CS563 Assignment 5: Programming with MPI

Instructor: Xinghui Zhao

Due: 11:59pm, February 23, 2020

1 Overview

The purpose of this assignment is to give you some practice on programming using MPI. The instructions use MPI+C, you can choose an alternative language which works with MPI, if you want.

1.1 MPI Installation

Follow the instructions below to install MPI on your computer (version numbers might be different):

1. Make sure you have `gcc` (C compiler) properly installed
2. Download the latest OpenMPI package `openmpi-3.0.0.tar.gz` at the following website: https://www.open-mpi.org/software/ompi/v3.0/
3. Unzip the tarball
4. In a terminal, navigate into the `openmpi-3.0.0` directory
5. Run the following command: `./configure --prefix=/usr/local`
   This step will take a few minutes and generates lots of outputs. It prepares the config.log file, which collects informations about you system, needed for the installation
6. Run the following command: `sudo make all install`
   This step installs the necessary packages for MPI
7. Test the installation by running `mpicc`, it should give you an error saying "no input files". If the system doesn't recognize the command, it means there is some problem with the installation

1.2 MPI Commands

1. Compile an MPI program

   `mpicc -o [executable name] [your mpi program].c`

2. Execute an MPI program

   `mpirun -np [number of processes] [executable]`

   You may use `man mpirun` to check all the options for the command.

2 MPI Applications

2.1 Revisit Roller Coaster

In Assignment 3 you have developed pseudo code for the Roller Coaster problem. In this assignment, you will implement your solution using MPI.

Suppose there are n passenger processes, and one car process. The passengers repeatedly wait to take rides in the car, which can hold C passengers ($C < n$). However, the car can go around the tracks only when it is full.

In your program, have passengers repeatedly take some number **R** of rides. Have the car take a fixed amount of time to go around the track, and have passengers delay for a random amount of time before taking another ride. Print a trace of the key events in your simulation.

2.2 Calculating Pi

We have discussed a sequential algorithm for calculating Pi in the class. Figure 1 shows the algorithm.

Your task is to parallelize this algorithm using manager-worker communication pattern, and implement it using MPI.

2.3 Submission

Submit the following on Blackboard:

```
npoints = 10000
circle_count = 0

do j = 1,npoints
  generate 2 random numbers between 0 and 1
  xcoordinate = random1
  ycoordinate = random2
  if (xcoordinate, ycoordinate) inside circle
  then circle_count = circle_count + 1
end do

PI = 4.0*circle_count/npoints
```

Figure 1: Sequential Code for Calculating Pi

1. Source code for each MPI application;
2. Sample output of each application.

3 Message Passing Application

In this section, you will use the message passing concepts and techniques we have learned so far to solve a more complex problem. For this application, you can choose any message passing model, use any programming language/library and you can implement this application using any communication patterns.

3.1 Stable Marriage

Let `Men` and `Woman` each be arrays of `n` processes. Each man ranks the women from 1 to n and each woman ranks the men from 1 to n. A `pairing` is a one-to-one correspondence of men and women. A paring is `stable` if, for two men m_1 and m_2 and their pared women w_1 and w_2, both of the following conditions are satisfied:

- m_1 ranks w_1 higher than w_2, or w_2 ranks m_2 higher than m_1;
- m_2 ranks w_2 higher than w_1, or w_1 ranks m_1 higher than m_2.

Expressed differently, a paring is unstable if a man and woman would both prefer each other to their current pair. A solution to the stable marriage problem is a set of n pairings, all of which are stable.

Write a program to solve this problem. The processes should communicate using message passing. The men should propose and the women should listen. A woman has to accept the first proposal she gets, because a better one might not come along; however, she can dump the first man if she later gets a better proposal. Your program should print a trace of key events as they happen. At the end, it should print the stable pairings.

3.2 Submission

Submit the following on Blackboard:

1. Source code;
2. A ReadMe file explaining how to run your program;
3. Sample output of the program.

4 Grading Scheme

This assignment will be graded out of 100. For your information, the grading scheme is shown in the following table.

Item	Percentage
Roller Coaster	30%
Calculating Pi	30%
Stable Marriage	40%