| | | | |
|---|---|---|---|
| COMP1680 (2023/24) | **Clouds, Grids and Virtualisation** | | **Contribution: 60% of course** |
| **Course Leader: Dr Catherine Tonry** | **Coursework 2: Open MP** | | **Deadline Date: Wednesday 13/12/2023** |
| This coursework should take an average student who is up-to-date with tutorial work approximately 30 hours<br><br>Feedback and grades are normally made available within 15 working days of the coursework deadline | | | |
| **Learning Outcomes:**<br>Characterise and critically evaluate high performance computing based architectures and their suitability for given applications.<br>Implement and execute applications using shared and distributed memory programming paradigms.<br>Describe and critically discuss the roles and applications of cloud and grid computing. | | | |

**Plagiarism is presenting somebody else's work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student's coursework; stealing coursework from another student and submitting it as your own work. Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University. Please see your student handbook for further details of what is / isn't plagiarism.**

All material copied or amended from any source (e.g. internet, books) must be referenced correctly according to the reference style you are using.

Your work will be submitted for plagiarism checking. Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence.

**Coursework Submission Requirements**

• An electronic copy of your work for this coursework must be fully uploaded by 23:30 on the Deadline Date of **Wednesday 13/12/2022** using the link on the coursework Moodle page for COMP1680.

• For this coursework you must submit a PDF document and a zip file of your code. In general, any text in the document must not be an image (i.e. must not be scanned) and would normally be generated from other documents (e.g. MS Office using "Save As .. PDF"). An exception to this is hand written mathematical notation, but when scanning do ensure the file size is not excessive.

• There are limits on the file size (see the relevant course Moodle page).

• Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions.

• You must NOT submit a paper copy of this coursework.

• All coursework must be submitted as above. Under no circumstances can they be accepted by academic staff via email or other means.

The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences. See http://www2.gre.ac.uk/current-students/regs

**Detailed Specification**

***This coursework is to be completed individually.***

To complete this assignment you will need the source code provided at the following URLs.

https://moodlecurrent.gre.ac.uk/mod/resource/view.php?id=2601163

https://moodlecurrent.gre.ac.uk/mod/resource/view.php?id=2601162

You are provided with a two C program codes (called jacobi2d.c and gauss2d.c) that solve a rectangular 2-dimensional heat conductivity problem using the Jacobi and Gauss-Seidel iterative methods.

This code can be compiled and linked to produce a conventional executable file called jacobiSerial and gaussSerial by using the following commands:

gcc jacobi2d.c –o jacobi2d.out

gcc gauss2d.c –o gauss2d.out

To run the executable use the slurm scripts provided.
https://moodlecurrent.gre.ac.uk/mod/resource/view.php?id=2601167
https://moodlecurrent.gre.ac.uk/mod/resource/view.php?id=2601165

As you implement each of the following 4 steps make sure that you retain and do not overwrite previous versions of your solutions. You must submit 4 versions of each code one for each step.

Compile and execute the codes using the University HPC. Note this is a shared resource with a queue may become busy near the hand in date so make sure you give plenty of time to run your code and don't leave it to the last minute. If you are unsure how to use the HPC please check the lab notes and the instructions on Moodle.

Please ensure you follow these tasks carefully and with the code provided, work based on other Jacobi and Gauss codes will not be marked and receive 1 mark and not be marked further.

Optimisation an incorrect parallelisation can break your code so check it still gives the same answer at every step.

Your report should be 2000words and should not just be screenshots of your code and results, you need to discuss what you have done and why. Also note your code is not marked only the report so make sure you document any changes to your code in your report.

Complete all the tasks required:

**Task 1 (25 Marks)**

You are required to compute a temperature distribution for a rectangular 2D head conduction problem simulating a plate with boundary conditions set at top 10°C, bottom 30°C, left 40°C and right 50°C with a range of problem sizes. To do this you are required to modify the codes to:

1. reflect the boundary conditions described above
2. report the execution time Record the run-time of your code under a range of problem sizes using different levels of compiler optimization (e.g. -O1, -O2 etc).

Document your changes to your code in your report. Though you submit your code, you do not receive marks for separately so it is important you highlight changes in your report. Explain your changes.

**Task 2 (30 Marks)**

You are then required to modify the applications you created in step 1 to produce a basic parallel version of the codes using OpenMP. The following commands will compile your parallel version on a platform that has OpenMP installed:

gcc -fopenmp jacobiOpenmp.c –o jacobiOpenmp

gcc -fopenmp gaussOpenmp.c –o gaussOpenmp

The parallel codes must include timers to report the parallel run-time of the code. This version must be tested to establish correct operation using 1, 2, 4, 8 and 16 threads, regardless of performance.

Include in your report, the print out of the temperatures for a 20x20 problem size for 1,2,4, 8 and 16 threads to demonstrate the code works correctly.

Document your changes to your code in your report. Though you submit your code, you do not receive marks for it so it is important you highlight changes in your report. Explain the changes made.

Run the Gauss-Seidel code for only 1 iteration using 1 and 2 threads for a 20x20 problem size. Output the temperatures along with the timings, include this in your report. Discuss the reasons for the differences in the solutions.

**Task 3 (30 Marks)**

Using the university HPC are to run performance tests with the OpenMP implementation you created in step 2. This will require that you remove most of the print output from the code and increase the problem size to provide sufficient work to demonstrate useful speedup. You are expected to provide speedup results:

1. for at least three problem sizes, you are unlikely to see much speedup for small domains, use at least a 100x100 grid and a consistent tolerance, maximum of $10^{-3}$.
2. for a range of number of threads (from 2 up to 8 threads) In calculating the speedup of your parallel code you should use the optimized single processor version of your code you produced in step 1 and compare to this. You will need to apply similar compiler optimizations to your parallel code. Please list your runtimes in a suitable unit.

Please report both tour timings and the speedup from the serial version. Comment on the speedup, how does it compare to the theoretical maximum.

**Task 4 (15 Marks)**

Using different OpenMP directives and clauses you are to further modify your OpenMP application to improve the parallel performance. You are expected to provide results that permit comparison with those you obtained in Step 3. Comment on the differences between optimising the Jacobi and Gauss-Seidel Methods. Make sure you document the changes made to the code and explain why you have done them.

**Deliverables**

• A PDF file with your report

•A ZIP file with the source code for your solutions.

Your report is required to provide details of your implementation of steps 1 to 4 as described above. The report should include discussion of your solutions and provide a clear description of; the code changes you have implemented, your compilation and execution processes and your test cases. For steps 3 and 4 you are expected to provide tabular and graphical results. Comment on the differences between the two methods and the effect on parallelisation. Your zip file should provide suitably named source code files for each of your implementations. The report should be approximately 2000 words, excluding code snippets and results tables.

**Grading Criteria**

You will receive marks based on your code snippets, timings and discussion of your results. Please note your report should be written in proper English and you may loose marks for unclear writing.

Marks allocated according to the following rubric:

| | Step 1<br>25 Marks | Step 2<br>30 Marks | Step 3<br>30 Marks | Step 4<br>15 Marks |
|---|---|---|---|---|
| Step 1: 0-12 Marks<br>Step 2,3: 0-14 Marks<br>Step4: 0-7 Marks | An unsatisfactory step, lacking either timings of the code or changes to boundary conditions. | An unsatisfactory step, lacking the required parallelisation steps to the code. | An unsatisfactory step, there is no evidence of timing the code or it is very limited. | An unsatisfactory step, there is little evidence of optimising the parallel code. |
| Step 1: 13-14 Marks<br>Step 2,3: 16-17 Marks<br>Step4: 8 Marks | A satisfactory step, however, there are errors in the code or lack of detail in the report. | A satisfactory step, however, there are errors in the code or lack of detail in the report. | A satisfactory step, however, there are errors in the timings or lack of detail in the report. | A satisfactory step, however, there are errors in the optimisation or lack of detail in the report |
| Step 1: 15-17 Marks<br>Step 2,3: 18-20 Marks<br>Step4: 9-10 Marks | A good step, however, the code is mostly correct and documented but there is a lack of discussion of the results. | A good step, however, the code is mostly correct and documented but there is a lack of discussion of the results. | A good step, however, the timings are mostly correct and documented but there is a lack of discussion of the results. | A good step, however, the optimisation is mostly correct and documented but there is a lack of discussion of the results. |
| Step 1: 18-19 Marks<br>Step 2,3: 21-23 Marks<br>Step4: 11 Marks | A very good step, however, the code and timings are correct but | A very good step, however, the code is correct but there are a few | A very good step, however, the timings is correct but | A very good step, however, the optimisation is correct but there are a few |

| | | | | |
|---|---|---|---|---|
| | there are a few limitations in the report | limitations in the report. | there are a few limitations in the report. | limitations in the report. |
| Step 1: 0-12 Marks<br>Step 2,3: 24-26 Marks<br>Step4:12-13 Marks | An excellent step, however, the code and timings are correct but there is a lack of true critical analysis in the report | An excellent step, however, the code is correct but there is a lack of true critical analysis in the report | An excellent , however, the timings is correct but there is a lack of true critical analysis in the report | An excellent step, however, the optimisation is correct but there is a lack of true critical analysis in the report. |
| Step 1: 0-12 Marks<br>Step 2,3: 27-28 Marks<br>Step4: 14-15 Marks | An outstanding step, the code and timings are correct, and the report details them fully. | An outstanding step, however, the code is correct, and the report details them fully. | An outstanding however, the timings is correct, and the report details them fully. | An outstanding step, however, the optimisation is correct, and the report details them fully. |

**If you are unsure about any of these instructions, then please email the module leader or make an appointment to see them as soon as possible.**