CS 677: Parallel Programming for Many-core Processors

Instructor: Philippos Mordohai (Philippos.Mordohai@stevens.edu)

Textbook

Programming Massively Parallel Processors: A Hands-on Approach
by David Kirk and Wen-mei Hwu
Morgan Kaufmann, 2012 (2nd edition)
ISBN: 978-0124159921

Office Hours: Tuesday 5-6 in Lieb 215 and by appointment

Objectives

The objective of the course is to provide the students with knowledge of the state-of-the-art hardware architectures and programming philosophies for gaming, machine learning, scientific computation, simulation, and visualization. The emphasis will be on the NVIDIA’s CUDA, which currently is the most widely used parallel computing architecture.

Evaluation

Homework assignments 40%
Midterm 15%
Participation in class 5%
Final project 40%

Policies

Collaboration Policy. Homework assignments will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems. Use of the Internet is allowed, but should not include searching for previous solutions or answers to the specific questions of the assignment. Any sources used must be cited. We will assume that, as participants in a graduate course, you will be taking the responsibility of making sure that you personally understand the solution to any work arising from collaboration.

Late Submission Policy. No late submissions will be allowed without consent from the instructor. If urgent or unusual circumstances prohibit you from submitting a homework assignment in time, please e-mail me explaining the situation.
Class Schedule

Week 1: Introduction to massively parallel programming and CUDA (Kirk & Hwu Ch. 1, 2 and 3)

Week 2: CUDA threads and atomics; CUDA memories (Kirk & Hwu Ch. 4 and 5)

Week 3: Performance considerations (Kirk & Hwu Ch. 6)

Week 4: More performance considerations and floating point considerations (Kirk & Hwu Ch. 6 and 7)

Week 5: Case study: MRI reconstruction (Kirk & Hwu Ch. 8) and timers.

Week 6: Project ideas; convolution, constant memory and cache (Kirk & Hwu 8)

Week 7: Parallel patterns: reduction trees, prefix sum; sparse matrix and vector operations; summed area tables (Kirk & Hwu Ch. 9 and 10, notes)

Week 8: Midterm; introduction to Thrust

Week 9: Project proposals; Case study: Molecular Visualization and Analysis; computational thinking (Kirk & Hwu Ch. 12 and 13)

Week 10: CUDA streams and more libraries (notes)

Week 11: OpenCL (Kirk & Hwu Ch. 14 and notes)

Week 12: Project mid-point presentations; More OpenCL

Week 13: Instructions for project presentations, posters and reports; binning; OpenGL interface; OpenACC; DirectCompute (notes and Kirk and Hwu Ch. 15)

Week 14: Project presentations

Note: case studies may be re-ordered or replaced during the semester according to technology changes, student interest and relevance to projects.