The program uses pthreads to implement a multi=threaded application to serially calculate the average angular distance between 50,000 stars. Mutex locks were used to prevent race conditions and to protect shared data from being read by more than one thread at a time. It includes the time.h library for the clock_t feature to measure time due its accuracy and reliability compared to the other timing methods I tried. In conclusion, the optimal number of threads to use with the least amount of time taken was 2 due to codespaces having two cores. As we can see in the graph, the more threads you use after two, the more time increases slightly.

## Time vs. Threads
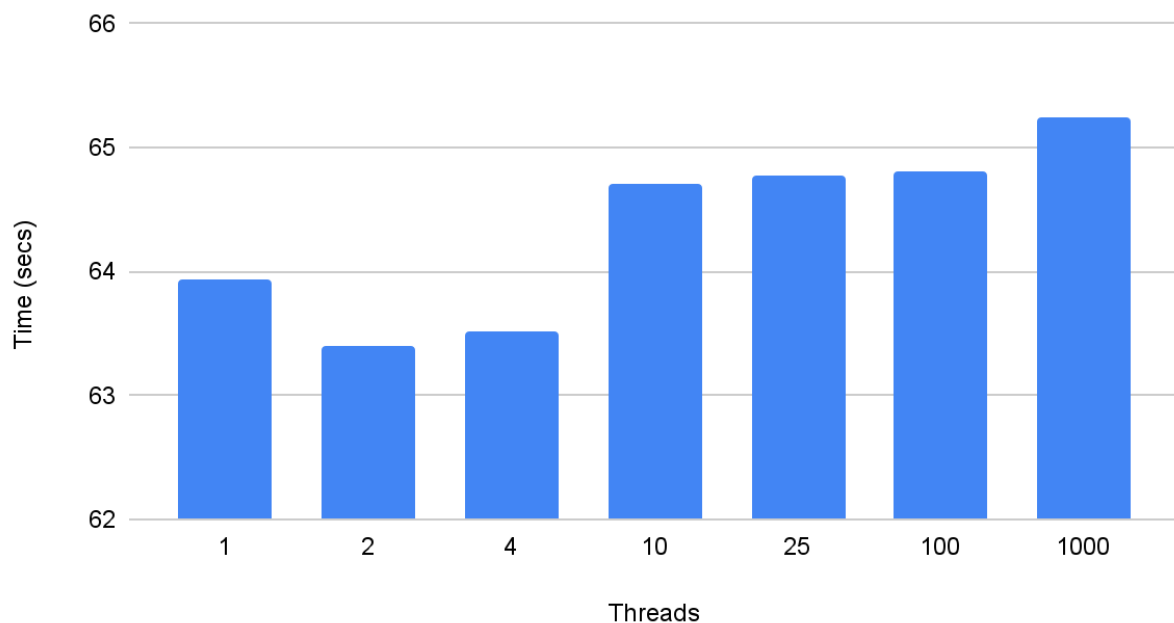


| Threads | Time | Average Distance | Minimum Distance | Max Distance |
|---|---|---|---|---|
| 1 | 63.92992 | 31.904232 | 0.000225 | 179.56972 |
| 2 | 63.400072 | 31.904232 | 0.000225 | 179.56972 |
| 4 | 63.512847 | 31.904232 | 0.000225 | 179.56972 |
| 10 | 64.710653 | 31.904232 | 0.000225 | 179.56972 |
| 25 | 64.77785 | 31.904232 | 0.000225 | 179.56972 |
| 100 | 64.811818 | 31.904232 | 0.000225 | 179.56972 |
| 1000 | 65.251164 | 31.904232 | 0.000225 | 179.56972 |