CSC410/510, Spring 2023

Group Project (Phase 2)

Parallel Implementation of Game of Life Using C/CUDA and Performance Analysis

A two-dimensional binary cellular automaton consists of a two-dimensional $n \times n$ grid together with an initial assignment of zeros and ones to cells of the grid and a "next-state" function for updating the values at the cells. The updating function is applied simultaneously to all the cells. The value of the next-state function at an individual cell will depend on the current value at the grid point and eight neighbors (four neighbors immediately above, below, to the right, and to the left of the cell, and the four diagonally adjacent neighbors: upper-left, upper-right, lower-left and lower-left neighbors).

John Conway's Game of Life is a well-known example. In it, cells with ones are "alive", while cells with zeroes are dead. Updating corresponds to finding the "next generation," and the new value of a cell depends on all eight immediately adjacent neighbors. The rules are the following:

a. Living cell with one or zero neighbors will die from loneliness.
b. Living cells with two or three neighbors will survive into the next generation.
c. Living cells with four or more neighbors will die from overcrowding.
d. Dead cells with exactly three neighbors will come to life.

You can assume that the number of rows is equal to the number of columns. Outside boarder of the $n \times n$ grid is "dead" in the sense that there is never an organism in these cells.

1. Write a serial C program that simulates a two-dimensional binary cellular automaton using the above rules. For example, you can let n = 100. The program reads the initial state from a file (input.txt). The initial state is a 100 × 100 matrix of 1s and 0s, where 1 denotes a living cell and 0 denotes a dead cell.
2. Use different initial states to verify the correctness of your implementation. One test case of the initial state is that the central 16 cells (a square with 4 rows and 4 columns) are alive and all other cells are dead. The iteration should stop if there is no change in the two consecutive generations.
3. Include more interesting examples to demonstrate the game of life. Some interesting games of life are included in the YouTube video https://www.youtube.com/watch?v=C2vgICfQawE. You are encouraged to include a few more challenging and interesting cases of game of life and visualize them for 10% bonus.
4. Design a parallel algorithm and modify the serial program into a parallel program written in C/CUDA. You program should work for any n and t values where n is the size of the matrix and t is the number of processes.
5. Write a report that (1) cover page, (2) problem description, (3) serial implementation and demonstration with at least five test cases, (4) parallel algorithm design and implement (partition, communications, boundary condition, thread organization, etc.), (5) screenshots to show users how to compile and execute your program, (6) compare your serial and parallel results of game of life, (7) performance analysis and discussion (serial runtime, parallel runtime, speedup, efficiency with tables and figures), (8) conclusions, (9) a brief description of the work and percentage of contribution of each group member, (10) references and open sources, and (11) appendix (serial and parallel source codes).

Submission: Upload the source codes of both serial and parallel programs (proper format) and the report (Word or PDF file) to Canvas before the deadline.

Student Name: ____________________

Project (Phase 1): Grading Policy

Grade components	**Points**	**Points earned**	**Comments**
Serial program and demonstration (correct results)	5		
Parallel algorithm and implementation that you should include in the report (partition, thread organization, efficient implementation)	10		
C/CUDA programing (comments, clarity, format)	10		
Compilation and Execution (Screenshots; consistent and correct results)	5		
HPC performance measure and analysis	10		
Report (cover page and all other components, format and clarity, correct results, contribution percentage)	10		
Total	50		
Bonus: Include challenging and interesting cases of game of life and visualize them for bonus.	10		