EECE 513: Fault-tolerant digital systems

Introduction to the course
(Karthik Pattabiraman)
Administrivia

• **Class hours:** Tuesday & Thursday 10:30-12:00 pm, MCLD 207

• **Office hours:** Wednesday 1-2 PM

• **Textbook:** No text book (I will distribute notes/slides online)

• **Reference Books:**
Prior Background

• **Mathematical background:** I assume you’ve had some basic exposure to probability theory
  – Both discrete and continuous probability distributions
  – Can pick up easily from the Trivedi book (Chap 1-4)
  – Talk to me if you don’t have this background (today)

• **Programming/architecture background**
  – At least one course in software design (e.g., software engg., operating systems) – proficient in programming
  – At least one course in hardware (e.g., architecture)
Class Structure

• **Lectures and paper readings (30 %)**
  – I will lecture roughly every 3 out of 4 classes
  – We will discuss papers in other classes (you need to submit paper reviews ahead of time – 15%)
  – Discussion leading counts for 5% of grade
  – Class participation counts for 10 % of grade

• **Assignments (30 %)**
  – Three assignments each counting for 10%
  – Due in late Sept, Oct and Nov respectively
Class Project

• Major component of course grade (40 %)

• To be done in teams of 2 (3 people allowed in a team if warranted – discuss with me first)

• You are encouraged to integrate it with your own research, but this is not necessary
  – But cannot be the same thing you do for research
  – Talk to your advisor first to ensure this is OK
Final project: Milestones

- In a week or two from today (by Sep 22\textsuperscript{nd})
  - Setup a time to talk to me about project
- By end of September: 2-page proposal (5%)
- By end of October: mid-term report (10%)
- By early December: Presentation (10%)
- By December 15\textsuperscript{th} -> final report (15%)
  - Must be in the form of a conference paper (10 pages)
  - Demo may be shown for implementation projects
Paper Readings

• We will read papers on fault-tolerance techniques at different layers of system stack
  – Each of you needs to submit a 2 page review of the papers by noon the day before the discussion
  – All reviews will be open to everyone after then.
  – Discussion leader summarizes each paper (5 mins), and the points raised by the class along with his/her own points for leading discussion.
  – No late submissions on the reviews, please.
Assignments

• Assignment 1: Basic concepts & probabilities
  – To be done with pencil and paper (maybe Excel)

• Assignment 2: Modeling of computer system
  – Using Mobius tool from UIUC (Markov/SAN modeling)
  – Demonstration of analytical modeling & simulation

• Assignment 3: Fault-injection and analysis
  – Use of simulator or prototype fault-injector
  – Requires some programming/scripting experience
Why Study Reliable Computing -1

• Traditional needs
  – Long-life applications (e.g. space missions )
  – Life-critical, short-term applications (e.g., aircraft engine control, fly-by-wire)
  – Defense applications and Nuclear industry

• Newer critical-computation applications
  – Health industry
  – Automotive industry
  – Industrial control systems, production lines
Why Study Reliable Computing -2

• **Networks and Internet**
  – Wired and wireless networked applications
  – Large data-centers such as Google, Amazon
  – E-commerce, Web 2.0 applications

• **Scientific computing**
  – Reliability is an important issue with the advent of large-scale machines (e.g., Blue Waters will have on the order of hundred thousand processors)
  – Check-pointing and recovery don’t scale well
Why Study Reliable Computing -3

• **Desktop computing**
  – Reliability problems can impact system security
  – Power consumption is becoming important, so we cannot overprovision as we used to in the past
  – Users don’t want to deal with system management

• **Ubiquitous computing**
  – We are seeing an increase in the number of devices around us, and are increasingly relying on their correct operation (e.g., smart phones, sensors ... )
Fault-tolerance

• The ability to provide continued correct operation despite the presence of faults
  – Encompasses a broad rage of techniques ranging from low-level devices to application software
  – Important to ensure that the service behaves as expected provided fault belongs to fault-model
  – There is no such thing as perfect fault-tolerance
    Every fault-tolerance technique has a coverage and a fault-model over which it is evaluated.
Hierarchical fault-tolerance - 1

- Hardware
  - Processor, Memory, Storage system

- Fault-tolerant Middleware

- Operating system

- Applications
  - Application program interface (API)
  - Compiler

What can be provided in software and application itself?

What can be provided by the Middleware layer?

What is typically provided in the operating system?

What can be provided in COTS hardware to ensure fail-silent behavior of system components (nodes, network)?

How to combine hardware and software fault tolerance techniques - (1) fast error detection in hardware, (2) high efficiency detection and recovery in software?

How to assess whether the achieved availability meets system requirements?
Hierarchical Fault-tolerance

SIFT

Applications

Fault-tolerant middleware

Operating system

Hardware

- Application program interface (API)
- Compiler
- CRC on messages, acknowledgment, watchdogs, heartbeats, consistency protocols
- Memory management, detection of process failures, hooks to support software fault tolerance for applications
- Error correcting codes, N_of_M and standby redundancy, voting, watchdog monitors, reliable storage (RAID, mirrored disks)

Checkpointing and rollback, application replication, Algorithm-based techniques, Process pairs, robust data structures, recovery blocks, N-version programming,

Checkpoints and rollback, application replication, Algorithm-based techniques, Process pairs, robust data structures, recovery blocks, N-version programming,
What will you learn in this course?

• Dependable systems design
  – Hardware dependability
  – Duplication and TMR
  – Software approaches
  – Parallel systems
  – Distributed systems
  – Case studies of real-world systems

• Dependability evaluation techniques
  – Combinatorial methods
  – Markov models
  – Stochastic Activity Networks (SANs)
  – Fault-injection
  – Formal methods
  – Statistical methods
Why take this course?

• Exposure to state of the art and traditional techniques in fault-tolerant systems design

• Deep understanding of design choices and trade-offs in real-world fault-tolerant systems

• Rigorous methods and tools for dependability evaluation (useful for industry and research)
  – Opportunity to learn to use tools of the trade
Other reasons to take this class

• Chance to explore a dependability idea related to your research or other interests
  – Talk to your research supervisor first (if applicable)
  – Talk to me if you don’t have a research area or don’t want to align with your research

• You get a chance to hone your research skills
  – Paper discussions, presentations etc. are important, especially for new graduate students
  – Also important for industry jobs, future opportunities
Policies etc.

• All homeworks are due at the beginning of the class on which they’re due (typically Tue/Thu)
• Late submissions will be penalized by 10% for everyday they are late up to 3 days maximum
• Please review UBC’s policy on plagiarism and academic conduct. Ignorance of the same will not be an acceptable excuse for lapses.
• You are expected to attend (and participate in) all lectures and paper discussions in class.
By end of class today ...

• Please fill out the form I’ve handed out
  – Due by Sep 8\textsuperscript{th}, 2011 (next class)
  – Helps me tailor the course content to your interest

• Read the paper on dependability concepts linked from the course website
  – Don’t have to go into details – I’ll do that in class
  – Will assume you’re familiar with the basic terms
Final thoughts

• The goal is to have fun while we learn!
  – I certainly hope you have fun in this course ...

• I am always open to suggestions and critical comments on the course. **Such critical comments will not impact your grade in any way.** On the other hand, good suggestions may even earn you some extra credit/cookies