Preservation in the Cloud

David S. H. Rosenthal
LOCKSS Program
Stanford University Libraries

http://www.lockss.org/
http://blog.dshr.org/

LOTS OF COPIES KEEP STUFF SAFE

Cloud Storage

- The Good News
  - Cheap, flexible, easy-to-use, available, reliable
  - Competitive marketplace of providers
- The Bad News (S3's version, others similar)
  - AMAZON ... SERVICES ... ARE PROVIDED “AS IS”
  - WE ... DO NOT WARRANT THAT ... THE DATA YOU STORE WILL BE SECURE OR NOT ... LOST OR DAMAGED.
- How can we leverage cloud storage
  - Taking advantage of the economics
  - Without trusting a service that disclaims all liability?

Preservation as a Cloud Biz

- Provider expects download >> upload
  - Margins: ~100% on download, ~33% on upload
- Preservation: download << upload
  - Preserved content access density very low
- Preservation is cost-effective cloud use
  - Like buying the supermarket loss-leader
- Preservation is a small niche cloud use
  - Otherwise providers will change pricing model
- Cloud technology won't target preservation
  - Will not deliver preservation-level bit reliability

Availability vs. Reliability

- Availability:
  - What proportion of requests get an answer
  - S3 refunds you if they don't make 99.9%
- Reliability:
  - What proportion of requests get the right answer
  - S3 says that's your problem
- Preservation needs extreme reliability
  - CERN study: 99.9999999% of bits OK after 6 months
  - Petabyte for a century needs 99.9999999999999%

Multiple Replicas in the Cloud

- Each copy in cloud will be unreliable
  - Need copies in multiple storage providers
  - Need to detect and repair damage to each copy
  - Overall reliability depends on time from damage to repair
- Audit 3 copies of 10TB 8 times per year
  - Storage costs $4500/mo (Amazon pricing)
  - Audit by extract from cloud & hash $3400/mo
- Audit in provider's compute service
  - No charge for data transfer, so much cheaper
  - But, can't trust provider – incentive to cover up failure

Audit vs. Stored Hashes

- Auditor stores hashes (e.g. Song & JaJa '07)
  - Auditor initially gets content, hashes it, remembers hash
  - Regularly asks provider to hash content, report result
  - Compares reported hash to stored hash
- Auditor trusts provider
  - Provider could get content, hash it once, remember hash
  - Report remembered hash every time, no failures ever
- Auditor has to be in ingest pipeline
  - Hard to be a true third party

Audit vs. Stored Challenges

- Auditor stores challenges (Shah et al., 2007)
  - Auditor gets content, chooses N random nonces
  - Computes, stores N pairs: nonce,hash(nonce,content)
  - N-1 audits: send nonce, get hash(nonce,content)
  - Then get content, validate hash(nonce,content), repeat
- Auditor doesn't trust provider
  - Provider has content now if hash(nonce,content) correct
- Auditor has to be in ingest pipeline
  - Hard to be a true third party

Mutual Audit

- Auditor manages mutual audit (cf. LOCKSS)
  - Auditor sends nonce1 to provider
  - Provider replies nonce2,hash(nonce1,nonce2,content)
  - Auditor sends each vote to other providers to check
    - Plus spurious votes to detect fraud
- Auditor trusts majority of providers
  - Providers judged by “jury of peers”
- Auditor not in ingest pipeline
  - True 3rd party audit, never sees content being audited

Conclusions

- Preservation in the cloud requires:
  - Greater reliability than providers will offer, thus requires
  - Replicas in diverse providers, thus requires
  - Audit & repair between replicas
- Audit of cloud replicas requires:
  - 3rd party auditing that does not trust cloud provider,
  - But takes place in the cloud environment
    - Auditing outside the cloud is too expensive
- No perfect solution available
  - LOCKSS protocol closest to meeting all requirements