SafeStore: A Durable and Practical Storage System

Ramakrishna Kotla

Lorenzo Alvisi, Mike Dahlin
University of Texas at Austin
Digital Data: High durability requirements

How can we store data safely for multiple decades?

 Applications: Photos, health, finance, digital library

 Cost of data loss: Invaluable, prosecution

- Can you afford to lose your family photograph?
Multi-decade scale durable storage

Flash memory

CD/DVD

Raid + Tape

Disk

Storage Service Provider

Is this a solved problem?
Challenge: Broad range of threats

- Hardware failures
  - Google loses 5-10% of disks every year [FAST’07]

- Software vulnerabilities
  - S/W vulnerabilities increased by 8x [CERT, 2006]
  - Ransomware rising [Washington post, May, 2006]

- Operator errors
  - Operator errors are the hardest to mask [USITS’03]
Challenge: Uncommon faults must be considered

20% of attacks are committed by insiders - [CERT, 2006]

Insider attacks

Fire destroys top CS research center - [BBC, 31 October 2005]

Disasters

Photopoint.com shuts down without notifying 1.5 million customers [CNET, 2006]

Organizational failures
SafeStore: A durable and practical storage system

- **Aggressive fault isolation**
  - Physical, geographical, admin faults: autonomous SSPs
  - User faults: Restricted storage interface

- **Outsourced SSPs: Exploit economies of scale**
  - Amazon S3: $0.15/GB/month (storage), $0.20/GB (network)

- **Challenge: Limited or no control over SSPs**
SafeStore: Outline

- SafeStore architecture: Aggressive fault isolation
- **SafeStore: Rethink storage mechanisms**
  - Replication mechanism
  - Audit mechanism
- Evaluation
Data replication: Challenges

- **Heterogeneous SSPs**
  - Replication mechanism, number of nodes, MTTF, MTTR

- **Failures**
  - Correlated: Operator errors, catastrophes, bankruptcy
  - Uncorrelated: Sector failures

- **How do we replicate data to maximize MTTDL?**
Hierarchical data encoding

- Intra-ssp and inter-ssp redundancy
  - Inter-ssp redundancy: correlated SSP failures
  - Intra-ssp redundancy: uncorrelated failures within SSP
  - Replication: Full replication, erasure coding, RAID 5

- Encoding: inter-ssp: (3,2), intra-ssp: (3,2)
  - Overall storage overhead factor: 9/4
**Ideal: Optimal inter-ssp/intra-ssp encoding**

- What is the right inter-ssp and intra-ssp encoding?
- Setup: 3 SSPs with 8 nodes each
  - Uncorrelated: 10 years/1 day; Correlated: 100 years/10 days

**Black-box SSP interface:** Number of nodes, replication mechanism, MTTF, MTTR are not exposed

Storage overhead: 4

Available storage overhead

Probability of loss in 10 years

- $10^{-08}$
- $10^{-07}$
- $10^{-06}$
- $10^{-05}$
- $10^{-04}$
- $10^{-03}$
- 0.01
- 0.1
- 1

- inter-ssp: (3,1), intra-ssp: (8,6)
Informed hierarchical encoding interface

- SSPs expose limited information: redundancy
  - Example: 1, 1.3, 4, 8 for (8,8), (8,6), (8,2), (8,1) encoding
  - Need not expose number of nodes, replication mechanism, MTTDL, MTTR
- Data owner can choose both inter-ssp and intra-ssp redundancies
- How do we optimally distribute overall redundancy between inter-ssp and intra-ssp redundancies?
  - Robust to non-uniform distribution of nodes across SSPs
  - Robust to skew in MTTDL, MTTR within SSPs
Informed hierarchical encoding

- **Heuristics: Close to optimal durability**
  - Maximize inter-ssp redundancy by choosing minimal intra-ssp redundancies (greater than 1)
  - Distribute remaining redundancy uniformly across intra-SSP redundancies (minimize std. dev.)

- **Intuition**
  - Substantial improvement with increasing inter-ssp redundancy
  - Diminishing returns with increasing intra-ssp redundancy
Informed hierarchical encoding: Intuition

- Substantial improvement with increasing inter-ssp
- Diminishing returns with increasing intra-ssp
Informed hierarchical encoding

- Close to optimal durability

- Setup: 3 SSPs with 8 nodes each
  - Correlated: MTTF of 100 years; MTTR of 10 days
  - Uncorrelated: MTTF of 10, 5, 3 years; MTTR of 2, 3, 5 days
SafeStore: Architecture
SafeStore: Rethinking storage mechanisms
- Replication mechanism: Informed hierarchical encoding
- Audit mechanism
Evaluation
Why do we need to audit data?

- **Autonomous SSPs: No/limited control**
  - SSPs lose data: latent faults, operator errors, cut corners
  - Examples: Google and hotmail lost data

- **Rethinking the contract: Trust but verify**
  - Need for end-to-end audit to quickly detect data losses
  - Reduce MTTR
**SSP Fault model**

- Honest and active SSPs: Proactively verify local data and notifies data losses
- Honest and passive SSPs: Verify only when asked
  - Overloaded, cut corners, operator errors
- Dishonest SSPs: Can lie if data is lost
  - Software bugs, malicious
**Simple audit mechanism**

- **Auditee**: Read data, compute hash, compare with original hash
- **SSPs can send dishonest answers**
  - S/W bugs, operator errors, cut corners, malicious
- **Not verifiable**: Does not work with dishonest SSPs
Inefficient end-to-end audit mechanisms

- Storage intensive: data stored at the auditor
- Other inefficient mechanisms
  - Network intensive: data read from auditee
  - Computationally intensive: zero knowledge proofs
  - Voting based mechanisms may not work

SSP(Auditee)

Data: \{d\}
Is \( h_t(d) = h(d) \)?

Auditor

Nonce: C
Instance: t

Challenge: C, t
Compute \( h(C+d) \)?

Response

\( \{h(C+d)/\text{Fail}\}_{K_{SSP}} \)

Verify \( h(C+d) \)?
SafeStore: Audit protocol

- **Regular audit**: For all data objects
  - Detect data losses with honest/passive SSPs
  - Forces SSPs to commit to non-repudiable assertion

- **Spot check**: Random subset of objects
  - Verify responses in regular audit to catch dishonest SSPs
Audit: Separate commitment from verification

SSP

Honest    Dishonest

Active    Passive

- **Cost effective**
  - Regular audit: Transfer hash responses for all objects
  - Spot check: Transfer data for a small subset of objects

- **Mean time to detect data loss**
  - Regular audit: Detect data losses quickly at passive SSPs
  - Spot check: Detect data losses at dishonest SSPs
SafeStore Audit: Effective end-end audit

- Remove incentives for SSPs to cut corners
  - Dishonest SSPs: Verifiable proof of misbehavior
  - Service level agreement

- SLA incentives: Rational SSPs behave honestly
  - Rational SSPs try to maximize their profit
  - Storage cost << Penalty for data loss (honest SSP)
  - Penalty for data loss (honest SSP) << Prob. of catching dishonest SSP X penalty for data loss (dishonest SSP)
Audit protocol

- Dishonest SSPs may not respond to spot check
  - Read data from other SSPs, data owner
- Spot check all objects
  - Local auditor: Read from co-located data owner
  - Stateful auditor: Stores pre-computed hashes
Data durability: Passive SSPs

- Durability with audit is close to the system with active SSPs
  - With 20% additional cost, MTTD < 10 days for 1TB of data
  - Cost: Network, CPU, IO

MTTD = 20 days
MTTD = 10 days
MTTD = 2 days (Active SSPs)
Data durability: Dishonest SSPs

- Audit provides 2 9’s of better durability with 20% additional cost
  - Local auditor: spot check(100%); Remote: spot check(1%)
Outline

- SafeStore: High durability architecture
- SafeStore: Rethinking storage mechanisms
  - Replication mechanism: Informed hierarchical encoding
  - End-end audit mechanism
- Evaluation
SSFS: A Durable Storage System prototype

- SSFS storage system
  - NFS 2.0, snapshot based versioned file system, recovery
- Remote storage i/f
  - Temporal isolation: Write once read many
  - Incremental block level updates
  - Fast recovery optimization: on-demand transfers
- Security: AES, rabin-key signature, SHA1
**End-end performance: Postmark benchmark**

- SSFS is within 10% of unreplicated NFS
- Within 40% for snapshot intensive workload
Is it economically viable?

- **Total cost of operation**
  - Storage, network, administrative, operational, profit
- **SafeStore: Amazon pricing model**
  - Storage: $150/TB/month
  - Network: $200/TB
  - Includes all costs and profit
- **Local storage: Unreliable storage**
  - Storage: $30/TB/month - Internet archive
  - Administrative: Jim Gray[Queue’03]
    - 1 Admin/10TB (Typical), 1 Admin/100 TB (Efficient)
  - No administrative, geographical, software diversity
**Economic viability**

- **Outsourced storage is economically attractive**
  - SafeStore’s cost is 2x better than typical unreliable local storage system’s cost
Conclusion

- **SafeStore**: Highly durable storage system
  - Aggressive fault isolation by outsourcing storage
  - Informed hierarchical encoding
  - Efficient end-end audit mechanism

- **Future work**
  - Security primitives have limited lifetimes


- [http://www.cs.utexas.edu/~kotla/SafeStore](http://www.cs.utexas.edu/~kotla/SafeStore)
BACKUP SLIDES
Informed hierarchical encoding

- Close to optimal durability
- Same setup: 3 SSPs, 8 nodes each
Non-uniform distribution of nodes

- Robust to non-uniform distribution of nodes
- Setup: 3 SSPs with 14, 7, and 3 nodes each
**Data durability: Passive SSPs**

- Durability of the system with audit is close to the system with active SSPs
  - With 20% additional cost, MTTD < 10 days for 1TB of data
  - **Cost: Network($0.20/GB), CPU, IO**
Passive SSPs: MTTD data loss

- MTTD data loss falls rapidly with audit cost
- Remote auditor with spot check of 1%
  - MTTD data loss of 8 days with 20% additional cost
Audit cost model

- Computation: $0.031 per million crypto ops
- Storage: $0.027 per million IO operations
- Network: $200/TB of data transfer
Data replication: Challenges

- Autonomous SSPs
  - Heterogeneity: Replication mechanism, MTTF, MTTR
  - Limited or no control on data management policies

- Failures
  - Correlated: Operator errors, catastrophes, organizational
  - Uncorrelated: Sector failures

- How do we efficiently replicate data across SSPs?
  - High mean time to data loss (MTTDL)
Black box SSP interface

- Autonomous SSPs expose minimal information
  - No information on replication mechanism, number of nodes, MTTF, MTTR
  - Intra-ssp encoding is fixed and unknown
  - Amazon S3 just exposes the storage cost

- Oblivious hierarchical data encoding
  - User controls only inter-ssp encoding

- Which one provides better durability using 3 SSPs with 8 nodes each?
  - inter-ssp: (3,2) and intra-ssp: (8,2)
  - inter-ssp: (3,1) and intra-ssp: (8,4)
Oblivious hierarchical encoding: Sub-optimal

- Oblivious hierarchical encoding
  - User can control only inter-ssp redundancy
  - intra-ssp redundancy is fixed

- Sub-optimal durability: Off by several 9’s

What is the right SSP storage interface?

- Ideal
- Oblivious: (8,2)
- Oblivious: (8,4)
- Oblivious: (8,8)

Probability of data loss in 10 years vs. Available storage overhead
Audit optimizations

- Voting based mechanisms may not always work
  - Data is erasure coded across SSPs
- Auditor may pre-compute $h(ch+d)$ for future challenges
- Data owner may store $h(ch+d)$ at auditor
  - Additional management cost
  - Auditor loses data and data owner is not online?
Other results

- **IOZONE: File system micro-benchmarks**
  - Read/write latency and throughput
  - Within 12% of unreplicated NFS

- **Low storage overhead**
  - Efficient block level versioning
  - Increases slowly with updates
Future work

- Security primitives have limited lifetimes
- Interpreting data over long durations