

CSC 6220: Parallel Computing I: Programming

ECE 5610: Introduction to Parallel and Distributed Systems

Term Project

Fall 2022

Assigned: Monday November 21, 2022

Deadlines: Monday December 12, 2022 (5:30pm)

Requirements: Write an MPI program that implements a variant of the odd-even sort described in Section 6.3.5 of the textbook (pages 248-250). The initial array is partitioned into equal size sub-arrays which are distributed to the processes (one per process). You need to modify the algorithm to perform odd-even transposition iterations as long as the sub-arrays are changing.

The following is a high-level pseudocode description of the variant of odd-even sort algorithm that you need to implement:

Algorithm 1 Modified odd-even sort parallel algorithm

```
1: done = FALSE
2: while done = FALSE do
3:   {Perform original odd-even iterations}
4:   if received subarray is the same as the current subarray then
5:     broadcast TRUE to all processes;
6:   else
7:     broadcast FALSE to all processes;
8:   if all processes broadcast TRUE then
9:     done = TRUE
```

Test the program on 8 processes. The input array should consist of 128 random integers from the range $[0, 128]$. The array is generated at process 0 which is responsible for partitioning the array and sending the sub-arrays to the other processors. Process 0 will keep its corresponding sub-array, so that it can participate in the algorithm. At the end of the computation, process 0 collects all the sub-arrays and writes the sorted array into a file called `result.txt`. You should record the output of your program in a file called `output.txt` using the `script output.txt` command.

Compare the execution times for your implementation of the modified odd-even sort parallel algorithm with those of the standard odd-even sort (given in the textbook, section 6.3.5, pages 248-250) and the serial quicksort. For this performance comparison you should use 8 processors and randomly generated integer arrays of sizes: 2^{16} , 2^{20} , and 2^{24} . The random integers should be in the range $[0, 128]$. Produce a plot showing the execution times of the three algorithms. Produce another plot to show the speedup obtained by the modified parallel odd-even sort with respect to the sequential quicksort. Write a short (max. 2 pages) report describing the implementation and the obtained results. The report should be typeset using LaTeX. The plots should be generated using gnuplot or other similar plotting software.

Submission: Use the Canvas (Project module). You should submit a zip file containing the source of the program, the makefile, `result.txt`, `output.txt`, the jobscript file you used to

execute the program, and the report. The program should be compiled and executed on the WSU grid.