Joshua Catalan

CSE 3320-003

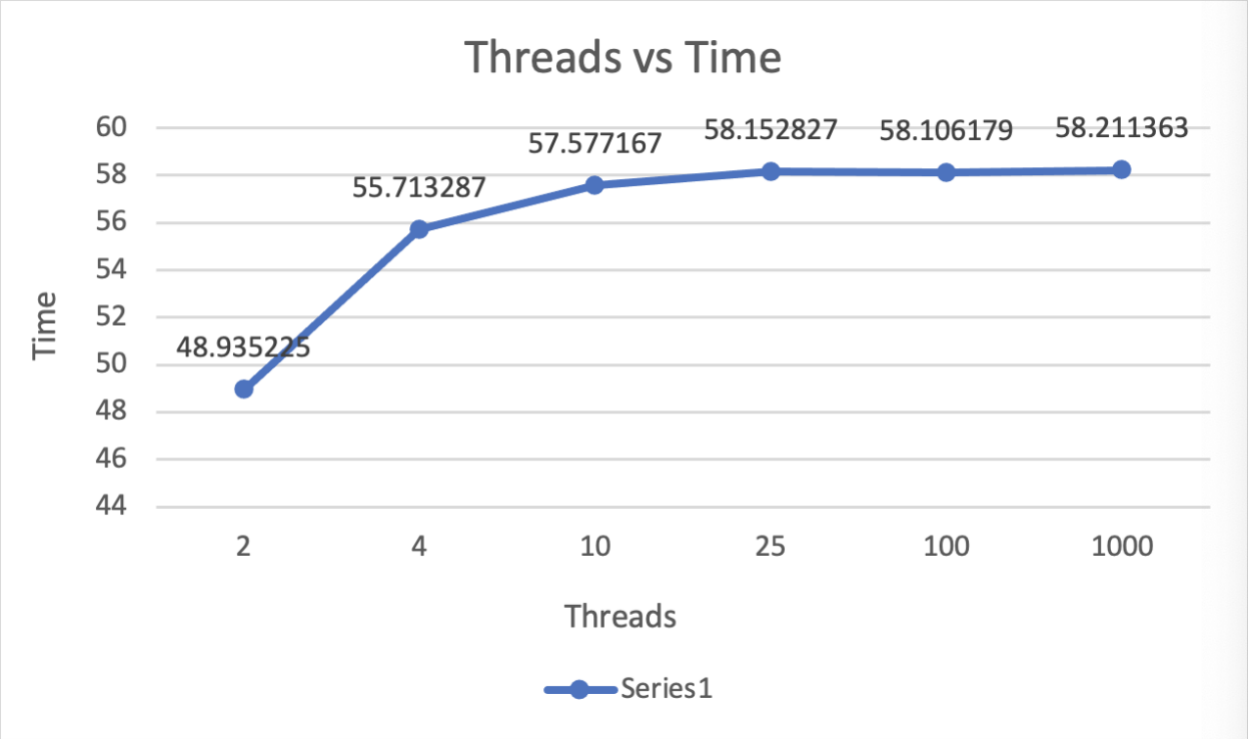
Star Catalogue Assignment Report

The purpose of this was assignment was to introduce threads to the Tycho Star Catalogue that was provided. Using threads can allow multiple tasks to be performed simultaneously within a single process, allowing for better performance. The original program calculated the average distance, minimum distance, and maximum distance for the number of records read (30,000). I had to use different amounts of threads such as 2, 4, 10, 25, 100, and 1000 threads and gather the amount of time elapsed for each thread. To implement the threads, I had to include the pthread.h library. I also included a mutex to help protect the shared variables from being accessed by multiple threads during execution. In doing so, it helped mitigate deadlocks and race conditions. For the timing method, I used the clock_t data type in the time.h library because it represents the number of clock ticks used by a processor to execute a program or a specific part of the program. I then subtracted the end time by the start time and divided that using the macro, CLOCKS_PER_SEC to get the value in seconds.

Table

# of Threads	Time Elapsed
2	48.935225
4	55.713287
10	57.577167
25	58.152827
100	58.106179
1000	58.211363

Graph



The one anomaly I found while recording the time elapsed for each thread was that the more threads, I used the more the time went up to a certain point. The time plateaued after using 25 threads. Upon further research, this may be due to Code spaces only using 2 cores.

In conclusion, the optimal number of threads to use is 2 threads because after that the time elapsed seems to increase. The optimal number of threads also depends on the task running as well as the hardware resources available on the given system. Running the program on Code spaces which has 2 cores, makes the optimal number of threads to use about 2 threads.