COLLEGE OF ARTS AND SCIENCES
UNIVERSITY OF THE PHILIPPINES LOS BAÑOS

1/F Wing C Physical Sciences Bldg., Harold Cuzner Royal Palm Ave.
College 4031, Laguna, Philippines
☎ Phone +63 49 536 2313
ics.uplb@up.edu.ph www.ics.uplb.edu.ph

INSTITUTE OF COMPUTER SCIENCE

CMSC 180: Introduction to Parallel Computing
Second Semester 2022-2023

Laboratory Exercise 04
Distributing Parts of a Matrix over Sockets

Introduction

Write a program that will create an open socket for sending and receiving data. Allow the program to read from the command line the port number to open and listen to. Also, let the program read from a configuration file a list of IP addresses and corresponding ports to communicate to. Run several instances of the program over several terminals or PCs. If the instances of your program are run on the same PC, then that PC's IP address must be in the configuration file, and each instance must open and listen to different ports. If the instances of your program are ran on different PCs, then the IP addresses of these PCs must be in the configuration file together with a specific port. You define your own port. Among the running instances, elect one instance as the master while the rest as the slave. The master will create a matrix M and distribute the corresponding submatrices to the slaves. You can create two configuration files for each master and slave.

Exercise Specifications

<u>Research Activity</u>:

A. Write the main program lab04 that includes the following:

1. Read *n*, *p* and *s* as user input (maybe from a command line or as a data stream), where *n* is the size of the square matrix, *p* is the port number, and *s* is the status of the instance (0 for master and 1 for slave)
2. *If s = 0 (master), then:*
    a. Create a zero $n\ x\ n$ square matrix **M.** Assigned a randomized non-zero value (1-1000) to grid points divisible by 10 such (0,0), (0,10), (10,0), (20,0),(10,10).
    b. Read the configuration file to determine the IP addresses and ports of the slaves and the number of slaves t-1;
    c. Divide your **M** into *t-1* submatrices of size ***n/t-1 x n*** each, m_1,m_2,...., m_t;
    d. Take note of the system $time\_before$;
    e. Distribute the t-1 submatrices m_1,m_2,...., m_{t-1} to the corresponding t-1 slaves by opening the port *p* and initiating communication with the IP and port of each slave;
    f. Receive the acknowledgement "ack" from each slave, for all t-1 slaves;
    g. Wait when all the t-1 slaves have sent their respective acknowledgements;
    h. Take note of the system time $time\_after$;
3. *Else if s=1 (slave):*
    a. Read from the configuration file what is the IP address of the master;

b. Wait for the master to initiate an open port communication with it by listening to the port assigned by the configuration file;

c. Receive from the master the submatrix m_i assigned to it;

d. Send an acknowledgement "ack" to the master once the submatrix has been received fully;

e. Try to interpolate the submatrix assigned to the slave.

4. Obtain the elapsed $time\_elapsed = time\_after - time\_before$ in the server;
5. Verify that each of the t-1 slaves received the correct submatrix.
6. Run all instances within one PC only but on different terminals.

B. Fill in the following table with your time reading:

n (size of the matrix)	**t (master & slaves)**	**Time Elapsed (seconds)**			**Average Runtime (seconds)**
		Run 1	**Run 2**	**Run 3**	
1,000	2+1				
1,000	4+1				
1,000	8+1				
1,000	16+1				
4,000	2+1				
4,000	4+1				
4,000	8+1				
4,000	16+1				
8,000	2+1				
8,000	4+1				
8,000	8+1				
8,000	16+1				

*Note: **+1** in t is optional*

C. Repeat A and use the table below, but run all slave instances on different PCs. What happened?

n (size of the matrix)	**t (master & slaves)**	**Time Elapsed (seconds)**			**Average Runtime (seconds)**
		Run 1	**Run 2**	**Run 3**	
1,000	2+1				
1,000	4+1				
4,000	2+1				

4,000	4+1				
8,000	2+1				
8,000	4+1				

*Note: **+1** in t is optional*

D. Is your implementation efficient? Did you use any of the communication techniques discussed in the lecture? If so, what is it (one-to-many broadcast, many-to-many broadcast, one-to-many personalized broadcast, many-to-many personalized broadcast)? If not, why not?

Note that because of part C, your report must include 2 tables.

Lab Report Guidelines

Submit a report on your answers to the research questions posted in this exercise. All laboratory reports and term projects must be written in a technical way. That means each must have the following sections:

1. Introduction,
2. Objectives,
3. Methodology,
4. Results and Discussion,
5. Conclusion,
6. List of Collaborators, *(Yes, you can collaborate with other students but make sure that you can explain your work)*
7. References, and
8. Appendices.

You will include in the appendices the respective fully commented source codes of your programs. Submit your report through the Google Classroom Laboratory Exercise 04 portal.