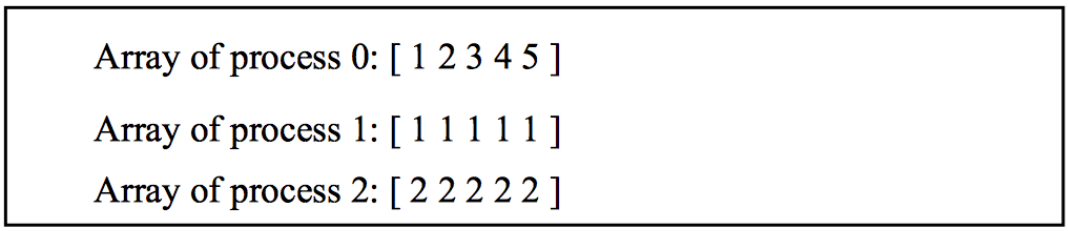
Homework 3: MPI Programing

Due: March 13, 2020

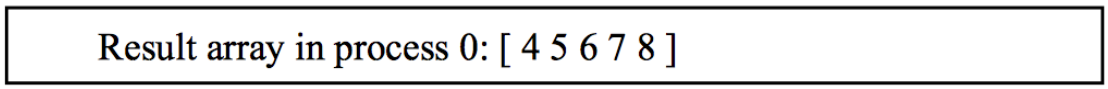
Background

MPI_Reduce() combines the elements provided in the *input buffer* of each process in the group, using the operation *op*, and returns the combined value in the *output buffer* of the process with rank *root*.

For example, there are 3 processes in a group, ranking from 0 to 2, and the process 0 is the *root* process, each process has an array, the length of the array is 5:



Use the array as input and use *MPI_SUM* as the operation of MPI_Reduce(), after doing the reduction, the result is:



Environment

The environment of this assignment is different from hw2. So, you must login the server with the original password at your first login.

You will need to run code on the cluster machines for this assignment.

The cluster consists of 4 machines you can use for this assignment.

Server IP Address: 166.111.68.163

Port: **2201** (different from hw2)

You can use `ssh {username}@166.111.68.163 -p 2201` to login the server.

Your username is **{your_student_id}** and original password is **{your_student_id}**.

For example, your username is 2018123123 and password is 2018123123 if your student ID is 2018123123.

The specification of each machine is:

Operating System	Ubuntu Server 16.04.1 LTS 64-bit
CPU	Intel(R) Xeon(R) CPU E5-2680 16 cores 1 thread per core
Memory	32 GB
C/C++ Compiler	gcc 7.4
MPI version	MPICH 3.2

The hostnames of 4 machines are: (maybe used when you want to test your own MPI program under different configurations)

node1 (the head node you log in) node2 node3 node4

Your HOME directory is shared among the 4 machines.

We have placed the starter code directory named ***bd_course_hw3*** under your own home directory on the server.

You can also extract the ***bd_course_hw3*** directory from the attachment in your local PC.

Description

There is a file named ***"reduce.c"*** in the code directory, which is a MPI program. In this program, each process has an array of integer, the elements of the array are generated randomly. The length of array ranges in [64, 1024, 16384, 262144, 4194304]. This program sums the element of the array into 1 array in process 0 using `MPI_Reduce()` in line 65 ~ line 74.

You should do the following tasks:

1. Use MPI_Send() + MPI_Recv() or MPI_Isend() + MPI_Irecv() to do the reduction with only single thread to sum the element of the array into 1 array. You should realize the reduction mainly in line 80 ~ line 83. Of course, you can add some variable or some other things as you want if needed. Make sure you do that correctly, because the program will check that. **Remember that you cannot use MPI_Reduce() when you realize it.**
2. Run the program using different amount of processes. You should record the runtime of MPI_Reduce() and "your_Reduce" implemented as before under different conditions (different amount of process & different length of array), and fill in the blanks in **"hw3_reslut.xlsx"**. We provide a script called ***run.sh*** in the code directory for you to test your program performance. After you run the script successfully, you will get all the data needed to filled in "**hw3_result.xlsx**".

Hint: You can try different optimization strategies and compare the performance. Try to optimize your reduction with only single thread as far as possible to make it faster. You can describe your different strategies in detail in your report.

If you don't want to use "reduce.c", feel free to write your own version.

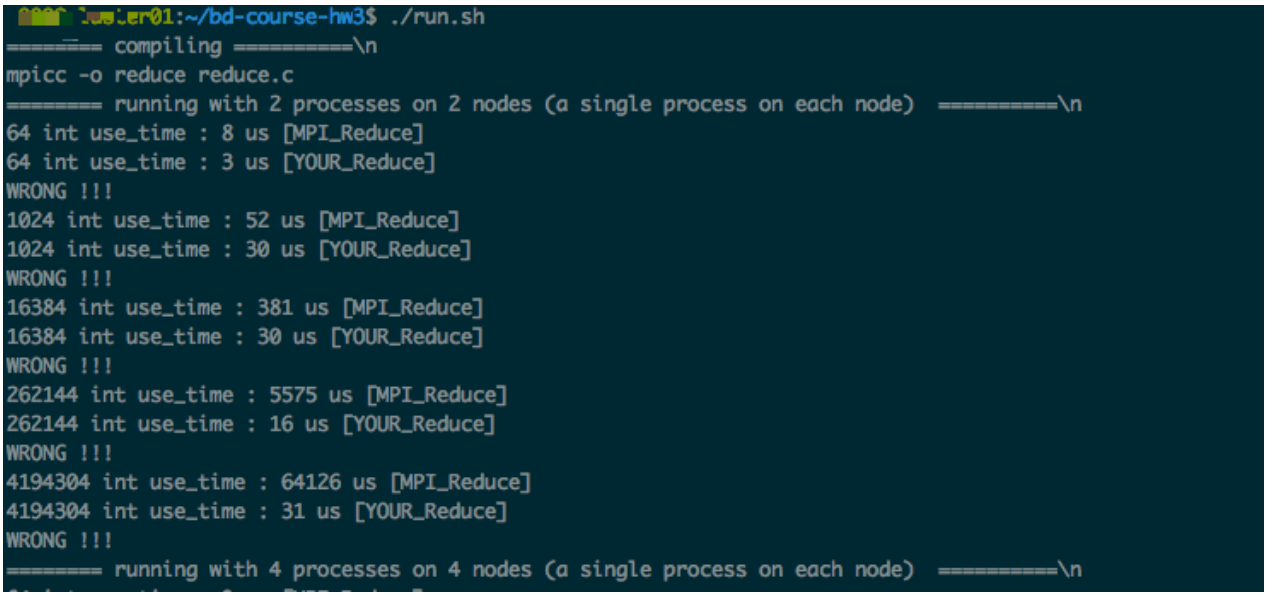
How to run the code

The code is implemented in C.

We provide a script called ***run.sh*** in the code directory for you to compile and test your program performance.

./run.sh

The result may like before:



We also provide a Makefile in the code directory for compiling. You can use **make** to compile code and if there is no error, you will get a ***reduce*** program.

You can use ***mpirun*** if you want to test your program under other conditions.

Remember:

You cannot modify file called host1, host2 or run.sh (when you submit the final version).

You should edit and compile your code on `node1` which is the server you login.

Hand-in

Please submit your code, your testing result and your report.

Please describe your implementation in detail in your report.

Before submitting the source files, make sure that all code is compilable and runnable! We should be able to simply make, then execute your programs without manual intervention.

Note

If you have any questions about this assignment, please contact me:

Email: lincheng96@qq.com