Syllabus

CS 553: Cloud Computing

http://www.cs.iit.edu/~iraicu/teaching/CS553-F17/

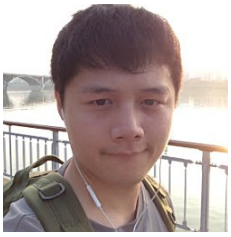
Ioan Raicu

Semester: Fall 2017
Lecture Time: Monday/Wednesday, 11:25AM - 12:40PM
Location: Stuart Building 104

Professor:

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

Teaching Assistants (cs553-s16@datasys.cs.iit.edu**):**



Jian Peng

- **Jian Peng** (jpeng10@hawk.iit.edu)
    - Office Hours Time: Monday/Wednesdays 10:15AM-11:15AM (SB007)
- **George Mathew (**gmathew1@hawk.iit.edu)
    - Office Hours Time: Tuesday/Thursday 12:45PM-1:45PM (SB007)

George Mathew

Summary Office Hours:

- Monday: 10:15AM-11:15AM (Peng/SB007)
- Monday: 12:45PM-1:45AM (Raicu/SB237D)
- Tuesday: 12:45PM-1:45AM (Mathew/SB007)
- Wednesday: 10:15AM-11:15AM (Peng/SB007)
- Wednesday: 12:45PM-1:45AM (Raicu/SB237D)
- Thursday: 12:45PM-1:45AM (Mathew/SB007)

Course Description

Cloud Computing is "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet." It has become a driving force for information technology over the past several years, and it is hinting at a future in which we won't compute on local computers, but on centralized facilities operated by third-party compute and storage utilities. Governments, research institutes, and industry leaders are rushing to adopt Cloud Computing to solve their ever-increasing computing and storage problems arising in the Internet Age. There are three main factors contributing to the surge and interests in Cloud Computing: 1) rapid decrease in hardware cost and increase in computing power and storage capacity, and the advent of multi-core architecture and modern supercomputers consisting of hundreds of thousands of cores; 2) the exponentially growing data size in scientific instrumentation/simulation and Internet publishing and archiving; and 3) the wide-spread adoption of Services Computing and Web 2.0 applications. This course is a tour through various topics and technologies related to Cloud Computing. Topics include distributed system models and enabling technologies, computer clusters for scalable Computing, virtual machines and virtualization of clusters and datacenters, design of cloud computing platforms, cloud programming and software environments (Workflow Systems, MapReduce, Google App Engine, Amazon AWS, Microsoft Azure, and emerging

cloud software stacks), grid computing and resource management, P2P computing with overlay networks, ubiquitous computing with clouds and the Internet of things, and data-intensive distributed computing. The course involves lectures, projects, programming assignments, and exams. Prerequsites: CS450 or CS455.

Required Texts

We will be using the textbook Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet by Kai Hwang, Jack Dongarra & Geoffrey C. Fox. (Required)

Prerequisites

CS450 (OS) or CS455 (Data Communications). Other courses that might contribute to having a better in depth understanding of this course are CS451, CS542, CS546, CS550, CS551, CS552, CS554, CS570, and CS595 (VMs). Many of these graduate courses are part of the Master of Computer Science Specialization in Distributed and Cloud Computing.

Detailed Course Topics

- Distributed System Models
- Parallel Computing
- Cloud Platform Architectures
- Cloud Programming
- Grid Computing
- Supercomputing Computing

Programming Assignments

There will be 3 programming assignments throughout the semester, each worth 15%~20% of the total grade, and each taking about 3~4 weeks to complete. These assignments can be completed in groups up to 2 students. The projects will require knowledge of Java, C and/or C++. It is expected that students know the basics of these languages, but students are free to generally choose the languages they want to implement their assignments in. These assignments must all work in a Linux environment (in which they will be graded in). These assignments must be submitted through BlackBoard; only 1 student (from the group) must submit the assignment.

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):

- **Amazon AWS (**https://aws.amazon.com**):** To be used for Programming Assignment #3 (likely $100 per student)
- **Chameleon** (https://www.chameleoncloud.org): To be used for Programming Assignment #1 and #2 (likely 1000 CPU hours per student)

Project

There will be 1 project which will be worth 10% of the grade. This assignment can be completed in groups up to 2 students, and will require about 4 weeks to be completed. It will involve students to read online material, compute and generate graphs, and write a report. This assignment must be submitted through BlackBoard; only 1 student (from the group) must submit the assignment.

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 4-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 4-day late pass, will be penalized 20% every day it is late. Assignments will not be accepted after the solutions have been posted.

Plagiarism Policy

Cheating will not be tolerated. We will use the MOSS: Measure Of Software Similarity system from Stanford (https://theory.stanford.edu/~aiken/moss/). It is used to automatically determine the similarity of programs (even if they are written in different programming languages). The supported languages are: C, C++, Java, C#, Python, Visual Basic, Javascript, FORTRAN, ML, Haskell, Lisp, Scheme, Pascal, Modula2, Ada, Perl, TCL, Matlab, VHDL, Verilog, Spice, MIPS assembly, a8086 assembly, a8086 assembly, MIPS assembly, HCL2.

You will receive a 0 on assignment; extremely serious offences will fail the course.

Some example screen shots from the MOSS system:

Moss Results

Tue Sep 8 23:29:31 PDT 2015

Options -l python -d -m 10

[How to Read the Results | Tips | FAQ | Contact | Submission Scripts | Credits]

File 1	File 2	Lines Matched
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (99%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/kn[redacted]1/ (99%)	86
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/k[redacted]1/ (76%)	/home/ubuntu/Projects/work/[redacted]/ (66%)	91
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (81%)	/home/ubuntu/Projects/work/[redacted]/ (82%)	69
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (70%)	/home/ubuntu/Projects/work/[redacted]3/ (61%)	70
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]1/ (69%)	/home/ubuntu/Projects/work/[redacted]2/ (40%)	71
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/k[redacted]9/ (56%)	/home/ubuntu/Projects/work/[redacted]4/ (50%)	43
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (62%)	/home/ubuntu/Projects/work/[redacted]9/ (55%)	67
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (55%)	/home/ubuntu/Projects/work/[redacted]/ (48%)	40
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/k[redacted]/ (54%)	/home/ubuntu/Projects/work/[redacted]9/ (55%)	40

/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]/ (68%)		/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]2/ (73%)	
4-71		2-66	
95-111		90-106	
74-91		69-86	
115-132		110-127	

/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]

```
>>>> file: LongJump.py
#[redacted]a

print("***** Long Jump Information System *****")
print("Please enter the names of competitors. (Press return when done.)")
print("Competitor no. 1:")
competitor = input()
b,c,g,h,d,k = 1,0,0,0,[],0
maxi,competitors = [],[competitor]
while True:
    b += 1
    print("Competitor no. "+str(b)+":")
    competitor = input()
    if competitor == "":break
    else:
        competitors.append(competitor)
print("Please enter the distances for each competitor.")
for each in competitors:
    print("Competitor " each ":" sep="")
    at1 = input("Attempt 1:\n")
    at2 = input("Attempt 2:\n")
    at3 = input("Attempt 3:\n")
    x = (at1+at2+at3).lower()
    if (at1+at2+at3).find("oul") != -1:
        x = (at1+at2+at3).lower()
    d.append(at1)
    d.append(at2)
    d.append(at3)
    if x.find("foul") == -1:
        maxi.append(max(eval(at1),eval(at2),eval(at3)))
```

/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]

```
>>>> file: LongJump.py

print("***** Long Jump Information System *****")
print("Please enter the names of competitors. (Press return when done.)")
print("Competitor no. 1:")
competitor = input()
b,c,g,h,d,k = 1,0,0,0,[],0
maximums,competitors = [],[competitor]
while True:
    b += 1
    print("Competitor no. "+str(b)+":")
    competitor = input()
    if competitor == "":break
    else:
        competitors.append(competitor)
print("Please enter the distances for each competitor.")
for each in competitors:
    print("Competitor " each ":" sep="")
    attempt1 = input("Attempt 1:\n")
    attempt2 = input("Attempt 2:\n")
    attempt3 = input("Attempt 3:\n")
    g = (attempt1+attempt2+attempt3).lower()
    if (attempt1+attempt2+attempt3).find("oul") != -1:
        g = (attempt1+attempt2+attempt3).lower()
    d.append(attempt1)
    d.append(attempt2)
    d.append(attempt3)
    if g.find("foul") == -1:
        maximums.append(max(eval(attempt1),eval(attempt2),eval(attempt3)))
    else:
        d.remove("foul")
    if not "foul" in d:
```

Exams

There will be two exams evenly spread out over the semester. Each exam will be worth 20% of the overall grade. If you are an in-class student, you must take the exams in class; for remote students, you can either come take it in class, or you can take the exam at an official testing center in a proctored environment. The exams will be individual, but students will be allowed to use their textbooks and any notes they have (on paper). No electronic

devices such as phones, eReaders, tables, or laptops will be allowed. Simple calculators can be used. The exams will have the following schedule:

- **Exam #1: Wednesday, October 11th, 2017 from 11:25AM – 12:40PM in Stuart Building 104**
- **Exam #1: Wednesday, November 29th, 2017 from 11:25AM – 12:40PM in Stuart Building 104**

THERE WILL BE NO MAKEUP EXAMS.

Grades

Grading Policies:

- **Programming Assignments (3):** 50% -- can use late day passes (PA1=15%, PA2=20%, PA3=20%)
- **Project (1):** 10% -- can use late day passes
- **Exam (2):** 40% -- NO MAKEUPS

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 typically falls around 80% (a solid B-grade). There are two separate (but similar) grading scales, one for undergraduate students, and one for graduate students.

Undergraduate Students:

- **A: 87% ~ 100%**
- **B: 75% ~ 86%**
- **C: 60% ~ 74%**
- **D: 50%~59%**
- **E: 0% ~ 49%**

Graduate Students:

- **A: 87% ~ 100%**
- **B: 75% ~ 86%**
- **C: 60% ~ 74%**
- **E: 0% ~ 59%**

Mailing lists

This course will use Piazza to facilitate discussions for assignments, at http://piazza.com/iit/fall2017/cs553/home (it has not been setup yet, more instructions will follow). Piazza should be the primary mechanism of communication between the students and the professor and the TAs. If you have a question and want to reach only the TAs and professor, send email to cs553-f17@datasys.cs.iit.edu. As a last resort, send individual emails directly (iraicu@cs.iit.edu) when you believe the message is not appropriate to be sent to the entire class, or to all the TAs and professor.