**nginx\_haproxy\_LVS三大软件介绍**

**1、常见负载均衡**

L4: tcp 负载均衡 LVS

L4-L7 Haproxy 有图形界面

L7:http nginx （1.9以后的版本也支持tcp负载均衡）

nginx国内中小企业集群比较流行的负载均衡软件，严格的说叫做反向代理

**服务器负载均衡有三大基本Feature：负载均衡算法，健康检查和会话保持**

**集群常用软件：**

开源软件：Nginx LVS Haproxy Keepalived Heartbeat（哈的be特）

商业集群硬件：F5 Netscaler（耐死改了） A10等，工作模式都相当于Haproxy的工作模式。20-100万一台

**高可用软件：**

Keepalived：简单，易用,vip漂移、也可以通过脚本控制服务的关闭

**只要两台机器都可以做高可用、可以和mysql nfs nginx rsync 等软件结合，**

Heartbeat（哈的be特）：功能多，可以控制服务的启动和关闭，可以配合drdb做数据同步，需要数据同步的高可用（数据库，存储）

**负载均衡：**

Nginx：并发不是特别大，总访问量不大，简单，易用，稳定，nginx是一个web服务器，可以做反向代理，可以做缓存，

早期版本只支持http负载均衡，1.9以后的版本也支持tcp负载均衡、

LVS：大并发，只是四层TCP的负载均衡、没有七层负载均衡功能

Haproxy：专业负载均衡和反向代理，支持L4 L7负载均衡

**Heartbeat 和 LVS 数据库和存储的负载均衡和高可用可以使用**

**Heartbeat 和 drdb 不仅进行VIP切换，还可以支持数据同步**

**keepalived 和 haproxy 配置mysql的高可用负载均衡**

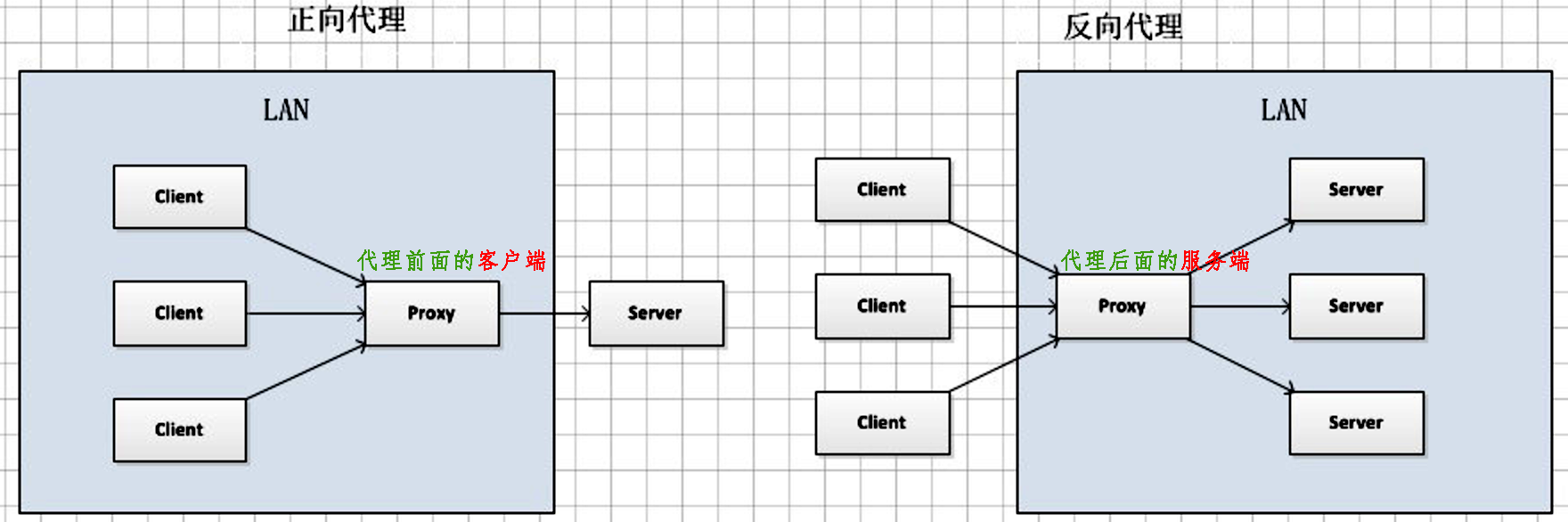
**集群常用软件：**

负载均衡：例如LVS 转发用户请求的数据包

反向代理：接收用户的请求、然后重新发起请求，去请求后面的节点

所以nginx严格的说是反向代理

正向代理和反向代理简单介绍：



正向代理中，proxy和client同属一个LAN，对server透明； 反向代理中，proxy和server同属一个LAN，对client透明。 实际上proxy在两种代理中做的事都是代为收发请求和响应，不过从结构上来看正好左右互换了下，所以把前者那种代理方式叫做正向代理，后者叫做反向代理

从用途上来区分

正向代理：正向代理用途是为了在防火墙内的局域网提供访问internet的途径。另外还可以使用缓冲特性减少网络使用率

反向代理：反向代理的用途是将防火墙后面的服务器提供给internet用户访问。同时还可以完成诸如负载均衡等功能

从安全性来讲：

正向代理：正向代理允许客户端通过它访问任意网站并且隐蔽客户端自身，因此你必须采取安全措施来确保仅为经过授权的客户端提供服务

反向代理：对外是透明的，访问者并不知道自己访问的是代理。对访问者而言，他以为访问的就是原始服务器

**两者的区别在于代理的对象不一样：正向代理代理的对象是客户端，反向代理代理的对象是服务端**

**haproxy与LVS的异同**

下面就这两种负载均衡软件的异同做一个简单总结：

　　1）两者都是软件负载均衡产品，但是LVS是基于Linux操作系统内核实现的一种软负载均衡，而HAProxy是基于第三应用实现的软负载均衡。

　　2）LVS是基于四层的IP负载均衡技术，而HAProxy是基于四层和七层技术、可提供TCP和HTTP应用的负载均衡综合解决方案。

　　3）LVS工作在ISO模型的第四层，因此其状态监测功能单一，而HAProxy在状态监测方面功能强大，可支持端口、URL、脚本等多种状态检测方式。

　　4）HAProxy虽然功能强大，但是整体处理性能低于四层模式的LVS负载均衡，而LVS拥有接近硬件设备的网络吞吐和连接负载能力。

综上所述，HAProxy和LVS各有优缺点，没有好坏之分，要选择哪个作为负载均衡器，要以实际的应用环境来决定。

HAProxy主要在于它有以下优点，总结如下：

　　1、HAProxy是支持虚拟主机的，通过frontend指令来实现

　　2、能够补充Nginx的一些缺点比如Session的保持，Cookie的引导等工作

　　3、支持url检测后端的服务器出问题的检测会有很好的帮助。

　　4、它跟LVS一样，本身仅仅就只是一款负载均衡软件；单纯从效率上来讲HAProxy更会比Nginx有更出色的负载均衡速度，在并发处理上也是优于Nginx的。

　　5、HAProxy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，不过在后端的MySQL slaves数量超过10台时性能不如LVS，所以向大家推荐LVS+Keepalived。

　　6、能对请求的url和header中的信息做匹配，有比lvs有更好的7层实现

**三大主流软件负载均衡器对比(LVS VS Nginx VS Haproxy)**

|  |
| --- |
| LVS： 1、抗负载能力强。抗负载能力强、性能高，能达到F5硬件的60%；对内存和cpu资源消耗比较低 2、工作在网络4层，通过vrrp协议转发（仅作分发之用），具体的流量由linux内核处理，因此没有流量的产生。 2、稳定性、可靠性好，自身有完美的热备方案；（如：LVS+Keepalived） 3、应用范围比较广，可以对所有应用做负载均衡； 4、不支持正则处理，不能做动静分离。 5、支持负载均衡算法：rr（轮循）、wrr（带权轮循）、lc（最小连接）、wlc（权重最小连接） 6、配置 复杂，对网络依赖比较大，稳定性很高。 |

|  |
| --- |
| Ngnix：  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；  2、Nginx对网络的依赖比较小，理论上能ping通就就能进行负载功能；  3、Nginx安装和配置比较简单，测试起来比较方便；  4、也可以承担高的负载压力且稳定，一般能支撑超过1万次的并发；  5、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。  6、Nginx对请求的异步处理可以帮助节点服务器减轻负载；  7、Nginx仅能支持http、https和Email协议，这样就在适用范围较小。  8、不支持Session的直接保持，但能通过ip\_hash来解决。、对Big request header的支持不是很好，  9、支持负载均衡算法：Round-robin（轮循）、Weight-round-robin（带权轮循）、Ip-hash（Ip哈希）  10、Nginx还能做Web服务器即Cache功能。 |

|  |
| --- |
| 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、能够补充Nginx的一些缺点比如Session的保持，Cookie的引导等工作 3、支持url检测后端的服务器出问题的检测会有很好的帮助。 4、更多的负载均衡策略比如：动态加权轮循(Dynamic Round Robin)，加权源地址哈希(Weighted Source Hash)，加权URL哈希和加权参数哈希(Weighted Parameter Hash)已经实现 5、单纯从效率上来讲HAProxy更会比Nginx有更出色的负载均衡速度。 6、HAProxy可以对Mysql进行负载均衡，对后端的DB节点进行检测和负载均衡。 9、支持负载均衡算法：Round-robin（轮循）、Weight-round-robin（带权轮循）、source（原地址保持）、RI（请求URL）、rdp-cookie（根据cookie） 10、不能做Web服务器即Cache。 |

**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Nigix/HAproxy作为反向代理负载均衡（或者流量不大都可以不选用负载均衡），因为其配置简单，性能也能满足一般的业务场景。 2、网站并发达到一定程度之后，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。。  注：Niginx与Haproxy比较：Niginx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保存等。具体选型看使用场景，目前来说Haproxy由于弥补了一些Niginx的缺点用户量也不断在提升。 |

**衡量负载均衡器好坏的几个重要因素：**

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

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。综合上述，haproxy是性能优越的负载均衡、反向代理服务器

**nginx负载均衡介绍**

Nginx除了可以用作web服务器外，

还可以用来做高性能的反向代理服务器，它能提供稳定高效的负载均衡解决方案。

nginx可以用轮询、IP哈希、URL哈希等方式调度后端服务器，同时也能提供健康检查功能。

目前有众多公司均已经部署使用nginx实现基于七层的负载均衡功能。

Nginx负载均衡

为了实现Nginx的反向代理以及负载均衡功能，应用中需要用到两个模块，HttpProxyModule和HttpUpstreamModule模块；

1. **upstream ngx\_http\_upstream\_module 定义节点和调度算法**
2. **http\_proxy 负责请求的转发，proxy\_pass 发给定义好的指定upstream名字**