CS230 - HW3

System Description:

Processor: Intel(R) Core (TM) i7-8565U CPU @ 1.80GHz
RAM: 8 GB
Operating System: Ubuntu (Version - 20.04.1)
Cores: 8 (Quad core processor, with hyperthreading enabled)

Below table summarizes the execution time to generate prime number till 10,000,000 for different number of threads in both single and multi-threaded mode:

Threading Mode	Number of Threads	Execution Time (Sec)
Single-Threaded	1	12.37380804
Multi-Threaded	1	12.67784915
Multi-Threaded	2	6.11113423
Multi-Threaded	4	3.29556469
Multi-Threaded	8	2.22296715
Multi-Threaded	16	2.15619553
Multi-Threaded	32	2.09740643
Multi-Threaded	64	2.08309155
Multi-Threaded	128	1.96178491

Table1: Execution time for different number of threads (single and multi-thread mode)

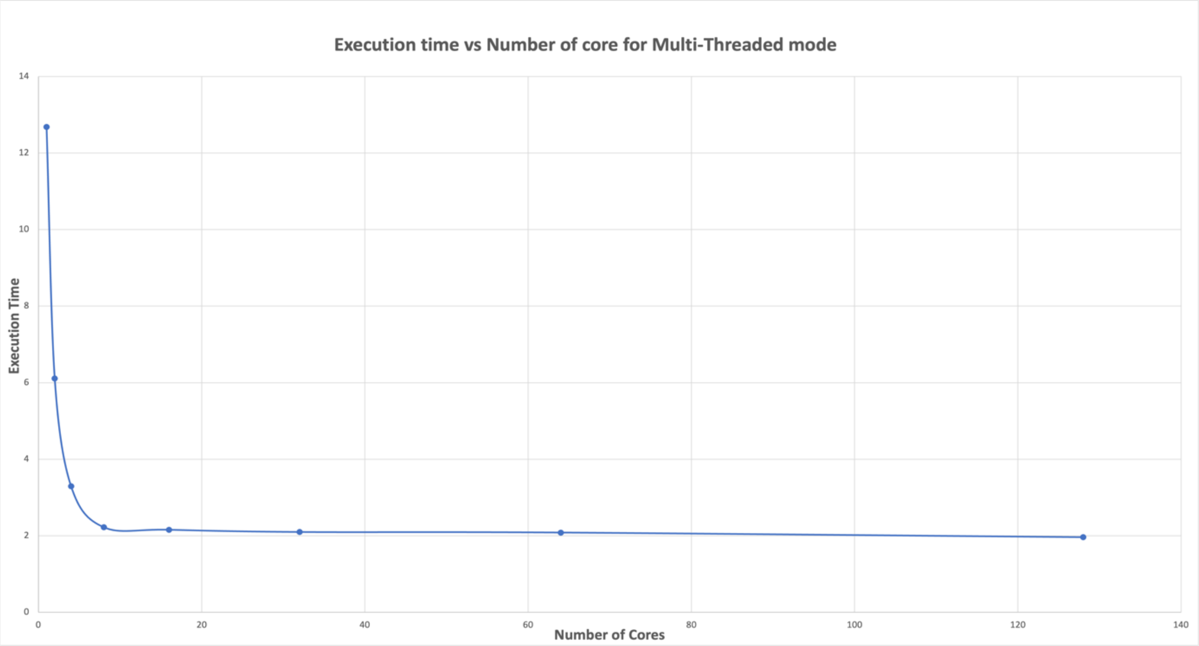

Figure1: Execution time for different number of threads (single and multi-thread mode)

Analysis of the performance of the programs with emphasis on **correctness and speedup:**

Speedup Analysis:

- As we increase the number of threads in multi-threading, we notice that the execution time decreases exponentially till 8 threads.
- As we increase the number of threads beyond 8, the execution time remains almost the same.
- Above behaviour can be attributed to the fact that our system has 8 cores, hence at max 8 threads be executed simultaneously (1 thread by 1 core) in ideal situation. Therefore, we see drastic improvement in the execution time as we increase the number of threads to 8. Moreover, in case the number of threads is more than 8, the threads after the 8th thread will get the core only when the core becomes free i.e. when the core finishes the earlier threads. Hence, we don't see improvement in execution time beyond 8 threads.

Correctness Analysis:

- Program is yielding correct result for both single and multi-threading cases by generating 664579 prime numbers which is the total number of prime numbers below 10,000,000.
- For single thread case the prime numbers are printed in order, however if number of threads is more than one, we are not getting prime numbers in order because the thread which got created later can finish earlier.

An analysis of **the single-thread version versus the multi-thread version with only one thread**. Is there a difference? How do you explain the results? Add plots and figures as necessary to explain your results

- As we can see from Table1, with only one thread, we see slight increase in execution time (around 0.3 sec) for multi-threaded mode compared to single-threaded mode.
- For single thread, both single-threaded and multi-threaded program will execute the instructions sequentially, however in case of multi-threading there is an additional overhead due to mutex lock and unlock. Hence, we see slight increase in execution time.