NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
COMPUTER SCIENCE SECTOR
COMPUTING SYSTEMS LABORATORY
www.cslab.ece.ntua.gr

PARALLEL PROCESSING SYSTEMS
9th Semester ECE, academic year 2023-24

Semester assignment

1 Familiarization with programming environment

1.1 Goal

The goal of this assignment is to get familiar with the lab compute resources (access to servers, compilation of programs, job submission etc.), through parallelization of a simple problem on shared memory architectures.

1.2 Conway's Game of Life

Conway's Game of Life takes place on a board of 2D tiles. Each tile can be either 'alive' or 'dead', simulating a living organism's existence (or not) in it, and can change state once at every given time step. At every time step, every 2D cell peaks into its state and the state of its neighbor cells (right, left, top, bottom, and diagonally) and updates its (new) state based on the following rules:

- If a cell is alive and has less than 2 live neighbors, it dies of 'loneliness'.
- If a cell is alive and has more than 3 live neighbors, it dies of 'overpopulation'.
- If a cell is alive and has 2 or 3 live neighbors, it survives this time-step.
- If a cell is dead and has exactly 3 neighbors, it becomes alive due to 'reproduction'.

1.3 Material

You are given a serial implementation of Conway's Game of Life in scirouter:
`/home/parallel/pps/2023-2024/a1/Game_Of_Life.c`.

1.4 Exercise

1. Gain access to the lab's clusters and solve any potential connection problems.
2. Get acquainted with compiling and submitting jobs in the cluster queues. For details about connecting, compiling submitting and executing see the "Lab Guide".
3. Develop a parallel version of Conway's Game of Life assuming a shared address space architecture with OpenMP.

4. Take performance (total execution time) measurements in one of the 'clones' for 1,2,4,6,8 cores and board sizes of 64×64, 1024×1024, and 4096×4096 (run 1000 generations/time-steps for all cases).
5. Gather, compare (via plotting), and comment on the results in an *exercise report*.
6. **Optional:** GoL is a very widely known kernel for creating shiny visual results. You can search the internet for specific value initializations that lead to interesting images, and plot their evolution using the relevant gif script we provide with the code.