CSCI503: Parallel Programming

Final Project

1 Introduction

The goals of your project should align with the goals of this course. Ultimately, parallel computing is employed to solve the speed and/or the size problem. What problem has your project solved? And were you successful in meeting this goal? Your efforts will be evaluated along the following criteria:

1. **Technical merits.** Does your project embody underlining technical strengths? In other words, is it too simple? As a graduate-level course, we expect a certain level of sophistication in how your team applies the various parallel programming paradigms. While it is understandable that projects are not always successful, we intend to measure your final product as a function of efforts, maturity, and completion. Therefore, it is in your best interest to complete the project.
2. **Creative merits.** On the surface, one could argue that this evaluation metric is subjective. We argue otherwise, and our views are as follows. Does your project do the minimum implementation (e.g. typing in an existing algorithm)? What is your experiment's setup? Is it a canned configuration or does your team make (extensive?) modifications to suit your project's objective? Do you look at the problem and solution through a different set of lenses?
3. **Presentation of your results.** The presentation of your final product is two-folded. First, each team will present the project to the class. Second, your team must include a formal project report. We expect the presentation and final report to be high quality — as if you were to submit this project to a conference.

Finally, we cannot stress this point enough. Your work must be original. If you use an external source (whether it's source code, algorithm or ideas), you must cite source where appropriate. In the past, a few students have submitted reports that are cut-and-paste from the Internet. The outcome of their final grade, unfortunately, has been extremely unpleasant. When in doubt, write the report using your own words and give credit where credit is due.

2 Presentation

We will provide a template for your presentation, but you are not required to use it. In general your presentation should include the following:

- Title, team members, and affiliation (e.g. department and university).
- Background and motivation (the problem)
- Research innovation (your solution)
- A description of your implementation
- Findings and discovery: performance, improvements over previous algorithms, unexpected results.
- Conclusion and future work.

You are free (and encouraged) to include graphics, charts and/or other means to visually stimulate your readers. Your presentation is an opportunity to show case your "research" and accomplishments. Needless to say you should proof-read your presentation and be prepared to answer questions.

3 Report

All projects must include an evaluation and analysis of performance or another metric of "success" where appropriate. The analysis should also discuss challenges, unexpected results, and the adequacy (or inadequacy) of the parallel programming algorithm, model, or technology with respect to the problem.

Please take a look at the paper *Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers*, *http://www.davidhbailey.com/dhbpapers/twelve-ways.pdf*, to learn what you should NOT do when reporting your performance data.

Your team must submit both a report and your code via Blackboard. The final report must include:

- A short description of the problem to be solved and why you chose it.
- A short discussion of your choices of parallel platform and programming model.
- A high-level description of your implementation.
- Measurements of performance of your parallel implementation on the target platform. For most problems, you should include measurements of performance of a *baseline* version, which could be a sequential version of your implementation or a naive parallel version. If your problem is a scientific computation using floating-point operations, we recommend that you report FLOPS (absolute metric) rather than speedups. For Hadoop applications, a measure of throughput is often more appropriate. Please include a description of the execution environment and the experimental methodology.

- Analysis of the measurements reported: How much parallelism were you able to exploit? What (if any) are the performance bottlenecks? How does your algorithm scale with the number of processes/threads/workers, for a fixed input data set?
- Lessons learned: Discuss any unexpected results, the adequacy of the programming model and target platform for your particular problem, and contrast it with your expectations from the project proposal.
- Future work: Discuss potential future work and/or applications of your project.

4 Grading

Grading breakdown for your final project will be as follows:

- Presentation [20 points]
- Project report [20 points]
- Code (plus building instructions) submitted via Blackboard [10 points]

Presentations date: Monday, May 9 8-10 a.m. (attendance is mandatory)
Location: MHP B7B Note: Location may change if we are able to get a classroom from 8 a.m. to 12 p.m., in which case an announcement will be sent prior to the presentation date.

Please bring a printed copy of your final report to the project presentation session. One report per team.