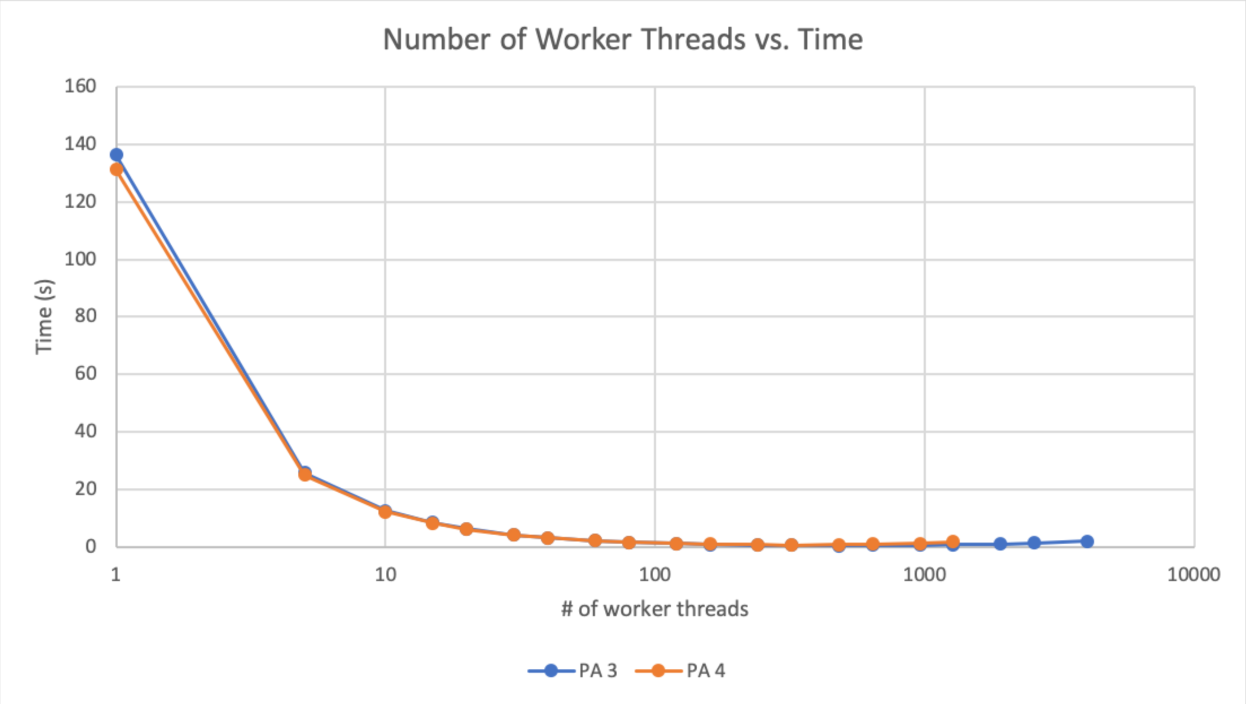
Sean Patlan

CSCE 313-508

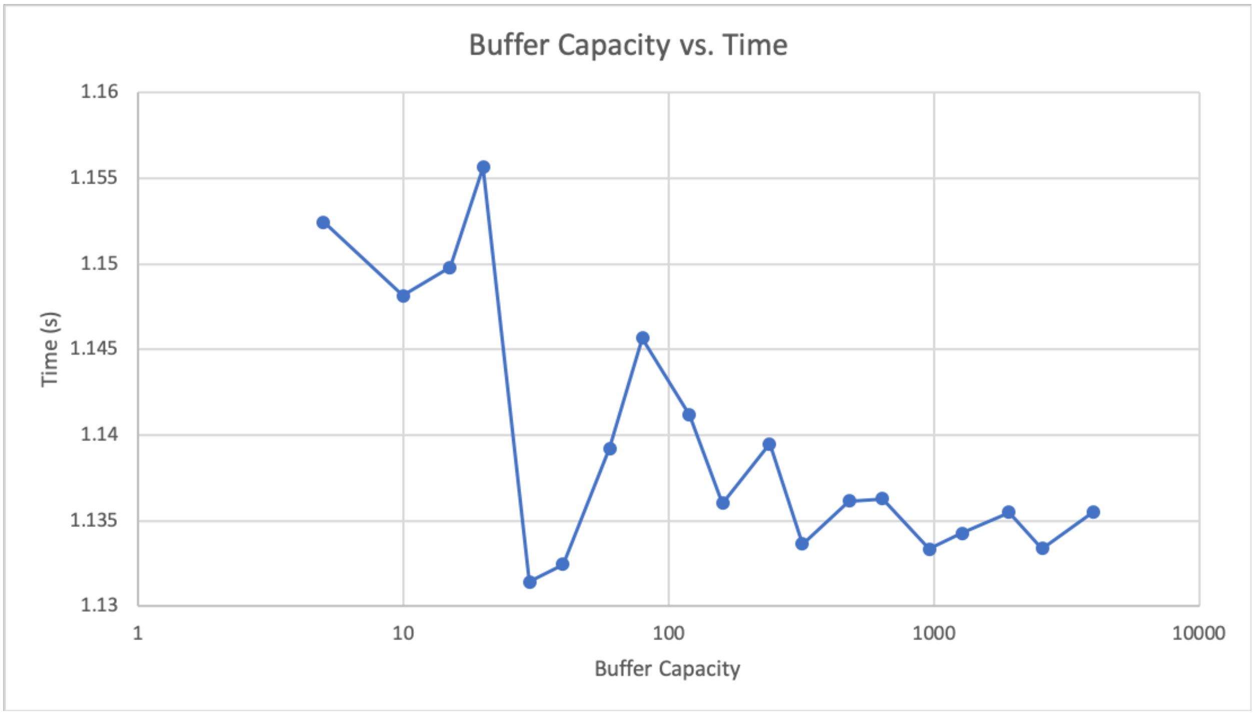
Programming Assignment #4 Report

My PA 4 implementation worked at essentially the same speed at PA 3 did with regards to the worker threads. To test the speed of the worker threads, I kept n constant at 10000 (the same as the PA 3 tests) and b constant at 100. I found that initially, with fewer worker threads, PA 4 worked slightly faster than PA 3 did. However, once the number of worker threads reached the hundreds, PA 3 surpassed PA 4 and operated slightly faster for the same number of worker threads. This is likely because the PA 4 implementation uses more threads than that of PA 3, so the kernel is having to do more context switches. In my opinion, this slight decrease in performance is worth the ability to produce and consume data at the same time, as that is more akin to real life computing. Same as PA 3, PA 4 began to slow down between 480 and 640 worker threads. Below is a graph comparing the two on a logarithmic scale:



Varying the b value in PA 4 had less of a visible effect on the speed of the program. To test its effects, I kept n constant at 10000 again and kept w constant at 120. I found that the b values don’t seem to directly affect the speed of the program. In fact, I even running the exact same command twice (without varying b) sometimes yielded very different operating speeds.

All of the trials in this case finished within 0.03 seconds of each other. I believe this is because threads are producing and consuming data concurrently, so the buffer is never full or empty for very long, making the exact capacity rather arbitrary. Below is a graph of the effect of b on the speed of the program:


Buffer Capacity vs. Time
Time (s)
Buffer Capacity