IBM Research - Haifa

Sketching Volume Capacities in Deduplicated Storage

Danny Harnik, Moshik Hershcovitch, Yosef Shatsky, Amir Epstein, Ronen Kat

Previous Works on Estimating Data Reduction

- **Plenty of previous works on data reduction estimation [HMNSV12], [XCS12], [HKMST13], [HKS16]…**
  - Data is currently **not reduced…**
  - Storage had **compression** and **deduplication** capabilities
  - How much space will my data require?

This Work

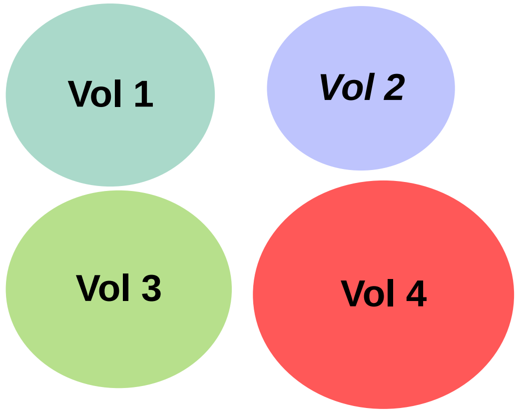
- Data is already in the storage system
- Data is **already reduced**
- **So we know everything about the data reduction, right?**
  - **Not quite**

- Stored physical capacity of entire system is known
- **Challenge**: report capacity at the granularity in which storage is managed
  - Volume / group / pool / file
  - W.l.o.g we will discuss volumes

Deduplication changes the picture

Before deduplication:
Each volume owns its capacity

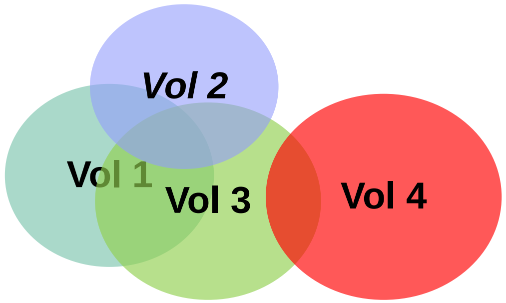
No Dedupe



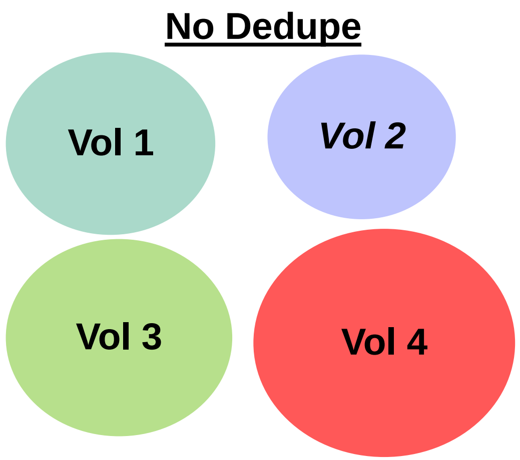
With Deduplication: Data is shared across multiple volumes…

- Which volume owns the data?
- Data reduction of a volume depends on the other data in the system!

With Dedupe

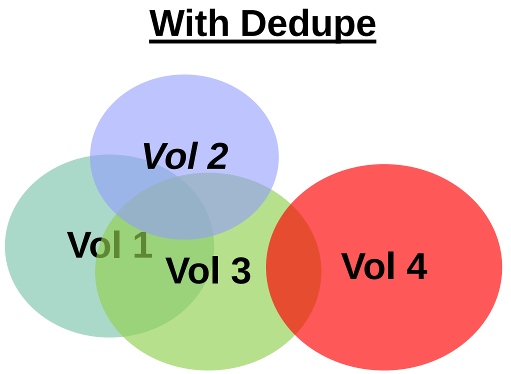


This Work



Estimate the following for every volume/group:

- **Reclaimable capacity -** How much capacity will be freed if a volume is moved out of a system
- **Capacity in another system**
- **Attributed capacity** – A fair sharing of capacities
- Breakdown to **dedupe and compression savings**



Motivation:

- The estimations are instrumental in addressing 3 different topics from the paper “**99 Deduplication Problems**”,
  - Shilane et al. (HotStorage 2016)

1. **Understanding capacities**
2. **Storage management** - including cross system space optimizations decisions/recommendations
3. **Tenant chargeback** – fair capacity billing

Why are volume capacities hard to compute?

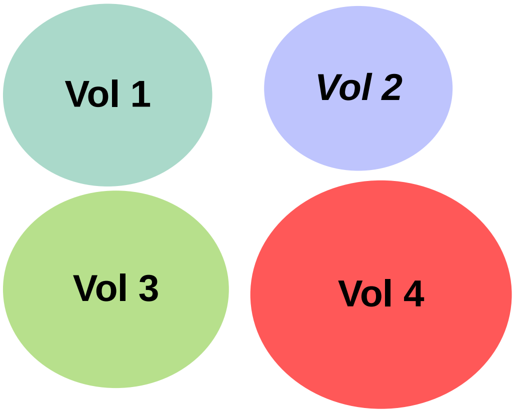
- All metadata exists in the system
- But it is too large to analyze efficiently…

- Cannot update volume stats locally on each I/O
  - An I/O to one volume can effect all other volumes in the system

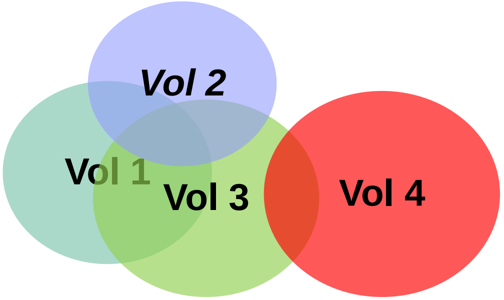
- Reclaimable is not additive!
  - Cannot deduce reclaimable of a group by the reclaimable space of the volumes in the group

  - Heuristics for reclaimable exist but they:
    a) Do not work for groups
    b) Can be grossly incorrect

No Dedupe



With Dedupe

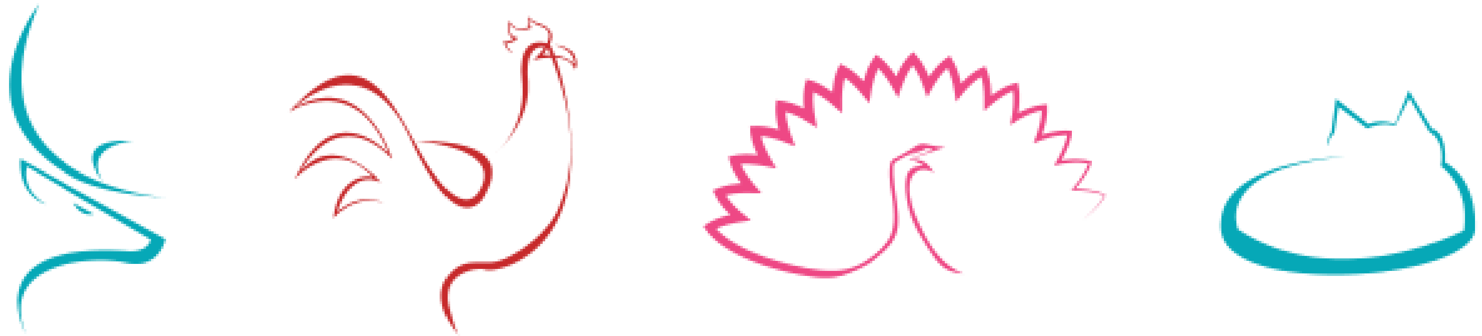


Our Solution: Volume Sketches

- Sketches come from the realm of streaming algorithms

- A sketch - information about the system which

a) Is as small as possible
b) Sufficient to get a decent estimation of what we want to measure



- We use a **content-aware** metadata sampling technique

- A variation of techniques introduced by Gibbons and Tirthapura [GT01] and Bar-Yosef et al. [BJKST02] for distinct elements estimation
  - Xie et al. use a close variant [XCS13] for deduplication
  - Our use case required some changes

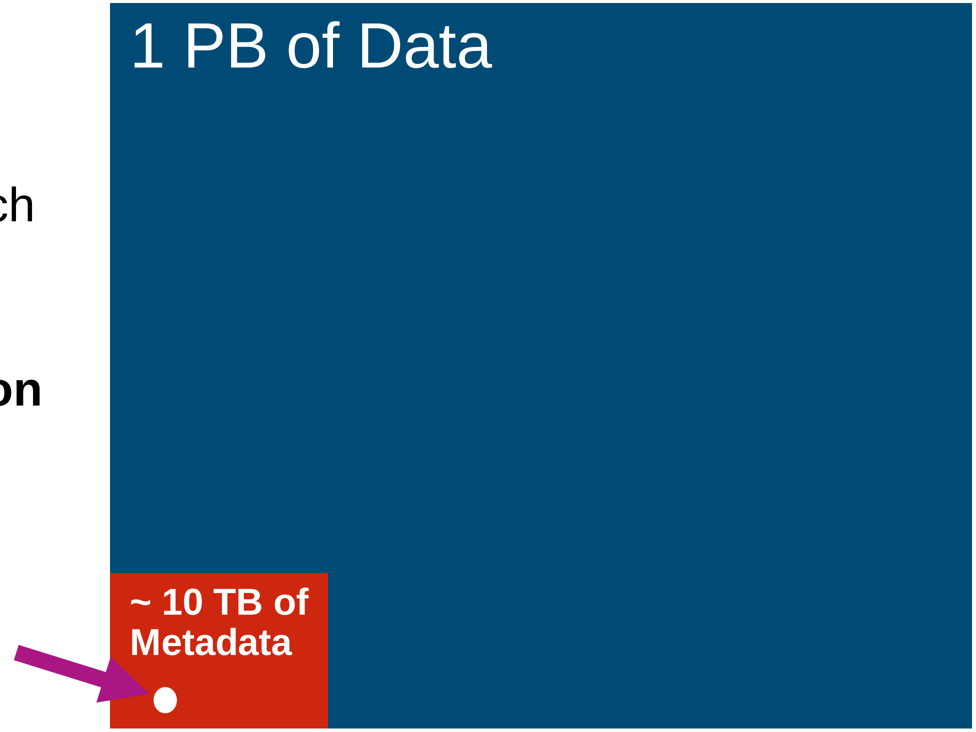
The Actual Method

- Data is split into chunks
  - Could be fixed or variable sized chunking
  - Compute a fingerprint per each chunk
    - A random cryptographic hash of its content
    - Standard method for identifying deduplication

- Does the fingerprint contain **k=13 leading zero bits**?
  - If **yes** then it is in the sketch
  - If **no** then ignore it

- Probability that a hash is in the sketch is **$1/2^k$= 1/8192**
- The **sketch size** is smaller than the written data by a factor of **~3.5 Million**
- This makes analyzing the sketch manageable even for very large systems

~300MB of sketch data



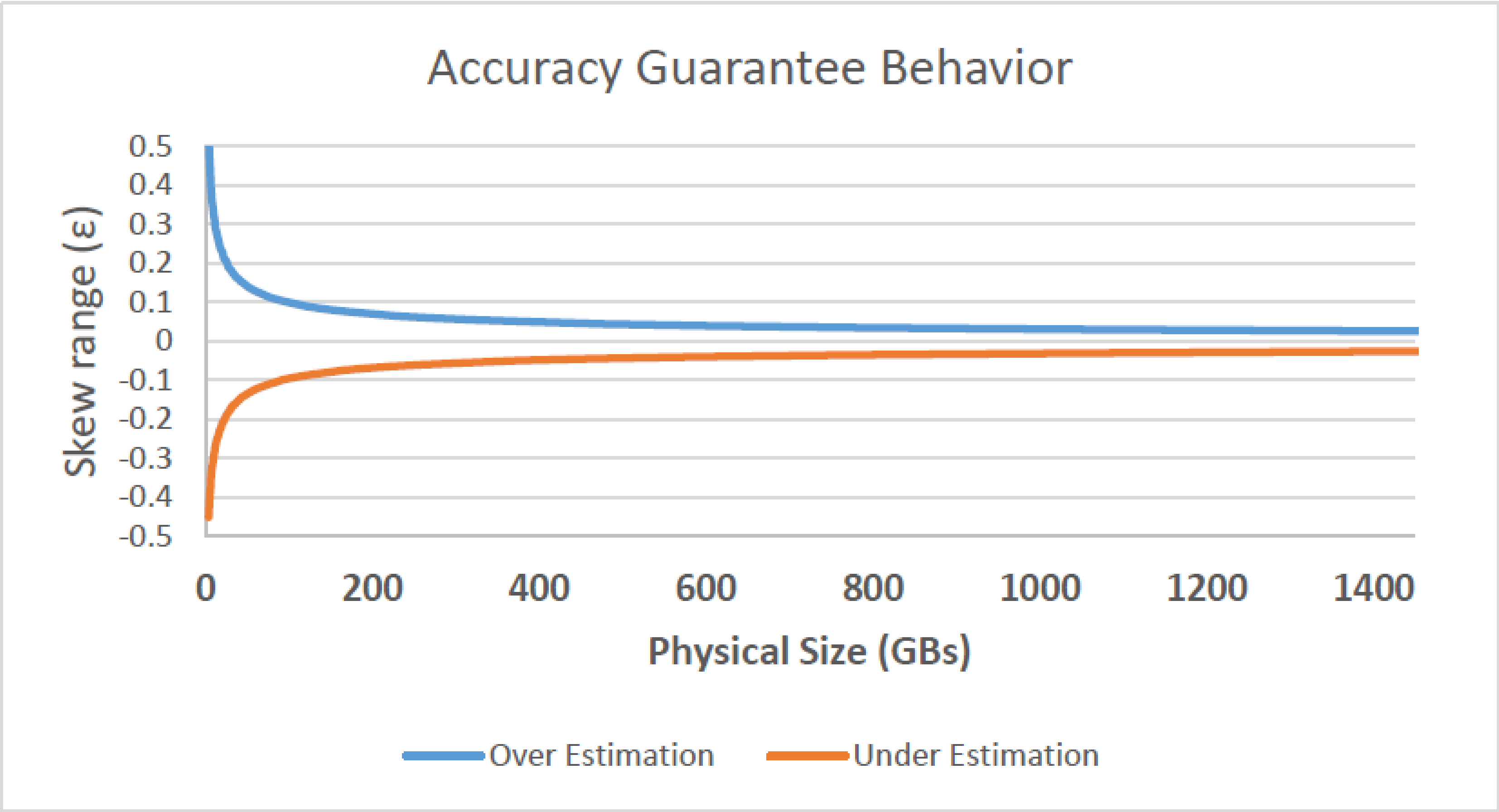
Notes on Sketches

- **Crucial property:** For every hash value **h** in the sketch, **all** the chunks in the system with fingerprint **h** will be monitored in the sketch

- To estimate a capacity measure simply estimate it on the sketch and then multiply by the sketch factor ($2^k = 8192$)

- Some subtleties when computing attributed, reclaimable, etc…
  - Requires a sketch per volume/group

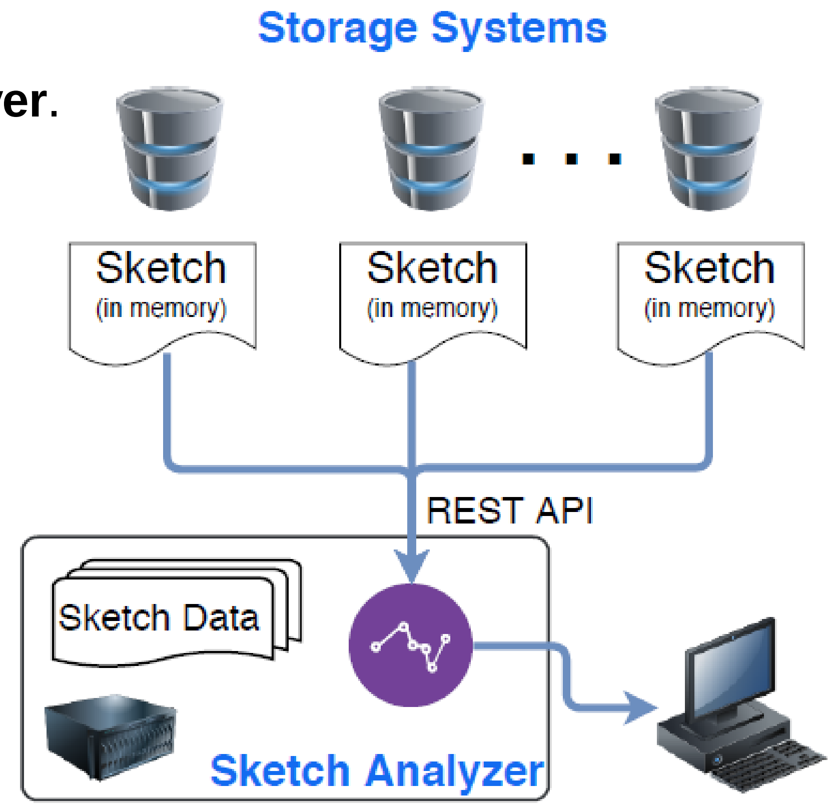
Estimation Accuracy



- Accuracy is a function of the **physical capacity** being estimated
- Larger capacity means higher accuracy
- Holds for all estimations (attributed, reclaimable, etc...)
- Proof is a modification of the multiplicative Chernoff bound

Design and Architecture

- Sketches are analyzed on an **external server**.
  - Avoids using extra CPU cycles on the storage systems
  - Easier to deploy
  - Ideal for cross system optimizations

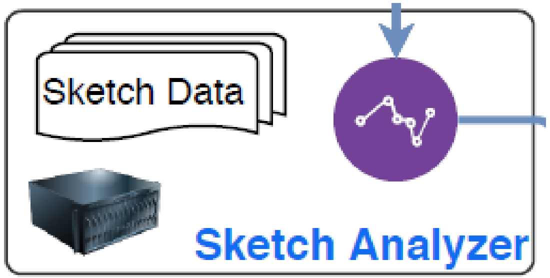


Sketch Collection

- In the storage system all sketch metadata is always **maintained in RAM.**
  - Avoids extra I/Os when fetching sketch
- Sketch is distributed on the system
  - As opposed to aggregated
- The sketches method is deployed in the **IBM FlashSystem A9000/A9000R**

- Note: Sketch does not represent a point in time snapshot of the system, but rather a fuzzy state

Sketch Analysis



- Runs in two main phases:

1. **Ingest phase** – aggregate the distributed sketch in data structures for
   - Volume sketches
   - Full system sketch

2. **Analysis phase** – compute the various measures for all volumes in a system
   - Can also query groups at this stage
     - Create a group sketch by merging the volume sketches
     - Run analysis on the group

- Emphasize analysis speed to support quick query times (e.g. on volume groups)
  - This is a crucial building block for next level optimizations that enumerate a large number of combinations

Evaluation – Workloads

- Used 3 types of data for evaluation

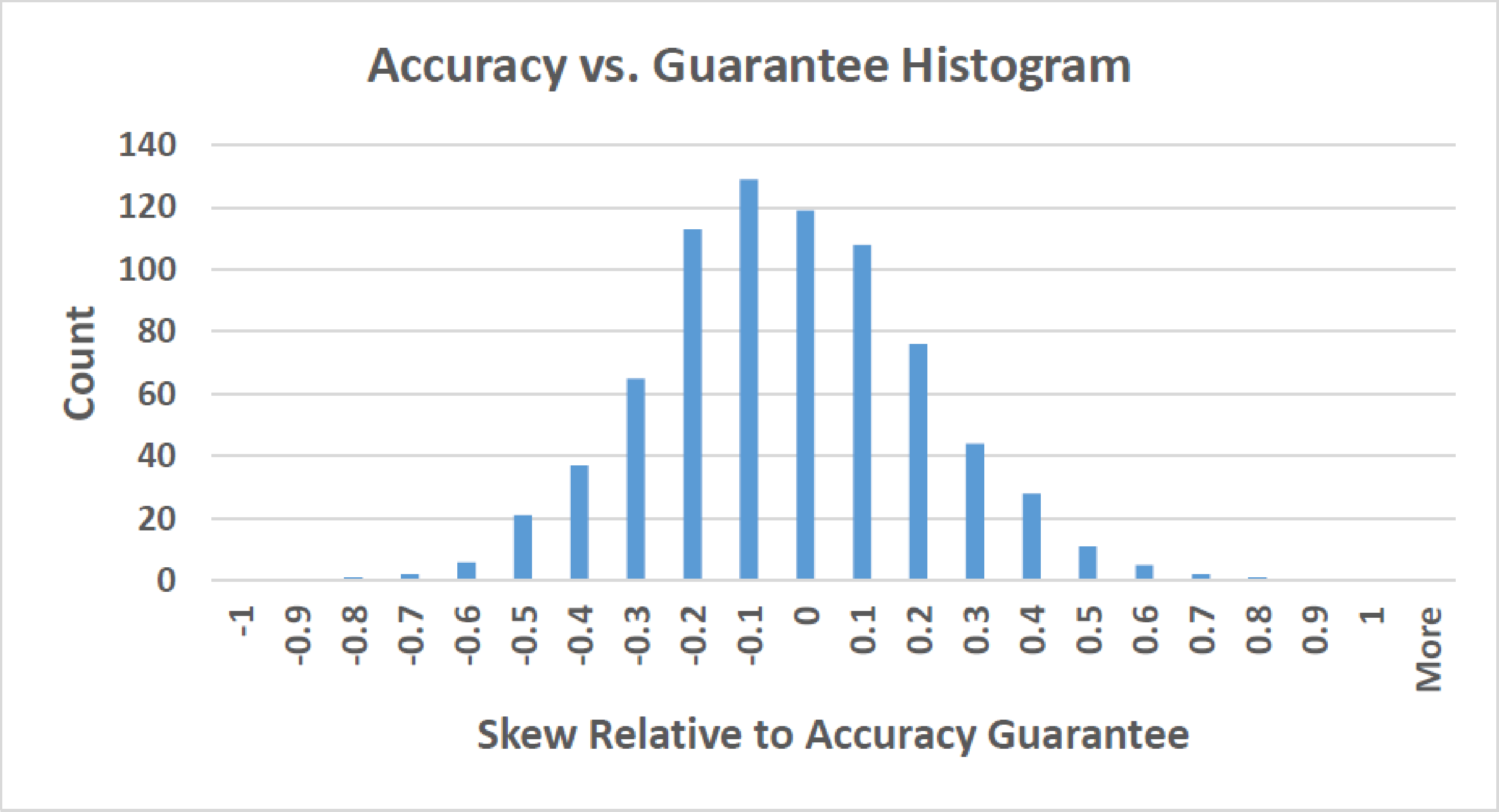
1. **Synthetic data** – various combinations of dedupe and compression ratios
   – Size up to 1.5 PB
2. **UBC-Dedupe Traces** – collected as part of the Meyer & Bollosky study [MB11].
   – 63TB of data written across 768 file systems
   – Include deduplication fingerprints (no compression data)
   – Available from the SNIA IOTTA
3. **Call home from field** – general stats about the sketches mechanism

- **Timing examples:**

	Number of volumes	Size (TB)	Ingest time (sec)	Analysis time (sec)
Synthetic	5	1500	89	0.93
UBC-Dedup	768	63	22	0.21
Field 1	3400	980	104	4.80
Field 2	540	505	65	2.70

Accuracy Evaluation

- Compare the reclaimable estimations for UBC-Dedup volumes vs. actual
- **Normalize difference by the accuracy guarantee**



Data Center Level Optimizations

- Our method is instrumental for cross system space optimizations
- As an example we implemented a greedy algorithm for space reclamation in an environment with multiple deduplicated storage systems.

- **The setting:**
  - 4 systems, each holding 192 random volumes from the UBC dataset
  - On average each system holds 7 TBs of physical space

- **Goal:** generate a plan that frees 1 TB of space from a source system
  - Plan includes: What volumes to move and where to move them to
  - Objective: minimize overall space consumption

- **Results:**
  - Algorithm ran between 30 to 55 seconds
  - Saving between 257GB to 296GB
    - Results depend on the source system…

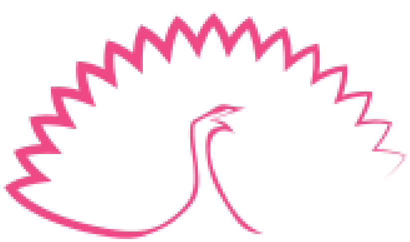
Summary

- Introduce sketching for managing capacities in systems with deduplication

- Brings clarity to capacities in a deduplicated world
- Opens the door to many space management applications

- Deployed in a real world all-flash storage system






Thank You !