Program Structures and Algorithms

Spring 2024

NAME: Gaurav Popat Gunjal

NUID: 002820367

GITHUB LINK: https://github.com/gunjalga/INFO6205

**Task: ASSIGNMENT 5 (Parallel Merge Sort)**

**Timings:**

**Array Length : 5000000**

For Sequential merge sort the time required to sort the array 10 times is 830ms

**Array Length : 3000000**

For Sequential merge sort the time required to sort the array 10 times is 425ms

**Array Length : 2000000**

For Sequential merge sort the time required to sort the array 10 times is 157ms

Graphical representation of timings at degree of parallelism = 8  
  
**A graph with numbers and dots

Description automatically generated**A graph with blue dots

Description automatically generated  
A graph with numbers and dots

Description automatically generated

A graph with numbers and a number of points

Description automatically generated with medium confidence

Graphical representation of timings at degree of parallelism = 4

A graph with numbers and a number of points

Description automatically generated with medium confidenceA graph with numbers and dots

Description automatically generatedA graph with numbers and dots

Description automatically generated

Graphical representation of timings at degree of parallelism = 2

A graph with numbers and circles

Description automatically generatedA graph with numbers and circles

Description automatically generatedA graph with numbers and a number of points

Description automatically generated with medium confidence

Graph of best time vs number of cores:

A graph with a line and a dotted line

Description automatically generated

**Conclusion:** Upon analyzing the graphs, it becomes evident that optimizing performance involves setting the cutoff value to N divided by the number of threads. As the thread count increases, performance shows improvement, but gains diminish gradually. However, this enhancement trend gradually plateaus, particularly noticeable on high-core CPUs, where further increases in thread count fail to yield significant performance gains.