Cardiff School of Technologies

Assessment Brief

Module Code	Module Title
CIS6007	Parallel and Distributed Systems
Academic Year	**Semester**
2023-2024	2

Module Leader email

Rehmat Ullah – rullah@cardiffmet.ac.uk

Content

Assessment Details

Assessment title	Abr.	Weighting
Game of Life using Parallel Programming Techniques	WRIT1	50%

Task/assessment brief:

The "Game of Life", or simply "*Life*", is an example of Cellular Automata developed by John Conway. The concepts involved are straightforward, but the emergent behaviour of the system can be complex and surprising!

The *universe* is a 2D array of elements, or *cells*. Initially some cells are filled in (*alive*, or *populated*) while the remainder of the cells are empty (*dead*, or *unpopulated*). This initial state, or pattern of alive and dead cells, is called the *seed* of the system.

The universe moves from one state to the next by applying a set of *rules* to every cell simultaneously. The rules outlined by Conway are…

- Any live cell with fewer than two live neighbours die, as if caused by underpopulation.
- Any live cell with two or three live neighbours lives on to the next generation.
- Any live cell with more than three live neighbours dies, as if by overpopulation.
- Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

Each new state is called a *generation*. More information about the Game of Life can be found at https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life.

For this assignment, you are required to implement 2 versions of Life – a normal serial version and a parallel version using Java. In each case you're required to setup the appropriate data structures and implement the algorithms to implement the above rules.

In addition to the implementation, you're also required to write a small report that outlines a set of experiments to run against which each of your implementations (sequential & parallel). The experiments should test different properties of your simulation – the array / universe size for example. As part of the report, you're required to compare the run-times and reflect on how the different parameters of each experiment affect performance. The report should be word processed and no longer than 1800 words. References and citations to material you research should be included where relevant.

You will also be required to explain your design and implementation in a short, informal **video presentation** lasting no more than 5-10 minutes. You will be required to talk about:

1. A walk-through of your code, explaining not just what's in the code, but *why* you chose the solution shown.
2. A demonstration your solution running.
3. A discussion of the timing results you obtain.

Word count (or equivalent):	2000

This a reflection of the effort required for the assessment. Word counts will normally include any text, tables, calculations, figures, subtitles and citations. Reference lists and contents of appendices are excluded from the word count. Contents of appendices are not usually considered when determining your final assessment grade.

Academic or technical terms explained:

Reflection: “Definitions of reflection are characterised as learning through experience toward gaining new insights or changed perceptions of self and practice” (Johns, 2007: 7)

Fault tolerance: Fault tolerance in distributed systems refers to the system's ability to continue operating and providing services in the presence of faults or failures.

Round trip latency: Round trip latency/delay/responsiveness refers to the total time it takes for a data packet to travel from a source to a destination and back again. It is the sum of the time it takes for the data to travel from the sender to the receiver (transmission and propagation delays), plus the time it takes for any acknowledgment or response to travel back from the receiver to the sender.

Parallelism: Parallelism involves the simultaneous execution of multiple tasks at the same time.

Concurrency: Concurrency involves the execution of multiple tasks, but not necessarily simultaneously.

Submission Details

Submission Deadline:	Thursday 4th April 2024	**Estimated Feedback Return Date**	This will normally be 20 working days after initial submission.
Submission Time:	By 4.00pm on the deadline day		
Moodle/Turnitin:	**Any assessments submitted after the deadline will not be marked and will be recorded as a non-attempt unless you have had an extension request agreed or have approved mitigating circumstances. See the School Moodle pages for more information on extensions and mitigating circumstances.**		
File Format:	The assessment must be submitted as a pdf or Word document (save the document as a pdf in your software) and submit through the Turnitin submission point in Moodle. **Your assessment document should be titled as follows:** **student ID number, module code and assessment ID, e.g. st12345678-CIS6007-WRIT1.docx** The required report (detailed above) including appendices must be submitted as a pdf or word document through the Turnitin submission point in Moodle. This must include a **link to (1) the code repository or shared OneDrive folder** you have used to store your implementation’s project files and **(2) a link to your video presentation**.		
Feedback	Feedback for the assessment will be provided electronically via Moodle. Feedback will be provided with comments on your strengths and the areas		

which you can improve. View the guidance on how to access your feedback.

All marks are provisional and are subject to quality assurance processes and confirmation at the programme Examination Board.

Assessment Criteria

Learning outcomes assessed

Learning Outcomes
[LO1] Demonstrate understanding of the theoretical concepts and abstractions to the design of novel and innovative distributed and parallel systems. [LO2] Critically evaluate the fundamental issues in the design of distributed algorithms, protocols and systems, such as timing, coordination and consensus. [LO3] Implement appropriate parallel and distributed software solutions to common problems found in computationally-heavy and data-driven domains.

Assessment Criteria	**100%**
1. Implementation	**50%**
Sequential Version (LO2, LO3)	20%
Parallel Version (LO2, LO3)	30%
2. Report	**30%**
Experiment Design (LO1, LO2)	15%
Discussion of Results (LO2)	15%
3. Video Presentation (LO1, LO2)	**20%**
Clarity and structure of presentation	10%
Demonstration of understanding of your implementation	10%

Other skills/attributes developed

This includes elements of the Cardiff Met EDGE (Ethical, Digital, Global and Entrepreneurial skills) and other attributes developed in students through the completion of the module and assessment. These will also be highlighted in the module guidance, which should be read by all students completing the module. Assessments are not just a way of auditing student knowledge. They are a process which provides additional learning and development through the preparation for and completion of the assessment.

Ethical	An awareness of the ethical use of modern parallel and distributed systems, conducted with integrity, societal considerations, and respect for the privacy of individuals and communities involved. By incorporating ethical principles into distributed systems particularly privacy concerns, organizations can foster transparency, trust, and accountability.
Digital	Demonstrate understanding of how modern parallel programming technologies can be applied to a given problem using latest programming tools and frameworks.
Global	Parallel and distributed systems acknowledge the global requirement posed by next generation

	internet applications such as high volume of data generation and the privacy associated with the personal data. This module takes into account the global issues such as responsiveness in latency sensitive applications such as autonomous cars, parallelism for performance to achieve such latency, bandwidth usage, and privacy in the context of parallel and distributed systems.
Entrepreneurial	The development of key theoretical and practical skills in parallel and distributed systems fosters an entrepreneurial mindset and equips students with the tools and knowledge necessary to navigate the dynamic landscape of next generation internet applications. Students can apply the knowledge of the cutting-edge topics in distributed systems such as Edge Cloud Computing, Edge Intelligence particularly Edge-based distributed federated learning and may bring innovation to overcome the challenges posed by conventional distributed systems.

Marking/Assessment Criteria

70 – 100% (1st)	An excellent parallel implementation is given that correctly and efficiently implements each rule in the given kernels. The implementation also correctly and efficiently manages the arrays / buffers for each generation of the system as well as correctly assigns data to kernel parameters and captured variables. A well written sequential version is also given. A detailed experiment design is given in the report that documents tests on a thorough range of properties. An excellent in-depth analysis of the results obtained, and scalability of the given solutions is also evident. An excellent understanding of the code is evident in the video presentation.
60-69% (2:1)	A very good parallel implementation is given that correctly implements each rule in the given kernels. The implementation also correctly manages buffer / array resources. A well written sequential version is also given. A detailed experiment design is given in the report that documents tests and a good range of properties, but this could be expanded. A very good analysis of the results obtained is also given. A very good understanding of the code is evident in the video presentation, though more in-depth discussion could have been included.
50-59% (2:2)	A good parallel implementation is given that correctly implements the rules in each kernel. Good management of array / buffer resources is also evident, though this could be more efficient. A good sequential version is also given, but could be more efficient. A good experiment design is given in the report that documents tests on a range of properties, but this needs to be expanded. A good analysis of the results obtained is also given. A reasonable understanding of the code is evident in the video presentation and the discussion of the code need to be expanded.
40-49% (3rd)	Only a basic parallel implementation is given that correctly implements the rules in each kernel, but this could be more efficient. Array / buffer resource handling could also be made more efficient. A working sequential version is also given but this needs to be more efficient. A basic experiment design is included in the report but few properties are tested. Some analysis of the results obtained, but this also needs to be expanded. Only a basic understanding of the code is evident in the video presentation.
35-39% (Narrow Fail)	Only a very basic parallel implementation is given that only partially implements the required rules in the given kernels. The implementation is also inefficient, both in terms of the code and handling of array / buffer resources. A partially functional sequential version is also given. Only a very basic experiment design is given in the report, with little to no in-depth discussion of the results obtained. In addition, little to no understanding of the code is evident in the video presentation.
<35% (Fail)	Little to no code has been implemented, and no meaningful experiment design, results or discussion are given – what has been submitted needs to be significantly expanded. Little to no understanding of the code or parallel programming techniques required are evident from the video presentation.

Further Information on assessment, referencing and grading can be found in the Module Handbook (on Moodle)

Cardiff Met

MetCaerdydd