作者：James-Jack  
链接：https://zhuanlan.zhihu.com/p/152526491  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

看完这篇文章，还不懂nginx，算我输

文章较长，阅读大概需要5分钟，全程干货，建议点赞收藏，以便下次阅读！

1. Nginx产生~

Nginx 同 Apache 一样都是一种 Web 服务器。基于 REST 架构风格，以统一资源描述符（Uniform Resources Identifier）URI 或者统一资源定位符（Uniform Resources Locator）URL 作为沟通依据，通过 HTTP 协议提供各种网络服务。

然而，这些服务器在设计之初受到当时环境的局限，例如当时的用户规模，网络带宽，产品特点等局限并且各自的定位和发展都不尽相同。这也使得各个 Web 服务器有着各自鲜明的特点。

Apache 的发展时期很长，而且是毫无争议的世界第一大服务器。它有着很多优点：稳定、开源、跨平台等等。

它出现的时间太长了，它兴起的年代，互联网产业远远比不上现在。所以它被设计为一个重量级的。

**但是它不支持高并发的服务器。在 Apache 上运行数以万计的并发访问，会导致服务器消耗大量内存。**

操作系统对其进行进程或线程间的切换也消耗了大量的 CPU 资源，导致 HTTP 请求的平均响应速度降低。

这些都决定了 Apache 不可能成为高性能 Web 服务器，**轻量级高并发服务器 Nginx 就应运而生了。**

俄罗斯的工程师 Igor Sysoev，他在为 Rambler Media 工作期间，使用 C 语言开发了 Nginx。

Nginx 作为 Web 服务器一直为 Rambler Media 提供出色而又稳定的服务。然后呢，Igor Sysoev 将 Nginx 代码开源，并且赋予自由软件许可证。

基于以下几点：Nginx火了

1. Nginx 使用基于**事件驱动架构，使得其可以支持数以百万级别的 TCP 连接。**
2. 高度的模块化和自由软件许可证使得第三方模块层出不穷（这是个开源的时代啊）。
3. Nginx 是一个**跨平台服务器，可以运行在 Linux、Windows、FreeBSD、Solaris、AIX、Mac OS 等操作系统上**。
4. 这些优秀的设计带来的极大的稳定性。

Nginx 的用武之地

关于代理

1. 说到代理，首先我们要明确一个概念，所谓代理就是**一个代表、一个渠道**；此时就涉及到**两个角色，一个是被代理角色，一个是目标角色。**
2. 被代理角色通过这个代理访问目标角色完成一些任务的过程称为代理操作过程；如同生活中的专卖店，客人到 adidas 专卖店买了一双鞋，这个专卖店就是代理，被代理角色就是 adidas 厂家，目标角色就是用户。
3. 简而言之，就是adidas老板找来专卖店这个代理来卖鞋子给客人这个目标角色。

正向代理：传说中的“翻墙”

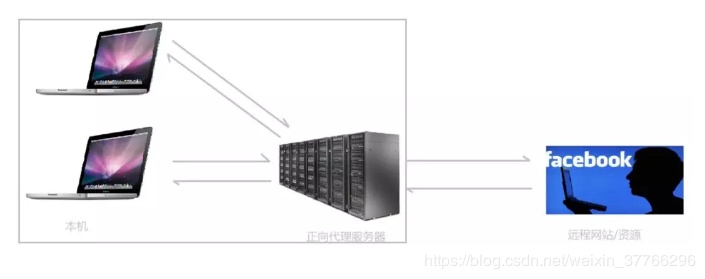
说反向代理之前，我们先看看正向代理，正向代理也是大家最常接触到的代理模式，我们会从两个方面来说关于正向代理的处理模式，分别从软件方面和生活方面来解释一下什么叫正向代理。

在如今的网络环境下，我们如果由于技术需要要**去访问国外的某些网站**，此时你会发现位于国外的某网站我们通过浏览器是没有办法访问的。

此时大家可能都会用一个操作 FQ 进行访问，FQ 的方式主要是**找到一个可以访问国外网站的代理服务器，我们将请求发送给代理服务器，代理服务器去访问国外的网站，然后将访问到的数据传递给我们**！

上述这样的代理模式称为正向代理，正向代理最大的特点是客户端非常**明确要访问的服务器地址**；\*\*服务器只清楚请求来自哪个代理服务器，而不清楚来自哪个具体的客户端；\*\*正向代理模式屏蔽或者隐藏了真实客户端信息。

来看个示意图（我把客户端和正向代理框在一块，同属于一个环境，后面我有介绍）：



**客户端必须设置正向代理服务器**，当然前提是要知道正向代理服务器的 IP 地址，还有代理程序的端口。  
**总结来说**：正向代理，“它代理的是客户端”，是一个位于客户端和原始服务器（Origin Server）之间的服务器，为了从原始服务器取得内容，客户端向代理发送一个请求并指定目标（原始服务器）。

然后代理向原始服务器转交请求并将获得的内容返回给客户端。客户端必须要进行一些特别的设置才能使用正向代理。

**正向代理的用途：**

1. 访问原来无法访问的资源，如 Google。
2. 可以做缓存，加速访问资源。
3. 对客户端访问授权，上网进行认证。
4. 代理可以记录用户访问记录（上网行为管理），对外隐藏用户信息。

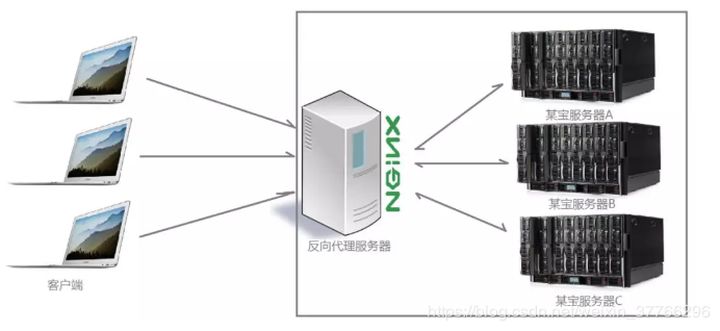
反向代理

明白了什么是正向代理，我们继续看关于反向代理的处理方式，举例如我国的某宝网站，每天同时连接到网站的访问人数已经爆表，单个服务器远远不能满足人民日益增长的购买欲望了。

此时就出现了一个大家耳熟能详的名词：分布式部署；也就是通过部署多台服务器来解决访问人数限制的问题。

某宝网站中大部分功能也是直接使用 Nginx 进行反向代理实现的，并且通过封装 Nginx 和其他的组件之后起了个高大上的名字：Tengine。

那么反向代理具体是通过什么样的方式实现的分布式的集群操作呢，我们先看一个示意图（我把服务器和反向代理框在一块，同属于一个环境，后面我有介绍）：



通过上述的图解大家就可以看清楚了，**多个客户端给服务器发送的请求，Nginx 服务器接收到之后，按照一定的规则分发给了后端的业务处理服务器进行处理了。**

此时请求的来源也就是客户端是明确的，但是**请求具体由哪台服务器处理的并不明确了**，Nginx 扮演的就是一个反向代理角色。

客户端是无感知代理的存在的，反向代理对外都是透明的，访问者并不知道自己访问的是一个代理。因为客户端不需要任何配置就可以访问。

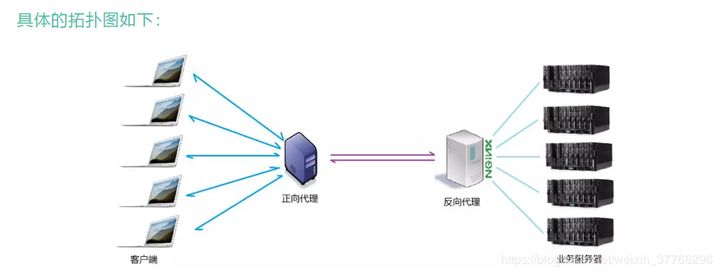
**反向代理，“它代理的是服务端”，主要用于服务器集群分布式部署的情况下，反向代理隐藏了服务器的信息。**

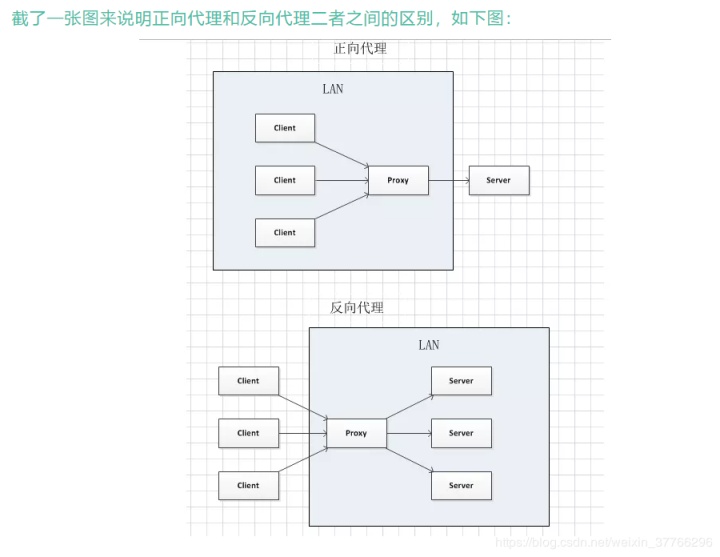
**反向代理的作用：**

保证内网的安全，通常将反向代理作为公网访问地址，Web 服务器是内网。  
负载均衡，通过反向代理服务器来优化网站的负载。

**项目场景**

通常情况下，我们在实际项目操作时，正向代理和反向代理很有可能会存在同一个应用场景中，正向代理代理客户端的请求去访问目标服务器，目标服务器是一个反向单利服务器，反向代理了多台真实的业务处理服务器。



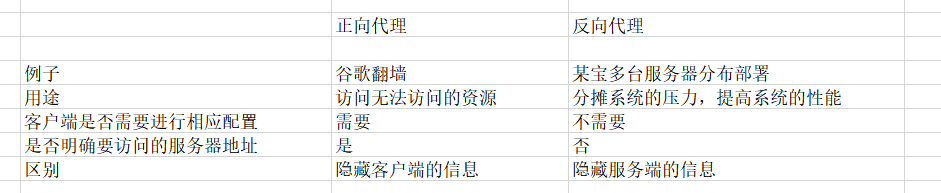


图解：

在正向代理中，Proxy 和 Client 同属于一个 LAN（图中方框内），隐藏了客户端信息。  
在反向代理中，Proxy 和 Server 同属于一个 LAN（图中方框内），隐藏了服务端信息。

实际上，Proxy 在两种代理中做的事情都是替服务器代为收发请求和响应，不过从结构上看正好左右互换了一下，所以把后出现的那种代理方式称为反向代理了。

**正向代理和反向代理区别总结：**



**负载均衡**

我们已经明确了所谓代理服务器的概念，那么接下来，Nginx 扮演了反向代理服务器的角色，它是依据什么样的规则进行请求分发的呢？不用的项目应用场景，分发的规则是否可以控制呢？

这里提到的客户端发送的、Nginx 反向代理服务器接收到的请求数量，就是我们说的负载量。

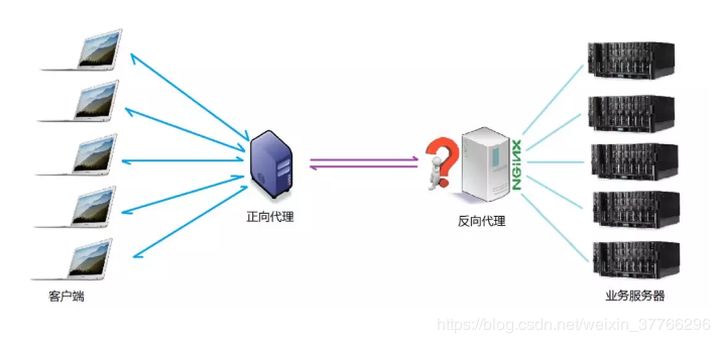
请求数量按照一定的规则进行分发，到不同的服务器处理的规则，就是一种均衡规则。

所以将服务器接收到的请求按照规则分发的过程，称为负载均衡。

负载均衡在实际项目操作过程中，有硬件负载均衡和软件负载均衡两种，硬件负载均衡也称为硬负载，如 F5 负载均衡，相对造价昂贵成本较高。

但是数据的稳定性安全性等等有非常好的保障，如中国移动中国联通这样的公司才会选择硬负载进行操作。

更多的公司考虑到成本原因，会选择使用软件负载均衡，软件负载均衡是利用现有的技术结合主机硬件实现的一种消息队列分发机制。



**Nginx 支持的负载均衡调度算法方式如下：**

**①weight 轮询（默认）**：接收到的请求按照顺序逐一分配到不同的后端服务器，即使在使用过程中，某一台后端服务器宕机，Nginx 会自动将该服务器剔除出队列，请求受理情况不会受到任何影响。

这种方式下，可以给不同的后端服务器**设置一个权重值（weight）**，用于调整不同的服务器上请求的分配率。

权重数据越大，被分配到请求的几率越大；该权重值，主要是针对实际工作环境中不同的后端服务器硬件配置进行调整的。

**②ip\_hash**：每个请求按照发起客户端的 ip 的 hash 结果进行匹配，这样的算法下一个**固定 ip 地址的客户端总会访问到同一个后端服务器**，这也在一定程度上解决了**集群部署环境下 Session 共享**的问题。

③fair：智能调整调度算法，动态的根据后端服务器的请求处理到响应的时间进行均衡分配。

响应时间短处理效率高的服务器分配到请求的概率高，响应时间长处理效率低的服务器分配到的请求少，它是结合了前两者的优点的一种调度算法。

但是需要注意的是 Nginx 默认不支持 fair 算法，如果要使用这种调度算法，请安装 upstream\_fair 模块。

④url\_hash：按照访问的 **URL 的 hash 结果分配请求**，每个请求的 URL 会指向后端固定的某个服务器，可以在 **Nginx 作为静态服务器的情况下提高缓存效率。**

同样要注意 Nginx 默认不支持这种调度算法，要使用的话需要安装 Nginx 的 hash 软件包。



很多人看到我总结了这篇文章，都来私聊我怎么样学好Nginx高性能服务器的知识，我的总结就是多阅读，多输出，每次我看完书我都会坚持写技术总结输出，一方面可以检测自己看的书是否真正的学进去，有句话说得好，如果你能把知识通俗易懂地讲给其他人听，并且其他人也能听懂你的讲解，说明你是真正地学会了；另一方面是想分享给其他需要的朋友看看，如果哪里讲得不好，也欢迎大家前来讨论，互相讨论的同时，也是提高自己的一种方法。

下面推荐一下我总结这篇文档看的书籍《精通Nginx》。

这本书主要是以模块为单位进行编写的，基本每个章节都是一个独立的模块，你可以从任意一个模块开始阅读。全书主要是有九个章节，理论和实践相结合，让读者更容易理解，这本书适用于任何对nginx有兴趣的读者，阅读这本书并不需要很深的nginx使用经验。下面附上目录信息给大家参考一下：

目录

第1章　安装Nginx及第三方模块

第2章　配置指南

第3章　使用mail模块

第4章　Nginx作为反向代理

第5章　反向代理高级话题

第6章　Nginx HTTP服务器

第7章　Nginx的开发 133

第8章　在Nginx中集成Lua 159

第9章　故障排除技巧