Game of Life

The problem

A board of NxM cells hosts a population of individuals. Each cell may host one individual or it may be empty. Population evolves in timesteps according to a set of rules, summarized as follows.

At each timestep:

- an individual in a cell with 2 or 3 alive neighbours stays alive
- an empty cell with exactly 3 alive neighbours becomes populated by a new individual
- an alive cell with less than 2 alive neighbours dies (becomes empty)
- an alive cell with more than 3 alive neighbours dies (become empty)

The board evolves for a number ITER of timesteps.

Assignment

You are required to implement a program computing in parallel, with parallelism degree nw, a given number of iterations on a NxM board with some initial random setting of the cells (alive or empty). The number of iteration, random number generator seed, dimensions of the board and the parallelism degree must be provided as command line parameters, being the pardegree the last one.

Ideally, you should provide three implementations:

A. the sequential one (to be used to compute speedups)
B. a parallel version implemented using C++ threads (only)
C. a parallel version implemented using OpenMP

Results

We wish to see scalability and speedup curves obtained on the Xeon PHI.

Some hints

- Please keep a look at the vectorization of the main (heaviest) computations in the code.
- In case, print the board on screen using some character for the individuals and nothing for empty cells. Check program that the program is working with small N M parameters than use larger values for N and M and suspend visualization.

- For the sake of simplicity, consider borders are always empty (all cells in first and last row/column are empty) and compute evolution of the internal cells only.

Figure 1: Display of two consecutive timesteps in a 30x60 board (step i, left, and i+1, right)