Tools of high performance computing 2020

Exercise 9

Return by Wednesday 25.3.2020 23:59 to Moodle.
Exercise session: **Cancelled.** You can consult the exercise assistant and lecturer using Moodle chat channel during session times or Moodle discussion forum.

Problem 1. *(6 points)*

We have *n* processes. Each process initializes an $n \times n$ matrix with zeros, except for the diagonal elements of the matrix that are initialized with the process's rank number. Each process sends to rank 0 an array containing all the elements of its diagonal. Process 0 overwrites the array sent by process *i* on the *i*-th column (row if using C) of its local matrix. At the end, process 0 prints its final matrix, on which each element should be the number of its row (or column).

In order to send a diagonal, a proper vector datatype should be created and set for reading the diagonal elements of a matrix with the right displacement and stride. When you are communicating to rank 0, keep in mind that you are sending a single vector datatype, but you want to receive an array of *n* elements, that have to be stored contiguously in its matrix row (or column).

Problem 2. *(18 points)*

Write an MPI program that iterates values $x_{i,j}(t)$ at points in a 2D grid. The iteration formula is

$$x_{i,j}(t+1)=\frac{1}{8}\left[x_{i-1,j}(t)+x_{i,j-1}(t)+x_{i+1,j}(t)+x_{i,j+1}(t)\right]+x_{i,j}(t)\ .$$

Initial condition at $t=0$ is $x_{i,j}(0)=i+j$. Use cartesian topology in the communication. You may assume that the processes form a square grid and that the computational 2D grid is evenly divided by the process grid. I.e. you have n_{p} processes and the 2D grid has the size $m\times m$. Then $n_{\mathrm{p}}=p\times p$ and m/p is an integer.