REPORT

Design:

The code starts with reading from a sample text file for the inputs. We must give the name of the input file. The input file contains the number of threads, array size, and the unsorted array. By using this we will create the array of threads asked for using the 'pthread_t' type. Now we will create threads using and direct to 'goSort ()' function to assign work to each thread to sort the array using the 'pthread_create ()' function in a for-loop. Before this, we will start the clock.

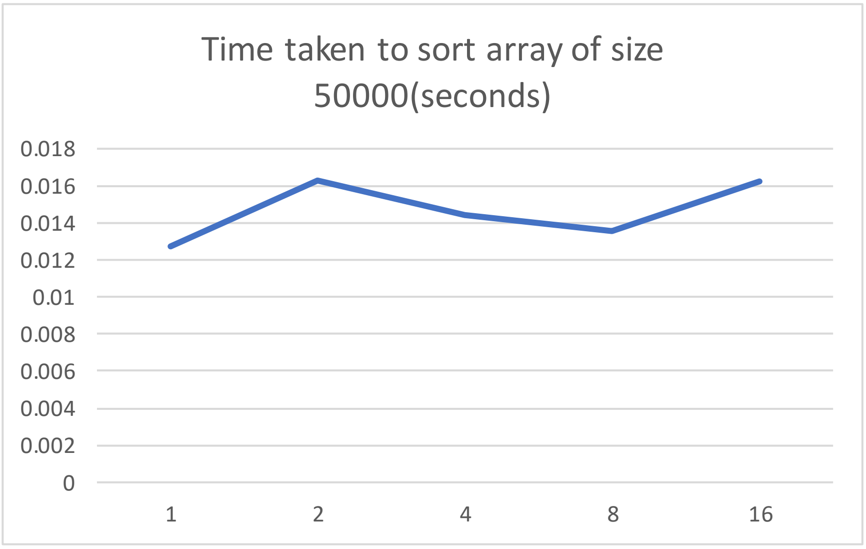
The 'goSort ()' function divides the array into parts equal to the number of threads and gives each part to each thread and then each thread does merge-sort on a given part concurrently and then ends the thread after completion. Now, we join back with the main thread using the 'pthread_join ()' function. We now merge all these parts, and we get the sorted array.

Comparison of performance:

(a) Varying the number of threads:

x-axis: number of threads varying as - 1, 2, 4, 8, and 16.

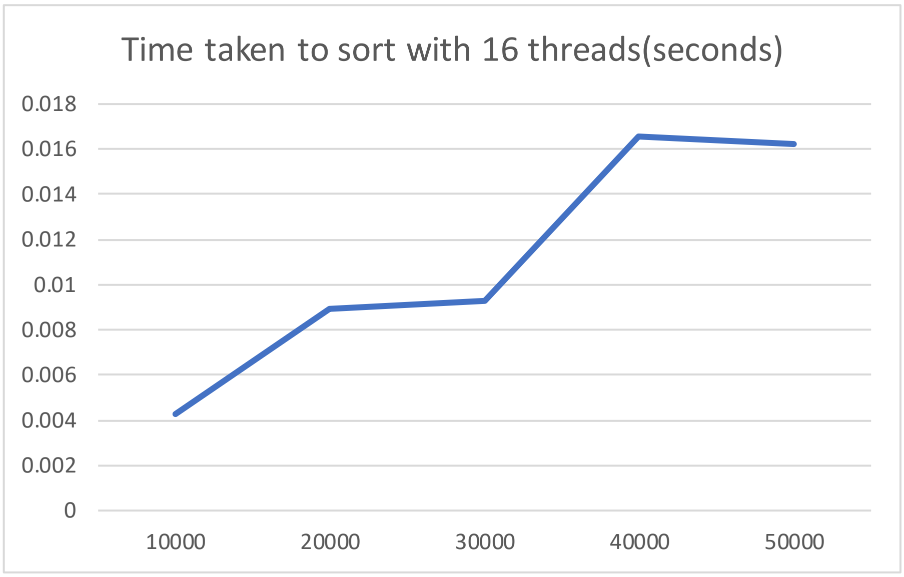
y-axis: time taken to sort a random array of size 50000.



(b) Varying the input array size:

x-axis: the array size varying as - 10000, 20000, 30000, 40000, 50000.

y-axis: time taken to sort with the total number of threads as 16.


Time taken to sort with 16 threads(seconds)