Middle East Technical University Department of Computer Engineering

CENG 478

Introduction to Parallel Computing

Spring 2018-2019

Assignment 3

Due date: 05.05.2019, 23:59

1. Matrix – Matrix Multiplication (50 Pts)

   a. You will implement a parallel algorithm that multiplicates two matrices in parallel using MPI processes and comment on your results

   b. Construct two matrices with length 3,200 x 3,200 A and B, where

   $$a_{i,j} = \frac{i+1}{j+1} \text{ and } b_{i,j} = i + 1 - \frac{j+1}{3200}$$

   c. Implement a parallel algorithm to multiplicate these two matrices. Find $C = A \cdot B$. Calculate and print the time consumed with MPI_Wtime(). Also print the Frobenius norm of the result. Frobenius norm of an m x n matrix C is defined as:

   $$||C||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |c_{i,j}|^2}$$

2. Tests and Report (50 Pts)

   Compile and run your code, calculate the time consumed for 1,2,4,8,16 and 32 processes on the given sample input. 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 performance further.

3. 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, 2_out.txt, … 16_out.txt, 32_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**[1].

   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.