- Middle East Technical University Department of Computer Engineering

CENG 478
Introduction to Parallel Computing

Spring 2019-2020

Assignment 1

Due date: 11.03.2020, 23:59

1. Hello World (0 Pts)

   a. The code is already implemented to show you how a parallel program can be executed.

   b. It is advised to read Hw0.pdf and test helloworld.c on SLURM before starting Hw1.

2. Parallel Maximum (40 Pts)

   a. You will implement a parallel algorithm that finds the maximum number in an array of float precision numbers using MPI. Your program should take the input file path as a string on argv[1], and input size as an integer on argv[2]. (i.e. for the default input "./input.txt" 1000000)

   b. Let n denote the length of array of float precision numbers, p the number of processes and p_i the i^{th} process. Assume that p is even and the array is distributed among the processes and each have almost the same amount of elements ($\cong n/p$).

   c. In the first part of the algorithm, each processor finds the maximum of its own segment of the array independently. For the second part which involves inter-process communication, the algorithm initially divides the processes into pairs and one process from each pair sends its maximum to the other, and the other compares the recieved one with its own max and decides which is the maximum. The same procedure is applied repeatedly in several stages until the result is obtained in one process. This part of the algorithm is illustrated in Figure 1.

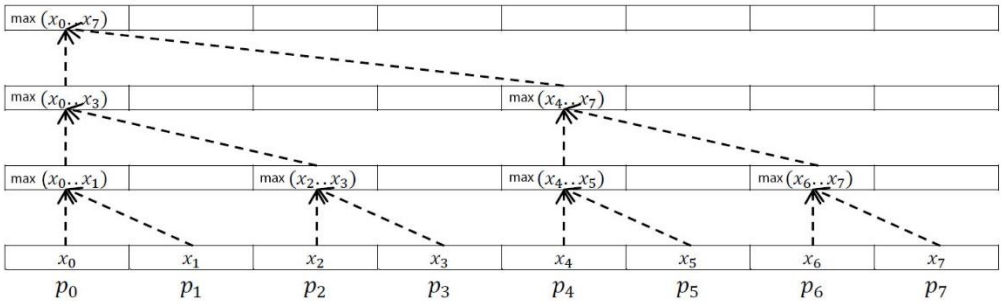

Figure 1: Overview of the algorithm x_i denotes the maximum found by p_i.

   d. Hint: You can implement the above algorithm using MPI_Send, MPI_Recv and MPI_Reduce methods.

3. Tests and Report (60 Pts)

a. Comment on the complexity of the above method. What is the improvement over a method that simply compares all the elements in a sequential process.

b. State your design choice. (Which MPI functions did you use and why?)

c. Your programs should print the following two lines:

Max_Value
Total_Time

d. To measure the time, you can use MPI_Wtime. Note that using MPI_FILE operations are not expected to increase performance because of the limitation of physical discs. You can read input with standard ANSI C functions on process 0 and distribute values on other processes. Just measure starting time after process 0 sends the values.

e. Run your implementation using 1, 4, 8, 16 number of MPI processes. Plot a "Time vs. Number of processors" graph. Plot a "Speed Improvement vs. Number of processors" graph. Comment on how the time and efficiency changes. What is the cause of this increase or decrease? Speculate on what can be done to improve the performance further.

4. Notes

a. You will submit a tar file consisting of your code file(s), your makefile, a **pdf** of your report and the outputs of your executions (i.e. 1_out.txt, 4_out.txt, 8_out.txt, 16_out.txt) via ODTÜClass. Note that the **comments and comparisons** on your report will **have a large effect on your grade**.

b. Note that **zombie processes** can pile up very quickly as you are trying to learn a new way of programming. Try to **control frequently** if there are any, with *squeue* command and kill them with *scancel* command to ensure that Slurm serves all the users **properly**.

c. Try to finish as early as possible. Since all of you work on the **same** HPC, the waiting times for the **queue** can be **huge** which will prevent you from testing your code **efficiently**.

d. Implementing MPI macros **does not mean** your code works in **parallel**. You have to design your algorithm so that it really works in parallel. For those who submits code that does not work in parallel will get **no credits** from this assignment.

e. You can still submit your work if the **deadline** is passed, however with an increasing **penalty** of **5*days*days**. (i.e. first day -5 points, second day -5*2*2=-20 points and so on). Note that even a minute late means that it is the other day.

f. We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations and will get zero.