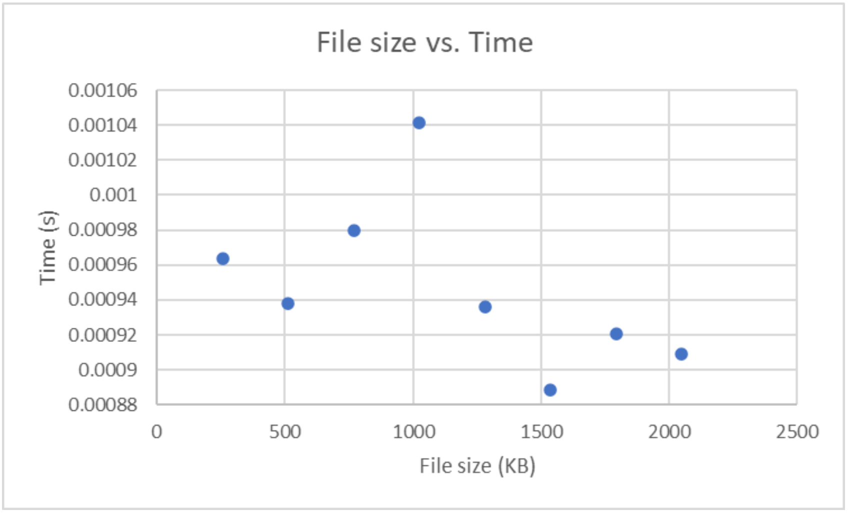
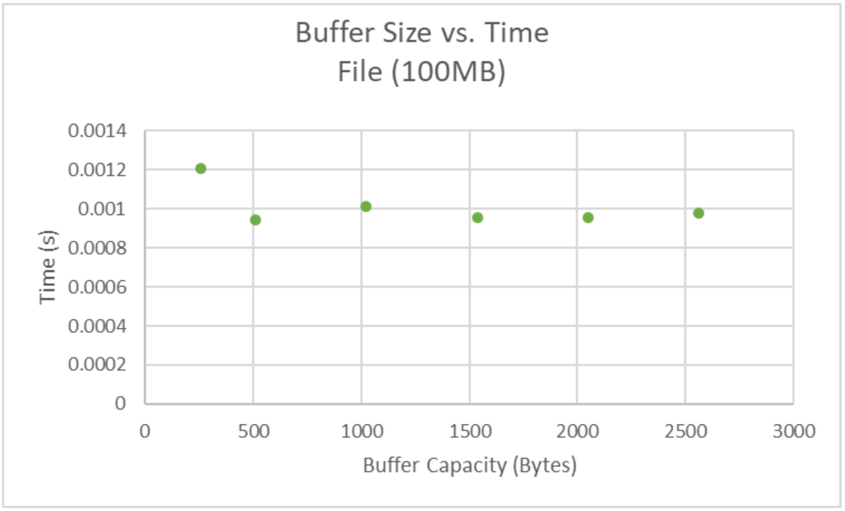
Cole McAnelly
PA1

I designed my client program in an easy-to-read manner, I separated each functionality required by the assignment into its own function, sometimes even using sub-functions to break up the repeated code so that I didn't have to copy and paste/write code twice. Then I wrote my own bash script to collect the execution times of the client program and write them to a csv file so that I could more easily create a graph. As you can see in the graphs of "File size vs. Time" and "Buffer Size vs. Time", there does not seem to be much corrilation between file size and time, but I believe this to be because the CPU scheduler would move my client program to the background while it waited, and did not count that toward execution time. In reality, the biggest bottleneck is the buffer capacity, and be increasing it we should see a decrease in operation time. I saw almost no noticable difference while transferring binary files and the csv files, and they were almost as fast as the individual point requests, which must be because instead of access the data in the file (like it would for a point), the server can just send the file contents with no processing (because its sending a binary). By far the longest operation was the gathering of 1000 data points, and I believe it to be because we had to request 2000 data points, which took massive amounts of time.


File size vs. Time
Time (s)
File size (KB)


Buffer Size vs. Time
File (100MB)
Time (s)
Buffer Capacity (Bytes)