

Program Structures and Algorithms – Assignment 5

Spring 2023 (SEC – 3)

NAME: Sharun Kumar Kakkad Sasikumar

NUID: 002774079

Tasks:

1. Develop a parallel sorting algorithm to sort each partition of an array in parallel. Two schemes will be considered to decide whether to sort in parallel: a cut-off and recursion depth, the number of available threads, or an appropriate combination of these.
2. Conduct experiments using various values of the cut-off to determine a suitable value. If the partition's element count is lower than the cut-off value, use the system sort.
3. Create the ParSort class in the sort.par package of the INFO6205 repository. The Main class is ready to use.
4. Create a report demonstrating the experiment results and drawing one or more conclusions about the effectiveness of parallelizing sorting with this method. The experiments should use arrays of sufficient size for parallel sorting to be beneficial. Experiment with many different array sizes (large enough to justify parallel sorting) and cut-off schemes.

Relationship Conclusion:

It can be safely concluded, with consistent results from the experiment's graphs, that, when choosing the cut-off for parallel sorting using system sort, a $\frac{\text{cutoff}}{\text{array size}}$ ratio between 0.35 to 0.45 can be chosen (an average ratio of approximately 0.4) is preferred. Additionally, when choosing the size of the thread pool, it is most efficient to select the thread count close to the number of physical processors available to JVM.

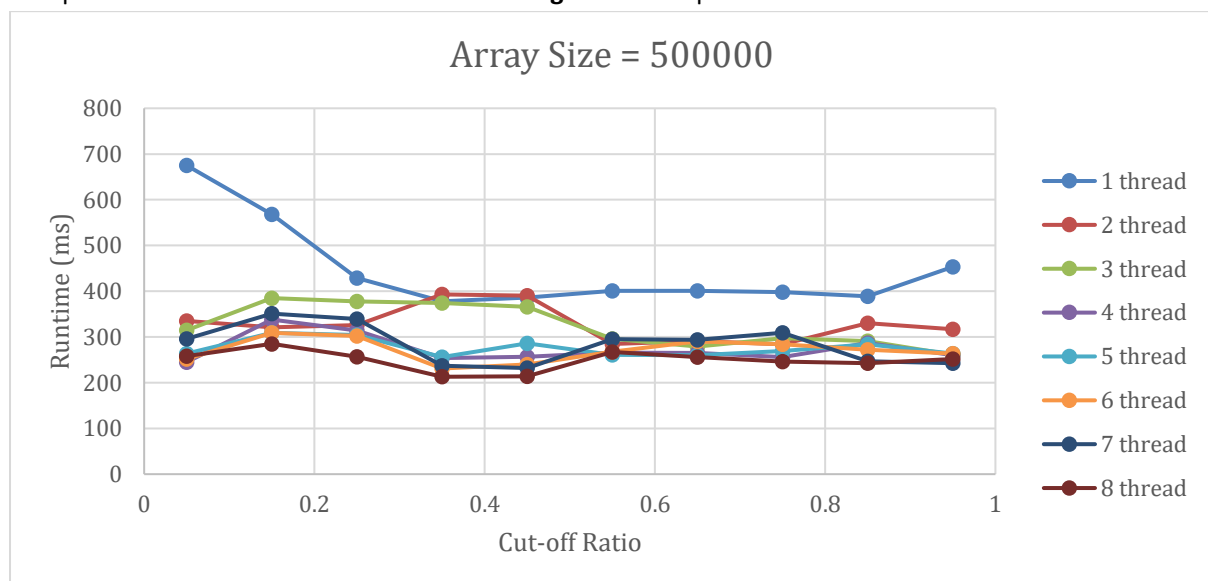
Optimally, $\frac{\text{cut off}}{\text{array size}} \cong 0.4$ AND thread pool \cong processor cores count

Code:

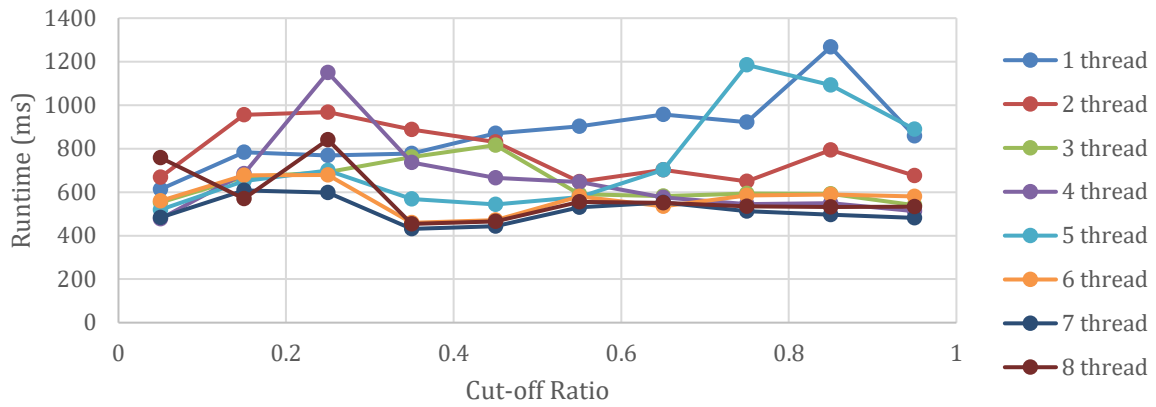
Submitted to GitHub Repository: <https://github.com/sharunkumar-ks/INFO6205/pull/4/files>

Graphical Representation:

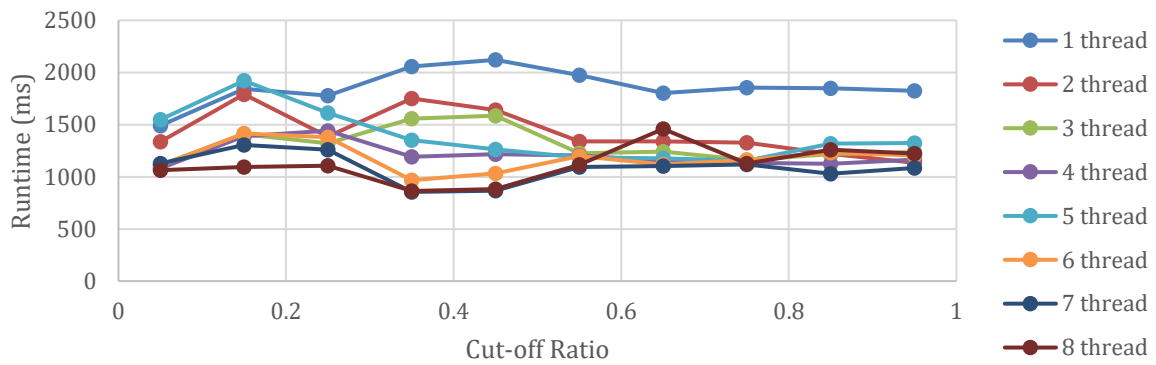
Complete data is available in the **Parallel Sorting.xlsx** file. Experiment was run on a machine with 8 cores.



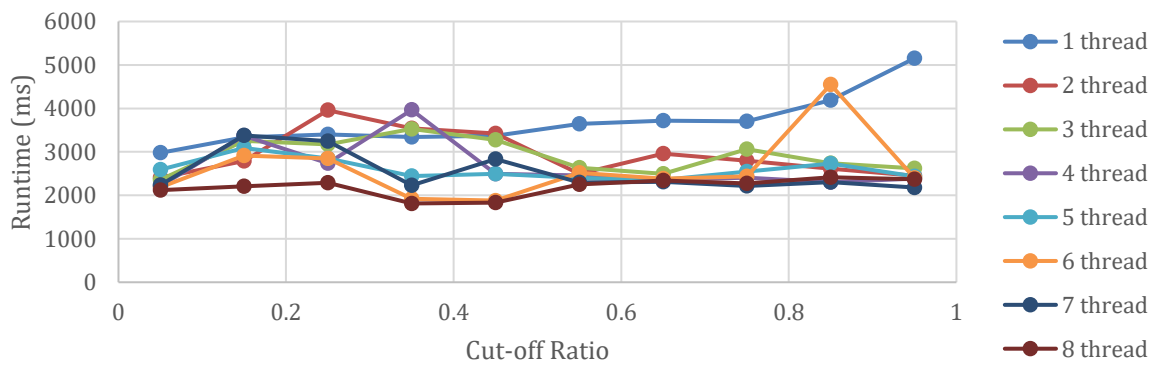
Array Size = 1000000



Array Size = 2000000



Array Size = 4000000



Array Size = 8000000

