CSC 4760/5760, Fall 2015
Programming Assignment 5
Assign date: November 2, 2015, Due: November 11 2015 at class

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]**. You can seed the grid using random function. Write a parallel MPI version of the game of life. Run the programs for 40 generation for a grid size of 40000X400000 on 16 and 32 processors. For 16 processor run the application twice with the following configurations

1. All sixteen processor on a single node
2. 8 processors per node

Measure the parallel execution times and compute the speedup. Write report comparing and discussing your MPI Implementations with OpenMP implementations and discussing the results of two configurations of 16 processors run.

Grading

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

Submission:

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