Agent-based Simulation of Fire Extinguishing

João Lourenço

April 25, 2022

Abstract

Please read this document carefully... be certain that it took me much longer to write it than it will take you to read it very carefully!

This is the project assignment for the course of Concurrency and Parallelism, edition 2021-22. This assignment is strongly based in the *EduHPC'19 Peachy Assignment* from the Trasgo Research Group of the Universidad de Valladolid[1].

The deadline for the project submission (code and report) is June 3, 2022, at 23:59.

The project submission is done by providing the repository URL and the commit ID of your project's code and report (the date/time of the commit ID must be before the deadline). More info on the project submission will be provided later.

1 Introduction

You are provided with a sequential code that simulates the evolution of a fire scenario with several focal points that is extinguished by multiple teams of firefighters.

The code provided computes a simplified simulation of fire emergencies, like for example in industrial facilities like oil refineries or chemical plants. Only a rectangular surface is considered in the simulation. This surface is modelled as a set of discrete and evenly spaced control points. A bidimensional array is used to represent the heat of each point in a given time step. Another one-dimensional array is used to store the information of a set of fire focal points. These points will be activated in fixed time instants during the simulation. Each focal point has a constant amount of heat until a team arrives to suffocate it. The program computes the evolution of the whole fire, taking into account the focal points and the behaviour of the teams of firefighters. This behaviour is simulated in a very simplified way, always commanding each team to move in the direction of its nearest focal point to suffocate it.

Figure 1 shows the output of the program for a surface of 30×30 points, after 14 iterations simulating the operations of two extinguishing teams. The following symbols are used to represent the state of each control point:

0: The point is at ambient temperature.

[1]Please remember that you are expected to develop your own project solution, and that any attempt of cheating implies immediate failure in the course!

```
Iteration: 14
+------------------------------------------------------------------------------------------+
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  .  +  +  1  1  1  +  +  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  .  +  1  1  2  3  2  1  1  +  +  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  .  +  1  2  4 (4) 4  2  1  1  +  +  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  .  +  1  2  3  4  3  2  1  1  +  +  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  .  +  1  1  2  2  2  2  1  1  +  +  .  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  .  +  1  1  1  2  1  1  1  1  +  +  .  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  .  +  +  1  1  1  1  1  1  +  +  +  .  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  .  +  +  1  1  1  1  +  +  +  +  .  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  .  +  +  +  +  +  +  +  +  +  .  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  .  .  +  +  +  +  +  +  +  .  .  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  .  .  .  .  +  .  .  .  .  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  .  .  .  .  .  .  .  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  .  .  .  .  .  .  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 [0] 0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 [.] 0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  .  .  .  .  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  .  .  .  .  .  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  .  .  .  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
| 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |
+------------------------------------------------------------------------------------------ +
Global residual: 5.503904
```

Figure 1: Example of output of the program in debug mode for an array of 30×30 positions and two extinguishing teams.

.:	The heat in the point is between 25 and 50 degrees.
+:	The heat in the point is between 50 and 100 degrees.
1–9:	The heat in the point is between the number multiplied by 100, and that number plus 100 degrees.
*:	The heat in the point is greater or equal to 1000 degrees.
( ):	The point is an active focal point.
[ ]:	One or more teams are visiting the point, or extinguishing the fire, if it is an active focal point.

The example represents two teams that have just extinguished a focal point at the bottom right of the surface (the residual heat is still dissipating) and now they are moving to the active focal point at the top left.

2 Sequential Code Description

2.1 Program arguments

The program receives the following arguments:

Argument	**Description**
size1, size2	Sizes of the bidimensional arrays of control points that stores the heat information of the surface.
maxIter	Maximum number of simulation iterations allowed.
numTeams	Number of extinguishing teams.
teamsDataList	Tuples of three numbers separated by spaces that indicate the initial position of each team and its extinguishing capacity (valid types are 1, 2, or 3).
numFocalPoints	Number of fire focal points.
focalPointsDataList	Tuples of four numbers separated by spaces that indicate for each focal point: position (x, y), activation time step, and constant amount of heat when it is active, before it is extinguished.

2.2 Surface file format

The program arguments introduced in the previous section can be also passed to the program by means of a text file. The first line of the file must contain the grid sizes `size1` and `size2`, and the maximum number of iterations `maxIter`. The second line contains the number of extinguishing teams. The following lines[2] contain the information of one team per line, its initial position (x, y), and its type (1, 2, or 3). Then, it appears a line with the number of focal points. The following lines contain the information of one focal point per line, the position (x, y), time step of activation, and the amount of heat. Students are encouraged to manually create or automatically generate their own test files. Several examples are provided along with the code.

Students are encouraged to manually create or automatically generate their own test files. Several examples are provided along with the code. You are allowed (and encouraged) to share your own wave files and simulation results with your colleagues, preferably publicly in the Piazza forum.

[2]Some of the input files merge multiple lines into one, which is possible, by using a space as separator..

2.3 Functional description

The program initializes all the control points of the surface with temperature 0 (ambient temperature), and then it starts the simulation loop. On each iteration there are several stages.

Activation of focal points: First, the focal points array is examined in order to check if any of them must be activated at that time step.

Heat propagation: Then, 10 steps of heat propagation are computed. The surface data is copied into an ancillary array, and the heat value of each position is updated using the copied values of its neighbours. The maximum difference between old and new heat values on any position of the surface is found.

Team movement: Each team calculates its distance to all the active focal points in order to decide in which direction should it move. Movement rules are slightly different depending on the type of the team.

Impact: After that, the result of the actions performed by each time is computed. If a team reaches an active focal point, it extinguishes it. It also reduces the amount of heat in the surroundings of its location according to an action radius that depends on the type of the team. Type-1 teams move fast (diagonally) but they have a short action radius. Type-2 and -3 teams move slower (steps can be just horizontal or vertical), but they carry extinguishing tools that allow them the reduction of heat in a longer range.

The simulation ends whether a fixed number of iterations is reached or the (residual) heat variation is under a given stability threshold.

Once finished, the program shows the elapsed time of the simulation and some results that are useful to check its correctness (final positions of the teams, and residual heat in the focal points).

2.4 Debug mode

If the source code is compiled with the `-DDEBUG` flag, a graphical representation of the results is presented, as discussed before. This representation can be useful to detect bugs, or just to know how the simulation evolves.

3 Project Development and Submission

3.1 Working rules

The following working rules apply. As this list is not exhaustive, in case of doubt do not hesitate in asking me.

- The project work will be done in grops of 3 students (unless explicitly authorized otherwise).

- Each group must create *one and only one* new repository by having one of the group member following the link `https://classroom.github.com/a/CXnQQMdt`.
- You will be asked to create a group. Name your group "`gNN_AAAAA_BBBBB_CCCCC`", where *gNN* is your group number (e.g., "`g07`"), `AAAAA`, `BBBBB`, and `CCCCC`, are the student numbers (IDs) of the three group members sorted in ascending order, i.e., lower (`AAAAA`) to higher (`CCCCC`). Example: `g07_12345_55555_98765`.
- Add your other two group members to the project.
- Remember that **your new repository must be private** and shared only among the group members
- The group must respect and use the "*Git Workflow*". Each member will have a local copy of the group's repository in his/her computer where individual work shall take place.
- The group shall make extensive use of branching (and merging). One branch–develop––test–merge for each work–task.
- Each student will commit his/her own individual work into the group shared repository.
- Commit your work frequently. Commit messages must describe clearly what was changed/added in that commit (and why). Commit messages are very important. Take you time to write good commit messages!
- Remember to push your (local) commits to your group remote master repository.
- Each project must have a `README.md` file identifying the group and its members, and with instructions on how to compile and run the test programs.
- The project report must have the look and feel of a *research article*, using the templates (Word or LaTeX) provided in the folder "`Report`" in the repository. If you ise LaTeX, please use the "`bare_jrnl_compsoc.tex`" with front size 10. If you use Word, leave the font size as is.
- The project report is limited to a maximum of 4 pages (see Section 3.4 for more info on the Project's Report). You may (and should) use Google to learn on how to write a research article.

3.2 Working methodology

You are given a reference sequential version of the code (`extinguishing.c`), and another identical[3] source file (`extinguishing_omp.c`) to develop the parallel OpenMP version. The following strategy is recommended:

1. Fork the original repository by accessing the link `https://classroom.github.com/a/CXnQQMdt`.

[3]The sequential version was modified for the `-DDEBUG` option, to introduce an *animation*-like behaviour.

2. Study the original source code and be sure you understand it. Please feel free to discuss the source code with your colleagues (and the teacher) in Piazza or, if necessary, directly with the teacher;
3. Find potential performance bottlenecks (maybe using a profiler);
4. Identify the best candidate statements/functions to exploit parallelism using OpenMP considering the observations from step 3;
5. Identify and remove the code dependences that may hinder the parallelization correctness and/or performance impact;
6. Use the OpenMP programming model to parallelize the program.
7. Run some preliminary tests to confirm the correctness of your parallelization approach;
8. Evaluate the impact of your parallelization;
9. Find additional potential performance bottlenecks (maybe using a profiler);
10. If any performance bottleneck is found, fix it and goto step 7;
11. If no performance bottleneck is found, look for additional (minor) blocks/functions that can be parallelized;
12. If some is found goto step 5;
13. Log into the cluster machine(s) (instructions will be provided later) and run **selected** performance tests there (remember that your CPU time in the cluster is limited);
14. If your results in the cluster do not match your expectations, go back to your development machine and to step 5;
15. Write your report. See Section 3.4 for suggestions/recommendations/rules for the report.

3.3 Grading Criteria

Remember my talk about the Bloom's Pyramid (see Figure 2). I will certainly try to assess both ***your group*** and ***your individual competences*** in all the levels, but I will certainly value more your outcome on the top three levels: *Analyze*, *Evaluate*, and *Create*.

This means that spending almost all of your project time programming and optimizing your and, and you do your final benchmarking in the last hours to quickly write your report, most certainly you will have a project grade **very much below** your expectations. So, please. . .

. . . leave **at least** one week for benchmarking your solution,
and for creating and writing your project report!

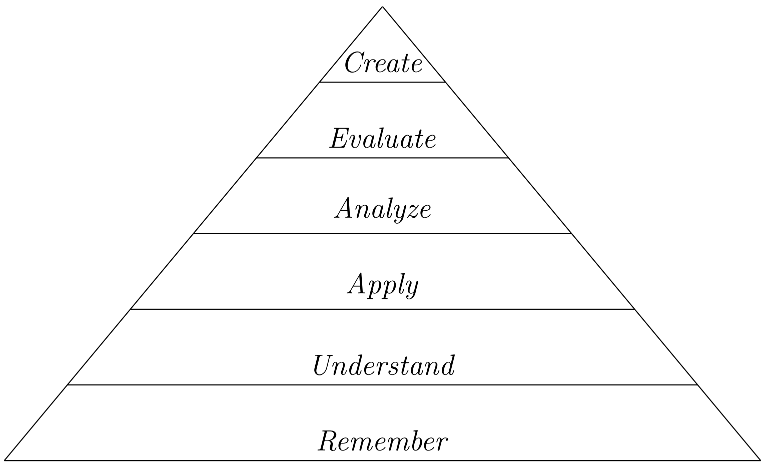

Figure 2: Bookm's Pyramid

3.3.1 Group grading criteria ($\mathcal{G}$)

The group grade ($\mathcal{G}$) is the same to all the elements of the group and will strongly depend on my appreciation of the group's work (on all the levels of the Bloom's Pyramid) **as perceivable from the written report**. Other relevant criteria include: easy of compilation and execution, cleanness and readability of the source code, correctness of the results, execution time/performance, scalability, etc.

3.3.2 Individual grading criteria ($\mathcal{S}$)

The individual grade ($\mathcal{S}$) is possibly different for each group member and **will strongly depend** on both, the work division as reported by the group, and on my own appreciation of the individual contribution of each group member (on all the levels of the Bloom's Pyramid) to the joint group's work as perceivable from the written report. Other relevant criteria include: compliance to the "*Git Workflow*", size and number of commits, meaningfulness of commit messages, etc.

3.3.3 Individual lab grading ($\mathcal{L}$)

The individual lab grade ($\mathcal{L}$) will be calculated as a weighted averaged following the formula:

$$\mathcal{L} = 0.7 \times (\mathcal{G}) + 0.3 \times (\mathcal{S})$$

3.3.4 Grade discussion

Both the group members and the professor may ask for an individual or collective (group) presentation and discussion of the project. As a result of such a presentation and discussion, both the group grade ($\mathcal{G}$) and the individual grade ($\mathcal{S}$) may be revised upwards or downwards.

3.4 Project Report Format, Structure, and Important Dates

The report shall be written **strictly** following the format provided in the `Templates` folder in the source repository and have the *look&feel* of a research paper. The report length is **limited to four pages** of text and graphics, plus **one extra page** for:

Section	**Description**
bibliography	Remember to add **all the documents and web-sites** relevant to your work to the bibliography section... and remember that you must cite them in the text wherever they were relevant;
acknowledgments	Please identify (number and name) your colleagues that were somehow helpful to your group while developing the project... also explain why/how they helped you;
individual contribution(s)	Please present your group working methodology and list clearly: i) how the work was divided between the group members, and ii) the relative contribution of each member (in percentage). For example, *A did whatever (20%), B did something else (30%), and C did a lot of stuff (50%).* If the group members cannot agree in the terms/contents for this sectin, add one (identified) separate statement for each group member.
comments, critics, and suggestions	All your comments, critics and suggestions are very welcome. My goal is to provide you with an interesting and educational project (and course) and all your feedback is very welcome (of course this section **will have no impact on your grade**).

Version history:

Date	**Version**	**Description**
2022-04-25	`1.0`	Initial version.