UCF Stands For Opportunity

CDA6530: Performance Models of Computers and Networks

Cliff Zou

Fall 2012

Course Information

- Teacher: Cliff Zou
  - Office: HEC243 407-823-5015
  - Email: czou@cs.ucf.edu
  - Office hour: TuTh 12pm – 1:30pm
- Course Webpage:
  - http://www.cs.ucf.edu/~czou/CDA6530/index.html
  - Syllabus is on the webpage and WebCourse
- UCF Tegrity for online lecture video streaming
  - WebCourse has a Tegrity link
  - Posted several hours after each class
- Use Webcourse@UCF for homework assignment and grading
  - Keep grade private
  - Homework submission
  - Also have a simple BBS channel

Objectives

- What you need for performance analysis:
  - Review practical probability theory
  - Review several useful random processes
  - Basic queuing theory
  - Practical analysis techniques
  - Useful tools (Matlab and basic usage of NS2)
  - Basic simulation techniques
    - Discrete-time simulation
    - Discrete event simulation
  - Example cases/papers on performance modeling
    - Prepare for your own research

Course Materials

- Reference textbook:
  - Introduction to Probability Models, Ninth Edition by Sheldon M. Ross.
  - Simulation, fouth edition by Sheldon M. Ross.
- Reference courses:
  - CMPSCI673 - Performance Evaluation, by Don Towsley, UMass.
  - Course: COMS6180 - Modeling and Performance Evaluation, by Visal Misra, Columbia Univ.
- Other references that we can find online

Course Introduction

- Coursework — times — approx %
  - Written homework — 2 — 20%
  - Programming projects — 5 — 60%
  - Midterm exam — 1 — 20%
- Written homework
  - Mainly for the first half course on knowledge: probability, random process, queuing theory
- Programming projects
  - Emphasis for the second half of the course
  - Example:
    - Basic usage of Matlab on statistical analysis
    - Internet worm propagation simulation
    - Networking simulation using NS2
    - Discrete-event simulation

Course Introduction

- Midterm exam
  - Focus on the first half content on probability, random process, queuing theory
- Programming projects
  - Simple project: each student individual work
  - Complex project: Two students as a group
  - No final exam (to reduce workload)

- Questions?