Grading

- Exams: 40%
  - Exam 1: 20% -- 10 March 7:00 to 10:00 p.m.
  - Exam 2: 20% -- 28 April 7:00 to 10:00 p.m.
- Online participation: 10% (new category)
  - watch the videos on Echo360
  - (it has stats that tell me what each student watched)

Grading

- Labs (Machine Problems): 35%
  - Passing Datasets 90%
  - Reasonable-looking answers to questions
- Project: 25%
  - Demo/Functionality/Coding Style: 50%
  - Performance Ranking with full functionality: 50%
  - Detailed Rubric will be posted

Lectures and Office Hours

All lectures are now to be viewed online via Echo360.

I’ll use the stats tracking to measure your participation, so please be sure to view the videos and keep up with the course.

Staff will hold office hours with Zoom (times may or may not need to shift).

Exam 2

Exam 2 rules and style will be different (similar to how 408 worked years ago):
- Do on your own at home.
- 3 hour block from among 24-30 hours (you pick and monitor your time, subtract any short breaks).
- Open everything except other people (please don’t discuss until the week after).
- Mostly multiple-choice questions.
  I may ask you to run some code (with RAI).
Project Proposals (Extra Credit, Not Required)

Here’s what you need to produce…

1. a description (a paragraph, a page, a URL—it’s a pitch to other students)

2. one or more input sets that require 1-20 minutes to solve on a CPU

Project Proposals

3. a way to evaluate correctness (yes/no) OR goodness (a number) for an output—must be quantitative and well-defined
   • diff is fine, but probably overly restrictive for GPU
   • GPUs often use a threshold on relative error: $1E-6$ or $1E-9$ for each output value, for example
   • Any domain-specific measure that groups can understand how to compute (code is best)

4. Timing suggestions (I reserve the right to overrule): for example, given the problem, does it make sense to include the cost of input-reorganization (single-use kernel) or to ignore such costs (multiple-use kernel)? What about copying to GPU memory and back? Timing will (of course) be the same for all groups doing any project, so it’s part of the project spec.

5. “correct” outputs for each input set

6. Pseudo-code or a sequential C/C++ implementation that compiles under RAI’s variant of Linux (pointing to a million-line code base is unlikely to attract other students)

7. Feel free to add spiffy front-end/back-end input creators / output visualizers, but those cannot be the basis for correctness / goodness.
Project Selection
Each group must identify the project that they want to do (one vote per group, which is binding). Projects that receive 7+ votes are allowed, and groups that voted for them cannot change their choices. Groups that did not vote for an allowed project can then pick any allowed project. Each member of team that proposed each allowed project gets +5 extra credit for their project grade (+1 overall).

Meaning of Extra Credit
Top line: Other students’ extra credit has no effect whatsoever on your grade.

Detail:
I assign grades ignoring extra credit. Extra credit may then raise a student’s grade.

Project Milestones
Specific dates will be posted on web page later. Roughly…
1. Look into related work.
2. Get a sequential version working in C.
3. Get a simple GPU version working.
4. Optimize the GPU version.
5. Write a final report.

Project Development
I will request UIUC Gitlab repositories for each group. (It has integrated issue-tracking—try it!) You are welcome to use your own Gitlab / Github / whatever, but
• be sure that you can deliver your code to us,
• be sure that it works on RAI, and
• please don’t make your repository public.
Project Development

I encourage you to try/use
• pair programming, code reading, and so forth
• OVER Zoom!
• (The fewer people you meet in person, the better.)
Please be helpful to those who have little/no prior experience with Gitlab/Github/etc.

Project Execution

We will evaluate your projects using RAI, so be sure to test them there!
You will receive RAI credentials soon.
RAI clients exist for Linux and Mac.
For Windows, we suggest WSL (Windows Subsystem for Linux) running Ubuntu (or your favorite variant of Linux), and using the RAI client that way.