CPEG467/667 - Fall 2020: Computational & Data Intensive Research Platforms & Applications – *Tentative Syllabus*

See https://sites.udel.edu/canvas/ for additional course information, including online access.
Information on Canvas is confidential. Do no share with anyone not participating in class.

Title: Computational & Data-Intensive Research Platforms &Applications

Time/Venue:

Tue/Thu 3:30pm - 4:45pm – course will be held online
First class: Sept 1, 2020
Last class: Dec 10, 2020
No classes: Nov 3 (Election Day); Nov 24, 26 (Thanksgiving Break)

Important Dates:

Project proposal due: Friday 8/11, 11:59PM (end of second week)
Final project report due: Friday, 12/11, 11:59PM (last week of classes – no extensions)
Midterm Exam: individual oral exam, Week of Oct 12
Final Exam: individual oral exam, Week of Dec 14

Instructor:

Rudolf Eigenmann eigenman@udel.edu
Guest lecturers from several disciplines involved in computational and data science

Communication and Office Hours:

We will reserve some time at the beginning of each lecture for questions and answers.
You can also send me email for a zoom appointment.
I will try to respond to emails within 24hrs. If I haven't, be sure to let me know in the next class.

Course Description:

An introductory course for students conducting computational and data-intensive research in all disciplines, providing an overview of relevant computer systems hardware, software, and applications. The course covers:

- Basics of hardware and system software (CPU and multicore, memory hierarchy, external storage, co-processors, wide-area networking, architectural models including cloud, operating systems, job schedulers, compilers and interpreters, performance evaluation environments)
- Application development and data management (programming languages, organizing programs and data, code management, parallel programming models, program

optimization, performance and scaling, checkpointing, algorithms, metadata, data formats, curation and archiving)

- Local and national resources (compute and storage resources, applying for resource use, getting help)
- Use of computational and data-intensive research methods in domain sciences

Prerequisites*:*

This course is available to members of computational and data-intensive (CDI) research teams of all sciences. There are no formal prerequisites. Attendees should be involved in CDI research, have experience in developing application code, and be proficient in a programming languages. If you are strongly interested in taking this course but do not have such background, contact the instructor.

Course format:

The course will provide high-level overviews of the topics**.** The emphasis in class and projects will be on applying the course material to the codes and data used in the students' research efforts. Where available, students will be asked to familiarize themselves with the material through online course content in advance. Some topics will be presented by domain experts.

Course text:

The course will use background material from diverse sources, including video and other online lectures, textbooks, and papers. There are no required books.
Recommended books:

- Introduction to High Performance Computing for Scientists and Engineers; Georg Hager; CRC Press; July 2010
- Introduction to parallel computing; Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar; Pearson, January 2003
- Programming Massively Parallel Processors: A Hands-on Approach; David B. Kirk and Wen-mei W. Hwu; Morgan Kaufmann; February 2010
- The Art of Multiprocessor Programming; Maurice Herlihy, Nir Shavit; MorganKaufmann; June 2012
- Designing Data-Intensive Applications. The Big Ideas Behind Reliable, Scalable, and Maintainable Systems; Martin Kleppmann; O'Reilly Media, March 2017
- MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop; Adam Shook and Donald Miner; O'Reilly Media, December 2012
- Data Analysis in the Cloud, Models, Techniques and Applications; Domenico Talia, Paolo Trunfio, Fabrizio Marozzo, September 2015
- Cloud Computing for Science and Engineering; Ian Foster, Dennis Gannon; The MIT Press Cambridge, 2017

Grading:

• Midterm (Beginning Oct 14):	20%
• Final exam (Week of Dec 14):	30%
• Projects:	25%
• Class Participation:	25%

Approximate schedule:

Week

1. Introduction, Languages and code generation
2. UNIX Shell, GitHub, version control
3. Shared memory architecture and programming
4. Distributed memory architecture and programming
5. Architecture as seen by software: GPUs and other co-processors
6. Architecture as seen by software: Clouds
7. *Midterm exams*
8. Local, national resources and access
9. Optimizations and algorithms
10. Debugging, testing, performance analysis
11. Computational and data-intensive domain sciences
12. Computational and data-intensive domain sciences
13. Computational and data-intensive domain sciences
14. Project presentations
15. Project presentations
16. Final exam

Exams:

The midterm and final exams will be individual, oral exams of 20 minutes, testing the understanding of the material learned up to that point in the courses. The exams will take the specific context of the student (discipline and skill level at the beginning of the course) into consideration. Contact the instructor to request an exam slot of 20 minutes, at least a week in advance of the midterm and final exam weeks.

Projects:

Every student will propose a computational or data-intensive application that you will execute on a parallel machine of your choice. Improve the application in some way that enhances scalability and measures the resulting performance. Describe your experience in 4-6 pages.

The objective of this project is that you gain experience in

- learning about a model and language for computational and/or data science
- working with a programming and execution environment for computational and data-intensive science.

- identifying a computational and storage system on which to run and measure the program,
- identifying code sections in a program that can be improved, and
- analyzing the program performance.

For obtaining a good grade, what counts is not so much that you can speed up your program, but that you demonstrate in your report an understanding of these five points. The expectation is that you spend two to three hours per week on your project.

Format and submission: At the end of the second week of classes, submit a project proposal of ½ to 1 page. The proposal should begin with the course number, a project title, and your name. Email the proposal to the instructor with subject line CPEG 467/667 Project Proposal.

By the end of the semester, expand this text into a 4-6 page report and submit by the project deadline (no extensions) with subject line CPEG 467/667 Final Project Report.

Policies:

Policy regarding class absences and late homework/projects: If you miss a class, make sure you ask another student in class to inform you of relevant information that may not be captured on the videos. If you have a conflict with an exam date that you know at the beginning of the semester, discuss alternative ways of taking the exam with the instructor within the first two weeks of classes. Conflicts that come up later, you will need to resolve with the party that created them. Exceptions to this policy require students to work with Assistant Dean Shermeyer to submit proof of extenuating circumstances so that the Dean's office emails the instructor an official Excused Absence Notification (EAN). If you receive an EAN, you need to then contact the instructor to discuss alternatives to taking the exam.

Be sure you complete your projects early, allowing for possible emergencies (such as illness or computer failed just before project submission).

UD has a number of policies to make learning and campus life as smooth and enjoyable as possible for everyone. Please familiarize yourself with these policies. http://www1.udel.edu/stuguide/

For questions related to COVID-19 please see https://www.udel.edu/home/coronavirus/return/