Deepansh Sachdeva (NUID: 001399788)

INFO 6205
Program Structures & Algorithms
Fall 2020
Assignment 5

Task

Implement a parallel sorting algorithm such that each partition of the array is sorted in parallel and draw a conclusion about the efficacy of this method of parallelizing sort.

Evidence

Using merge sort logic for parallel sort to get average times for different cutoff values:

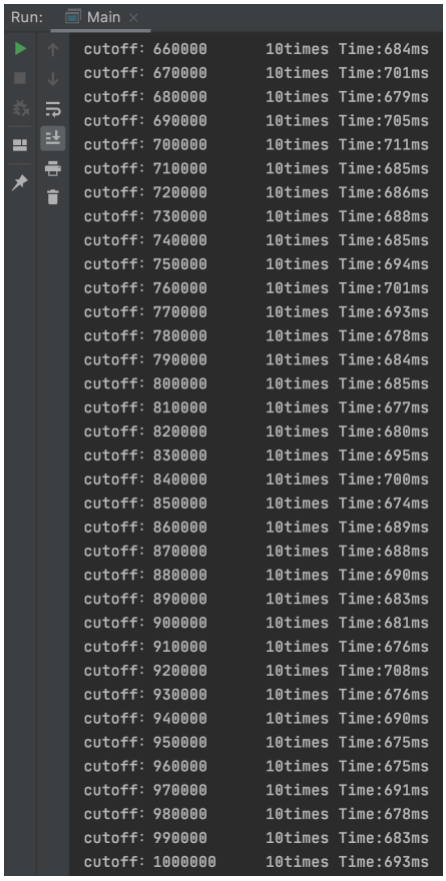

Figure 1 Execution log

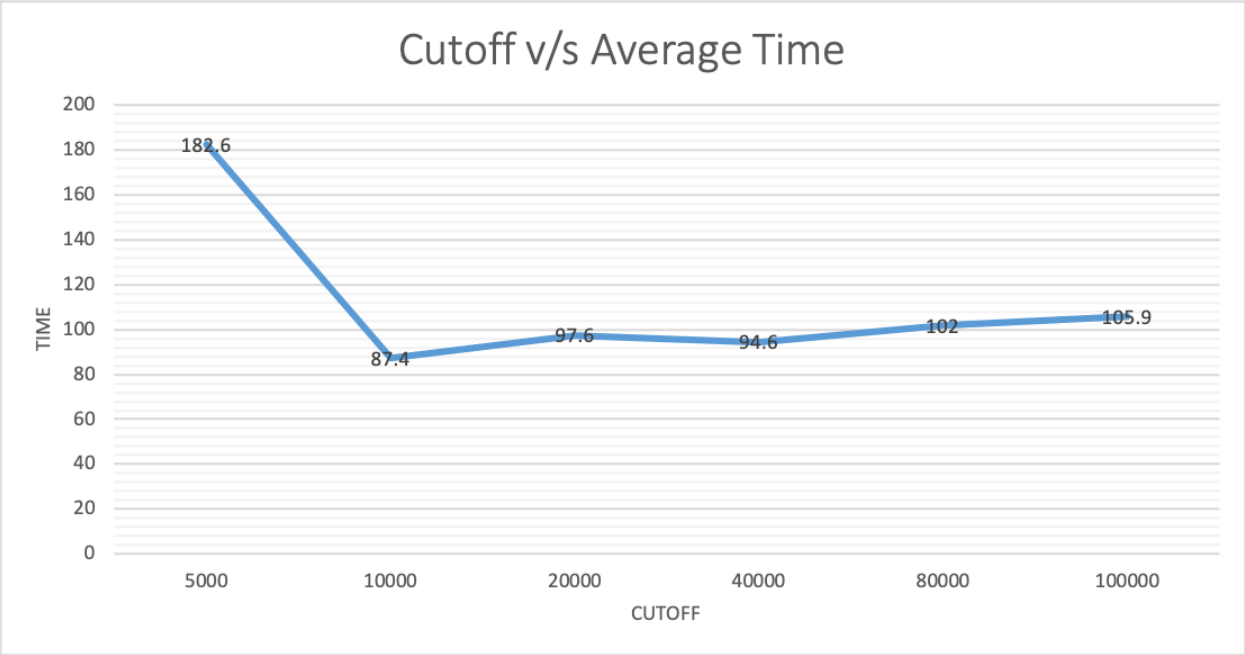

Figure 2 Graph plot of cutoff vs time

Conclusion

1. Implemented parallel sort using divide and conquer (merge sort) method by join and fork in java.
2. Cutoff is introduced to use system sort for array size less than cutoff and use merge sort instead.
3. Average time for various cutoff remains almost constant but considerably faster with 10,000 cutoff.
4. Parallel / Asynchronous merge sort is more efficient than normal merge sort.