Geo-distribution in Storage

-Jason Croft and Anjali Sridhar

Outline

- Introduction
- Smoke and Mirrors
- RACS – Redundant Array of Cloud Storage
- Conclusion

Introduction

Why do we need geo-distribution?

- Protection against data loss
- Options for data recovery

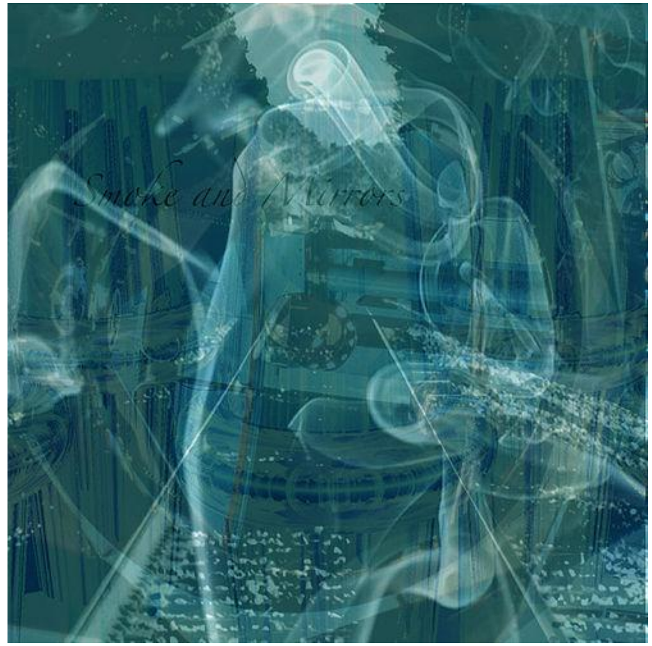
Cost ?

- Physical
- Latency
- Manpower
- Power
- Redundancy/Replication

How to Minimize Cost ?

- Smoke and Mirror File System
  – Latency
- RACS
  – Monetary cost
- Volley
  – Latency and Monetary cost

Applications?



Smoke and Mirrors: Reflecting Files at a Geographically Remote Location Without Loss of Performance

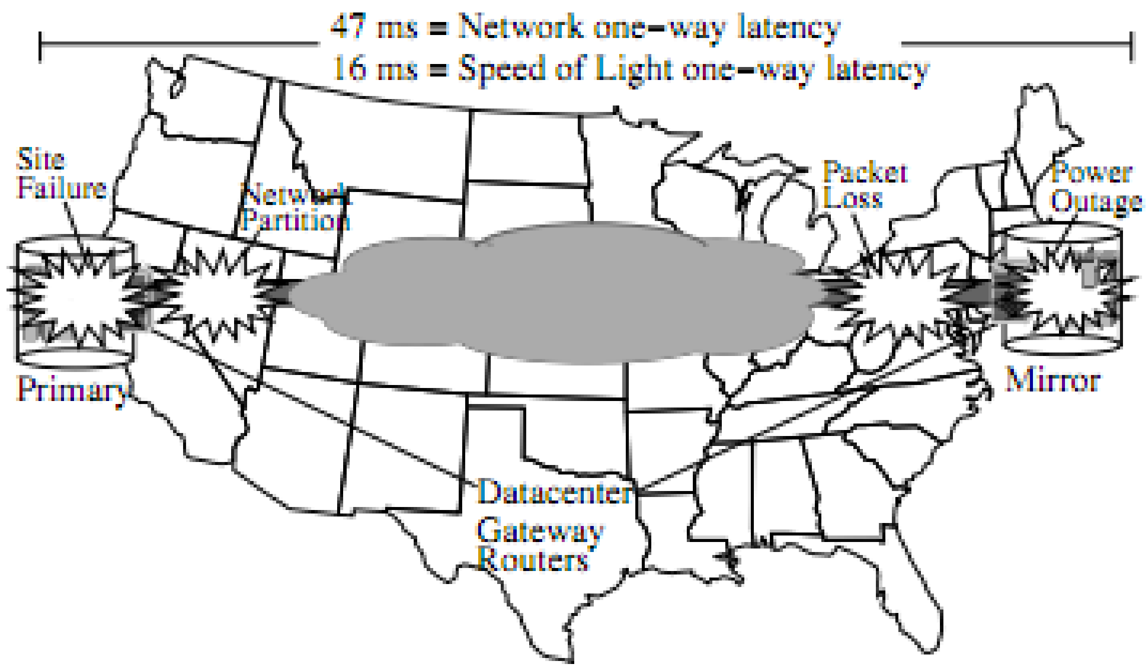
-Hakim Weatherspoon, Lakshmi Ganesh, Tudor Marian, Mahesh Balakrishnan, and Ken Birman,

Cornell University, Computer Science Department & Microsoft Research, Silicon Valley ,FAST 2009

Smoke and Mirrors

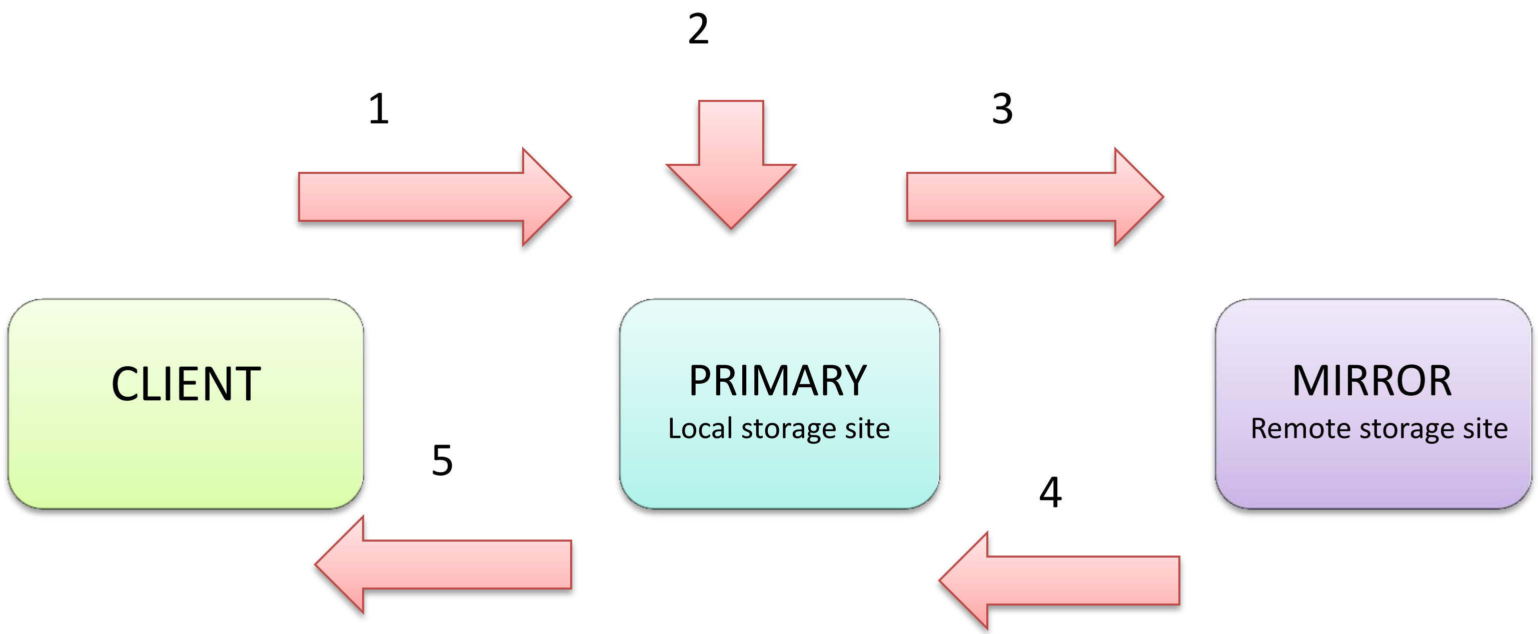
- Network sync tries to provide reliable transmission of data from the primary to the replicas with minimum latency
- Sensitive to high latency but require fault tolerance
- US Treasury, Finance Sector Technology Consortium and any corporation using transactional databases

Failure – Sequence or Rolling disaster



The model assumes wide area optical link networks with high data rates which has sporadic , bursty packet loss . Experiments are based on observation of TeraGrid, a scientific data network linking supercomputers.

Synchronous



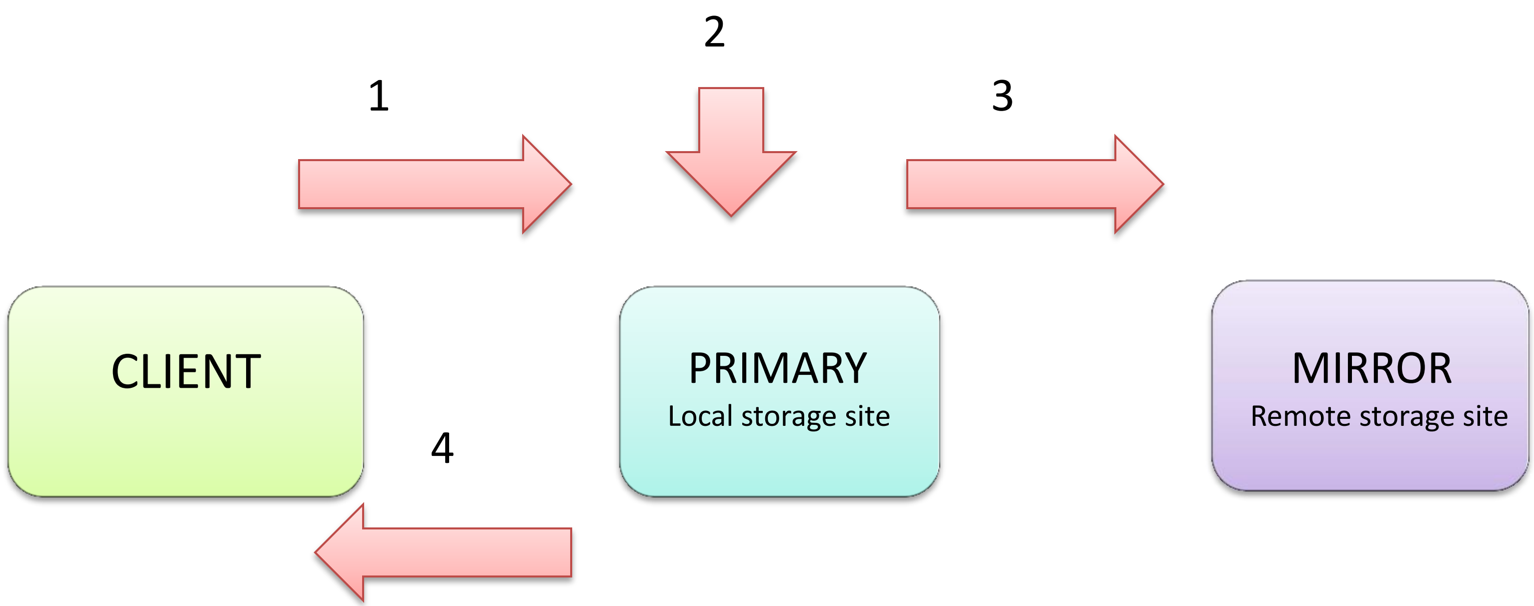
Disadvantage

- Low performance due to latency

Advantage

- High reliability

Asynchronous



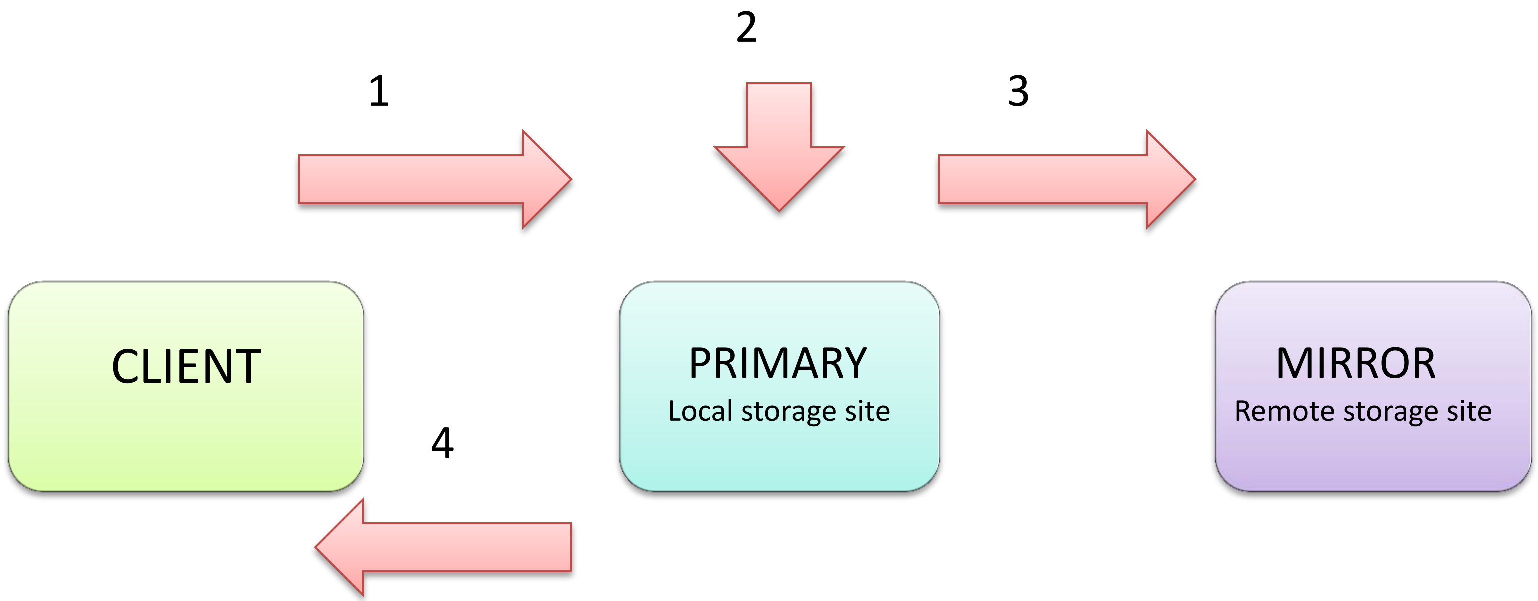
Advantage

- High performance due to low latency

Disadvantage

-Low reliability

Semi-synchronous



Advantage

-Better reliability than asynchronous

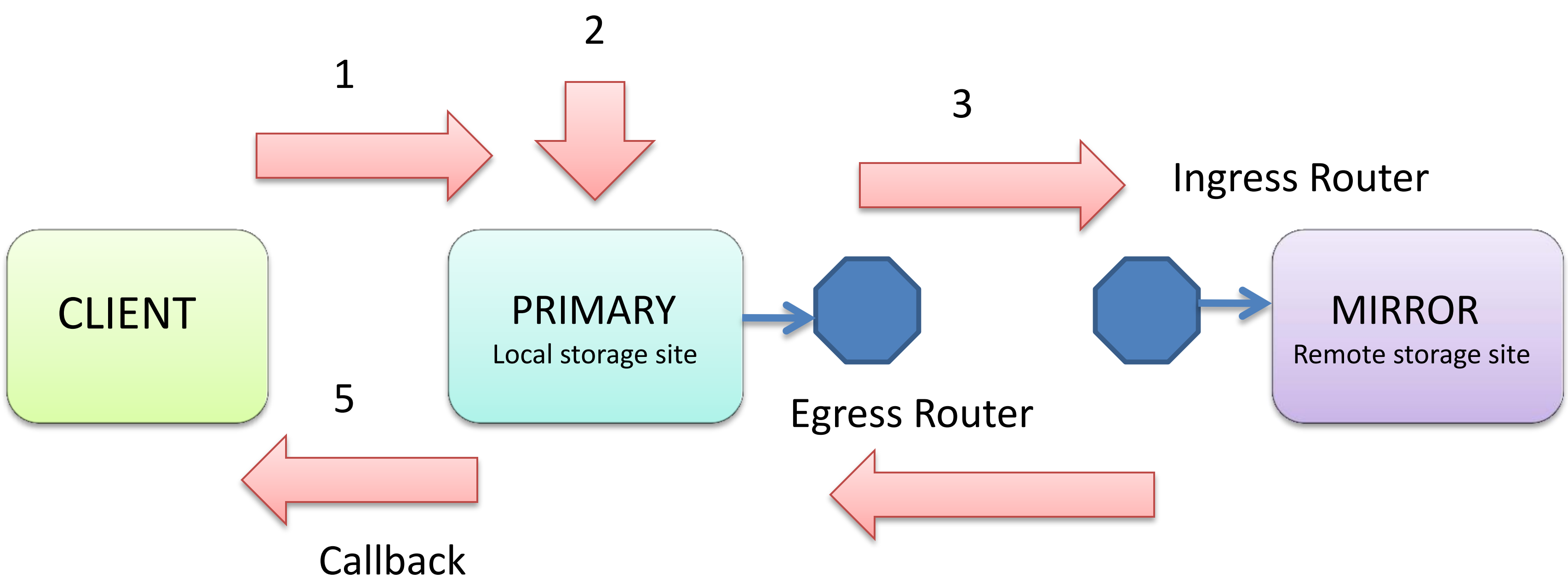
Disadvantage

- More latency than synchronous

Core Ideas

- Network Sync is close to the semi-synchronous model
- It uses egress and ingress routers to increase reliability
- The data packets along with forward error correcting packets are “stored” in the network after which an ack is sent to the client
- A better bet for applications

Network Sync



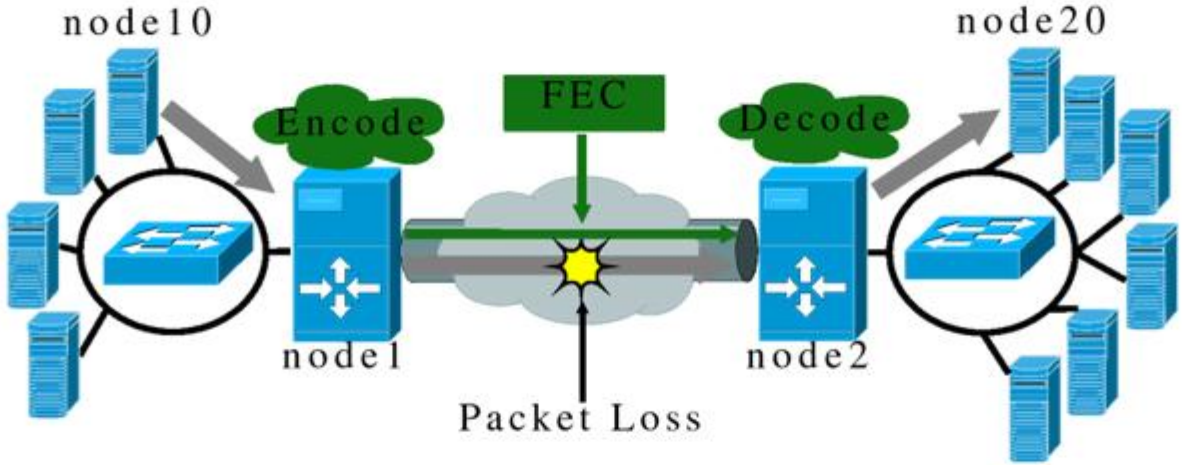
Ingress and Egress Routers are gateway routers that form the boundary between the datacenter and the wide area network.

FEC protocol

- (r,c) – r packets of data + c packets of error correction
- Example - Hamming codes (7, 4)

Bit #	1	2	3	4	5	6	7
Transmitted bit	p_1	p_2	d_1	p_3	d_2	d_3	d_4
p_1	Yes	No	Yes	No	Yes	No	Yes
p_2	No	Yes	Yes	No	No	Yes	Yes
p_3	No	No	No	Yes	Yes	Yes	Yes

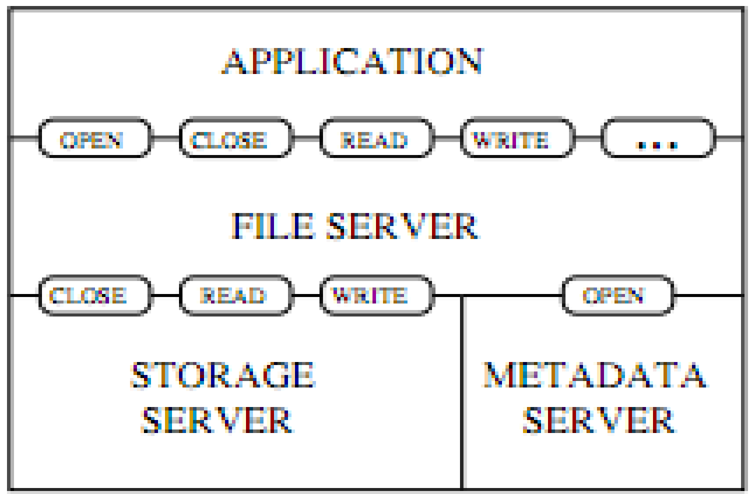
Maelstrom



http://fireless.cs.cornell.edu/~tudorm/maelstrom/

- Maelstrom is a symmetric network appliance between the data center and the wide area network
- It uses a FEC coding technique called layered interleaving designed for long haul links with bursty loss patters
- Maelstrom issues callbacks after transmitting a FEC packet

SMFS Architecture



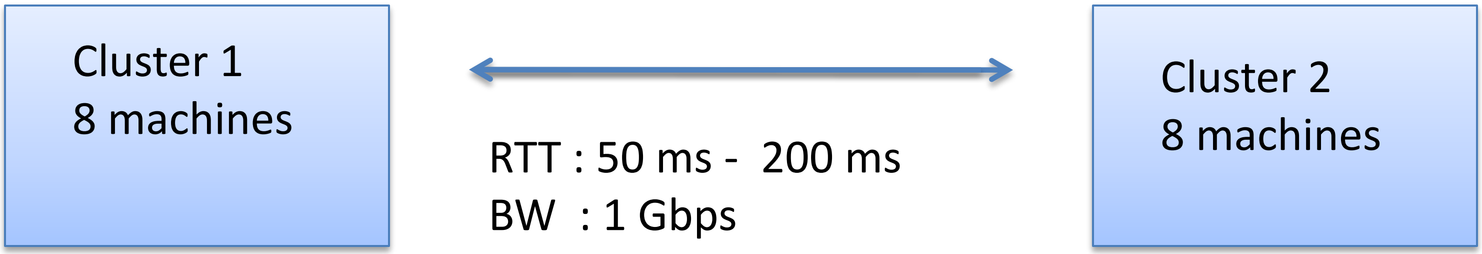
- SMFS implements a distributed log structured file system
- Why is log-structured file system ideal for mirroring?
- SMFS API - create(), append(), read(), free()

Experimental Setup

- Evaluation metrics
  - Data Loss
  - Latency
  - Throughput

- Configurations
  - Local Sync (semi-synchronous)
  - Remote Sync (synchronous)
  - Network Sync
  - Local Sync + FEC
  - Remote Sync + FEC

Experimental Setup 1 - Emulab



RTT : 50 ms - 200 ms
BW : 1 Gbps
(r,c) : (8,3)
Duration: 3mins
Message size: 4KB
Users: 64 testers
Num of runs: 5

Data Loss

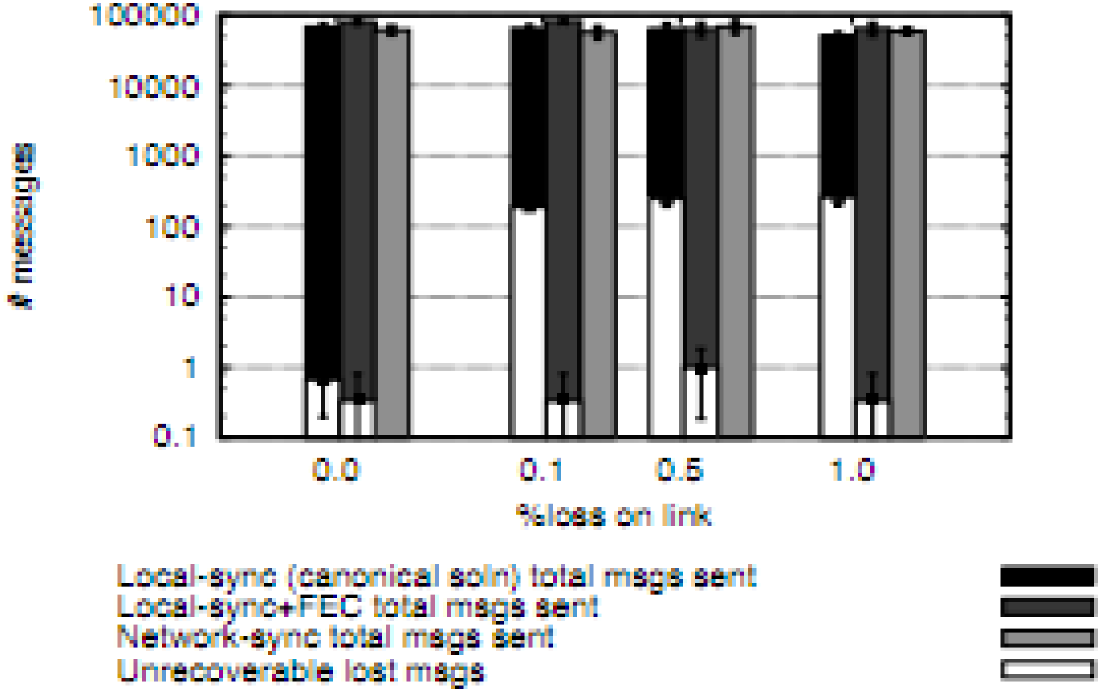


Figure 6: Data loss as a result of disaster and wide-area link failure, varying link loss (50ms one-way latency and FEC params $(r, c) = (8, 3)$).

Data Loss

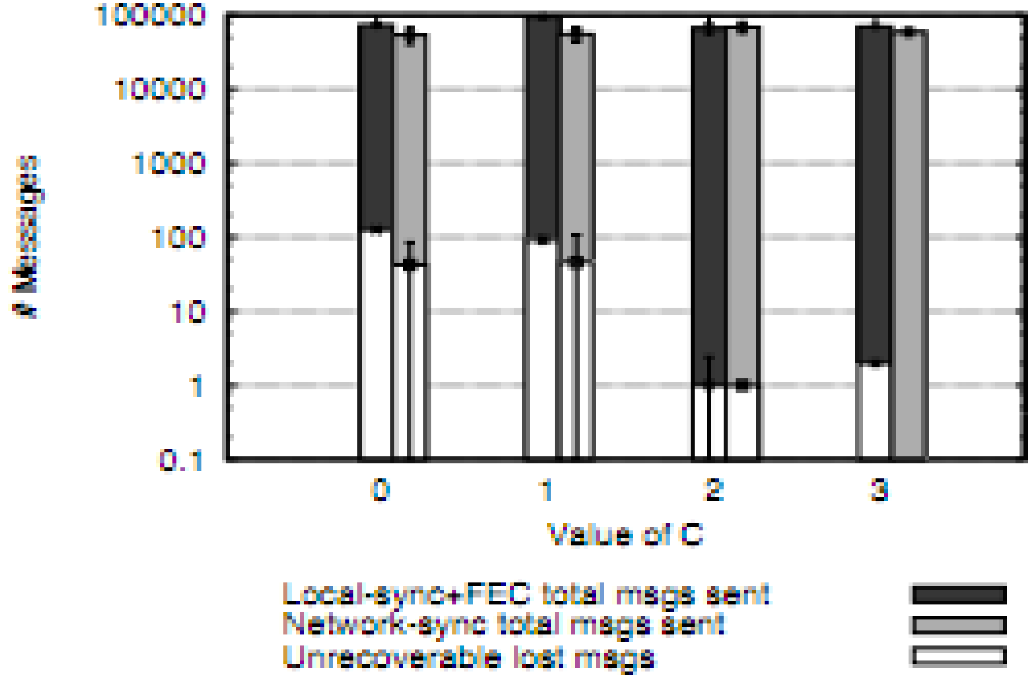


Figure 7: Data loss as a result of disaster and wide-area link failure, varying FEC param c (50ms one-way latency, 1% link loss).

Latency

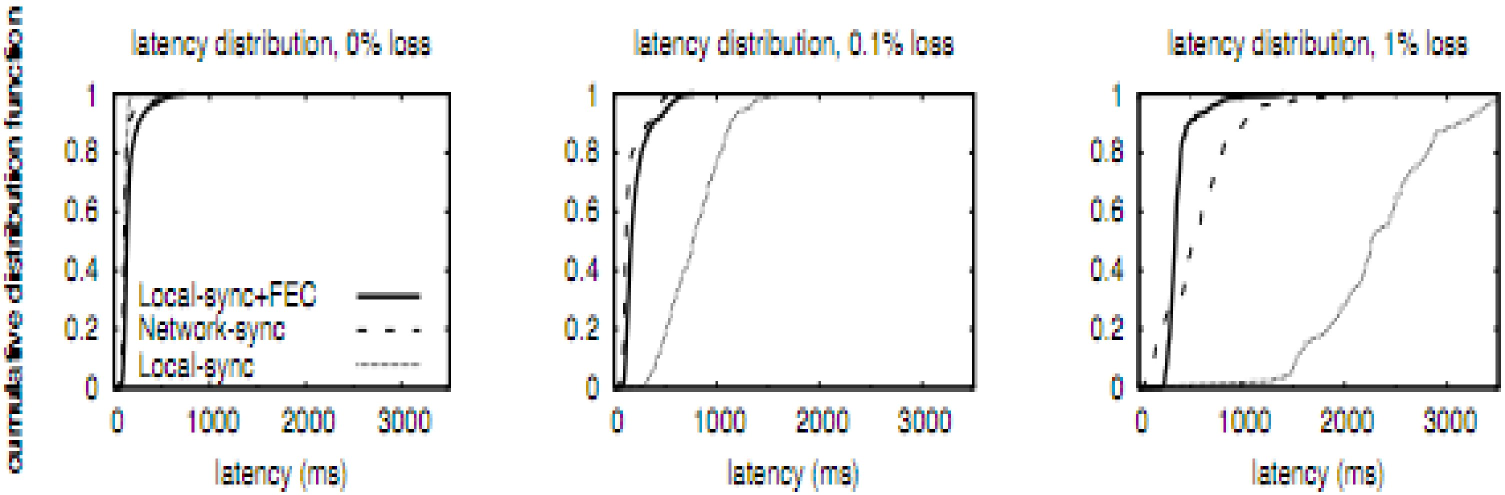


Figure 8: Latency distribution as a function of wide-area link loss (50ms one-way latency).

Throughput

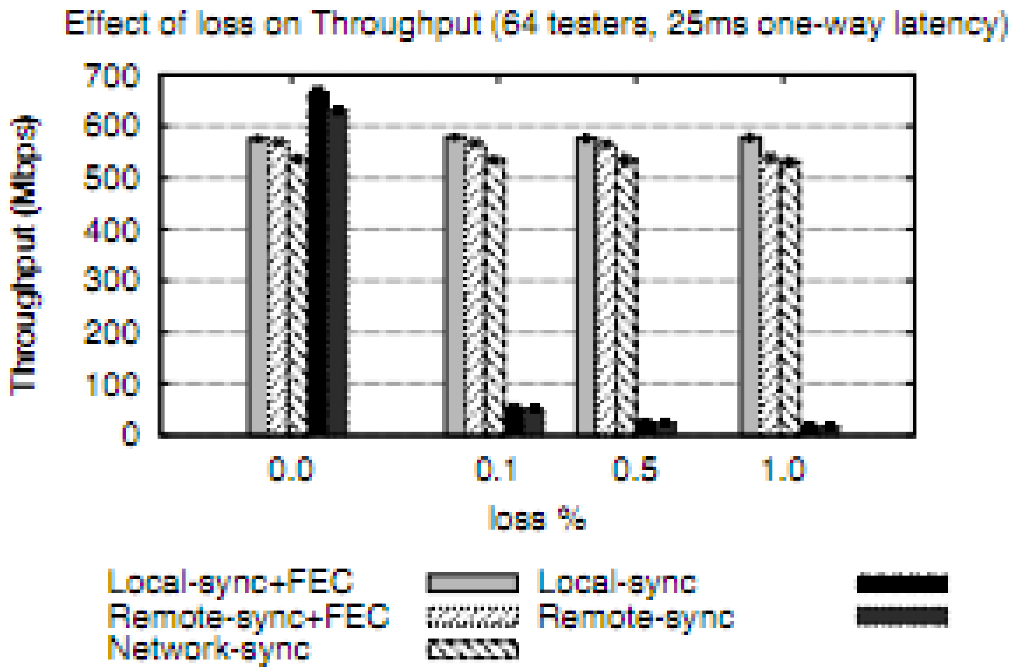


Figure 9: Effect of varying wide-area one-way link *loss* on Aggregate Throughput.

Experimental Setup 2 - Cornell National Lambda Rail (NLR) Rings

- The test bed consists of three rings:-

1) Short (Cornell -> NY -> Cornell)- 7.9ms
2) Medium (Cornell ->Chicago -> Atlanta - > Cornell)- 37ms
3) Long (Cornell->Seattle -> LA -> Cornell) - 94 ms

- The NLR (10Gbps) wide area network that is running on optical fibers is a dedicated network removed from the public internet.

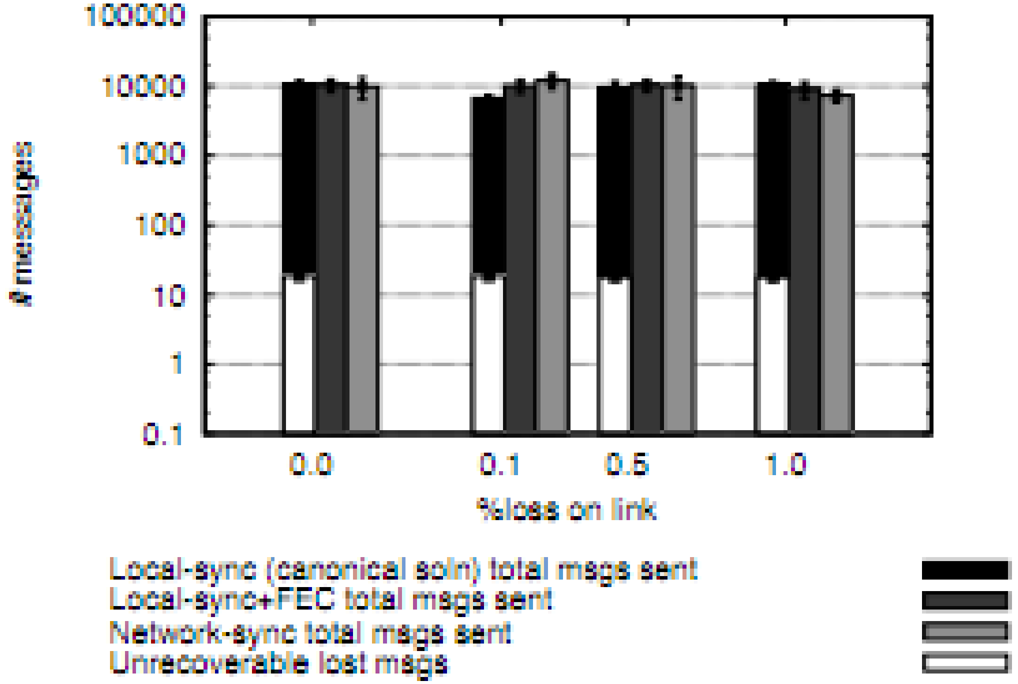


Figure 12: Data loss as a result of disaster and wide-area link failure (Cornell NLR-Rings, 37 ms one-way delay).

Discussion

- Is it a better solution than semi-synchronous?

  Is there overhead due to FEC?
- Single site and Single provider – thoughts?
- Is the Experimental setup that assumes link loss to be random, independent and uniform a representation of the real world?

RACS: A Case for Cloud Storage Diversity

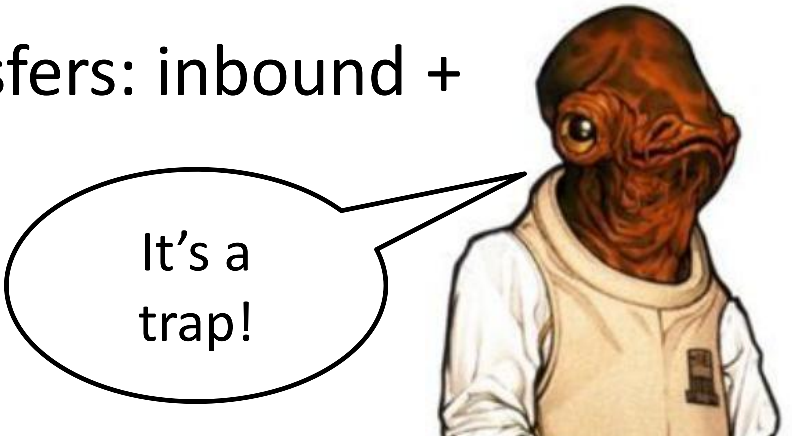
Hussam Abu-Libdeh, Lonnie Princehouse,
Hakim Weatherspoon
Cornell University

Presented by: Jason Croft
CS525, Spring 2011

Main Problem: Vendor Lock-In

- Using one provider can be risky
  - Price hikes
  - Provider may become obsolete

- **Data Inertia**: more data stored, more difficult to switch
  - Charged twice for data transfers: inbound + outbound bandwidth



Secondary Problem: Cloud Failures

- Is redundancy for cloud storage necessary?
  - Outages: improbable events cause data loss
  - Economic Failures: change in pricing, service goes out of business
- In cloud we trust?

Too Big to Fail?

- Outages

DANGER
A Subsidiary of Microsoft

The Planet
The Global IT Hosting Leader

amazon web services

rackspace

Gmail by Google

- Economic Failures

EMC²

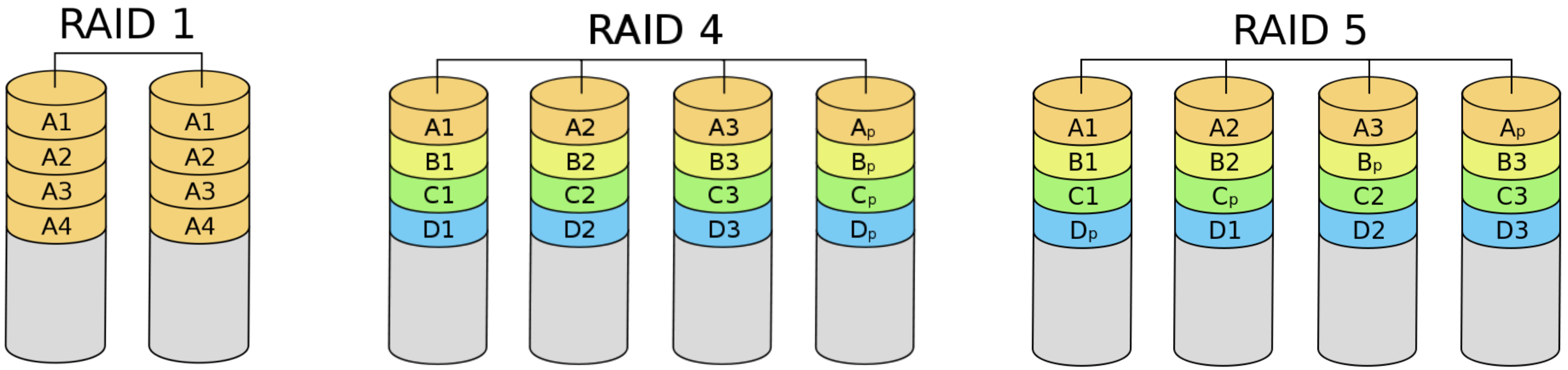
mozy

thelinkup

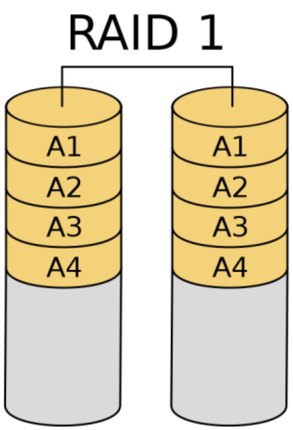
Solution: Data Replication

- RAID 1: mirror data

- **Striping**: split sequential segments across disks
  - RAID 4
  - RAID 5



DuraCloud: Replication in the Cloud



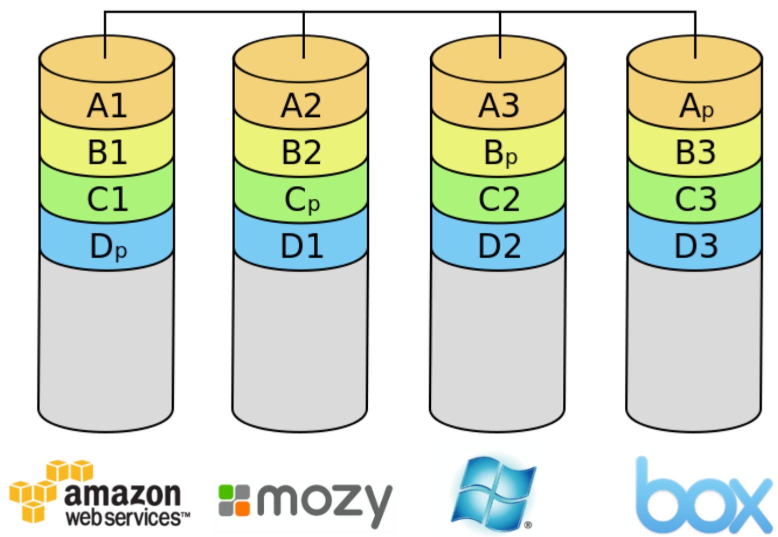
- Method: mirror data across multiple providers
- Pilot program
  - Library of Congress
  - New York Public Library – 60TB images
  - Biodiversity Heritage Library – 70TB, 31M pages
  - WGBH – 10+TB (10TB preservation, 16GB streaming)

http://www.duraspace.org/fedora/repository/duraspace:35/OBJ/DuraCloudPilotPanelNDIIPPJuly2010.pdf

DuraCloud: Replication in the Cloud

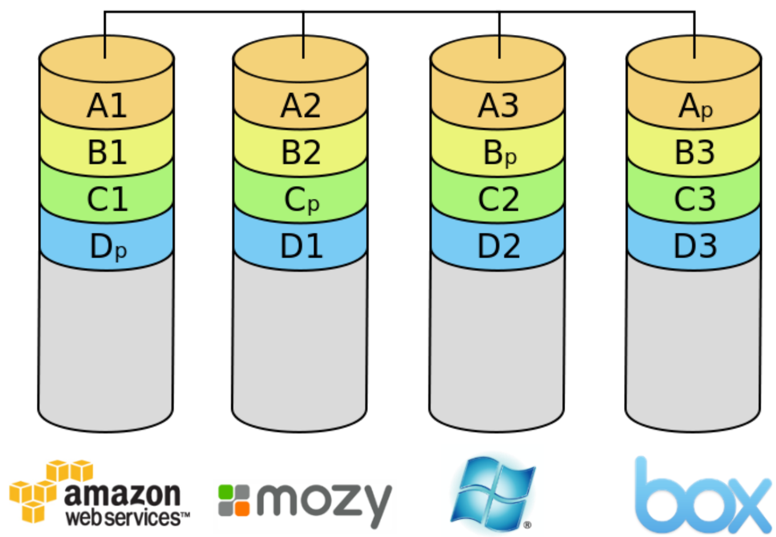
- Is this efficient?
- Monetary cost
  - Mirroring to *N* providers increases storage cost by a factor of *N*
- Switching providers
  - Pay to transfer data twice (inbound + outbound)
  - Data Inertia

Better Solution: Stripe Across Providers



- Tolerate outages or data loss
  - SLAs or provider's internal redundancy not enough
  - Choose how to recover data

Better Solution: Stripe Across Providers

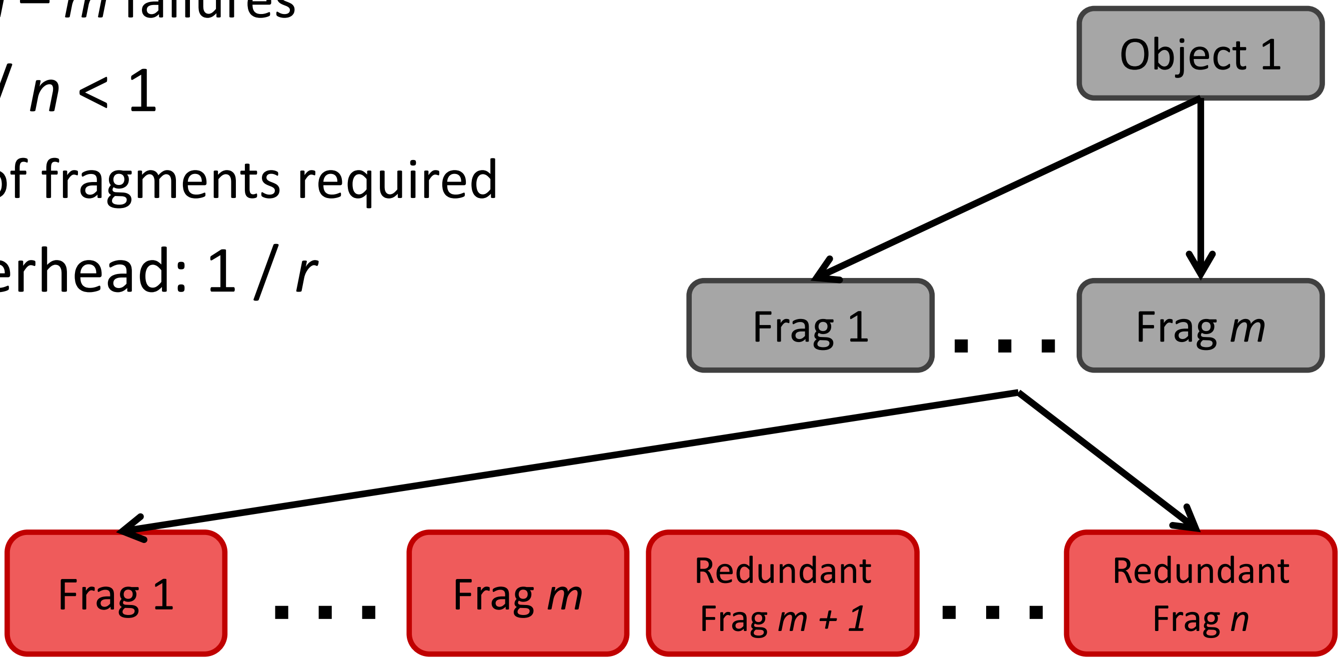


- Adapt to price changes
  - Migration decisions at lower granularity
  - Easily switch to new provider
- Control spending
  - Bias data access to cheaper options

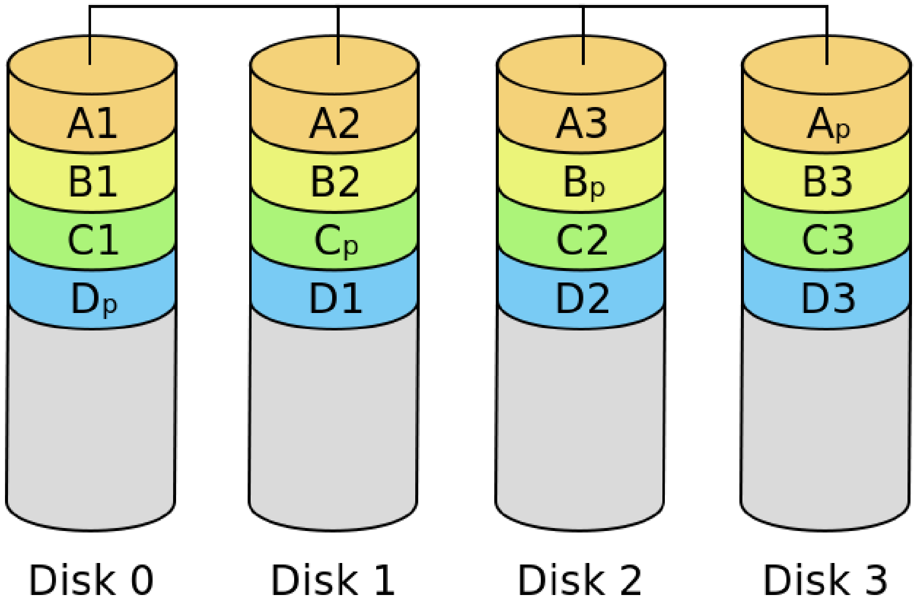
How to Stripe Data?

Erasure Coding

- Split data into *m* fragments
- Map *m* fragments onto *n* fragments (n > m)
  - *n* – *m* redundant fragments
  - Tolerate *n* – *m* failures
- Rate *r* = *m* / *n* < 1
  - Fraction of fragments required
- Storage overhead: 1 / *r*



Erasure Coding Example: RAID 5



(m = 3, n = 4)

Rate: r = ¾

Tolerated Failures: 1

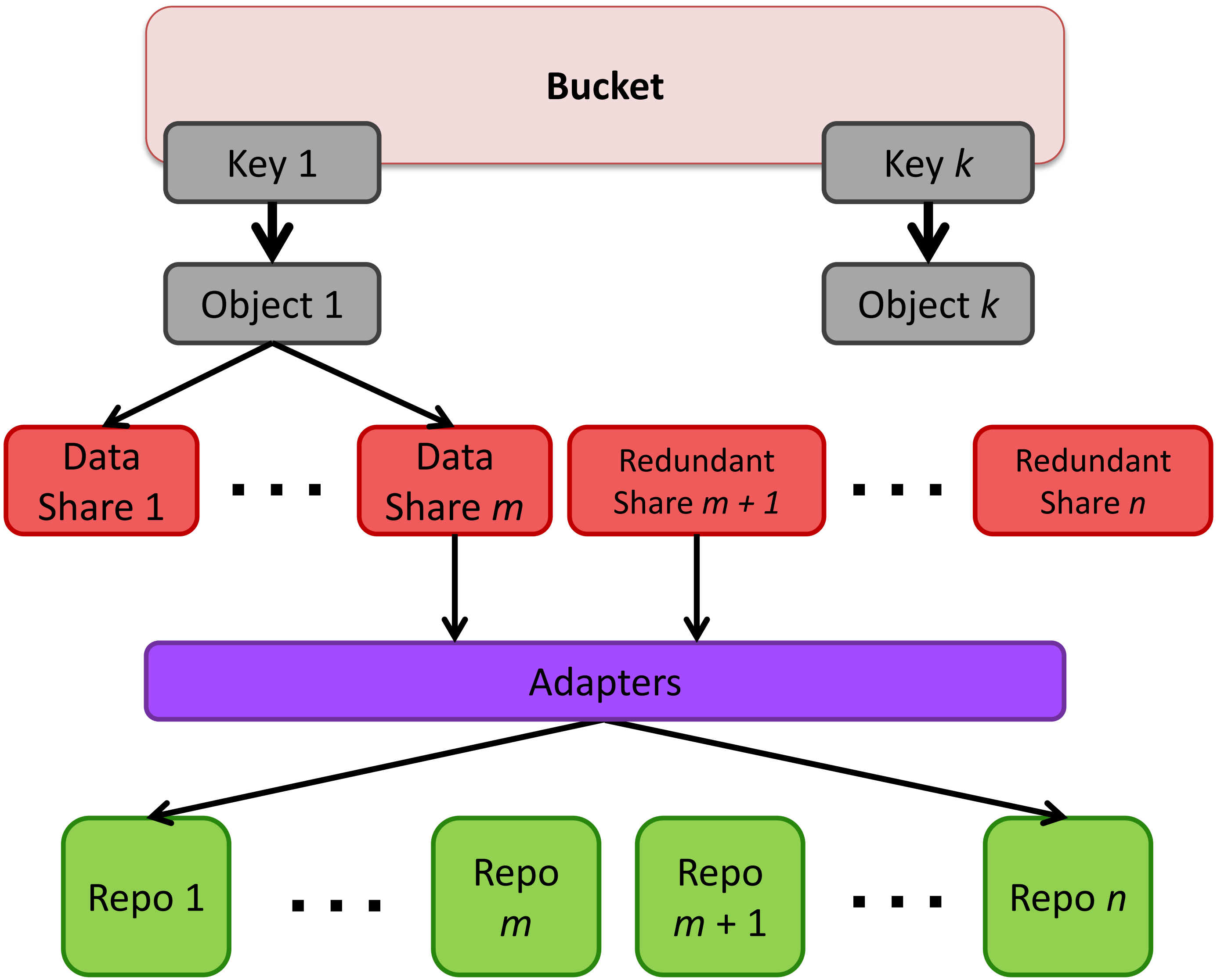
Overhead: 4/3

RACS Design

- Proxy: handle interaction with providers
  - Need *Repository Adapters* for each provider's API
  - E.g., S3, Cloud Files, NFS
  - Problems?
- Policy Hints: bias data towards a provider
- Exposed as S3-like interface

put	*bucket, key, object*
get	*bucket, key*
delete	*bucket, key*
create	*bucket*
delete	*bucket*
list	keys in *bucket*
list	all buckets

Design


Bucket
Key 1
Key k
Object 1
Object k
Data Share 1
. . .
Data Share m
Redundant Share m + 1
. . .
Redundant Share n
Adapters
Repo 1
. . .
Repo m
Repo m + 1
. . .
Repo n

Distributed RACS Proxies

- Single proxy can be a bottleneck
  - Must encode/decode all data
- Multiple proxies introduces data races
  - S3 allows simultaneous writes
  - Simultaneous writes can corrupt data in RACS!
- Solution: one-writer, many-reader synchronization with Apache Zookeeper
  - What about S3's availability vs. consistency?

Overhead in RACS

- ≈ n/m more storage
  - Need to store additional replicated shares
- ≈ n/m bandwidth increase
  - Need to transfer additional replicated shares
- n times more put/create/delete operations
  - Performed on each of n repositories
- m times more get requests
  - Reconstruct at least m fragments

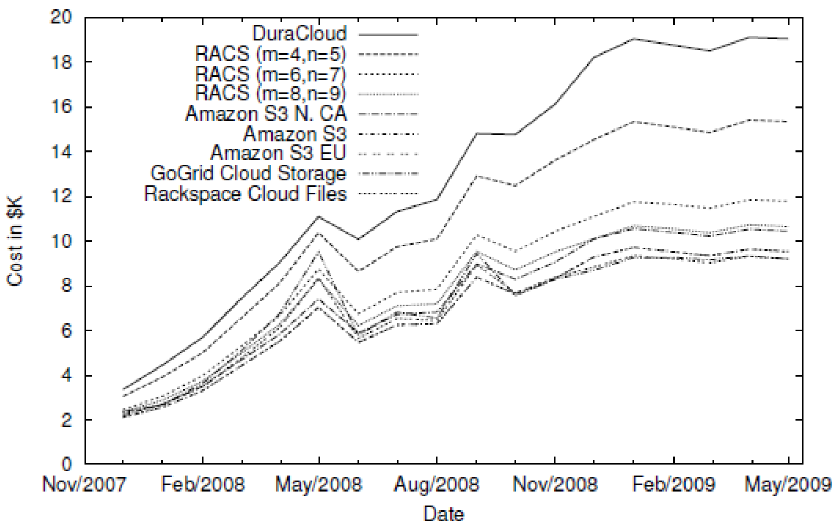
Demo

- Simple (m = 1, n = 2)
    - Allows for only 1 failure

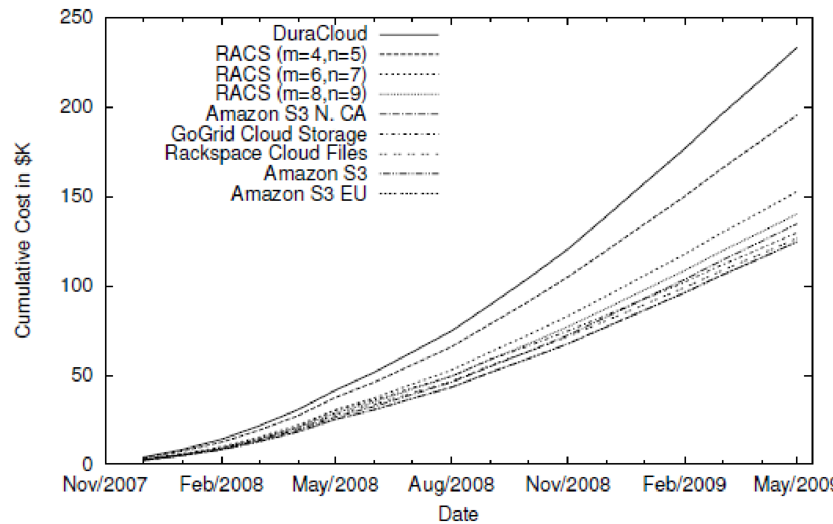
- Repositories:
    - Network File System (NFS)
    - Amazon S3

Findings

- Cost dependent on RACS configuration
- Trade-off: storage cost vs. tolerated failures
  - Cheaper as *n*/*m* gets closer to 1
  - Tolerate less failures as *n*/*m* gets closer to 1



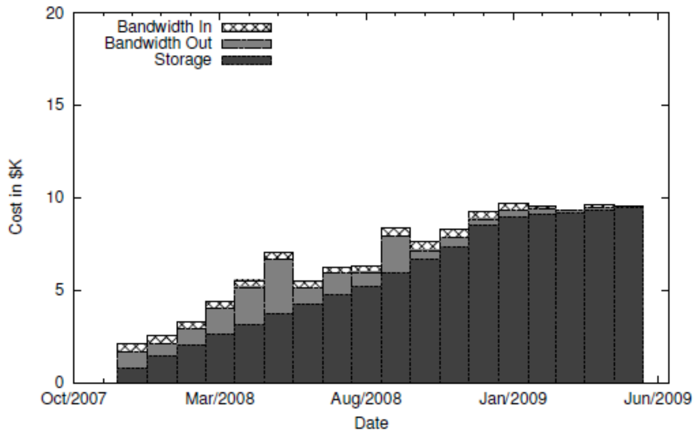
(a) Monthly costs with different storage providers



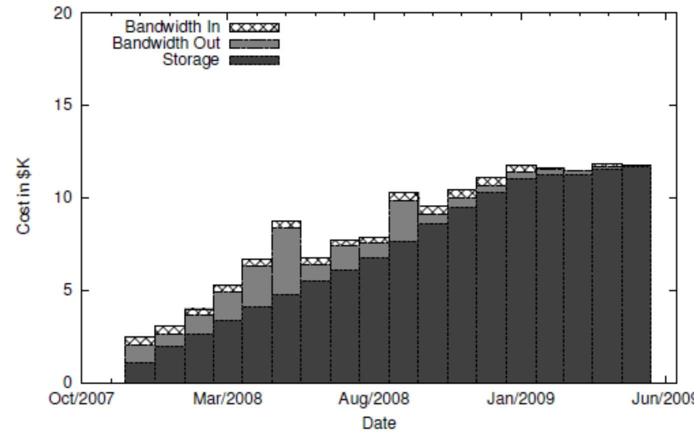
(b) Cumulative costs with different storage providers

Findings

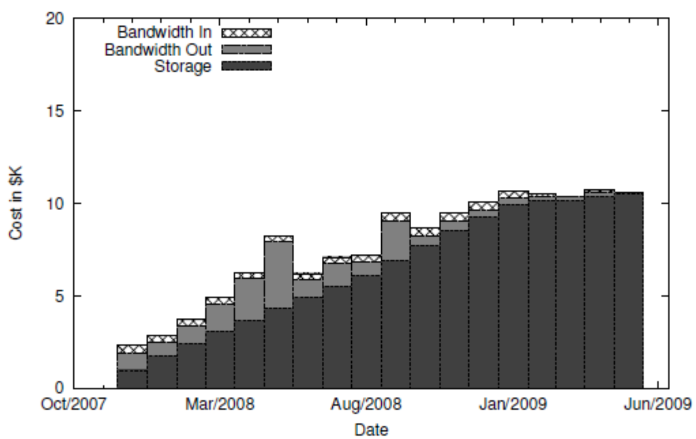
- Storage dominates cost in all configurations



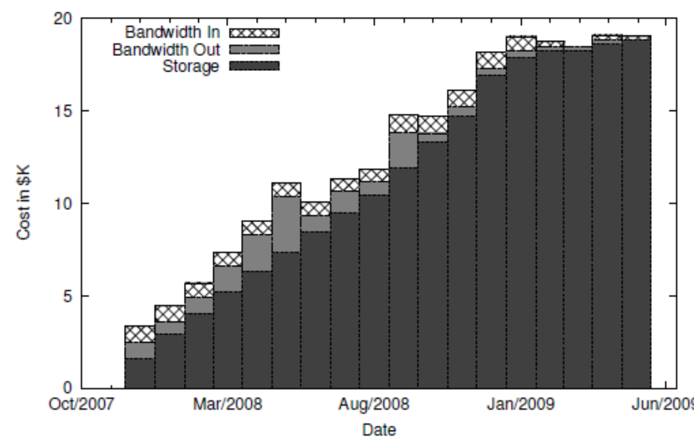
(a) Monthly costs breakdown with Amazon S3



(b) Monthly costs breakdown with RACS(m=4,n=5)



(c) Monthly costs breakdown with RACS(m=8,n=9)



(d) Monthly costs breakdown with DuraCloud

Discussion Questions

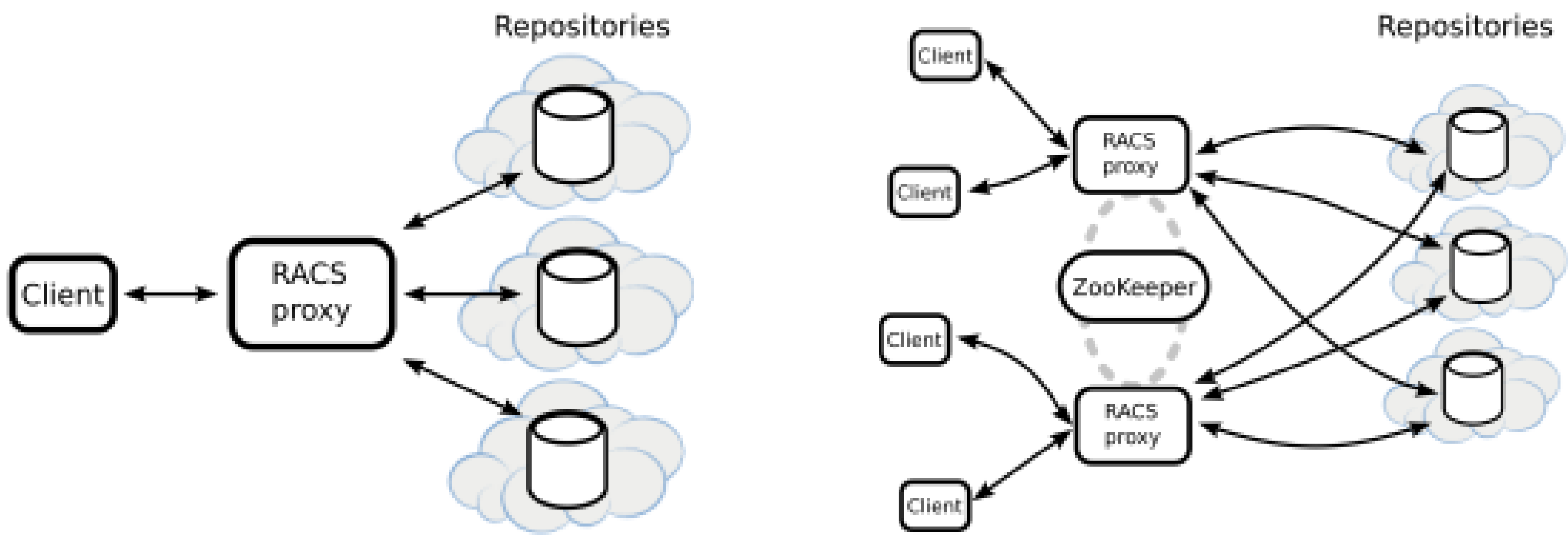
- How to reconcile different storage offerings?
  - Repository Adapters
  - Standardized APIs
- Do distributed RACS proxies/Zookeeper undermine S3's availability vs. consistency optimizations?
- Is storing data in the cloud secure?
  - Data privacy (HIPAA, SOX, etc.)
- If block-level RAID is dead, is this its new use?
- Are there enough storage providers to make RACS worthwhile?

Additional Material

- Amazon Outage: http://status.aws.amazon.com/s3-20080720.html, http://status.aws.amazon.com/s3us-20080720.html
- Maelstrom: http://fireless.cs.cornell.edu/~tudorm/maelstrom/
- R. Appuswamy et al. Block-level RAID is dead. In *HotStorage* ‘10.
- RACS: http://www.cs.cornell.edu/projects/racs/
- Rackspace Outage: http://www.youtube.com/watch?v=hX9qhPhhZs4
- Smoke and Mirrors: http://fireless.cs.cornell.edu/~tudorm/maelstrom/
- Smoke and Mirror Presentation: http://www.usenix.org/media/events/fast09/tech/videos/weatherspoon.mov
- A View of Cloud Computing (CACM, Apr ’10): http://cacm.acm.org/magazines/2010/4/81493-a-view-of-cloud-computing/fulltext
- H. Weatherspoon and J. D. Kubiatowicz. Erasure Coding vs Replication: A Quantitative Comparison. In *IPTPS* ’02.

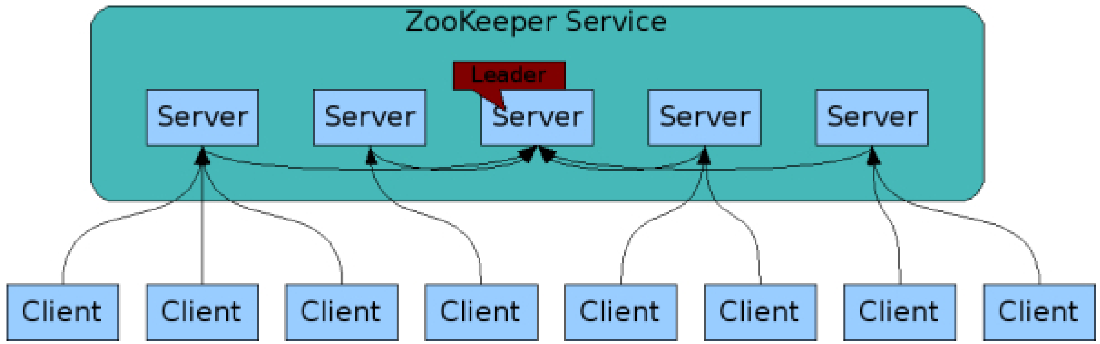
Backup Slides

Design



Repositories
Client
RACS proxy
Repositories
Client
Client
RACS proxy
ZooKeeper
Client
RACS proxy
Client

Zookeeper



- Goal: high performance and availability, strictly ordered access
  - Good for read-dominated loads
- Transactions marked with timestamp, applied in order
- Atomic updates

Example: Internet Archive

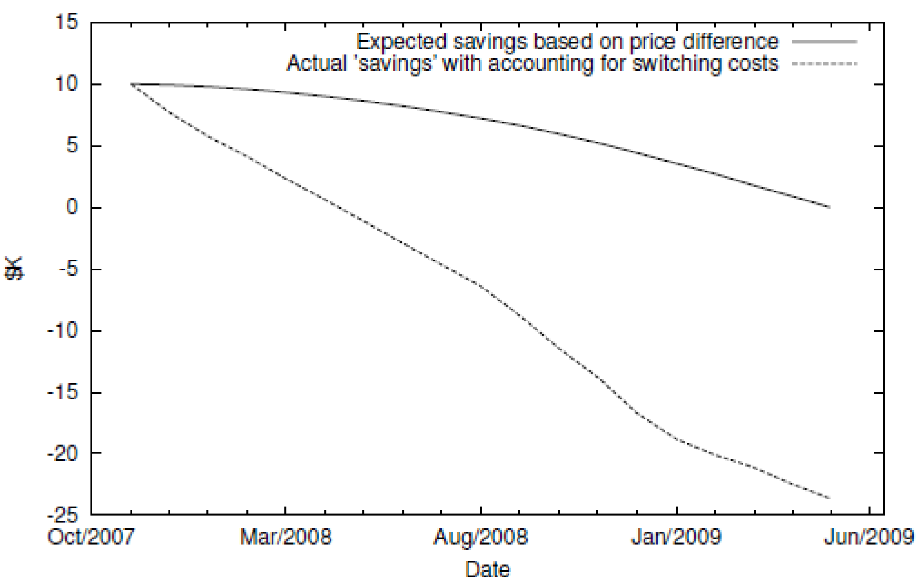
- Internet Archive, or the "Wayback Machine"
  - Permanent storage of snapshots of the Web
- Trace HTTP/FTP interactions over 18 months
- Findings:
  - Volume of data transfers is dominated 1.6:1 by reads
  - Requests are domianted 2.8:1 by reads

Example: Internet Archive

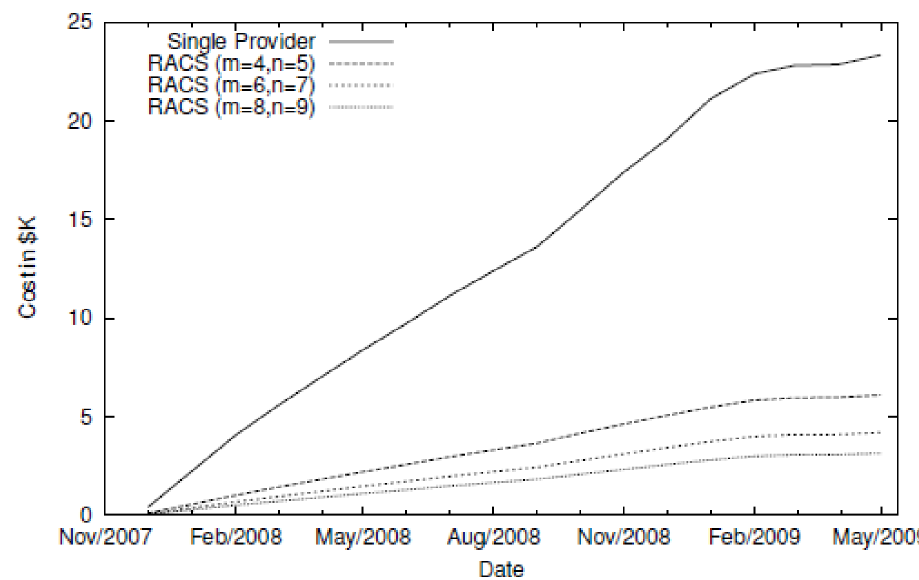
- Single provider: $9.2K – 10.4K per month
- Striping with 9 providers: +$1000 per month (11%)

Finding: Don't Wait to Switch

- Longer with one provider, more expensive it is to switch
- Can cost as much as $23K to switch providers (accounting for bandwidth)



(a) Month-by-month switching benefit (non-RACs solution)



(b) Month-by-month switching costs for various configurations

Figure 7: The cost of switching the Internet Archive's storage provider

Finding: RACS is Cheaper

- Scenario: if price doubles
- Cost to switch is cheaper as *n*/*m* is closer to 1

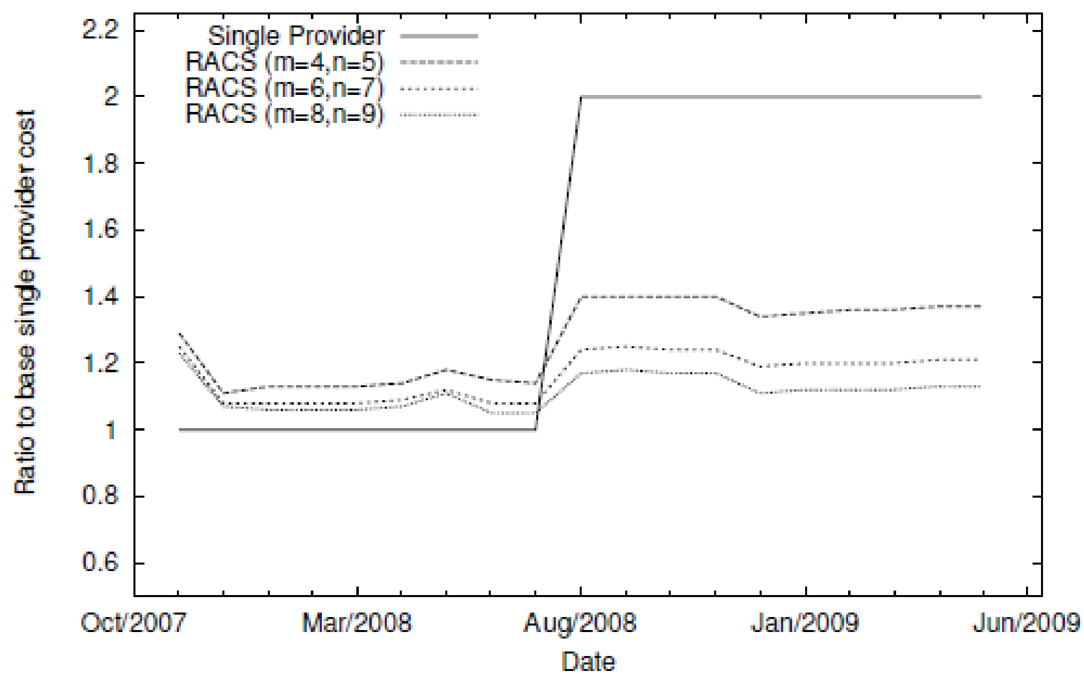


Figure 8: Tolerating a vendor price hike