ICHEC
Irish Centre for High-End Computing

PRACE Course: Intermediate MPI

Day 2

10/11/2022

1 Cartesian topology

Consider a n by m matrix with integer elements. Write a code that does the following:

1. the matrix is split in $p \times q$ blocks with each block distributed on a process, see fig. 1. (p·q=number of processes.)

2. create a Cartesian topology that matches the above block distribution. Use MPI_Dims_create for the division of processors in the grid and periodicity in both dimensions.

3. for each process find its Cartesian neighbours.

4. allocate submatices and initialise with random values of 0 and 1 at each process. Be sure that running the code on 1 process or on more, with the same starting seed, results in the same random distribution for your matrix. **Hint:** let process 0 generate the data for each sub-matrix and then send it to the right process. An alternative is to put the random number generator in the proper state in each process.

5. add a function that computes for the sum of all elements in the local block.

6. reduce this in the process 0.

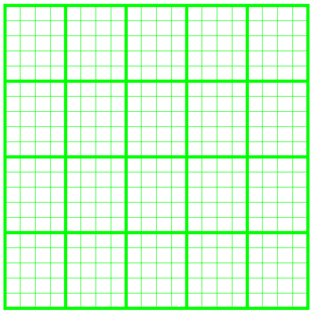

Figure 1. A matrix and its sub-blocks. Each sub-block shall be associated to a process from the Cartesian topology.

2 Derived datatypes

Create derived datatypes which help one pass around a matrix column, matrix row and a sub matrix. The code will contain:

1. Use dynamic allocation for your original matrix. In C be careful how you allocate the matrix. You need a contiguous block of memory for it.

2. In order to create a new type for each of the three situations. (use MPI_Type_vector)

3. Write a ping-pong program but use the derived data types as the message.