Outline

• What?
• Why?
• Who?
• How?
Outline

• What?
• Why?
• Who?
• How?
Computational Science

The use of advanced computing capabilities to solve complex scientific problems.

Images: SCI Institute, Utah
Modeling & Simulation
0.975 billion years
Unified Particle Physics for Real-Time Applications

Miles Macklin  Matthias Müller  Nuttapong Chentanez  Tae-Yong Kim

NVIDIA
Big Data & Data Science
How much Data?

- Google: 10 Exabytes of storage (2013)
- Twitter: 300+ Petabytes (2013)
- Facebook: 300+ PB (2013)
- eBay: 150 PB (2014)
- NOAA: 17 PB of climate data (2014)
- CERN’s LHC:
  600M events / sec, 1 MB per event.
  Filtered down to ~ 5 GB / sec, 160 PB / year (2015)
An example from my own Research
The Scientific Challenge

How is the mammalian brain wired?

Ramón y Cajal, 1905

Harvard Center for Brain Science
Connectome Workflow
The Data Challenge

- Pixel resolution: 4 nm
- Slice thickness: 30 nm
- 1 mm$^3$: 2 Petabytes

Images: Josh Morgan
1,500 μm³ hand traced volume.
1 / 600,000 of 1 mm³
6 months x 2 postdocs

Kasthuri, et al. 2015
Outline

- What?
- Why?
- Who?
- How?
Outline

- What?
- Why (do we care about Parallel Algorithms)?
- Who?
- How?
Why Parallelism?
Why Parallelism?

Transistors
Clock Speed
Watts
Operations / Clock

Intel CPU Trends
(sources: Intel, Wikipedia, K. Olukotun)

Herb Sutter
What Does the End of Moore's Law Mean for Gaming?

SHAMUS YOUNG | 31 AUGUST 2015 12:00 PM

This is a very strange time for computer technology. I mean, it’s always a strange time for one reason or another.
Multi- & Many-Core Processors

Intel Core i7-980X Extreme
6 cores
1.17B transistors

NVIDIA GTX 580 SC
512 cores
3B transistors

http://en.wikipedia.org/wiki/Transistor_count
Challenges of Parallelism

Parallel architectures and systems require:

• ... rethinking software design
• ... rethinking algorithms
• ... learning new skills
• ... learning new strategies
• ... learning new tools
Mozilla offices,
2015
(David Baron)
Two types of programmers…

• Two kinds of scientific programmers

  • Those using single processors

  • Those who can use up to 100 processors

• Big steps for programmers

  • From 1 to 2 processors

  • From 100 to 1000s of processors

After CS205
Outline

• What?
• Why?
• Who?
• How?
Ray Jones  
thouis@seas.harvard.edu

- BA in CS @ University of Utah
- PhD @ MIT
  - Image analysis for large-scale cell-based experiments (CellProfiler project)
- 10+ years industry experience.
- At Harvard since 2012
  - High-throughput (multi-TB / day) image analysis for Connectomics
TFs

- Philippe Tillet (Head TF)
- Adi Peleg
- Kevin Chen
- Zezhou (Alex) Liu
- Raahil Sha
Outline

- What?
- Why?
- Who?
- How?
CS 205 Goals

• Learn parallel computational thinking and tools.

• Implement data-intensive computations on multicore, cloud, and GPU systems.

• Practice modern software engineering.
  • Open, Collaborative, & Dynamic.

• Apply these techniques to a problem of interest to you (Final Project).
Course Topics

Modern Architectures
- Multithreaded / Multiprocessor

Cloud Computing
- Amazon’s EC2 & Spark

High-Performance Parallel Computing
- GPU computing with OpenCL

Applications in Scientific Computing
- Modeling, Simulation, and Data Science
Act I: Cloud Computing

- Overview of parallel computing
- “Embarrassingly Parallel” Computing
- Spark & Amazon’s EC2
- Advanced Spark
Act II: Thinking Parallel

- Analysis of parallel algorithms
- Multicore / Multiprocessor
- Abstraction and parallel primitives
- Understanding how CPU & memory architecture affect computing
Act III: Heterogenous and GPU Computing

- Alternative computing environments
- Parallel / GPU programming with OpenCL
- Abstraction and modularity
- Performance analysis and optimization
- Advanced parallel strategies
Act IV: Applications

- Image & video processing
- Text mining
- Machine learning
- Numerical algorithms
Prerequisites

Programming experience in a higher-level language

• Python, C, C++, Fortran, Java, etc.

Willingness to learn new software

• This can be time consuming.
• We will be on the bleeding edge!
• We’ll try to make it fun!
YOU'RE FLYING!
HOW?

PYTHON!
Software Engineering

Open & Collaborative:

- GitHub - submit HW via Pull Request.
- Peer evaluations via Code reviews (on the PR).
- Collaboration is encouraged!
  - Work together, look at each other’s code (with permission), etc.
- BUT!!! do your own work!

Simple rule: no cut & paste.
CS205: Computing Foundations for Computational Science

Computation has long been an important tool for scientists, but the past two decades have seen a true revolution in the practice of science. Computation, in the form of both simulation and analysis, has joined theory and experimentation as the oft-quoted “third pillar” of science.

This is an applications course highlighting the use of modern computer architectures in solving scientific problems. The class emphasizes making effective use of modern architectures, particularly parallel, multicore, and GPU-based processing, with a strong emphasis on “parallel thinking.” You will learn the fundamentals of scientific computing including abstract thinking, algorithmic development, and assessment of computational approaches, using a series of open source tools and libraries for data analysis, modeling, and visualization of real scientific problems.

Instructor:
Thouis “Ray” Jones

Staff:
Philippe Tillet
Adi Peleg
Next Steps

- HW 0 - out tomorrow
  - Due dates: next Tuesday and Friday (Sept 8 & 11)
  - Get signed up for GitHub, Amazon, & Piazza.
- Read syllabus carefully
- Do readings (posted on Piazza)
  - Post comments to Piazza before lectures
- IACS Seminar on Fridays, 1:00 pm, G115
  - (Free lunch @ 12:30)