HW1: Parallel Odd-Even Transposition Sort

Description:

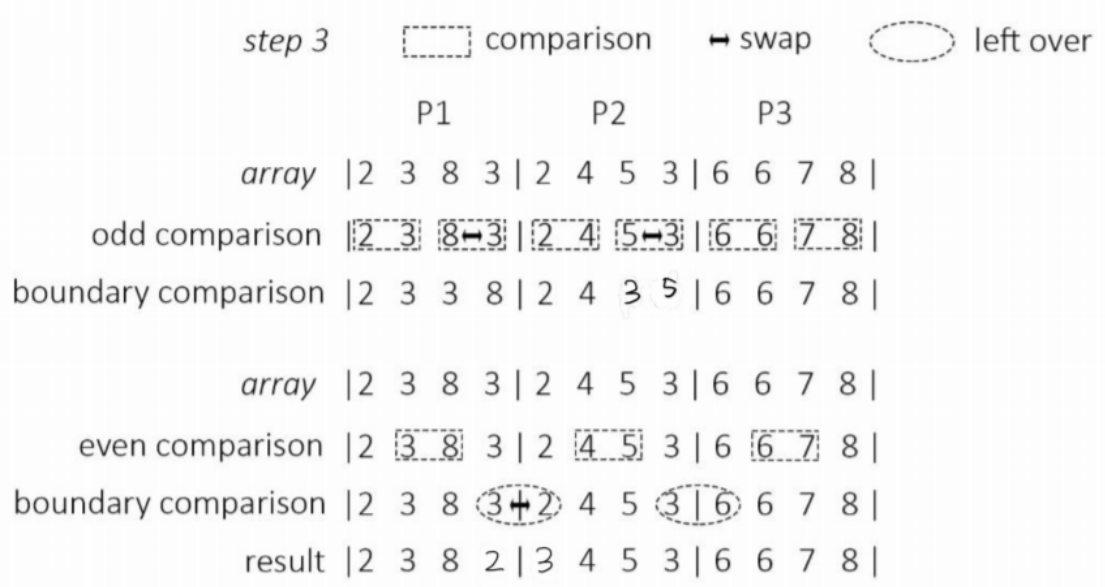
In this homework, you are required to write a parallel odd-even transposition sort by using MPI. A parallel odd-even transposition sort is performed as follows:

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

1. Insides 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 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, swaps 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 gure is an example to design your MPI program:

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



Requirement

- The array data should be randomly generated.
- 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 identies your name, student id, assignment id, implementation version, running time of the whole program. (see following as an example.)

- 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 MPI program. (Only 1-20 cores comparison is enough)
  - [ ] the size of the random generated array. (Small, Medium, Large)
  - [ ] MPI vs Sequential
  - [ ] More if you have
- You need to include a figure describing the flow chart of your MPI program
- The report should be in appropriate format, with a title page, introduction session to introduce the basic problem and task, method session to describe your parallel implementation, result session to compare performance under different configurations, and a conclusion session which concludes your experiment results.

Where and What to Turn in Your Homework

- Please turn in below several material:
  - Report
  - Source codes for you program for both sequential and MPI implementation
- Zip all your material and submitted them to BlackBoard. And make sure to name it studentid.zip
- No late submission is accepted.

Tips

1. Try to start doing your homework earlier. Since the current server has very limited resource, and everybody will wait in a queue to submit their jobs. If you start your homework too late, you may not be able to finish your experiments.
2. Do not run your program without submitting it using the qsub command. Although we use some tricks to forbid direct running, but some of you may find some ways to run it directly, and it will ruin other's student experiment data and you will not be able to get a valid experiment data.