Nebojsa Kilibarda
1001934192

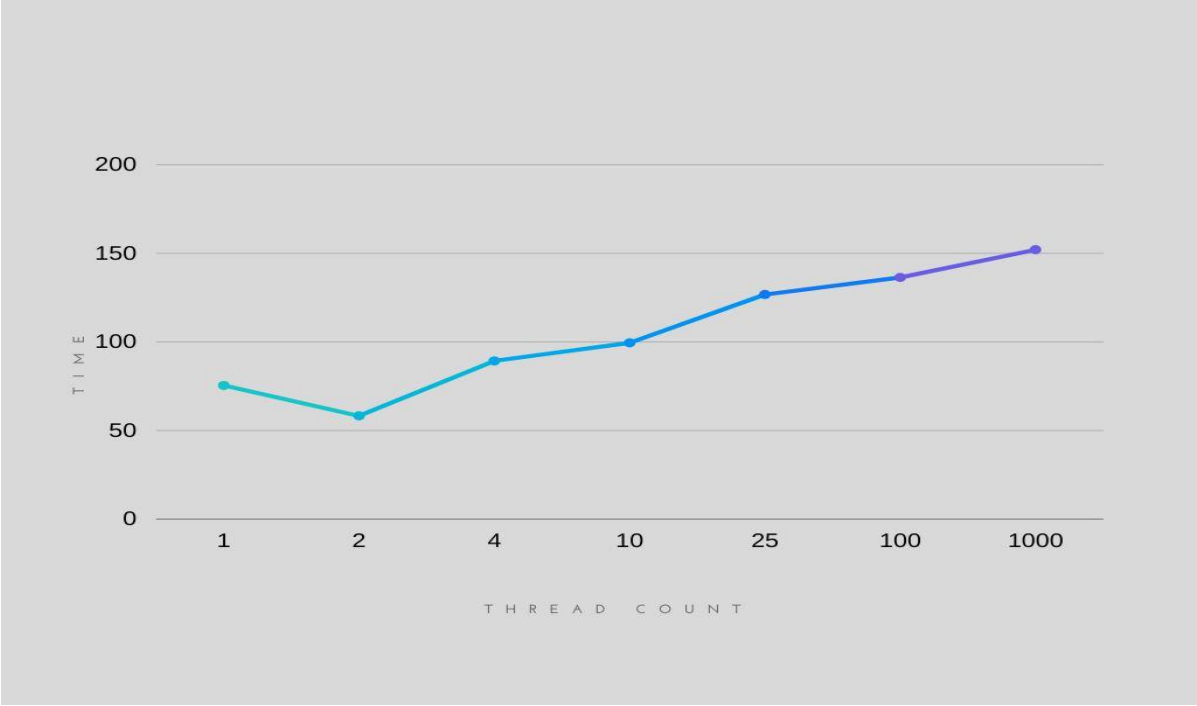
Executive Summary of Program #2

1. This program calculates angular distances between stars in the tycho system. It's a multithreaded program with threading performed in parallel. It allows the user to specify the amount of threads to use, which are then used to count the distances between the stars in parallel. The program uses the pthread.h library, although there may be more optimal solutions. In addition, mutexes are used to ensure no race conditions occur at large numbers of threads.
2. Aside from the pthread.h library, which provides the create, join, and exit functions for threads, time.h is used to instantiate a timer for the running time of the threads. This method is preferable in my opinion because it allows us to time the specific portion of the code that is of interest to us (In this case, the specific time between the creation of the first thread and the termination of the last one)
3. The results from the program are noticeably affected by the Codespaces used for the implementation. The Codespaces provide a dual core CPU for calculations which makes two threads the optimal solution when running this program. This is because a larger number of threads will be 'stuck' waiting for the mutex to unlock, thus reducing the cost efficiency. In addition, the Codespace environment is unreliable at times due to high use, with frequent disconnections from the server and other interruptions. When running the program on my local machine, the results deviate less between runs, and the program performs better time wise.

Table 1.A

#threads	1	2	4	10	25	100	1000
time	75.2357	58.0143	89.1136	99.3241	126.618	136.260	151.934

Graph 1.B



4. Two anomalies are present in the resulting data. Namely, on certain runs the running time goes for an unexpectedly long time, perhaps due to the Codespaces environment. Instead of running for 73-75 seconds, it would take over 100. Additionally, the average distance found slightly differs between different thread numbers, providing values that are slightly incorrect.

5. In conclusion, the optimal number of threads will depend on the CPU running the program. In our case, since Codespaces is used, and it provides 2 CPUs, we get the best running time when using two threads. On my local machine, which has an 8-core cpu, 8 threads performed the best.