Assignment – Homework 2
Class – Computer Science 446: Section 1001
Author – Samuel Mouradian
Professor – Christos Papachristos
Assignment Due Date – 10 / 03 / 2024

Homework 2 Response Questions

Run your code with a different combination of:

i) Amount of values (e.g. 1000, 1000000, …)
ii) Requested Threads for each amount (1 2 4 8 16)

Then, answer the following questions:

1. What are your observations?

   As the number of threads increases, the elapsed time decreases, because the computation is divided among the threads. However, if there is a large sample size, the elapsed time can increase.

2. Do you find the observed behavior reasonable? Why / Why not?

   Yes, because threading improves performance - up to a certain point, that is. If there are too many threads, or if there is too large of a data set, then it's possible the performance will be slower.

3. What kind of considerations did you make for your implementation of the *Critical Section*? Provide reasoning / justification:

   The critical section was mainly focused around the "totalSum" variable. Since a lock was placed around the variable, it allowed the threads to access and modify the value individually instead of all at once.

4. [EXTRA] What do you think would happen if instead of having the Threads loop over the **int** array to compute a local **arraySum**, and finally update the **totalSum**, we were instead just directly adding to **totalSum** each time we were accessing an array element within each Thread (i.e. within the *Thread's* **for** loop)? Why?

   If the "totalSum" value was updated every time we accessed an array element within each thread, then there would be incorrect values being summoned by other threads that the totalSum has been set to. Furthermore, there would be a drastic effect in the execution time – it would be much slower.