WITS UNIVERSITY

School of Computer Science and Applied Mathematics
UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

COMS4040A & COMS7045A: MPI Lab 3

May 19, 2025

Objectives

- Learn how to use MPI collective communication functions and MPI derived datatypes.
- Design and implement parallel algorithms for some common problems using the MPI point-to-point and collective communication functions.

Programming

1. Write MPI programs to implement `MPI_Allreduce` in two different ways, that is, a tree structured global sum followed by broadcast, and butterfly structured global sum (see `Lec11` slides). First write your programs for the special case in which `comm_sz` is a power of two. Can you modify your program so that it will handle any number of processes?
2. Implement matrix-vector multiplication using a block-column distribution of the matrix. You can have process 0 read in the matrix and simply use a loop of sends to distribute it among the processes. Assume the matrix is an $n \times n$ square matrix and that n is evenly divisible by `comm_sz`.
3. Learn how to distribute and collect data across multiple processes using `MPI_Scatter` and `MPI_Gather`. To practice, write an MPI program that performs the following:
   (a) The root process (rank 0) starts with an array of N integers, where N is divisible by the number of processes.
   (b) Use `MPI_Scatter` to divide this array into equal parts and distribute them to all processes.
   (c) Each process multiplies each of its received elements by 2.
   (d) Use `MPI_Gather` to collect the modified subarrays back into a single array on the root process.

4. Write a simple MPI program based on Example 4 discussed in `Lec12` slides that defines an MPI derived datatype for the Particle struct, then use the root process to initialize an array of particles with random values for mass and position, and finally distribute the particles evenly among the processes.