

PA2 - Report

Tom King

To run the program, simply run make and then use ./client followed by needed flags

The flags are -d (datarequest) -p patientnum -t time -e ecgno -f (file name) -m message size -n (new channel)

I did all my testing on the linux tamu servers.

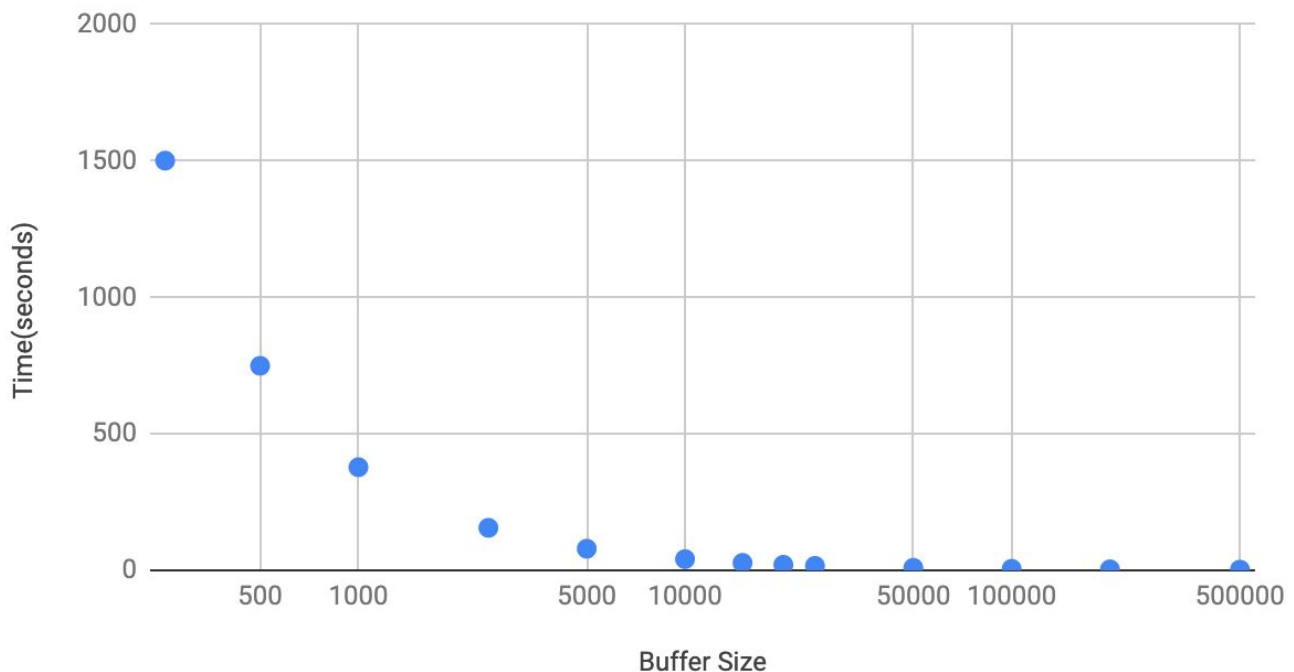
The d flag transfers the entire patient file specified (default 1), times it and compares with the original. Similarly, the f flag transfers the specified file in binary form, times it and compares with the original. The program starts off by forking to call data server and sends quit message to ensure a safe exit after it is done. The program also creates a new channel on the c flag and tests it out.

Timing data: 1.csv took 77.16 seconds to transfer on the linux.tamu.edu server.

y1.csv (1 MB) created with truncate -s command took 15.17 seconds to be transferred at the base 256 byte buffer size.

I designed it very simply. There are three main modes, data transfer, file transfer and a new channel mode. I also modified dataserver and fiforequest channel to support the bonus part of the assignment which is to find out how to stop the bottlenecking of the file transfer. The answer is the buffer size and as you can see in the chart below, it was very effective. I transferred a 100 MB file created with truncate and it was very slow at first but it got fast really quickly. I increased the buffer starting at 256 to see the results and it's almost linear proving that the buffer size was the bottleneck.

Time(seconds) vs. Log(Buffer Size)



Buffer Size	Time(seconds)
256	1501.02
500	749.23
1000	377.585
2500	155.54
5000	78.94
10000	40.86
15000	27.05
20000	19.98
25000	16.12
50000	8.93
100000	5.28
200000	3.487
500000	2.32