NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTAMENTO DE
INFORMÁTICA

DEPARTAMENT OF
COMPUTER SCIENCE

Parallel Version of *Game of Life* with OpenMP

João Lourenço

April 14, 2021

Resumo

In this class you will learn/remember some basic concepts of concurrency and how to process data in parallel using the C programming language.

1 Introduction

The Game of Life[1][2] is a *zero players game*, that was invented in 1970 by the British mathematician John Horton Conway. Conway developed an interest in a problem which was made evident in the 1940's by mathematician John von Neumann, who aimed to find a hypothetical machine that had the ability to create copies of itself and was successful when he discovered a mathematical model for such a machine with very complicated rules on a rectangular grid. Thus, the Game of Life was Conway's way of simplifying von Neumann's ideas. It is the best-known example of a cellular automaton which is any system in which rules are applied to cells and their neighbours in a regular grid. Martin Gardner popularised the Game of Life by writing two articles for his column "Mathematical Games" in the journal Scientific American in 1970 and 1971.

You have been working with OpenMP in the past labs. In this lab you are expected to take a larger and more complex program and parallelize it with openMP.

2 Lab Work

2.1 Sequential version of the *Game of Life*

Clone the given version of a sequential implementation of the Game of Life in your device (your own laptop or lab workstation) with the command

```
git clone https://bitbucket.org/joaomlourenco/game_of_life_seq.git
```

and compile it using the command `make`. `make` is a command that builds a project given in a project specification text file named `Makefile`. Open the text file `Makefile` with your favorite text editor and use the web to learn a bit about `make` and `Makefile`. You'll need it again in the short future.

[1] https://en.wikipedia.org/wiki/Conway's_Game_of_Life
[2] http://web.stanford.edu/~cdebs/GameOfLife/

Campus de Caparica
2829-516 CAPARICA | Portugal
T: +351 212 948 536
di.secretariado@fct.unl.pt
di.fct.unl.pt

This project dependes on the "libpcre" library. In Linux you can install it with

```
apt intall libpcre
```

In macOS (with *HomeBrew*) can install it with

```
brew install libpcre3-dev
```

And in Windows can install it with

I have no idea... is you know please share with us!

The given `Makefile` is rather versatile and should work for both Linux and macOS. If you adapt it for Windows, please share your adaptations in Piazza.

2.2 OpenMP version of the *Game of Life*

The given version of the Game of Life is sequential. **Your job is to make a new parallel version of this program using OpenMP!**

Please follow these steps:

1. Compile and experiment with the given version. Carefully study (and make sure you understand) the given source code.
2. Change this program to use *Terminal ANSI Escape Codes* to:
   (a) Clear the screen (terminal) at the very beginning.
   (b) Position the cursor at coordinates $(0, 0)$ before printing the board, so that a board is printed overlapping the last board. This makes it easier to observe the evolution of the system.
3. Change the program to include a new optional flag "-p" with an integer as argument (from *pause*) that will make a pause for the given number of miliseconds before drawing the next frame.
4. Change the program to include a new optional flag "-q" (from *quiet*) so that only the last board (final state) is printed.
5. Create a parallel version of the program by using OpenMP.
6. Experiment your parallel version with different boards, board sizes and number of processors (remember you can use e.g. "`OMP_NUM_THREADS=2 ./glife 100 tests/1.in`" to run a simulation with just 2 processors). What is the achieved speedup? And efficiency? And cost? IS there a relation between the board size and the speedup achieved?

Please remember to use GIT appropriately. Leran to work with branches and make a branch when you start a new phase from the list above, and merge your branch to your main version when the development is finished and appropriately tested/validated.

Acknowledgments

The text from the Introduction is an adaptation from the text in http://web.stanford.edu/~cdebs/GameOfLife/.