COSC 420 - High-Performance Computing
Project 1

Dr. Joe Anderson

Due: 11 October

1 Description

Working in groups of two or three, you will solve (at least) three problems from the problem database found at projecteuler.net. There are hundreds(!) of of problems to choose from, so you must also adhere to the following restrictions:

1. No two groups may submit solutions to the same problem. When you assign yourselves to a group on MyClasses, change the name of your group to reflect the three problems you chose (by ID from the Project Euler website), and email the course instructor with your selection.
2. Your problems should show a range of difficulty (as rated by the Project Euler webpage): you should solve problems in at least the 10th, 40th, and 60th percentiles (mouse over the yellow bar in the "Difficulty" column to see the percentile). Note that you need to make an account and be signed in to see the difficulty ratings!
3. Choose problems that allow you to take advantage of parallelism. In your project description and documentation, describe your algorithmic approach to the problem, and analyze the speedup and efficiency of a parallelized solution.
4. Choose problems on diverse topics. That is, they should not all be focused on number theory, graph theory, strings, etc. If you are in doubt, consult with the instructor to see if you're selection is diverse enough.

Your solutions **must use parallelism and parallel programming**. Include with your code `sbatch` scripts to take advantage of the HPC cluster. This means that you must work to develop an algorithm that has a definite theoretical speedup over the sequential version. In your documentation, make this explicit, with an explanation of your approach. Document which parallel programming constructs you use; try to take the most advantage of the MPI tooling that you can. Try using custom MPI types, various point-to-point communication, collectives, and topologies to help your program be more efficient.

Also include details of how your solutions evolved as you developed them. What went wrong? What are the inherent difficulties in solving the problems, and how did you overcome them? If the problems are "too hard" for simple brute force, what analytical tools did you employ to reduce your computational burden?

Design and implement separate programs to solve each of the three problems (but they may of course share code for common, useful utilities that you write yourselves). The solution to the problem should be printed out to standard output in a way that it can be checked by submitting it to the `projecteuler.net` site.

Record both the time and number of operations performed by your solution, as appropriate for the problem; count an operation as being any single arithmetic operation on one of the data elements (comparison, addition, multiplication, assignment, etc.). For the sake of getting the fastest solution possible, you may use a flag or command-line parameter to turn the benchmarking features on and off.

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

2 Submission

Submit a `.zip` file called `Project1[LastName].zip` (with `[LastName]` replaced with your own last name) containing your source code, `Makefile`, documentation, and testing output, then upload it to the course MyClasses submission page. Your submission must include checkable output (i.e. can be entered into the Project Euler page to see if the solution is correct).

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. Currently, the most recent problems (and consequently some of the least-solved) include problems 716-725. Your group may submit a solution to any of these, in addition to the three above, for a bonus of 10 points.

2. Problem #696 is currently the least-solved problem on the site, only solved by 99 users! *Any number of groups may submit a solution to this problem*, for a bonus of 10 points, similar to the above. This and the problems listed above will be available to submit all semester! But your solution must be original, and must take advantage of the HPC cluster.

3. For five bonus points, use GitHub, GitLab, BitBucket, or another public source-control platform to host your group's code and organize your joint efforts. Be sure to document that you did this in your README.