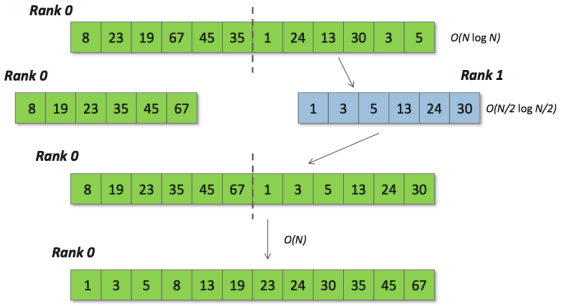
CS546 Parallel and Distributed Processing

Program Assignment 1

Submission:
Due by 11:59pm of 9/24/2019
Total points 100 - Late penalty: 10% penalty for each day late
Please upload your assignment on Blackboard with the following name:
CS546_SectionNumber_LastName_FirstName_PA1.
Please do NOT email your assignment to the instructor and/or TA!

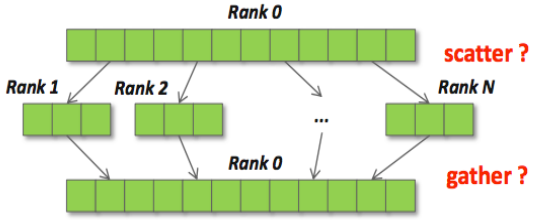
NOTE: for all the below assignment use the example from the lecture resource https://anl.app.box.com/v/2019-iit-mpi-lecture . Modify it and the whole package should compile.

Exercise 1:
We described the 2-processes sorting using MPI send/receive in the class. An example of parallel sort using MPI send/receive is:

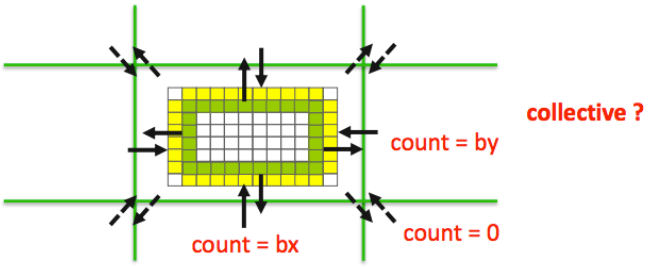


However, in the 2-processes version of sorting, only the first two processes are active, all other processes are waiting. In this exercise, please implement the sorting algorithm for an arbitrary number of processes. Hint: please start from blocking_p2p/sort_2_procs.c.

Exercise 2:
In question 1, we described a version of parallel sorting using send/receive APIs, rank 0 used to sends a chunk to every process. Finally, rank 0 received the sorted chunk from every process. In this assignment, please implement this algorithm using scattering and gathering collective APIs.



Exercise 3:
In the stencil example in the class, the algorithm used non-blocking send/receive calls for each direction. In this exercise, please Implement this algorithm using a single collective call(Alltoallv). HINT: please start from nonblocking_p2p/stencil.c



Exercise 4 (optional):
Please start from nonblocking_p2p/stencil.c, try to use derived datatypes with nonblocking send/receive instead of manually packing/unpacking. This is an optional exercise, there are no extra points for this question.

Running Code:
The whole package should compile and run as is.

What to turn in:
1. A written report of the assignment in PDF format. This is your chance to explain your approach, your optimizations, any insights gained, problems encountered, etc. Your report should include performance results for your solution.
2. Your source code and instructions to build it along with instructions to run the program.
3. Output of the file with timings of your testing. It should be consistent with your report.

Grading:

Out of 100 possible points we split the points as follows:
o Report: 40 points
- 10 points readability, use of English, organization
- 10 points for clarity of plots / figures
- 20 points for performance evaluation and analysis of results

o Source code: 60 points
- 10 points for readability and cleanness of code
- 10 points for given instructions and in-code comments
- 30 points for correctness (if it is not even compiling you lose all points)
- 10 points for performance and possible optimizations you have done

Note: We encourage collaboration between you and your classmates. Discuss various approaches and techniques to better understand the questions. However, we do NOT allow copying solutions or code. This is considered as cheating and falls under IIT code of honor. Penalties will be enforced. Please make sure you write your own solutions. GOOD LUCK!