• **Semester:** Spring 2015
• **Lecture:** Monday/Wednesday, 11:25AM - 12:40PM
• **Location:** Stuart Building 113
• **Professor:** Dr. Ioan Raicu ([iraicu@cs.iit.edu](mailto:iraicu@cs.iit.edu))
  – **Office Hours Time:** Wednesday, 12:45PM - 1:45PM, SB237D
• **Teaching Assistant:** Ke Wang
  – **Office Hours Time:** Monday 10:15AM – 11:15AM, SB002
• **Teaching Assistant:** Tonglin Li
  – **Office Hours Time:** Tuesday 12:45PM – 1:45PM, SB002
• **Teaching Assistant:** Dongfang Zhao
  – **Office Hours Time:** Thursday 12:45PM – 1:45PM, SB002
• **URL:** [http://www.cs.iit.edu/~iraicu/teaching/CS554-S15/](http://www.cs.iit.edu/~iraicu/teaching/CS554-S15/)
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, 2009
• Postdoc: Northwestern University
• Funding/Awards:
  – NSF, DOE, NASA (~$1.3M)
• Over 70+ Collaborators:
  – DOE Labs: ANL, ORNL, LANL, PNNL, LBL, FNAL
  – Academia: UIUC, UChicago, Northwestern, John Hopkins, UCBerkeley, Notre Dame
  – Industry: Amazon, Microsoft, Google, Cleversafe
• My students work at:
  – Microsoft, IIT, NetApp, Hortonworks, Dell, Amazon, Nokia, Here, ANL, etc
• More info: http://www.cs.iit.edu/~iraicu/index.html
• Research Focus
  – Emphasize designing, implementing, and evaluating systems, protocols, and middleware with the goal of supporting data-intensive applications on extreme scale distributed systems, from many-core systems, clusters, grids, clouds, and supercomputers

• People
  – Dr. Ioan Raicu (Director)
  – 6 PhD Students (spanning 2nd to 6th year)
  – 7 MS Students
  – 5 UG Students

• Contact
  – iraicu@cs.iit.edu
Who are you?

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

• What do you want to get out of this course?
Course Overview

- 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, homework, and a major project
Prerequisites

• Coursework
  – Required: CS450
  – Recommended: CS542, CS546, CS451, CS550, CS551, CS552, CS553, and CS570

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

- Paradigms
- Parallel Programming Systems
- Job Management Systems
- Storage Systems
Course Topics

• Paradigms
  – Supercomputing (e.g. IBM BlueGene/P/Q, Cray XT6)
  – Grid Computing (e.g. XSEDE, OSG)
  – Cloud Computing (e.g. Amazon AWS, Google App Engine, Windows Azure)
  – Many-core Computing (e.g. NVIDIA GPUs, Xeon Phi)
Course Topics

- Parallel Programming Systems
  - MapReduce (e.g. Hadoop)
  - Workflows (e.g. Swift)
  - MPI (e.g. MPICH)
  - OpenMP
  - Multi-Threading (e.g. PThreads)
• Job Management Systems
  – Batch scheduling (e.g. Condor, Slurm, SGE, PBS)
  – Light-weight Task Scheduling (e.g. Falkon, Sparrow, MATRIX)
Course Topics

• Storage Systems
  – File Systems (e.g. EXT3)
  – Shared File Systems (e.g. NFS)
  – Distributed File Systems (e.g. HDFS, FusionFS)
  – Parallel File Systems (e.g. GPFS, PVFS, Lustre)
  – Distributed NoSQL Key/Value Stores (e.g. Cassandra, MongoDB, ZHT)
  – Relational Databases (e.g. MySQL)
Computer systems that can be used for development of projects (more information about access to these will be passed in the first several lectures):
  – 20-node Linux cluster
  – Amazon AWS - $100 credit per student

Other systems that could be used, on as needed basis:
  – IIT/CS SCS Linux Cluster (512-cores x64)
Research Papers
Reading and Discussion

• 1~2 papers per lecture
• Serve as background to the lecture
• Serve as basis for discussion
• Quizzes will be from the reading assignments
• Students will be assigned to lead the discussion
• Major quarter long project
  – Topic of choice of the student (from a given list)
  – Can work in groups of up to 3 students
  – 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
Useful Software for your Projects

- Operating systems: Linux
- Scripting: BASH
- Source control: SVN
- Programming languages: Java, C/C++
- Job submission systems: GRAM, PBS, Condor, Cobalt, SGE, Falkon, Sparrow
- Programming models: MapReduce (Hadoop/Spark), MPI (MPICH), Multi-Threading (PThreads), Workflows (Swift)
- File systems: FUSE
- Parallel file systems: GPFS, PVFS, Lustre
Useful Software for your Projects (cont)

- Distributed file systems: GPS, HDFS, FusionFS, Ceph, GlusterFS
- Data services: GridFTP
- Grid middleware: Globus
- Cloud middleware: Nimbus, Eucalyptus, OpenNebula, Open Stack
- Key/Value Stores: Chord, Tapestry, ZHT, Casandra, MongoDB, MemCached, DynamoDB
- Simulation environments: GridSim, SimGrid, OptorSim, GangSim, Bricks, SimMatrix, PeerSim, CODES/ROSS
- Virtualization: Oracle Virtual Box, XEN, VMWare
• Quizzes: 20%
• Project Proposal: 10%
• Mid-Semester Progress Report: 10%
• Final Oral Presentation: 25%
• Final Project Report: 25%
• Participation: 10%
Grade Scale

- A: 85% ~ 100%
- B: 70% ~ 89%
- C: 60% ~ 69%
- E: 0% ~ 59%
Late Policy

• Assignments will be due at 11:59PM on the day of the due date, through BlackBoard
• There will be a 15 minute grace period
• There will also be a 7-day late pass, where students can submit late assignments without penalty
  – the late pass can be used in 1-day increments spread out over multiple assignments
• Any late submissions beyond the grace period and beyond the 7-day late pass, will be penalized 10% every day it is late
Course Outcomes

- Understand the importance of data-intensive computing
- Understand the difference between cluster, grid, clouds, and supercomputing.
- 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 systems research papers
- Make a formal presentation on a technical topic
- Write up a formal report on the project
• Required texts
  – None
  – Readings will be from online material
• We will be using BlackBoard minimally, mostly to post grades
• Mailing list
Questions

• Write me:
  – iraicu@cs.iit.edu

• Skype me:
  – ioan.raicu

• Call me:
  – 1-312-567-5704

• Reach all TAs:
  – cs554-s15@datasys.cs.iit.edu