Midwestern State University
Parallel Programming – Fall 2021 | Assignment # 1

Team Assignment (**You are allowed to work in teams of two**)

Step 1:

TO BE SOLVED, TESTED AND GRADED IN STAMPEDE2

Read how to compile and execute sequential programs in Stampede2.

- Write a sequential program that will visit each one of the locations of a 640000 integer array and will compute the addition of all numbers in this array. (*You will not receive credit for this code, however you will lose 50 points if not delivered*).
- The content of this array is pretty much the first 640000 consecutive numbers, starting at 1.
- Read about how to time routines in Linux, and time the section where the summation is executed, this will be your serial time. (Explained in class)
- You can verify your result with the Gauss Technique (google it).
- Read the TACC guide and familiarize with the compilation of serial code.
    - This command must be at the TOP of your code. In simple words need to tell me how to compile your code on stampede.
- Must provide *.c file and script for sequential code.
- Make sure that you name your script accordingly.
- File name: NameLastNameSerial.c
    - The first two lines of this program must be your names and last names
    - The second line must have the command(s) that I need to type in TACC to compile & execute your program

Step 2:

[70 points]

For your first parallel program, you need to write an MPI version that will compute the summation of all the first consecutive 640000 numbers (same problem statement as above), however this time you will use:

a) 4 nodes, 64 cores, time it (use MPI_Wtime)
b) Only MPI_Send and MPI_Recv allowed for this solution

- Must provide *.c file and script for parallel code (V1)
- Make sure that you name your script accordingly.
- File name: NameLastParallelV1.c
    - The first two lines of this program must be your names and last names
    - The second line must have the command(s) that I need to type in TACC to compile & execute your program

Your program must fulfill the following conditions.

AT PROCESS ZERO

1. Process zero, is the only one process that owns the array of samples, nobody else owns the 640000 samples

2. Process zero will fill the array of samples with numbers as follows:
    a. Location [0] = 1
    b. Location [1] = 2
    c. :
    d. Etc until you reach the last location
3. Process zero will then distribute the piece of information (NOT THE WHOLE ARRAY) that each one of the other processes need to compute their corresponding summation
4. Process zero must be also the ONLY ONE process that owns an array named "Summation". The content of this array will be determined as follows:
    a. Location zero of our summation array has the local summation computed by P0
    b. Location one of our summation array has the local summation computed by P1
    c. Location two of our summation array has the local summation computed by P2
    d. Etc
5. Once all other processes have sent the info to process zero, process zero will compute the final addition, and will send a message via IO to the user telling him/her
   The summation of all numbers is: XXXXXXXXXX

AT THE OTHER PROCESESS

1. Compute the summation for their corresponding section
2. Will send the result to the right location at P0

Step 3:
[30 points]

Provide a MS document with the times for

a. Sequential
b. Parallel [MPI_Send and MPI_Recv]

Make sure that the times are in the same scale so that we can compare them easily and quickly see which one is larger, smaller, etc.

It should be clear that when executed, the code should report these values. If the values are not reported by your code, then they do not exists and the values of your word document will be ignored and not validated.

Deliverables:

- Assignment Due: Wednesday September 22, at 8:00 am via d2L.
- Must deliver hard copy of all your code at the beginning of class (-30 if not delivered on time)
- All pages must be stapled in the following order.
    - Serial Code
    - Script for Serial
    - Parallel Code (V1)
    - Script for V1

- (-20 if not respected)
- If your code does not compile → zero points
- If your code crashes → zero points
- If your code is not properly timed → -20 points
- If serial code is not delivered → (-50)
- If Word document with times is not delivered → (-25)
- If I cannot compile or execute your program because files are missing → zero points. I am not adding anything to run/execute your assignment. It is your responsibility to give me everything that is needed to successfully run/excute your program.

Anti-procrastination rule:

1) Questions associated with the assignment are very welcome. Please familiarize with my office hours and approach me ASAP. No, I will not write your code. It is your assignment.
2) No, I will not use your flash drive, be prepared to share your screen via zoom.
3) 24 hours before the assignment is due, I will stop providing feedback, tips, hints. No Exceptions.