Parallel Computing/Programming Assignment #1: Game of Life

Christopher D. Carothers
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th Street
Troy, New York U.S.A. 12180-3590
Email: chrisc@cs.rpi.edu

February 2, 2016

DUE DATE: Noon, Tuesday, February 9th, 2014

1 Assignment Description

For this assignment, your are to write a Conway's Game of Life program except there will be a twist.

1.1 Basic Rules

The Game of Life is an example of a Cellular Automata where universe is a two-dimensional orthogonal grid of square cells (with WRAP AROUND FOR THIS ASSIGNMENT), 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 at each and every cell:

- Any live cell with fewer than two live neighbors dies, as if caused by under-population.
- Any live cell with two or three live neighbors lives on to the next generation.
- Any live cell with more than three live neighbors dies, as if by over-population.
- 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 using a Random Number Generator (RNG) at time 0. Use the RNG provided in the assignment 1 code template. The first generation is created by applying the above rules simultaneously to every cell in the seedbirths and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a "tick" The rules continue to be applied repeatedly to create further generations.

Note, a "tick" starts with $Cell(0, 0)$ and ends with $Cell(N-1, N-1)$ in the serial case. When we get to parallel things will get much more interesting!

1.2 Adding Additional Randomness

Using the uniform distribution provided in RNG, if value return by the RNG is greater than a set threshold, then perform the above described basic rules for each cell. Otherwise, randomly pick state of *LIVE* or *DEAD*.

1.3 Experiments

- Run a 16x16 cell universe for 100 ticks with a threshold of 0%.
- Run a 16x16 cell universe for 100 ticks with a threshold of 25%.
- Run a 16x16 cell universe for 100 ticks with a threshold of 50%.
- Run a 16x16 cell universe for 100 ticks with a threshold of 75%.
- Run a 16x16 cell universe for 100 ticks with a threshold of 90%.

Plot each of these experiments in a cell graph (yes, you'll have 256 data points). In your write-up describe how these plots differ.

1.4 Assignment 1 Code Template

Is available on `kratos.cs.rpi.edu` under `/home/chrisc/assignment1-template.tar.gz`. To untar it to your home directory, do the following:

- `zcat /home/chrisc/assignment1-template.tar.gz | tar -x`.

That will create an `template` sub-directory. The the name by doing `mv templateassignment1`. Inside the `assignment1` sub-directory you will see the code files which will be covered in class.

YES, YOU MUST USE THE TEMPLATE FOR THIS ASSIGNMENT. FAILURE TO DO SO WILL RESULT IN A SIGNIFICANT GRADE PENALTY.

2 HAND-IN INSTRUCTIONS

Keep/put your assignment C code and write-up with graphs in PDF format on the class server, `kratos.cs.rpi.edu` under the `assignment1` sub-directory on your Linux account.

We will use an automated script to collect the assignments. So it is extremely important you place your solution in the right directory name, `assignment1`.

PLEASE PUT YOUR ASSIGNMENT UNDER THE `assignment1` SUB-DIRECTORY OR ELSE IT WILL NOT BE COLLECTED CORRECTLY AND COULD RESULT IN SUBSTANTIAL POINT LOSS OR NOT GRADED AT ALL.

ADDITIONALLY, KEEP A BACKUP COPY OF YOUR CODE, DATA and REPORT/GRAPHS ON YOUR OWN SYSTEM IN CASE KRATOS.CS.RPI.EDU FAILS DURING THE ASSIGNMENT PERIOD.