CSI 702

High Performance Computing

http://www.scs.gmu.edu/∼jwallin/c702s06

Dr. Brett Berlin and Dr. John Wallin

Wednesday 7:20-10:00, Innovation Hall Rm 133

Prerequisites

CSI 700 and 701, OR Permission of Instructor.

Description

This course emphasizes the role of computation as a fundamental tool of discovery in the development of science, and the role that computational power has in furthering this discovery. In the course, we will explore how scientists can effectively use high performance computing, computer clusters, and grids to solve scientific problems. During the semester, we will explore the software and hardware systems associated with these technologies, along with techniques need to take effective advantage of them. Several individual programming assignments will be completed in this course, as well as exams and a possible group project.

Topics

- **Software Optimization, Design, and Testing** - complex code development, standards, specifications, validation, verification, optimization
- **High Performance Computing Architectures** - clusters, grids, and machine architectures
- **Parallel Programming -** algorithms, libraries, and languages used in parallel programming
- **Grids and Clusters -** configurations, queuing jobs, issues with using systems

Texts

- *Code Complete: A Practical Handbook of Software Construction* by Steve McConnell *(recommended)*
- *Scientific Computing - An Introductory Survey* by Michael Heath *(recommended)*
- *Software Optimization for High Performance Computing* by Kevin R. Wadleigh and Isom L. Crawford *(recommended)*
- *Using MPI* by William Gropp, Ewing Lusk, and Anthony Skjellum *(recommended)*
- *Grid Computing : Making the Global Infrastructure a Reality* by Fran Bergman, Geoffrey Fox, and Anthony J.G. Hey *(recommended)*
- *The Grid 2: Blueprint for a New Computing Infrastructure* by Ian Foster and Carl Kesselman *(recommended)*

Grading

Each student must complete a number of short computational assignments and a group computational project. A final exam and midterm will be given. Short quizzes may be given in class as part of the class participation grade. No makeup will be given for missed in-class quizzes. The weights for each section are:

- 50% for homework & group project
- 20% final exam
- 20% for midterm exam
- 10% for class participation/quizzes

Honor Code

As in any class, you are allowed to study with other students. However, tests (in class and take-home) and homework assignments must be completed on your own unless you have written permission to do otherwise from your instructor. SPECIFICALLY - YOU MAY NOT COPY ANY TEXT, COMPUTER CODE, OR MATERIAL AND REPRESENT IT AS YOUR OWN WORK. In some assignments, you will be given written permission to use existing code in your projects, but but you MUST cite the source it came from. Any violations of this will be considered plagiarism and will be treated as a violation of GMU's academic honor code. Other violations of GMU's honor code will be treated similarly. Please see the Honor Code page for details.

Instructor Information

jwallin@gmu.edu
S&T I, room 109
http://www.scs.gmu.edu/∼jwallin
Office hours - Wed 5:00-6:00, Thurs 3:00-4:00, other times by appointment or drop in as available.
berlin1@bigplanet.com
Office hours by appointment.

Preliminary Class Schedule

- **January 25 -** Lecture 1: Goals and History of High Performance Computing
  1. why do we need faster computers? - dimensions, resolution, physical realism
  2. early computers and computer centers
  3. Cray - optimizing computers with vector pipelines
  4. Early parallel machines - the problems with proprietary design
  5. Commodity computing - Beowulf clusters

6. Parallel machines - revisited
7. Grids

- **February 1 -** Lecture 2: Computer Architectures - Grids, Clusters, and SMD
  1. clusters - topology, communication, and limitations
  2. grids - topology, communication, and limitations
  3. SMD - topology, communication, and limitations
- **February 8 -** Lecture 3: Software Design Cycle and Problem Taxonomy
  1. what are the characteristics of good software? trade-offs in optimization
  2. when is high performance necessary?
  3. communication vs bandwidth- the curse of latency
  4. problem taxonomy - why are some problems harder than others?
  5. case studies in parallelism
- **February 15 -** Lecture 4: Software Optimization for Single Processor Machines
  1. how do compilers change programs into executable code? a short summary of compilation
  2. modern CPU design - pipelines, dual cores, speculative execution
  3. cache misses and paging
  4. data io
  5. setting performance goals
  6. profiling and targeted optimization
- **February 22 -** Lecture 5: Software Optimization on Clusters - MPI
  1. a short review of cluster architecture
  2. software for running programs on clusters
  3. introduction to MPI
     (a) minimal commands in MPI
     (b) language variation of the language
     (c) global messages
     (d) point to point communication

- **March 1 -** Lecture 6: Intermediate and Advanced MPI
  1. asynchronous communication
  2. defining data types
  3. creating virtual topologies
  4. profiling in MPI
- **March 8 - MIDTERM EXAM**
- **March 15 - SPRING BREAK**
- **March 22 -** Lecture 7: Using Grids for Computing
  1. free software for creating grids
  2. commercial software for grids
  3. characteristics of programs suitable for grids
  4. data transfer
  5. case studies
- **March 29 -** Lecture 8: Software Optimization for Grids
  1. adapting programs to run on a grid
  2. Perl and shell scripts
  3. using scripts to manage a suite of programs
  4. managing the project using scripts
- **April 5 -** Lecture 9: Optimizing Software for Grids
  1. Globus
  2. Grid Forum
- **April 12 -** Lecture 10: SMD architectures
  1. the Altix - hardware overview
  2. internal CPU design
  3. communication design and performance
  4. OpenMP

- **April 19 -** Lecture 11: Optimizing SMD Programs
  1. tools for optimization and profiling
  2. programming with shared memory
  3. Unified Parallel C (UPC)
- **April 26 & May 3 -** Lectures 12 & 13 - Advanced Topics
  1. managing the flood of data from high throughput computing
  2. simple solutions - naming conventions, meta-data, logs
  3. visualization
  4. access and archiving data
  5. Verification and Validation
  6. Future Directions in Computing
- **May 10 - FINAL EXAM 7:30 - 10:15pm**