COSC 462 Fall 2024 : Programming Assignment 1

Date: Sept 20, 2024 **Total points: 30** **Due: 6:00 PM, Sept 27, 2024**

1. Point-to-Point Communication:

Using calls to MPI_Send and MPI_Recv, write a code that performs the following:

1. (5 points) Let each processor initialize a variable with a unique integer. Each processor should send this integer to its right neighbor (wrap around the boundary processors, that is, processor 0 and processor $p-1$ are topological neighbors). Each processor which should then print out it's rank and the value of the integer it received. Repeat with left neighbors. Use 8 processes (MPI ranks) for this problem.

2. (5 points) The cost of communicating a message of size m between two processors can be modelled as:

   $$T_{comm} = O(\tau + \mu m)$$

   Using different message lengths (use large message sizes - don't use one integer or float or double), make a plot of T_{comm} vs. m. Since each node has 48 cores, start the job on two nodes, that is, 96 ranks mapping each rank to a CPU core but communicate only between rank 0 and rank 95. ONLY use MPI_Wtime to measure the communication time. From your data, evaluate the constant τ and μ for the parallel architecture you ran your executions on.

2. Collective Communication: Let each process initialize a variable with a unique integer. Use 8 processes (MPI ranks) for this problem.

1. (5 points) Gather all the integers on processor with rank 0 and print out the values from rank 0.

2. (5 points) Gather all the integers on all the processors and print out their values from each rank.

3. (5 points) Use an MPI reduction operation on the integer on each processor to sum them all up on processor with rank 0 and print out the sum on rank 0. Repeat but this time print out the product of all the integers on rank 0.

4. (5 points) Use an MPI reduction operation on the integer on each processor to sum them all up on *all* processors and print out the sum on each rank. Repeat but this time print out the product of all the integers on *all* ranks.

In each case, the output to the screen should be identifiable by the rank of the processor which printed it out.