Recovery-Oriented Computing

Aaron Brown, Dan Hettenna, David Oppenheimer, Noah Treuhaft, Leonard Chung, Patty Enriquez, Susan Housand, Archana Ganapathi, Dan Patterson, Jon Kuroda, Mike Howard, Matthew Mertzbacher, Dave Patterson, and Kathy Yelick

University of California at Berkeley

In cooperation with

George Candea, James Cutler, and Armando Fox

Stanford University

http://roc.CS.Berkeley.EDU/

RECOVERY-ORIENTED COMPUTING

Agenda

- Schedule Wednesday

1:00 Intro, ROC talks

3:00 Break

3:30 OceanStore

6:00 Dinner

7:30 Panel Session: "Challenges and Myths in Operating Reliable Internet Services"

- Wireless Interect access during breaks (no access during talks/panels)
  - Network name is "nest"

Target is Services

- **Companies like Amazon, Google, Yahoo, …**
- **Also Internal ASP model**
  - Enterprise IT as services
- **Since providing a single service, can do things differently**
  - Fascinating solutions to hard problems
  - Change software while continually providing service
- **Since providing service, availability is killer metric**
- **Plausible model for future of IT?**

The past: goals and assumptions of last 15 years

- **Goal #1: Improve performance**
- **Goal #2: Improve performance**
- **Goal #3: Improve cost-performance**
- **Assumptions**
  - Humans are perfect (they don't make mistakes during installation, wiring, upgrade, maintenance or repair)
  - Software will eventually be bug free (good programmers write bug-free code, debugging works)
  - Hardware MTBF is already very large (~100 years between failures), and will continue to increase

Today, after 15 years of improving performance

- **Availability is now the vital metric for servers**
  - near-100% availability is becoming mandatory
    - » for e-commerce, enterprise apps, online services, ISPs
  - but, service outages are frequent
    - » 65% of IT managers report that their websites were unavailable to customers over a 6-month period
      - 25%: 3 or more outages
  - outage costs are high
    - » social effects: negative press, loss of customers who "click over" to competitor
    - » $500,000 to $5,000,000 per hour in lost revenues

Source: InternetWeek 4/3/2000

New goals: ACME

- **Availability**
  - 24x7 delivery of service to users
- **Change**
  - support rapid deployment of new software, apps, UI
- **Maintainability**
  - reduce burden on system administrators (cost of ownership ~5X cost of purchase)
  - provide helpful, forgiving sysadmin environments
- **Evolutionary Growth**
  - allow easy system expansion over time without sacrificing availability or maintainability

Where does ACME stand today?

- **Availability: failures are common**
  - Traditional fault-tolerance doesn't solve the problems
- **Change**
  - In back-end system tiers, software upgrades difficult, failure-prone, or ignored
  - For application service over WWW, daily change
- **Maintainability**
  - human operator error is single largest failure source?
  - system maintenance environments are unforgiving
- **Evolutionary growth**
  - 1U-PC cluster front-ends scale, evolve well
  - back-end scalability still limited

Recovery-Oriented Computing Philosophy

"If a problem has no solution, it may not be a problem, but a fact, not to be solved, but to be coped with over time"

— Shimon Peres

- **Failures are a fact, and recovery/repair is how we cope with them**
- **Improving recovery/repair improves availability**
  - UnAvailability = $\frac{\text{MTTR}}{\text{MTTF}}$ *(assuming MTTR much less than MTTF)*
  - 1/10th MTTR just as valuable as 10X MTBF
- **If major Sys Admin job is recovery after failure, ROC also helps with sys admin**
- **If necessary, start with clean slate, sacrifice disk space and performance for ACME**

ROC Approach

- **Work with companies to get real data on failure causes and patterns**
  - David Oppenheimer's survey 3 sites
  - Patty Enriquez's survey of FCC switch failure data
- **Develop ACME benchmarks to test old systems & new ideas to measure improvement**
  - Fastest to Recover from Failures v. Fastest on SPEC
  - Poster session database benchmark: Chang & Brown
  - Friday: Fault Insertion in glibc (CS 294/444A class)
  - Friday: Automating Root Cause Analysis (294/444A)
- **Support for humans to operate services**
  - Aaron Brown's talk on Undo for SysAdmin

ROC Approach

- Cluster technology that enables partition systems, insert faults, test outputs
- ISTORE(ROC-I): Cluster of 64 PCs modified with ability for HW isolation, fault insertion, monitoring, diagnostic processors
- Cluster of 40 IBM PCs: each with 2 GB DRAM, 1 gigabit Ethernet, gigabit switch,HW monitor, each running Vmware virtual machine monitor (software layer)

ISTORE HW update

- **Difficulties with dual power supplies**
  - power drops during startup, flaky powerup of dual power supplies
  - led to rework and 2nd revision of backplane
  - continued problems getting both power supplies to work together
  - decision to move on using one out of two power supplies for now
- **Now testing bricks before fabricating remainder of backplanes.**

ISTORE Software Update

- Development for diagnostic processors (DPs)
  - Sensor library: software API to access temperature, vibration, humidity and other sensors
  - DP network protocol: reliable connection-based protocol over CAN-bus hardware
  - Remote logging interface: can log system events from a brick on a remote PC
    - » Useful for debugging and for sensor data analysis
  - Brick coordination protocol: synchronization and coordination between bricks, used for
    - » Power-up phase, to avoid power surge
    - » Accessing devices shared by a shelf on the backplane

ISTORE Network

- 4 100 Mb Ethernets interfaces/brick:
  - Bandwidth through striping
  - Failure resistant: striping automatically adjusts for failed links
- Two ISTORE programming models
  - Cluster with Linux PCs + TCP
    - » Cluster servers, such as Apache run
    - » Problem: Pentium can saturate 2 of 4 links
  - As Parallel Java (Titanium) platform
    - » User level UDP for lower overhead communication
      - Recently rebuilt due to eliminate concurrency problems
    - » Protection from language model