Assignment 2: Parallelising the Simulation of Forest Fire Spread and Recovery

Due: **Week 12 Tuesday** (26/11/2024, 11:59pm Irish time)

Mark: **30%**

1 Aim

Design a parallelised solution that leverages data parallelism and task parallelism, using appropriate synchronisation mechanisms, and analyse the performance of the parallel implementation using Amdahl's Law, Gustafson- Barsis Law, and the Karp-Flatt metric.

2 Documents to Submit

- You should submit **a report (PDF)** called **ID#_Report.pdf**.
    - For example: if you are student number is 12345678, then you should name your report: **ID12345678_Report.pdf**
- You can submit more than one JAVA file, if needed.
    - In your java file(s) you should include prominent comments at the start of the code that contain your ID number and your Name.
- No ZIP Files.

3 Assignment Specification: What You Need to Do

Your task is to design a parallelised version of a simulation that leverages data parallelism and task parallelism, using appropriate synchronisation mechanisms. You are also tasked to analyse the performance of your parallel implementation.

3.1 Background: Simulating Forest Fire Spread and Recovery

The simulation represents a large forest as a **2D grid** where each cell can be in one of the following states:

- **Empty** (no trees)
- **Tree** (healthy tree)
- **Burning** (tree on fire)

Each time step in the simulation will apply rules for fire spread, tree growth, and recovery. The rules for the simulation are:

- Fire Spread
    a. A **Burning** cell turns into an **Empty** cell in the next time step.
    b. A **Tree** cell will start **Burning** if one of its neighbours is **Burning**.
        i. `burnProbability`: represents the likelihood that a **Tree** cell will start **Burning** in the next time step if one or more of its neighbouring cells are **Burning**.
        ii. For each **Tree** cell that has at least one **Burning** neighbour, a random value between 0 and 1 is generated. If this value is less than or equal to `burnProbability`, that **Tree** cell will start **Burning** in the next time step.
- Tree Growth and Recovery
    a. An **Empty** cell may grow a **Tree** based on a probability
        i. `growthProbability`: represents the chance that an **Empty** cell in the grid will turn into a **Tree** in the next time step.
        ii. For each **Empty** cell, a random number is generated between 0 and 1 at each time step. If this number is less than or equal to `growthProbability`, that cell will become a **Tree** in the next time step.
    b. A **Burning** cell becomes **Empty** in the next time step after it has burned out.
- Boundary Conditions
    a. The simulation wraps around boundaries, so the grid is treated as a torus (or a donut shape).
    b. **Top and Bottom Edges**: Cells in the top row treat cells in the bottom row as their "north" neighbours, and cells in the bottom row treat cells in the top row as their "south" neighbours.
    c. **Left and Right Edges**: Cells in the leftmost column consider cells in the rightmost column as their "west" neighbours, and vice versa for cells on the rightmost column.
    d. **Corners**: follow both rules above, meaning that each corner cell considers its diagonal opposite on the grid as one of its neighbours.

3.2 Problem Statement: Parallelising the Simulation

You are tasked with developing a **parallel solution** for a forest fire simulation. Your goal is to leverage **parallel** programming **techniques**, such as data parallelism and task parallelism, to achieve efficient simulation performance. Given that fire spread relies on the state of neighbouring cells, you will need to implement **synchronisation mechanisms** like mutexes and semaphores.

You are also required to analyse the performance scalability of your solution.

3.2.1 Implementation of Parallel Fire Simulation (50%)

Your first task is to provide a parallel solution to the forest fire simulation. You should take into account the following:

- Divide the forest grid across multiple threads: 2, 4, 8.
- Use synchronisation to ensure that cells are consistently updated across threads.
- Create a separate task that records the number of cells in each state (**Tree**, **Empty**, **Burning**) at each time step.
- **Grid Size and Steps**: The grid size[1] and the number of simulation steps[2] should be adjustable.
- **Probability Parameters**: `growthProbability`[3] (for tree growth) and `burnProbability`[4] (for trees catching fire from neighbours) should be set.

Please note: your report should clearly describe the design and implementation of your parallel solution, including explanations of how parallel and synchronisation techniques are applied to the forest fire simulation. It should link to specific code segments to demonstrate the implementation. Your code should also be well commented. Failure to provide clear, structured explanations that directly reference your code will result in a significant penalty.

3.2.2 Performance Analysis (30%)

Your second task is to measure the performance of the parallel solution. This requires you to:

- Record the execution time of the simulation with varying numbers of threads.
- Calculate the speedup and efficiency.
- Analyse the performance using Amdahl's Law, Gustafson's Law, and the Karp-Flatt metric, showing both theoretical and practical impacts of parallelism on this problem.

[1] Start small (e.g., 10*10) while developing. Use larger grids (500*500 or more) for evaluating performance. Please note if you are encountering significant slowdowns or memory crashes due to large sizes, you can use smaller.

[2] Similar to the grid size, you should use small number while developing (e.g., 10, 20, or 50), and larger while evaluating performance (e.g., 400, 600, or 1000).

[3] Start by setting the growth probability to around 0.01 to 0.1 for typical forest recovery, meaning that the forest regrows slowly over time. For faster regrowth, you could try a value closer to 0.2 or 0.3, but avoid excessively high values.

[4] Start by experimenting with values between 0.1 and 0.5. A value closer to 0.1 means that the fire spreads more slowly, while a value closer to 0.5 means it spreads quickly across the forest.

3.3 Detailed Report (Quality 20%)

Submit a detailed report that:

- Presents your findings, including code structure, parallelisation strategy, synchronisation choices, and performance analysis.
- Discuss where parallelism improved the performance and where overhead or contention limited it.

20% is dedicated to the quality of your report. This evaluates how clearly, accurately, and professionally you communicate your work. This includes the structure and organisation of the report, clarity of writing, quality of analysis, accuracy in presenting data or findings, and overall presentation.

4 Assessment Criteria

The table provided outlines the assessment criteria and grading breakdown for evaluating your performance. Utilise these guidelines as a reference and consistently strive for excellence in your work.

Table 1 Assessment Criteria

Criteria	Excellent (80%-100%)	Proficient (60%-79%)	Satisfactory (40%-59%)	Limited (20%-39%)	Inadequate (0%-19%)
Parallel Solution (50%)					
Performance Measurement (30%)					
Report Quality (20%)					

Final Mark: (<Score>/100)*30%

4.1 Scoring Scale

Excellent:

- Represents an exceptional level of understanding and execution of the assignment's requirements.
- Achieves a performance that consistently exceeds the highest standards set by the assignment's objectives.
- Characterised by exceptional attention to detail, a comprehensive approach, and minimal to no errors.

Proficient:

- Shows strong competence in understanding and executing the assignment's requirements.
- Consistently meets or slightly exceeds the assignment's objectives.
- Characterised by strong attention to detail and a good depth of coverage.

Satisfactory:

- Meets the basic expectations for the assignment but may lack some depth or precision.
- Characterised by satisfactory quality, with moderate proficiency in fulfilling the assignment's requirements.
- May have occasional errors or incomplete coverage of the assignment's expectations.

Limited:

- Displays some understanding of the assignment's requirements and objectives.
- Achieves work that is somewhat effective but may have shown inconsistencies or gaps in execution.
- Characterised by a decent quality of work that may fall short in terms of depth, precision, or comprehensive coverage.

Inadequate:

- Fails to meet the minimum expectations for the assignment.
- Demonstrates significant deficiencies, resulting in incomplete, poorly executed work.
- Characterised by multiple errors, omissions, and inadequate coverage.

5 Note on Plagiarism and Cheating

Plagiarism and cheating are **serious academic offenses** that undermine the integrity of the learning process and the reputation of both individuals and institutions. It is crucial that all group members understand and adhere to the principles of academic honesty throughout this assignment:

1. **Original Work:** All work submitted must be original and created by the student. Copying, reusing, or paraphrasing content from external sources without proper attribution is strictly prohibited.
2. **Proper Citation:** If you refer to external sources, including textbooks, online materials, or classmates' work, ensure that you provide appropriate citations. Failure to do so can be considered plagiarism.
3. **Collaboration vs. Copying:** Collaboration between students is encouraged and expected, but there is a clear distinction between collaborative work and copying. Collaboration involves sharing ideas and discussing concepts to enhance understanding. Copying involves duplicating another person's work or ideas without adding your own input.

Consequences for plagiarism and cheating may include a failing grade for the assignment, academic probation, or more severe disciplinary actions as per university policies. Please take this note seriously and maintain the highest standards of academic integrity in your work. If you have any questions or uncertainties about what constitutes plagiarism or cheating, seek clarification from your instructor before submitting your assignment.