**C100K下内核调优**

**1、文件句柄数量受限**

**1.1、单一进程的文件句柄数量受限**

当前用户每个进程最多允许同时打开1024个文件，除去每个进程必然打开的标准输入、标准输出、标准错误、服务器监听socket、进程间通讯的unix或socket等文件，剩下的可用于客户端socket连接的文件数就只有大概1024-10=1014个左右。也就是说默认情况下，基于Linux的通信程序最多同时允许1014个TCP并发连接

想要支持更高数量的TCP并发连接的通讯处理程序，就必须修改Linux对当前用户的进程可同时打开的文件数量的软限制(soft limit)和硬限制(hard limit)

**软限制**是指Linux在当前系统能承受的范围内进一步限制用户能同时打开的文件数

**硬限制**是指根据系统硬件资源状况(主要是系统内存)计算出来的系统最多可同时打开的文件数量

/etc/security/limits.conf

**\* soft nofile 1048576**

**\* hard nofile 1048576**

在linux kernel 2.6.25之前通过ulimit -n(setrlimit(RLIMIT\_NOFILE))设置每个进程的最大打开文件句柄数不能超过NR\_OPEN（1024\*1024）,也就是100多w（除非重新编译内核），而在25之后，内核导出了一个sys接口可以修改这个最大值（/proc/sys/fs /nr\_open）

**1.2、整个系统的文件句柄数量受限**

cat /proc/sys/fs/file-max

file-max表示系统所有进程最多允许同时打开的文件句柄数，是Linux系统级硬件限制。通常，这个系统硬限制是Linux系统在启动时根据系统硬件资源状况计算出来的最佳的最大同时打开文件数限制，如果没有特殊需求，不应该修改次限制

要修改它，需要对 /etc/sysctl.conf 文件，增加一行内容：

fs.file-max = 1048576

**2、端口数量受限(服务器一般不调)**

一般来说，对外提供请求的服务端不用考虑端口数量问题，只要监听某一个端口即可。可客户端要模拟大量的用户对服务器发起TCP请求，而每一个请求都需要一个端口，为了使一个客户端尽可能地模拟更多的用户，也就要使用客户端拥有更多可使用的端口

cat /proc/sys/net/ipv4/ip\_local\_port\_range

修改 /etc/sysctl.conf 文件，增加一行内容

**net.ipv4.ip\_local\_port\_range = 1024 65535**

**3、TCP参数调优**

要想提高服务端的性能，以达到完美高并发的目的，需要对系统的TCP参数进行适当的修改优化

**net.ipv4.tcp\_tw\_reuse = 1**

当服务器需要大量TCP连接之间切换时，会产生大量处于TIME\_WAIT状态的连接。TIEM\_WAIT意味着连接本身是关闭的，但是资源还没有释放。将net.ipv4.tcp\_tw\_reuse设置为1是让内核在安全时尽量回收连接，这比重新建立新连接要便宜很多

**net.ipv4.tcp\_fin\_timeout = 15**

这是处于TIEM\_WAIT状态的连接在回收前必须等待的最小数据。改小它可以加快回收

**Net.core.rmem\_max = 16777216**

**Net.core.wmem\_max = 16777216**

提高TCP的最大缓冲区大小，其中:

Net.core.rmem\_max：表示接受套接字缓冲区大小的最大值(以字节为单位)

Net.core.wmem\_max：表示发送套接字缓冲区大小的最大值(以字节为单位)

**net.ipv4.tcp\_rmem = 4096 87380 16777216**

**net.ipv4.tcp\_wmem = 4096 65536 16777216**

提高Linux内核自动对socket缓冲区进行优化的能力，其中：

net.ipv4.tcp\_rmem：用来配置读缓冲的大小，第1个值为最小值，第2个值为默认值，第3个值为最大值。

net.ipv4.tcp\_wmem：用来配置写缓冲的大小，第1个值为最小值，第2个值为默认值，第3个值为最大值。

**net.core.somaxconn = 4096**

每个网络接口接受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。默认为1000

**net.core.netdev\_max\_backlog = 4096**

表示socket监听(listen)的backlog上限。Backlog是socket的监听队列，当一个请求尚未被处理或建立时，他会进入backlog。而socket server可以一次性处理backlog中的所有请求，处理后的请求不再位于监听队列中

**net.ipv4.tcp\_max\_syn\_backlog = 20480**

表示SYN队列的长度，默认为1024，，加大队列长度是8192，可以容纳更多等待连接的网络连接数

**net.ipv4.tcp\_syncookies = 1**

表示开启SYN Cookies，当出现SYN等待队列溢出时，启用cookies来处理，可以防范少量SYN攻击，默认为0，表示关闭

**net.ipv4.tcp\_max\_tw\_buckets = 360000**

表示系统同时保持TIME\_WAIT套接字的最大数量，如果超过这个数字，TIME\_WAIT套接字将立刻被清除并打印警告信息。默认180000

**net.ipv4.tcp\_no\_metrics\_save = 1**

一个TCP连接关闭后，把这个连接曾经有的参数比如慢启动门snd\_sthresh、拥塞窗口snd\_cwnd，还有srtt等信息保存到dst\_entry中，只要dsr\_entry没有失效，下次新建立相同连接的时候就可以使用保存的参数来初始化这个连接

**net.ipv4.tcp\_syn\_retries = 2**

表示在内核放弃建立新连接之前发生SYN包的数量，默认为4

**net.ipv4.tcp\_synack\_retries = 2**

表示在内核放弃内核连接之前发生SYN+ACK包的数量，默认为5

完整的TCP参数调优配置如下所示：

net.ipv4.tcp\_tw\_reuse = 1

net.ipv4.tcp\_fin\_timeout = 15

net.core.rmem\_max = 16777216

net.core.wmem\_max = 16777216

net.ipv4.tcp\_rmem = 4096 87380 16777216

net.ipv4.tcp\_wmem = 4096 65536 16777216

net.core.netdev\_max\_backlog = 4096

net.core.somaxconn = 4096

net.ipv4.tcp\_max\_syn\_backlog = 20480

net.ipv4.tcp\_syncookies = 1

net.ipv4.tcp\_max\_tw\_buckets = 360000

net.ipv4.tcp\_no\_metrics\_save = 1

net.ipv4.tcp\_syn\_retries = 2

net.ipv4.tcp\_synack\_retries = 2

其他一些参数

**vm.min\_free\_kbytes = 65536**

确定系统开始回收内存的阀值，控制系统的空闲内存。值越高，内核越早开始回收内存，空闲内存越高

**vm.swappiness = 0**

控制内核从物理内存移出进程，移到交换内存。该参数从0到100，当该参数=0，表示只要有可能就尽量避免交换进程移出物理内存；该参数=100，这告诉内核疯狂的将数据移出物理内存移到swap缓存中

Linux内核调优主要参数

修改进程打开文件句柄数

开启重用

TIME\_WAIT时间，允许将TIME\_WAIT socket用于新的TCP连接

调整TCP最大缓冲区

套接字缓冲区大小

允许送到队列的数据包的最大数目

SYN队列长度

SYNcookies

加快网络性能的协议

控制内核从物理内存移出进程，移到交换内存

内存管理