



链滴

# Nginx、LVS 及 HAProxy 负载均衡软件的优缺点详解

作者: [gentoo666](#)

原文链接: <https://ld246.com/article/1513271675009>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

**摘要:** Nginx/LVS/HAProxy 是目前使用最广泛的三种负载均衡软件，一般对负载均衡的使用是随着网站规模的提升根据不同的阶段来使用不同的技术，具体的应用需求还得具体分析，本文总结三者之间的优缺点。

**编者按** 负载均衡 (Load Balancing) 建立在现有网络结构之上，它提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力，同时能够提高网络的灵活性和可用性。目前使用最为广泛的负载均衡软件是 Nginx、LVS、HAProxy，本文作者结合自己的实践经验总结了三者各自的优缺点。文章来自 ha97 网。

**以下为原文:**

Nginx/LVS/HAProxy 是目前使用最广泛的三种负载均衡软件，本人都在多个项目中实施过，参了一些资料，结合自己的一些使用经验，总结一下。

一般对负载均衡的使用是随着网站规模的提升根据不同的阶段来使用不同的技术。具体的应用需求还得具体分析，如果是中小型的 Web 应用，比如日 PV 小于 1000 万，用 Nginx 就完全可以了；如机器不少，可以用 DNS 轮询，LVS 所耗费的机器还是比较多的；大型网站或重要的服务，且服务器较多时，可以考虑用 LVS。

一种是通过硬件来进行，常见的硬件有比较昂贵的 F5 和 Array 等商用的负载均衡器，它的优点是有专业的维护团队来对这些服务进行维护、缺点就是花销太大，所以对于规模较小的网络服务来说时还没有需要使用；另外一种就是类似于 Nginx/LVS/HAProxy 的基于 Linux 的开源免费的负载均衡件，这些都是通过软件级别来实现，所以费用非常低廉。

目前关于网站架构一般比较合理流行的架构方案：Web 前端采用 Nginx/HAProxy+ Keepalived 作负载均衡器；后端采用 MySQL 数据库一主多从和读写分离，采用 LVS+Keepalived 的架构。当然根据项目具体需求制定方案。

下面说说各自的特点和适用场合。

**Nginx 的优点是:**

1. 工作在网络的 7 层之上，可以针对 http 应用做一些分流的策略，比如针对域名、目录结构，的正则规则比 HAProxy 更为强大和灵活，这也是它目前广泛流行的主要原因之一，Nginx 单凭这点利用的场合就远多于 LVS 了。

- 
-

Nginx 对网络稳定性的依赖非常小，理论上能 ping 通就能进行负载功能，这个也是它的优势一；相反 LVS 对网络稳定性依赖比较大，这点本人深有体会；

- 
- 
-

3. Nginx 安装和配置比较简单，测试起来比较方便，它基本能把错误用日志打印出来。LVS 的置、测试就要花比较长的时间了，LVS 对网络依赖比较大。

4. 可以承担高负载压力且稳定，在硬件不差的情况下一般能支撑几万次的并发量，负载度比 LVS 相对小些。

5. Nginx 可以通过端口检测到服务器内部的故障，比如根据服务器处理网页返回的状态码、超等等，并且会把返回错误的请求重新提交到另一个节点，不过其中缺点就是不支持 url 来检测。比如户正在上传一个文件，而处理该上传的节点刚好在上传过程中出现故障，Nginx 会把上传切到另一台务器重新处理，而 LVS 就直接断掉了，如果是上传一个很大的文件或者很重要的文件的话，用户可能因此而不满。

6. Nginx 不仅仅是一款优秀的负载均衡器/反向代理软件，它同时也是功能强大的 Web 应用服务器。LNMP 也是近几年非常流行的 web 架构，在高流量的环境中稳定性也很好。

7. Nginx 现在作为 Web 反向加速缓存越来越成熟了，速度比传统的 Squid 服务器更快，可以考虑用其作为反向代理加速器。

8. Nginx 可作为中层反向代理使用，这一层面 Nginx 基本上无对手，唯一可以对比 Nginx 的就有 lighttpd 了，不过 lighttpd 目前还没有做到 Nginx 完全的功能，配置也不那么清晰易读，社区资也远远没 Nginx 活跃。

9. Nginx 也可作为静态网页和图片服务器，这方面的性能也无对手。还有 Nginx 社区非常活跃第三方模块也很多。

**Nginx 的缺点是:**

1. Nginx 仅能支持 http、https 和 Email 协议，这样就在适用范围上面小些，这个是它的缺点。

2. 对后端服务器的健康检查，只支持通过端口来检测，不支持通过 url 来检测。不支持 Session 直接保持，但能通过 ip\_hash 来解决。

LVS: 使用 Linux 内核集群实现一个高性能、高可用的负载均衡服务器，它具有很好的可伸缩性

Scalability)、可靠性 (Reliability)和可管理性 (Manageability)。

**LVS 的优点是：**

1. 抗负载能力强、是工作在网络 4 层之上仅作分发之用，**没有流量的产生**这个特点也决定了它在负载均衡软件里的性能最强的，对内存和 cpu 资源消耗比较低。
2. 配置性比较低，这是一个缺点也是一个优点，因为没有可太多配置的东西，所以并不需要太多接触大大减少了人为出错的几率。
3. 工作稳定，因为其本身抗负载能力很强，自身有完整的双机热备方案，如 LVS+Keepalived，不过在项目实施中用得最多的还是 LVS/DR+Keepalived。
4. 无流量，LVS 只分发请求，而流量并不从它本身出去，这点保证了均衡器 IO 的性能不会受到大流的影响。
5. 应用范围比较广，因为 LVS 工作在 4 层，所以它几乎可以对所有应用做负载均衡，包括 http、数据库、在线聊天室等等。

**LVS 的缺点是：**

1. 软件本身不支持正则表达式处理，不能做动静分离；而现在许多网站在这方面都有较强的需求这个是 Nginx/HAProxy+Keepalived 的优势所在。
2. 如果是网站应用比较庞大的话，LVS/DR+Keepalived 实施起来就比较复杂了，特别后面有 Windows Server 的机器的话，如果实施及配置还有维护过程就比较复杂了，相对而言，Nginx/HAProxy+Keepalived 就简单多了。

**HAProxy 的特点是：**

1. HAProxy 也是支持虚拟主机的。
2. HAProxy 的优点能够补充 Nginx 的一些缺点，比如**支持 Session 的保持，Cookie 的引**；同时支持通过获取指定的 url 来检测后端服务器的状态。
3. HAProxy 跟 LVS 类似，本身就只是一款负载均衡软件；单纯从效率上来讲 HAProxy 会比 Nginx 更出色的负载均衡速度，在并发处理上也是优于 Nginx 的。
4. HAProxy **支持 TCP 协议的负载均衡转发**，可以对 MySQL 读进行负载均衡对后端的 MySQL 节点进行检测和负载均衡，大家可以用 LVS+Keepalived 对 MySQL 主从做负载均衡。
5. HAProxy 负载均衡策略非常多，HAProxy 的负载均衡算法现在具体有如下 8 种：

**① roundrobin**，表示简单的轮询，这个不多说，这个是负载均衡基本都具备的；

**② static-rr**，表示根据权重，建议关注；

**③ leastconn**，表示最少连接者先处理，建议关注；

**④ source**，表示根据请求源 IP，这个跟 Nginx 的 IP\_hash 机制类似，我们用其作为解决 session 问

的一种方法，建议关注；

**⑤ ri**，表示根据请求的 URI；

**⑥ rl\_param**，表示根据请求的 URI 参数 'balance url\_param' requires an URL parameter nam

；

**⑦ hdr(name)**，表示根据 HTTP 请求头来锁定每一次 HTTP 请求；

**⑧ rdp-cookie(name)**，表示根据 cookie(name)来锁定并哈希每一次 TCP 请求。

**⑨ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑩ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑪ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑫ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑬ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑭ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑮ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑯ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑰ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑱ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑲ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**⑳ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**㉑ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**㉒ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**㉓ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**㉔ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**㉕ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**㉖ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**㉗ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**㉘ ip\_hash**，表示根据 IP 地址来锁定每一次 TCP 请求。

**Nginx 和 LVS 对比的总结：**

1. Nginx 工作在网络的 7 层，所以它可以针对 http 应用本身来做分流策略，比如针对域名、目结构等，相比之下 LVS 并不具备这样的功能，所以 Nginx 单凭这点可利用的场合就远多于 LVS 了；Nginx 有用的这些功能使其可调整度要高于 LVS，所以经常要去触碰触碰，触碰多了，人为出问题几率也就大。

2. Nginx 对网络稳定性的依赖较小，理论上只要 ping 得通，网页访问正常，Nginx 就能连得通这是 Nginx 的一大优势！Nginx 同时还能区分内外网，如果是同时拥有内外网的节点，就相当于单拥有了备份线路；LVS 就比较依赖于网络环境，目前来看服务器在同一网段内并且 LVS 使用 direct 式分流，效果较能得到保证。另外注意，LVS 需要向托管商至少申请多一个 ip 来做 Visual IP，貌似不能用本身的 IP 来做 VIP 的。要做好 LVS 管理员，确实得跟进学习很多有关网络通信方面的知识，不再是一个 HTTP 那么简单了。

3. Nginx 安装和配置比较简单，测试起来也很方便，因为它基本能把错误用日志打印出来。LVS 的安装和配置、测试就要花比较长的时间了；LVS 对网络依赖比较大，很多时候不能配置成功都是因网络问题而不是配置问题，出了问题要解决也相应的会麻烦得多。

<p>4. Nginx 也同样能承受很高负载且稳定，但负载度和稳定度差 LVS 还有几个等级：Nginx 处理有流量所以受限于机器 IO 和配置；本身的 bug 也还是难以避免的。</p>

<p>5. Nginx 可以检测到服务器内部的故障，比如根据服务器处理网页返回的状态码、超时等等，且会把返回错误的请求重新提交到另一个节点。目前 LVS 中 ldirectd 也能支持针对服务器内部的情况来监控，但 LVS 的原理使其不能重发请求。比如用户正在上传一个文件，而处理该上传的节点刚好在传过程中出现故障，Nginx 会把上传切到另一台服务器重新处理，而 LVS 就直接断掉了，如果是上一个很大的文件或者很重要的文件的话，用户可能会因此而恼火。</p>

<p>6. Nginx 对请求的异步处理可以帮助节点服务器减轻负载，假如使用 apache 直接对外服务，么出现很多的窄带链接时 apache 服务器将会占用大量内存而不能释放，使用多一个 Nginx 做 apache 代理的话，这些窄带链接会被 Nginx 挡住，apache 上就不会堆积过多的请求，这样就减少了相当的资源占用。这点使用 squid 也有相同的作用，即使 squid 本身配置为不缓存，对 apache 还是有很大帮助的。</p>

<p>7. Nginx 能支持 http、https 和 email (email 的功能比较少用)，LVS 所支持的应用在这点上比 Nginx 更多。在使用上，一般最前端所采取的策略应是 LVS，也就是 DNS 的指向应为 LVS 均衡，LVS 的优点令它非常适合做这个任务。重要的 ip 地址，最好交由 LVS 托管，比如数据库的 ip、webservice 服务器的 ip 等等，这些 ip 地址随着时间推移，使用面会越来越广，如果更换 ip 则故障会接而至。所以将这些重要 ip 交给 LVS 托管是最为稳妥的，这样做的唯一缺点是需要的 VIP 数量会比较。Nginx 可作为 LVS 节点机器使用，一是可以利用 Nginx 的功能，二是可以利用 Nginx 的性能。当这一层面也可以直接使用 squid，squid 的功能方面就比 Nginx 弱不少了，性能上也有所逊色于 Nginx。Nginx 也可作为中层代理使用，这一层面 Nginx 基本上无对手，唯一可以撼动 Nginx 的就只有 lighttpd 了，不过 lighttpd 目前还没有能做到 Nginx 完全的功能，配置也不那么清晰易读。另外，中代理的 IP 也是重要的，所以中层代理也拥有一个 VIP 和 LVS 是最完美的方案了。具体的应用还得具分析，如果是比较小的网站 (日 PV 小于 1000 万)，用 Nginx 就完全可以了，如果机器也不少，可用 DNS 轮询，LVS 所耗费的机器还是比较多的；大型网站或者重要的服务，机器不发愁的时候，要多考虑利用 LVS。</p>

<p><strong>现在对网络负载均衡的使用是随着网站规模的提升根据不同的阶段来使用不同的技术</strong></p>

<p>第一阶段：利用 Nginx 或 HAProxy 进行单点的负载均衡，这一阶段服务器规模刚脱离单服务、单数据库的模式，需要一定的负载均衡，但是仍然规模较小没有专业的维护团队来进行维护，也没有需要进行大规模的网站部署。这样利用 Nginx 或 HAProxy 就是第一选择，此时这些东西上手快，置容易，在七层之上利用 HTTP 协议就可以。这时是第一选择。</p>

<p>第二阶段：随着网络服务进一步扩大，这时单点的 Nginx 已经不能满足，这时使用 LVS 或者商用 Array 就是首要选择，Nginx 此时就作为 LVS 或者 Array 的节点来使用，具体 LVS 或 Array 的是选是根据公司规模和预算来选择，Array 的应用交付功能非常强大，本人在某项目中使用过，性价比也高于 F5，商用首选，但是一般来说这阶段相关人才跟不上业务的提升，所以购买商业负载均衡已经为了必经之路。</p>

<p>第三阶段：这时网络服务已经成为主流产品，此时随着公司知名度也进一步扩展，相关人才的能力以及数量也随之提升，这时无论从开发适合自身产品的定制，以及降低成本来讲开源的 LVS，已经成首选，这时 LVS 会成为主流。</p>

<p>最终形成比较理想的基本架构为：Array/LVS — Nginx/Haproxy — Squid/Varnish — AppServer。</p>