CSC 4760/5760, Fall 2015
Programming Assignment 4
Assign date: October 16, 2015, Due: October 26 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 sequential and parallel OpenMP version of the game of life. Run the programs for 100, 200, 400 generation for a grid size of 40000X400000 (during development and testing use smaller grid and generation). Measure the parallel execution time on 2, 4, 8, 16, and 32 threads. Compute the speedup. The problem size and number of thread should be passed as command line arguments to the program. Profile your sequential code to determine the serial fraction of the code. Compute the theoretical maximum speed up for 8 and 16 threads and compare it with observed speedup. Write a report discussion the following points:

1. Partitioning strategy
2. Speedup and scalability with changing processor number.
3. Comparison of theoretical speedup and observed speed explaining the variation observed or not.

Bonus Point (10 Points):

Submit an animation (mpeg or other format) of your grid as it changes over generations along with brief report how you generated the animation of. You will need to store the grid in a file after every generation in file. I will provide more information on the animation next class.

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.