CS 179 Spring 2017

Project Information

You will be given four weeks to design and implement a GPU-accelerated project of your choice, working alone or in pairs. Teams of three must be approved by a TA and must be proposing a substantially-sized project. If you do choose to work with others, we will expect a proportional increase in the project's scope. A general rule of thumb is to have at least one CUDA module per person. For example, a two-person team can create a raytraced visualization of an N-body simulation. In this case, one person will write a CUDA based raytracer and the other person will write a CUDA based N-body simulator.

Due Dates

In addition to turning in the code to compile a working executable at the end of the term, there are a few additional tasks we would like you to complete: a proposal, a CPU-based proof-of-concept, and a final write-up.

- Proposal: 11:59 PM, (Mon) 5/22/2017
- CPU Demo: 3 PM, (Friday) 6/2/2017
- Final Version & Write-Up: 3 PM, (Friday) 6/9/2017

Proposal

- 1-3 sentence summary of project
- 1-3 paragraph explanation of project with background
- Why is this challenging? Has it been done before? What tricky things are you going to have to figure out? 1-2 paragraphs
- What are the deliverables? Goals? 1 paragraph
    - If you're in a two-person team, clearly describe the CUDA module each team member is building
- Timeline: What are you going to do each week?

CPU Demo

Because debugging your projects on the GPU could be fairly difficult, we are asking you to provide a rough CPU-based proof-of-concept a week before the final version is due. This CPU demo should have all of the final expected features but with the CUDA

portions written as CPU loops. This portion should be clearly marked using comments that explain how the CPU loops will be translated into CUDA.

Write-Up

Your final project should be submitted with a comprehensive readme outlining everything we need to know about the program. This document should be clear and concise.

A sufficient readme should have all of the following:

- Installation/Usage instructions
- Explanation of what the program does
- Expected results
- Analysis of performance

Project ideas
Raytracer
Random numbers
16bit fast transpose
Diffeq solver
Finite automaton
Parallelize an algorithm. Work must be original.
Cryptocurrency
Multi machine computation
Matrix factorization
Branch and bound
Interval analysis

(see the powerpoint)

Resources

There will be in-class office hours on Fridays until the final due date. These will be in addition to normal office hours. Please utilize these office hours for getting feedback and help with projects.

Have fun!