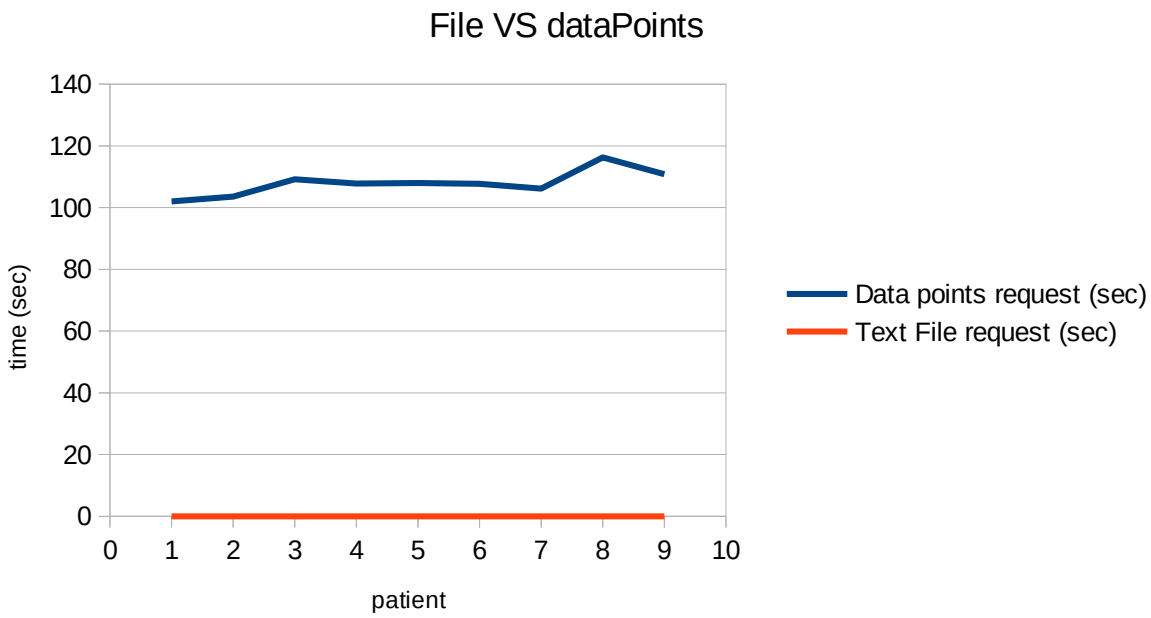
Mauro Lopez

Threading and Synchronization

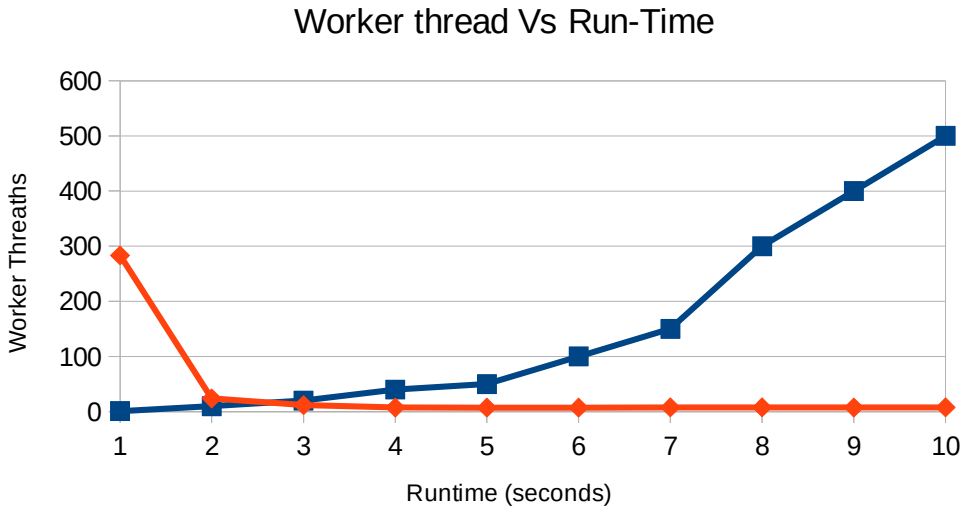
Introduction:

In this programming assignment we begin to improve on our initial design created in PA2. We saw that by looking at the graphs, we can see we bottle neck at some points of the program. This happen, when we asked for individual data points for a patient. We were only using, one channel, thus leading to a huge run time. With threading, we can see it help raise the speed. This was due to the fact that our process was running in parallel, and each data point, was getting collected simultaneously.

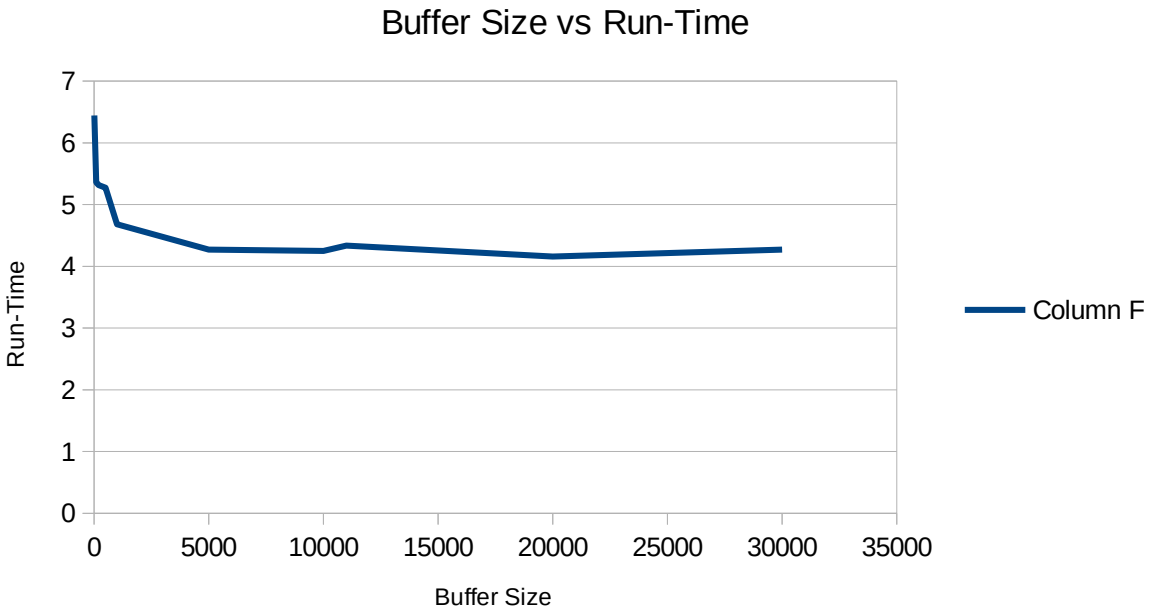
Graphs before PA4:



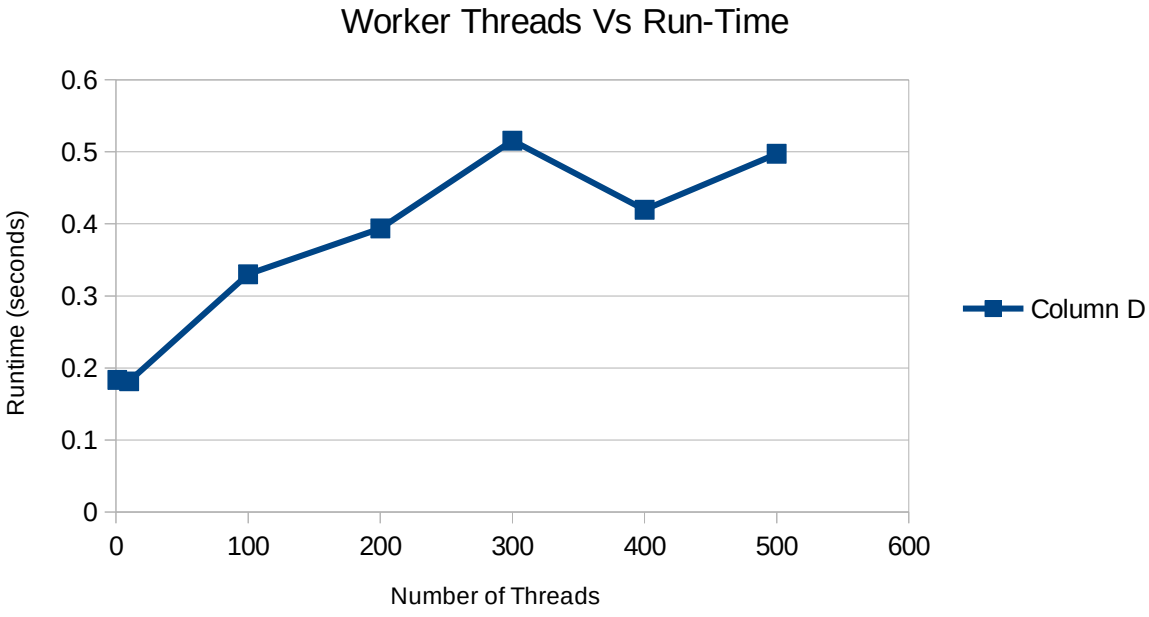
Graphs After PA4:



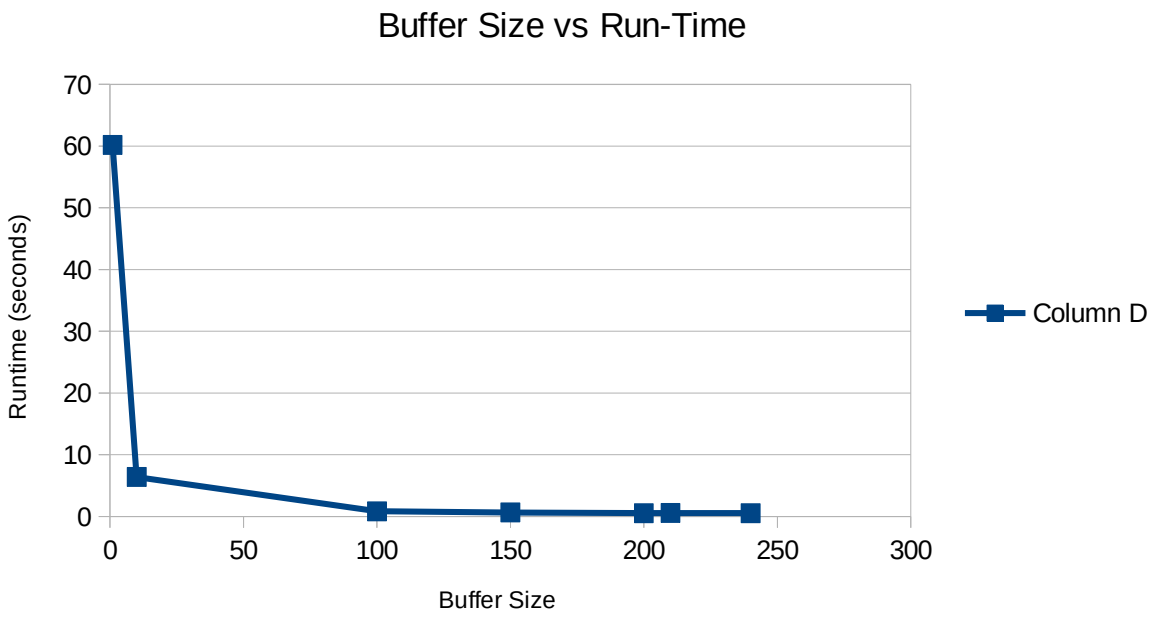
Mauro Lopez



File Request:



Mauro Lopez



Conclusion:

Data Request:

Taking a closer look at my data request program we can see the relationship between increasing the number of threads and the runtime. We can see that the run time decreases in an exponential format. This matches our expectation of working with threads. This run time allows me to see how having multiple threads running all at once, can in fact decrease run time since they are all executing a piece of the entire program. If we leave the number of threads constant and focus on the bounded buffer size, we can see that having a bigger buffer does decrease our run time. This makes sense, since not a lot of worker threads will have to wait until the buffer is empty, since its relatively big.

File Request:
Looking at the graph we actually see the opposite of decreasing run time. It seems that the more worker threads we have running, the more time we spend over the entire program. Taking a close look at this scenario, I can see that this was probably due to the fact that our number of producers were relatively less than our worker. This means that our producer couldnt put enough task in the buffer, so a lot of those extra worker threads would have to check the queue and wait for a process to come to the queue.

When we set the number of threads constant, and change the buffer size. We can see that having a smaller buffer, can make the run time really high. Increasing the buffer size, we see a decrease in the run time. This is a result of the program asking the FIFO for very limited amounts of data. Even with 500 threads, we are limited by how much we get from our FIFO. In both cases we see a linear pattern, even though they are of opposite trends.