Mid-Project Report

## Members:

### Ameer Hamza, 18100216

### Moize Ahsan, 18100202

## Progress:

Our project is to parallelize the game “Conway’s Game of Life”. We are implementing the whole game in GOLANG. We have used a portion of our code from another subject whose reference has been given in literature references. The serial code has been implemented and provided with this report.

The code expects the following command-line arguments:

1. The input file containing the text as follows: If the text is ‘1’, it means the cell is live. If the text is ‘0’, the cell is dead. If the text is ‘\n’, this is the end of the current row and we should start reading the next row.
2. Number of rows in the grid
3. Number of columns in the grid
4. Number of iterations is the number of subsequent generations to calculate

Regarding the parallel code for the game, it is still in progress. We have discussed the details we decided to implement in the ‘Refinement and Changes’ section.

## Prototype/Partial Implementation:

We have provided the serial code in the file ‘game\_of\_life.go’ along with a sample ‘input’ named file which contains a 30x30 grid of 1s and 0s. The number of rows and columns passed as command-line arguments should be the same as those of the grid in input file. The corresponding output is shown in the ‘output’ file. The number of iterations used is 10.

## Refinement and Changes:

As already discussed, it is a grid based game. So grid decomposition and dynamic work pooling techniques will be used to divide the work among all the processors. The grid would be divided into blocks and the master process will make a work pool of available blocks to compute. If a process is free, it will ask the master process to assign it a block to compute the next generation for. Once the next generation is computed, it will communicate the changes to the master process and wait for the master to assign any other block to compute the next generation. The master process will wait until all the processes have communicated the changes to it. Once that generation is finalized, the master will again make the pool of the available work and processes will start computing another generation. This process will continue until the specified generations of the grid have been computed. This is our proposed solution for the time being, any more refinements would be further communicated.

## Literature References:

1 .<https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life>

2 .Portion of code from Advanced Programming (CS 300)

3. Section 2.3 of “An Intro to Parallel Programming” coursebook for workpool and decomposition techniques.