Can We Afford To Preserve Large Databases?

David S. H. Rosenthal

LOCKSS Program
Stanford University Libraries
http://www.lockss.org/

The Story So Far ...

- Threat Model
  - Broad range of poorly understood threats to data
- Petabyte for a century example:
  - Required performance beyond our ability to measure
- Now read on ...
  - What's different about large databases?
  - At $1M/replica, price-performance matters
  - Minimal number of replicas is an economic priority
- How well can we take decisions in this area?

Preservation *is* Fault Tolerance

- Bits *can* be copied perfectly
  - This doesn't mean they always *will* be copied perfectly
  - Perfect preservation is neither *guaranteed* nor *free*
  - In fact, at a large enough scale, it is *impossible*
  - How much loss can we tolerate?
- Everything that can possibly go wrong, will
  - How often will things go wrong?
  - How well will we tolerate things going wrong?
- We want better, more affordable preservation
  - Must *predict*, *measure* & *trade-off* cost and performance

Black Box Model

- Preservation system viewed as black box
  - Put bits in once
  - Get bits out repeatedly over time
  - Are the bits the same?
- Inside the box can be whatever you want
  - As many replicas, backups, ... as you want
  - Whatever audit and repair mechanisms you want
- Measure preservation *delivered to end user*
  - Who doesn't care about the replicas, backups, audits ...

Threat Model

- Media failure
- Hardware failure
- Software failure
- Network failure
- Obsolescence
- Natural Disaster
- Operator error
- External Attack
- Insider Attack
- Economic Failure
- Organization Failure

Rules of Thumb

- Safer data but higher cost from:
  - More replicas
    - BFT: 3*f*+1 replicas survive *f* simultaneous faults
  - More independent replicas
    - Less correlation between faults, therefore
    - Fewer simultaneous faults
  - More frequent audits of replicas
    - Some faults instantly visible, others *latent*
    - Shorter lifetime of latent faults, therefore
    - Lower probability of coinciding faults

How Safe Do We Need To Be?

- Keep a petabyte for a century
  - With 50% chance of remaining completely undamaged
- Consider each bit decaying independently
  - Analogy with radioactive decay
- That's a bit half-life of 10^{18} years
  - One hundred million times the age of the universe
- That's a rather demanding requirement
  - Hard to measure
  - Even *very* unlikely faults will matter a lot

How Likely Are The Threats?

Examples:

- Hardware
  - Schroeder 2007
  - Pinheiro 2007
- Software
  - Prabhakaran 2005
  - Yang 2006
- Operator Error
  - "Most important cause of data loss"
- Internal Attack
  - Secret Service report
  - Under-reported
- External Attack
  - Software mono-culture
  - Flash worm

Example: Disks

- Manufacturers specifications:
  - 10^6 hours MTTF
  - 10^{-14} unrecoverable bit error rate
- Schroeder & Pinheiro FAST '07 papers:
  - Field replacement rate 2-20 times the MTTF value
  - No "bathtub curve" of early failures
  - Enterprise disks 10x expensive, no more reliable
  - No correlation between temperature & failure
  - Significant autocorrelation – very bad for RAID
  - Significant long-rage correlation
  - SMART data logging not useful for failure prediction

Example: Software

File system code is carefully written & tested:

- Iron File System (Prabhakaran 2005):
  - Fault injection using pseudo-driver below file system
  - Bugs and inconsistencies in ext3, JFS, ReiserFS, NTFS
- FiSC (Yang 2006):
  - Model checking of file system code
  - 33 severe bugs in ext3, JFS, ReiserFS, XFS
  - Could destroy / in each file system
- Take away message:
  - The more you look, the more you find

Example: Insider Attack

- Political interference (Hansen 2007):
  - 2006 Earth Science budget *retroactively* reduced 20%
  - "One way to avoid bad news: stop the measurements!"
  - Suppose the data itself turned out to be "inconvenient" ...
    - Remove it (e.g. EPA pollution database)
    - Alter it?
- Independent replicas essential
  - Independently administered in different jurisdictions
  - Mutually audited so they're *tamper evident*

Realism

- Perfect preservation - not at any price
  - Threats too prevalent, diverse, poorly understood,
  - Real systems are inevitably imperfect
- How imperfect is adequate?
  - How much will it cost?
- How adequate is what we can afford now?
  - Won't know unless we can measure performance
- Kaizen: improve cost-performance thru time
  - Need preservation benchmarks to drive market
  - Learn from incidents c.f. NASA's ASRS

Benchmarking Preservation

- We need to benchmark a system we've built
  - Does it meet the 10^{18} year bit half-life target?
- We need to see about five bits flip
  - Watch a petabyte of data for 1000 years?
    - Too late to be useful
  - Watch an exabyte of data for a year?
    - Too expensive to be feasible
- Other ideas?
  - Fault injection?
  - Accelerated aging?

Suppose We Had Benchmarks

- Varying, uncertain *time value of money*
  - Postpone replication, but adds to risk
- Rapid, predictable decrease in *cost-per-byte*
  - Postpone replication, but adds to risk
- Rapid increase in *total demand* for storage
  - Replicate now, before competitors grab funding
- Varying, uncertain *future funding probability*
  - Repeated economic triage inevitable
- Endowment is the only safe mechanism

Service Level Agreements

- Create dataset, endow it, hand off to service:
  - Service level agreement to specify quality of preservation
  - Otherwise market captured by Potemkin services
- How to write the agreement?
  - How can we require performance we can't measure?
- How to audit compliance with agreement?
  - LOCKSS: mutual audit protocols *between replicas*
  - Other ideas? Audit preservation *delivered to end user*?

Transfer Of Custody

- Liability Disclaimers are endemic:
  - AMAZON DOES NOT WARRANT THAT AMAZON WEB SERVICES ... WILL BE ACCESSIBLE ON A PERMANENT BASIS OR WITHOUT INTERRUPTION OR THAT THE DATA YOU STORE IN ANY SERVICE ACCOUNT WILL NOT BE LOST OR DAMAGED
- Liability Disclaimers are viral:
  - You can't accept liability for your suppliers' products
  - Disclaiming lowers your competitors' costs
- Transfer of custody without liability:
  - Can it be meaningful?

Fundamental Problem

- We specify system performance levels
  - E.g. HIPPA
- That we don't know how to measure
  - Against threats we know we don't understand well
- But we assume will be met
  - So we don't plan for not meeting them

Research Agenda

- Better data on incidence of threats
  - disk behavior, bugs, operator errors, attacks, ...
- Better algorithms & architectures
  - a "better than BFT" model?
  - "better than TPM" hardware support for preservation?
  - highly independent replica architectures?
- Better cost-performance models
  - Define, measure "performance" of preservation systems?
  - Taking decisions with dynamic costs & performances?
  - Transferring custody of data vs. liability disclaimers?