

# Nginx 简介

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,它具有有很多非常优越的特性：

- **作为 Web 服务器：**

相比 Apache、Nginx 使用更少的资源，支持更多的并发连接，体现更高的效率，这点使 Nginx 尤其受到虚拟主机提供商的欢迎，能够支持高达 50,000 个并发连接数的响应。

- **作为负载均衡服务器：**

Nginx 既可以在内部直接支持 Rails 和 PHP，也可以支持作为 HTTP 代理服务器对外进行服务。Nginx 用 C 编写,不论是系统资源开销还是 CPU 使用效率都比 Perlbal 要好的多。

- **作为邮件代理服务器：**

Nginx 同时也是一个非常优秀的邮件代理服务器（最早开发这个产品的目的之一也是作为邮件代理服务器），Last.fm 描述了成功并且美妙的使用经验。

而在微服务中，每个功能可能都是一个独立的服务，如果通过传统的方式配置每个应用，配置文件会变得很复杂多变，所以我们需要服务的注册与发现。

## 注册与发现的原理

服务的注册与发现是通过服务注册表实现的，应用端（每个服务）通过配置文件向服务注册表提交自己的注册信息，当服务启动时，服务注册表会检索到该应用，并将该应用的网络地址添加到表中。同样当服务终止，服务注册表会删除服务的地址。

服务注册表是通过心跳机制实现的。当其他应用访问已注册的服务时，负载均衡会通过服务注册表，实现服务的发现。

# 为什么要使用 Nginx

微服务架构中，众多服务被拆分解耦，并部署到不同的容器以及服务器中，你可能使用了一个服务发现系统（例如 etcd 或 Eureka）或者一个资源管理框架来管理所有这些服务，可如果你想让你的用户去从互联网访问你的某些微服务，你就必需使用一个反向代理服务器，从而使你的众多微服务能够被访问到。

还有一个问题是当你拥有多个服务实例时，你希望能够轻松地连接到它们，将你的请求在它们中高效地分发，并以最快的方式执行，所以不同服务实例之间的负载均衡也是非常重要的问题。

而 Nginx 作为一款优秀的反向代理服务器和负载均衡服务器，他的诸多优秀特性成为了我们选择他的原因：

- **模块化设计**

Nginx 采用高度模块化设计，使得具有较好的扩展性，在 Nginx 中，除了少量的核心代码，其他一切皆为模块。所有模块间是分层次、分类别的，官方 Nginx 有五大类型的模块：核心模块、配置模块、事件模块、HTTP 模块、Mail 模块。

- **高可靠性**

高可靠性是指服务可靠性。Nginx 是一个高可靠性的 Web 服务器，这也是我们为什么选择 Nginx 的基本条件。Nginx 采用一个主进程(master)和 N 个工作进程(worker)的工作模式，而 worker 进程才是真正复制相应用户请求的进程。配置了缓存时还会有缓存加载器进程(cacheloader)和缓存管理器进程(cachemanager)等。所有进程均是仅含有一个线程，并主要通过“共享内存”的机制实现进程间通信。

- **支持热部署**

Nginx 使用主进程和 worker 工作进程的机制，使得 Nginx 支持热部署，这个热部署包括不

停止服务更新配置文件、更新日志文件、以及更新服务器程序版本，也称为平滑升级。

- **低内存消耗**

Nginx 对于内存的消耗是非常小的，特别是对于非活动连接。Nginx 对于非活动连接是指，当我们开启持久连接功能时，用户连接不再发送数据后就会立即转为非活动连接，直到持久连接超时时间到达才销毁。在一般的情况下，10000 个非活跃的 HTTPKeep-Alive 连接在 Nginx 中仅消耗 2.5M 的内存，这也是 Nginx 支持高并发连接的基础。

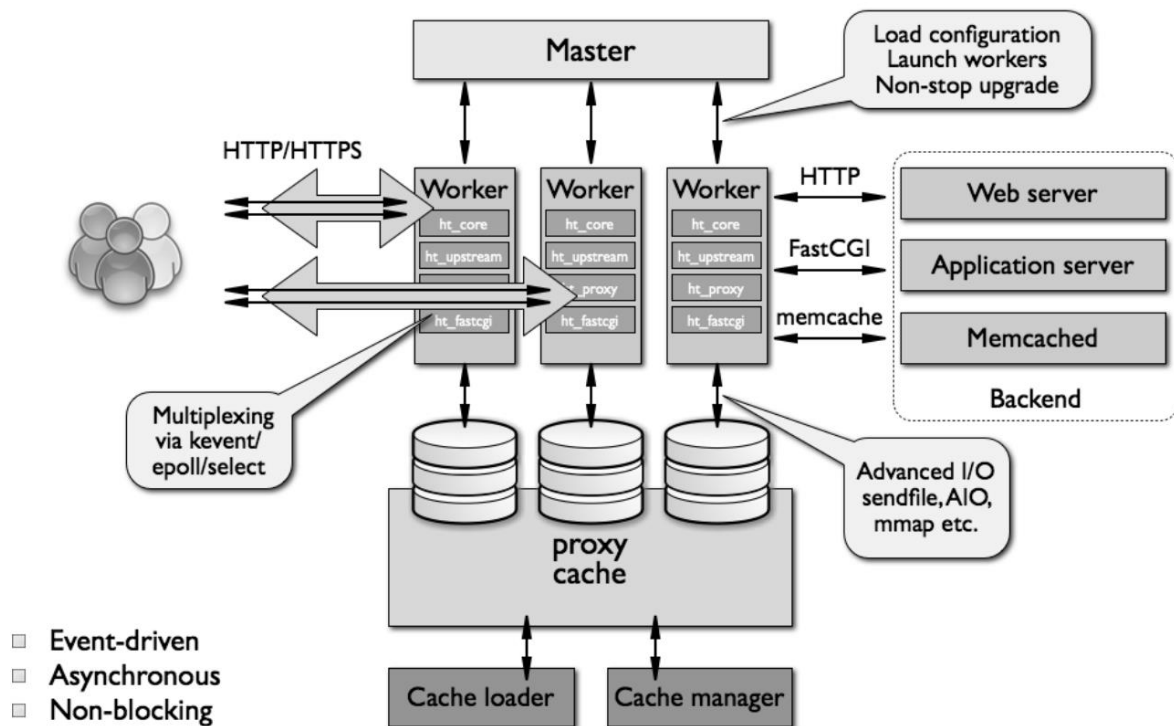
- **高扩展性**

Nginx 的设计具有扩展性，它完全是由多个不同功能、不同层次、不同类型且耦合度极低的模块组成。因此，当对某一个模块修复 Bug 或进行升级时，可以专注于模块自身，无须在意其他。Nginx 支持磁盘异步 I/O（AIO）、内存映射机制（MMAP）、事件驱动机制（Event-driven）、单线程 N 请求等等。

- **高并发**

Nginx 是异步非阻塞的。在需要进程等待的过程中，这些闲置的进程就空闲出来待命，而 webserver 的工作性质决定了每个 request 的大部份生命都是在网络传输中，实际上花费在 server 机器上的时间片不多，因此就表现为几个进程解决了高并发的問題。

# Nginx 架构



Nginx 在启动后，会有一个 master 进程和多个 worker 进程。master 进程主要用来管理 worker 进程，并不处理网络请求，主要负责调度工作进程，也就是图示的三项：加载配置、启动工作进程及非停升级。

服务器实际处理网络请求及响应的是工作进程（worker），在类 unix 系统上，Nginx 可以配置多个 worker，而每个 worker 进程都可以同时处理数以千计的网络请求。多个 worker 进程之间是对等的，他们同等竞争来自客户端的请求，各进程互相之间是独立的。一个请求，只可能在一个 worker 进程中处理，一个 worker 进程，不可能处理其它进程的请求。worker 进程的个数是可以设置的，一般我们会设置与机器 cpu 核数一致，这里面的原因与 Nginx 的进程模型以及事件处理模型是分不开的。代理（proxy）设计，可以说是 Nginx 深入骨髓的设计，无论是对于 HTTP，还是对于 FastCGI、memcache、Redis 等的网络请求或响应，本质上都采用了代理机制。所以，Nginx 天生就是高性能的代理服务器。

# Nginx 与 Tomcat 对比

- **Nginx 优点：**

负载均衡、反向代理、处理静态文件有优势。Nginx 处理静态请求的速度高于 apache。

- **Tomcat 优点：**

动态解析服务器，处理动态请求，是编译 JSP\Servlet 的容器，Nginx 有动态分离机制，静态请求直接就可以通过 Nginx 处理，动态请求才转发请求到后台交由 Tomcat 进行处理。