

● 博客

● Nginx

- 轻量级，采用 C 进行编写，同样的 web 服务，会占用更少的内存及资源
- 抗并发，nginx 以 epoll and kqueue 作为开发模型，处理请求是异步非阻塞的，负载能力比 apache 高很多，而 apache 则是阻塞型的。在高并发下 nginx 能保持低资源低消耗高性能，而 apache 在 PHP 处理慢或者前端压力很大的情况下，很容易出现进程数飙升，从而拒绝服务的现象。
- nginx 处理静态文件好，静态处理性能比 apache 高三倍以上
- nginx 的设计高度模块化，编写模块相对简单
- nginx 配置简洁，正则配置让很多事情变得简单，而且改完配置能使用 -t 测试配置有没有问题，apache 配置复杂，重启的时候发现配置出错了，会很崩溃
- nginx 作为负载均衡服务器，支持 7 层负载均衡
- nginx 本身就是一个反向代理服务器，而且可以作为非常优秀的邮件代理服务器
- 启动特别容易，并且几乎可以做到 7*24 不间断运行，即使运行数月也不需要重新启动，还能够不间断服务的情况下进行软件版本的升级
- 社区活跃，各种高性能模块出品迅速

Apache

1. apache 的 rewrite 比 nginx 强大，在 rewrite 频繁的情况下，用 apache
2. apache 发展到现在，模块超多，基本想到的都可以找到
3. apache 更为成熟，少 bug，nginx 的 bug 相对较多
4. apache 超稳定
5. apache 对 PHP 支持比较简单，nginx 需要配合其他后端用
6. apache 在处理动态请求有优势，nginx 在这方面是鸡肋，一般动态请求要 apache 去做，nginx 适合静态和反向。
7. apache 仍然是目前的主流，拥有丰富的特性，成熟的技术和开发社区

总结

两者最核心的区别在于 apache 是同步多进程模型，一个连接对应一个进程，而 nginx 是异步的，多个连接（万级别）可以对应一个进程。一般来说，需要性能的 web 服务，用 nginx。如果不需要性能只求稳定，更考虑 apache，后者的各种功能模块实现得比前者，例如 ssl 的模块就比前者好，可配置项多。epoll (freebsd 上是 kqueue) 网络 IO 模型是 nginx 处理性能高的根本理由，但并不是所有的情况下都是 epoll 大获全胜的，如果本身提供静态服务的就只有寥寥几个文件，apache 的 select 模型或许比 epoll 更高性能。当然，这只是根据网络 IO 模型的原理作的一个假设，真正的应用还是需要实测了再说的。

更为通用的方案是，前端 nginx 抗并发，后端 apache 集群，配合起来会更好。