Assignment 3: OpenMP and MPI

Background:

The **Game of Life** is a cellular automaton devised by John Conway. The "game" is actually a zero-player game, meaning that its evolution is determined by its initial state, needing no input from human players. It is played on a grid where each cell can be either **alive** or **dead**. The state of each cell evolves in discrete time steps based on the a set of rules, for more details see [1].

Requirement

- Implement Game of Life as described in **on a** grid of given size with a border of permanently dead cells. Parallelize it using OpenMP and/or MPI.
- Code should be of decent quality: documented, commented, proper error handling, proper use of the C language
- To verify the correctness of your implementation you are encouraged to use several verification patterns, see below.
  - For all benchmark executions of the code the board should be of size 3000x3000. You should start with the 'grower' pattern (see below), where cell (0,0) of the pattern should be at cell (1500,1500) of the board. You should then run the code for 5000 iterations.
  - Write a report containing the full code of the implementation, a motivation of your chosen implementation, data distribution and communication methods, the population size at the end of a benchmark run, a speedup curve for at least 1, 2, 4, 8, 16, 32, 64, and 128 processes/thread, and a discussion of this speedup curve

You are free to use as many nodes of the snellius cluster as you can get. The use of GPUs is NOT permitted.

Implementation hints

On Canvas I provide a set of header files [download link] with different cell patterns of increasing complexity. They are used to ensure that your implementation of the Game of Life rules is correct. You can do a '#include' of the patterns to incorporate them into your code.

- Verify that the 'beehive' pattern has a population of exactly 6 in all generations
- Verify that the 'glider' pattern has a population of exactly 5 in all generations until the pattern leaves the board

[1] https://conwaylife.com/wiki/Conway's_Game_of_Life

- Verify that the 'grower' pattern has a population of 49 at generation 10, and a population of 138 at generation 100.
- There are a number of programs and websites to help you visualize the Game of Life. The one I use is https://golly.sourceforge.net. The file set on Canvas also contains .rle files of the patterns. Most Game of Life programs, including Golly, can read these .rle files.

NOTE: As mentioned before, code should be of decent quality. Emphatically part of this is handling of error conditions from MPI function calls, `` `malloc()` `` calls, file opening, and other functions that can return errors. In the context of this assignment, error handling can be as simple as printing a warning and stopping the program. we will deduct points for missing error handling.

NOTE: You will be running the experiments on the snellius cluster. Such clusters are heavily used, so it may take some time before your job is run. Keep this in mind while observing the deadline. Start early enough to do all the runs you need. Early development can also be done on your own computer.

Submission:

Submit a short report in Canvas before the deadline, including the graphs and answers to the questions.

Note: Submitting after the deadline will result in losing points (1 point for every 30 minutes after the deadline)

Code and Scripts:

Provided as tar file in canvas.