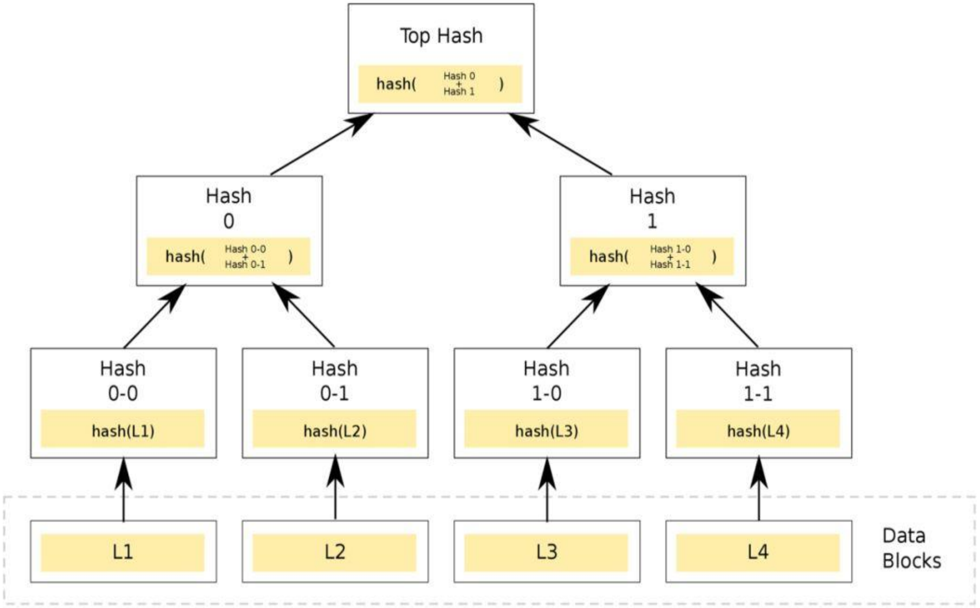
Data Domain Cloud Tier: Backup here, backup there, deduplicated everywhere!

Abhinav Duggal Fani Jenkins Philip Shilane
Ramprasad Chinthekindi Ritesh Shah
Mahesh Kamat
Dell EMC

slide made by wgl

Background

- **Merkle tree**
  - L0-Data Block
  - LP-hash(hash+hash)
  - L6-Top



Motivation

➢ For Customer

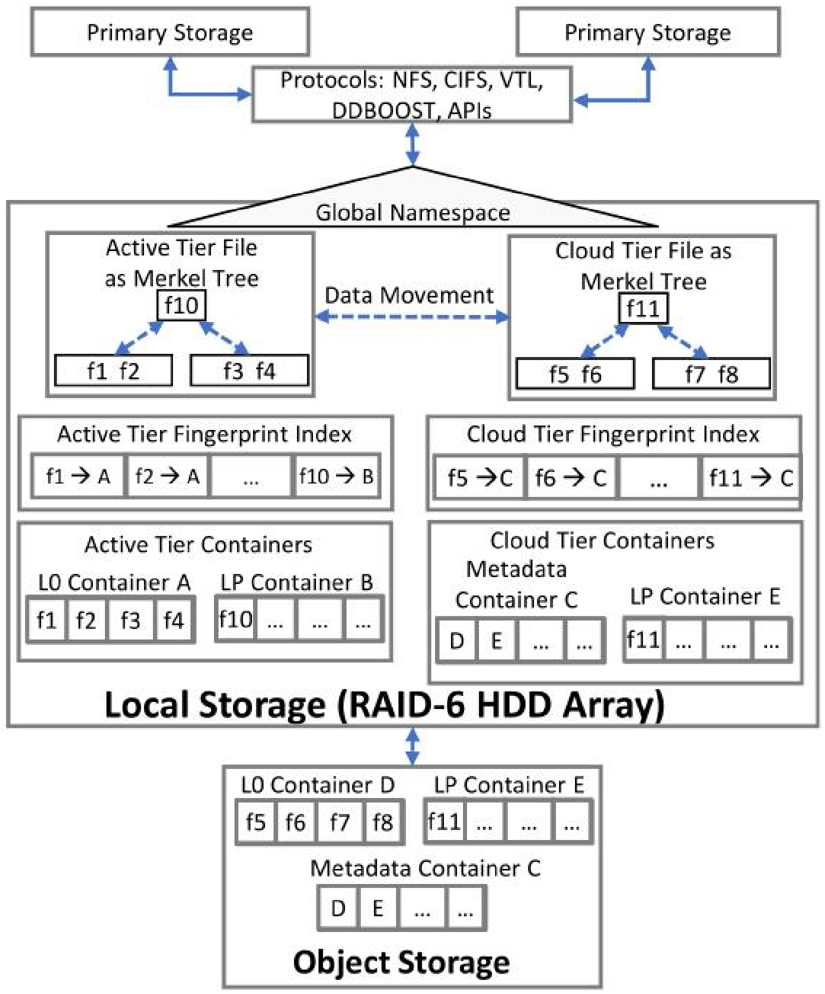
- Firstly a customer wishes to free up space on their active tier by migrating files to the cloud and wishes to determine how much space will be saved.
- They wish to transfer the unique content to the cloud tier to preserve the benefits of deduplication during transfer.
- Finally, as a customer deletes files from a cloud tier, unreferenced chunks must be removed to free space and reduce storage costs

Overview

- **Design**
  - Active & Cloud Tier Architecture
  - Estimate Freeable Space(Perfect Hashing and Physical Scanning)
  - Seeding for ....& File Migration for ..... & File Restore
  - Active & Cloud Tier Garbage Collection
- **Evaluation**
  - Deployed Systems Evaluation
  - Experiments on Internal Systems
- **Conclusion**

Design

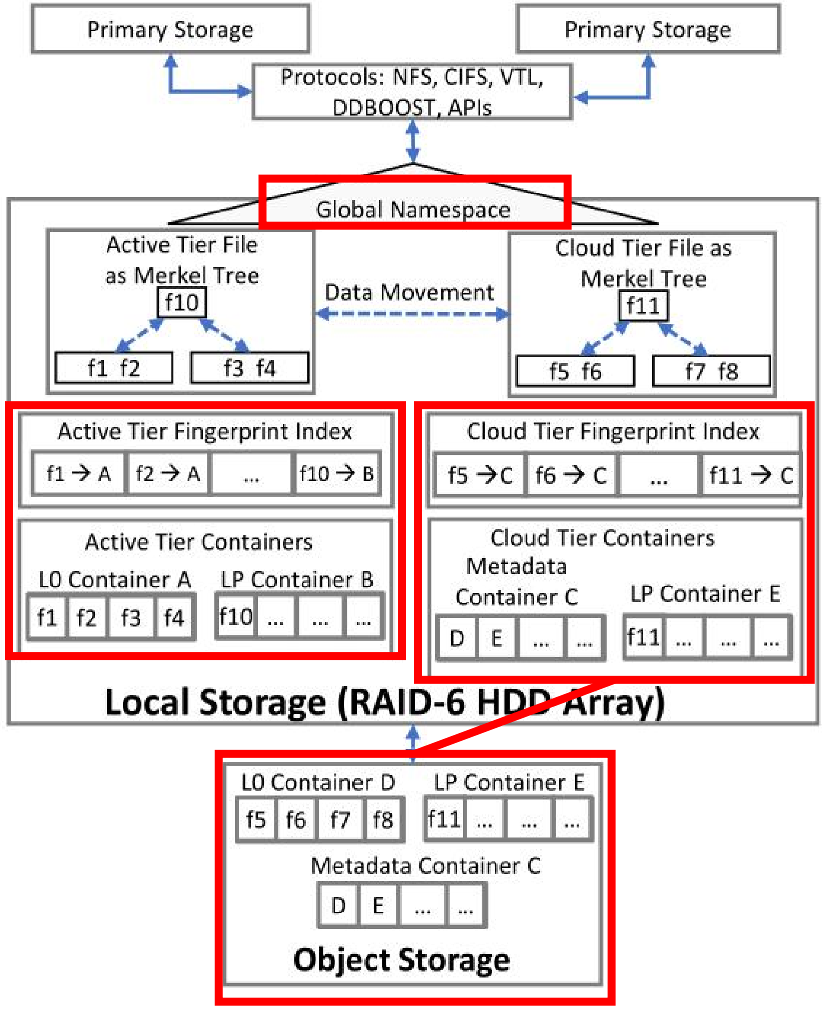
- **Architecture**
  - An active tier where customers backup their primary data
  - A cloud tier where selected backups are transitioned to cloud storage and retained long term
  - A global namespace spans both, active tier and cloud tier files.



Architecture

➢ **Global namespace**
- contains the L6 chunks of every file

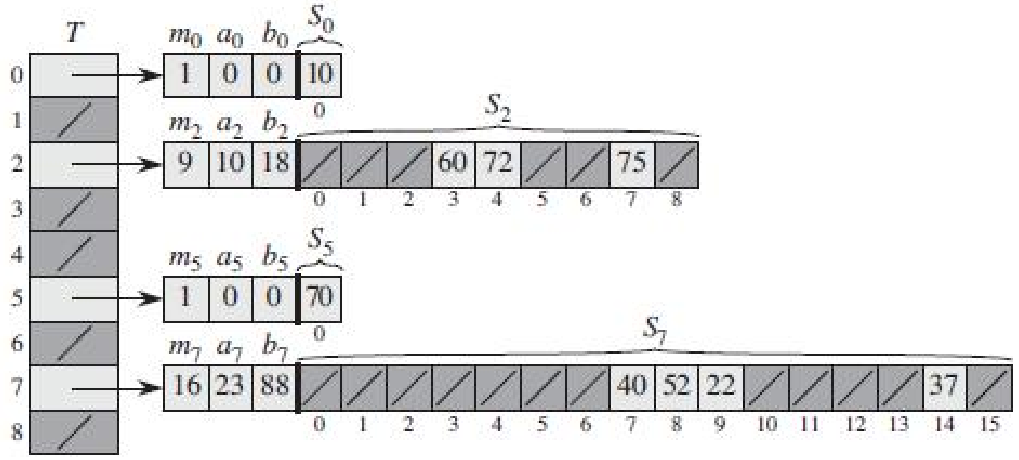
➢ **Active tier**
- L0-Container
- LP-Container
- Active tier fingerprint index (fp-L0&LP)



➢ **Cloud tier**
- L0-Containers & LP-Container
- Metadata Container
- Cloud tier fingerprint index (fp-Metadata)

Per-Design

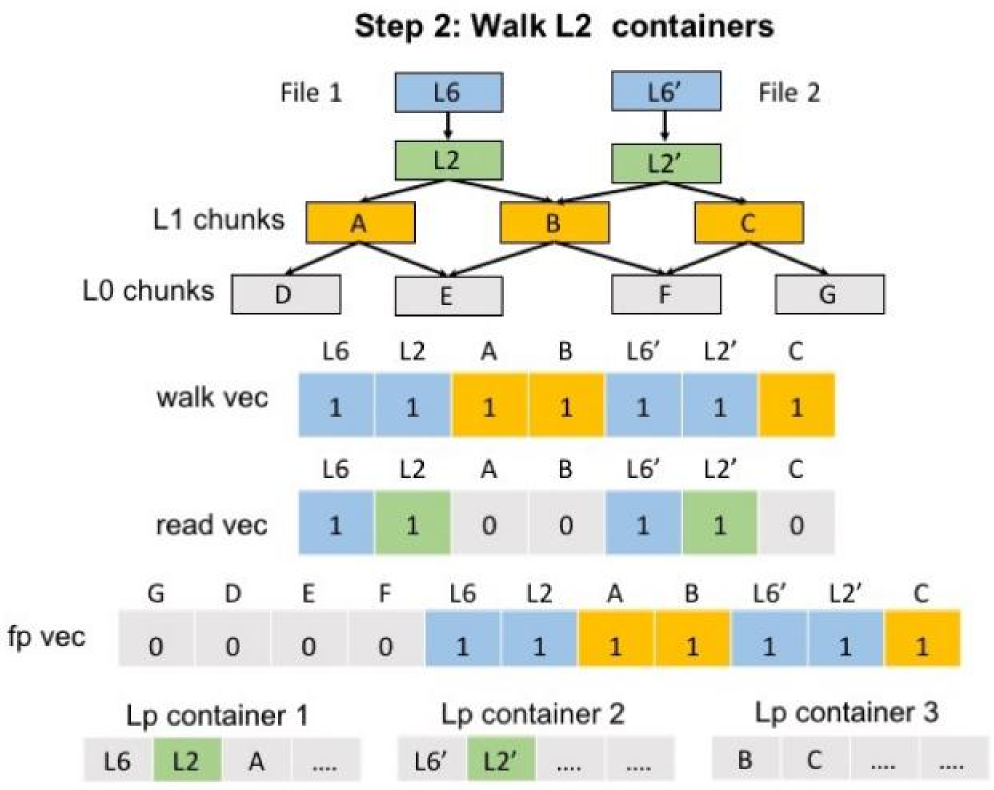
➢Perfect Hashing



- Perfect hash vector = Perfect hash function + a bit vector
- Without the compactness of the perfect hashing representation, we would not have sufficient memory to reference all fingerprints in the system.

Per-Design

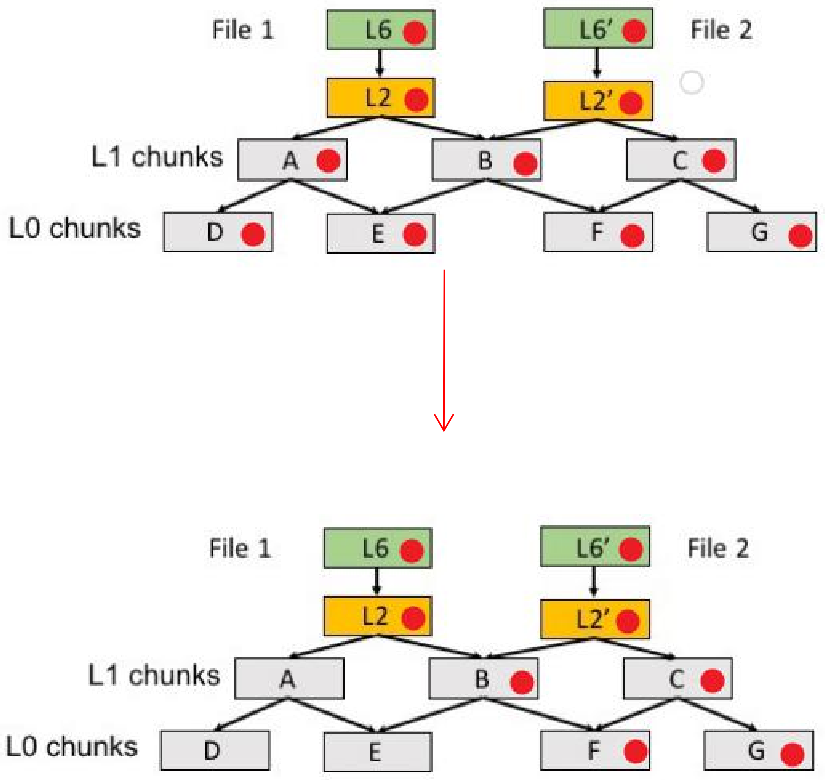
- **Physical Scanning**
  - Depth first manner
    - Enumerating same chunk
  - Breadth first manner
    - LP : walk vector、read vector
    - LP & L0 : fingerprint vector
  - Blue: previous step
  - Green :read vector
  - Yellow :walk and fingerprint vectors



Design

➢Estimate Freeable Space

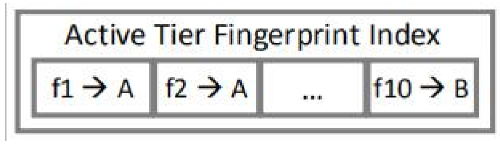
- Firstly walk migration file through the Merkle tree and Mark the chunks
- Then, walk all the remaining files
- Unmark the chunks referenced by these remaining files.
- Sum up the chunk sizes for any chunks still marked in the perfect hash vector



Design

➢Seeding(no deduplicate)

- Firstly, Build an in-memory perfect hash vector (resume)



- Traverse the Merkle trees in breadth first manner and mark the corresponding chunks live in the perfect hash vector(1)

- Walk the containers, pick the chunks which are marked live, pack them in destination containers, and write them to cloud tier.(1-0)

- Data containers (L0) & Containers for metadata (LP and Metadata-Containers) &Cloud tier fingerprint index

- Update namespace

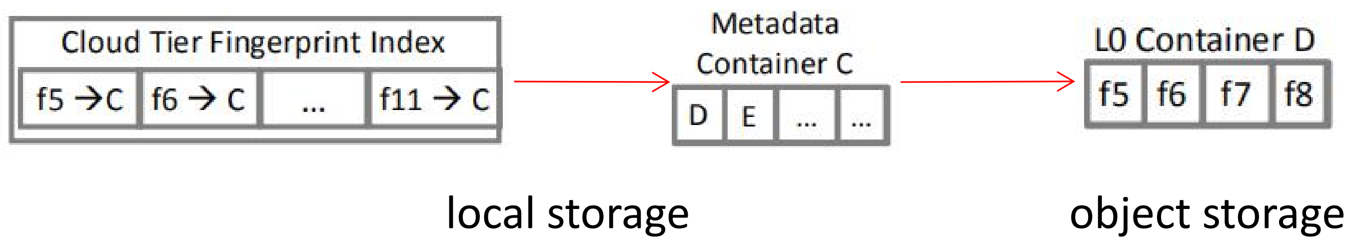
Design

- **File Migration**
  - Traverse the Merkle trees in depth-first search manner
  - Check by caching and metadata prefetching technique
  - Read and pack chunks which are not present in the cloud tier to form new L0-Containers
  - New Merkle trees are formed in a bottom-up manner

Design

➢File Restore

- For new L0 fingerprints



- Write the relevant L0 chunks to L0-Containers in the active tier

Design

- **Active Tier Garbage Collection**

  Copy forward:

  - Firstly Read the source containers from disk
  - Perfect hash vector to determine which L0 chunks are live(1)
  - Finally, we decrypt and uncompress the compression regions, encrypt and compress the live chunks into new compression regions, and pack them into destination containers

Design

➢Cloud Tier Garbage Collection

- The challenge for cloud GC is that L0 containers are not local and reading them from the cloud is expensive

- Compression regions-level

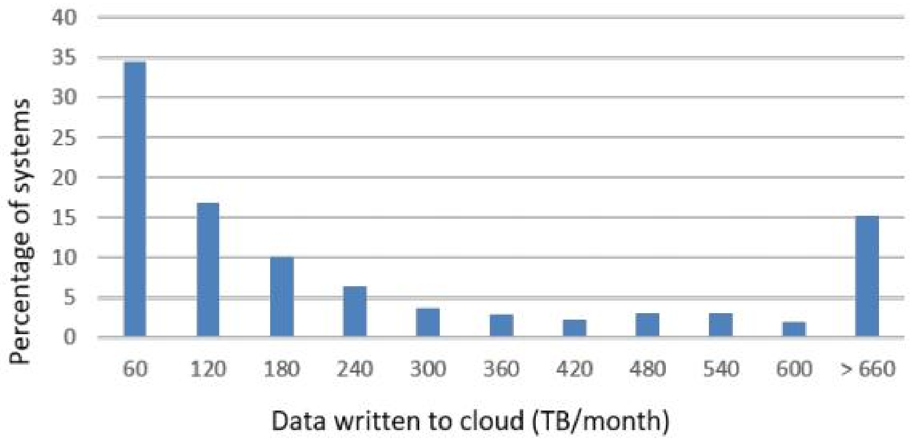
- We only delete a compression region in a cloud container when it is completely unreferenced(shortcoming)

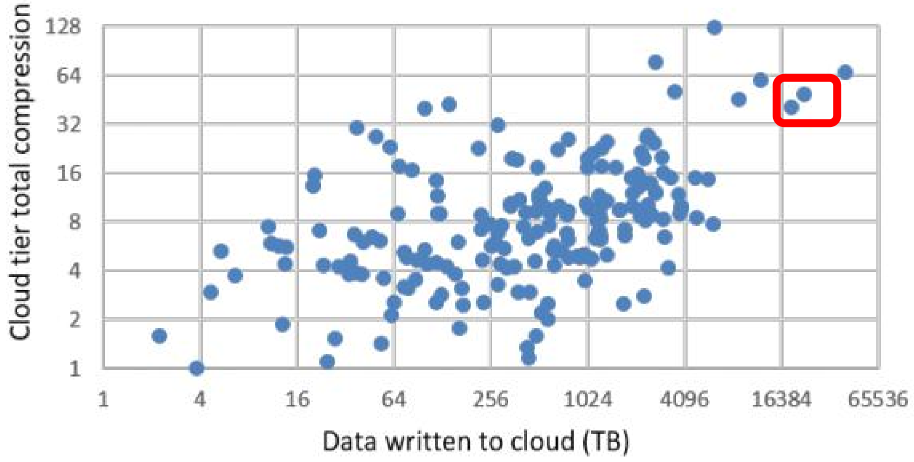
Evaluation

randomly selected 200 systems for analysis.

➢Deployed Systems Evaluation

Total compression ratio =deduplication ratio × local compression ratio.





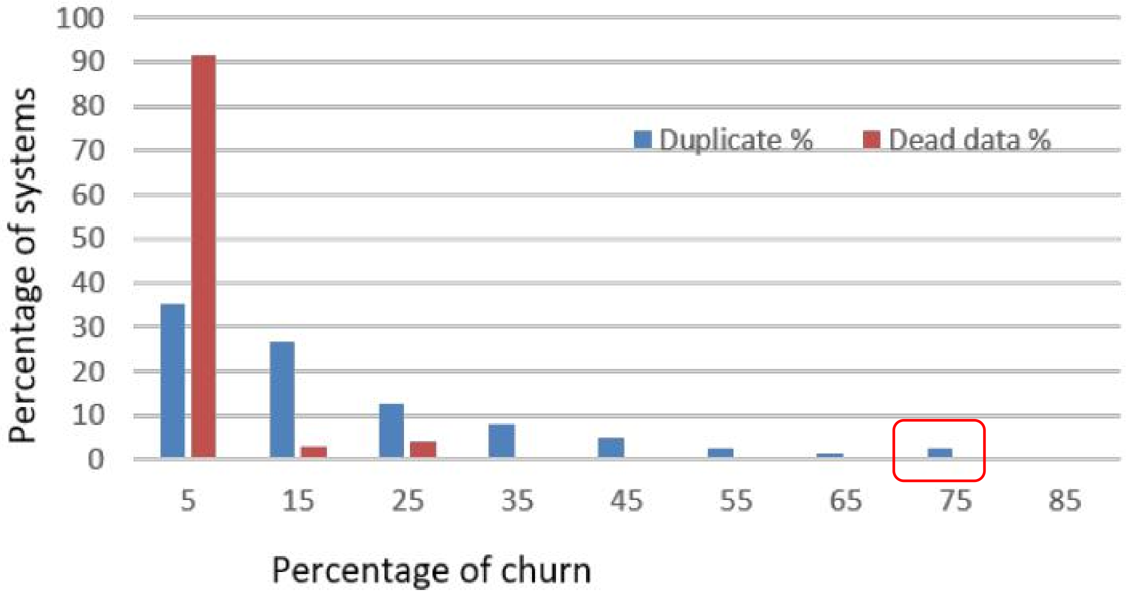
Some customers are writing about 500TB logically per month while others are writing 100TB or less.

More data is moved to the cloud tier, the more total compression is achieved

Evaluation

customers who ran GC at-least once (40% of our selected 200 systems)

➢ Field GC Analysis

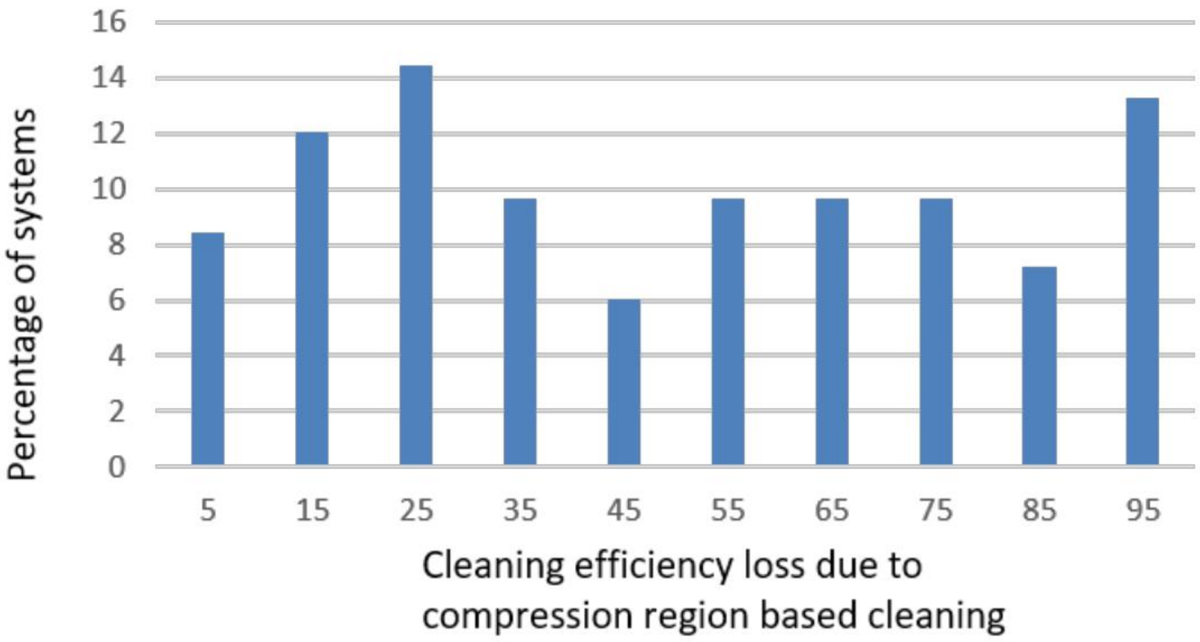


- Unreferenced chunks due to file deletions
- Duplicate chunks.
- churn (bytes deleted within a time period)

The churn due to file deletions is relatively low

Evaluation

➢Cleaning Efficiency Loss



The cleaning efficiency loss is almost 100% in some cases

Evaluation

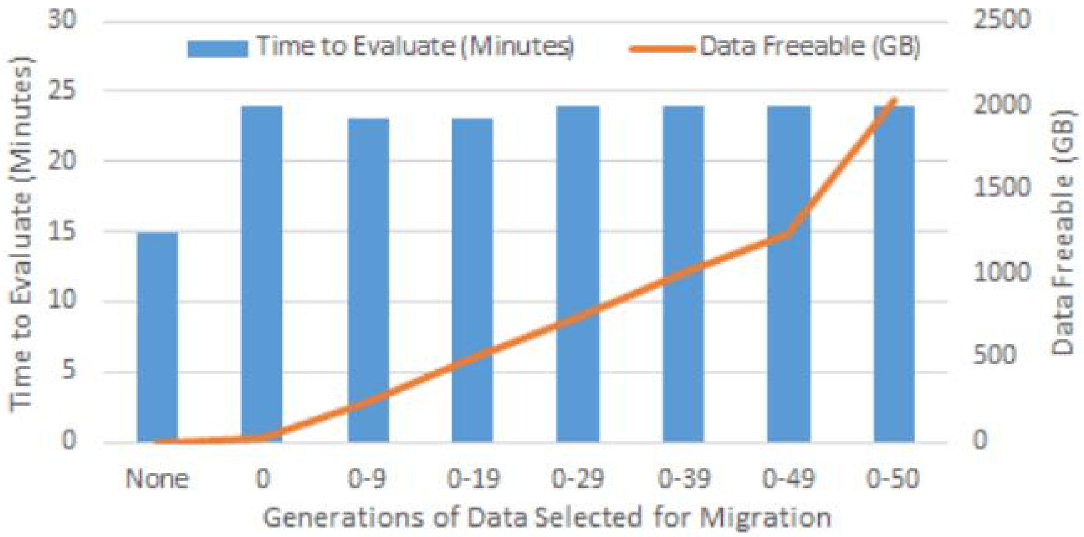
➢ Experiments on Internal Systems

- Amazon S3 (public cloud) and ECS (private cloud)
- Synthetic load generator(Gen 0-N)

Evaluation

generations 0-50 of backups that average 24GB each for a total logical size of 120TB.

➢Freeable Space Estimation



When all generations are selected, the amount of freeable space jumps

- Evaluation time is consistently about 24 minutes
- The amount of freeable space increases

Evaluation

➢File Migration and Restore performance

Hardware	DD-Mid	DD-High
Memory	192GB	384GB
CPU(cores * GHz)	8 * 2.4GHz	24*2.5GHz
Active tier capacity	288TB	720TB
Cloud tier capacity	576TB	1440TB

Table 1: Data Domain hardware for experiments

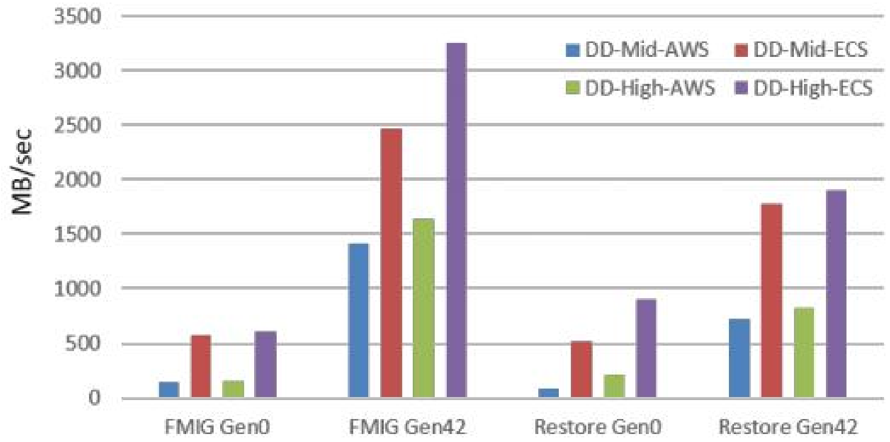
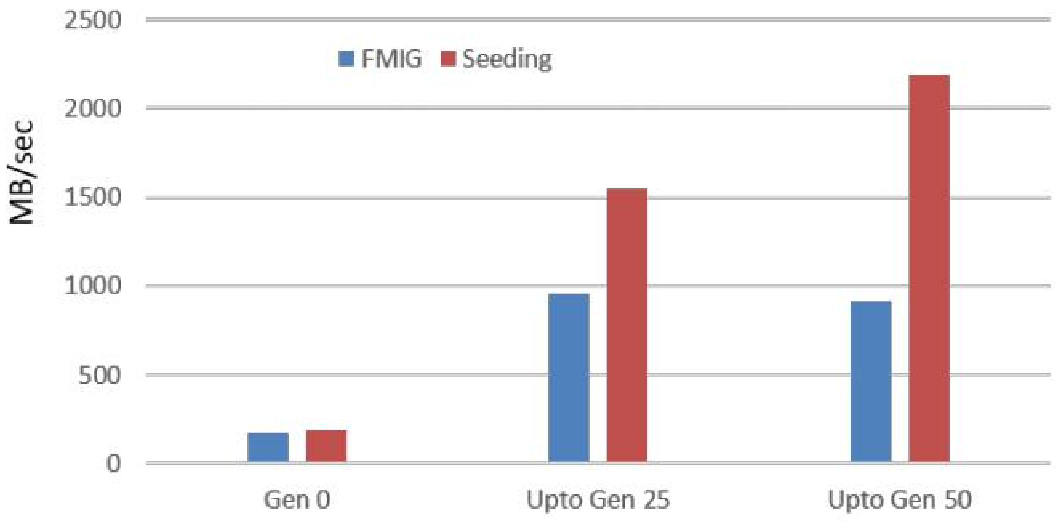


Figure 9: File Migration and Restore performance

- Gen42 migration & restore from object storage is better than Gen0
- Restoring from object storage is typically slower than writing

Evaluation

➢File Migration vs. Seeding Performance



The same containers repeatedly

- L0 performance is similar.
- As we transfer Gen0-25 and Gen0-50, seeding is faster than file migration by a 2x factor

Conclusion

- **Data Domain**
  - Based on customer's process
  - Active & Cloud Tier Architecture
  - Estimate Freeable Space(Perfect Hashing and Physical Scanning)
  - Seeding for ....& File Migration for ..... & File Restore
  - Active & Cloud Tier Garbage Collection
- **Evaluation**
  - Field GC Analysis & Cleaning Efficiency Loss
  - Freeable Space Estimation & File Migration and Restore, Seeding performance