COMS4040A & COMS7045A: MPI Programming Exercise 1

2020-5-18

Objectives

- Apply the basic MPI functions to write simple MPI programs
- Compile and run MPI program on a cluster

Questions

1. In `sieve_mpi.c`, we are using a broadcast to send the next sieve k to all the processes. How would you eliminate this broadcast? Implement your idea based on `sieve_mpi.c`.

2. We made an assumption $n/p > \sqrt{n}$ in the sieve of Eratosthens example given in the class. Based on `sieve_mpi.c`, implement the sieve of Eratosthens without the above assumption.

3. Write an MPI program that computes the sum of an array of random numbers. Implement it using the following data distribution methods, respectively.
   - Block distribution
   - Cyclic distribution
   - Block-cyclic distribution

4. Write a simple MPI program in which two processes send packets of information back and forth a number of times and record the amount of time required (often called MPI **pingpong** program). The packets of information consist of an array of dummy floating point numbers which vary in length from one floating point number to 10,000 floating point numbers. These packets are sent back and forth from one process to another 100 times and the total amount of time required is recorded. From this timing data the average amount of time per send/receive is computed, as well as the transfer rate (bandwidth) in bytes/sec.

5. A small college wishes to assign unique identification numbers to all of its present and future students. The administration is thinking of using a six-digit identifier, but is not sure that there will be enough combinations, given various constraints that have been placed on what is considered to be an "acceptable" identifier. Write parallel programs both in MPI and OPenMP to count the number of different six digit combinations of the numerals 0–9, given these constraints:
   - The first digit may not be a 0.
   - Two consecutive digits may not be the same.
   - The sum of the digits may not be 7, 11, or 13.