ECE 408: Applied Parallel Computing **Spring 2019 ZJUI**
Prof. Steven S. Lumetta

Project Planning Timeline

Tuesday 9 April (same day as the midterm)
Let me know your teams (or be assigned to one).

Thursday 11 April
In-class presentations about the possible projects. Each team selects one of the three projects (one team per project; we can flip coins if more than one team wants one project).

Friday 19 April (11:59:59 p.m.): Milestone 1
Obtain input data samples. Identify viable sequential algorithms. 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 algorithm names and links.

Friday 26 April (11:59:59 p.m.): Milestone 2
Implement a working sequential solution in C for at least one of the sequential algorithms. This sequential version will be the baseline for your parallel implementation, although it may be possible to do a better job on a CPU (you just need a point for comparison). Soon after this checkpoint, you should talk with the faculty contact for your project and try to verify that the answers produced by your sequential code are reasonable.

What to turn in: You should be using SVN to manage your code, but be sure that there's a copy committed along with at least some input data. Provide instructions as to how to execute your sequential code on your sample data set. Explain any difficulties that you encountered in implementing your sequential solution, and comment on the potential improvements (algorithmic or simply use of vector units/SSE, for example) to the sequential performance. Also summarize the output that you've seen and the timing you've observed using your sequential code (just rough numbers—no need to be too careful yet).

Friday 10 May (11:59:59 p.m.): Milestone 3
Implement an initial CUDA implementation and develop a methodology for checking that your parallel code produces correct results (by comparing it with your sequential code's output). Note that exact matches to floating-point computations are unlikely, so you may need to set a threshold on fractional error observed or something similar. Talking to the domain experts may help you to make this check more reasonable and useful.

What to turn in: Again, both your sequential and CUDA implementations should be committed to SVN by the deadline. In the writeup, present data samples and validation results for sequential output. You should have a large enough sample that you can claim that your sequential code produces correct results. Present and explain these data in your writeup. Also present 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.

Friday 17 May (11:59:59 p.m.): Milestone 4
Implement at least three optimizations to your CUDA implementation and measure the impact of each on the overall performance.

What to turn in: 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). Performance comparisons with sequential execution time as a function of input size (if appropriate), explanation of any interesting aspects of performance, and mention of any unexplained performance behavior (these are to be investigated for the final report).

Tuesday 21 May
In-class presentations of project results: 25 minutes for each team: 20 presentation and 5 Q/A. These should be accessible to other class members, so you may assume some knowledge of CUDA and programming, but no knowledge of your application, algorithms, or other information. Start with background/motivation, then state the problem, explain the sequential algorithm chosen as a baseline, illustrate the basic CUDA approach (where is the parallelism?), talk about challenges, and talk about optimizations. Each person should play a part in the presentation.

Thursday 23 May: Exam 2
Review session (if desired) for exam; exam 2 held in evening.

Friday 31 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 in the presentation and everything that you are able to accomplish after the presentations. Include details of how your results are validated to make sure that what you produce is useful to the faculty contact. Remember that the intent of the projects is to give you potential starting points for doing some enabling work and starting to get involved in research in the summer or next year, so do your best to make what you've done clear. I will share your reports with the contact person for your project, so while you do need to write up the details of your optimizations (for me), you also need to be sure to highlight the benefit to overall application runtime for a more general audience.