Internet Services

Conclusion to Internet Services
Richard Martin
Rutgers University
Department of Computer Science
May. 2003

Course Structure

• Critique Papers: ↔
  • Examples of good and poor research
• Student Presentations: 🤧
  • Practice presenting ideas
• Position Papers 🤧
  • Practice writing and reviewing
• Project ?
  • Practice experimentation and evaluation

Themes in the Course

• Availability & Reliability
• Network Performance
• End System Performance
• Location Based Services
• Not Covered:
  • Content distribution
  • Security
  • User access
  • Architectures (hardware and software)

Availability

• Hard to model at right level of accuracy
  • Stochastic reasoning appropriate?
  • Probabilistic reasoning appropriate?
• Little empirical data to supporting work
  • Beginning to understand these tradeoffs
• 10 years on, software and operators still major sources of outages
  • still at 399s?
• "Time heals all wounds?"
Human Impact

- Hard to quantify availability
  - Analytic models of operator actions
    - State machines
    - Fault trees
    - Action grammars
  - Likelihoods of errors & impact
- Restricting actions?
  - How to do this in a computer system?
    - E.g. Complexity of user interface for shell vs. car
    - Is this possible?

Software Availability

- 30 years of software engineering
  - People still use print to debug
  - Software still horrible
  - Mythical man-month still relevant
  - Xtreme programming good, but doesn't attack essence

Quantify Intuition

- High availability software exists
  - What is being programmed?
  - How is it created?
  - What is its architecture?
  - What is not done?
- Can these processes be
  - Described?
  - Checked?
  - Automated?
Internet Tomography

- Measuring the real internet
- Creating realistic models
- Understanding current situation
- Implications for services?
  - How can you use this information?
  - # of access points?
  - Distribution of access points?
  - Likelihood of failure?
  - 5 nines not such a big deal? (4 is OK?)

End Performance

- Is performance a solved problem?
  - Under normal operating conditions?
- When do other problems demand more attention?
  - Security, availability, billing Ø
- Performance under overload not well understood

Overload and QoS

- Over-engineering the worst case by safety factor
  - traditional engineering approach
- Computer systems have other options
  - Denial of service
    - Admissions control
    - Drop
  - Degradation of service
    - lower QoS over load
- How to easily implement?

Location based Services

- All talk, poor motivation
  - Still no killer app
    - Compelling reason to care about user's location
- Most likely will come from something
  - stupid
    - geocaching
    - graffiti
- Will drive a lot of infrastructure
  - Change the way people perceive location
not covered:
WAN content distribution

- Is this theme played out?
  - Akamai, Fast-forward, Inktomi
  - Squid
- What happened to VoD?
  - One student paper on multi-cast
  - Bandwidth savings vs. other stuff

Not Covered: Security

- Hard to do justice in this class
- Huge problem or red-herring?
  - Headache for everyone
  - Permeates all levels and systems (language, net, OS)
- Quantify how secure a given system is?
- A global methodology for securing systems

Not Covered: Architectures

- SMP/MPP/Cluster debate is over?
- Software architectures ossified
  - 2-3 Tiers
  - Historical Accident?
  - What if we had a clean slate?

New Themes

- Personal Search and Retrieval
  - I have too much stuff
- What environments or systems make services more programmable?
  - Can your grandmother set-up a web page?
  - Run a service?
  - Create a web-store (eBay?)
- Web Services
  - Offer client/server over the web in a standardized way.
Summary

Good luck on the projects!

Due Sunday May 11th