Proposal

Joseph Conway's Game of Life via Multithreading

Team members

Devan Patel, Kyle Koceski, and Daniel Fuentes

March 6, 2014

0.1 Team members

1. Devan Patel - devanp92@ufl.edu
2. Daniel Fuentes - danielffuentes@ufl.edu
3. Kyle Koceski - kkoceski93@ufl.edu

0.2 Designated Submitter

Devan Patel

0.3 Project Overview

Joseph Conway's Game of Life program was implemented without parallelism; however, we wish to create a version with multithreading to produce a more efficient model. The program depicts how life creates, survives, and destroys depending on a cell's neighbors in an infinite grid. For each cell, there are eight surrounding cells that could either be alive or dead; the current cell dies if there are less than two or more than three cells that are alive around it. It survives if either there are exactly two or three cells alive around it. Lastly, a cell is birthed if the current cell is dead and there are exactly three cells alive around it. It is evident that the cells in the grid form and dissipate erratically and we would like to show that via multithreading.

Our version would consist of a web server where the user will be allowed to input the grid size, initial number of alive cells, and maximum number of cells to be created throughout the game. However, lives will be collectively represented by threads.

0.4 How This Project Relates To Concurrent Programming

As stated in the above, there will a single thread dedicated to a life. Each thread will know its neighbor and be able to access its current state (alive or dead). Inasmuch this will allow the current thread to determine its state. It should be noted that a Thread Pool will contain each life and the number of maximum threads will be determined by the user input (there will be a limit for the number of threads allowed). From this, we will be able to produce comparisons between a threaded and sequential version with the given information and conclude if parallelism has produced a more efficient model than its sequential counterpart.

0.5 Hardware and Software Used

There will be no additional hardware required other than a multi-core laptop since our product is purely software based. The software that we plan on using are as follows:

- Backend - Java 1.7
- Frontend - JSP 2.2, CSS, HTML5, and XML.
- Testing - Apache JMeter