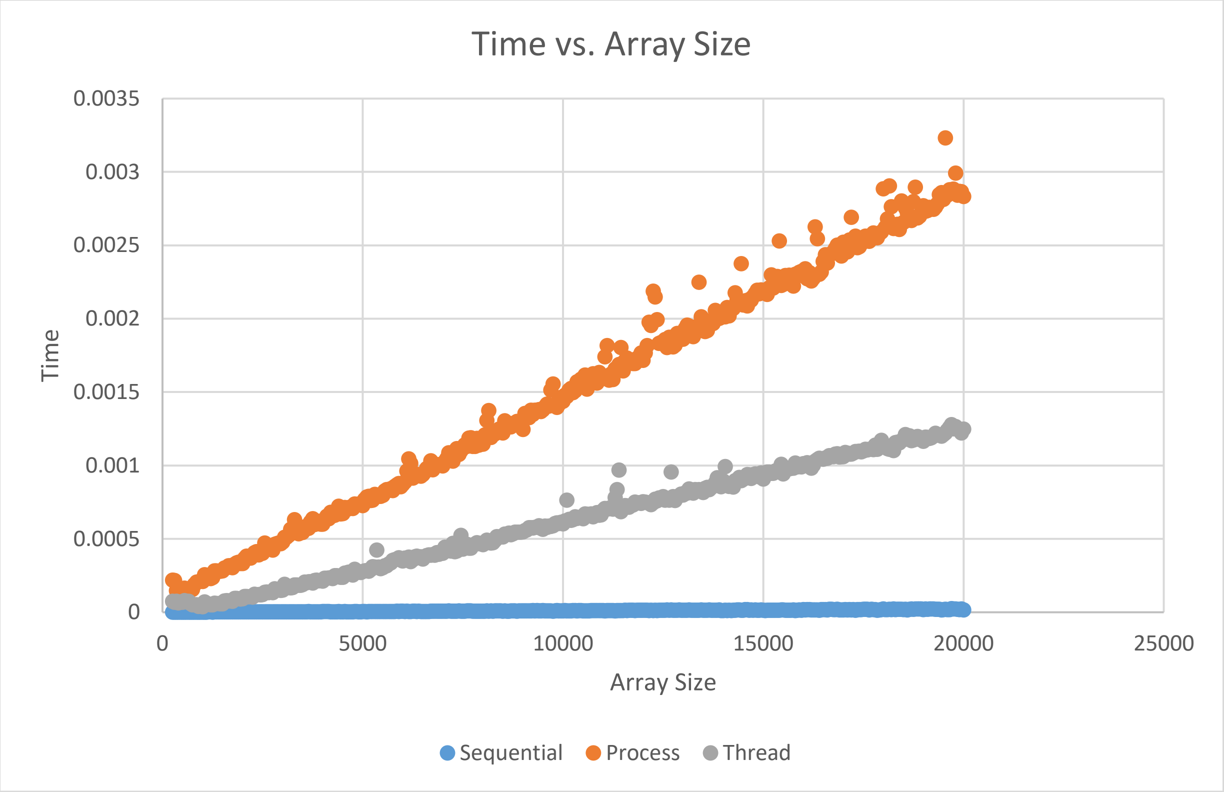
CS214 Systems Programming
Karen Lee – kwl47 – Section 07
Michael Loh – ml1328 – Section 03

SPOOKY SEARCHING: RESULTS

Part 1: Experimenting with Array Size

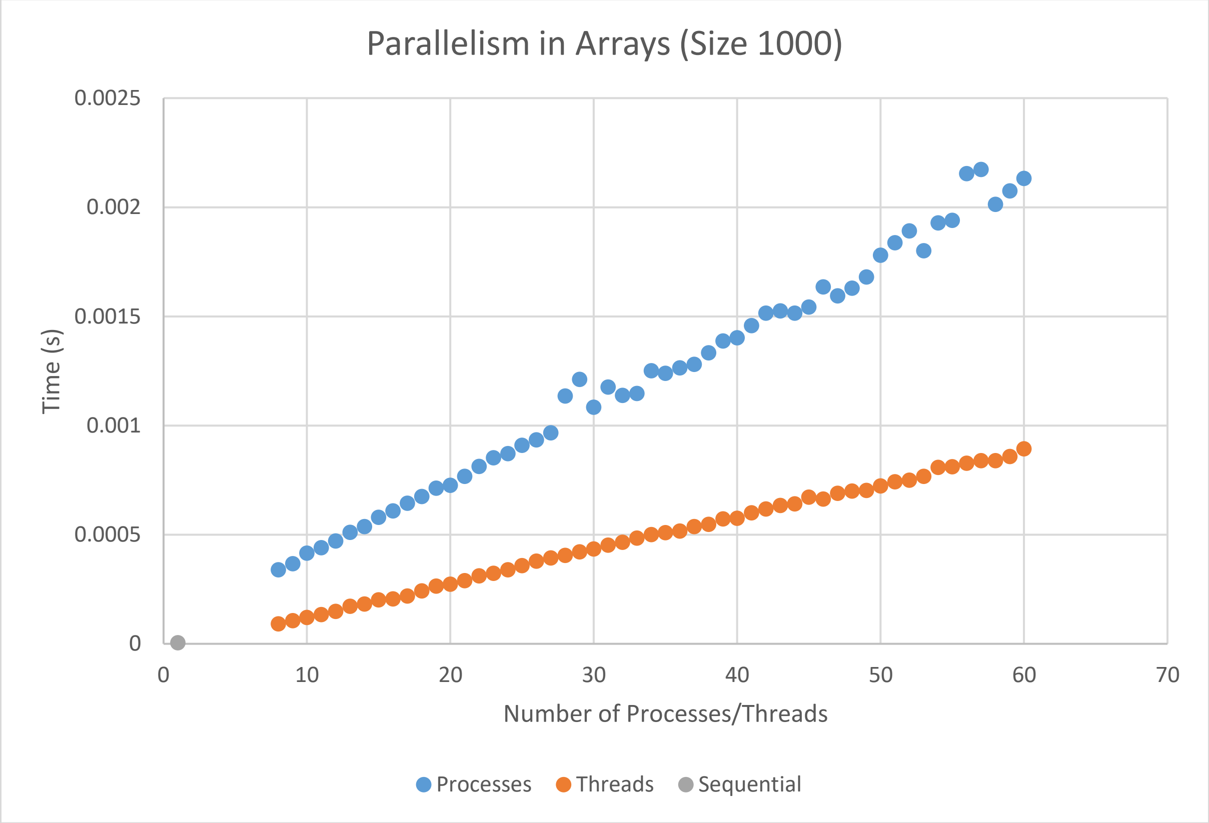
Workload A



- In this graph, we can see that sequential search is by far the fastest way to search an array at larger sizes.
- Multithreading is slower than sequential search, but is still much faster than multiprocessing.
- The standard deviation of the multiprocess search is by far the highest, followed by threads. As array size increases, the standard deviation increases too.
- Increasing array necessitates a greater amount of processes/threads, which is probably the cause of the slow down for multithreading and multiprocessing.

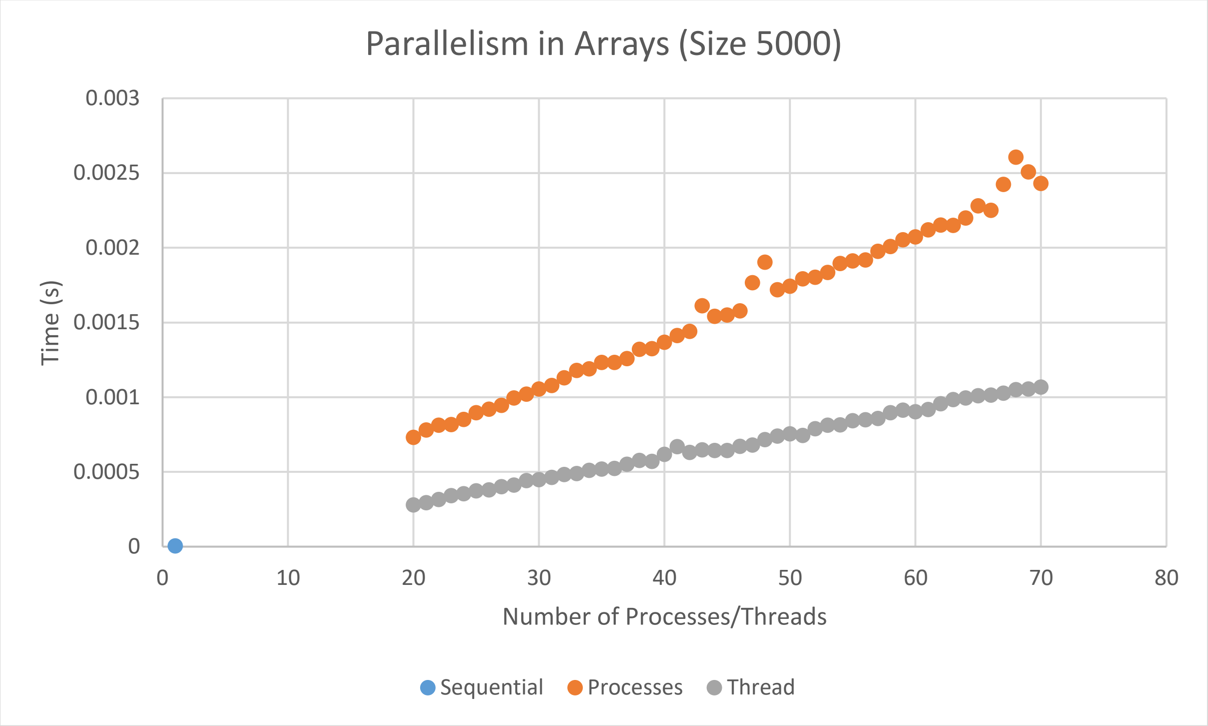
Part 2: Experimenting with Number of Branches

Workload B



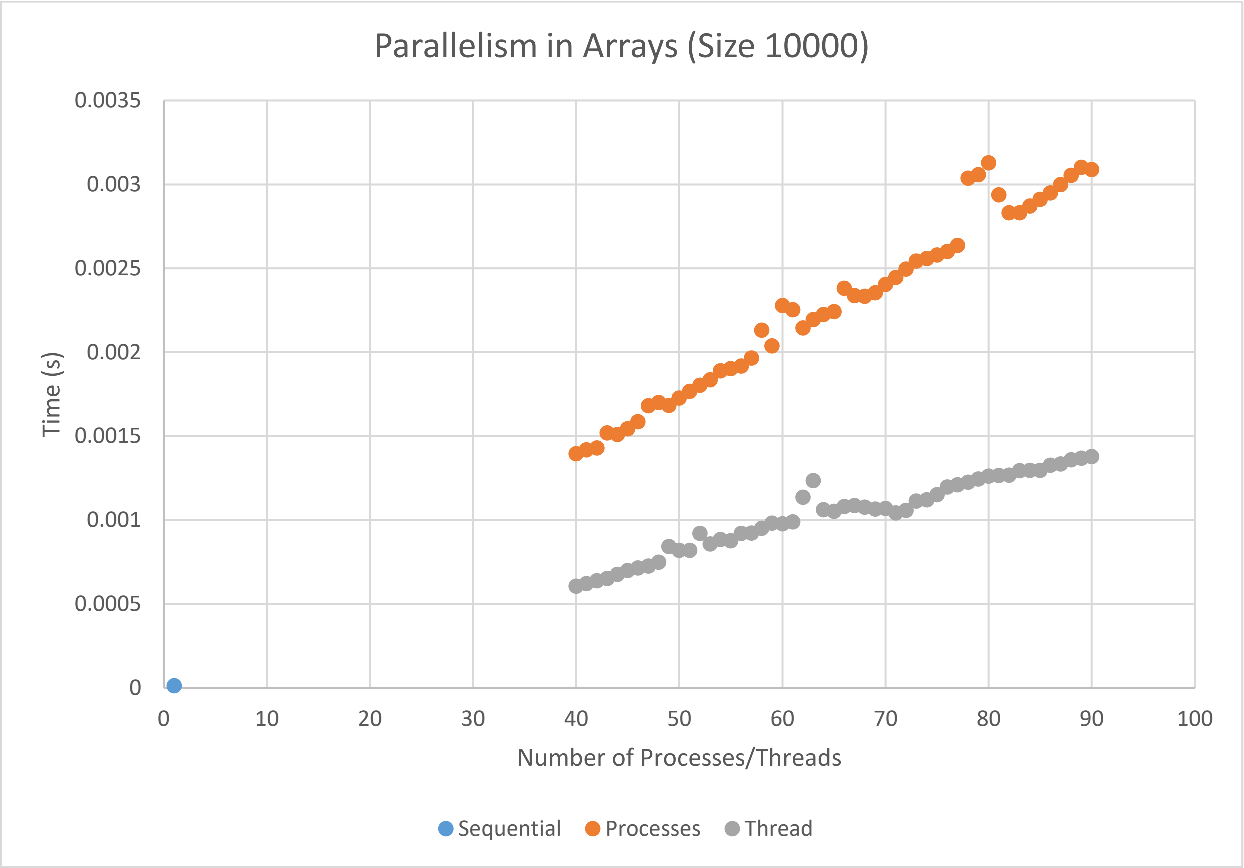
- By adding more threads and processes, the time it takes to search the array takes longer.
- Processes remain slower than threads.
- The rate of increase for both in this workload appears to be linear.

Workload C

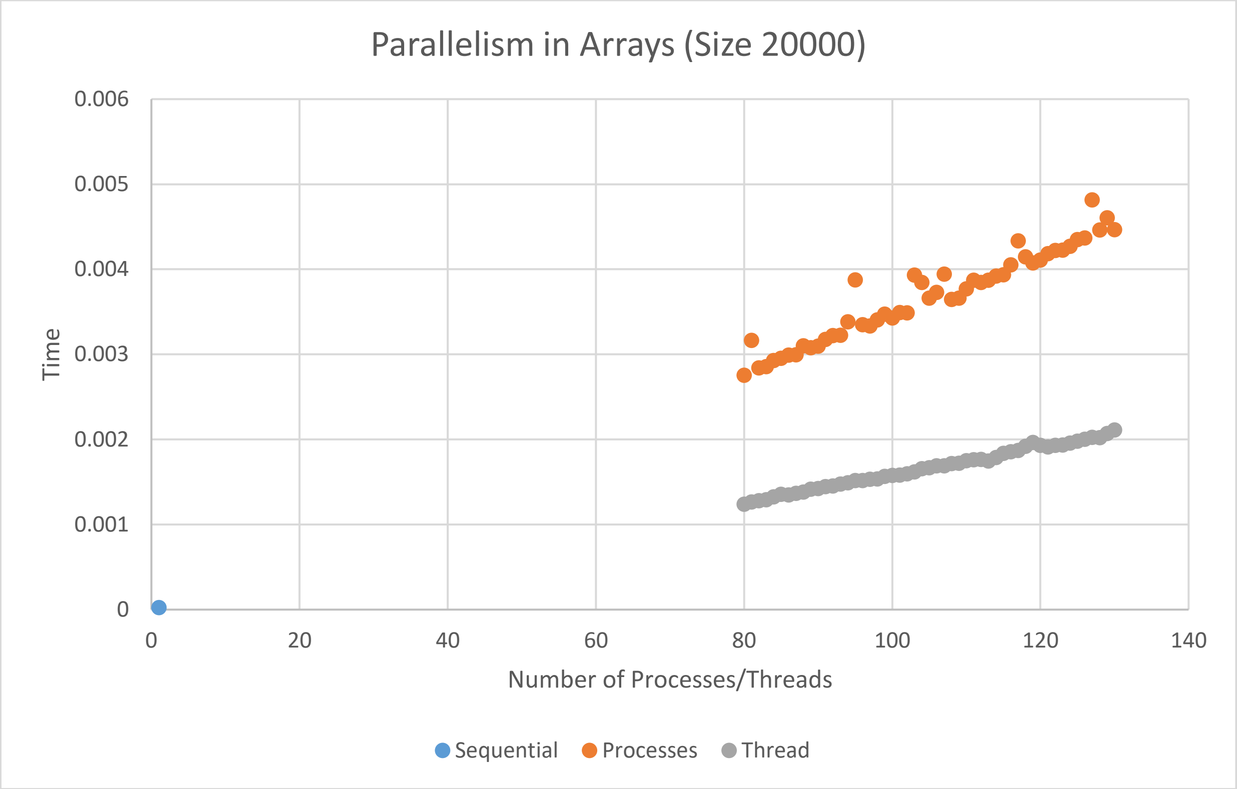


- Despite increasing the array size by a factor of 5 from Workload B, the results of this workload are more or less the same.

Workload D



- In workload D, we continue seeing the trend of how more threads/processes equates to longer time to search. Workload E
- With larger arrays, the standard deviation continues to rise as well.



- Workload E, our largest test, still shows the same linear trend as all of the workloads testing parallelism.
- Since all the tests on parallelism produced more or less similar results, we can probably attribute the rise in the standard deviation to the increased array size necessitating more processes/threads.