EE/CSCI 451
Spring 2019
Programming Homework 4
Assigned: March 1, 2019
Due: March 16, 2019, before 11:59 pm
Total Points: 50

1 Login to HPC

- The host is: hpc-login**3**.usc.edu
- Username and password are the same as your email account
- **Do not** run your program in the login node.
- After login, use the 'srun' command to run your program on a remote node. For example:
  srun **-n**4 ./run

1.1 MPI Examples

The "mpi_examples" folder includes the source codes used in discussions. To run an mpi program, for example, the 'scatter.c', follow the steps:

1. Login to HPC
2. Setup MPI toolchain: type 'source /usr/usc/openmpi/default/setup.sh'
3. mpicc -o run -O3 scatter.c
4. srun **-n**4 ./run

2 Pass Message in a Ring [20 points]

Write an MPI program that passes a value around 4 processes using the following steps.

1. Process 0 initializes $Msg = 451$ and prints value of Msg

2. Process 0 sends the value of Msg to Process 1

3. Process 1 receives the value of Msg, increases it by 1, prints the value and sends the current value of Msg to Process 2

4. Process 2 receives the value of Msg, increases it by 1, prints the value and sends the current value of Msg to Process 3

5. Process 3 receives the value of Msg, increases it by 1, prints the value and sends the current value of Msg to Process 0

6. Process 0 receives the value of Msg from Process 3 and prints the value

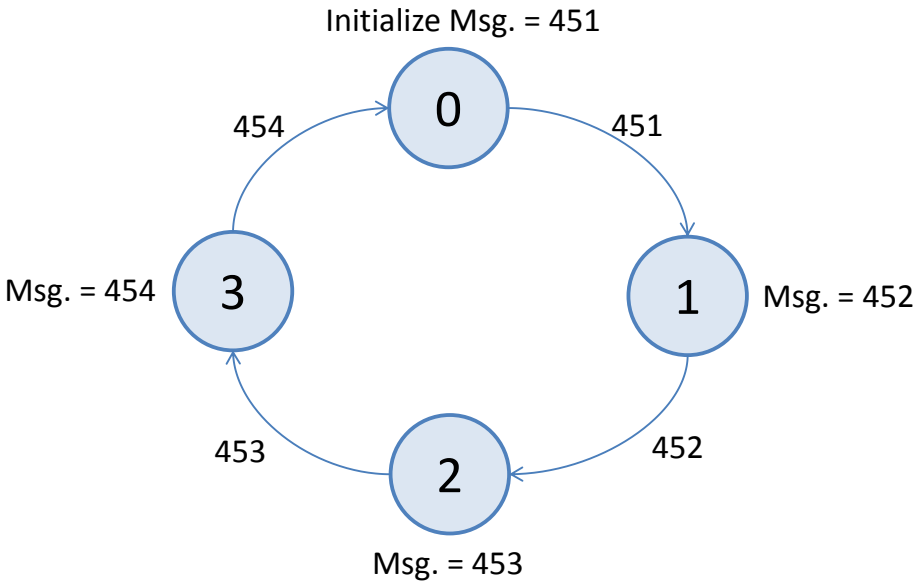

Figure 1: Example diagram

Name this program as 'p1.c'. Figure 1 illustrates the steps. The output messages look like:

- Process 0: Initially $Msg = 451$
- Process 1: $Msg = 452$
- Process 2: $Msg = 453$
- Process 3: $Msg = 454$
- Process 0: Received $Msg = 454$. Done!

3 Add 64 numbers using 4 processes [30 points]

In the "number.txt" file, you can find 64 numbers. Your task is to write an MPI program with 4 processes to compute the sum of these 64 numbers. There are 3 approaches:

1. Approach 1, name this program as p2_1.c:
   - Each process reads the entire *array*.
   - Do in parallel: Process 0 computes $\sum_{i=0}^{i=15} array[i]$; Process 1 computes $\sum_{i=16}^{i=32} array[i]$; Process 2 computes $\sum_{i=32}^{i=47} array[i]$; Process 3 computes $\sum_{i=48}^{i=63} array[i]$.
   - Process 1,2,3 send their partial *sum* to Process 0.
   - Process 0 computes the sum of all the partial *sum*s and prints it out.
2. Approach 2, name this program as p2_2.c:
   - Process 0 reads the *array*
   - Process 0 broadcasts the entire *array* to every process
   - Do in parallel: Process 0 computes $\sum_{i=0}^{i=15} array[i]$; Process 1 computes $\sum_{i=16}^{i=32} array[i]$; Process 2 computes $\sum_{i=32}^{i=47} array[i]$; Process 3 computes $\sum_{i=48}^{i=63} array[i]$.
   - Process 0 uses MPI_SUM reduction to sum these partial *sum*s.
   - Process 0 prints out the result.
3. Approach 3, name this program as p2_3.c:
   - Process 0 reads the array and scatters the entire *array* to every process using the **scatter** operation.
   - Each process sums up the portion of the *array* it receives.
   - Process 0 uses the **gather** operation to gather these partial *sum*s, computes the sum of all the partial *sum*s and prints it out.

4 Submission Instructions

- Code: 'p1.c', 'p2_1.c', 'p2_2.c', 'p2_3.c'. Make sure your program is runnable. (10+20 pts)
- Report: Write clearly how to compile and run your code. Screenshot of the execution time and performance on hpc. (20 pts)

You may discuss the algorithms. However, the programs have to be written individually. Submit the code and the report via ee451spring2019@gmail.com. Please make sure to include your name, student ID and the homework number in the PDF, and name your PDF file lastname_firstname_pa#.