A Fresh Look at the Reliability of Long-term Digital Storage

Mema Roussopoulos
Harvard University
November, 2006

The need for long-term digital storage

- Emerging web services
  - Email, photo sharing, web site archives
- Regulatory compliance and legal issues
  - Sarbanes-Oxley, HIPAA, intellectual property litigation
- Many other fixed-content repositories
  - Scientific data, intelligence info, libraries, movies, music
- Digital objects versus their analog counterparts

Why is long-term storage hard?

Long-term content suffers from more threats than short-term content

- Large-scale disaster
- Human error
- Media faults

- Component faults
- Economic faults
- Attack
- Organizational faults

- Media/hardware obsolescence
- Software/format obsolescence
- Lost context/metadata

Why is long-term storage hard?

- Large-scale disaster
- Human error
- Media faults

- Component faults
- Economic faults
- Attack
- Organizational faults

- Media/hardware obsolescence
- Software/format obsolescence
- Lost context/metadata

Why is long-term storage hard?

- Large-scale disaster
- Human error
- Media faults

- Component faults
- Economic faults
- Attack
- Organizational faults

- M ob
- S ob
- Lo

•Hardware: "Time Warner loses tapes" [Reuters, May 2005]

•Software: uninstalling required driver

•Infrastructure: turning off air conditioning system

Why is long-term storage hard?

- Large-scale disaster
- Human error
- Media faults

- Component faults
- Economic faults
- Attack
- Organizational faults

- M
ob
- Sc
ob
- Lo

•Sudden irrecoverable loss: DISK CRASH

•Bit rot: gradual accummulation of bit errors

• "100-year" CD myth

•Disk sectors gone wrong, firmware bugs, vibration…

Why is long-term storage hard?

- Large-scale disaster
- Human error
- Media faults

- Component faults
- Economic faults
- Attack
- Organizational faults

- M ob
- S ob
- L

•Hardware: power loss, fried controller card

•Software: disk firmware bugs affect data

•Network failures: ingestion of data may fail

•External license servers/companies

•Domain names vanish/reassigned

Why is long-term storage hard?

- Large-scale disaster
- Human error
- Media faults

- Component faults
- Economic faults
- Attack
- Organizational faults

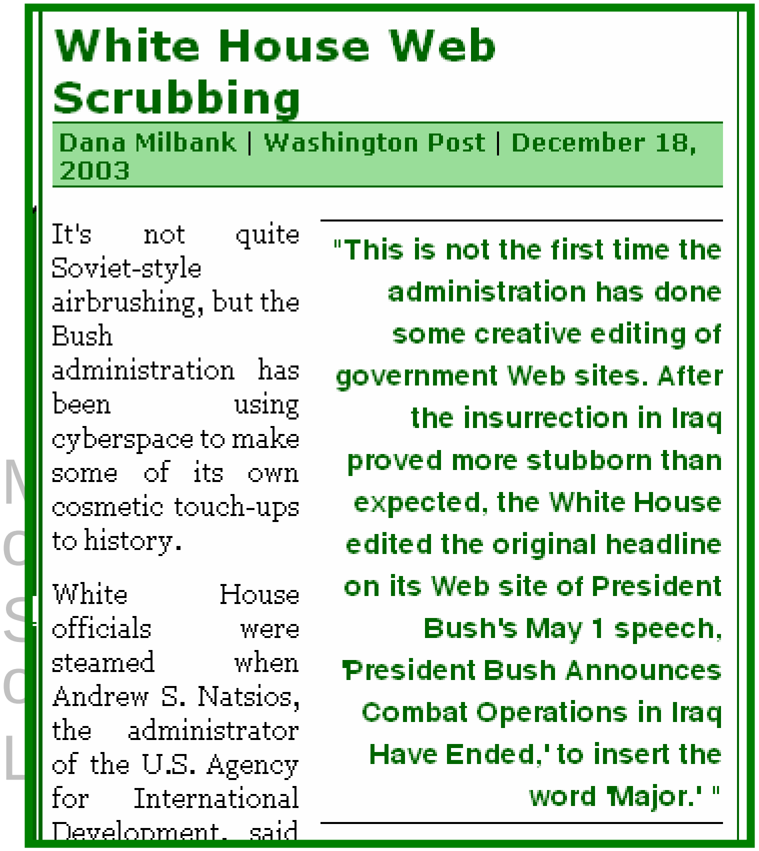
GOT MONEY????

- Software/format obsolescence
- Lost context/metadata

Why is long-term storage hard?

- Large-scale disaster
- Human error
- Media faults

- Component faults
- Economic faults
- Attack
- Organizational faults



White House Web Scrubbing

Dana Milbank | Washington Post | December 18, 2003

It's not quite Soviet-style airbrushing, but the Bush administration has been using cyberspace to make some of its own cosmetic touch-ups to history.

White House officials were steamed when Andrew S. Natsios, the administrator of the U.S. Agency for International Development, said

"This is not the first time the administration has done some creative editing of government Web sites. After the insurrection in Iraq proved more stubborn than expected, the White House edited the original headline on its Web site of President Bush's May 1 speech, 'President Bush Announces Combat Operations in Iraq Have Ended,' to insert the word 'Major.' "

Why is long-term storage hard?

- Large-scale disaster
- Human error
- Media faults

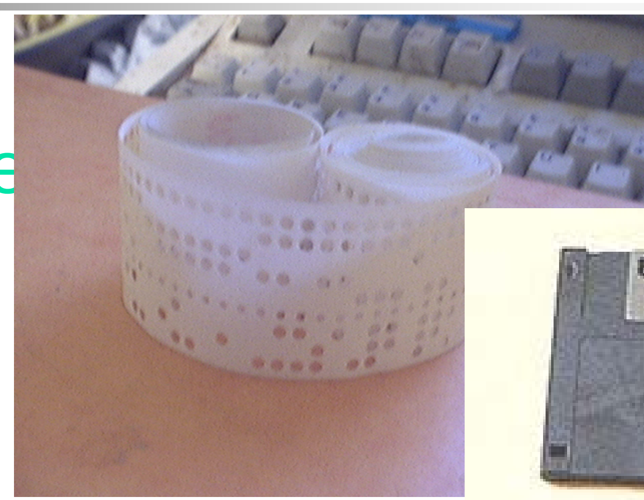
- Component faults
- Economic faults
- Attack
- Organizational faults

- M
  ob
- S
  ob
- Lost context/metadata

•Organizations come and go

•No data "exit strategies"

Why is long-term storage hard?



- Large-scale disaste
- Human error
- Media faults

- Component faults
- Economic faults
- Attack
- Organizational faults

- Media/hardware obsolescence
- Software/format obsolescence
- Lost context/metadata

What to do?

- Create lots of replicas of content to be preserved
- Increase probability that at least one replica will survive in long run
- Simple, intuitive, necessary but…
  - Not sufficient

Why is replication insufficient?

- Assumption of replica independence
  - e.g., a large-scale disaster wipes out all nearby replicas
  - Geographic dispersal not enough
  - Need administrative independence, component independence, etc.
- Assumption of fault visibility
  - Latent faults lurk subversively until data accessed
  - Archival workloads don’t access all data frequently
  - Accrue over time until too late to fix

Can we model long-term reliability?

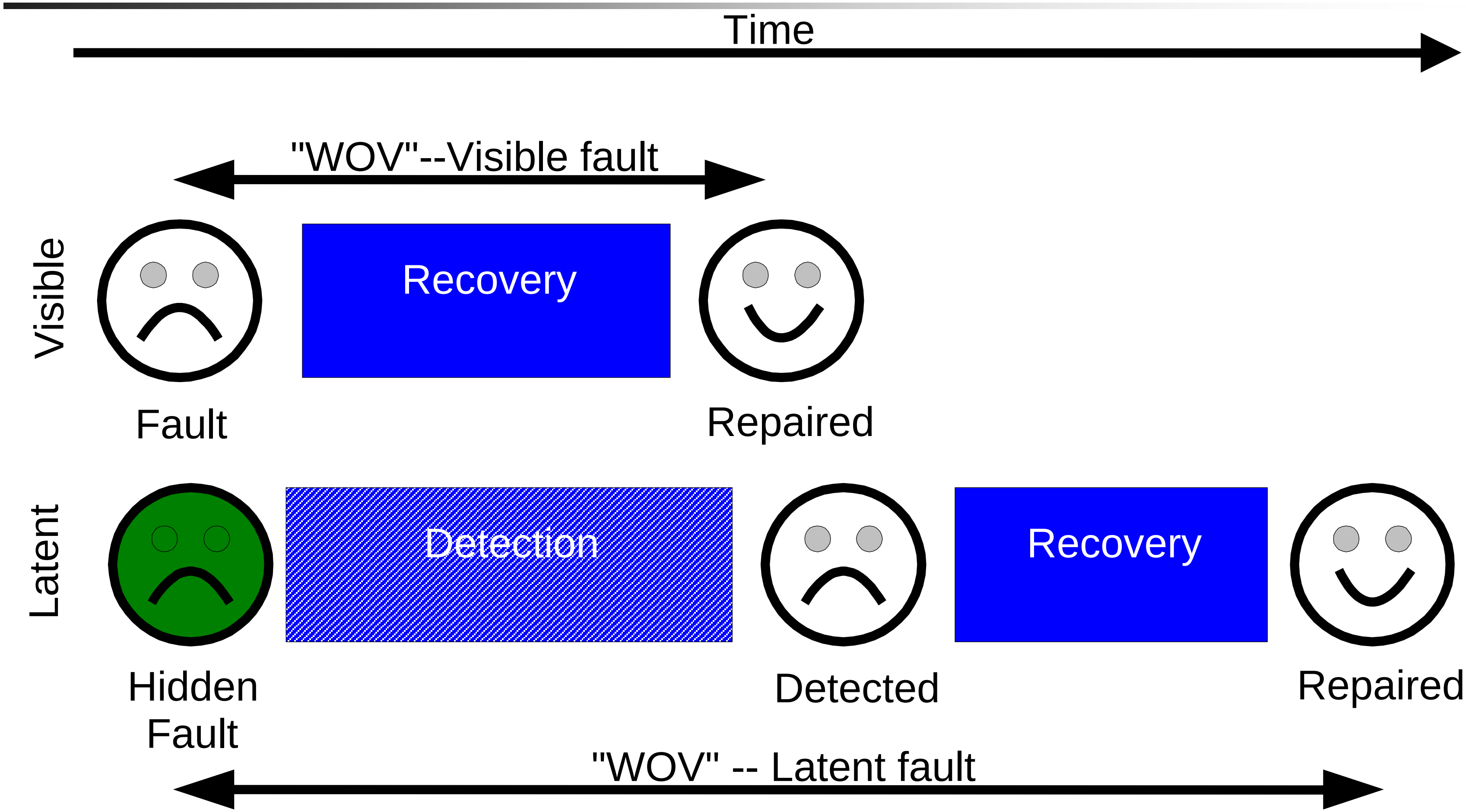
- Abstract reliability model for replicated data
  - Applies to all units of replication
  - Applies to many types of faults
- Extend RAID model
  - Account for latent as well as visible faults
  - Account for correlated faults: temporal and spatial
- Simple, coarse model
  - Suggest and compare strategies (choose trade-offs)
  - Point out areas where we need to gather data
- See EuroSys 2006 paper [Baker et al]

Our current approach

- Start with two replicas, then add more
- Derive MTTDL of mirrored data in the face of
  - Both immediately visible and latent faults
- Data loss occurs
  - If copy fails before initial fault can be repaired
- Time between fault and its repair is
  - *Window of Vulnerability* (WOV)

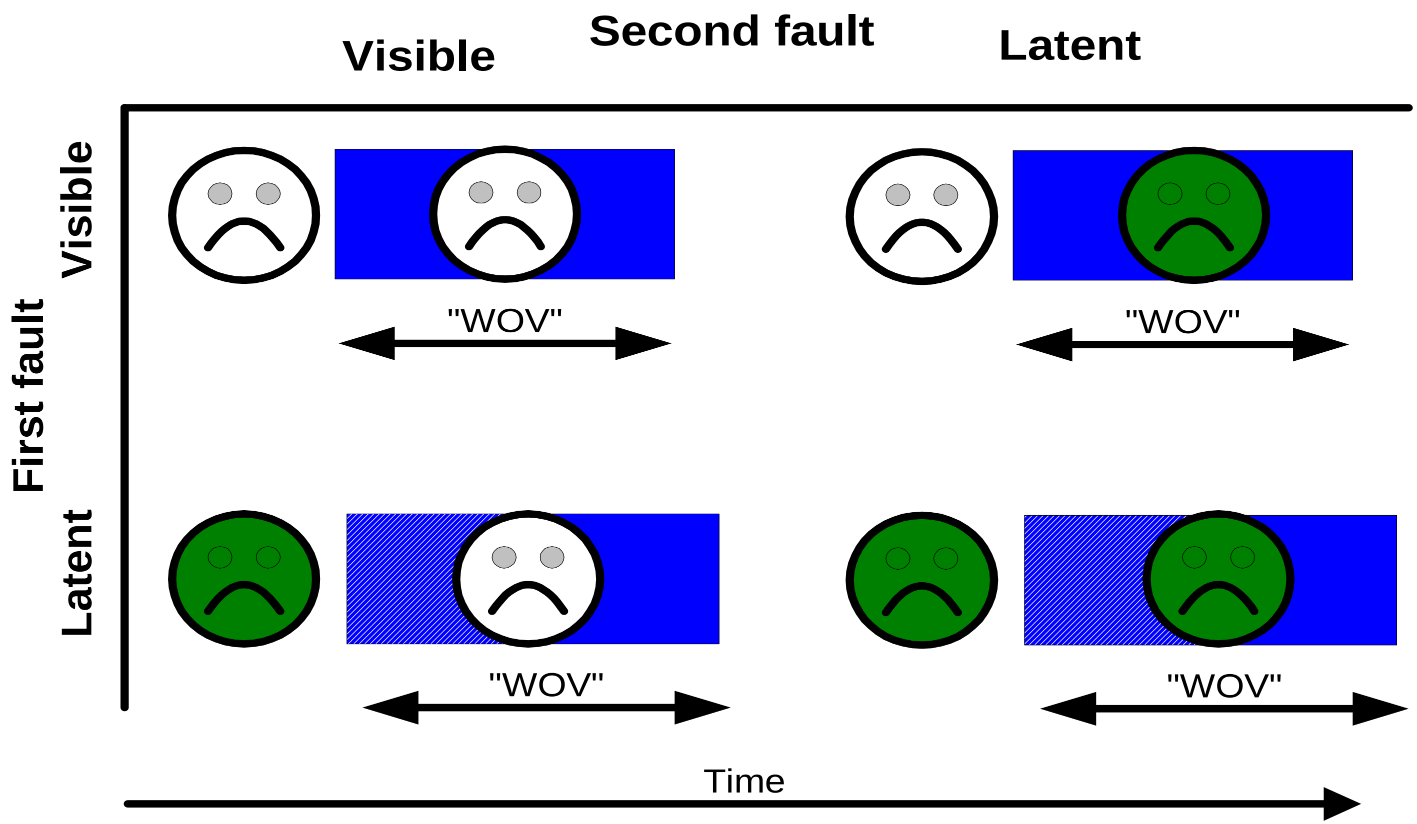
Window of vulnerability

Temporal overlap of faults



- Want detection time to be small

Data loss cases with 2 replicas

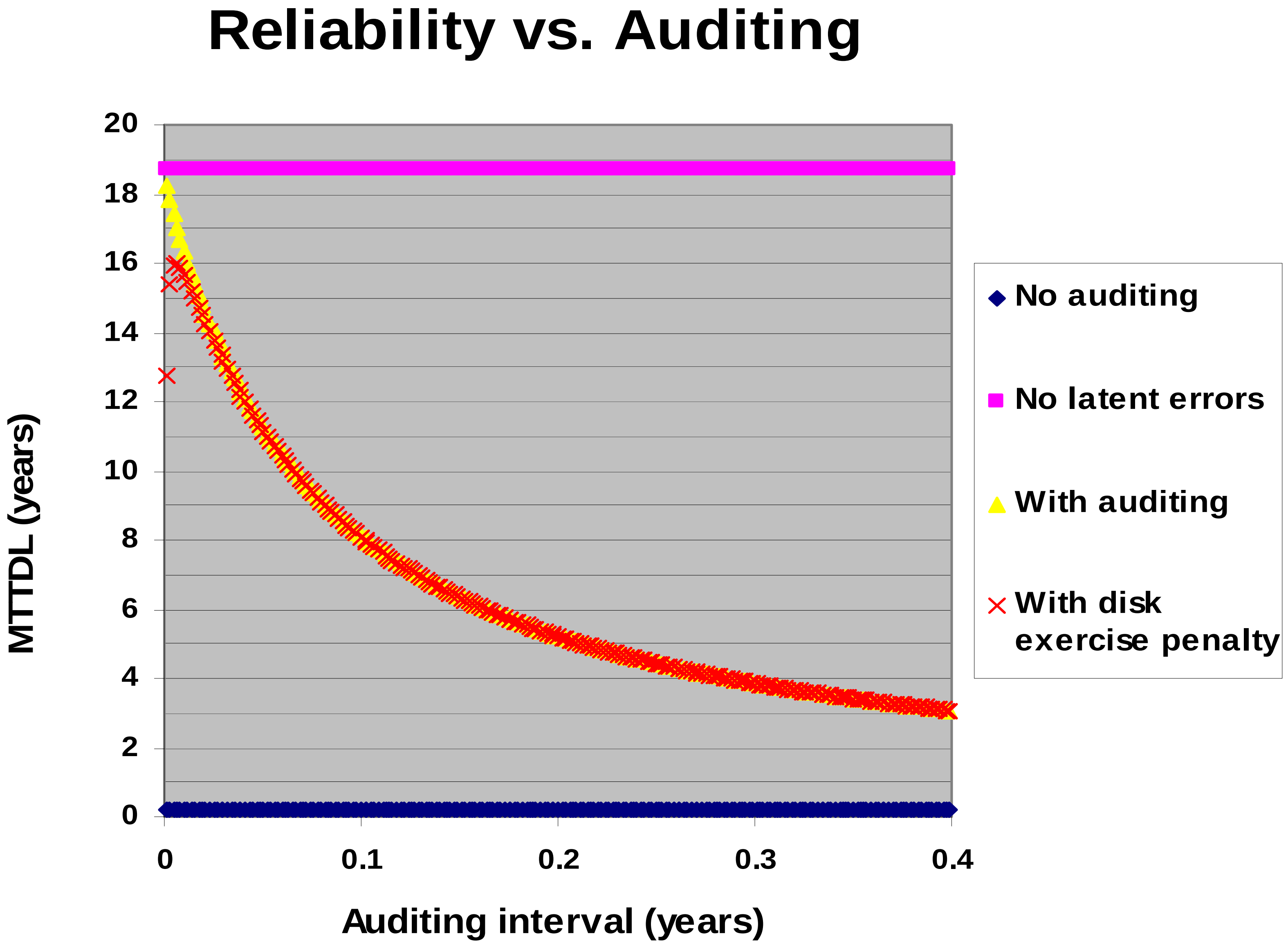


- Overall probability = sum of each case

Example using the model

- How much does it help to shorten detection time?
- Portion of real archive (www.archive.org)
  - Monthly snapshots of web pages
  - 1.5 million immutable files
  - 1795 200GB SATA drives, “JBOD”
  - Mean time to visible (disk) failure: 20 hours
  - Almost 3 years of monthly file checksums
  - Mean time to latent fault 1531 hours

Scenario: audited replicated archive



Dynamic long-term architecture

- Large time-scale => Failure is inevitable
- Independent replicas
  - Geographic, administrative, platform
  - Gains from extra replication offset by correlations
- Inexpensive audit of content
  - Fix latent faults at all levels before they accrue
  - *Content must be accessible to do this cheaply!!*
  - Backup to high-latency off-line media is not a solution
  - Includes "repairing" endangered content/metadata
- Keeping data static requires a dynamic system!