

蔷薇

爱你一万年^_^

[博客园](#)[首页](#)[新随笔](#)[联系](#)[订阅](#)[管理](#)

Nginx 相关介绍(Nginx是什么?能干嘛?)

Nginx的产生

没有听过Nginx? 那么一定听过它的"同行"Apache吧! 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。它是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。

由于:

- 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
- 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~)
- 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
- 这些优秀的设计带来的是极大的稳定性

所以, Nginx火了!

Nginx的用武之地

Nginx是一款自由的、开源的、高性能的HTTP服务器和反向代理服务器; 同时也是一个IMAP、POP3、SMTP代理服务器; Nginx可以作为是一个HTTP服务器进行网站的发布处理, 另外Nginx可以作为反向代理进行负载均衡的实现。

关于代理

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, 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。

公告

昵称: 失恋的蔷薇
园龄: 3年2个月
粉丝: 279
关注: 0
[+加关注](#)

<		2020年9月			
	日	一	二	三	四
30	31	1	2	3	
6	7	8	9	10	
13	14	15	16	17	
20	21	22	23	24	
27	28	29	30	1	
4	5	6	7	8	

搜索

最新随笔

1.Python 实现国产SM3加密集

2.Docker 主要操作命令

3.Pycharm & WebStorm 如何
中词汇同时高亮

4.JS 返回前一个月的当天日期

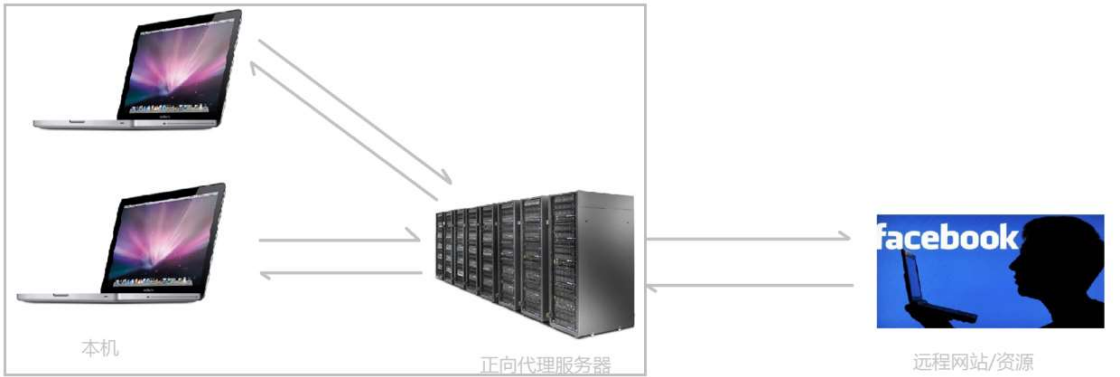
5.VSCode 如何同步设置

正向代理

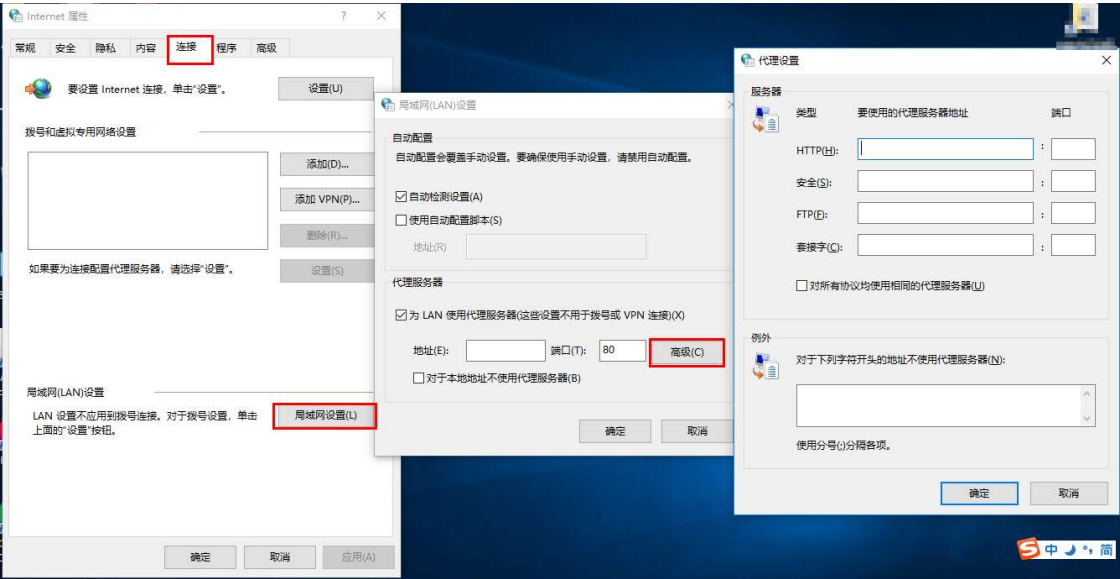
说反向代理之前，我们先看看正向代理，正向代理也是大家最常接触的到的代理模式，我们会从两个方面来说关于正向代理的处理模式，分别从软件方面和生活方面来解释一下什么叫正向代理。

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。如图。



总结来说：正向代理，“它代理的是客户端，代客户端发出请求”，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

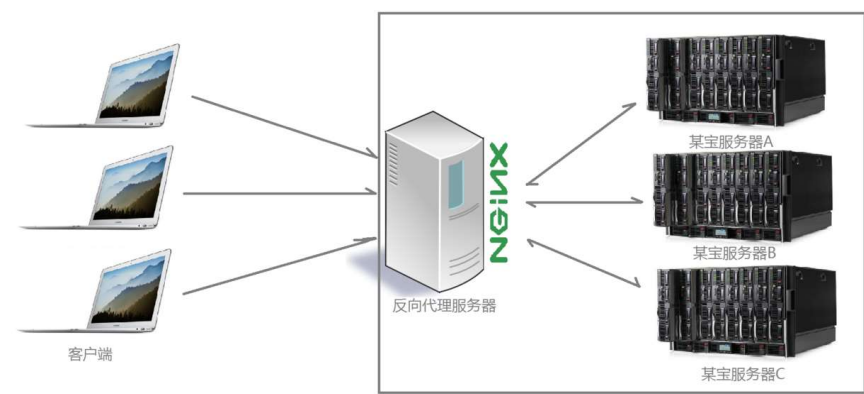
反向代理

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，举例如我大天朝的某宝网站，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，有兴趣的童鞋可以访问Tengine的官网查看具体的信息：<http://tengine.taobao.org/>。那么

积分与排名
积分 - 141568
排名 - 4968
随笔分类
Apache(5)
Django(24)
Docker(1)
Java(2)
Linux(8)
MongoDB(3)
MySQL(14)
Nginx(3)
Oracle(1)
PyCharm(6)
Python(35)
Redis(2)
REST Framework(2)
Scrapy(4)
其他(4)
前端(10)

随笔档案
2020年9月(1)
2020年6月(1)
2020年3月(1)
2020年1月(1)

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后端的业务处理服务器进行了处理。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。

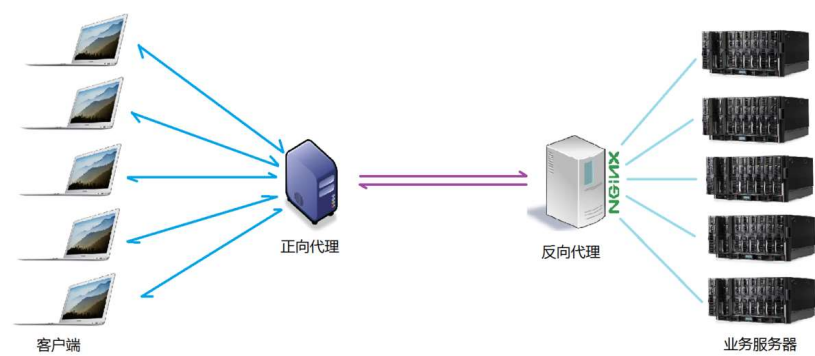
客户端是无感知代理的存在的，反向代理对外都是透明的，访问者并不知道自已访问的是一个代理。因为客户端不需要任何配置就可以访问。

反向代理，"它代理的是服务端，代服务端接收请求"，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：
- (1) 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
 - (2) 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在一个应用场景中，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。具体的拓扑图如下：

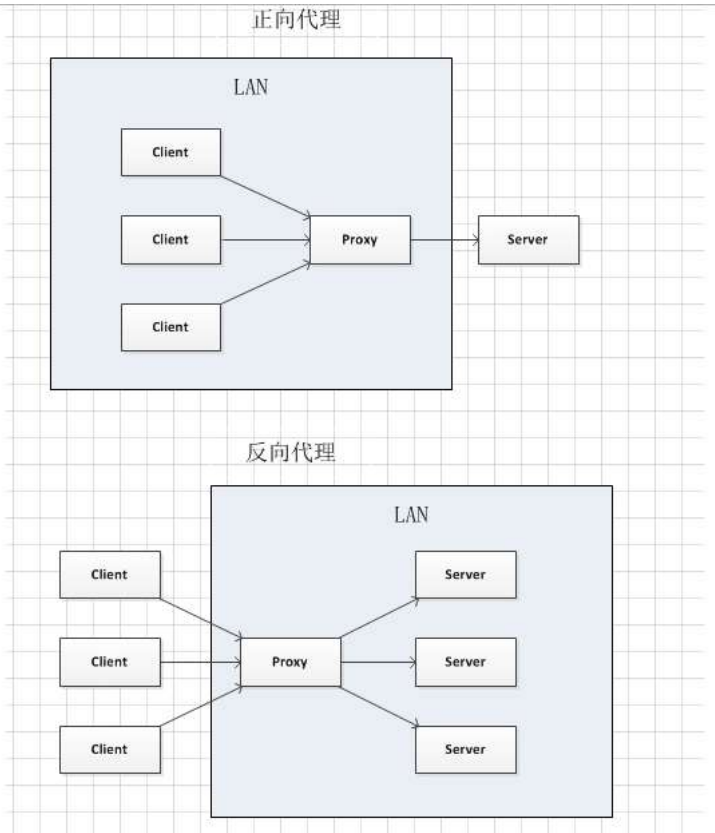


二者区别

截了一张图来说明正向代理和反向代理二者之间的区别，如图。

2019年11月(1)
2019年8月(3)
2019年7月(1)
2019年6月(2)
2019年5月(1)
2019年2月(3)
2019年1月(4)
2018年12月(2)
2018年11月(6)
2018年10月(4)
2018年9月(1)
2018年8月(5)
2018年7月(11)
2018年6月(9)
2018年5月(4)
2018年4月(7)
2018年3月(7)
2017年12月(7)
2017年9月(1)
2017年8月(1)
2017年7月(11)

最新评论
1. Re:Apache Windows下Ap 步骤
1、文章不错，已转载；



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。

负载均衡

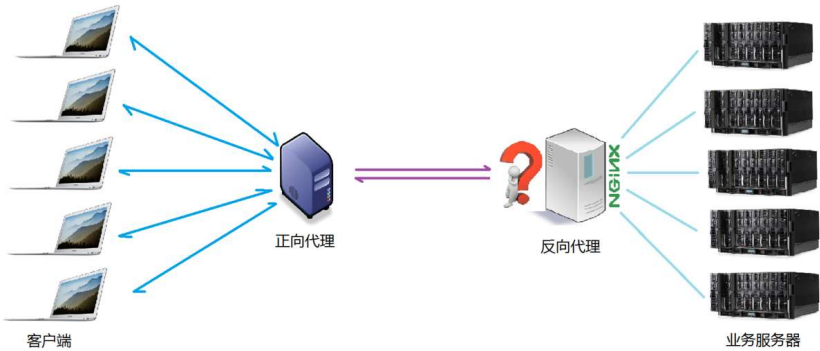
我们已经明确了所谓代理服务器的概念，那么接下来，Nginx扮演了反向代理服务器的角色，它是以依据什么样的规则进行请求分发的呢？不同的项目应用场景，分发的规则是否可以控制呢？

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：

2、已发红包，另外能加个微信

2. Re: Docker 主要操作命令
从刘江的django教程追过来的
圳工作

3. Re: Nginx 相关介绍(Nginx
干嘛?)
已转载，并注明了出处，再次感

4. Re: Nginx 相关介绍(Nginx
干嘛?)
讲解的通俗易懂，很透彻，学习

5. Re: Django 信号signal
看到在刘江老师博客中，你评论
号和消息的用途，发现你挺细聊

阅读排行榜

1. Nginx 相关介绍(Nginx是什
嘛?)(217609)
2. Apache Windows下Apact
(53834)
3. Python 实例方法、类方法、
的区别与作用(51190)
4. Apache Linux下Apache安
25)
5. Python minidom模块(DO
析XML)(36209)

1. weight轮询（默认，常用，具有HA功效！）：接收到的请求按照权重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。
2. ip_hash（常用）：每个请求按访问ip的hash结果分配，这样每个访客固定访问一个后端服务器，这也在一定程度上解决了集群部署环境下session共享的问题。
3. fair：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；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。
4. url_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。

几种常用web服务器对比

对比项\服务器	Apache	Nginx	Lighttpd
Proxy代理	非常好	非常好	一般
Rewriter	好	非常好	一般
Fcgi	不好	好	非常好
热部署	不支持	支持	不支持
系统压力	很大	很小	比较小
稳定性	好	非常好	不好
安全性	好	一般	一般
静态文件处理	一般	非常好	好
反向代理	一般	非常好	一般

至此。转载请注明出处，记得扫码打赏支持哦，谢谢！

[本站相关链接：[Nginx 一个高性能的HTTP和反向代理服务器](#)、[Nginx 安装与部署配置以及Nginx和uWSGI开机自启](#)、[基于Linux和Nginx的Django部署](#)]



分类： Nginx

[好文要顶](#)[关注我](#)[收藏该文](#)

评论排行榜

1. Nginx 相关介绍(Nginx是什么?)(103)
2. Nginx 安装与部署配置以及SGI开机自启(11)
3. PyCharm 在PyCharm中使(9)
4. Python 实例方法、类方法、的区别与作用(8)
5. Django Windows+Apachsgi环境下部署(3)

推荐排行榜

1. Nginx 相关介绍(Nginx是什么?)(142)
2. Python 实例方法、类方法、的区别与作用(9)
3. Nginx 安装与部署配置以及SGI开机自启(6)
4. Apache Windows下Apact(5)
5. VSCode 如何同步设置(3)