

CS 294-8

Principles of Fault Tolerant Computing

Kathy Yelick

<http://www.cs.berkeley.edu/~yelick/294>

CS294, Yelick

Introduction, p1

Today's Agenda

- Motivation and trends
- Examples of failures
- Background in reliability computing
- Course Overview
- Administrivia

CS294, Yelick

Introduction, p2

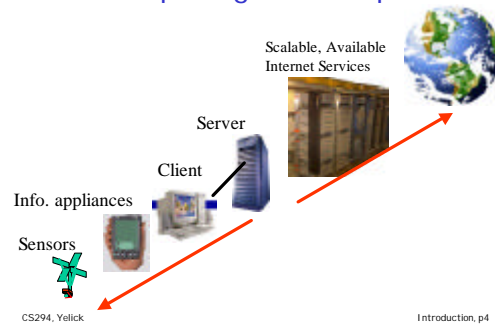
Ubiquitous Computing

- Computing everywhere:
 - Desktop, Laptop, Palmtop, Cars, Cellphones
- Input devices everywhere:
 - Sensors, cameras, microphones
- Connectivity everywhere:
 - Rapid growth of bandwidth in the interior of the net
 - Broadband to the home and office
 - Wireless technologies such as CMDA, Satellite, laser
- Increased reliance on computers is inevitable
- Computer systems will become invisible only when they are reliable

CS294, Yelick

Introduction, p3

Computing Landscape



CS294, Yelick

Introduction, p4
Culler, 1999

The "Post-PC" Era

PostPC Era Divides built on two technologies:

- 1) Mobile Consumer Electronic Devices
 - e.g., successor to PDA, Cell phone, wearable computers
- 2) Infrastructure to Support such Devices
 - e.g., successor to Big Fat Web Servers, Databases

CS294, Yelick

Introduction, p5

The problem space: big data

- Big demand for enormous amounts of data
 - today: enterprise and internet applications
 - online applications: e-commerce, mail, web, archives
 - enterprise decision-support, data mining databases
 - future: richer data and more of it
 - computational & storage back-ends for mobile devices
 - more multimedia content
 - more use of historical data to provide better services
- Two key application domains:
 - storage: public, private, and institutional data
 - search: building static indexes, dynamic discovery

CS294, Yelick

Introduction, p6

Application: Tornado Response

- CAPS at University of Oklahoma
- Currently 12 radars in Oklahoma area
 - Improve warning time: saved 800 lives?
- Two problems:
 - Real-time computation and response
 - Some local to one radar
 - Better algorithms involve coordination
 - Archival of data for experimentation and long term analyses (“data mining”)
 - Petabytes per year

CS294, Yelick

Introduction, p7

Application: Smart Buildings

- Buildings adapt to occupants and save energy
 - Save \$55 billion in the U.S.
 - Reduce carbon emissions by 35 million metric tons
- Sensors with wireless connections
- Integrated with server to record history information and do prediction

CS294, Yelick

Introduction, p8

Application: Earthquakes

- Reduce the risk and improve response to earthquakes
- Use millions of MEMS sensors in buildings, ground, bridges, etc.
- Front-end processing in sensor
- Tied to backend data bases and computational models
- Building will “self-diagnose” after an earthquake.

CS294, Yelick

Introduction, p9

Application: Transportation

- Traveler information service
 - Limited for exists in Europe
 - Prototype from Path project in LA
- “Mine” sensor data from roads to predict travel times
- Traffic manage apply controls (traffic ramp meters) in real time
- Improve long term highway planning

CS294, Yelick

Introduction, p10

Summary of Post-PC Era

- Computing and data in the extremes: tiny devices and enormous “utility-style” servers
- Applications entertainment and business will continue
- New applications that make computing transparent in the environment require reliability

CS294, Yelick

Introduction, p11

Today's Agenda

- Motivation and trends
- **Examples of failures**
- Background in reliability computing
- Course Overview
- Administrivia

CS294, Yelick

Introduction, p12

eBay Crash

- eBay: giant internet auction house
 - A top 10 internet business
 - Market value of \$22 billion
 - 3.8 million users as of March 1999
 - Bidding allowed 24x7
- June 6, 1999
 - eBay system is unavailable for 22 hours with problems ongoing for several days
 - Stock drops by 6.5%, \$3-5 billion lost revenues
 - Problems blamed on Sun server software
 - Similar to EECS server downtime?
- Shorter downtimes common

CS294, Yelick

Introduction, p13

EECS Servers Crash

- Department servers are offline for 3-7 days
 - Cause is disk failure coupled with incompatible RAID software
- Power failure also resulted in lost data in a separate event
 - UPS not purchased due to staff turnover

CS294, Yelick

Introduction, p14

Ariane 5 Rocket Crash

- Ariane 5 and its payload destroyed about 40 seconds after liftoff
- Error due to software bug:
 - Conversion of floating point to 16-bit int
 - Out of range error generated but not handled
- Testing of full system under actual conditions not done due to budget limits
- Estimated cost: 120 million DM

CS294, Yelick

Introduction, p15
Risks Digest

The Therac-25 Failure

- Therac-25 is a linear accelerator used for radiation therapy
- More dependent on software for safety than predecessors (Therac-20, Therac-6)
- Machine reliably treated thousands of patients, but occasionally there were serious accidents, involving major injuries and 1 death.
- Software problems:
 - No locks on shared variables (race conditions).
 - Timing sensitivity in user interface.
 - Wrap-around on counters.

CS294, Yelick

Introduction, p16
Fox and Dill, 1999

Tele Denmark

- Tele Denmark Internet, ISP
- August 31, 1999
 - Internet service down for 3 hours
 - Truck drove into the power supply cabinet at Tele Denmark
 - Where were the UPSs?
 - Old ones had been disconnected for upgrade
 - New ones were on the truck!

CS294, Yelick

Introduction, p17
Risks Digest & rec.humor

Lampson: Systems Challenges

- Systems that work
 - Meet their specs
 - Always available
 - Adapt to environment & evolve over time
 - Made from unreliable components
 - Grow without practical limit
- Credible simulations or analysis
- Writing good specs
- Testing
- Performance
- Understanding when it doesn't matter

"Computer Systems Research: Past and Future" -Butler Lampson, Microsoft
SOSP Keynote, Dec. 1999

CS294, Yelick

Introduction, p18

Hennessy: The "New World" Focus

- Availability
 - Both appliance & service
- Maintainability
 - Two functions:
 - Enhancing availability by preventing failure
 - Ease of SW and HW upgrades
- Scalability
 - Especially of service
- Cost
 - per device and per service transaction
- Performance
 - Remains important
 - But its not SPECint

"Back to the Future: Time to Return to Longstanding Problems in Computer Systems?" -John Hennessy, Stanford FCRC Keynote, May 1999

CS294, Yelick

Introduction, p19

Today's Agenda

- Motivation and trends
- Examples of failures
- **Background in reliable computing**
- Course Overview
- Administrivia

CS294, Yelick

Introduction, p20

Aspects of Reliability

- **Safety:** "First, do no harm"
- **Fault tolerance:** faults should (at worst) result in graceful degradation
- **Predictability:** behavior should be a function of inputs and environment; should be reproducible
- **Timeliness:** real-time constraints, Quality of Service (QoS) guarantees

CS294, Yelick

Introduction, p21

Availability vs. Reliability

- Jim Gray's 85 paper (see class web page) distinguishes these
 - Reliability is measured by mean time between failures (MTBF)
 - Availability is a function of MTBF and mean time to recover (MTTR)
$$\text{MTBF}/(\text{MTBF}+\text{MTTR})$$
 - A system may have a high MTBF, but low availability

CS294, Yelick

Introduction, p22

Fault Recovery

- How quickly is the fault detected?
- How soon can recovery begin?
 - Does it require human intervention
 - How is the sysadmin notified?
- How long does recovery take?
 - Restore from backup?
 - Purchase new HW?

CS294, Yelick

Introduction, p23

Two Keys to Availability

- Redundancy
 - "The one good idea"
 - May be in software, hardware, data structures (state), programmers, etc.
- Modularity
 - Reduce the size of the failure unit (FRU)
 - Change failure model from continuous to discrete
 - 90% of client machines available 90% time

CS294, Yelick

Introduction, p24

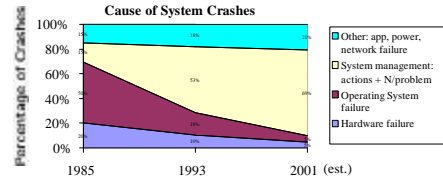
Causes of Faults: Tandem

- In Gray's '85 survey of Tandem customers
 - 30% were “infantile” failures
 - The rest were broken into (roughly):
 - Administration 42%
 - Software 25%
 - Hardware 18%
 - Environment (power, etc.) 14%
 - Unknown 3%

CS294, Yelick

Introduction, p25

Causes of Faults: Vax



- VAX crashes '85, '93 [Murp95]; extrapol. to '01
- System Management includes:
 - Multiple crashes per problem
 - Sysadmin Actions: set params, config, bad app install
- HW/OS 70% in '85 to 28% in '93. In '01, 10%?
 - Sysadmin increasingly important

CS294, Yelick

Introduction, p26

The Fail-Fast Principle

- Reliable systems should be built from components that fail fast
 - No answer is better than the wrong answer
 - Improves latency of fault detection
 - Fault containment is better if modules stop

CS294, Yelick

Introduction, p27

The Heisenbug Hypothesis

- Software faults can be divided into
 - Bohrbugs – easily reproduced
 - Hiesenbugs – transient, hard to produce
- Conventional wisdom is that most bugs in running systems are Heisenbugs
 - Is this true? SW and HW? Open source too?
- HW redundancy (processor pairs) can help with Hiesenbugs

CS294, Yelick

Introduction, p28

Today's Agenda

- Motivation and trends
- Examples of failures
- Background in reliable computing
- Course Overview
- Administrivia

CS294, Yelick

Introduction, p29

What is Fault Tolerance?

- A better title might have been “reliable” or “available” computing
- We will be looking at:
 - The “classics” (Gray, Lamport, Birman,...) in distributed computing
 - Recent results (Coding-based replication, practical byzantine fault tolerance,...)
- Avoid overlap with 262AB (Coda and Bayou possible exceptions)
- Not software verification/quality: See Wolfgang Pree's course instead

CS294, Yelick

Introduction, p30

What is Meant by “Principles”?

- We will study
 - Models of distributed systems and faults
 - Distributed algorithms
 - Reasoning techniques
- Things that every system designer should know, aside from the experience papers

CS294, Yelick

Introduction, p31

Course Goals

- Prepare for research in reliability
 - Put structure on past work
- Identify major open problems and possible approaches
 - Can cheap hardware be used in place of expensive humans? Bugs? Maintenance?
 - What is the user's view? Are “weak” consistency models acceptable?
 - To what extent can self-monitoring, self-healing systems help?

CS294, Yelick

Introduction, p32

Administrivia

- Class times:
 - Tuesday 3:30-5:00 here in 380
 - Thursday: seminar in 306 3:30-4:30
 - Except this week: Thursday 3:30 in 380
 - Thursday: discussion 4:30-5:30 in 380
- Work
 - Readings (some write-ups)
 - Read ISTORE paper for Thursday
 - Small homework assignments
 - Project: presentations/poster + paper

CS294, Yelick

Introduction, p33

This Week

- Read ISTORE paper for Thursday
- Homework 1 due next Tuesday:
 - Anatomy of a failure
- Read Grapevine and Porcupine papers for next Tuesday
- Read M. Baker paper for 9/7

CS294, Yelick

Introduction, p34