Nginx做web服务器**[Linux](http://lib.csdn.net/base/linux" \o "Linux知识库" \t "_blank)**内核参数优化

Nginx提供web服务时Linux内核参数调整是必不可少的，其中在优化方面就需要我们格外的注意。在下面就是对Linux内核参数优化的详细介绍，希望大家有所收获。

关于Linux内核参数的优化：

net.ipv4.tcp\_max\_tw\_buckets = 6000

timewait的数量，默认是180000。

net.ipv4.ip\_local\_port\_range = 1024 65000

允许系统打开的端口范围。

net.ipv4.tcp\_tw\_recycle = 1

启用timewait快速回收。

net.ipv4.tcp\_tw\_reuse = 1

开启重用。允许将TIME-WAIT sockets重新用于新的TCP连接。

net.ipv4.tcp\_syncookies = 1

开启SYN Cookies，当出现SYN等待队列溢出时，启用cookies来处理。

net.core.somaxconn = 262144

web应用中listen函数的backlog默认会给我们内核参数的net.core.somaxconn限制到128，而Nginx内核参数定义的NGX\_LISTEN\_BACKLOG默认为511，所以有必要调整这个值。

net.core.netdev\_max\_backlog = 262144

每个网络接口接收数据包的速率比内核处理这些包的速率快时，允许送到队列的数据包的最大数目。

net.ipv4.tcp\_max\_orphans = 262144

系统中最多有多少个TCP套接字不被关联到任何一个用户文件句柄上。如果超过这个数字，孤儿连接将即刻被复位并打印出警告信息。这个限制仅仅是为了防止简单的DoS攻击，不能过分依靠它或者人为地减小这个值，更应该增加这个值(如果增加了内存之后)。

net.ipv4.tcp\_max\_syn\_backlog = 262144

记录的那些尚未收到客户端确认信息的连接请求的最大值。对于有128M内存的系统而言，缺省值是1024，小内存的系统则是128。

net.ipv4.tcp\_timestamps = 0

时间戳可以避免序列号的卷绕。一个1Gbps的链路肯定会遇到以前用过的序列号。时间戳能够让内核接受这种“异常”的数据包。这里需要将其关掉。

net.ipv4.tcp\_synack\_retries = 1

为了打开对端的连接，内核需要发送一个SYN并附带一个回应前面一个SYN的ACK。也就是所谓三次握手中的第二次握手。这个设置决定了内核放弃连接之前发送SYN+ACK包的数量。

net.ipv4.tcp\_syn\_retries = 1

在内核放弃建立连接之前发送SYN包的数量。

net.ipv4.tcp\_fin\_timeout = 1

如果套接字由本端要求关闭，这个参数决定了它保持在FIN-WAIT-2状态的时间。对端可以出错并永远不关闭连接，甚至意外当机。缺省值是60秒。2.2 内核的通常值是180秒，你可以按这个设置，但要记住的是，即使你的机器是一个轻载的WEB服务器，也有因为大量的死套接字而内存溢出的风险，FIN- WAIT-2的危险性比FIN-WAIT-1要小，因为它最多只能吃掉1.5K内存，但是它们的生存期长些。

net.ipv4.tcp\_keepalive\_time = 30

当keepalive起用的时候，TCP发送keepalive消息的频度。缺省是2小时。