

Concurrent Computing Coursework

Josh Felmeden nk18044

Antoine Ritz,

December 3, 2019

Functionality and Design

Our solution was built up by initially creating a single threaded solution to the problem. This version iterates through the board bitwise, and for each bit gathers all of the 'neighbours' for the cells (the 8 directly adjacent cells). From this, the logic is applied and the cell is updated if necessary. This is repeated for the desired number of turns.

From this, we created a multi-threaded solution. We split the board up into strips and passed each strip to a worker. However, each worker would also need information from the lines directly above and below its strip of cells (called *halo lines*). We decided to pass these halo lines wrapped around the strips, so that the workers are able to calculate each cell correctly. Once they have completed their strip, they return it to the distributor function. The function reconstructs the world and begins the process

again for the desired number of turns.

Experiments and Critical Analysis

Stage 1a – Single Thread

Stage 1b – Divide and Conquer

Stage 2a – User Interaction

Stage 2b – Periodic Events

Stage 3 – Division of Work

Stage 4 – Cooperative Problem Solving