CS 424/624 Reliable Software Systems

Lecture 2: Empirical Study

Prof. Ryan Huang
Administrivia

- Sign up paper presentations
- Look for course project teammates
- Join “#paper-discussion” channel on Slack
  - Can use Anonymous bot
Importance of Studying System Failures

• We often learn more about reliability design from systems that failed than systems that succeeded
  - Many surprises…

• Study ≫ number game

• A good study on system failures yields deep insights that inspire solutions
  - Which problems are important but overlooked, what the common patterns are, etc.
  - E.g., the concept of margin of safety
Regular meetings to review and analyze all major incidents

- A written report to summarize what went wrong, how the issue was resolved, what to improve, etc.
- Typically called Root Cause Analysis (RCA) or Postmortem Analysis
- Reports of very serious outages are usually posted publicly
Numerous papers that study failures on different kinds of systems
- Some focus on a sub-type of failures

A Common Research Topic In Academia

Why do Internet services fail, and what can be done about it?

David Oppenheim, Athanasios Carzaniga, and David A. Patterson
University of California at Berkeley, EECS Computer Science Division
387 Soda Hall, 4170 Berkeley, CA 94720-1776, USA
(davidopp, archanap, patterson)@eecs.berkeley.edu

Abstract
In 1999, the Cisco published its landmark study on the cause of failures of telecom systems and the techniques Telecom used to prevent such failures [1]. Unfortunately, these lessons haven’t resulted in improved fault-tolerant services as os the new kid on the 2006-conference

Why Does a Cloud-Scale Service Fail Despite Fault-Tolerance?

Peng Huang, Xinlin Jia, William J. Bolosky1, Yuanyuan Zhou
University of California, San Diego
Microsoft Research2, Microsoft2

Abstract
The sheer scale and complexity of the cloud mean that even millions of research into fault-tolerance and software engineering for reliability, billions of dollars of invest-

Why Does the Cloud Stop Computing?

Lessons from Hundreds of Service Outages

Hayati S. Osman, Mingzhe Hao, and Risa O. Semino
University of Chicago

Abstract
We conducted a cloud outage study (COOS) of 72 popular Internet services. We analyzed 1427 headline news and public-post-mortem reports that detail 2055 published outages that

Learning from Mistakes — A Comprehensive Study on Real World Concurrency Bug Characteristics

Shan Lu, Sooyeon Park, Eunsoon Seo and Yuanzhen Zhou
Department of Computer Science,
University of Illinois at Urbana-Champaign, Urbana, IL 61801
(shanlu,specs,seom2,yzhou)@illinois.edu

Abstract
The reality of multi-core hardware has made concurrent programming pervasive. Unfortunately, writing correct concurrent programs is

Simple Testing Can Prevent Most Critical Failures

An Analysis of Production Failures in Distributed Data-intensive Systems

Dong Yuan, Ye Yao, Xin Zhang, Guilherme Bensa Rodrigues, Xu Zhao,
Yongfei Zhang, Peiyuan U. Zufan, Michael Stone
University of Toronto

Abstract
Large, production-quality distributed systems still fail periodically, and do so sometimes catastrophically, where most or all users experience an outage or data loss. We present the results of a comprehensive study of 138 widely deployed, open-source, data-intensive systems. Our investigations focus on the accuracy and effectiveness of existing approaches to testing, and we analyze 54 outages that were identified using our method.

Understanding, Detecting, and Localizing Partial Failures in Large System Software

Chang Lou, Phong Nguyen, and Scott Smith
Johns Hopkins University

Abstract
Partial failures occur frequently in complex systems and are a common reason for system downtime and data loss, despite the existence of advanced testing and monitoring technologies. We propose a new approach to understanding and localizing partial failures. Our system, FailureFinder, uses a combination of static analysis and dynamic monitoring to identify the cause of a partial failure. We evaluate FailureFinder on a number of real-world systems and show that it is able to accurately identify the root cause of failures in a variety of scenarios.
Today: Two Early Empirical Studies

- One is a classic work that basically started this line of research in systems community

- The second is one of first systematic studies on OS bugs
Why Do Computers Stop and What Can Be Done About It?
• Jim Gray
  - A pioneer computer scientist
  - Received the Turing Award in 1998
    ▪ for seminal contributions to database and transaction processing research and technical leadership in system implementation
  - “ACID”, granular database locking, “five-minute rule” in caching, …
  - Disappeared with his sailboat in the waters in San Francisco in Jan 2007
Mean Time *Between* Failure (MTBF)

Mean Time To Recovery (MTTR)

Mean Time To Failure (MTTF)
### Background: Reliability vs. Availability

- **Reliability:** systems not doing wrong things
  - proportional to MTBF
- **Availability:** systems doing right things on time
  - related to MTBF and MTTR

#### Definition 1

\[
\text{Availability} = \frac{E[\text{uptime}]}{E[\text{uptime}]+E[\text{downtime}]}
\]

#### Definition 2

\[
\text{Availability} \approx \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}
\]
An Empirical Study of Operating Systems Errors
How Complex Systems Fail