SOFT354 Coursework Assignment 2019/2020

Assignment Brief

Assignment Title:	Parallel Computing and Distributed Systems Coursework
Submission Deadlines:	**4pm, 16th January 2020**
Submission:	Online (DLE)
Contribution to Module Grade:	60%
Individual/Group Assignment:	Individual
Module:	SOFT354
Module Leader:	Dr Mario Gianni

Requirements

For this coursework, you are required to

- identify a problem in which parallel computing allows for better performance;
- provide a serial implementation of an algorithm that solves the problem you have identified;
- provide a parallel implementation in CUDA of an algorithm that solves the problem you have identified;
- produce a report.

The problem can be something related to what was covered during our practical sessions or a particular computational topic that you find interesting. Some suggestions are

- Linear Algebra (e.g., Matrix multiplication, matrix inversion).
- Image processing (e.g., filtering, denoising, feature detection).
- Sorting and Graph Theory (e.g., merge sort, Dijkstra).
- Machine Learning (e.g., Neural Networks, Support Vector Machine, K-Means).
- Simulation of dynamical systems (e.g., n-body simulation, computational fluid dynamics, “flocking” simulations).

Your implementation in CUDA of the algorithm can either be a parallelisation of the serial algorithm, or a completely separate parallel algorithm that solves the same problem.

The report must include the following sections:

- **Introduction**:

    - Explain the topic you are investigating in the coursework. You should provide some background in which you will provide details on the computational process involved, the applicability domain and the current state of the art, if applicable. You should include some references that indicate the source of your information (i.e. books, journals and websites). In particular, you must explain what parallel computation using CUDA / GPU involves, and how it can be beneficial to the problem you are focusing on.
- **Implementation**:
    - Describe the serial version of the algorithm that you implemented. Discuss how you moved from the serial algorithm to the parallel one. If applicable, you can explain here the parallelisation methodology and the particular parallel design you have adopted.
- **Evaluation**:
    - After the serial and the parallel implementations are ready, you must perform a comparison in terms of performance between the two and discuss the results. For example,
        - calculate the speedup of the parallel version over the serial one;
        - calculate and discuss the efficiency;
        - demonstrate the scalability of the program by modifying the problem size.

      Remember to be methodologically precise in taking measurements:
        - indicate which part of the code you are considering;
        - indicate whether you are taking into account overheads;
        - report an average of multiple measurements;
        - include charts where applicable.
- **Conclusion**:
    - Summarise your findings and discuss any further work that could be undertaken to further improve your program.
- **References**:
    - The bibliography used in the report. Harvard style of referencing is preferred.

Word limit: There is no strict limit on the length of your report. As a guide, aim for between 5 and 10 A4 pages using a size 12 font, with a reasonable number of figures / tables. This corresponds (approximately) to a word count of between 2,500 and 5,000 words.

Submission Details

Submission is online via the DLE. Two files must be submitted:

- The report, in Word or PDF format.
- A zip file with two directories that contain the CUDA and the serial version of the solution (make sure all source code is included).

Deadline and Marking

The deadline for submission is **4pm, 16th January 2020** via the submission link on the SOFT354 DLE page. **If the marker has difficulty compiling or running your code, or some aspect of what you have done is unclear, you may be required to attend a short viva with me to explain your work.** Feedback and marks will be returned within 20 working days.

Assessment Criteria

This coursework is worth 60% of your mark for the module. It will be marked according to the following scheme:

Code	Max. mark
CUDA	**25%**
Coding style	**5%**
Self-explanatory use of variable names, correct use of datatypes, modularity, comments in the code.	
Correctness, quality, performance and efficiency	**20%**
Behaviour of the algorithm (e.g., for each input it produces the expected output), code optimization, time and space complexity.	
Serial	**15%**
Coding style	**5%**
Self-explanatory use of variable names, correct use of datatypes, modularity, comments in the code.	
Correctness, quality, performance and efficiency	**10%**
Behaviour of the algorithm (e.g., for each input it produces the expected output), code optimization, time and space complexity.	
Report	**Max. mark**
Introduction	**10%**
Good justification of the choice of problem, accurate background information, reference.	
Implementation	**15%**
Clear explanation of the algorithms used and how the parallel algorithm makes use of the features of CUDA.	
Evaluation	**25%**
Suitable choice of measures to compare the algorithms, and correct application of these measures. Results should be presented clearly and discussed. Soundness of the findings.	

Conclusion	**10%**
Concise summary of the work and possible future work.	

Plagiarism

This is an **individual** assignment and must reflect the work of that individual. Thus, while you may discuss this assignment in general with your colleagues and give each other technical help (e.g. diagnosing compiler errors), your code and report must be entirely your own work.

The University treats plagiarism very seriously. If you cannot satisfy me that your work is your own, formal plagiarism procedures will be started.

The penalty for submitting work which is wholly or partially the work of someone else is usually, at least, a mark of zero for the assignment. Do not be tempted to help a colleague by giving them your code or design, as both parties will be guilty of an assessment offence and both face the risk of a zero mark. Please refer to your student handbook for guidance as to what constitutes original / individual work.

Module Learning Outcomes Assessed

- **ALO-2**: Relate the concepts, the logic, and the requirements of the domain in which parallel computation is beneficial to its applicability domain.
- **ALO-3**: Apply software engineering tasks to real design of parallel systems and implement the design on dedicated software applications capable to run on specific parallel architecture.