# Nginx、HAProxy、LVS三者的优缺点

**对于做软件负载均衡，我们都知道主流的方案有LVS、Haproxy、Nginx。**

**那么我们该如何选择呢？**

### HAProxy、Nginx、LVS、Apache比较：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **比较** | **HAProxy** | **Nginx** | **LVS** | **Apache** |
| 简介 | 高可用、负载均衡且基于**TCP和HTTP**应用的代理，支持高并发，多集群反代。 | 高性能**http和反向代理服务器、邮件代理服务器，支持高并发，轻量级Web，低系统资源消耗。** | Linux虚拟服务器，常用VS/NAT、VS/TUN和VS/DR，三种模式负载均衡。 | 高性能Web服务器，支持代理，市场份额很高。 |
| 优点缺点 | 1、抗负载能力强，负载均衡速度高。  2、**支持session保持，Cookie引导，可通过url检测后端服务器健康状态**。  3、也可做MySQL、Email等负载均衡。  4、一般不做Web服务器的Cache。 | 1、抗负载能力强。  2、http、https、Emai协议功能较好，处理相应请求快。  3、Web能力强，配置简单，**支持缓存功能、适用动静分离**，低内存消耗。  4、**不支持session直接保持，但可通过ip\_hash解决，通过端口对后端服务器健康检查**。 | 1、抗负载能力强。  2、**通过vrrp转发（仅分发）效率高**，流量通过内核处理，没有流量产生。（理论）  3、相当稳定可靠。  4、**不支持正则，不能做动静分离，配置略复杂，需要IP略多**。 | 1、Web处理能力强，市场份额很高。（不过后期Nginx在Web方面越来越好，份额也在增长）  2、Rewrite强大，并支持很多模块，扩展很方便。  3、Bug少，历史久，比较稳定。  4、处理动态请求比Nginx好。 |
| 支持算法 | 1、轮循  Round-robin  2、带权轮循  Weight-round-robin  3、原地址保持）  Source  4、RI请求URL  rdp-cookie（根据cookie） | 1、轮循  Round-robin  2、带权轮循  Weight-round-robn  3、Ip哈希  Ip-hash | 1、rr（轮循）  2、wrr（带权轮循）  3、lc（最小连接）、  4、wlc（权重最小连接） | 通过相关模块实现代理 |
| 官网 | www.haproxy.com | nginx.org | www.linuxvirtualserver.org | http://www.apache.org/ |
| 是否免费 | 免费 | 免费 | 免费 | 免费 |
| 虚拟主机 | 支持 | 支持 | 不支持 | 支持 |
| 适用性 | 七层（常用）、四层（不常用） | 七层（常用） | 四层（常用） | 不常用 |
| 量级 | 七层重量级，四层轻量级 | 七层重量级，四层轻量级 | 四层重量级 | 功能略差 |
| 常用热备 | Keepalived+其它 | Keepalived+其它 | Keepalived+其它 | Keepalived+其它 |
| 负载均衡参考标准 | 1、效率，并发处理能力、处理数据能力。  2、成本、需求、平台。 | | | |

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；

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。

### 一、Nginx特点

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，它的正则规则比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单凭这点可利用的场合就远多于LVS了。
2. **Nginx对网络稳定性的依赖非常小**，理论上能ping通就就能进行负载功能。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。

Nginx同时还能区分内外网，如果是同时拥有内外网的节点，就相当于单机拥有了备份线路；

**LVS就比较依赖于网络环境，**很多时候不能配置成功都是因为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。目前来看服务器在同一网段内并且LVS使用direct方式分流，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，貌似是不能用本身的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，就不再是一个HTTP那么简单了。

1. **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的配置、测试就要花比较长的时间了，LVS对网络依赖比较大。
2. **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、超时等等，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**。比如用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服务器重新处理，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，用户可能会因此而不满。

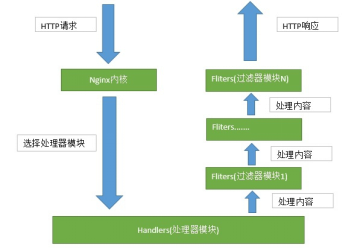
6、**Nginx也是功能强大的Web应用服务器既Cache功能（静态网页和图片服务器）。**LNMP也是近几年非常流行的web架构，在高流量的环境中稳定性也很好。**Nginx现在作为Web反向加速缓存越来越成熟**了，速度比传统的Squid服务器更快，可以考虑用其作为反向代理加速器。

1. **Nginx可作为反向代理服务器使用，这一层面Nginx基本上无对手**，唯一可以对比Nginx的就只有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，社区资料也远远没Nginx活跃。

 什么是nginx的异步处理：  
        squid同步处理：浏览器发起请求，而后请求会立刻被转到后端，于是在浏览器和后台之间就建立了一个通道。从请求发起直到请求完成，这条通道都是一直存在的。  
        **nginx异步处理：浏览器发起请求，请求不会立刻转到后端，而是请求数据(header)先收到nignx上，然后nginx再把这个请求发到后端，后端处理完成后把数据返回到nginx上，nginx将数据流发到浏览器。**  
    使用异步处理的好处：  
        1. 假设用户执行一个上传文件操作，因为用户网速又比较慢，因此需要花半个小时才能把文件传到服务器。squid的同步代理在用户开始上传后就和后台建立了连接，半小时后文件上传结束，由此可见，后台服务器连接保持了半个小时；而nginx异步代理就是先将此文件收到nginx上，因此仅仅是nginx和用户保持了半小时连接，后台服务器在这半小时内没有为这个请求开启连接，半小时后用户上传结束，nginx才将上传内容发到后台，nginx和后台之间的带宽是很充裕的，所以只花了一秒钟就将请求发送到了后台，由此可见，后台服务器连接保持了一秒。同步传输花了后台服务器半个小时，异步传输只花一秒，可见优化程度很大。  
        2. 在上面这个例子中，假如后台服务器因为种种原因重启了，上传文件就自然中断了，这对用户来说是非常恼火的一件事情，想必各位也有上传文件传到一半被中断的经历。用nginx代理之后，后台服务器的重启对用户上传的影响减少到了极点，而nginx是非常稳定的并不需要常去重启它，即使需要重启，利用kill -HUP就可以做到不间断重启nginx。  
        3. 异步传输可以令负载均衡器更有保障，为什么这么说呢？在其它的均衡器（lvs/haproxy/apache等）里，每个请求都是只有一次机会的，**假如用户发起一个请求，结果该请求分到的后台服务器刚好挂掉了，那么这个请求就失败了；而nginx因为是异步的，所以这个请求可以重新发往下一个后台，下一个后台返回了正常的数据，于是这个请求就能成功了。还是用用户上传文件这个例子，假如不但用了nginx代理，而且用了负载均衡，nginx把上传文件发往其中一台后台，但这台服务器突然重启了，nginx收到错误后，会将这个上传文件发到另一台后台，于是用户就不用再花半小时上传一遍。**        4. 假如用户上传一个10GB大小的文件，而后台服务器没有考虑到这个情况，那么后台服务器岂不要崩溃了。用nginx就可以把这些东西都拦在nginx上，通过nginx的上传文件大小限制功能来限制，另外nginx性能非常有保障，就放心的让互联网上那些另类的用户和nginx对抗去吧。  
        用异步传输会造成问题：  
            后台服务器有提供上传进度的功能的话，用了nginx代理就无法取得进度，这个需要使用nginx的一个第三方模块来实现。  
 **Nginx upstream支持的分配策略及原理：  
        1. 轮询(默认)：每个请求按照顺序逐一分配到不同的后端服务器。如后端服务器down掉，就切换到另一台并剔除down的后端主机  
        2. weight：指定轮询几率（权重），weight和访问比率成正比，用于后端服务器性能不均的情况。  
        3. ip\_hash：每个请求按照访问ip的hash结果分配，不同ip的请求被分配到后端不同的服务器上，可以解决session的问题。**

淘宝的前端使用的Tengine就是基于nginx做的二次开发定制版。

Nginx常规的HTTP请求和响应流程图：



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，算是它的缺点。

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的直接保持，但能通过ip\_hash来解决。**

### 二、HAProxy特点

HAProxy免费开源，稳定性也是非常好。

**经过官方测试统计，haproxy 单位时间处理的最大请求数为20000个，可以同时维护40000-50000个并发连接，最大数据处理能力为10Gbps，**这个数值作为软件级负载均衡器是相当惊人的。 综上述，haproxy是性能优越的负载均衡、反向代理服务器。

**不能做Web服务器即Cache。**

**HAProxy的特点是：**

1、 **支持两种代理模式：TCP（四层）和HTTP（七层），支持虚拟主机；**

2、**HAProxy的优点能够补充Nginx的一些缺点，比如支持Session的保持，Cookie的引导；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有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）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支持连接拒绝，因为维护一个连接的打开的开销是很低的，有时我们很需要限制攻击蠕虫（attack bots），也就是说限制它们的连接打开从而限制它们的危害。

这个已经为一个陷于小型DDoS攻击的网站开发了，而且已经拯救了很多站点，这个优点也是其它负载均衡器没有的。

6、haproxy支持全透明代理（已具备硬件防火墙的典型特点）:

可以用客户端IP地址或者任何其他地址来连接后端服务器。

这个特性仅在Linux 2.4/2.6内核打了tcp proxy补丁后才可以使用。

这个特性也使得为某特殊服务器处理部分流量同时又不修改服务器的地址成为可能。

7、自带强大的监控服务器状态的页面，实际环境中我们结合Nagios进行邮件或短信报警。

8、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问题的一种方法，建议关注；

⑤ ri，表示根据请求的URI；

⑥ rl\_param，表示根据请求的URl参数’balance url\_param’ requires an URL parameter name；

⑦ hdr(name)，表示根据HTTP请求头来锁定每一次HTTP请求；

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

### 三、LVS（linux virtual server）特点

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)。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处理所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。

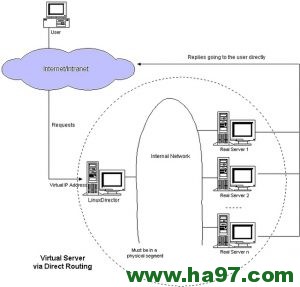
  **LVS的优点是：**

1. 1、抗负载能力强、性能高，能达到F5的60%，对内存和CPU资源消耗比较低，**工作在网络4层**，通过VRRP协议（**仅作代理分发之用**），具体的流量是由linux内核来处理，因此没有流量的产生，而流量并不从它本身出去，这点保证了均衡器IO的性能不会收到大流量的影响。对内存和cpu资源消耗比较低。
2. 配置性比较低（配置复杂），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（**LVS+Keepalived**），不过我们在项目实施中用得最多的还是LVS/DR+Keepalived。
4. 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。

6、支持多种负载均衡算法：rr(轮询)，wrr(带权轮询)、lc(最小连接)、wlc(带权最小连接)

7、LVS工作模式有4种：  
        (1) nat 地址转换  
        (2) dr 直接路由  
        (3) tun 隧道  
        (4) full-nat

LVS DR(Direct Routing)模式的网络流程图：



**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。

### 四、总结

**衡量负载均衡器好坏的几个重要的因素：**  
    1. 会话率 ：单位时间内的处理的请求数  
    2. 会话并发能力：并发处理能力  
    3. 数据率：处理数据能力

### 三大主流软件负载均衡器适用业务场景：

**1. 网站建设初期，可以选用Nginx、HAProxy作为反向代理负载均衡(流量不大时，可以不选用负载均衡)，因为其配置简单，性能也能满足一般业务场景。**

**如果考虑到负载均衡器是有单点问题，可以采用Nginx+Keepalived/HAproxy+Keepalived避免负载均衡器自身的单点问题。**

**2. 网站并发到达一定程度后，为了提高稳定性和转发效率，可以使用lvs，毕竟lvs比Nginx/HAProxy要更稳定，转发效率也更高。**

#### Nginx与HAProxy对比：

**nginx只支持七层，用户量最大，稳定性比较可靠。**

**Haproxy支持四层和七层，支持更多的负载均衡算法，支持session保持等。**

#### Nginx和LVS对比：

1. 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，那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。

而使用多一个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，这样就减少了相当多的资源占用。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。

**2、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均衡器，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，使用面会越来越大，如果更换ip则故障会接踵而至。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数量会比较多。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。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，可以用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，要多多考虑利用LVS。**