

提高Linux上socket性能Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E6_8F_90_E9_AB_98Linu_c103_645051.htm 在开发 socket 应用程序时，首要任务通常是确保可靠性并满足一些特定的需求。利用本文中给出的 4 个提示，您就可以从头开始为实现最佳性能来设计并开发 socket 程序。本文内容包括对于 Sockets API 的使用、两个可以提高性能的 socket 选项以及 GNU/Linux 优化。为了能够开发性能卓越的应用程序，请遵循以下技巧：

- * 最小化报文传输的延时。
- * 最小化系统调用的负载。
- * 为 Bandwidth Delay Product 调节 TCP 窗口。
- * 动态优化 GNU/Linux TCP/IP 栈。

技巧 1. 最小化报文传输的延时 在通过 TCP socket 进行通信时，数据都拆分成了数据块，这样它们就可以封装到给定连接的 TCP payload（指 TCP 数据包中的有效负荷）中了。TCP payload 的大小取决于几个因素（例如最大报文长度和路径），但是这些因素在连接发起时都是已知的。为了达到最好的性能，我们的目标是使用尽可能多的可用数据来填充每个报文。当没有足够的数来填充 payload 时（也称为最大报文段长度（maximum segment size）或 MSS），TCP 就会采用 Nagle 算法自动将一些小的缓冲区连接到一个报文段中。这样可以通过最小化所发送的报文的数量来提高应用程序的效率，并减轻整体的网络拥塞问题。尽管 John Nagle 的算法可以通过将这些数据连接成更大的报文来最小化所发送的报文的数量，但是有时您可能希望只发送一些较小的报文。一个简单的例子是 telnet 程序，它让用户可以与远程系统进行交互，这通常都是通过一个 shell 来进行的。如

果用户被要求用发送报文之前输入的字符来填充某个报文段，那么这种方法就绝对不能满足我们的需要。另外一个例子是 HTTP 协议。通常，客户机浏览器会产生一个小请求（一条 HTTP 请求消息），然后 Web 服务器就会返回一个更大的响应（Web 页面）。解决方案您应该考虑的第一件事情是 Nagle 算法满足一种需求。由于这种算法对数据进行合并，试图构成一个完整的 TCP 报文段，因此它会引入一些延时。但是这种算法可以最小化在线路上发送的报文的数量，因此可以最小化网络拥塞的问题。但是在需要最小化传输延时的情况中，Sockets API 可以提供一种解决方案。要禁用 Nagle 算法，您可以设置 TCP_NODELAY socket 选项，如清单 1 所示。

```
清单 1. 为 TCP socket 禁用 Nagle 算法
int sock, flag, ret; /* Create
new stream socket */
sock = socket( AF_INET, SOCK_STREAM, 0 );
/* Disable the Nagle (TCP No Delay) algorithm */
flag = 1;
ret = setsockopt( sock, IPPROTO_TCP, TCP_NODELAY, (char
*)&flag, sizeof(flag) );
if (ret == -1) { printf("Couldnt
setsockopt(TCP_NODELAY)\n"); exit(-1); }
```

提示：使用 Samba 的实验表明，在从 Microsoft? Windows? 服务器上的 Samba 驱动器上读取数据时，禁用 Nagle 算法几乎可以加倍提高读性能。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com