SOFT354 Coursework Assignment 2016 / 2017

Assignment Brief

Assignment Title:	Parallel Computing and Distributed Systems
Submission Deadlines:	4pm, 25th January 2017
Submission:	Online (DLE)
Contribution to Module Grade:	60%
Individual / Group Assignment:	Individual
Module:	SOFT 354 - Parallel Computing and Distributed Systems
Module Leader:	Dr Robert Merrison-Hort

Requirements

For this coursework you are required to produce a report with two supporting code implementations. You will need to identify a problem in which parallel computing allows for better performance. This problem can be something related to what was covered during labs (e.g., vector multiplication) or a particular computational topic that you find interesting – there are some suggestions at the end of this document.

You are then required to produce the following:

- A **serial implementation** of an algorithm that solves the problem you have identified.
- A **parallel implementation** of an algorithm that solves the problem you have identified. You can choose whether to base this on CUDA or MPI. This can either be a parallelisation of the serial algorithm, or a completely separate parallel algorithm that solves the same problem.
- A report containing 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** or **MPI** 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.
  - **Comparison and discussion**: After the serial and parallel implementations are ready, you must perform a performance comparison between the two and discuss the results; e.g., 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; 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 you used for the report. Please use the Harvard style of referencing.

Word limit: There is no strict limit on the length of your report – write as much or as little as you need to in order to explain what you have done. 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. You should submit two files:

- Your report, in Word or PDF format.
- A zip file with two directories that contain your serial and parallel programs (make sure all source code is included).

Deadline and Marking

The deadline for submission is 4pm, 25th January 2017 via the submission link on the SOFT354 DLE page. **If I have 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 (50%)	Well written, making appropriate use of suitable data types and CUDA / MPI functions. Code should be neatly written, with sensible variable names, indenting, and comments where necessary.	**50%**
Report (50%)	**Introduction:** Good justification of the choice of problem, accurate background information, references.	**10%**
	Implementation: Clear explanation of the algorithms used and how the parallel algorithm makes use of the features of CUDA / MPI.	**15%**

	Comparison / Discussion: Suitable choice of measures to compare the algorithms, and correct application of these measures. Results should be presented clearly and discussed.	**15%**
	Conclusion: Concise summary of the work and possible future work.	**10%**

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.

Potential Computational Problems

You can select one of the problems below, or find another that you find interesting. In either case, please confirm with me (via email) what problem you have selected before starting work.

- **Image processing**: e.g. applying a filter, feature detection, video processing, computer vision.
- **Linear algebra**: e.g. matrix-matrix multiplication, inverting a matrix.
- **Machine learning**: e.g. neural networks / deep learning; support vector machines.
- **Physics simulations:** e.g. n-body simulation, computational fluid dynamics, “flocking” simulations.
- **Cryptography:** e.g. encryption, decryption, attacks.