ECE 408 (CS 483) - Applied Parallel Programming

Instructors:

Applied Parallel Programming has been taught by Professor Wen-Mei Hwu in the past, who designed the class material and remains the course director. It is currently being taught by Professor Sanjay Patel during the Fall 2018 semester. There is usually 1 TA who assigns and grades the MPs.

Prerequisites:

ECE 220 (previously ECE 190) is listed as a prerequisite, as most of this course is taught in C. However, it is recommended that students acquire some more hands-on experience coding in C or C++ than ECE 220, as they will get more out of the course. Courses such as CS 225 and/or ECE 391 would provide a particularly strong background for this course.

When to Take It:

The course is typically offered during both the Fall and Spring semesters. Most students take this technical elective during their Junior or Senior year.

Class Content:

The primary framework used in this class is CUDA, a C-like language maintained by NVIDIA that supports programming on GPUs and GPU clusters.

Much of the class is spent comparing and contrasting serial vs. parallel implementations of various widely-used computing algorithms. Each algorithm typically introduces new CUDA features, which students utilize in the MPs.

Some of the main topics covered in this course include:

- CUDA programming language
- Parallel software design
  - Performance enhancements
  - Scalability
- Processor architecture features and constraints
- Applications of parallel programming
- State of the parallel programming industry

This course has also included a few guest lectures in past semesters, from people who use parallel programming in their work or research. Fall 2018 the guest lecturers included representatives from Jaunt XR.

Work:

For 4 credit hours, this class is not terribly work intensive during the semester. Students will need to put in extra time or focus during lectures to understand the underlying CUDA architecture and algorithm structure. The lecture material prepares students very well for the assignments, with important portions of the project are typically explained and analyzed thoroughly in lecture before the due date. Having a strong understanding of the problem and the parallel solution helps incredibly in test and MP performance.

Lectures are typically 75 minutes, 2 days a week, and there are no discussion or lab sections. The assignments consist of 6-7 MPs that are all written in CUDA and executed on a GPU cluster (either CSL GEM or WebGPU). As of Fall 2018, each MP has a set of questions for each problem, worth 10% of the MP. The questions ask the student to comment on the performance of their own code as well as to compare the characteristics of differing algorithms to solve the same problem.

This course also includes a final project component where students work in teams of 3-4 to apply the concepts they have learned throughout the semester. Teams are set to participate in a class-wide "competition" project. The competition is typically a standard problem that can be parallelized in a wide variety of manners, with students competing to design the most efficient parallel implementation. For Fall 2018, the competition was to design an optimized neural-network convolution layer forward pass. The final project is primarily graded on the checkpoint reports and optimizations used over the ~2 months given for the project. The ranking of speed accounts for 10% of the final project grade.

As of Fall 2018, there are two exams throughout the semester, which are 3-hour timed paper exams.

The exams have multiple choice, short answer, and code problems where you fill in the missing parameters. The content of the exams covers algorithms used in MPs as well as unused options brought up in lecture. Another significant topic is the effect of certain parameters and algorithms on the performance of the program.

Grade breakdown (for Fall 2018):
• Exam 1: 20%
• Exam 2: 20%
• Labs (Machine Problems): 35% (5% each)
  • Correctness (number of datasets passed): 90%
  • Report/questions: 10%
• Project: 25%
  • Milestone 1: 5%
  • Milestone 2: 10%
  • Milestone 3: 10%
  • Milestone 4: 30%
    • Optimization 1: 10%
    • Optimization 2: 10%
    • Optimization 3: 10%
  • Milestone 5 (Final optimizations): 30%
    • Optimization 4: 10%
    • Optimization 5: 10%
    • Optimization 6: 10%
• Performance Ranking: 10%
• Report Style: 5%

**Life After:**

If you really enjoyed this course, check out CS 420 (ECE 493) - Parallel Progrmg: Sci & Engrg.

If you are interested in working on the CUDA language itself or contributing to open parallelism platforms such as OpenMP, consider applying for internships at companies such as NVIDIA, AMD, and Intel.

If you are open to applying CUDA or parallel computing in industry, there are a wide variety of opportunities available.

In addition, expertise in parallel programming is highly valued in the academic community, particularly in research areas that deal with large amounts of data such as computational genomics and biomedical imaging. There are many opportunities for undergraduate and graduate students to get involved with research on campus if they are competent in parallel software design.