CS 595:
Data-Intensive Computing

Syllabus

Ioan Raicu
Computer Science Department
Illinois Institute of Technology

CS 595: Data-Intensive Computing
August 22nd, 2011
Semester: Fall 2011
Lecture Time: Monday/Wednesday, 11:25AM - 12:40PM
Location: Stuart Building 106
Professor: Dr. Ioan Raicu (iraicu@cs.iit.edu, 1-312-567-5704)
  - Office Hours Time: Wednesday, 12:45PM - 1:45PM
  - Office Hours Location: Stuart Building 237D
Teaching Assistant: TBA
  - Office Hours Time: TBA
  - Office Hours Location: TBA
URL: http://www.cs.iit.edu/~iraicu/teaching/CS595-F11/
Who am I?

• **Current position:**
  – Assistant Professor at Illinois Institute of Technology (CS)
    • Director of the Data-Intensive Distributed Systems Laboratory (DataSys)
  – Guest Research Faculty, Argonne National Laboratory (MCS)

• **Education:** PhD, University of Chicago, March 2009

• **Funding/Awards:**
  – NSF CAREER, 2011 – 2015 ($450K)
  – NSF/CRA CIFellows, 2009 – 2010 ($140K)
  – NASA GSRP, 2006 – 2009 ($84K)

• **Over 70+ Collaborators:**
  – Ian Foster (UC/ANL), Rick Stevens (UC/ANL), Rob Ross (ANL), Marc Snir (UIUC), Arthur Barney Maccabe (ORNL), Alex Szalay (JHU), Pete Beckman (ANL), Kamil Iskra (ANL), Mike Wilde (UC/ANL), Douglas Thain (ND), Yong Zhao (UEST), Matei Ripeanu (UBC), Alok Choudhary (NU), Tevfik Kosar (SUNY), Yogesh Simhan (USC), Ewa Deelman (USC), and many more…

• **More info:** [http://www.cs.iit.edu/~iraicu/index.html](http://www.cs.iit.edu/~iraicu/index.html)
Who are you?

• Background?
  – Math/CS/ECE?
  – UG/MS/PhD?

• What do you want to get out of this course?
Data Intensive Computing is critical to advancing modern science
  - Applies to cluster computing, grid computing, supercomputing, and cloud computing

Increasing gap between compute capacity and storage bandwidth

Need for advanced techniques to manipulate, visualize and interpret large datasets

Building large-scale distributed systems is hard
  - network (e.g., transport, routing)
  - algorithmic (e.g., data distribution, resource management)
  - social (e.g., incentives)
• Understand methods and approaches to:
  – Design, implement, and evaluate distributed systems
• Topics include:
  – Resource management (e.g. discovery, allocation, compute models, data models, data locality, virtualization, monitoring, provenance), programming models, application models, and system characterization
• Course involves:
  – Lectures, outside invited speakers, discussions of research papers, homeworks, and a major project
Prerequisites

• Topics
  – Programming (C, C++, or Java)
  – Networking
  – Operating systems
  – Distributed systems
Course Topics

- Distributed Systems
- Supercomputing
- Grid Computing
- Cloud Computing
- Many-core Computing
- Data Intensive Computing
- Storage Systems
- Distributed and Parallel File Systems
Course Topics (cont)

- Parallel I/O
- Local Resource Management
- Scientific Computing and Applications
- Parallel Programming Systems and Models
- MapReduce
- Data-Intensive Computing with GPUs
- Data-Intensive Computing with Databases
Computer Usage

• fusion.cs.iit.edu
  – request account by sending email to iraicu@cs.iit.edu
  – AMD, 48-cores @ 1.9GHz, 64GB RAM, 1Gb/s network, Linux Suse 11.2 x64

• csrocks.cs.iit.edu
  – accounts have already been requested, you will be notified of instructions on how to access the CSROCKS cluster
  – 15 nodes, 1Gb/s network, Linux
• IIT/CS SCS Linux Cluster (512-cores x64)
• IBM BlueGene/P at Argonne National Laboratory (160K PPC)
• SiCortex at Argonne National Laboratory (5832 MIPS)
• Amazon EC2
• Windows Azure
• 1~2 papers per lecture
• Each paper must be summarized in writing
• Serve as background to the lecture
• Serve as basis for discussion
  – Each paper will have a student discussion leader
Homeworks

• 1~5 assignments
• Will give hand-on experience with some specific technology or theoretical concept
• Generally will have 1~3 week(s) to complete
• Must be completed individually
Projects

• Major quarter long project
  – Topic of choice of the student
  – Can work in groups
  – May require the following things:
    • Reading research papers
    • Using open source software
    • Implementation of a real/simulated system
    • Analysis of theoretical work
    • Performance evaluation of theoretical/real systems
    • Written report(s)
    • Oral presentation(s)
Project Ideas

• Distributed file systems
• Data aware scheduling algorithms
• Distributed operating systems
• Distributed job management systems
• Parallel programming languages
• Distributed workflow systems
• Distributed monitoring systems
• Scientific computing with GPUs
• Scientific computing with MapReduce
• Distributed caching strategies
• Distributed cache eviction policies
• Distributed hash tables
• Virtualization impact for data-intensive computing
Useful Software for your Projects

- **Operating systems:** Linux, Windows
- **Scripting:** BASH
- **Source control:** SVN
- **Programming languages:** Java, C/C++
- **Job submission systems:** GRAM, PBS, Condor, Cobalt, SGE, Falkon
- **Programming models:** MapReduce (Hadoop), MPI (MPICH), Multi-.Threading (PThreads), Workflows (Swift, Pegasus/DAGMan, Nimrod, Taverna, BPEL)
- **File systems:** FUSE
• **Parallel file systems:** GPFS, PVFS, Lustre
• **Distributed file systems:** GPS, HDFS
• **Data services:** GridFTP
• **Grid middleware:** Globus
• **Cloud middleware:** Nimbus, Eucalyptus, OpenNebula
• **Distributed hash tables:** Chord, Tapestry
• **Simulation environments:** GridSim, SimGrid, OptorSim, GangSim, Bricks
• **Virtualization:** Sun Virtual Box, XEN, VMWare
• Participation in paper discussions (including writeups for papers): 15%
• Homeworks: 15%
• Project Proposal: 5%
• Mid-quarter oral presentation: 10%
• Final oral presentation: 15%
• Final Project Report: 40%
Course Outcomes

- Understand the importance of data-intensive computing
- Understand the difference between cluster computing, grid computing, supercomputing, and cloud computing
- Understand how to build large scale distributed systems
- Understand applications that require data-intensive computing
- Understand trends in many-core computing and challenges that will come with them
- Build distributed systems
- Be familiar with multiple programming models
- Read and understand a research paper
- Make a formal presentation on a technical topic
- Write up a formal report (and a research paper) on the project
• Required texts
  – None
  – Readings will be from online material
• We will be using BlackBoard minimally, mostly to post grades
• Mailing list
  – Sending email to cs595-f11@datasys.cs.iit.edu
  – More info at:
    • http://datasys.cs.iit.edu/mailman/listinfo/cs595-f11
• Write me:
  – iraicu@cs.iit.edu
• Skype me:
  – ioan.raicu
• Call me:
  – 1-312-567-5704
• Mailing list
  – cs595-f11@datasys.cs.iit.edu