

Project Planning Timeline

Input data samples and sequential algorithms have been provided for each project. Please be sure to take a look at them and understand them as soon as possible—hopefully, you have already done so as part of choosing your project.

You should already have received notice of Gitlab repositories. Feel free to use it to share code amongst your teammates, or to use something else (non-public, of course). Gitlab does support issue tracking, so unless your team has a strong preference for other version control systems, I suggest making use of it. Also note that you must use Gitlab to turn in your code for milestones.

For GPU execution, we will test your project code using the RAI interface. You should each have credentials for using RAI soon, and a brief set of instructions for use (there are also public introductions on the Github page and from previous semesters). Each project will have at least one docker with inputs for the project. We may create additional dockers if more inputs become available. Dockers should be available soon, but you need not wait: the docker just means that files have already been uploaded to the server—you can include any files that you want in your directory when you execute under RAI.

Thursday 9 April (11:59:59 p.m.): Milestone 1

Look into existing parallel approaches (you may want to look more broadly than GPU solutions). Write up what you've found in a one-page summary including links or citations to any implementations and place a copy in your Gitlab repository. For the projects that provide a sequential solution (#1 and #3), be sure that each of your team members is able to execute the sequential version and understand how to compare the results. For the projects without sequential versions (#2 and #4), be sure that each of your team members understand the input format and the functions that you plan to use for perceptron implementations and final output calculations; you may also want to start writing some of the code that you'll need for milestone 2.

Thursday 23 April (11:59:59 p.m.): Milestone 2

Implement an initial CUDA implementation that produces correct / reasonably competitive results based on your project's evaluation criteria. Be sure to record timing information, as you will need it for comparison with your optimized versions in the final report.

What to turn in: Submit a working version of your CUDA implementation (as a directory that can be launched with your project's docker under RAI) to your group's Gitlab repository. Also include (as a writeup) initial timing results for your CUDA implementation and speculate as to how you can improve performance through optimization. Be specific about expected reuse, coalescing, control divergence, and so forth. Keep in mind that the writing you do now is likely to be reused in your final report.

Tuesday 28 April: Exam 2

Take-home, as discussed earlier.

Thursday 7 May (11:59:59 p.m.): Milestone 3

Implement at least three optimizations to your CUDA implementation and measure the impact of each on the overall performance.

What to turn in: Submit a working version of your CUDA implementation (as a directory that can be launched with your project's docker under RAI) to your group's Gitlab repository. Also include (as a writeup) detailed explanations of the optimization strategies, the changes necessary to the code, the expected impact on performance (why one expects a benefit, and how much—similar to the in-class analysis), the actual (measured) benefit to performance, any difficulties in realizing the implementation, and any issues with variations in the correctness of the results (or a statement that the results are indistinguishable or identical, based on your comparison methodology from the earlier milestone). Explanation of any interesting aspects of performance, and mention of any unexplained performance behavior (these are to be investigated for the final report).

Tuesday 12 May (11:59:59 p.m.): Project Reports Due

These should be about 10 pages long (including figures, text, and references to any sequential or parallel literature that you learned from and/or built upon) and should cover everything that you have been able to accomplish for the project.

Notes:

- The deadline for final projects gives you some flexibility to work around your final schedule but leaves a few days for us to grade, so please do not ask for an extension. Instead, if your team has too many finals in the last day or two, plan to finish earlier.
- Milestones are intended to ensure that you are putting time into the project and not planning to do everything at the last minute. For this reason, we will grade the milestones for completion (in other words, whether or not you turned something reasonable-looking in), but not in detail. Each will be worth 10 points (of 100 total).