

Towards Automated Problem Analysis of Large-Scale Storage Systems

Priya Narasimhan
Greg Ganger
Chuck Cranor

Carnegie Mellon University

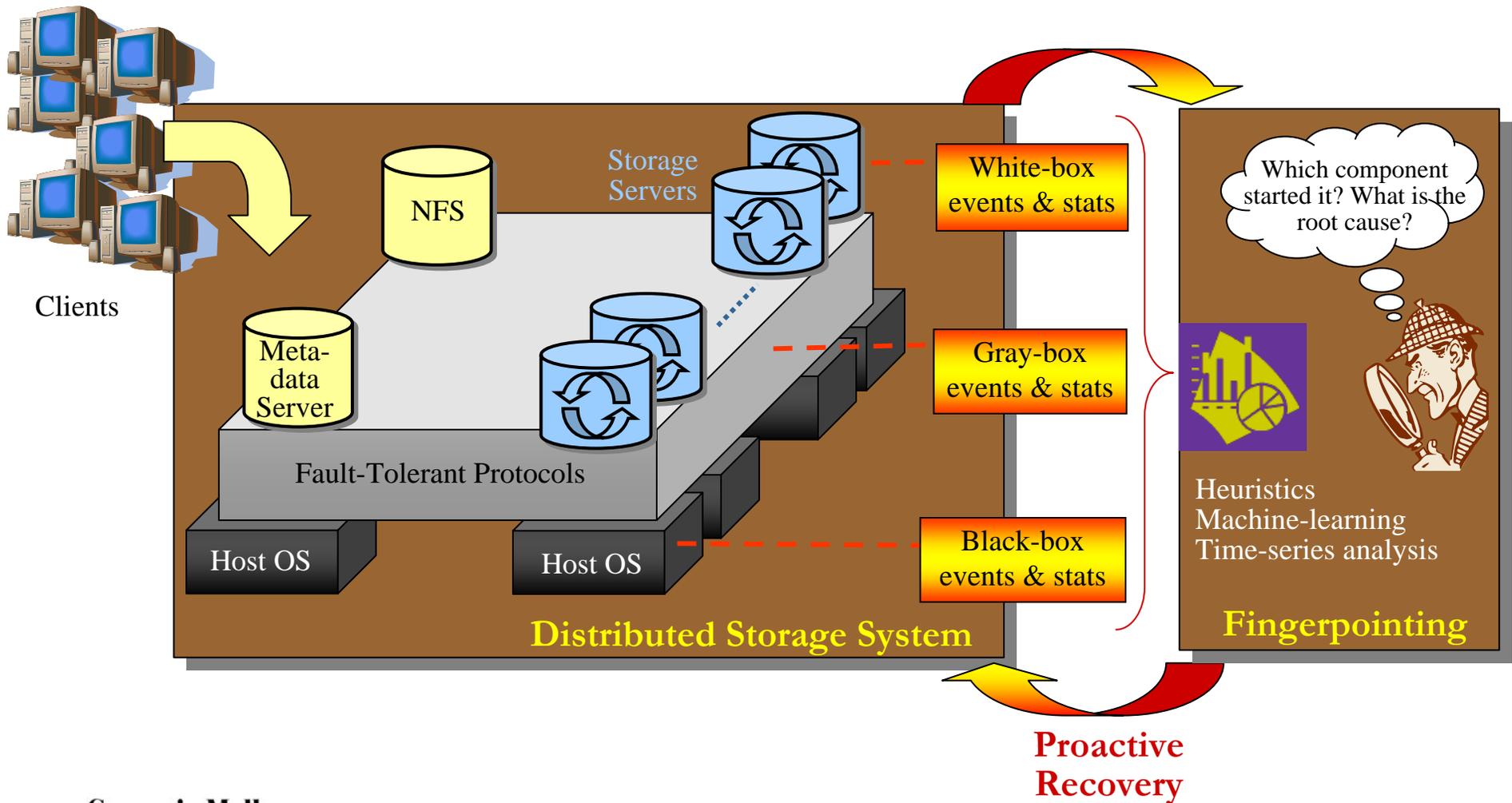


Automated Failure Analysis

- Failure analysis difficulty...
 - Creates major problems for administrators
 - Worsens as scale and complexity grows
- Goal: automate it and get proactive
 - Failure detection and prediction
 - Problem determination (or **fingerpointing**)
- How: Instrumentation plus statistical-analysis tools



Fingerprinting Approach



Major Accomplishments



- Gathered list of failures in distributed storage systems
- Developed initial fingerprinting approach
 - Black-box instrumentation to collect performance metrics
 - Algorithms to analyze the monitored metrics, using
 - Anomaly detection
 - Machine learning
 - Time-series analysis and correlation
- Successfully fingerprinted failures in two fault-tolerant distributed storage systems

Target List of Problems

- Poor performance
 - Caching disabled, working poorly
 - Blocksize, striping, rotation factor (configuration error)
 - Increased serialization of I/O and network requests
 - Data structures or algorithms run lower than expected
- Resource leaks
 - Leak of protocol resource (e.g. file handles)
 - Memory leak
- Correctness
 - Cache consistency
 - Files end up empty or have wrong contents
 - Directory entries are missing
- Hangs
 - Infinite loop on operation
- Miscellaneous - Server won't start
 - Port in use, bad saved state

Gathered from
bug-tracking
databases for

- Ursa Minor
- CODA
- CITI Linux NFSv4
- OpenAFS

Experimental Methodology

- Instrumented distributed fault-tolerant storage systems
 - CMU's Ursa Minor (supported by the PASIS read/write protocols)
 - MIT's Castro-Liskov BFS (supported by BFT protocols)
 - Tested with the iozone file-system benchmark
- Black-box instrumentation through the SAR toolkit
 - Packets/second, total/free/used memory, context-switch rate, CPU usage, number of bytes read, number of bytes written
- Techniques applied so far
 - Simple machine-learning techniques [[USENIX SysML 2007](#)]
 - Heuristics and time-series analysis [[CMU-PDL-TR-06-107](#)]
 - Correlation-based techniques

Fingerprinting Approach

- **Local anomaly detection + global fingerprinting**
 - **Local** (node-level) anomaly detector
 - Mean- and variance-based thresholds to flag anomalies in metrics
 - Sends time-stamped anomalies and anomaly counts to fingerprinter
 - **Global** (system-wide) fingerprinting
 - For each metric, count anomalies in an x -second window
 - Trigger fingerprinting if anomaly counts exceed a specific threshold
 - Fingerprint faulty node using
 - Heuristic approach, k -means clustering, k -nearest neighbor
- **Global correlation on the raw time-series data**
 - Pair-wise correlation between pairs of servers/nodes
 - Using windowed versions of time-series of all performance metrics
 - Faulty node's pair-wise correlation will stand out as different

The Elephant in the Room

- False positives, false positives, false positives,
- Workload changes can be mistakenly identified as anomalies
 - Sudden burst of clients (flash-crowd)
- System reconfigurations can be mistakenly identified as anomalies
 - Addition/removal of nodes
 - Upgrades
- Mode changes can be mistakenly flagged as anomalies
 - Off-peak vs. peak-load
 - Backup of the system
- Of course, the anomaly-detector itself has a false-positive rate

Initial Insights

- Can exploit replication in fingerprinting
 - Allows for peer comparison in approach
 - Algorithms can also be used in a non-replicated setting
- False-positive rate dependent on the tuning of the algorithms
- Fault manifestations can travel
 - Performance problem observed on a (faulty) node can propagate to other (non-faulty) nodes
- No single performance metric is sufficient for root-cause analysis
- Different failures have different signatures

Next Steps

- **More experimentation**
 - Target other benchmarks
 - Expand failure-injection campaign
 - Include non-replicated targets
- **Use data from real system usage**
 - Deployed instance of Ursa Minor used continuously
- **Leverage white-box and gray-box data**
- **Improve algorithms**
 - Leverage topology and dependency information
 - Automated tuning to reduce false-positive rates



Publications

- Soila Pertet, Rajeev Gandhi and Priya Narasimhan, "[Fingerpointing Correlated Failures in Replicated Systems](#)", *USENIX Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML)*, Cambridge, MA, 2007
- Michael P. Kasick, Priya Narasimhan, Kevin Atkinson, Jay Lepreau, "[Towards Fingerpointing in the Emulab Dynamic Distributed System](#)", *USENIX Workshop on Real, Large Distributed Systems (WORLDS)*, Seattle, WA, 2006
- Eno Thereska, Brandon Salmon, John Strunk, Matthew Wachs, Michael Abd-El-Malek, Julio Lopez, Gregory R. Ganger, "[Stardust: Tracking Activity in a Distributed Storage System](#)", *International Conference on Measurement and Modeling of Computer Systems, (SIGMETRICS)*, Saint-Malo, France, 2006
- Eno Thereska, Dushyanth Narayanan, Anastassia Ailamaki, Gregory R. Ganger, "[Observer: Keeping System Models from Becoming Obsolete](#)", *Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-07-101*
- Soila Pertet, Rajeev Gandhi and Priya Narasimhan, "[Group Communication: Helping or Obscuring Failure Diagnosis?](#)", *Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-06-107*
- Raja R. Sambasivan, Alice X. Zheng, Eno Thereska, Gregory R. Ganger, "[Categorizing and differencing system behaviors](#)", *IEEE Workshop on Hot Topics in Autonomic Computing*, Jacksonville, FL, 2007
- Eno Thereska, Anastassia Ailamaki, Gregory R. Ganger, Dushyanth Narayanan, "[Observer: keeping system models from becoming obsolete](#)", *IEEE Workshop on Hot Topics in Autonomic Computing*, Jacksonville, FL, 2007

Project Participants

- Faculty

- Priya Narasimhan
- Greg Ganger
- Chuck Cranor

- Post-doc

- Alice Zheng

- Graduate students

- Michael Abd-El-Malek
- Soila Pertet
- Raja Sambasivan
- Eno Thereska

- Undergraduate students

- Keith Bare
- Michael Kasick



<http://www.pdl.cmu.edu/ProblemAnalysis>