

Arturo Villalobos

CSCE 313-503

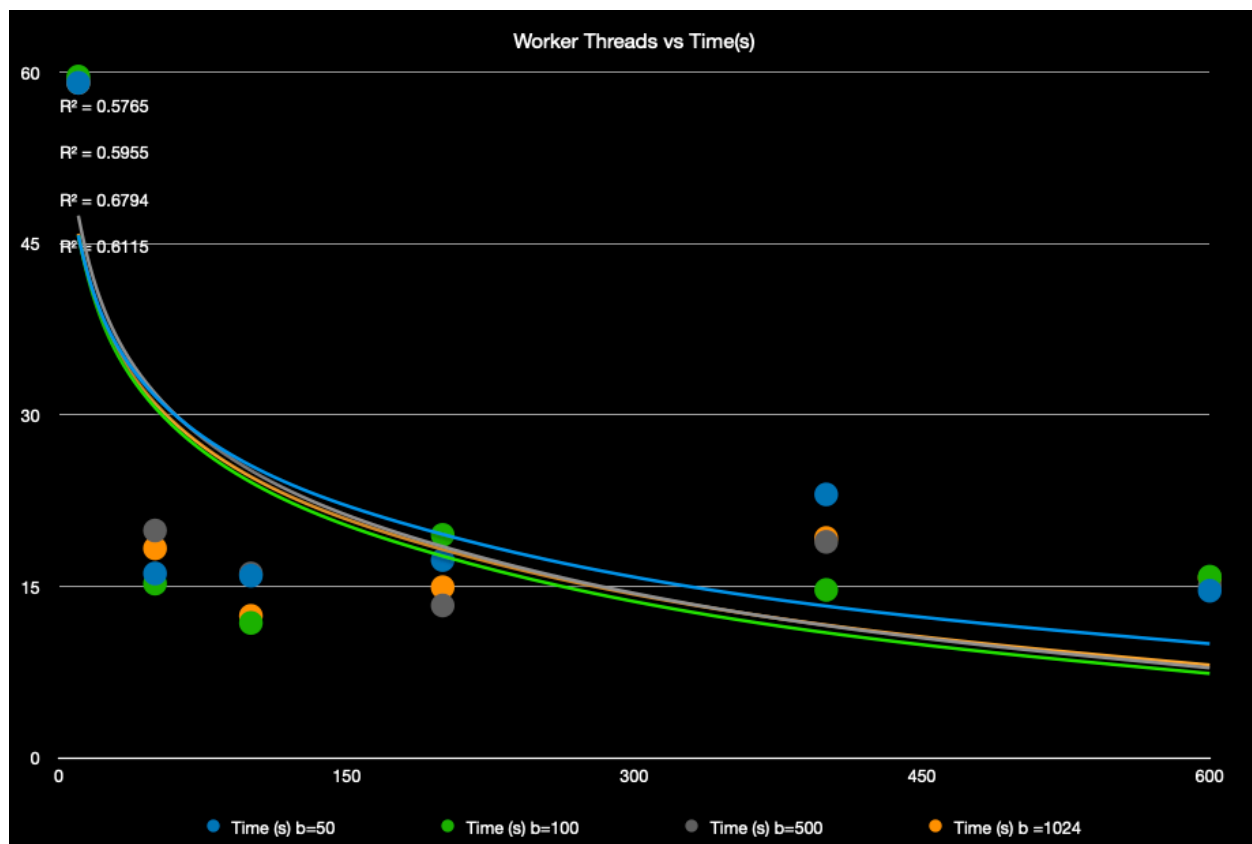
Professor Ahmed

4-8-21

PA4

Video Link: <https://drive.google.com/drive/folders/1VijKo2zrNHKLStKWkmt74ZBKsS8Cz53e?usp=sharing>

When considering the making histograms within our programs the most obvious parameters that would speed up the application are the amount of patients we get data on , and the amount of worker threads we make. After testing, I decided to make these charts with -p being set to all 15 patients , -n being set to 15000 , and varied the buffer capacity and the amount of threads. The resulting graph shows that at 100 worker threads , regardless of buffer capacity, we find the greatest increase in performance. Buffer capacity does not seem to increase or decrease the performance in any significant way. At 400 worker threads , we see a decrease in performance across the board and this would be the point of diminishing returns, meaning the overhead caused by context switching significantly impacts the performance. Graph shown below:



File Transfers are a different story altogether. We already saw that in PA1 we found that an increase in message buffer capacity increases performance. I decided to set m to 256 across all tests, and instead focused on worker thread count and request buffer capacity parameters. The graph again shows that there is no significant difference in performance across different request buffer values, however there is an inverse relationship between performance and worker thread count. As the amount of threads increases, the amount of time taken for the program to run also increases (lower performance). This can be seen immediately after only adding 50 threads. The fact is that the context switch overhead outweighs any performance gained by having more threads. In fact, a file transfer with 1 worker thread, all other parameters kept the same, is faster than all other values of worker thread. Graph shown below:

We can see the same chart again but an increase in message buffer capacity to different values and request buffer capacity being set the same to 100. While there is now some distinction between different tests with lower m values being across the board lower performing, the context switch overhead outweighs any benefits gained from higher m values. Graph shown below:

