

HW1: Parallel Odd-Even Transposition Sort

Description :

In this homework, you are required to write a parallel odd-even transposition sort by using MPI. A parallel odd-even transposition sort is performed as follows:

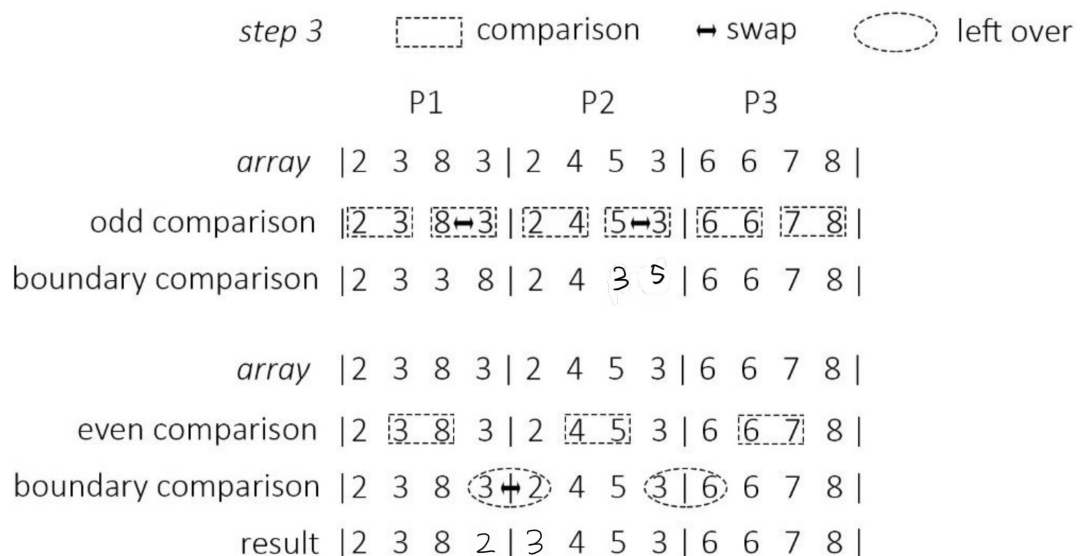
/ Initially, m numbers are distributed to n processes, respectively.*/*

1. Inside each process, compare the odd element with the posterior even element in odd iteration, or the even element with the posterior odd element in even iteration respectively. Swap the elements if the posterior element is smaller.
2. If the current process rank is P , and there are some elements that are left over for comparison in step 1, Compare the boundary elements with process with rank $P-1$ and $P+1$. If the posterior element is smaller, swap them.
3. Repeat 1-2 until the numbers are sorted.

You need to use MPI to design the program. The number of processors used to execute the program is n that is much less than m .

The following figure is an example to design your MPI program:

- **P1, P2, P3** are three different processes
- There are three operators that might need the communication among processes, i.e., *comparison*, *swap*, *left over* (boundary elements that should be compared).



Requirements

- The array data should be **randomly** chosen.
- You need to implement two versions of the tasks including a **MPI version** and a **Sequential version**.

- The implementations of both versions need to be submitted in separate files.
- You need to print the following information that identifies your **name**, **student id**, **assignment id**, **implementation version**, **running time of the whole program**. (see following as an example.)

```
Name: [redacted]
Student ID: [redacted]
Assignment 2, Mandelbrot Set, MPI implementation.
runTime is [redacted]
```

- You need to print out the 20-dims input array and the output array as running results of the submitted code.
- You need to specify the **command line** that you compile and run your program in your report.
- You need to **compare the performance** of different implementations or configurations in your report:
 - the number of cores used in the program
 - the size of the array
 - MPI vs Sequential
 - More if you have
- You need to include the two figures describing the structure of your MPI program and Sequential program individually.
- The report should be written in appropriate format which you could refer to a example report.

Where and What to Turn in Your Homework

- Please turn in a paper includes
 - Report
 - Performance analysis
 - The source code of your program
- Send your source code to blackboard, zip all your submitted files and names it studentID.zip
- No late homework assignment submission!!!

Due:

23:55, 10/06/2019