CS 4230: Parallel Programming (3 units)

Schedule: T-Th 2:00PM – 3:20 PM, WEB 1230
Location: WEB 1230

Instructors:
Mary Hall, MEB 3252, mhall@cs.utah.edu; Mon: 3-4PM; Thur: 4-5PM
Mark Baranowski, MEB 2180, mark.s.baranowski@gmail.com; Tues: 12:45-1:45PM; Wed: 4-5PM

Course Summary
This course is a comprehensive exploration of parallel programming paradigms, examining core concepts, focusing on a subset of widely used contemporary parallel programming models, and providing context with a small set of parallel algorithms. In the last decade, this area has been the subject of significant interest due to a number of factors. Most significantly, the advent of multi-core microprocessors has made parallel computing available to the masses.

At the high end, supercomputer architectures are diverse, and parallel programming strategies tend to be influenced heavily by the architecture. We will talk about current strategies for how to achieve performance portability, such that the same code can perform well across platforms. Desktop and server architectures and embedded devices can also be thought of as multiprocessors. While the programming strategies vary in these distinct markets, the underlying concepts for how to write correct and performant parallel code remain the same.

Grading

<table>
<thead>
<tr>
<th>Percent</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>40%</td>
<td>Programming Projects (P1, P2, P3, P4)</td>
</tr>
<tr>
<td>20%</td>
<td>Written homeworks and in-class assignments</td>
</tr>
<tr>
<td>25%</td>
<td>Midterm and final exams</td>
</tr>
<tr>
<td>10%</td>
<td>Mock SC Cluster Competition</td>
</tr>
<tr>
<td>5%</td>
<td>Class participation, including working examples in class</td>
</tr>
</tbody>
</table>

Assignments
Individual assignments such as written homeworks and programming assignments will be submitted on Canvas. These must be your own work, and will adhere to the departmental policy on cheating. In-class programming and the mock SC project will be done in groups. Students will be asked to work problems in class and will be graded on class participation. Late homeworks are not allowed. Late programming assignments will incur a 20% penalty per day.

Classroom Behavior
All students are expected to maintain professional behavior in the classroom setting, according to the University of Utah Student Code, which is posted at http://www.regulations.utah.edu/academics/6-400.html.

Students should read the Code carefully and know that they are responsible for the content. According to Faculty Rules and Regulations, it is the faculty responsibility to enforce responsible classroom behaviors, beginning with verbal warnings and progressing to dismissal from class and a failing grade. Students have the right to appeal such action to the Student Behavior Committee.
Inclusivity Statement

It is our intent that students from all diverse backgrounds and perspectives be well-served by this course, that students' learning needs be addressed both in and out of class, and that the diversity that the students bring to this class be viewed as a resource, strength and benefit. It is our intent to present materials and activities that are respectful of diversity: gender identity, sexuality, disability, age, socioeconomic status, ethnicity, race, nationality, religion, and culture. Your suggestions on how we can make the course more inclusive and welcoming are encouraged and appreciated.

We also expect students to treat others in the class, including the teaching staff, with the same level of respect.

We take incidents of discrimination, bias, and harassment seriously. We will file reports with the Office of Equal Opportunity, Affirmative Action, and Title IX (OEO) about such incidents. If you are unsure what differentiates free speech and professional behavior from discrimination, bias, and harassment we are happy to have an open, judgment-free, and confidential conversation with you, or refer you to the OEO.

Students with Disabilities

The University of Utah seeks to provide equal access to its programs, services and activities for people with disabilities. If you will need accommodations in the class, reasonable prior notice needs to be given to the Center for Disability Services, 162 Union Building, 581-5020 (V/TDD). CDS will work with you and the instructor to make arrangements for accommodations.

Syllabus (subject to change)

August 20: L1: Course Introduction
Go over syllabus and provide context for course.

August 22: L2: Parallel Algorithm Basics
Learn about parallelism correctness and performance challenges. In-class exercises.

August 27: L3: Introduction to Parallel Architectures
Shared memory vs. Distributed memory, parallel programming models and execution models.

August 29: L4: Data-Parallel Programming with OpenMP
Programming Shared-memory architectures using OpenMP pragmas. Implement parallel pi code variations and submit.

Sept. 3: L5: Parallelism correctness
Learn the concepts of reordering transformations and data dependences.

Sept 5: L6: Data Dependencies in Practice
Apply your knowledge of data dependences and OpenMP to parallelize code examples.

**Sept. 10: L7: Thread Building Blocks**  
Library-based shared-memory parallel programming.

**Sept. 12: L8: Memory Hierarchy Optimizations, 1**  
Reordering transformations to improve data locality. Modeling performance based on loop order.

**Sept. 17: L8: Memory Hierarchy Optimizations, 2**  
Loop tiling and its performance impact.

**Sept. 19: L9: Task Parallelism**  
Learn about task parallelism. In-class programming assignment.

**Sept. 24: L10: Midterm Review**  
Work representative questions in class.

**Sept. 26: MIDTERM – in class**

**Oct. 1: L11: SIMD and SIMD instruction sets**  
Learn about vector instruction sets on CPUs, and how to program them using OpenMP.

**Oct. 3: L12: SIMD in-class programming assignment**  
Add SIMD instructions to CNN example

**FALL BREAK**

**Oct. 15: L13: Distributed Memory Architectures and Introduction to MPI**  
Learn how to program clusters using Message Passing Interface.

**Oct. 17: L14: MPI Communication Examples, Communicators**  

**Oct. 22: L15: MPI Pattern**  
Look at another MPI pattern to see how to program it.

**Oct. 24: L16: MPI Scatter, Gather, Scatterv, non-contiguous blocks**  
Establish communication patterns that are different from sending/receiving contiguous blocks.

**Oct. 29: L17, MPI, cont.**  
Final MPI lecture, complete Cannon algorithm description.
Oct. 31: **L18: Introduction to GPUs and CUDA**
Learn about Graphics Programming Units and how to program them.

**Nov. 5: L19: GPU Parallel Optimizations**
Learn how to develop correct GPU thread programs.

**Nov. 7: L20: GPU Parallel Pattern**
Reduction operation on GPUs.

**Nov. 12: L18: GPU Memory Hierarchy Optimization**
Learn how to optimize GPU code for unique memory hierarchy, global memory coalescing, shared memory and registers.

**Nov. 14: L19: Sparse linear algebra pattern**
Learn how to write spare linear algebra code for GPUs.

**Nov. 19: L20: Sparse, 2**

**Nov. 21: L21: Deep Learning**

**THANKSGIVING BREAK**

**Dec. 3: Finals preparation and Mock SC Student Cluster Competition**
Read the reproducibility paper from 2017 and be prepared to discuss. Prepare for interview.

**Dec. 5: Mock SC Student Cluster Competition**
Group interviews

**Dec. 10, 1PM, FINAL, In class**