Nicole Gizzo
Application Programming Using Java
Homework 4 Analysis

For the fourth homework in Application Programming Using Java, we updated our previous version of the Game Of Life simulation in order to handle massive grids with millions of cells. In order to implement a larger scale of the Game Of Life simulation, small portions of the simulation was ran simultaneously using multiple threads. In doing so, this will reduce the runtime significantly. To experiment with the actual performance of my program, I ran the simulation for different grid sizes, and for each of those grids, a range of ticks from 1 to 16. All simulations were ran for 100 ticks. It is important to note that these simulations were run on my personal computer, which only has two cores. Since my computer only has two cores, it can only run at most four threads in parallel.

The first time analysis that was recorded was for a grid with the dimensions of 1,000 x 1,000. This was ran 5 times for the threads 1, 2, 4, 8, and 16, each ran with 100 ticks. The runtime recorded for this grid size with various number of threads can be seen in Figure 1. This graph shows a shows a relative speed up as you increase the threads from 1 to 8 threads, but there is a huge jump in runtime when you increase the threads form 8 to 16 threads. Since I previously mentioned that my computer could only run at most 4 threads in parallel, this jump was expected. I believe the slow down was caused from the other threads having to be called after the initial four threads. The slower runtime could also be explained from the threads interfering with each other, and having to switch between threads rather than running them all at once in parallel. The overhead of these extra threads caused the performance decrease.

The next time analysis that was recorded was for a grid with a size of 10,000 x 10,000. This was also ran 5 times for the threads 1, 2, 4, 8 and 16, each with 100 ticks. The runtime for this grid size with a various number of threads can be seen in Figure 2. The results from this graph differed from what I hypothesized the results to be. I expected the graph to be visually similar to the previous graph seen in Figure 1. I anticipated the simulation to run slower for a single thread since the graph is significantly larger, but I did not expect the simulation to run faster with 16 threads than 4 or 8 threads. Initially, I figured this was an error on my part for running the simulation with the wrong number of threads. However, when I recomputed the runtime for a 10,000 x 10,000 grid with 16 threads I got the same results.

The final set of simulations that were recorded was for grids with a size of 20,000 x 20,000. This was also ran 5 times for the threads 1, 2, 4, 8 and 16, each with 100 ticks. The performance analysis for this grid size can be seen in Figure 3. The results for this graph met my expectations in that you would see an initial high runtime for thread 1, which would decrease for two threads and run the fastest for four threads. The behavior in this graph also matched my expectations in that there was an increase in runtime when ran with 8 threads and an additional increase for 16 threads.

The reason why I limited the size of my grids to a 20,000 by 20,000 was due to a limitation in the amount of memory on my computer. With the limitations of memory and processors of my personal computer, I believe that the performance of my simulation is satisfactory despite these challenges. My program runs as expected in that for grids of a large size, it will run best when four or eight threads are used, and will run slower for a single thread or 16 threads and higher.

Figures:

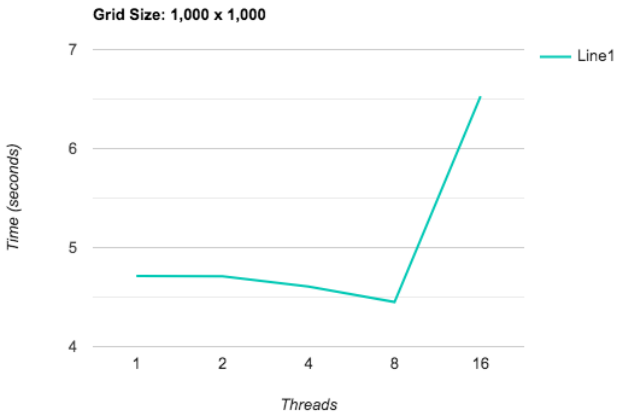

Figure 1- The runtime plot for a 1,000 by 1,000 grid. This plot shows the runtime for a grid of this size that was run once for each thread (1,2,4,8,16) with 100 ticks.

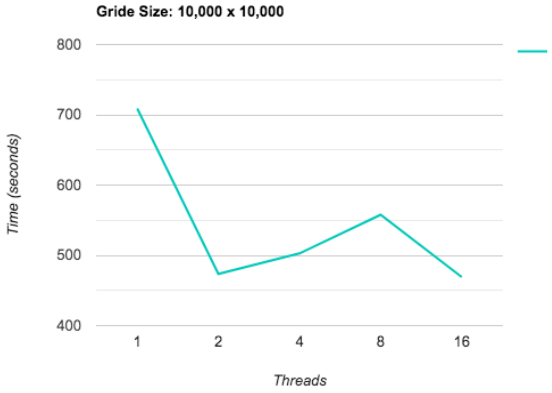

Figure 2- The runtime plot for a 10,000 by 10,000 grid. This plot shows the runtime for a grid of this size that was run once for each thread (1,2,4,8,16) with 100 ticks.

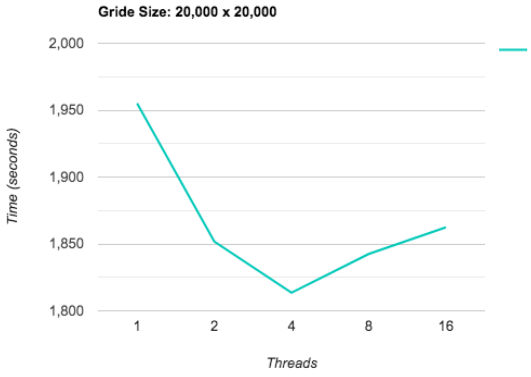

Figure 3- The runtime plot for a 20,000 by 20,000 grid. This plot shows the runtime for a grid of this size that was run once for each thread (1,2,4,8,16) with 100 ticks.