CSC 410 Parallel Programming
Dr. Christer Karlsson
Benjamin Kaiser
Programming Assignment 2
Part 2 Conway's Game of Life

**Description of Program**

This program is a parallel implementation of Conway's Game of Life. It is supposed to take a bunch of parameters on the command line and use them to start the game board off. These parameters also indicated how long to run the game and when to print the game board. The parameters included the number of live cells on the board, and the size of the board.

**Description of algorithms and libraries used**

The only two things that were really used in this program were the OpenMPI library for C++ and then also the ping-pong algorithm. According to open-mpi.org, the OpenMPI library is "an open source Message Passing Interface implementation that is developed and maintained by a consortium of academic, research, and industry partners. Open MPI is therefore able to combine the expertise, technologies, and resources from all across the High Performance Computing community in order to build the best MPI library available. Open MPI offers advantages for system and software vendors, application developers and computer science researchers."

In other words, it is a distributed processing library that runs the exact same code on all of the processes.

The algorithm is described pretty in depth in my Homework 3 submission which I have also included in this submission. Essentially what the main algorithm is is a for loop that goes through each row and checks the 8 cells around and counts to see whether or not they are alive.

**Description of functions and program structure**

The program has three functions including main. Main is where all the fun happens. All of the calculations and stuff for each process is done there.

The other two functions are simply assistance functions. The first is "generateAlive". This function generates the random coordinates in the grid where cells are supposed to live.

The second function is "fillGrid". This function is supposed to take the row or rows assigned to a particular process and if the randomly generated coordinates fall within that row, then it places a 1 instead of a 0 in that spot.

The program was split into two files. These are life.cpp and life.h. Life.h contains all of the function prototypes and such that I could use. Life.cpp contains the actual code that the program uses.

**Description of testing and verification process**

The only real testing and running that I did was running the program. I did notice that the program appears to work properly if the number of rows <= the number of processors but anything is very non-deterministic and wasn't 100% sure why. I probably have a condition in one of my if statements wrong. The number of live cells and the number of times printing as well as the print iteration number.

**Data**

No data was collected really and nothing was really specified as near as I can tell for data collection.

**Analysis of Data**

Since no data was collected, I can't really analyze it.

**Description of Submission**

The contents of the submission of this program are as follows:

- Ping_pong.cpp - source code for part 1
- Pingpongresults.txt - data for part 1
- Pingpongresultssingle.txt - data for part 1
- Life.cpp - source code for part 2
- Life.h - source code for part 2
- PingPongAnalysis.pdf - data analysis for part 1
- Homework3.pdf