Laborator 4

Programare paralela si concurenta

Ciprian Paduraru

(Stencil computation)

Read “Game of life” project description – page 1 of MPI_game_of_life_project.pdf file.

1. Create an iterative program for running “game of life”. To reuse the code, implement a class named “GameOfLife” with the following functions:
    - Initialize(const float lifeProbability, const float& Rect rect) – random initialization within rect, each cell is alive with probability = lifeProbability.
    - Update(const float& Rect rect) – updates a new cycle according to rules
    - DebugPrint(const float& Rect rect) prints the table at console within rect

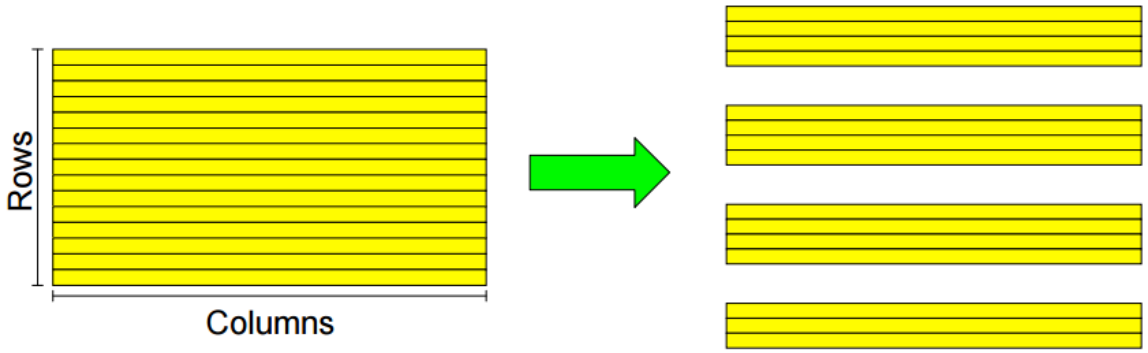
    Notes:

    - Use a define or template for the number of rows/columns.
    - Beware that you should use a double buffering mechanism – you can’t update any cell until all read operations ends (even in the serial implementation).

    Start with a small board like 4x4, and small number of steps to debug correctly.

    Check the implementation provided in GameOfLife.h and its usage in 1.cpp.

2. Use a 2D decomposition row blocks for splitting tasks between processes.



Notes:

- For simplicity, reuse the class created at 1. You can do initialization in parallel OR send initial table from master.
  Every process can have it's own instance of GameOfLife representing a single data block.
- Because an intermediate block needs a row above and a row below to compute correctly, you have to create 2 ghost lines for each process.

  We'll denote ghost lines by 0 and NUM+ 1 while the usual block lines with indices in interval [1,NUM]. Each process P (except first and last – which will execute just a half of this) should execute the following before Updating:

  ISend(Row[1], P-1);

  ISend(Row[NUM], P+1);

  IRecv(Row[0], P–1);
  IRecv(Row[NUM+1], P+1);

  Note that you should use asynchronous operation for avoiding deadlocks!

Check the implementation provided in 2.cpp.

3. Improve implementation of 2 such that communication of rows exchange and computation of some part of the table are overlapping.

   Check the implementation provided in 3.cpp.

4. Virtual Topologies in MPI

   Discussion about different topologies available in MPI. For more details:

   - Graph topologies: Check topologies_carthesian.pdf file in this folder.
   - Carthesian topologies: Check topologies_graph.pdf.
   -

5. Parallelize game of life using a 2d grid topology in MPI.