COSC 420 - High-Performance Computing
Project 3

Dr. Joe Anderson

Due: 9 December 2019

1 Description

Working in groups of two or three, you will implement an algorithm or data structure of your choice, subject to instructor approval. Your project should have some aspect of Monte-Carlo method or random simulation.

Some suggested topics:

- Large-scale traffic network flow simulation
- N-body simulation (Barnes-Hut algorithm)
- Ising model, using Metropolis sampling
- Cache hit/miss simulation
- Parallel mesh generation
- Parallel Fast Fourier Transoform
- Heat-equation PDE solver

For whichever topic you implement, also implement some visualization of the algorithm and the data it processes. Recording timing data to display the efficiency of your algorithm, when given more computational hardware. The visualization/demonstration can be a sequential, perhaps that renders the results of a longer-running parallel computation.

Finally, you will give a short (10-15) minute presentation of your software in the HPCL for the class final. Your presentation should:

- Build up the basic concepts needed to understand the problem
- Explain how the problem would be solved sequentially
- Explain how a solution can be implemented in parallel
- Give real-world applications where the problem is useful, and demonstrate your software on a "toy" version of that problem.

Include a `README` file to document the problem, its solution, your code, and other required information. Include a `Makefile` to compile the project. Be sure to include full and thorough documentation.

2 Submission

Submit a `.zip` file called `Project3[GroupName].zip` containing your source code, `Makefile`, and documentation, then upload it to the course MyClasses submission page.

Second, print out your code and documentation, to be turned in on the due date (typically at the start of the following Lab). Include printouts/logs of successful operations of your program.

3 Bonus

If any bonus are completed, be sure to note it in the README file and provide appropriate output to demonstrate its correctness.

1. (5 pts) Document your project thoroughly, and make it generally available on a standard source-control site like Github.
2. (10 pts) Create a GUI to run the visualization part of the program, which allows the user to tweak the problem data and/or parameters, then display the results.