Computing Foundations for Computational Science

Harvard CS205
Spring 2015
Instructor: Professor HT Kung
Mondays and Wednesdays from 2:30-4pm, Pierce Hall 209

Computational science grows in popularity each day. Via parallel and distributed computing, advances in the area have enabled practitioners in diverse fields, including physical sciences, biotechnology, medicine, finance, and engineering, to discover and recognize principles and patterns in data. To understand, harness, and further these powerful capabilities, students must grasp both relevant computer science foundations and programming skills. To this end, this course consists of foundational modules and programming tasks essential to the theory and practice of data science:

Foundational modules

- Data-driven model learning
- Parallel computing, GPU
- Distributed computing, MapReduce
- Graphlab
- Neuromorphic computing for asynchronous data flows
- IO complexity and management
- Compressive sensing and dimension reduction
- Optimistic concurrency control
- Consensus protocols
- Distributed machine learning

Programming tasks

- MapReduce
- OpenCL
- MPI
- Graphlab
- Distributed learning

The course instruction has two components: lectures and labs. Lectures will focus on teaching the foundational modules based on research literature. The labs will provide assistance on the programming tasks, and will use server clusters at Harvard as well as remote resources in the cloud. In addition, labs will have access to state-of-the-art 3D cameras for data acquisition. Students will learn to use open source tools and libraries and apply them to data analysis, modeling, and the visualization of machine learning and scientific computing problems.

Students will complete weekly quizzes on assigned reading materials, practice skills through programming tasks, and implement a final project (in 3- or 4-person teams) using concepts and skills learned in the course.

Prerequisites: (1) programming experience (Python and CS50 should be fine); (2) basic knowledge in systems programming and machine organization (e.g., CS61); (3) familiarity in algorithms (e.g., CS124); and (4) maturity in mathematics (e.g., undergraduate linear algebra and statistics). For students with strong interest in the subject matter, one or two of these four requirements may be waived.