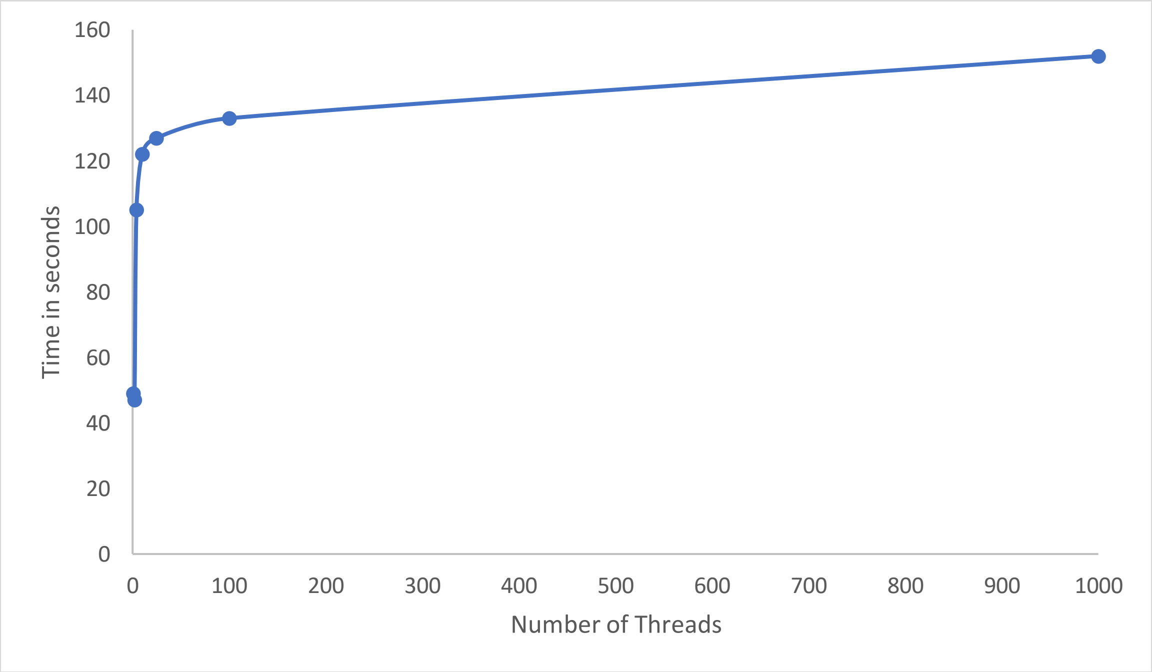
In the assignment, we are tasked to multi-thread the application that will calculate the average angular distance between 30,000 stars in the Tycho Star Catalogue using pthreads. In doing so, we can determine between 2, 4, 10, 25, 100, and 1000 on what is the optimal number of threads that will take the least amount of time to do calculate the minimum, maximum, and mean angular distance.

To get started, I first implemented my timing method. I included the sys/time.h library to get the time. By using the function gettimeofday before creating the threads and using the function again after all the calculations, I was able to get how many seconds the program took by subtracting the start time by the end time. I chose this timing method because I wanted the time that it took when the program starts until when it ends. Next, I implemented the pthread.h library so that the program will be able to have a threading interface to test out how the thread amount affects the time.

Thread Count	Elapsed Time (in seconds)
1	48.999283
2	46.999668
4	104.999756
10	122.000526
25	126.999779
100	132.999954
1000	152.000031



By looking at the results of the timing how long it takes to calculate for each thread count, we can see that they follow the trend at the beginning where it looks like the more treads there are, the longer it takes. However, the is very noticeable when the number of threads increase from 100 to 1000 threads. We would expect the more threads to take a more considerable time. From the data, we can also see that for 2 thread counts, it records the fastest time compared to all the other thread counts. To add onto the timing problem, when running on code spaces, one thread appears to be faster than 2 threads. However, when the code is ran locally, it has a noticeable faster time at 2 threads.

In conclusion from the data, the optimal number of threads is 2 threads. This is because it takes the fastest time to do the calculations compared to the rest of the thread counts.