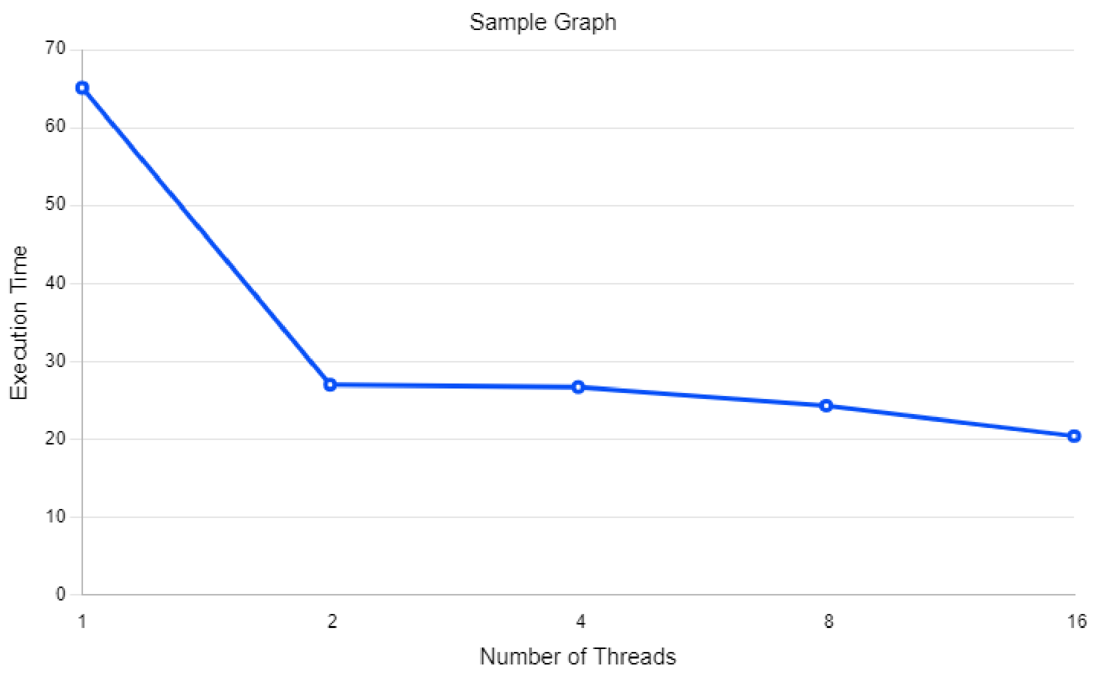
Programming Homework 2 Report

Part 1.1

averaged after 3 runs, n=2048 for p:

- 16, t=20.4s
- 8, t=24.3s
- 4, t=26.7s
- 2, t=27.0s
- 1, t=65.1s
- C[100][100]=230,092,800 for all experiments

observation: as the number of threads decreases, the execution time generally increases slightly except when there is only 1 thread, execution time almost doubles.



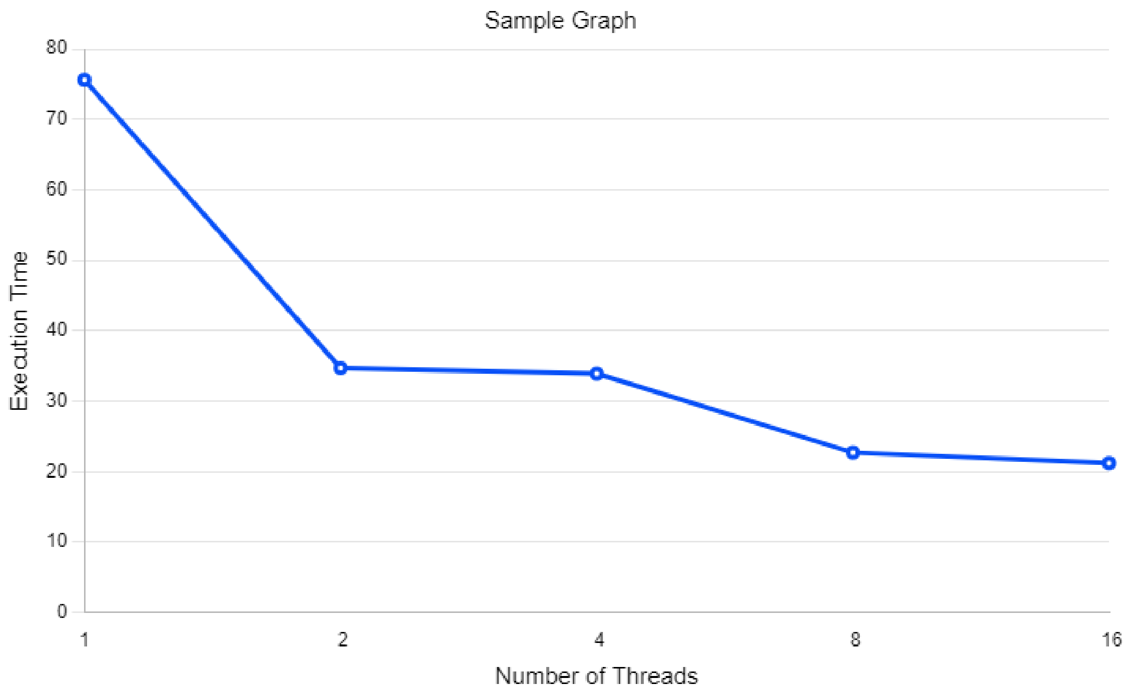
strategy: dividing C into n/p number of blocks, where each thread is responsible for calculating and updating each block in C.

Part 1.2

averaged after 3 runs, n=2048 for p:

- 16, t=21.2s
- 8, t=22.7s
- 4, t=33.9s
- 2, t=34.7s
- 1, t=75.6s
- C[100][100]=230,092,800 for all experiments

observation: once again, as the number of threads decreases, the execution time increases slightly except when there is only 1 thread, where execution time almost doubles.



strategy: each thread is responsible for a n/p sized block in A, and each entry in A contributes to some output in C. We incrementally update C as each entry in A is being multiplied by the corresponding element in B and add that to C. Locks are needed to ensure the integrity of C as the same entry might be updated by multiple threads.

Key Difference: in Problem 1.1, there is a smaller benefit to using a larger number of threads; 2,4,8.16 all give similar results. This contrasts Problem 1.2, where 8,16 yield a group of similar results, 2,4 share another tuple of similar results, and a purely serial program requires the longest execution time. One possible explanation is that there are more operations in 1.2 that would benefit from the increased number of threads in 1.2, whereas in 1.1 the overhead of forming new threads is significant enough to reduce the overall performance gains of the added number of threads.

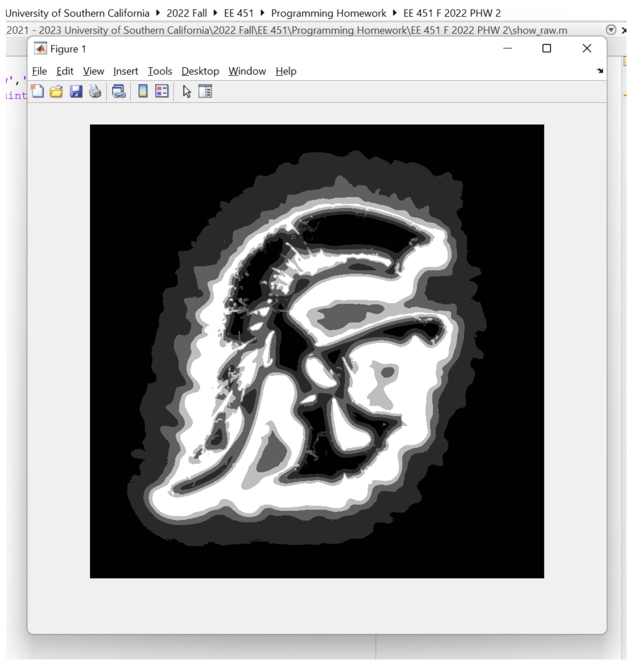
Part 2

averaged after 3 runs for p:

- 8, t=1.33s
- 4, t=1.74s
- 1, t=1.80s

observation: the serial version is almost on par with the parallel version. Again, this might be due to the fact that there are not enough operations to parallelize to cover the expensive cost of the pthreads overhead.

image: apologies for the late submission, but here is the screenshot!


University of Southern California ▸ 2022 Fall ▸ EE 451 ▸ Programming Homework ▸ EE 451 F 2022 PHW 2
2021 - 2023 University of Southern California\2022 Fall\EE 451\Programming Homework\EE 451 F 2022 PHW 2\show_raw.m
Figure 1
File Edit View Insert Tools Desktop Window Help