COMS3008A: Parallel Computing
Exercises 7 & 8

2020-10-7

1 Objectives

- Apply the basic MPI functions to write simple MPI programs
- Compile and run MPI programs locally; compile and run MPI programs using MSL cluster
- Apply MPI point-to-point and collective communication functions to write MPI programs.

2 Problems

1. How do you launch multiple processes to run an MPI program? Compile and run the example codes in Lecture 7. Are the processes running the same code or the same tasks in these example programs?

2. Suppose the size of communicator `comm_sz = 4`, and **x** is a vector with $n = 26$ elements.

   (a) How would the elements of **x** be distributed among the processes in a program using a block distribution?

   (b) How would the elements of **x** be distributed among the processes in a program using a block cyclic distribution with block size $b = 4$?

3. Using the logical interconnect structure we set in token ring example, and using point-to-point communications only, implement an efficient broadcast operation that has a time complexity $O(\log p)$, instead of $O(p)$, where p is the number of processes in the communicator.

4. Program `sieve_mpi.c` gives an implementation of sieve of Eratosthens algorithm. In `sieve_mpi.c`, we are using a broadcast to send the next sieve k to all the processes. How would you replace this broadcast by using MPI point-to-point send and receive functions?

5. Complete the different MPI versions of the *numerical integration using trapezoidal rule* problem discussed in Example 5, Lec8 slides.

6. Implement the matrix-vector multiplication problem given in Example 6, Lec8 slides.

7. Write a simple program that performs a scatter operation that distributes the individual elements of an array of integers among the processes, each process modifies the value it receives, and then a gather operation is followed to collect the modified data where they are stored in the original array.

8. Write a simple program that performs an all-to-all operation that distributes $k(k \geq 1)$ individual elements of an array from each process to all processes (using `MPI_Alltoall` function).

9. Implement the odd-even transposition sort using MPI according to the parallel algorithm given in Example 9, Lec8 slides.

10. How would you implement quicksort using MPI (Lec8 slides)?

11. Suppose `comm_sz = 8` and the vector $\mathbf{x} = (0, 1, 2, \ldots, 15)$ has been distributed among the processes using a block distribution. Implement an allgather operation using a butterfly structured communication (Pg.37 (b), Lec8 slides) and point-to-point communication functions.