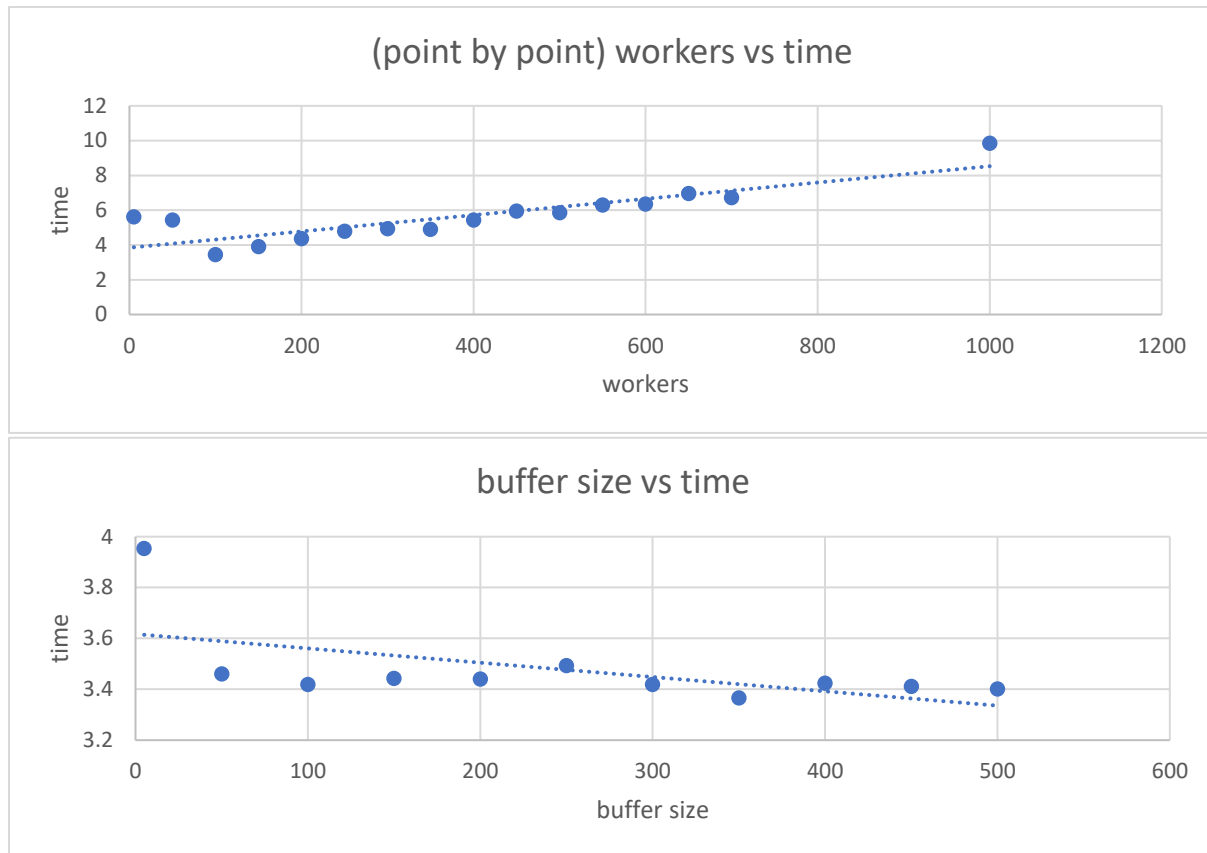
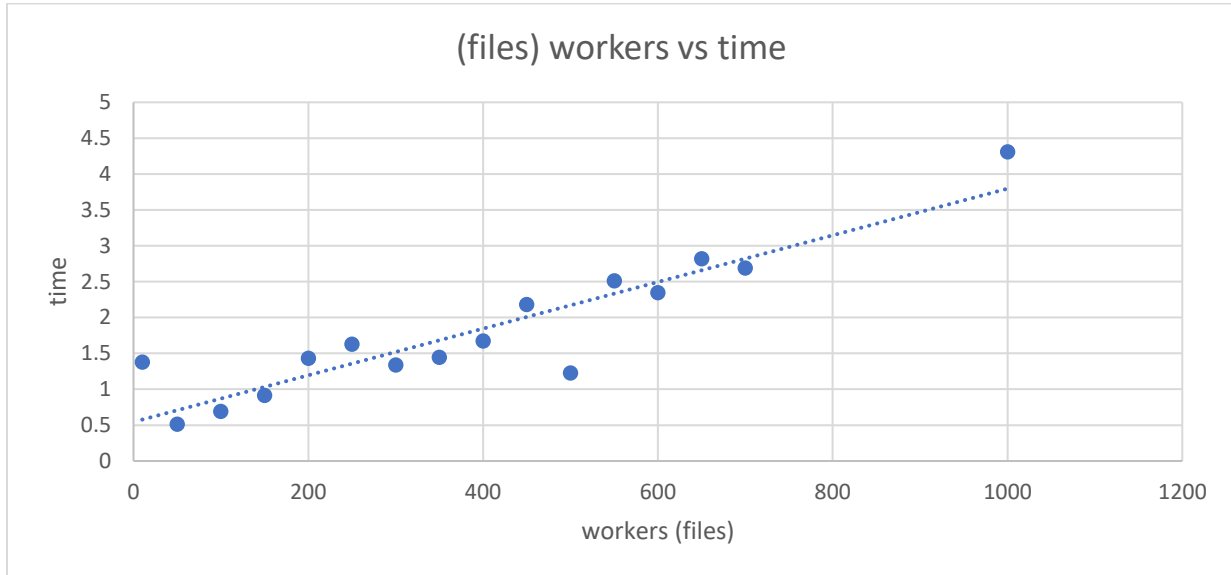


PA3 Threading and Synchronization



My data comparing how the buffer size and the number of workers affects the collection of data in a point by point manner used 5 patients, 15000 data points, and 100 for either the buffer size or workers, depending on which was not being tested. First, looking at the time taken by the different amounts of workers, it seems having more worker threads increases the time taken by the program to collect the data in a linear fashion. This is only after a certain point, as in this case the benefit of multiple workers can be seen until the number of workers exceeds around 100. While initially counter-intuitive, 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.

This contrasts the time saved by increasing the buffer size capacity. Increasing the 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.



Similarly, to the point by point workers vs time graph, the increase in workers increases the time taken by the program to finish a transfer. Overhead is once again an issue, which each increase in worker threads also coming with more needed context switches adding to the overall runtime. This can lead to inefficient use of the CPU after a certain threshold specific to the run conditions. In the graph above, this point can be seen around 100, where more workers than that increases the overhead too greatly to be considered efficient use of the CPU.