CS240A: Applied Parallel Computing Spring 05

HW 2 – Game of Life

Prof. John R. Gilbert *Apr 18, 2005*

Problem Description

Implement a parallel simulation of John Conway's Game of Life in 2 languages on a 2D torus. This means that it is still on a square grid, but the cells on the topmost row depend on the bottom row, and those on the leftmost column depend on the rightmost column.

The rules are as follows:

- If nonempty, a cell with 1 or 0 neighbours dies, due to loneliness.
- If nonempty, a cell with 4 or more neighbours dies, due to overpopulation.
- If nonempty, a cell with 2 or 3 neighbours survives.
- If empty, a cell with 3 neighbours becomes populated.

Look at `http://www.bitstorm.org/gameoflife/` for more information and a sequential implementation.

Requirements

You will do the problem in one of (UPC / CAF) or MPI, assigned randomly in class. You will also implement this problem in one other language of your choice.

Your program should read the starting state of the Life grid from an input file. It should run for a specific number of iterations and then print its final state to an output file.

The data formats for the input and output files are the same. The format is as follows. On the first line, the number of occupied cells (ncells), the number of iterations (niter) and the grid size (gridsize) are specified, separated by a space. The next n lines contain the co-ordinates of the occupied cells. In the output file, niter and gridsize are the same as the input, but ncells will be different.

```
1st line: ncells niter gridsize
ncells lines: x y
```

Write a 1–2 page report on your implementation describing your data–redistribution techniques, sequential implementation and data structures, workload balancing. You should get reasonable performance on very sparsely populated grids. You may not want to explicitly check every grid cell in your highest performance code.

Plot a log-log performance graph with 1, 2, 4, 8 and 16 processors on a grid size of at least 1000, but preferrably higher (This depends on your platform). Depending on your languages, you may have to report your graphs for different platforms:

- MPI: datastar, cluster2, X1
- OpenMP: datastar, X1
- UPC: X1, cluster2
- CAF: X1
- Matlab*P: beowulf