CM30225 Parallel Computing
Assessed Coursework Assignment 2

Set:	Thu 17 Nov 2022
Due:	8 p.m. (UK time), Monday 9 Jan 2023, via Moodle
Marks:	10% of course total
Set by:	Russell Bradford
Environment:	Azure, MPI and C

The objective of this assignment is to give you experience of using MPI on a distributed memory architecture; plus a feel for how parallel problems scale on such architectures.

The problem to solve is the same as Assignment 1. But, now, you will be using MPI on a distributed memory architecture.

As before, run your code on varying sizes of arrays and varying numbers of cores, but in a distributed memory configuration.

A complete solution consists of

- properly commented source code, together with any extra information you feel will help, such as describing your approach to the parallelisation of your solution, how you thought about messaging overhead, and so on
- excerpts from the results of your **correctness testing** of your code with a description of your testing process
- comments, graphs or any other relevant details of your **scalability investigation**, with particular reference to speedup and efficiency

Hand in your writeup as a PDF, *not* as DOC, DOCX or DOC variant (or MS spreadsheets, or Apple formats)

Hand in the code in a text (C) file that can be compiled, separate from your writeup, *not* as a listing in the PDF (check the upload to Moodle did not convert your program file to HTML)

All answers must be placed on Moodle by 8 p.m. on Monday 9 Jan 2023. CHECK YOUR HANDIN. Moodle has been known to lose coursework.

Feedback

Feedback on this assignment will be provided via Moodle, normally within three semester weeks. There will be general feedback that applies to many people and some individual feedback. Please read your individual feedback in the context of the general feedback.

Anonymous Marking

This unit will be marked anonymously, so you may wish not to put your name on your submission.

Notes

- This assignment is testing you on the use of MPI in C, so any use of other constructs (pthreads, SWAR, OpenMP, monitors etc.) will be marked down
- It is your choice on how to parallelise the solution, e.g., how to partition the workload
- The Azure VM can provide up to $4 \times 44 = 176$ cores on four nodes
- See the Unit web page and BUCS' web pages for how to use Azure
- If you have problems with Azure, email `hpc-support@bath.ac.uk` and give as much detail as you can about the problem. Note this is purely for problems with the machine: any other problems are your own!
- Note that the system will be running unattended over the Christmas period, and so has the potential to break and not be fixed promptly (as has happened in the past). Also note that the queue will get long. Neither is an excuse for a late hand-in
- This coursework is not a complex software engineering exercise: keep it clean and simple (a single `.c`/`.h` plus a single PDF is enough)
- This coursework is not intended to assess your ability to write C, but excessively poor code will hamper your ability to get high marks (always compile with maximum warnings set, e.g., `-Wall -Wextra -Wconversion` for gcc)
- Keep the code listing to a maximum of 80 columns width: super-wide lines are harder to read and tries to put too much into a single line
- Note that good commenting (helpful and informative, not excessive) is *very* important in parallel code as the computation is happening simultaneously across many chunks of code and is not limited to the piece of code in front of you
- Don't use screenshots, they are always difficult to read. If you need to include relevant data, cut and paste it into your document
- Don't include hundreds of data files and log files: use your judgement as to what is the right level of information
- Don't leave this assignment to the last minute: parallel programming is more subtle than you think
- The mark that appears on Moodle is provisional, and subject to revision. The SAMIS mark is definitive

Assessment Criteria

- **Normal standards for coding, testing and commenting apply**
- High marks will be gained by versions that are properly tested (with good evidence) and have a good scalability investigation including thoughtful commentary on topics including speedup, efficiency, scalability and other relevant information
- Medium marks will be gained by versions that work but have a limited amount of testing, or a limited scalability investigation, or the comments on scalability are weak
- Low to failing marks will be gained by versions that don't work or have poor testing, or have poor or no scalability investigation, or poor or no comments on scalability

This is individual coursework. Plagiarism is not acceptable: all work must be your own.
`https://library.bath.ac.uk/referencing/plagiarism`

If you have a good reason for an extension to the deadline for this coursework, please apply to the the Director of Studies, Zack Lyons: forms to apply for an extension are on Moodle (on the Student Zone page, not the Unit page).