

Concurrency and Parallelism 2018-19

(3) Game of Life in Cilk+

João Lourenço

September 26, 2018

Abstract

In this class you will learn about Cilk+ and improve your knowledge on the development of parallel programs.

1 Introduction

The Game of Life¹² 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.

¹https://en.wikipedia.org/wiki/Conway's_Game_of_Life

²<http://web.stanford.edu/~cdebs/GameOfLife/>

2 Lab Work

2.1 About Cilk+

Please download and read carefully the document available in CLIP, in the section “*Textos de Apoio*” named “**Structured Parallel Programming - App B (Cilk+).pdf**”. This document will teach you the basics of *Cilk+*.

2.2 Given Version

You may find at

`https://bitbucket.org/joaomlourengo/game_of_life.git`

a running version of the Game of Life.

Clone the given version in your Linux device (own laptop, lab workstation, or as a last resort, in the “node9” server used in the last lab class) with

`git clone https://bitbucket.org/joaomlourengo/game_of_life.git`

and compile it using the command `make`. `Make` is a command that builds a project given in a project specification file named `Makefile`. Have a look at this text file and learn a bit about `make` and `Makefile`. You’ll need it again in the short future.

Try running the program with the input files provided in the directory (folder) “`tests/`”.

2.3 Work plan

The given version of the Game of Life is multithreaded, using *pthread*s and the *fork/join* pattern.

Your job is to make a new parallel version of this program using Cilk+!

Please follow these steps:

1. Compile and experiment with the given version. Study the source code and understand what is the policy for the number of threads created by this program.
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 all the boards are printed overlapping the last board, and it becomes easier to observe the evolution of the system.
3. Change the program to include a new optional flag “-s” (from *silent*) so that only the last board (final state) is printed.
 4. Change your program to use Cilk+ instead of *pthread*s. It is possible to adapt the code to make a trivial replacement of the `pthread_create()` by a `cilk_spawn()`. But is this a good option? Is this the best option? Please build your argument to defend whichever design decision you make for your *Cilk+* version.
 5. Experiment with different boards and board sizes. What can you conclude about the relative speed of the *pythraeds* vs. the *Cilk+* versions? What can you conclude about performance gains of the *Cilk+* version relative to the board size?

Please remember to use GIT appropriately. Learn 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/>.