COL380_Assignment1

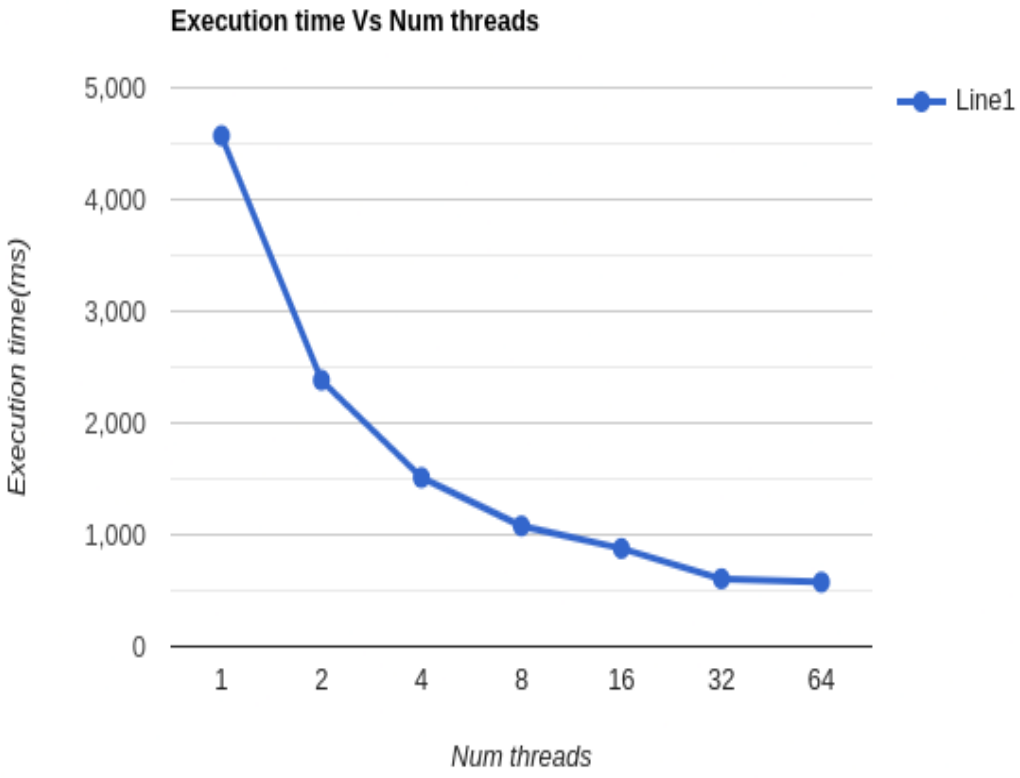
Swaraj Gawande

January 2022

1 Outline

As the number of threads free does not remain the same I used number of partitions as the parameter to distribute work among tasks. Each task passes through all the all the data and classifies the data corresponding to the partition number/bucket number only to avoid data races.
As the algorithm is recursive the sequential sort *seqSort*, which is essentially quicksort, is applied when the size of the array is less than threshold. In addition to this if the original size of array is such that we could not select p*p pseudo splitters, then we use *seqSort* directly in the array. After the classification for a given partition is completed further recursive call or sequential sort is done. After assigning work for p tasks the program waits for execution of each task is completed. Following is the graph which shows the scalability of the program. As the number of threads is increased provided that there is sufficient number of CPUs, the use of the hardware increases proportionally. And thus we could see a decrease in execution time with increasing count of threads.

2 Graph


Execution time Vs Num threads
5,000
4,000
3,000
2,000
1,000
0
Execution time(ms)
1
2
4
8
16
32
64
Num threads
Line1