Syllabus

CS 554

Data-Intensive Computing

http://www.cs.iit.edu/~iraicu/teaching/CS554-F13/index.html

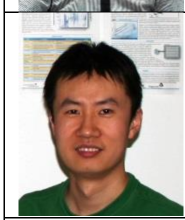
Semester: Fall 2013
Lecture Time: Monday/Wednesday, 11:25AM - 12:40PM
Location: Stuart Building 238

Professor:

- **Dr. Ioan Raicu** (iraicu@cs.iit.edu, 1-312-567-5704)
  - Office Hours Time: Wednesday 12:45PM-1:45PM (SB237D)

Teaching Assistant:

- **Ke Wang** (kwang22@hawk.iit.edu)
  - *Area of Research:* Distributed Scheduling Systems and Simulations
  - *Office Hours Time:* Monday/Tuesday 12:45PM-1:45PM (SB002)



Teaching Assistant:

- **Tonglin Li** (tli13@iit.edu)
  - *Area of Research:* Distributed NoSQL Key/Value Stores
  - *Office Hours Time:* Thursday 10AM-11AM, Friday 12:45PM-1:45PM (SB002)

Other PhD Students (and general area of research) who can help with course projects:

- **Dongfang Zhao (**dzhao8@hawk.iit.edu**) –** Distributed File Systems
- **Scott Krieder (**skrieder@iit.edu**) –** Many-Core Computing
- **Iman Sadooghi (**isadoogh@iit.edu**) –** Cloud Computing

Office Hours Summary:

- **Monday:** 12:45PM-1:45PM (Wang SB002)
- **Tuesday:** 12:45PM-1:45PM (Wang SB002)
- **Wednesday:** 12:45PM-1:45PM (Raicu SB237D)
- **Thursday:** 10AM-11AM (Li SB002)
- **Friday:** 12:45PM-1:45PM (Li SB002)

Overview

This course is a tour through various research topics in distributed data-intensive computing, covering topics in cluster computing, grid computing, supercomputing, and cloud computing. We will explore solutions and learn design principles for building large network-based computational systems to support data intensive computing. This course is geared for junior/senior level undergraduates and graduate students in computer science.

Course Goals

The support for Data Intensive Computing is critical to advancing modern science as storage systems have experienced an increasing gap between its capacity and its bandwidth by more than 10-fold over the last decade. There is an emerging need for advanced techniques to manipulate, visualize and interpret large datasets. Building large scale distributed systems that support data-intensive computing involves challenges at multiple levels, from the network (e.g., transport, routing) to the algorithmic (e.g., data distribution, resource management) and even the social (e.g., incentives). This course is a tour through various research topics in distributed systems, covering topics in cluster computing, grid computing, supercomputing, and cloud computing. We will explore solutions and learn design principles for building large network-based computational systems to support data intensive computing. Our readings and discussions will help us identify research problems and understand methods and general approaches to design, implement, and evaluate distributed systems to support data intensive computing. Topics include resource management (e.g. discovery, allocation, compute models, data models, data locality, virtualization, monitoring, provenance), programming models, application models, and system characterization. Our discussions will often be grounded in the context of deployed distributed systems, such as the TeraGrid, Amazon EC2 and S3, various top supercomputers (e.g. IBM BlueGene/P, Sun Constellation, Cray XT5), various cloud platforms (e.g. Amazon AWS, Google App Engine, Windows Azure, and private clouds), and various software/programming platforms (e.g. Hadoop/MapReduce, Dryad, Sphere/Sector, Swift/Falkon, and Parrot/Chirp). The course involves lectures, outside invited speakers, discussions of research papers, and a major project (including written reports and oral presentations). Prerequisite: CS450.

Required Texts

None; readings will be from published research online material.

Prerequisites

CS450 (Operating Systems). Other courses that might contribute to having a better in depth understanding of this course are CS542, CS546, CS451, CS550, CS551, CS552, CS553, and CS570; these courses are not required, and the introductory content of some of those courses will be repeated in this class. Many of these graduate courses are part of the Master of Computer Science Specialization in Distributed and Cloud Computing.

Knowledge of the following topics is recommended to get the most out of this class:

1. Programming (C, C++, or Java)
2. Networking
3. Operating systems
4. Architecture
5. Distributed systems

Computer Usage

Computer systems that can be used for development of projects (more information about access to these will be passed in the first several lectures):

- **15-node (150-core) private cloud using virtualization and Linux**
- **Amazon AWS - $100 credit per student**

Other systems that could be used, on as needed basis:

- IIT/CS SCS Linux Cluster (512-cores x64)
- IIT/CS CUDA Linux Cluster (80-cores x64 with NVIDIA GPUs)
- IBM BlueGene/P at Argonne National Laboratory (160K-cores PPC)
- SiCortex at Argonne National Laboratory (5832-cores MIPS)

Detailed Course Topics

Lecture 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)
- 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)
- 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. Casandra, MongoDB, ZHT)
  - Relational Databases (e.g. MySQL)

Projects

There will be a major semester long project that will require the implementation of a real/simulated system, a written report, and an oral presentation. Projects can fall in a number of different areas, which are related to data-intensive distributed computing. These areas will be expanded with specific projects in each area.

- 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

The students will be given 10~20 project descriptions (<1 page in length for each project), and students must pick 1 project to develop further into a proposal (2~3 pages). Students can work in teams of 2 on these projects, and multiple teams can work on the same project (independently). These projects will generally be of high interest to the DataSys Laboratory and are likely sub-projects from existing ongoing research projects. These projects will have constant interaction with PhD students in the DataSys lab who are leading the projects at large. Students who obtain excellent results on their projects will be encouraged to submit their work as a publication to highly respected conferences and workshops. Two such events that will be located in Chicago in May 2014 are IEEE CCGrid 2014 and GCASR 2014.

Some major projects and tools you should familiarize yourself with are (as they might be useful to your projects):

- Operating systems: Linux
- 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)
- File systems: FUSE
- Parallel file systems: GPFS, PVFS, Lustre
- 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
- Simulation environments: GridSim, SimGrid, OptorSim, GangSim, Bricks, SimMatrix, PeerSim
- Virtualization: Oracle Virtual Box, XEN, VMWare

Schedule

Tentative schedule:

- 08/19/2013: Syllabus
- 08/21/2013: Introduction to Distributed Systems
- 08/26/2013: Introduction to Distributed Systems
- 08/28/2013: Project Brainstorming
- 08/30/2013: **Group formation due;** Last day to add/drop class
- 09/02/2013: NO CLASS (Labor Day)
- 09/04/2013: TBA
- 09/09/2013: **Project Proposal Due**
- 09/11/2013: Guest Lecture
- 09/16/2013: TBA
- 09/18/2013: TBA
- 09/23/2013: TBA
- 09/25/2013: TBA
- 09/30/2013: TBA
- 10/02/2013: TBA
- 10/07/2013: TBA
- 10/09/2013: TBA
- 10/14/2013: TBA
- 10/16/2013: **Project Midterm Progress Report Due**
- 10/21/2013: TBA
- 10/23/2013: TBA
- 10/28/2013: TBA
- 10/30/2013: TBA
- 11/04/2013: TBA
- 11/06/2013: TBA
- 11/11/2013: TBA
- 11/13/2013: TBA
- 11/18/2013: Guest Lecture
- 11/20/2013: Guest Lecture

- 11/25/2013: TBA
- 11/27/2013: NO CLASS (Thanksgiving)
- 12/02/2013: **Project Final Presentations** (likely 10AM – 8PM with some breaks)
- 12/04/2013: **Project Final Reports Due**
- 12/11/2013: Grades Due

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.

Mailing lists

This course will use Piazza to facilitate discussions around the projects, at http://piazza.com/iit/fall2013/cs554/home. The previous announced mailing list has been removed.

Grades

There are no exams. Grading Policies:

- Homework: 20%
- Project Proposal: 10%
- Mid-semester Progress Report: 10%
- Final Oral Presentation: 30%
- Final Project Report: 30%

The following grading scale will be used. The scale will be adjusted downwards based on the overall performance of the entire class. Traditionally, in my classes, the class average score will typically fall in the B-grade range.

- **A: 85% ~ 100%**
- **B: 70% ~ 84%**
- **C: 60% ~ 69%**
- **E: 0% ~ 59%**

Course Outcomes

When a student completes this course, s/he should be able to:

1. Understand the importance of data-intensive computing
2. Understand the difference between cluster, grid, clouds, and supercomputing.
3. Understand how to build large scale distributed systems
4. Understand applications that require data-intensive computing
5. Understand trends in many-core computing and challenges that will come with them
6. Build distributed systems
7. Be familiar with multiple programming models
8. Read and understand systems research papers
9. Make a formal presentation on a technical topic
10. Write up a formal report (or even a research paper) on the project