请求都必须经过调度器2、调度器必须等待实际服务器的HTTP响应，并将它反馈给用户（前两种方式不需要经过调度反馈，是实际服务器直接发送给用户）

特性：1、调度策略丰富。例如可以为不同的实际服务器设置不同的权重，以达到能者多劳的效果。2、对反向代理服务器的并发处理能力要求高，因为它工作在HTTP层面。3、反向代理服务器进行转发操作本身是需要一定开销的，比如创建线程、与后端服务器建立TCP连接、接收后端服务器返回的处理结果、分析HTTP头部信息、用户空间和内核空间的频繁切换等，虽然这部分时间并不长，但是当后端服务器处理请求的时间非常短时，转发的开销就显得尤为突出。例如请求静态文件，更适合使用前面介绍的基于DNS的负载均衡方式。4、反向代理服务器可以监控后端服务器，比如系统负载、响应时间、是否可用、TCP连接数、流量等，从而根据这些数据调整负载均衡的策略。5、反射代理服务器可以让用户在一次会话周期内的所有请求始终转发到一台特定的后端服务器上（粘滞会话），这样的好处一是保持session的本地访问，二是防止后端服务器的动态内存缓存的资源浪费。

### 4. IP负载均衡(LVS-NAT)

因为反向代理服务器工作在HTTP层，其本身的开销就已经严重制约了可扩展性，从而也限制了它的性能极限。

NAT服务器:它工作在传输层，它可以修改发送来的IP数据包，将数据包的目标地址修改为实际服务器地址。

从 Linux2.4内核开始，其内置的Neftilter模块在内核中维护着一些数据包过滤表，这些表包含了用于控制数据包过滤的规则。可喜的是，Linux提供了iptables来对过滤表进行插入、修改和删除等操作。更加令人振奋的是，Linux2.6.x内核中内置了IPVS模块，它的工作性质类型于Netfilter模块，不过它更专注于实现IP负载均衡。

实验证明使用基于NAT的负载均衡系统。作为调度器的NAT服务器可以将吞吐率提升到一个新的高度，几乎是反向代理服务器的两倍以上，这大多归功于在内核中进行请求转发的较低开销。但是一旦请求的内容过大时，不论是基于反向代理还是NAT，负载均衡的整体吞吐量都差距不大，这说明对于一睦开销较大的内容，使用简单的反向代理来搭建负载均衡系统是值靠虑的。

这么强大的系统还是有它的瓶颈，那就是NAT服务器的网络带宽，包括内部网络和外部网络。当然如果你不差钱，可以去花钱去购买千兆交换机或万兆交换机，甚至负载均衡硬件设备，但如果你是个屌丝，咋办？

一个简单有效的办法就是将基于NAT的集群和前面的DNS混合使用，比如５个100Mbps出口宽带的集群，然后通过DNS来将用户请求均衡地指向这些集群，同时，你还可以利用DNS智能解析实现地域就近访问。这样的配置对于大多数业务是足够了，但是对于提供下载或视频等服务的大规模站点，NAT服务器还是不够出色。

### 5. 直接路由(LVS-DR)

NAT是工作在网络分层模型的传输层（第四层），而直接路由是工作在数据链路层（第二层），貌似更屌些。它通过修改数据包的目标MAC地址（没有修改目标IP），将数据包转发到实际服务器上，不同的是，实际服务器的响应数据包将直接发送给客户羰，而不经过调度器。

LVS-DR 相较于LVS-NAT的最大优势在于LVS-DR不受调度器宽带的限制，例如假设三台服务器在WAN交换机出口宽带都限制为10Mbps，只要对于连接调度器和两台实际服务器的LAN交换机没有限速，那么，使用LVS-DR理论上可以达到20Mbps的最大出口宽带，因为它的实际服务器的响应数据包可以不经过调度器而直接发往用户端啊，所以它与调度器的出口宽带没有关系，只能自身的有关系。而如果使用LVS-NAT，集群只能最大使用10Mbps的宽带。所以，越是响应数据包远远超过请求数据包的服务，就越应该降低调度器转移请求的开销，也就越能提高整体的扩展能力，最终也就越依赖于WAN出口宽带。

总的来说，LVS-DR适合搭建可扩展的负载均衡系统，不论是Web服务器还是文件服务器，以及视频服务器，它都拥有出色的性能。前提是你必须为实际器购买一系列的合法IP地址。

### 6. IP隧道(LVS-TUN)

基于IP隧道的请求转发机制：将调度器收到的IP数据包封装在一个新的IP数据包中，转交给实际服务器，然后实际服务器的响应数据包可以直接到达用户端。目前Linux大多支持，可以用LVS来实现，称为LVS-TUN，与LVS-DR不同的是，实际服务器可以和调度器不在同一个WANt网段，调度器通过 IP隧道技术来转发请求到实际服务器，所以实际服务器也必须拥有合法的IP地址。

总体来说，LVS-DR和LVS-TUN都适合响应和请求不对称的Web服务器，如何从它们中做出选择，取决于你的网络部署需要，因为LVS-TUN可以将实际服务器根据需要部署在不同的地域，并且根据就近访问的原则来转移请求，所以有类似这种需求的，就应该选择LVS-TUN。

## 三、负载均衡算法

### 1.静态均衡算法

#### a、轮询法

将请求按顺序轮流地分配到每个节点上，不关心每个节点实际的连接数和当前的系统负载。

优点：简单高效，易于水平扩展，每个节点满足字面意义上的均衡；

缺点：没有考虑机器的性能问题，根据木桶最短木板理论，集群性能瓶颈更多的会受性能差的服务器影响。

#### b、随机法

将请求随机分配到各个节点。由概率统计理论得知，随着客户端调用服务端的次数增多，其实际效果越来越接近于平均分配，也就是轮询的结果。优缺点和轮询相似。

#### c、源地址哈希法

 源地址哈希的思想是根据客户端的IP地址，通过哈希函数计算得到一个数值，用该数值对服务器节点数进行取模，得到的结果便是要访问节点序号。采用源地址哈希法进行负载均衡，同一IP地址的客户端，当后端服务器列表不变时，它每次都会落到到同一台服务器进行访问。

优点：相同的IP每次落在同一个节点，可以人为干预客户端请求方向，例如灰度发布；

缺点：如果某个节点出现故障，会导致这个节点上的客户端无法使用，无法保证高可用。当某一用户成为热点用户，那么会有巨大的流量涌向这个节点，导致冷热分布不均衡，无法有效利用起集群的性能。所以当热点事件出现时，一般会将源地址哈希法切换成轮询法。

#### d、加权轮询法

不同的后端服务器可能机器的配置和当前系统的负载并不相同，因此它们的抗压能力也不相同。给配置高、负载低的机器配置更高的权重，让其处理更多的请；而配置低、负载高的机器，给其分配较低的权重，降低其系统负载，加权轮询能很好地处理这一问题，并将请求顺序且按照权重分配到后端。

加权轮询算法要生成一个服务器序列，该序列中包含n个服务器。n是所有服务器的权重之和。在该序列中，每个服务器的出现的次数，等于其权重值。并且，生成的序列中，服务器的分布应该尽可能的均匀。比如序列{a, a, a, a, a, b, c}中，前五个请求都会分配给服务器a，这就是一种不均匀的分配方法，更好的序列应该是：{a, a, b, a, c, a, a}。

优点：可以将不同机器的性能问题纳入到考量范围，集群性能最优最大化；

缺点：生产环境复杂多变，服务器抗压能力也无法精确估算，静态算法导致无法实时动态调整节点权重，只能粗糙优化。

#### e、加权随机法

与加权轮询法一样，加权随机法也根据后端机器的配置，系统的负载分配不同的权重。不同的是，它是按照权重随机请求后端服务器，而非顺序。

#### f、键值范围法

根据键的范围进行负债，比如0到10万的用户请求走第一个节点服务器，10万到20万的用户请求走第二个节点服务器……以此类推。

优点：容易水平扩展，随着用户量增加，可以增加节点而不影响旧数据；

缺点：容易负债不均衡，比如新注册的用户活跃度高，旧用户活跃度低，那么压力就全在新增的服务节点上，旧服务节点性能浪费。而且也容易单点故障，无法满足高可用。

### 2.动态均衡算法

#### a、最小连接数法

根据每个节点当前的连接情况，动态地选取其中当前积压连接数最少的一个节点处理当前请求，尽可能地提高后端服务的利用效率，将请求合理地分流到每一台服务器。俗称闲的人不能闲着，大家一起动起来。

优点：动态，根据节点状况实时变化；

缺点：提高了复杂度，每次连接断开需要进行计数；

实现：将连接数的倒数当权重值。

#### b、最快响应速度法

根据请求的响应时间，来动态调整每个节点的权重，将响应速度快的服务节点分配更多的请求，响应速度慢的服务节点分配更少的请求，俗称能者多劳，扶贫救弱。

优点：动态，实时变化，控制的粒度更细，跟灵敏；

缺点：复杂度更高，每次需要计算请求的响应速度；

实现：可以根据响应时间进行打分，计算权重。

#### c、观察模式法

观察者模式是综合了最小连接数和最快响应度，同时考量这两个指标数，进行一个权重的分配。

## 四、参考文献

《六种常用的web负载均衡技术》——bugyun

《负载均衡》——百度百科

《负载均衡算法及分类》——[Linias](https://xianshang.blog.csdn.net/)