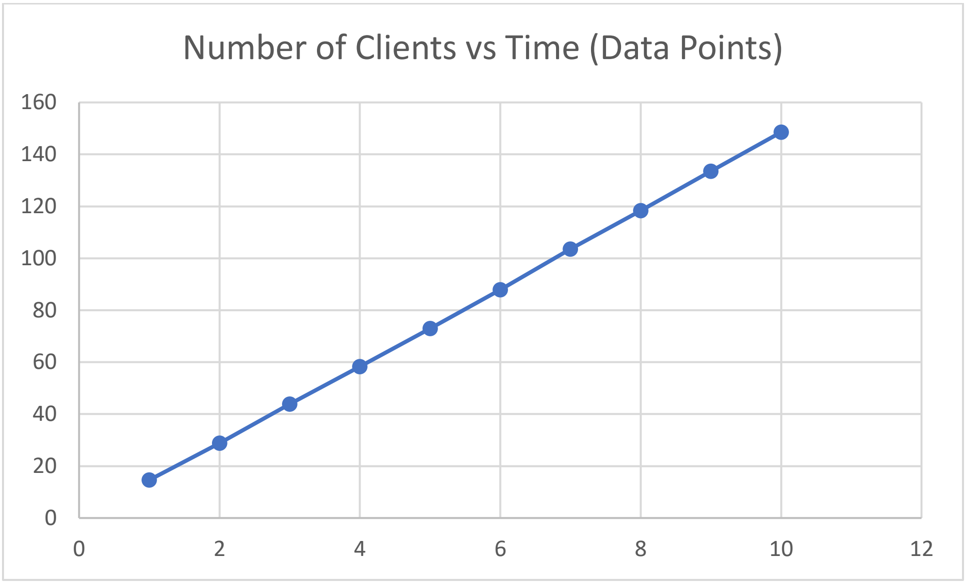
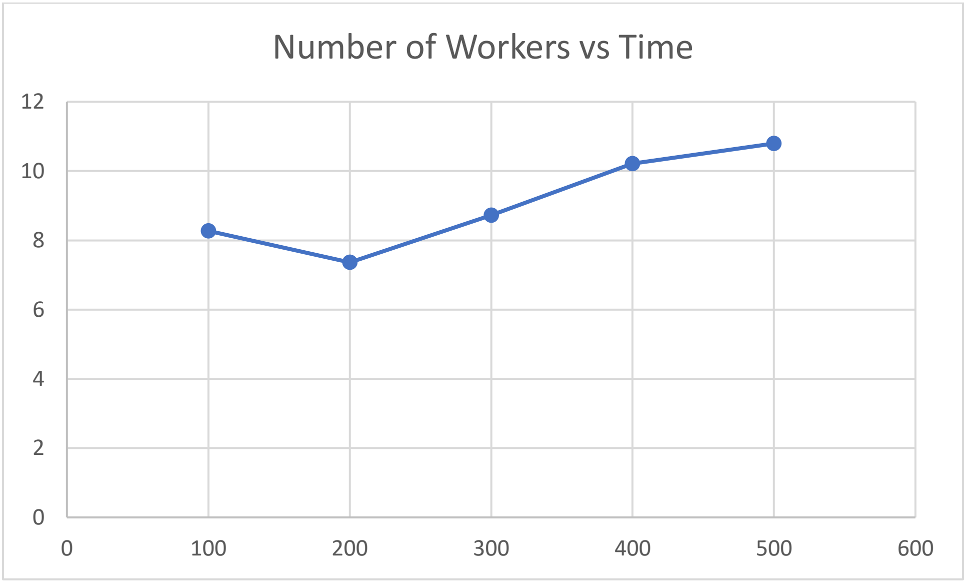
Sean Neal
426006716
11/24/2019

PA5: The Data Server Moved Out

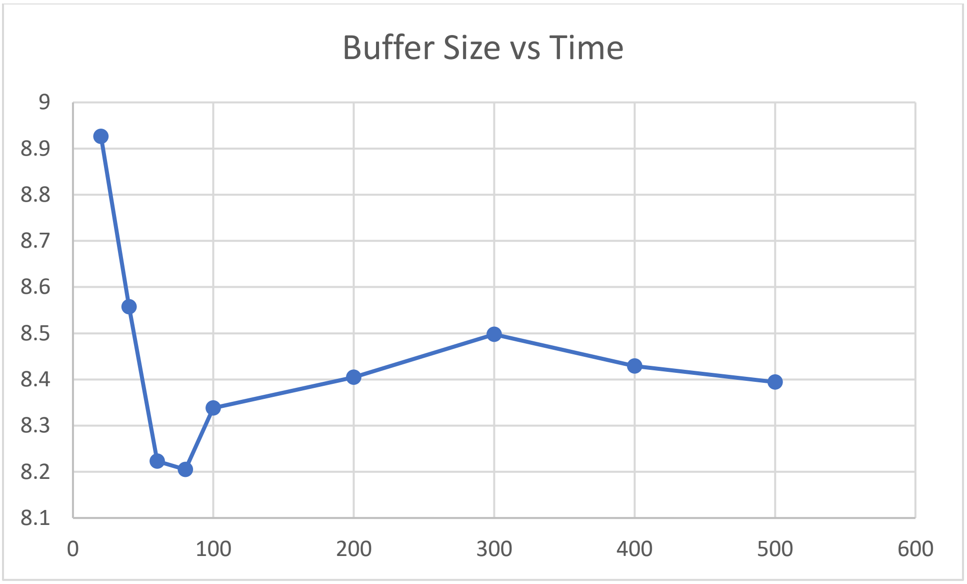


When testing the time it took for various numbers of clients to use the server together, I used a baseline of 15000 data points for 15 clients, using 50 workers and a buffer size of 100. From my tests of 1 through 10 clients, I saw a linear increase in time it took to process the requests of the clients. It seemed consistent with requiring roughly 15 more seconds per client of the server.

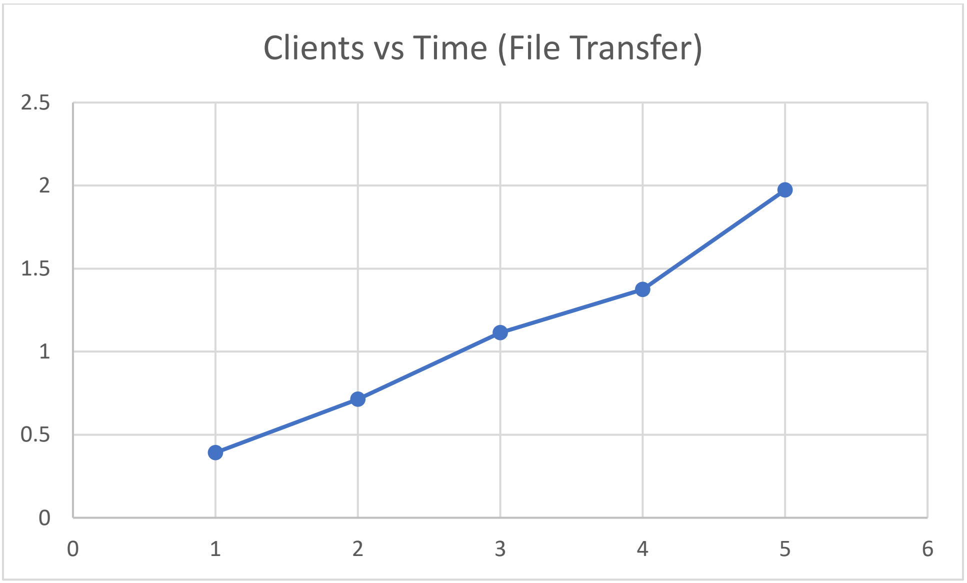


When testing the time it took for various numbers of workers, I used only 1 client at a time. I noticed that up to around 200 workers, the addition of workers saved significant time in processing the data point requests. However, at 300 workers and beyond, the addition of more workers only added time to the execution of the program. I believe this drop off in performance enhancement is caused due to the increase in context switches. Each thread forces the CPU to do more context switches, eating up valuable time. There becomes a point where the workers do not do enough work to save time over the additional overhead they produce from context switches. In the graph above, at around 200 workers the increases in the overhead cost is too great to be considered an efficient use of the CPU.

Sean Neal
426006716
11/24/2019



Increasing the buffer capacity linearly lowers the time taken for reducing the context switches needed. By allowing the threads to carry more information, the need for context switches drops and the CPU can be used much more efficiently. However, once the buffer is over around 100, increasing it does not affect runtime as dramatically. This is because the buffer, once past this cut off point, is no longer the limiting factor.



As seen in the graph for multiple clients requesting data points, the increase in time it took for multiple clients to request a full file also increased linearly as the number of clients went up. This is to be expected as the number of clients increases linearly, and thus the number of requests on the server also increases linearly.

Sean Neal
426006716
11/24/2019

There is a slowdown in the tests between TCP/IP when compared to the times these tests took when running on FIFO. This is due to the fact that FIFO has a much simpler protocol stack when compared to TCP/IP. TCP/IP has layers to it, which the information must go through to be sent to the client or server. While FIFO is simply first in first out for the information, TCP/IP requires significantly more overhead per information packet, resulting in the longer execution times.