EEC 289Q – Modern Parallel Computing

4 units – Winter Quarter; alternate years
Lecture: 3 hours
Prerequisite: Required: ECS 30; optional but desirable: EEC 170 or ECS 154a.
Grading: assigned projects (30%), final project (60%), classroom participation (10%).

Catalog Description:

Exploration of the architecture of modern parallel computers, their programming models, and their programming systems.

Expanded Course Description:

This course focuses on modern parallel computing and for this particular offering of the course, GPU computing. We will explore using the programmable GPU as a parallel computer, primarily using the CUDA programming language (an extension to C/C++). We will cover the architecture of the GPU and its programming model; the CUDA programming language; fundamental data structures and algorithms on the GPU; numerous application domains and how they can be expressed on the GPU; programming models and high-level languages for GPU computing; and current research challenges in GPU computing. We expect that students who successfully complete this course will be ready to use GPU computing in their own projects and research, and/or be ready to conduct GPU-computing research on their own.

We assume that students who take the course will have experience using the C/C++ programming language; prior experience in any of computer architecture, computer graphics, algorithms, and data structures will also be useful. All assignments will use C/C++.

The assigned projects in this class will contain significant design elements that allow students to design and implement parallel solutions to computationally challenging problems, to analyze and improve their performance, and to use these solutions to understand the architectures, programming models, and programming systems of these modern parallel computers.

1. History of modern parallel computing
   - Hardware predecessors (vector machines, massively parallel machines, graphics processors)
   - Software predecessors
2. Architecture of modern parallel processors
3. Programming model of modern parallel processors
4. Programming systems for modern parallel processors
5. Fundamental parallel primitives
6. Survey of computational motifs ("dwarfs") and parallel implementation strategies
7. Optimization techniques
8. Heterogeneity and multi-node issues
9. Application case studies
10. Future directions

Course Overlap:

EEC 171, EEC 270, and EEC 277 all summarize the material in this class at a high level (for each of these courses, roughly 5-10% of class time is spent on modern parallel computers, mostly from the hardware perspective). ECS 158 covers the programming of numerous parallel architectures, one of which is the GPU. ECS 223 covers parallel algorithms, some of which could be applicable in this course.