CSC 6740, spring 2019
Programming Assignment 6
Assign date: April 2, 2019, Due: April 18 2019 11:30 pm

The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square *cells*, each of which is in one of two possible states, *alive* or *dead*. Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

1. Any live cell with fewer than two live neighbors dies, as if caused by under-population.
2. Any live cell with two or three live neighbors' lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by overcrowding.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

The initial pattern constitutes the *seed* of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed, births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a *tick* (in other words, each generation is a pure function of the preceding one). The rules continue to be applied repeatedly to create further generations.

For infinite two dimensional grid you can assume that the grid is wrapped around in both x and y direction. If the grid dimension is **NxM** then **cell[i][0]** is adjacent to **cell[i][M-1],** similarly **cell[0][j]** is adjacent to **cell[N-1][j]**.

Write a program, which implements a hybrid MPI+OpenMP version of the game of life. Your program should take five command line arguments. It should take the name of the input grid file as 1st command line argument, the number MPI processes and number of threads/process as 2nd and third arguments. The fourth argument should be the number of generation and the fifth argument should be the name of the output file. Your program should write down the state of the grid in the output file after the simulation is complete.

Input and output file format: The input and output will be space delimitated ASCII file. The first two entries of the of the file will be the grid dimensions (row and column) and the rest of the file grid cell state (1 for alive and 0 for dead).

Example file: 4 4
0 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0

The example file depicts a four by four grid where the 3rd cell in the 1st row is alive and rest are dead and so on.

Execution: Run your code on 8 (2 processes, 4 thread/per process), 16 (4 processes, 4 thread/per process), 32 (4 processes, 8 thread/per process), 32 (2 processes, 16 thread/per process) for the given input file for 40 generations. I will provide the input file (for developing and testing the program create your own input file).

Measure the parallel execution times and compute the speedup. Write report comparing and discussing different executions of your hybrid runs and comparing it with your OpenMP and MPI implementations from assignment 2.

Grading

1. Program: 80 Points
2. Report: 20 Points

Submission:

Submit a zip file (no WinZip) containing your code (source, header and make files only no data file or executable) and a README file (containing instruction how to compile and run). The zip should also contain the report in pdf format. You also need to submit a printed copy of your report.