Abstract

High-performance computing (HPC) is becoming increasingly important for efficient research in biomedical informatics. Meanwhile, all computers have become parallel—from laptops to supercomputers. This course teaches students how to program parallel computers. The emphasis is on techniques that improve the efficiency of the program and the productivity of the programmer. This course will be taught in conjunction with the collaborative on-line course *Applications of Parallel Computing* taught through the XSEDE project and the University of California, Berkeley. Registered students will receive accounts on XSEDE, a national supercomputing resource, for the duration of the course, for the assignments and projects. Students will work on final projects that apply parallel computing techniques to their research in biomedical informatics.

1 Instructor Contact Information

E-mail  eric.peskin@nyumc.org

Office Phone  212-263-2287

Office Location  TRB (227 East 30th Street), #7-37

Office Hours  Fridays 2 – 3pm

2 Needed Background

Prior programming experience is assumed. Ideally, students should have some programming experience in C or similar language. The programming assignments will use C or FORTRAN. Picking up C along the way is certainly possible, but will require extra effort by the student. The instructor will also help.
3 Course Logistics

This course will be taught in conjunction with the collaborative on-line course *Applications of Parallel Computing* taught through the XSEDE project and the University of California, Berkeley. The course includes online video lectures, auto-graded quizzes and homework assignments. Registered students will receive free accounts on XSEDE, an NSF-sponsored national supercomputing resource, for the duration of the course.

The local instructor at NYULMC will guide and advise the students, provide supplemental material where appropriate, help those students learning C on the fly, guide the final projects, and assign grades.

A significant portion of the grade will rest on a final project. Students are encouraged to work in pairs for this project. The projects should apply the parallel-computing techniques learned in the course to research of interest to the students. The project consists of code, a written report, and an oral presentation.

There are two on-line lectures per week. In addition to the online lectures, the course will meet locally on Mondays from 3pm to 5pm.

4 Schedule of Topics

A tentative weekly schedule is as follows:

Week 1:  
- Introduction
- Single Processor Machines
- Homework 0

Week 2:  
- Introduction to Parallel Machines and Programming Models
- Sources of Parallelism and Locality in Simulation - Part 1

Week 3:  
- Sources of Parallelism and Locality in Simulation - Part 2
- Shared Memory Programming: Threads and OpenMP
- Programming Homework 1

Week 4:  
- Distributed Memory Machines and Programming
- Partitioned Global Address Space Programming with Unified Parallel C

Week 5:  
- Introduction to GPUs by Bryan Catanzaro
- Dense Linear Algebra - Part 1
- Programming Homework 2 (Part 1)

Week 6:  
- Dense Linear Algebra - Part 2
- Graph Partitioning
Week 7:  
- Automatic Performance Tuning and Sparse-Matrix-Vector-Multiplication (SpMV)  
- Hierarchical Methods for the $N$-Body Problem  
- Programming Homework 2 (Part 2)

Week 8:  
- Structured Grids  
- Cloud Computing with MapReduce and Hadoop, by Matei Zaharia  
- Final Project Proposal

Week 9:  
- Programming Homework 3  
- Architecting Parallel Software with Patterns, by Kurt Keutzer  
- Parallel Fast Fourier Transform (FFT)

Week 10:  
- Dynamic Load Balancing  
- Blood Flow Simulation at the Petascale and Beyond, by Richard Vuduc

Week 11:  
- Simulating the Brain, Cat out of bag, by Rajagopal Ananthanarayanan  
- Parallel Climate Modeling by Michael Wehner

Week 12:  
- Frameworks for Structured Software Development by John Shalf  
- Exascale Computing by Katherine Yelick

Week 13:  
- Final Project Presentation

Week 14:  
- Final Project Report

5 Evaluation Criteria

Table 1 shows the graded items and the weight of each item’s contribution to each student’s overall score for the course.

<table>
<thead>
<tr>
<th>Weight</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 %</td>
<td>Quizzes</td>
</tr>
<tr>
<td>40 %</td>
<td>Homework</td>
</tr>
<tr>
<td>50 %</td>
<td>Project</td>
</tr>
</tbody>
</table>

Cheating and plagiarism will not be tolerated. You must cite your sources. *If you submit someone else’s work as your own, you will fail this course.*