CS 553:

Cloud Computing

Syllabus

Ioan Raicu
Computer Science Department
Illinois Institute of Technology

CS 553: Cloud Computing
August 21st, 2017

Introductions

- **Professor: Ioan Raicu**
  - Office Hours Time: Monday/Wednesday 12:45PM-1:45PM (SB237D)
  - More Information:
    - http://www.cs.iit.edu/~iraicu/
    - http://datasys.cs.iit.edu/
- **TAs (cs553-f17@datasys.cs.iit.edu):**
  - **Jian Peng**
    - **Office hours Tuesday/Thursday 12:45pm-1:45pm**
  - **More TAs to be announced**

Ioan

Course Overview

- This course is a tour through various topics and technologies related to Cloud Computing
- Explore solutions and learn design principles for building large network-based systems, to support compute and data intensive computing across geographically distributed infrastructures
- Discussions often grounded in real Cloud Computing systems:
  - Amazon EC2 and S3, Microsoft Azure, Google AppEngine, OpenStack, Google's MapReduce, Yahoo’s Hadoop, Spark, etc

Course Overview (cont)

- Understand methods and approaches to:
  - Design, implement, and evaluate cloud computing systems
- Course involves:
  - Lectures, outside invited speakers, programming assignments, projects, and exams
- Prerequisites:
  - Required: CS450 (Operating Systems) or CS455 (Data Communication)
  - Recommended: CS550 (Advanced Operating Systems)
  - Helpful: CS451, CS542, CS546, CS550, CS551, CS552, CS554, CS570, and CS595 (VMs)
- Required texts:
  - Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet by Kai Hwang, Jack Dongarra & Geoffrey C. Fox.

Course Topics

- Distributed System Models
- Parallel Computing
- Cloud Platform Architectures
  - Amazon AWS
  - Microsoft Azure
  - Google App Engine
  - Google MapReduce / Yahoo Hadoop
  - Spark
  - OpenStack
- Cloud Programming
- Grid Computing
- Supercomputing

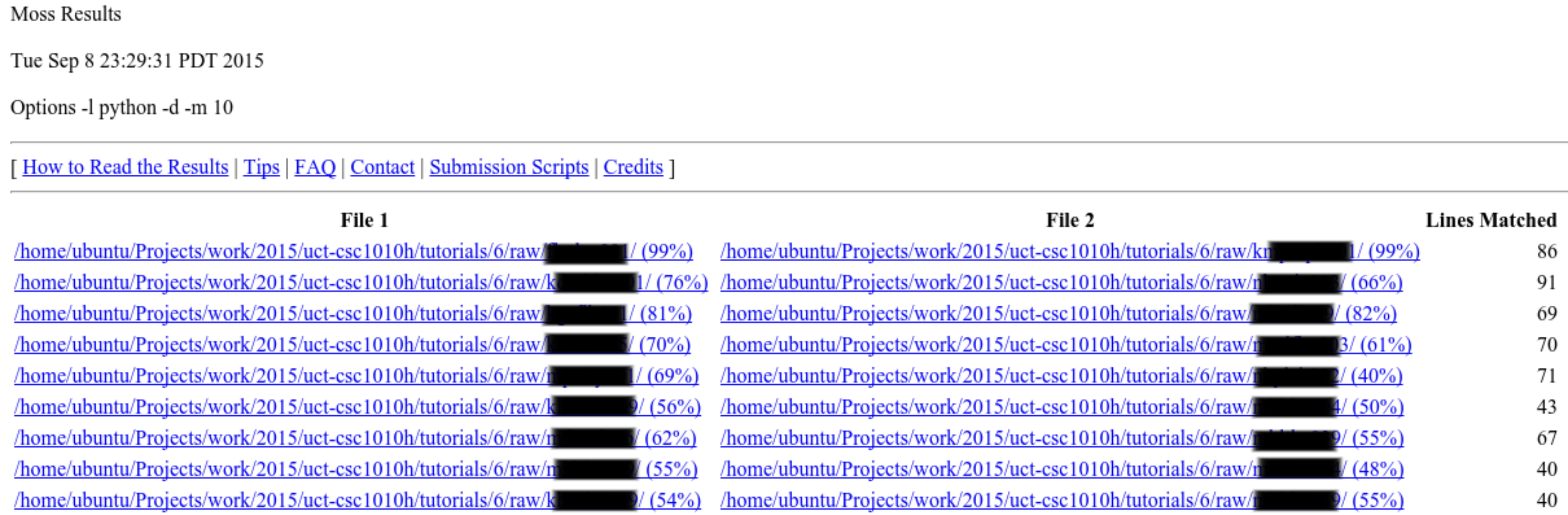
Assignments

- Programming Assignments
  - 3 assignments
  - Will give hands on experience with cloud computing programming
  - Can work in groups of 2
  - Expected to know (or learn quickly) some of these languages and systems:
    - Linux, Virtual Machines, Amazon AWS, Google App-Engine, Hadoop, Swift, multi-threading, sockets, C/C++, Java, Python, Bash
- Project
  - 1 assignment
  - Will enforce theoretical foundation of cloud computing technologies
  - Can work in groups of 2

Cheating will not be tolerated

- MOSS: Measure Of Software Similarity https://theory.stanford.edu/~aiken/moss/
- Automatic system for determining the similarity of programs
- Supports many languages:
  - 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**

MOSS Plagiarism Detection



Moss Results

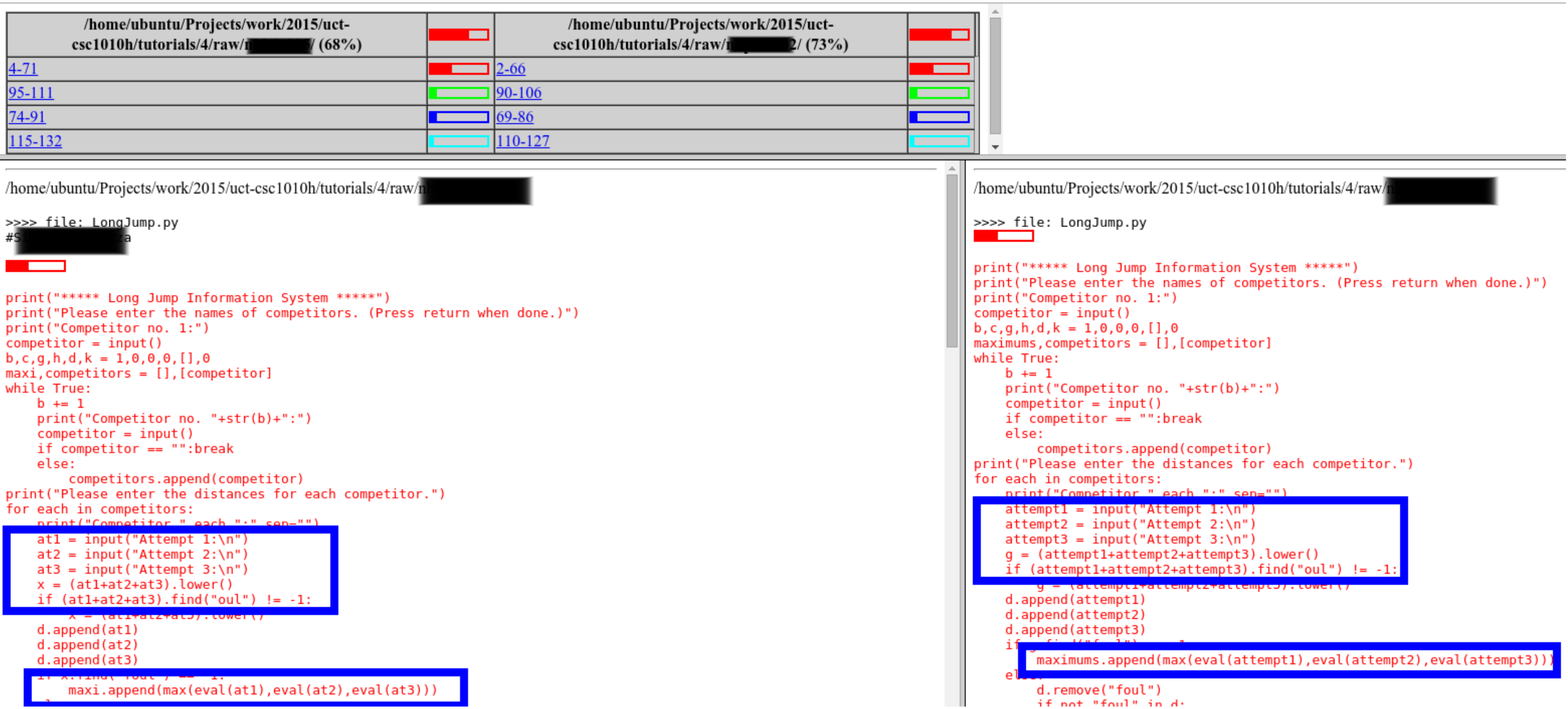
Tue Sep 8 23:29:31 PDT 2015

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/[redacted]/ (99%)	86
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (76%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (66%)	91
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (81%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (82%)	69
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (70%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (61%)	70
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (69%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (40%)	71
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (56%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (50%)	43
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (62%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (55%)	67
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (55%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (48%)	40
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (54%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (55%)	40

MOSS Plagiarism Detection



/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]/ (68%)	🟥	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]/ (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
#S[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

- 2 exams
- The exam 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 are worth 20% each, for a total of 40% of the grade
- Schedule:
  - **Exam #1: Wednesday, October 11th, 2017 from 11:25AM – 12:40PM in Stuart Building 104**
  - **Exam #2: Wednesday, November 29th, 2017 from 11:25AM – 12:40PM in Stuart Building 104**
- **There will be no makeup exam.**

Late Policy

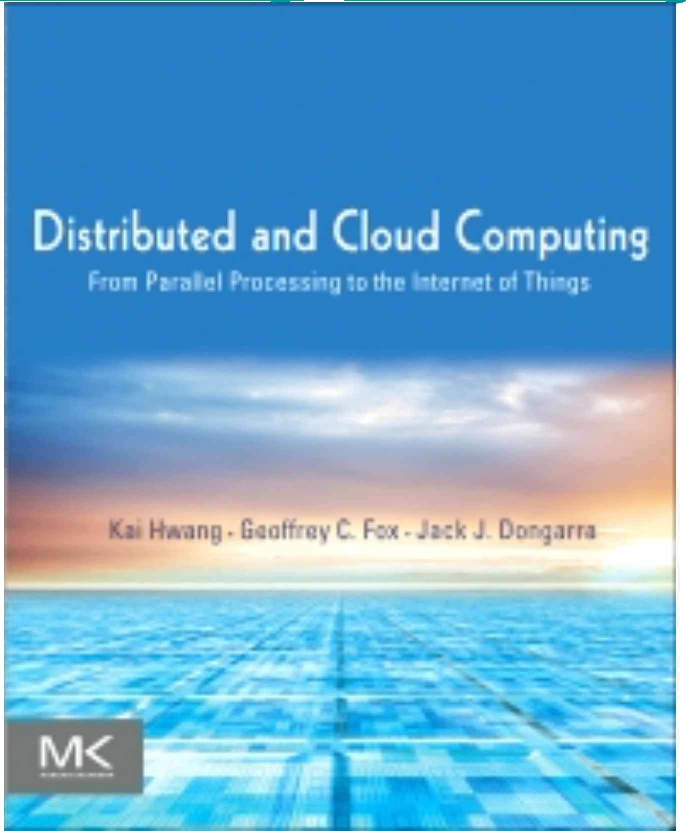
- Assignments will be due at 11:59PM on the date they are due; there will be a 15 minute grace period
- There will also be a 4-day late pass, where students can submit late assignments without penalty
  - The late pass can be used in 1-day increments spread out over multiple assignments.
  - Any late submissions beyond the grace period and beyond the 4-day late pass, will be penalized 20% every day it is late
- Exams
  - There will not be any makeup exams; do not miss any exam or you will get a 0

Grading

- Breakdown:
  - Programming Assignments (3): 50% -- can use late day passes (PA1=15%, PA2=15%, PA3=20%)
  - Project (1): 10% -- can use late day passes
  - Exam (2): 40% -- NO MAKEUPS
- Scale:
  - **A:** 87% ~ 100%
  - **B:** 75% ~ 86% ➔ class average
  - **C:** 60% ~ 74%
  - **E:** 0% ~ 59%

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.


Distributed and Cloud Computing
From Parallel Processing to the Internet of Things
Kai Hwang · Geoffrey C. Fox · Jack J. Dongarra
MK

Questions

- Write me:
  - iraicu@cs.iit.edu
- Call me:
  - 1-312-567-5704
- Write the Tas and me:
  - cs553-f17@datasys.cs.iit.edu
- Online discussion forum:
  - http://piazza.com/iit/fall2017/cs553/home