\*tcpsyncookies

是一个开关，是否打开SYN Cookie功能，该功能可以防止部分SYN×××。

tcpsynackretries和tcpsynretries定义SYN的重试次数。

YN Cookie是对TCP服务器端的三次握手做一些修改，专门用来防范SYN Flood×××的一种手段。它的原理是，在TCP服务器

接收到TCP SYN包并返回TCP SYN + ACK包时，不分配一个专门的数据区，而是根据这个SYN包计算出一个cookie值。这个

cookie作为将要返回的SYN ACK包的初始序列号。当客户端返回一个ACK包时，根据包头信息计算cookie，与返回的确认序列

号(初始序列号 + 1)进行对比，如果相同，则是一个正常连接，然后，分配资源，建立连接。

原理：在Tcp服务器收到Tcp Syn包并返回Tcp Syn+ack包时，不专门分配一个数据区，而是根据这个Syn包计算出一个cookie值。在收到Tcp ack包时，Tcp服务器在根据那个cookie值检查这个Tcp ack包的合法性。如果合法，再分配专门的数据区进行处理未来的TCP连接。

默认为0，1表示开启

net.ipv4.tcpfintimeout

修改timewait状的存在时间，默认的2MSL

注意：timewait存在且生存时间为2MSL是有原因的，见我上一篇博客为什么会有timewait状态的存在，所以修改它有一定的风险，还是根据具体的情况来分析。

tcpretries1

放弃回应一个TCP连接请求前﹐需要进行多少次重试。RFC 规定最低的数值是3﹐这也是默认值

tcpretries2

TCP失败重传次数,默认值15,意味着重传15次才彻底放弃.可减少到5,以尽早释放内核资源.

---------------------------------------------------------------------------------------------------------------------------------------------------------------

nf\_conntrack

该模块在kernel 2.6.15（2006-01-03发布） 被引入，支持ipv4和ipv6，取代只支持ipv4的ip\_connktrack，用于跟踪连接的状态，供其他模块使用。

最常见的使用场景是 iptables 的 nat 和 state 模块：

nat 根据转发规则修改IP包的源/目标地址，靠nf\_conntrack的记录才能让返回的包能路由到发请求的机器。

state 直接用 nf\_conntrack 记录的连接状态（NEW/ESTABLISHED/RELATED/INVALID）来匹配防火墙过滤规则。

在服务器访问量大时，如果内核netfilter模块conntrack相关参数配置不合理，就会导致新连接被drop掉

推荐bucket至少 262144，max至少 1048576，不够再继续加。

net.netfilter.nf\_conntrack\_count 的数字持续超过 nf\_conntrack\_max 的 20% 就该考虑调高上限了；

测试没问题后可以写入配置文件 vim /etc/sysctl.d/90-conntrack.conf :

net.netfilter.nf\_conntrack\_buckets = 262144

net.netfilter.nf\_conntrack\_max=1048576

net.nf\_conntrack\_max=1048576

net.netfilter.nf\_conntrack\_tcp\_timeout\_fin\_wait=30

net.netfilter.nf\_conntrack\_tcp\_timeout\_time\_wait=30

net.netfilter.nf\_conntrack\_tcp\_timeout\_close\_wait=15

net.netfilter.nf\_conntrack\_tcp\_timeout\_established=300#########################################################################################

如何修改内核参数:::

Linux在系统运行时修改内核参数(/proc/sys与/etc/sysctl.conf)，而不需要重新引导系统，这个功能是通过/proc虚拟文件系统实现的。

在/proc/sys目录下存放着大多数的内核参数，并且设计成可以在系统运行的同时进行更改, 可以通过更改/proc/sys中内核参数对应的文件达到修改内核参数的目的(修改过后，保存配置文件就马上自动生效)，不过重新启动机器后之前修改的参数值会失效，所以只能是一种临时参数变更方案。(适合调试内核参数优化值的时候使用，如果设置值有问题，重启服务器还原原来的设置参数值了。简单方便。)

但是如果调试内核参数优化值结束后，需要永久保存参数值，就要通过修改/etc/sysctl.conf内的内核参数来永久保存更改。但只是修改sysctl文件内的参数值，确认保存修改文件后，设定的参数值并不会马上生效，如果想使参数值修改马上生效，并且不重启服务器，可以执行下面的命令：#sysctl –p

常用的内核参数：

net.ipv4.tcpsyncookies = 1#表示开启SYN Cookies。当出现SYN等待队列溢出时，启用cookies来处理，可防范少量SYN×××，默认为0，表示关闭；

net.ipv4.tcptwreuse = 1#表示开启重用。允许将TIME-WAIT sockets重新用于新的TCP连接，默认为0，表示关闭；

net.ipv4.tcptwrecycle = 1#表示开启TCP连接中TIME-WAIT sockets的快速回收，默认为0，表示关闭。

net.ipv4.tcpfintimeout = 30#表示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。

net.ipv4.tcpkeepalivetime = 1200#表示当keepalive起用的时候，TCP发送keepalive消息的频度。缺省是2小时，改为20分钟。

net.ipv4.iplocalportrange = 1024 65000#表示用于向外连接的端口范围。缺省情况下很小：32768到61000，改为1024到65000。

net.ipv4.tcpmaxtwbuckets = 5000#表示系统同时保持TIMEWAIT套接字的最大数量，如果超过这个数字，#TIMEWAIT套接字将立刻被清除并打印警告信息。默认为180000，改为5000。#对于Apache、Nginx等服务器，上几行的参数可以很好地减少TIMEWAIT套接字数量，#但是对于Squid，效果却不大。此项参数可以控制TIMEWAIT套接字的最大数量，避免Squid服务器被大量的TIMEWAIT套接字拖死。\*

-------------------------------------------------------------------------------------------------------------------------------------------------------------

如何尽量处理TIMEWAIT过多

sysctl改两个内核参数就行了，如下：

net.ipv4.tcp\_tw\_reuse = 1

net.ipv4.tcp\_tw\_recycle = 1

简单来说，就是打开系统的TIMEWAIT重用和快速回收，至于怎么重用和快速回收，这个问题我没有深究，实际场景中这么做确实有效果。用netstat或者ss观察就能得出结论。

还有些朋友同时也会打开syncookies这个功能，如下：

net.ipv4.tcp\_syncookies = 1

打开这个syncookies的目的实际上是：“在服务器资源（并非单指端口资源，拒绝服务有很多种资源不足的情况）不足的情况下，尽量不要拒绝TCP的syn（连接）请求，尽量把syn请求缓存起来，留着过会儿有能力的时候处理这些TCP的连接请求”。

如果并发量真的非常非常高，打开这个其实用处不大。