来源：伯乐在线 - 独自等待\_天涯

如需转载，请发送「转载」二字查看详情；

[如有好文章投稿，请点击 → 这里了解详情](http://mp.weixin.qq.com/s?__biz=MzA5ODM5MDU3MA==&mid=402922032&idx=1&sn=3a36315ebb17544d454c1a24fc897d62&scene=21#wechat_redirect)

NGINX以高性能的负载均衡器，缓存，和web服务器闻名，驱动了全球超过 40% 最繁忙的网站。在大多数场景下，默认的 NGINX 和 Linux 设置可以很好的工作，但要达到最佳性能，有些时候必须做些调整。本文将讨论当调优系统时要考虑的一些NGINX和Linux设置。

有太多可以调优的设置，但本文只涵盖一小部分设置，这些设置对大多数使用者有优化的好处。本文不包括那些设置，那些需要必须深入理解 NGINX 和 Linux，或者需要 Nginx 支持团队或专业服务团队指导才能做的设置。专业服务团队已经和很多全球热门网站共事，调优Nginx 以达到最高性能，他们可以与你一起共事，充分利用NGINX或NGINX部署。

### **简介**

本文假设你已经对 NGINX 架构和配置的概念有一个基本的理解，本文不会重复 NGINX 文档，但会提供各种选项的概述和相关文档的链接。

调优的时候要遵循的一个准则：一次只改一个设置，如果对性能无有效提升，就改回默认值。

我们先从讨论Linux调优开始，因为一些操作系统配置的设置决定了如何优化Nginx的配置。

### **调优Linux 的配置**

Linux 内核（2.6以上）的设置已经适用于大多数场合，不过对一些设置的调整，会有更大的收益。如果系统配置太低，检查错误信息的内核日志则会提示建议升级。在这里我们只涉及最有可能在常规工作负载下调优很有收益的配置。调整配置的更多细节，请参考Linux文档。

#### **缓冲区队列**

下面的设置与连接及其如何排队相关。如果传入连接速率很高，导致性能参次不齐（例如一些连接似乎停滞了），改变这些配置会有效。

* net.core.somaxconn：排队等待连接的最大数目，由NGINX可接受的数目决定。默认值通常很低，但可以接受，因为NGINX 接收连接非常快，但如果网站流量大时，就应该增加这个值。内核日志中的错误消息会提醒这个值太小了，把值改大，直到错误提示消失。

注意： 如果设置这个值大于512，相应地也要改变NGINX listen指令的backlog参数。

* net.core.netdev\_max\_backlog ： 在提交到CPU前网卡中数据包缓冲的速率，高带宽下提高这个值可提高性能。检查内核日志文件中有关这个设置的错误，根据网卡文档中的建议修改这个值。

#### **文件描述符**

文件描述符是操作系统资源，用于表示连接、打开的文件，以及其他信息。NGINX 每个连接可以使用两个文件描述符。例如，如果NGINX充当代理时，通常一个文件描述符表示客户端连接，另一个连接到代理服务器，如果开启了HTTP 保持连接，这个比例会更低（译注：为什么更低呢）。对于有大量连接服务的系统，下面的设置可能需要调整一下：

* sys.fs.file\_max —— 文件描述符系统级别的限制
* nofile —— 用户级别文件描述符限制，在 /etc/security/limits.conf 文件中修改。

#### **临时端口**

当NGINX充当代理时，每个到上游服务器的连接都使用一个短暂或临时端口。可能需要修改这些设置：

* net.ipv4.ip\_local\_port\_range —— 端口值的起止范围。如果你发现用尽端口号，可以增大端口范围。一般端口号设置是1024到65000。

### **调优NGINX配置**

以下是一些可以影响性能的NGINX指令。如上所述，我们只讨论自己能调整的指令。我们建议你在没有NGINX团队指导下，不要调整别的指令。

#### **工作进程**

NGINX可以运行多个工作进程，每个都可处理大量并发连接。可以控制工作进程数，用下面的指令管理它们的连接：

* worker\_processes —— NGINX工作进程数（默认值是1）。在大多数情况下，一个CPU内核运行一个工作进程最好，建议将这个指令设置成自动就可以。有时可能想增大这个值，比如当工作进程需要做大量的磁盘I/O。
* worker\_connections —— 每个工作进程可以处理并发的最大连接数。默认值是512，但多数系统有充足的资源可以支撑更多的连接。合适的设置可以根据服务器的大小和流量的性质决定，可以通过测试修改。

#### **长连接**

长连接对性能有很大的影响，通过减少CPU和网络开销需要开启或关闭连接。NGINX终止所有客户端连接，创建到上游服务器独立的连接。NGINx支持客户端和上游服务器两种长连接。下面是和客户端的长连接相关的指令：

* keepalive\_requests-单个客户端长连接可以请求的数量，默认值是100，但是当使用压力测试工具从一个客户端发送多个请求测试时，这个值设更高些特别有用。
* keepalive\_timeout—空闲长连接保持打开状态的时间。

下面是和上游服务器长连接的相关指令：

* keepalive –每个工作进程中空闲长连接到上游服务器保持开启的连接数量。没有默认值。

要使用连接到上游服务器的长连接，必须要配置文件中下面的指令。

proxy\_http\_version 1.1;

proxy\_set\_header Connection "";

#### **访问日志**

记录每个请求会消耗CPU和I/O周期，一种降低这种影响的方式是缓冲访问日志。使用缓冲，而不是每条日志记录都单独执行写操作，NGINX会缓冲一连串的日志记录，使用单个操作把它们一起写到文件中。

要启用访问日志的缓存，就涉及到在access\_log指令中buffer=size这个参数。当缓冲区达到size值时，NGINX会把缓冲区的内容写到日志中。让NGINX在指定的一段时间后写缓存，就包含flush=time参数。当两个参数都设置了，当下个日志条目超出缓冲区值或者缓冲区中日志条目存留时间超过设定的时间值，NGINX都会将条目写入日志文件。当工作进程重新打开它的日志文件或退出时，也会记录下来。要完全禁用访问日志记录的功能，将access\_log 指令设置成off参数。

#### **Sendfile**

操作系统的sendfile()系统调用可以实现从一个文件描述符到另一个文件描述符的数据拷贝，通常实现零拷贝，这能加速TCP数据传输。要让NGINX使用它，在http或server或location环境中包含sendfile指令。NGINX可以不需要切换到用户态，就把缓存或磁盘上的内容写入套接字 ，而且写的速度非常快，消耗更少的CPU周期。注意，尽管使用sendfile()数据拷贝可以绕过用户态，这不适用于常规的NGINX处理改变内容的链和过滤器， 比如gzip。当配置环境下有sendfile指令和激活内容更改过滤器的指令时，NGINX会自动禁用sendfile。

#### **限制**

你可以设置多个限制，防止用户消耗太多的资源，避免影响系统性能和用户体验及安全。 以下是相关的指令：

* limit\_conn and limit\_conn\_zone—NGINX接受客户连接的数量限制，例如单个IP地址的连接。设置这些指令可以防止单个用户打开太多的连接，消耗超出自己的资源。
* limit\_rate–传输到客户端响应速度的限制（每个打开多个连接的客户消耗更多的带宽）。设置这个限制防止系统过载，确保所有客户端更均匀的服务质量。
* limit\_req and limit\_req\_zone– NGINX处理请求的速度限制，与limit\_rate有相同的功能。可以提高安全性，尤其是对登录页面，通过对用户限制请求速率设置一个合理的值，避免太慢的程序覆盖你的应用请求（比如DDoS攻击)。
* max\_conns上游配置块中服务器指令参数。在上游服务器组中单个服务器可接受最大并发数量。使用这个限制防止上游服务器过载。设置值为0（默认值）表示没有限制。
* queue (NGINX Plus) – 创建一个队列，用来存放在上游服务器中超出他们最大max\_cons限制数量的请求。这个指令可以设置队列请求的最大值，还可以选择设置在错误返回之前最大等待时间（默认值是60秒）。如果忽略这个指令，请求不会放入队列。

### **缓存和压缩可以提高性能**

NGINX的一些额外功能可用于提高Web应用的性能，调优的时候web应用不需要关掉，但值得一提，因为它们的影响可能很重要。 它们包括缓存和压缩。

#### **缓存**

一个启用NGINX缓存的情景，一组web或者应用服务器负载均衡，可以显著缩短对客户端的响应时间，同时大幅度降低后端服务器的负载。缓存本身就可以作个专题来讲，这里我们就不试图讲它了。参阅NGINX Plus管理手册的NGINX内容缓存。

#### **压缩**

所以使用更小的网络带宽。然而尽管压缩数据会消耗CPU资源，但当需要减少网络带宽使用时这样做非常有效。需要注意的是，不能对已压缩的文件再压缩例如JPEG 文件。有关更多的信息，请参阅“NGINX Plus管理指南”中的压缩和解压缩。