**大文件传输实验报告**

辛云星 李 任

卫晓滨 杨 浩

1. 实验目的
2. 通过实验熟悉windows socket编程；
3. 掌握利用TCP实现文件传输的方法；
4. 深入TCP协议原理，加速传输效率，保证传输可靠。
5. 实验思路
6. 保证传输可靠性

TCP协议保证了端到端的正确传输，但是并不能保证应用程序之间的正确传输，造成应用程序收发错误的情况有右二：

* 1. 网络紊乱

此时发送方发送的数据到不了接收方，会相应地收到网络错误或超时信息，而接收方则一直处于等待状态，白白浪费资源。解决方案是在程序中及时响应各种错误返回，并设置心跳机制，用来监听连接状态。

* 1. 主机或程序崩溃

此问题主要发生在最后一个包，发送方发送完最后一个包后立即返回，若此时接收方宕机，则接收不到这个包，而发送方因为早已返回也无从得知。解决方案是有序、安全断开连接，并在应用层对最后一个包回复确认。

1. 提高传输速度

可从操作系统和协议两方面提高传输速度。

* 1. 操作系统

套接字收发请求均通过系统调用实现，而系统调用会引进程发上下文切换、页面调度等开销，所以要尽量减少系统调用。

* 1. TCP协议

TCP协议有流控和Nagel机制，通过调节参数，可使传输效率最大化。同时，也可用多线程和内存映射文件方法加快传输。

1. 实验过程
2. 顺序关闭连接

关闭顺序由：

closesocket（）

变为：

shutdown（）

closesocket（）

closesocket不管TCP缓冲区还有没有待发送数据，直接返回；而shutdown先将TCP缓存的数据发送，再发送fin，告诉接收方已无数据发送，但仍可接受数据。这样当接收方也用shutdown表示不再发送后，连接可以正确断开。

1. 应用层确认

发送方调用shutdown后，阻塞等待接收方的fin，所接受到则比表示文件数据传输完整，否则报错。

1. 心跳机制

建立一个单独线程，保持收发双方的TCP连接。如果网络紊乱或主机崩溃或进程崩溃，则TCP双方能及时反馈，进而断开文件传输，避免空等。

1. 调节TCP buffer和Process buffer大小

这两个参数是减少操作系统调用的关键，可以通过setsockopt设置。测试多组数据，找到最优参数。

1. Nagel

Nagel具体指捎带、延迟确认机制，是为减少网络碎片、减少网络拥塞设计。但由于程序中发送方只负责发，接收方只负责收，所以没有捎带的机会，Nagel只会带来无谓开销，应当取消该机制。

1. 多线程

多线程收发可以加速文件传输效率。

1. 内存映射文件

大文件IO极大影响传输效率，最好能用内存映射方式代替之。该程序使用文件IO方式。

1. 断点续传

收发方各维持一个log文件，记录个线程传输到的位置。当连接断开又恢复时，继续从该位置开始传输。该程序未实现此功能。

1. 测试环境

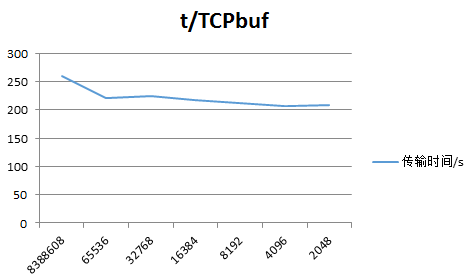
Mingw64

Win10 64bit 4GB HDD5400rpm

Loopback

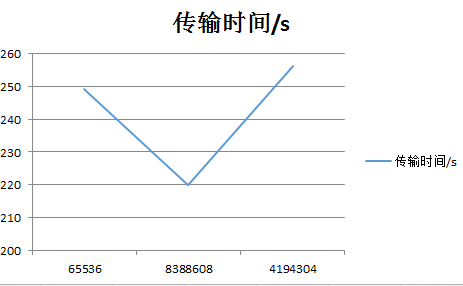
文件6,828,792,465 B（~6.8G）

1. 测试结果
2. Process buffer = 8M，查看传输时间与TCP buffer的关系。



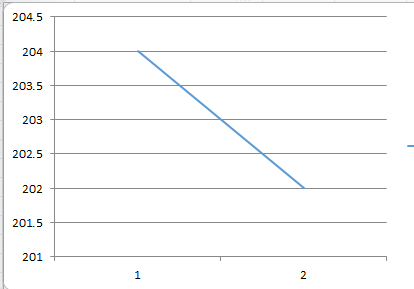
看到当TCP buffer大于64K时传输时间较大，越小效率越高，最后基本保持稳定。

1. TCP buffer = 64K，查看传输时间与Process buffer的关系。



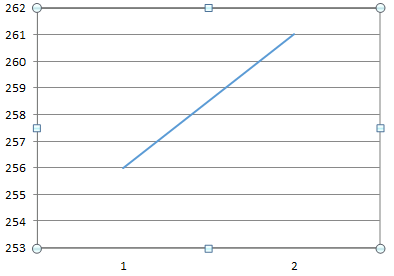
看到，当Process buffer = 8M时，传输时间最低。通过前两个实验可以认为（TCP buffer，Process buffer） = （64K，8M）是文件传输最快的两个参数，此时传输速度是32M/s。而5400rpm HDD传输速度为60-90M/s，为什么没达到极限速度呢，很可能是loopback的带宽和大文件IO所限。

1. Nagel



横坐标=1表示使用Nagel，2表示不用Nagel。两者差距2s，用不用Nagel对程序基本没影响。为什么呢？推测还是受loopback带宽和大文件IO所限。

1. 多线程



横坐标表示线程数量。由一线程增至两线程，传输时间增加5s，若继续增加线程数目，则传输时间增加更多。这是因为此时限制程序传输效率的并不是程序收发速度，而是loopback带宽和大文件IO。而每个线程都操作一个大文件IO，势必会对程序效率造成影响。