CS5105 Parallel Computing

Assignment 3

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
Deadline: 15th November 2020 11:59PM

Roll number: ____________________ Name: ____________________

Question:	1	2	3	4	5	6	7	8	9	10	11	Total
Marks:	1	1	1	1	1	1	2	2	1	2	3	16
Score:												

Instructions:

- Unless specified in the question, the programs should be able to run on any arbitrary number of processes/threads.
- Unless specified in the question, use a random number generator to initialize the required arrays/matrices.

1. (1 mark) Write an MPI program to compute the sum of an array of elements. Computation load should be balanced across processes. Input: Number of elements in the array. Output: Array sum (Print only once).
2. (1 mark) Write an MPI program to exchange an array of elements. Each process has an array of N elements. Rank 0 sends data to Rank 1, Rank 1 to Rank 2, ..., Rank n-1 to Rank n, and Rank n to Rank 0. Input: Number of elements in the array. Output: Array elements of all processes.
3. (1 mark) Write an MPI program to compute the sum of an array of elements. Assume that the array cannot be stored on a single process i.e., you cannot have int array[N] in one process, and then send the data to other processes. Initialize the array with its index values i.e., array[i] = i. Distribute the array such that computation work across processes is balanced. Input: Number of elements in the array. Output: Array sum (Print only once).
4. (1 mark) Write an MPI program to compute the sum of an array of elements. Assume that the array cannot be stored on a single process i.e., you cannot have int array[N] in one process, and then send the data to other processes. Initialize the array with its index values i.e., array[i] = i. Distribute the array such that the amount of computation work across processes is balanced. Your program should support block distribution, cyclic distribution, and block cyclic distribution of the array. Read about distribution of data. Why do you think

various distribution schemes are required? Input: Number of elements in the array. Array distribution = block, cyclic, block cyclic. Output: Array sum (Print only once).

5. (1 mark) Write a threaded program (pthread/any thread library) to compute the sum of an array of elements. Can you write the program without using locks? Input: Number of elements in the array. Output: Array sum (Print only once).

6. (1 mark) Write an OpenMP program to compute the sum of an array of elements. Input: Number of elements in the array. Output: Array sum (Print only once).

7. (2 marks) Write an MPI+OpenMP program to compute the sum of an array of elements. Use OpenMP to compute the local_sum, and MPI to compute the global_sum. Input: Number of elements in the array. Output: Array sum (Print only once).

8. (2 marks) Write an MPI program to compute the product of two vectors i.e.,

   $$V^T V$$

   . Input: Size of the vector. Output: Scalar value (Print only once).

9. (1 mark) Write an OpenMP program to multiply two matrices. Input: Matrix_1 size: m X n Matrix_2 size: n X p. Output: Array sum (Print only once).

10. (2 marks) Each process has an array of N elements. Start at Rank 0. Generate a random number. Compute X = (random_number) % (number of processes)). Send the array of data from Rank 0 to Rank X. At Rank X, generate a random number and compute new_X using the same formula. Send data from Rank X to Rank new_X. Continue this process for m iterations. If you think such a program cannot be written, explain the reasoning behind it as comments in the code file. Input: Number of elements in the array. Iteration count, m. Output: Array elements of all processes.

11. (3 marks) Implement iittp_barrier() in MPI. The barrier should work both across processes and threads within a process i.e., all threads across processes should synchronize after the call. Do not use MPI_Barrier().

Space for rough work/graffiti/doodles