Project 1 - OpenMP

Project 1 requires developing and submitting an OpenMP program and a written report. Project 1 is worth 20% of the total assessment and must be done individually. The due date for the submission is nominally **11.59pm on the Friday of Week 8**. Submissions received after this date will be deemed late. A penalty of 10% of marks per working day late (rounded up) will be applied unless approved prior to submission. The weekend does not count as working days, e.g. a submission on the weekend will be late but attract no loss of marks. A submission anytime on Monday will attract 10% loss of marks and so on. Please let the subject coordinator know if there is anything unclear or of concern about these aspects.

Assessment Tasks

1) Research parallel methods for the following problem: determine the minimum of the maximum flows between all pairs of vertices in a directed graph with weighted edges, where each weight is a real value greater than 0.

   1.1) For clarity, let removal of weight `w` from the graph mean removing any set of edges whose total weight equals `w`. Then, the maximum flow from a vertex `u` to a vertex `v` in such a graph is the minimum amount of weight required to be removed from the graph in order to eliminate all paths in the graph from `u` to `v`. The problem is to find the minimum of these maximum flows over all pairs of vertices in the graph.

This problem has particular importance in the design of future high performance interconnection networks for supercomputer architectures, because it represents a quantity that is similar to the more fundamental network bi-section width: the minimum number of edges required to be cut in order to separate the network roughly into two halves. We may cover some of these concepts in more detail later in the subject.

2) Choose/devise a parallel method, either that you have found in the literature, in part or in full, or that you have devised yourself, in part or in full, and implement it using OpenMP in the C/C++ language such that it will compile and run on a 4 socket cluster node provided via the Spartan HPC at Unimelb.

   2.1) Make sure to think about achieving as much performance as you can on a 4 socket cluster node on Spartan; that may require you to consider its NUMA characteristics, and other OpenMP techniques as discussed in the lectures.

3) Design and apply an appropriate methodology for testing the performance of your parallel implemention on a 4 socket cluster node. Your experiments should at the very least, i.e. to pass, show measurements of speedup versus number of threads/cores. You may also show more detailed measurements that demonstrate that the techniques you have implemented are indeed contributing to the performance of your approach; in order to achieve a higher grade in the project.

4) Write a minor report (2000 words not including figures, tables, diagrams or references) with the following sections and details:

   4.1) Introduction (250 words): define the problem as above in your own words and discuss the parallel technique that you have implemented. Present the technique using parallel pseudo-code, assuming a PRAM style parallel algorithm syntax as shown in lecture slides. Cite any relevant literature that you have made use of.

   4.2) Methodology (350 words): discuss the experiments that you will use to measure the performance of your program, with mathematical definitions of the performanc e measures and/or explanations using diagrams, etc.

   4.3) Experiments (350 words): show the results of your experiments, using appropriate charts, tables and diagrams that are captioned with numbers and referred to from the text. The text should be only enough to explain the presented results so it clear what is being presented, not to analyse result.

   4.4) Discussion and Conclusion (1050 words): analyze your experimental results, and discuss how they provide evidence either that your parallel techniques were successful or otherwise how they were not successful or, as may be the case, how the results are inclusive. Provide and justify, using theoretical reasoning and/or experimental evidence, a prediction on the performance you would expect using your parallel technique as the number of sockets in the single node were to increase, i.e. as the number of cores of such a single node were to increase to a much larger number; taking architectural aspects and technology design trends into account as best as you can - this may require some speculation.

4.5) References: cite literature that you have cited in preparing your report.

Use the latest ACM Conference Style guide for all aspects of formatting your report, i.e. for font size, layout, margins, title, authorship, etc.

Assessment Criteria

Assessment is divided between your written report and the degree of understanding you can demonstrate through your selection and practical implemention of a parallel solution to the problem. Your ability to implement your proposed parallel solution, and the depth of understanding that you show in terms of practical aspects, is called "software component". In assessing the software component of your project the assessor may look at your source code that you submit and may compile and run it to verify results in your report. Programs that fail to compile, fail to provide correct solutions to any problem instances, and/or fail to provide results as reported may attract significant loss of marks. The remaining aspects of the project are called "written report". In assessing the written report, the assessor will focus soley on the content of the written report, assessing a range of aspects from presentation to critical thinking and demonstration of a well designed and executed methodology for obtaining results and drawing conclusions.

The assessment of software component and written report is weighted 40/60, i.e. 40% of the project marks are focussed on the software component and 60% of the project marks are focussed on the written report.

Assessing a written report requires significant qualitative assessment. The guidelines applied for qualitative assessment of the written report are provided below.

Quality Assessment Guidelines

A general rubric that we are using in this subject is provided below. It is not criteria with each criterion worth some defined points. Rather it is a statement of quality expectations for each grade. Your feedback for your written assessment should make it clear, with respect to the quality expectations below, why your submission has received a certain grade, with exact marks being the judgement of the assessor. Bear in mind that for some assignments, independent discussions (paragraphs) in the assignment may be attracting independent assessment which later sums to total the final mark for the assignment. As well, please bear in mind that assessors are acting more as a consistent reference point than as an absolute authority. Therefore while you may disagree with the view point of the assessor in the feedback that is given, for reasons of consistency of assessment it is usually the case that such disagreement does not lead to changes in marks.

Quality expectations:

- >=80% - H1. A very good, excellent or outstanding discussion, with at most only minor improvements to conceptual expression or wording that can be identified. A grade in this range is generally considered to reflect the possibility of continuing with research higher degree study in the future and usually about 10% to 20% of students would be awarded this grade.
- 70%-79% - H2. A good discussion with no significant shortcomings, however there are one or more aspects of the discussion that can be clearly improved. Some concepts may be awkwardly expressed or in doubt.
- 65%-69% - H3. A reasonable discussion that addresses the question but with one aspect of the discussion that is significantly poor in writing style, understanding or missing all together. Usually 75% of students would receive a grade of H3 or above.
- 50%-64% - P. The discussion does not entirely address the question - it is considered to be off topic in some ways, and there is more than one aspect that is significantly poor in writing style, understanding or missing all together.

- 0%-49% - F. The discussion shows a clear lack of understanding/effort, or clearly misunderstood or underestimated what was expected and/or has significant writing style issues. Usually less than 5% of students would receive this grade.

When considering writing style, The “Five C’s of Writing” is adapted here as a guideline for writing/assessing a discussion:

- Clarity - is the discussion clear in what it is trying to communicate? When sentences are vague or their meaning is left open to interpretation then they are not clear and the discussion is therefore not clear.
- Consistency - does the discussion use consistent terminology and language? If different terms/language are/is used to talk about the same thing throughout the discussion then it is not consistent.
- Correctness - is the discussion (arguably) correct? If the claims in the discussion are not logically sound/reasonable or are not backed up by supporting evidence (citations), or otherwise are not commonly accepted, then they are not assumed to be correct.
- Conciseness - is the discussion expressed in the most concise way possible? If the content/meaning/understanding of the discussion can be effectively communicated with less words, then the discussion is not as concise as it could be. Each sentence in the discussion should be concisely expressed.
- Completeness - is the discussion completely covering what is required? If something is missing from the discussion that would have significant impact on the content/meaning/understanding of what it conveys, then the discussion is incomplete.

Submission

Your submission must be via Canvas (the assignment submission option will be made available closer to the deadline) and be either a single ZIP or TAR archive that contains the following files:

- **solution.cc**: Ensure that your solution is a single file, and include comments at the top of your program that show how to compile and run your program, including examples how to test the correctness and performance of your program, using additional test case input files if you require.
- **Report.pdf**: The only acceptable format is PDF
- test case inputs: Any additional files, providing test case inputs, that are needed to follow the instructions you provide in your solution’s comments section.