01 开篇词：你为什么要学 Nginx？

更新时间：2019-11-26 14:43:41

> 一个人追求的目标越高，他的才力就发展得越快，对社会就越有益。——高尔基

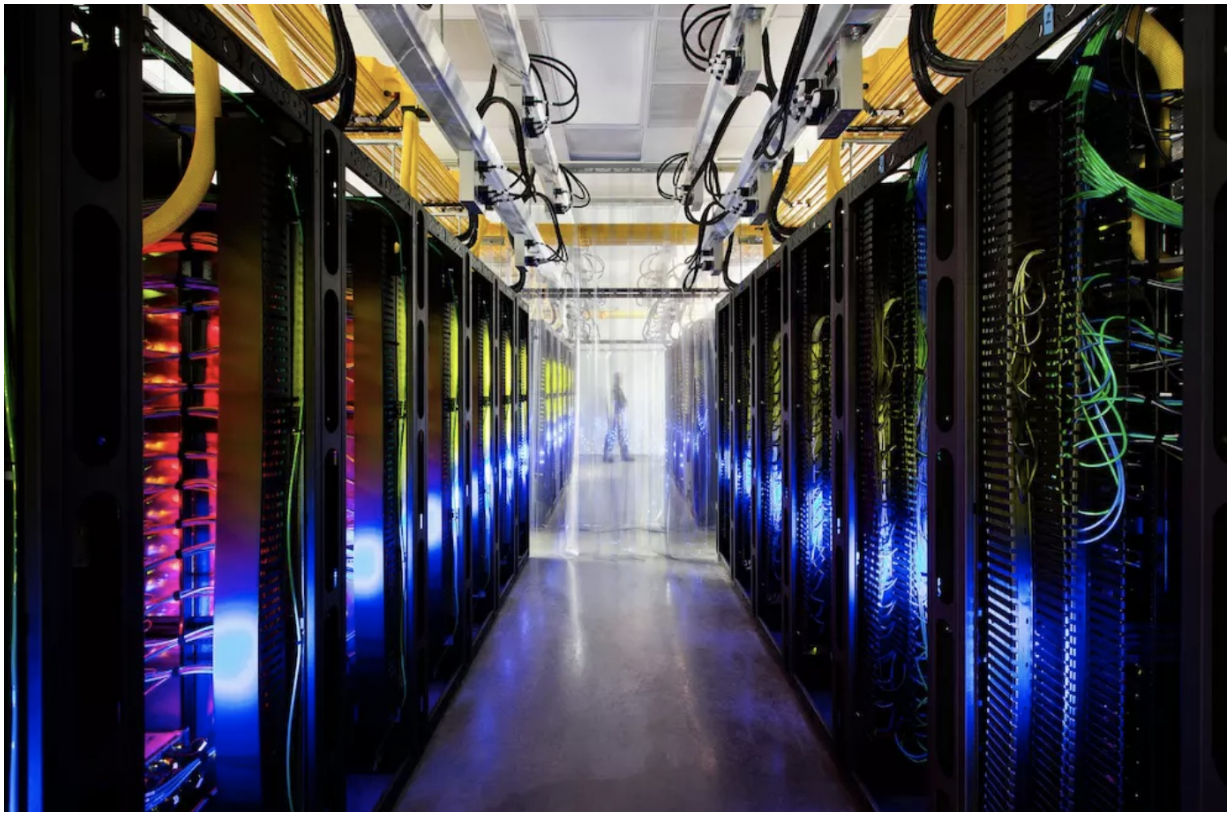
大家好，欢迎大家学习 《Nginx 从入门到实战》 专栏。在这个专栏系列中，我将带领大家从浅入深、由表及里，深入学习和理解 Nginx 。

简单来说 Nginx 其实就是一款轻量级的 Web 服务器、反向代理服务器，由于它的内存占用少，启动极快，高并发能力强，在互联网项目中广泛应用。那么你可能会问了：“不是说 Nginx 吗？怎么又扯出来一个 Web 服务器呢？啥是服务器？”。别着急，在学习 Nginx 之前你还真得知道什么是服务器这个概念，下面我们先来看下到底啥是服务器。

啥是服务器

服务器的英文叫 Server ，顾名思义就是为其他人服务的。我们可以把所有为其他用户提供服务的机器或软件都称作服务器。

Server 可以指硬件，比如谷歌公司对用户提供服务的主机，下图是谷歌公司的一个数据中心图片，走道两侧亮着红黄蓝颜色的笨重机器就是服务器。



Server 也可以是一个软件。比如我们本次专栏要介绍的 Nginx，它就是一个对用户提供 HTTP 服务的 Server。又比如我们常听到的网易邮箱服务器，它就是专门为用户提供邮件服务的。这些软件都运行在一个个物理机器上面，专门对外提供对应的服务。

WEB服务器

我们上面唠叨了一下什么是服务器。下面我们说一下啥是 Web 服务器，在上面我们也说了 Nginx 其实就是一个 轻量级的 Web 服务器，那么什么是 Web 服务器呢？

我们平时可以打开浏览器访问微博的网站，获取各种娱乐圈各种劲爆消息。这个看似简单的行为，背后有隐藏了那些细节呢？

上图就是一个典型的 web 请求流程，主要分为五个步骤：

1. 浏览器本身作为一个客户端，当你输入 www.weibo.com 的时候，向 DNS 服务器发出域名请求服务；
2. DNS 服务器将域名将对应的 IP 地址返回给浏览器；
3. 浏览器使用 IP 地址找到对应的服务器后，建立 TCP 连接，向服务器发送 HTTP 请求；
4. 服务器接收到请求之后才开始处理，返回 HTTP 响应；
5. 浏览器收到来自服务器的响应后开始渲染页面，最后断开与该服务器之间的 TCP 连接。

我们所介绍的 Web 服务器就是在第 3 和 4 步骤中发挥作用的。它的作用很简单，概括地讲，主要完成三个工作：

1. 接收请求；
2. 处理请求，生成响应 ；
3. 发送响应。

自己写个web服务器

上面我们也讲到了，虽然 Web 服务器这个名字听起来很高大上，其实它的原理非常的简单，我们也可以自己写一个服务器。下面以 go 语言为例 DIY 一个自己的服务器：

```go
package main

import (
    "fmt"
    "log"
    "net/http"
)

func sayHelloName(w http.ResponseWriter, r *http.Request) {
    r.ParseForm()
    fmt.Fprintf(w, "当前请求的地址为:"+r.URL.Path)
}

func main() {
    http.HandleFunc("/", sayHelloName)
    err := http.ListenAndServe(":8888", nil)
    if err != nil {
        log.Fatal("ListenAndServe: ", err)
    }
}
```

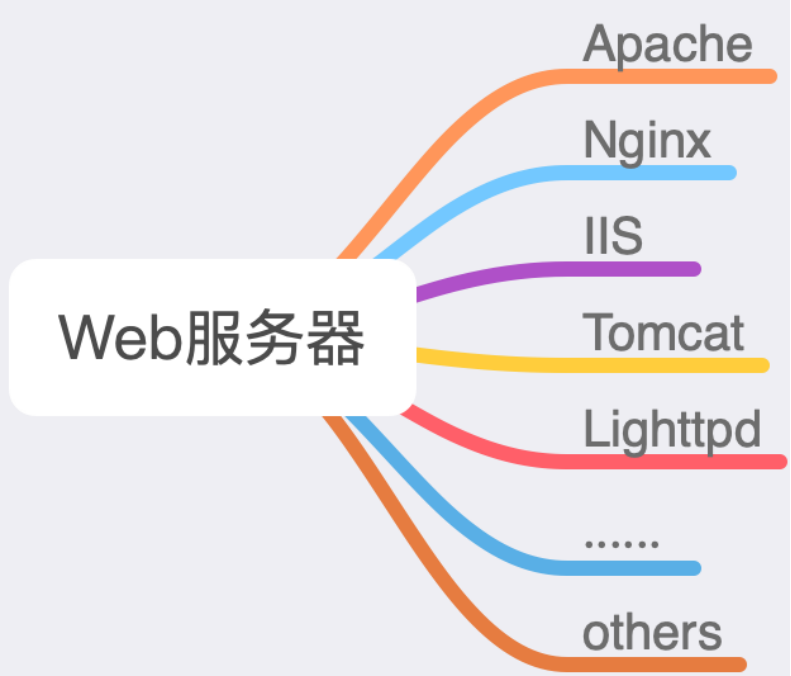
这就是一个很简单的 web 服务器，它监控本机的 8888 端口。当我们访问 localhost:8888 的时候，浏览器上会输出我们当前的访问路径：

我们只是使用了几行代码就可以完成 Web 服务器的基本功能。是不是很简单呢？

> 哈哈，当然了，一个成熟的 Web 服务器肯定还有其他各种各样的功能，但是原理都是一样。所以，一定要淡定，学会 Web 服务器并没有想象中的那么困难。

WEB服务器的分类

其实，现在的市面上面有很多类似于 Nginx 的 Web 服务器，如下图:

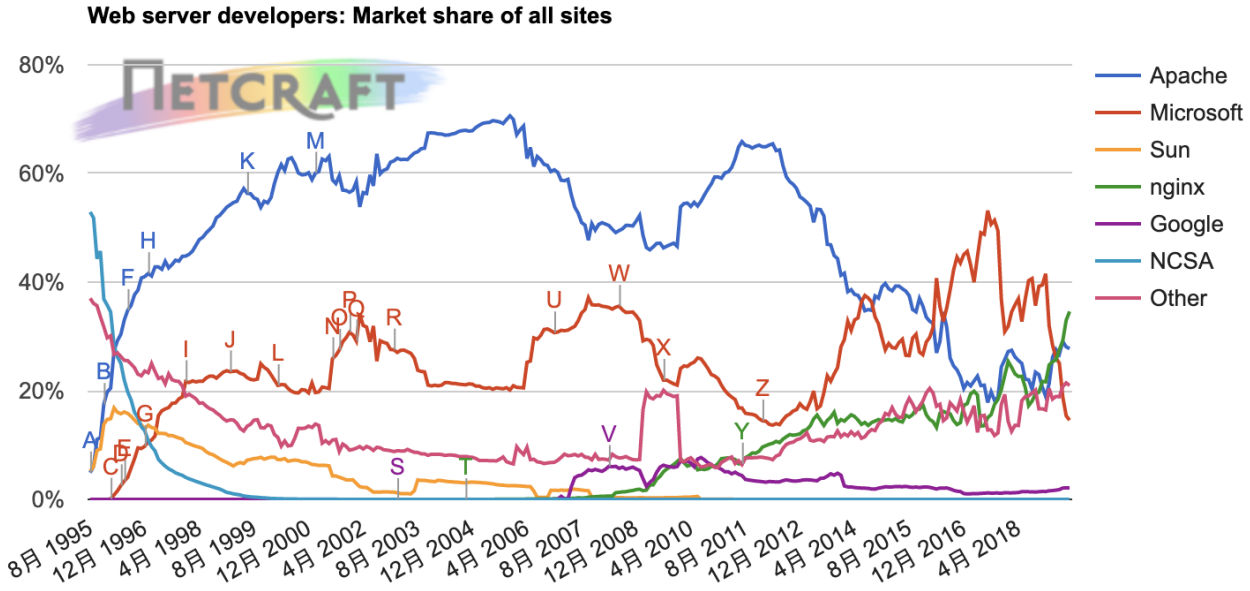


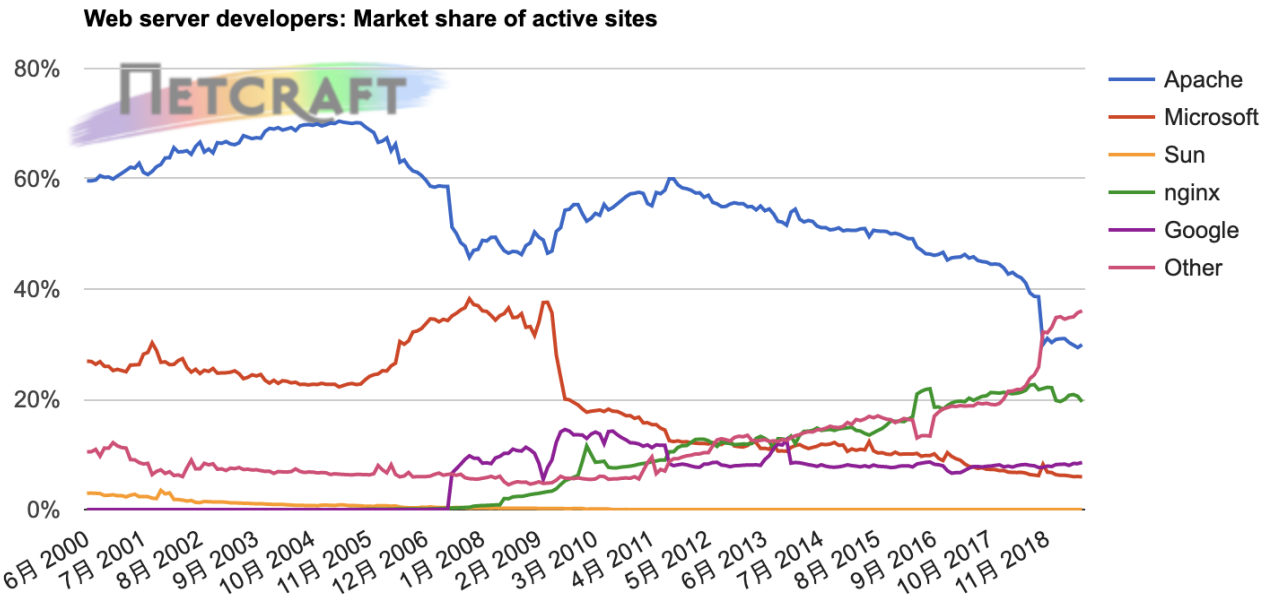
几款服务器比较

下面我们简单地比较一下几款流行的 Web 服务器软件的区别。

1. Apache 是一款历史悠久的开源 Web 服务器软件。拥有很多的第三方模块，你想使用的功能基本上都可以找到，避免自己重复造轮子。采用多进程方式处理请求，每个请求都对应于一个进程。在高并发的情况下，多进程处理方式特别的消耗资源，所以天然不适合高并发场景；
2. Nginx 是年轻(相比 Apache)的开源 Web 服务器软件。它也采用 C 语言编写，代码运行效率很高。它采用了 epoll (以 Linux 为例) 事件处理机制作为模型，能够保证非常高的并发量；
3. IIS 是微软的 Web 服务器软件，这个可是收费的哦~~~

我们看一下 Netcraft 公司在2019年7月份统计的全球 Web 服务器占用率调查：





从这两张图中可以看到，Nginx 开发者数量已经明显的超过了其他 Web Server 开发者数量，独占鳌头。

在 Web Server 市场使用率中，Nginx 的占有率一直在稳步的上升，超过了 IIS，与 Apache 的差距越来越小。

所以，使用和学习 Nginx 已经成为每一个后端工程师必备的技能了。好了，铺垫了这么多，下面我们正式来看下 Nginx 倒底是个什么东西。

Nginx简介

Nginx是啥东东

Nginx 是由战斗民族的一位名叫 Igor Sysoev 的程序猿开发的，开源、高性能的 HTTP 服务器和反向代理服务器，也可以作为一个 IMAP/POP3 代理服务器。也就是说，Nginx 不仅可以托管网站，进行 HTTP 服务处理，还可以作为反向代理服务器。

Nginx 出现的初衷是为了解决著名的 C10K 问题而出现的。和传统的 Web Server 不一样，Nginx 使用了**异步事件**处理机制架构。这种架构可以轻松高效地处理大量的请求，并且非常的节省内存。高性能是 Nginx 最大的优点。

> 啥？不知道什么是 C10K 问题，我严重怀疑你是从外星球来的，来，看这里。
>
> 不知道什么是 异步事件 处理机制？没关系，我会在后面的小节中分析的，包学会。

为什么选择Nginx

占用内存小。这得益于 Nginx 使用 C 语言编写，能够高效使用 CPU、内存等系统资源。并且作者自己造了很多的轮子，比如 Nginx 自己实现了内存管理系统，动态数组机制等。Nginx 作者对**内存的使用控制简直到了丧心病狂**的地步，所以非常的节省系统资源，特别是内存；

高并发。在 Linux 系统上，Nginx 使用了 epoll 机制，能够高效处理大量的连接数。理论上，Nginx 可以同时处理的最大连接数取决于你的机器的物理内存，上不封顶；

高可靠性。我认为 Nginx 的高可靠性主要体现在两方面:

(一)：Nginx 使用了 **Master-Worker 机制**，真正处理请求的是 Worker 进程。Master 进程可以监控 Worker 进程的运行状况，当某个 Worker 进程因意外原因退出的时候，Master 会重新启动 Worker 进程；

(二)：Nginx 的内部框架非常优秀。它的各个模块都非常简单，所以也非常的稳定。

热部署。可能大家觉得这个原因并不重要，其实在实际的线上环境是非常重要的。代码上线之后，我们只需要执行 `nginx -s reload` 命令就可以完成 Nginx 的重启，其他的交给 Nginx 就可以了，你可以安心去喝咖啡了。

> 如果没有亲身体验过这种痛苦，你是无法理解的这是**多么痛的领悟**。

Nginx能干啥

上面我们 '吹嘘' 了 Nginx 的这么多优点，那么 Nginx 究竟能干什么呢？

HTTP 服务器。作为一款优秀的 Web 服务器，那么提供 HTTP 服务显然是它的首要任务；

负载均衡。Nginx 提供了多种负载均衡策略，实现了**7层负载均衡**。针对不同的情形，我们可以选择合适的策略。另外我们也可以自己实现特殊需求的负载均衡策略；

反向代理。Nginx 是一款非常优秀的反向代理服务器。

Talk is cheap，Show me your code, 下面我们就进入 Nginx 的世界吧…

}