Toward automated problem analysis of large-scale storage

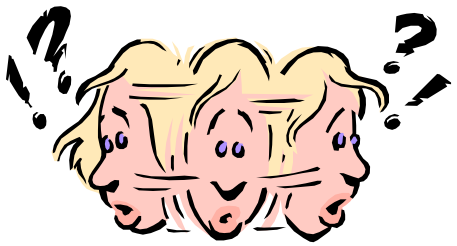
Priya Narasimhan, Greg Ganger, Chuck Cranor

Automated Problem Analysis

- Problem analysis is very difficult
  - for administrators and debuggers alike
  - ever-worsening as scale grows
- Goal: automate it and get proactive
  - failure detection and prediction
  - problem determination (causes and effects)
- How: instrumentation plus statistical tools

Problem determination is very difficult

- Goal: identify failed components and root cause
  - guiding repair and prevention of recurrence
- Very challenging in large-scale environment
  - can have multiple manifestations with a single cause
  - can have multiple causes for a single manifestation
  - problems and/or their manifestations can “travel” among communicating components
  - … and might just be workload change



A bit more on approach

- Collect and orchestrate instrumentation
  - explicitly reported “events”
    - significant changes/hiccups observed by components
  - activity tracking

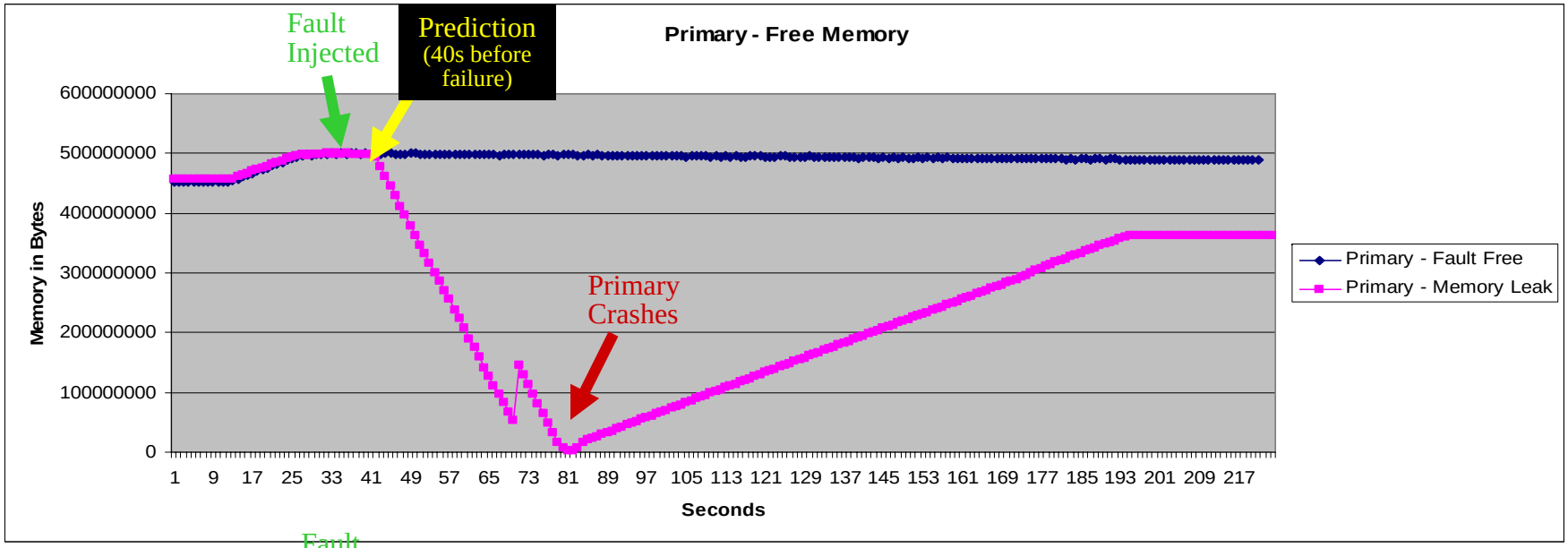
Activity tracking instrumentation

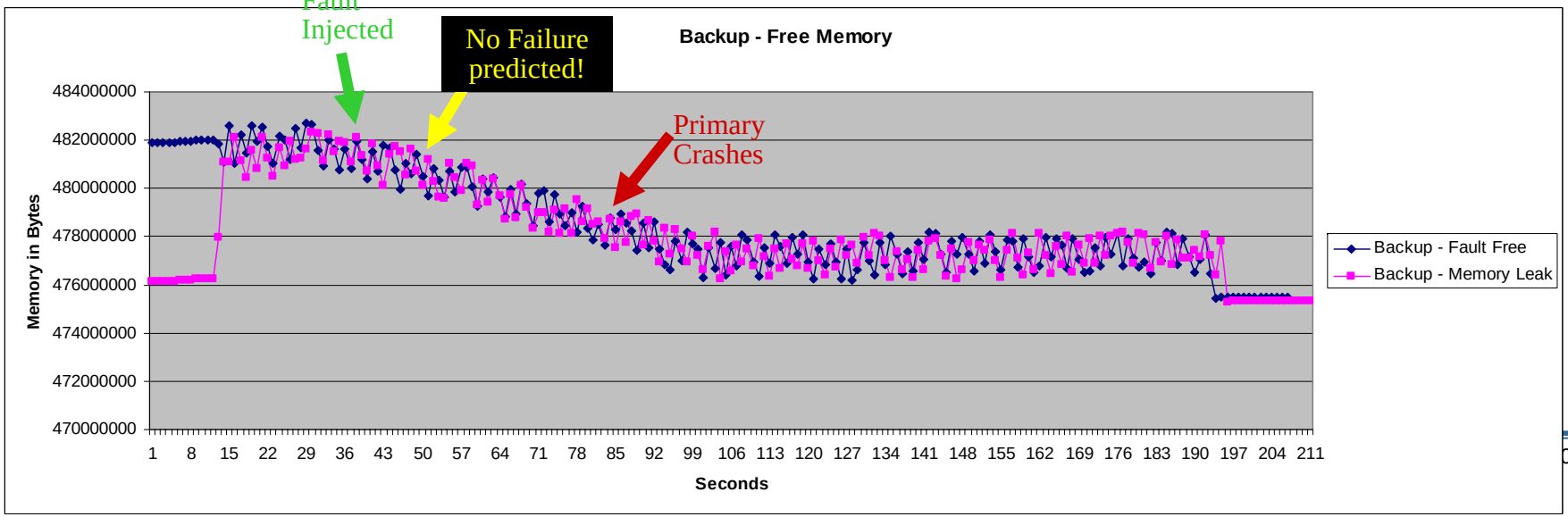
- Tracks requests along execution path
  - trace records logged (at ~200 strategic points)
  - < 5-6% overhead on performance and storage space
- Post-processed for data needed
  - per-workload, per-resource demands
  - per-request flow and latency maps
- Multiple uses
  - problem diagnosis, of course
  - also for guiding tuning (automatically or otherwise)

A bit more on approach

- Collect and orchestrate instrumentation
  - explicitly reported “events”
    - significant changes/hiccups observed by components
  - activity tracking
- Statistical tools (e.g., machine learning)
  - there are many, with different strengths/weaknesses
  - example usages
    - sample usage of resource X periodically and mine for anomalies in the time series data
    - categorize requests according to measured characteristics and watch for changes in (1) the set of categories or (2) the percentages of requests in each category

One “simple” example





Summary

- End goal: self-healing (autonomics; Self-* Storage)
  - automatically detect, diagnose, and repair problems
  - or, better, replace “detect” with “predict” and be proactive
- Least understood aspect: the “diagnose” step
  - need deep instrumentation
  - automatically identify failed components (to repair)
  - automatically identify root causes (to prevent recurrences)
- Some key sub-questions
  - trade-off between instrumentation detail and accuracy
  - pros/cons of different algorithms for different problems

For more information:
http://www.pdl.cmu.edu/

Greg.Ganger@cmu.edu

Director, Parallel Data Lab