Commutativity-guaranteed Docker Image Reconstruction towards Effective Layer Sharing

Sisi Li, Ao Zhou, Xiao Ma, Mengwei Xu, Qingyuan Jiang, Shangguang Wang

Beijing University of Posts and Telecommunications

Background

Container

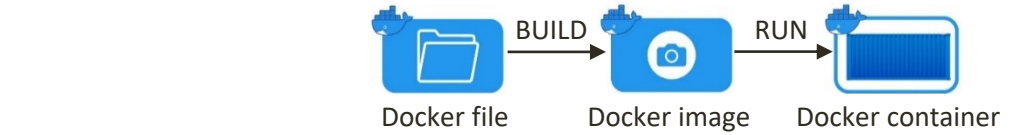
Application deployment evolvement with containers.

IBM Cloud Functions | Azure Functions | Knative | Alibaba Cloud | Openwhisk

Image

Rapid growth of container images.

E.g.: Docker Hub stores more than 2 million public images occupying 1 PB storage.



Service

Frequent service deployment and migration.

E.g.: Google starts average 7,000 containers per second.

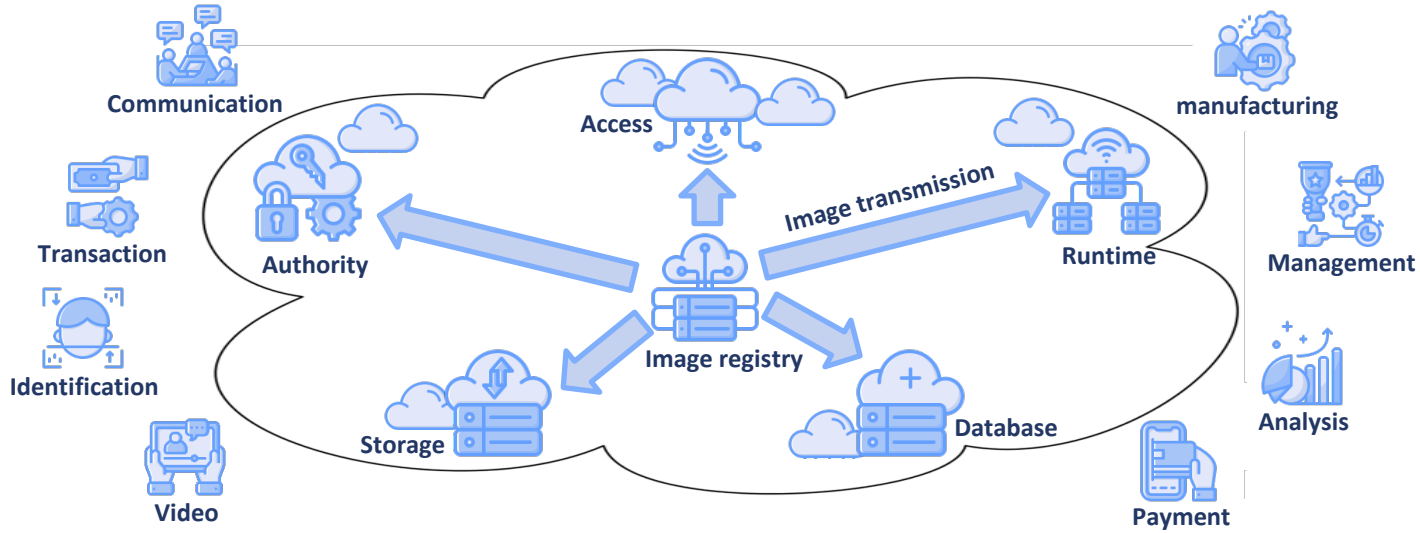
Background

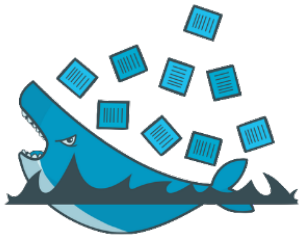
Container

Image

Service

- Huge pressure on storage and network of container images.

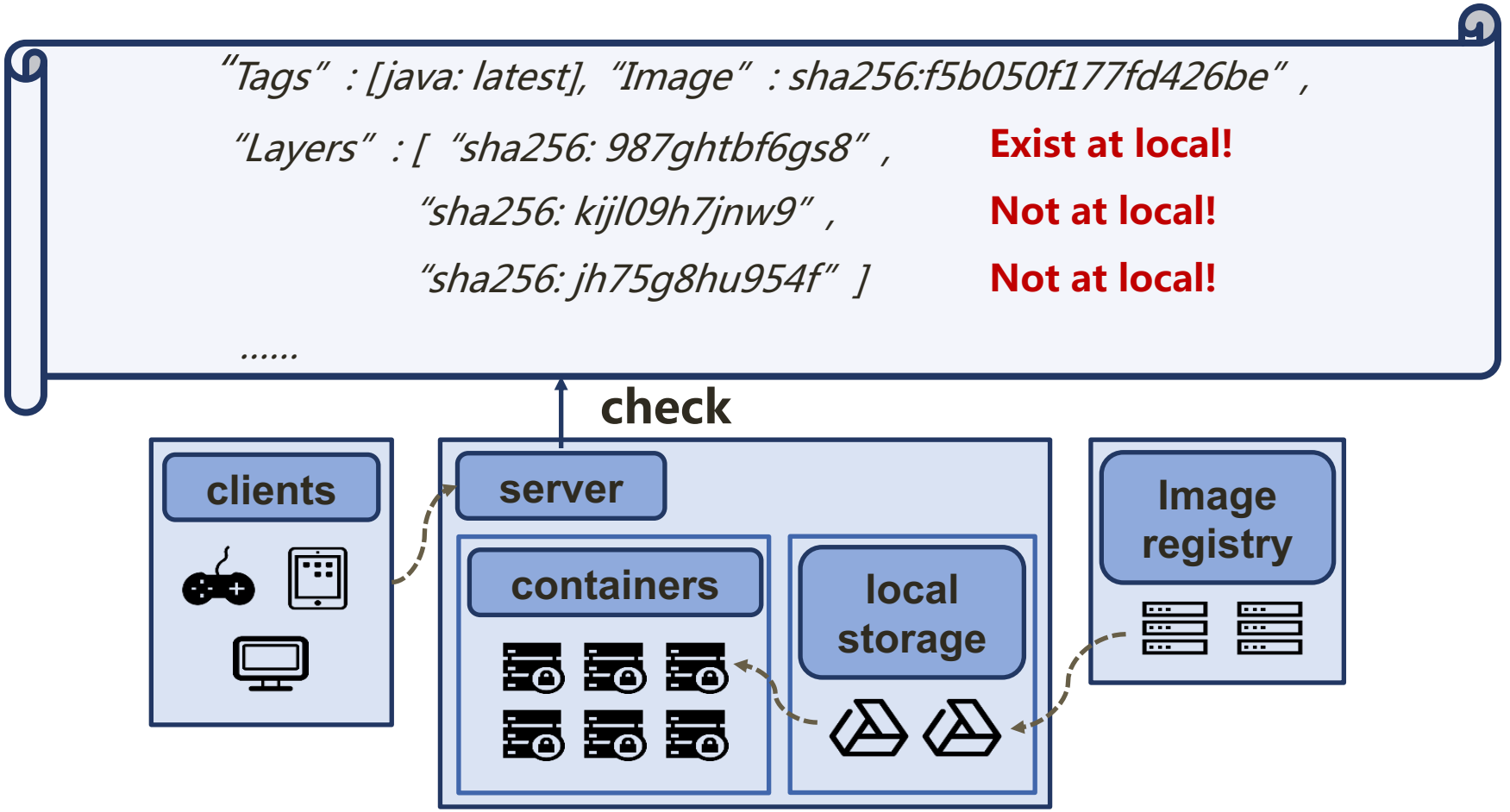




How to relieve the burden of storage and network caused by container images?

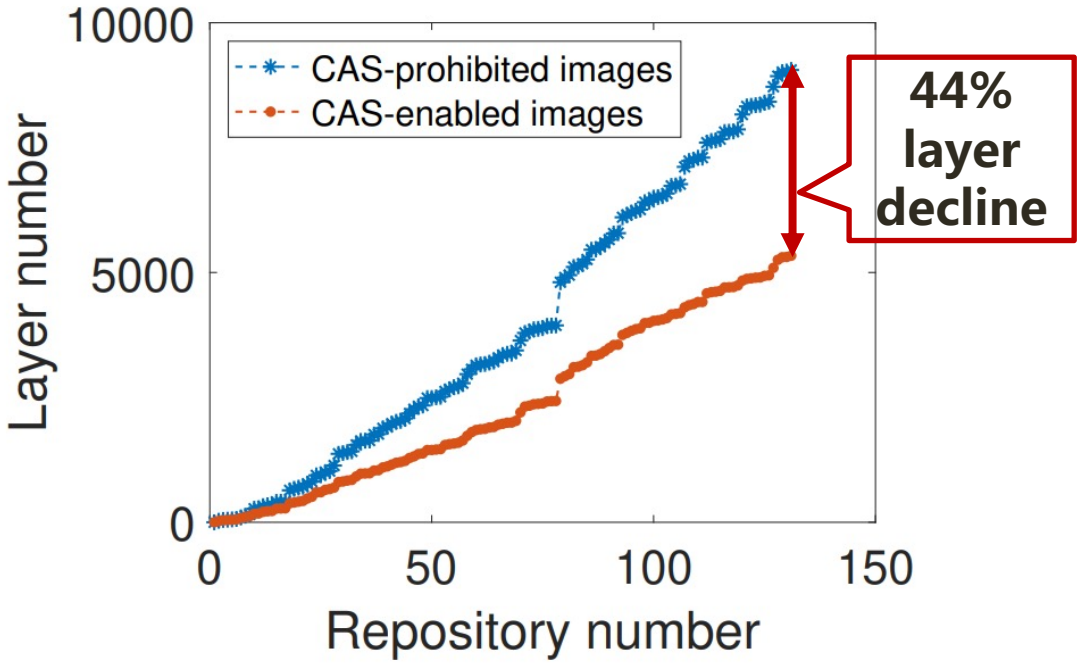
Content Addressable Storage (CAS)

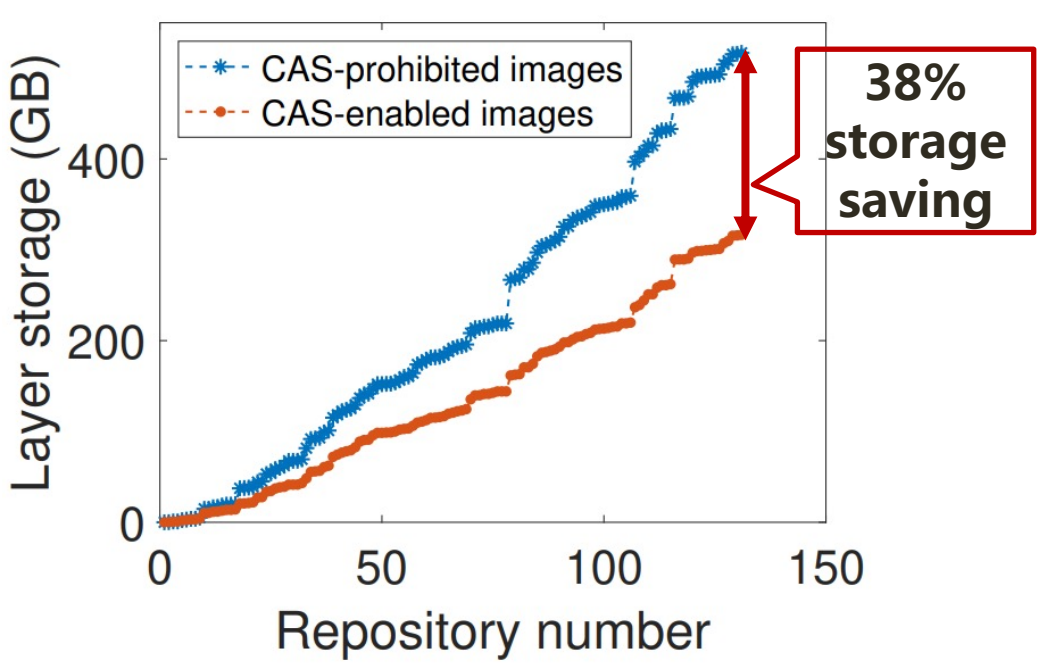
- CAS used in Docker **benefits to both storage and network!**
- CAS allows **identical layer sharing** among different images at the local storage.



Evaluation of Content Addressable Storage

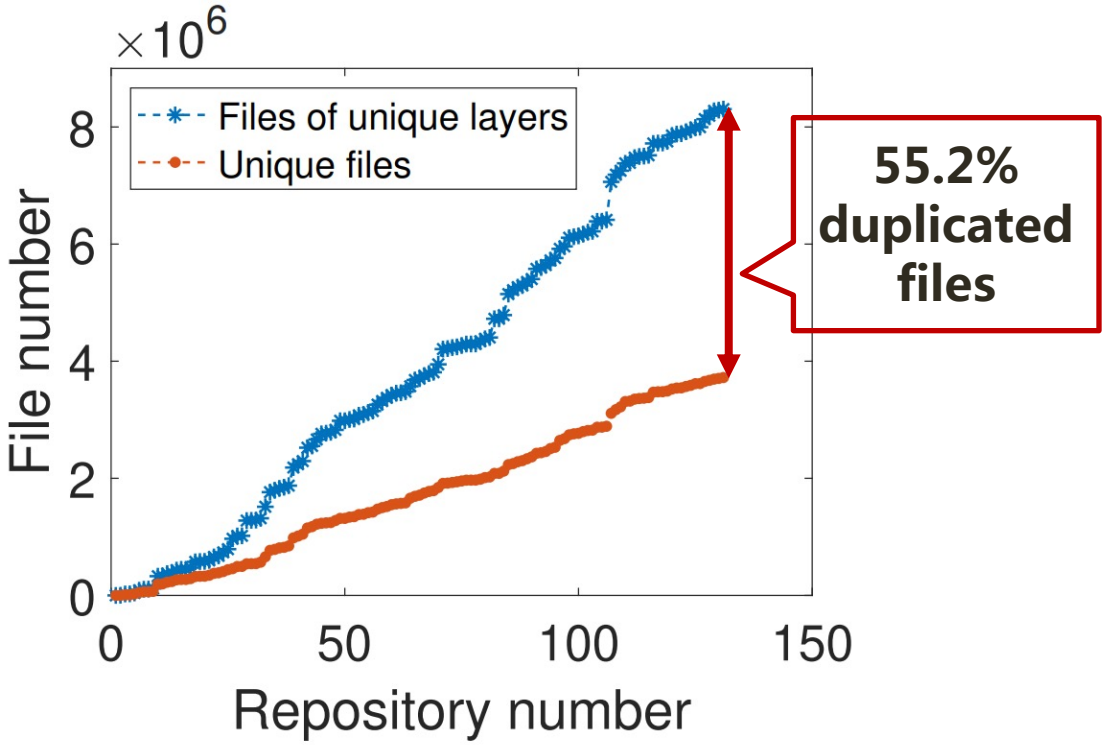
- **ImageNet: 2,200 images including 8,305,000 files among the most popular 130 repositories from Docker Hub.**

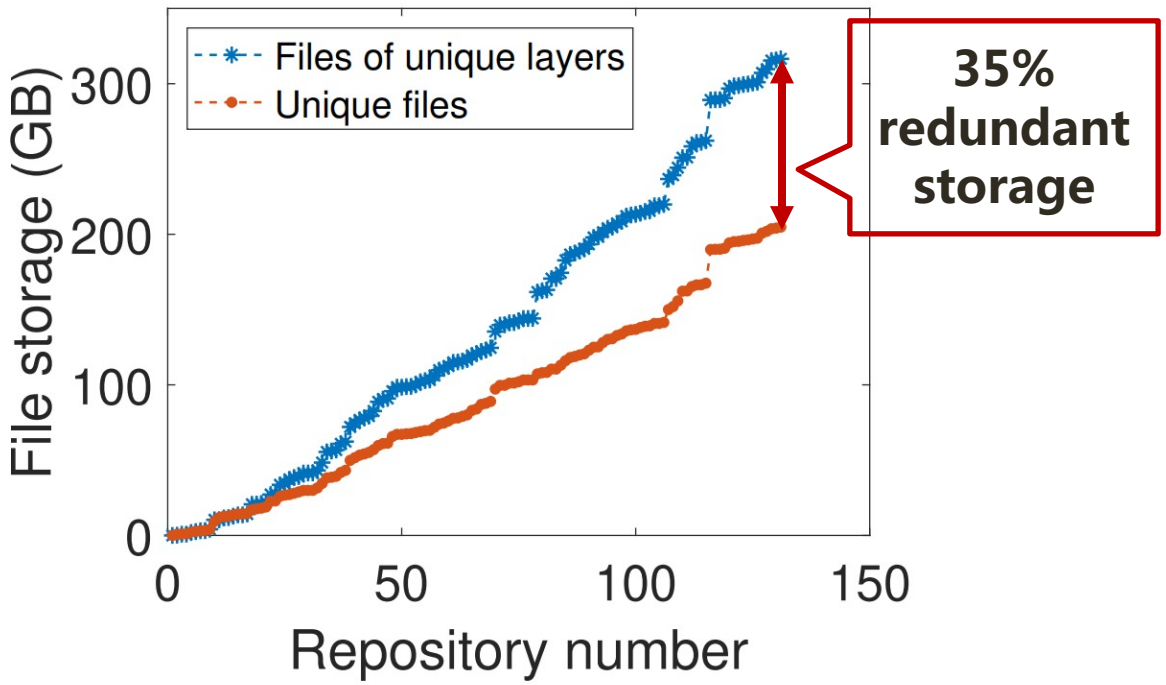




- **The saving gets larger with the increase of repositories.**

Image File Redundancy





- **The redundancy gets larger with the increase of repositories.**

Layer sharing enables partial files reused, while can not eliminate all the file redundancy. The potential of layer sharing remains to be explored.

Image Layer Similarity

How many layers are similar?

Layer similarity: the level of file redundancy between layers

Quantification metrics:

(1) Jaccard index 0 → 1 (More identical files)

(2) Deduplication ratio 1 → 2 (More redundant storage)

- **Not all layers have identical files. (Jaccard index = 0)**
- **Partial layers have high proportion of identical files. (Jaccard index > 0.7)**

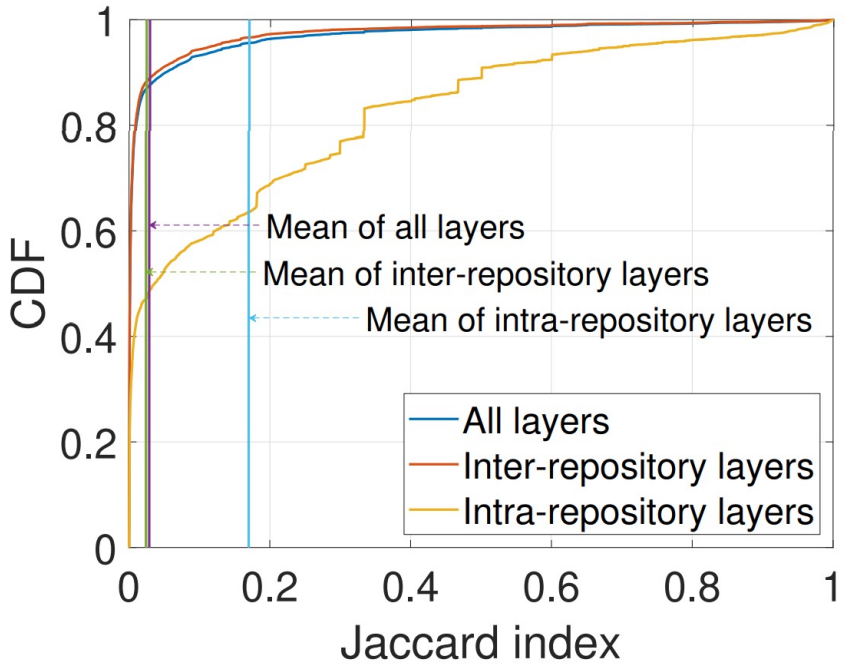
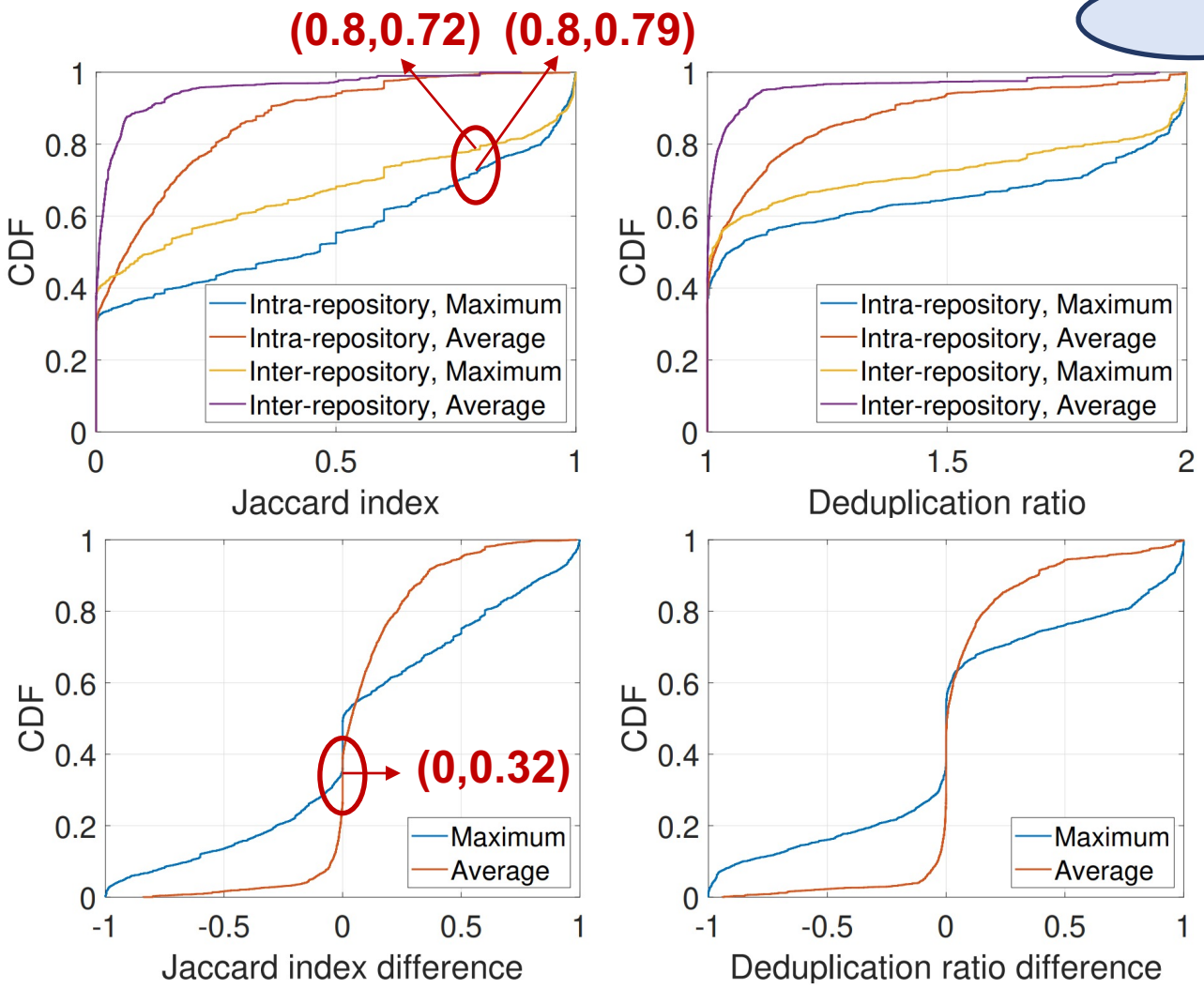


Image Layer Similarity

How much similar are layers within repository and across repositories?



Metric difference of the layer: maximum/average metric within repository - maximum/average metric across repository

- **Layers within the same repositories are more similar than layers across the repositories.**

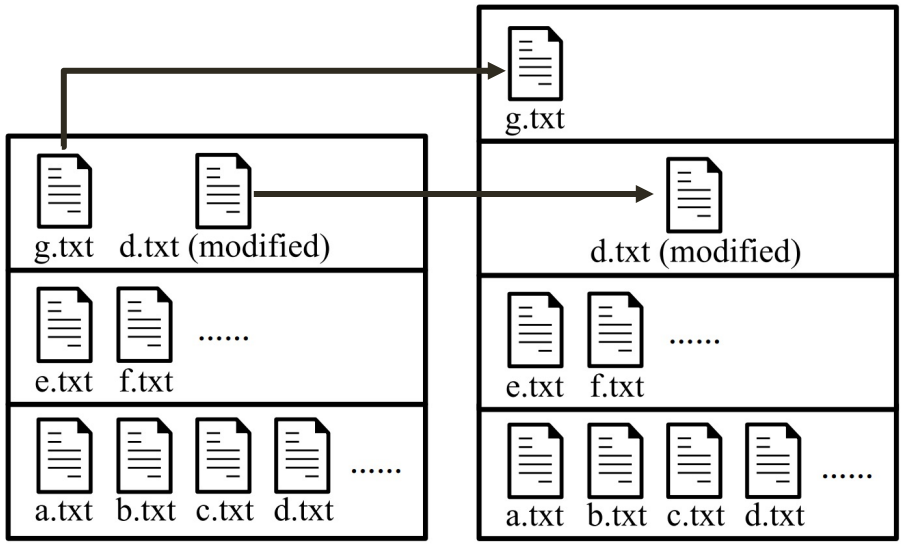
Image Reconstruction

Limited layer sharing

High similarity of layers

Image reconstruction:

Regrouping the files to create identical layers towards effective layer sharing.



An example of image reconstruction

Questions:

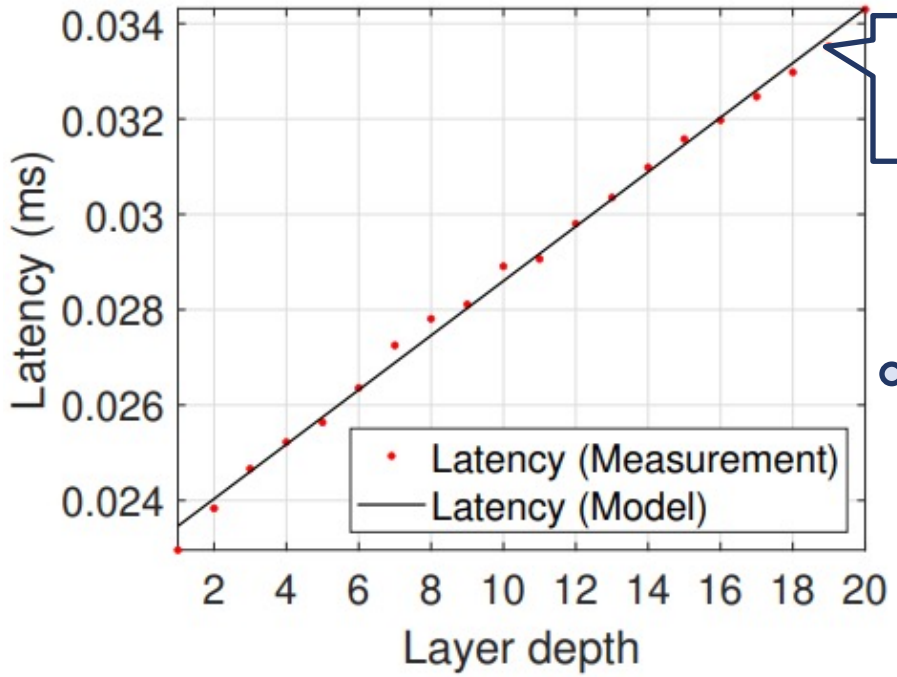
How many layers in the reconstructed image?

Which layer does each file belong to?

What is the order of layers?

Challenge#1

- **How many layers in the image affects the operation latency.**



Latency = 0.001526× Layer depth + 0.3087

How to trade off the storage and the operation overhead?

Operation and Storage Weighted Cost

Optimization goal: min Cost = α · operation cost + β · storage cost

- **Operation cost : summary latency of each layer**

$$C_o^i = \sum_{D=1}^{N} \mu(D) = \sum_{D=1}^{N} (a \cdot D + b) = a \cdot \frac{N(N+1)}{2} + b \cdot N.$$

Latency of each layer

- **Storage cost: incremental storage of unique layers**

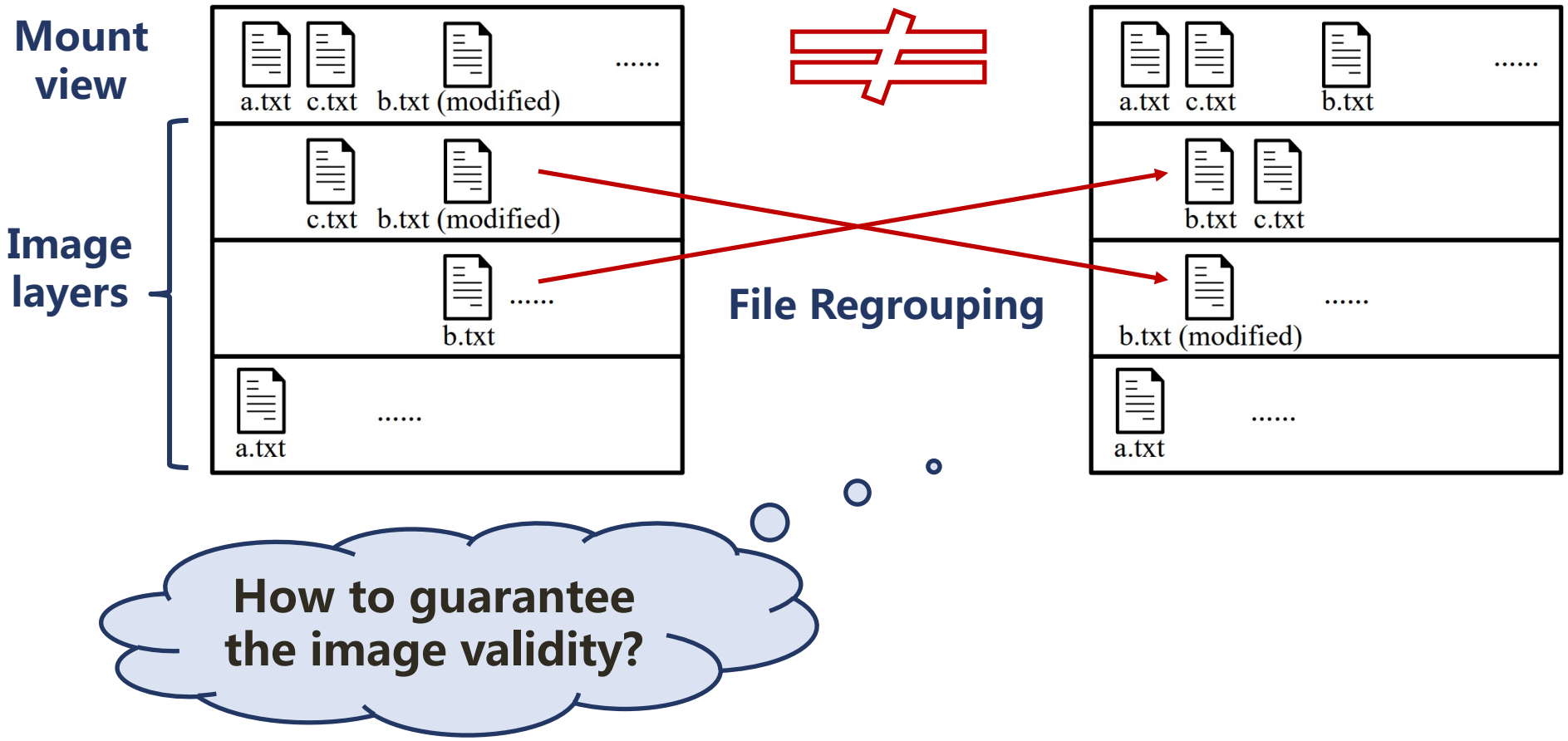
$$C_s^i = \sum_{n \in N} \left\{ \sum_{k \in M} (\xi_{k,n} \cdot S_k) \cdot min\left(\prod_{j \in J} \sum_{k \in M \cup K} |\xi_{k,n} - y_{k,j}|, 1 \right) \right\}.$$

Layer size

Whether the layer is unique

Challenge#2

- Partial files have dependency. (Such as b.txt (modified) and b.txt.)
- File location and layer order affect the mount view of image and make it invalid.



Commutativity Model

- **If two files have dependency, they are defined as noncommutative.**
- **The relative layer position of noncommutative file pairs cannot be changed.**

Commutativity constraint:

Commutativity of file k and k'

$$\delta_{k,k'} \cdot \lambda_{k,k'} = \delta^*_{k,k'} \cdot \lambda_{k,k'}$$

Relative position of file k and k'

Relative position after reconstruction

Image
Layer3: c.txt, b'.txt (modified on b.txt), …
Layer2: b.txt, …
Layer1: a.txt, …

E.g.:

The b.txt and b'.txt are noncommutative: $\lambda_{b,b'} = 1$.

The relative position of b.txt and b'.txt: $\delta_{b,b'} = -1$.

Then $\delta^*_{b,b'} = -1$. The file b'.txt must be in the upper layer to the file b.txt.

Problem Formulation

Objective: Minimize the weighted operation cost and storage cost

$$\mathbf{P}: \min_{N,\Xi} C = \alpha C_o^i + \beta C_s^i.$$

$$s.t.\ C1: sgn(\theta_k^i) = \sum_{n=1}^{N} \xi_{k,n}, \forall k \in M \cup K.$$

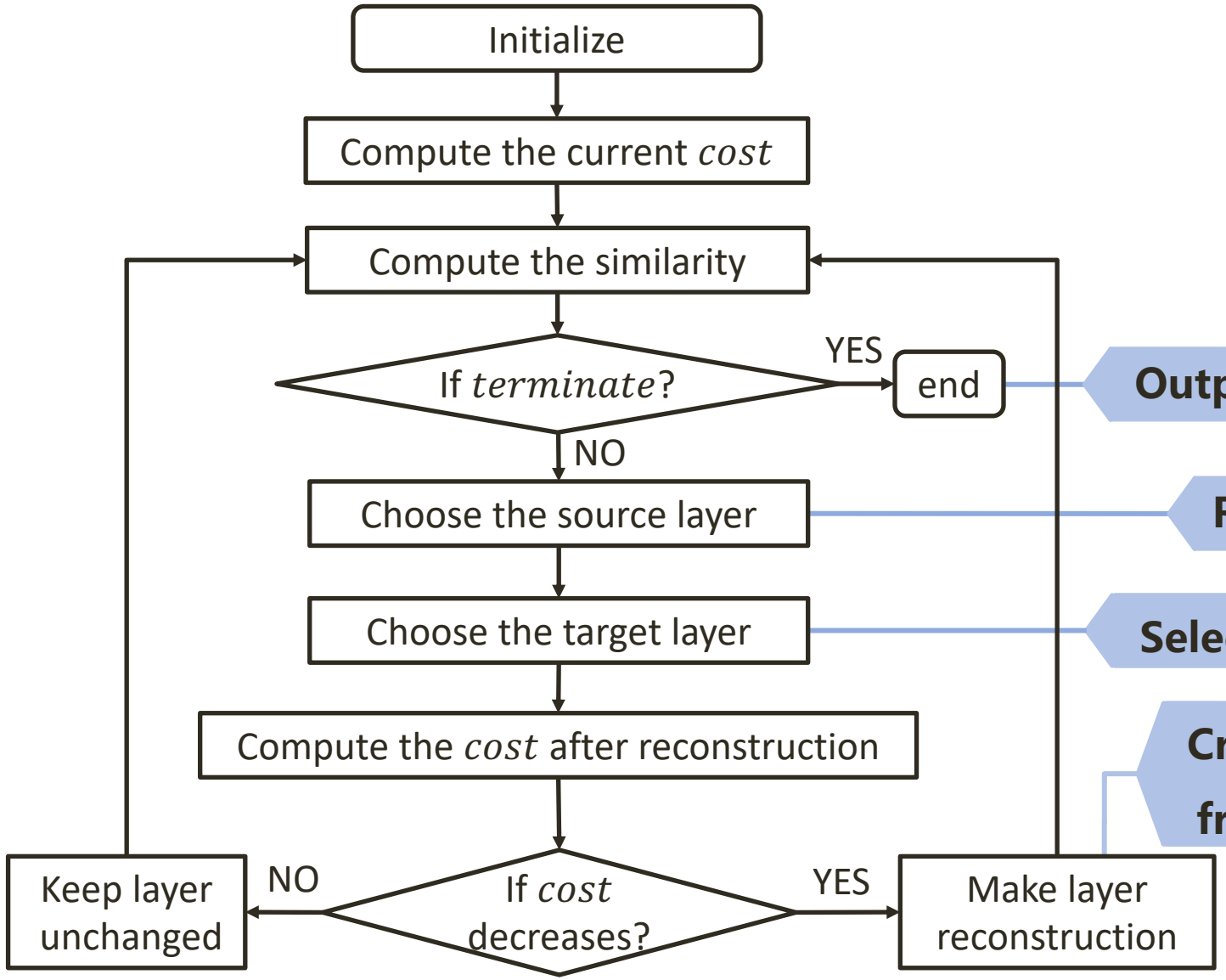
Constraint 1: file consistency

$$C2: \delta_{k,k'}^i \cdot \lambda_{k,k'}^i = sgn(\sum_{n=1}^{N} n\xi_{k,n} - \sum_{n=1}^{N} n\xi_{k',n}) \cdot \lambda_{k,k'}^{i_0},$$

$$\forall k, k' \in M.$$

Constraint 2: file commutativity

Similarity-aware Image Reconstruction Algorithm



Output metadata of the reconstructed image

Regroup files for each layer

Select target layer only within the repository

Create sharing layer with the identical files from the source layer and the target layer

Evaluation

- **Settings**

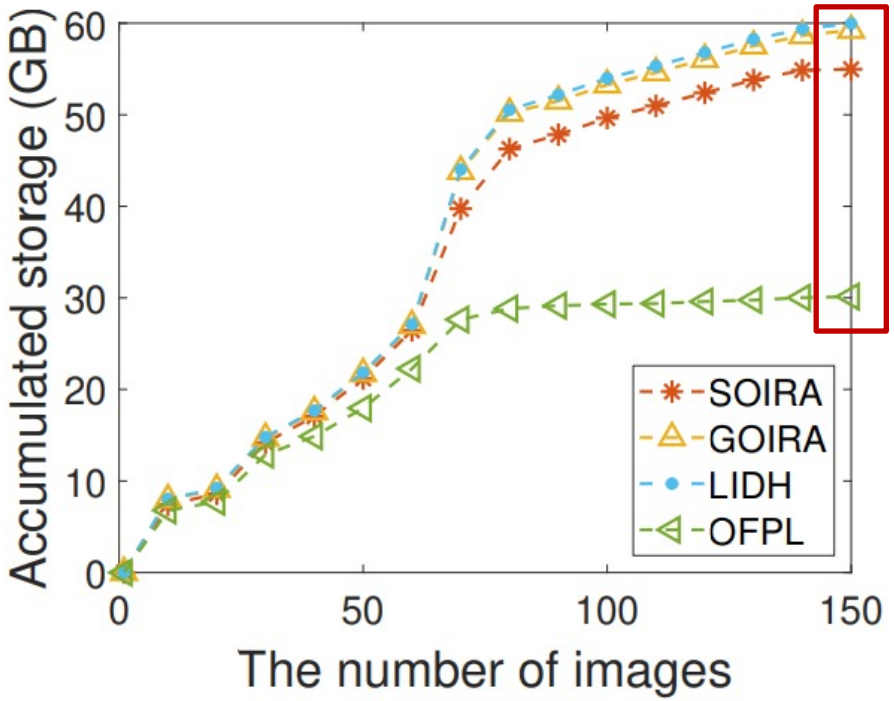
  
  Server: eight core Intel Xeon Processor, 2.5GHz, 11GB RAM

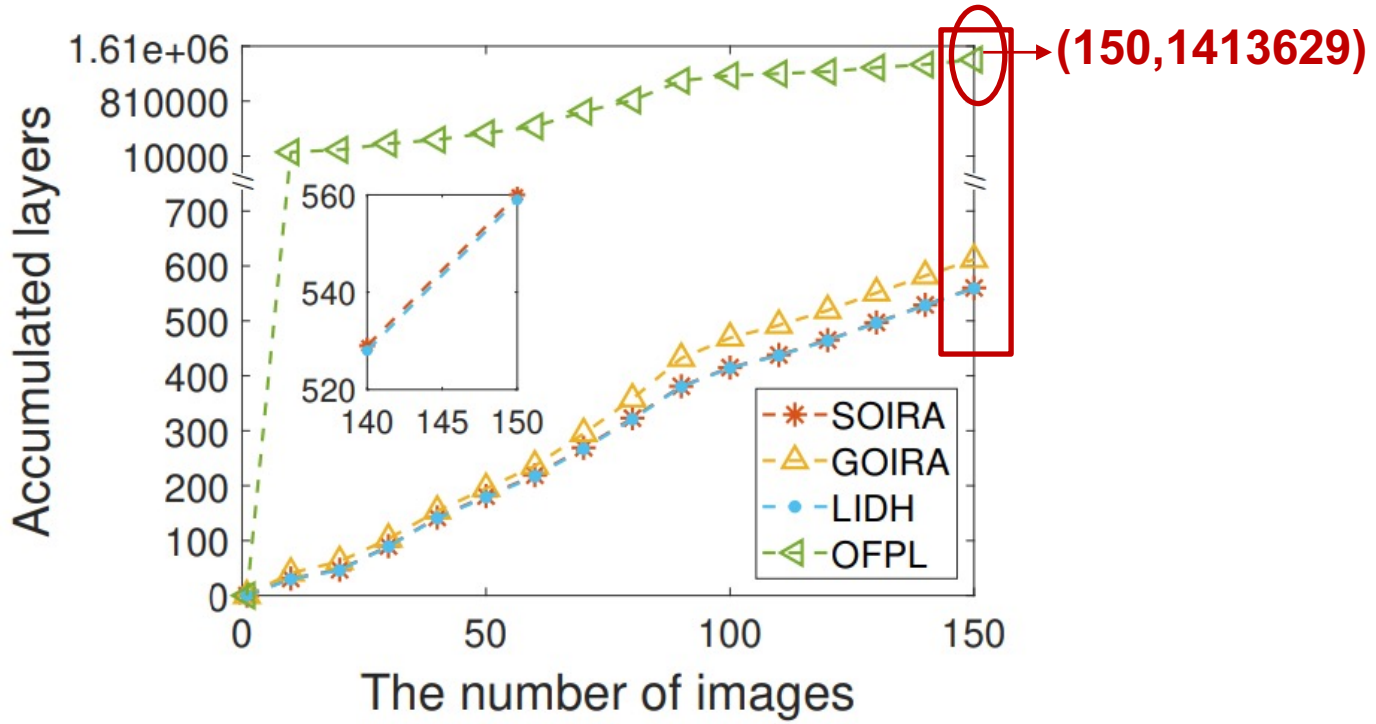
  
  dataset: ImageNET

- **Baselines**

  1. **Greedy Offline Image Reconstruction Algorithm (GOIRA).**
  2. **Layered Images from Docker Hub (LIDH).**
  3. **One-File-Per-Layer (OFPL).**

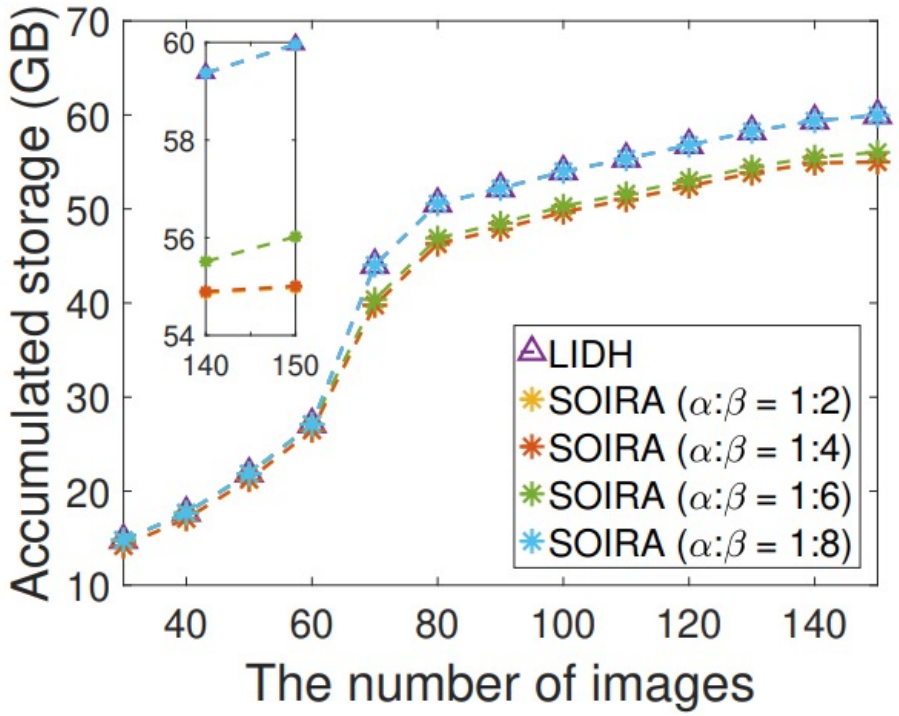
Storage and Operation Overhead

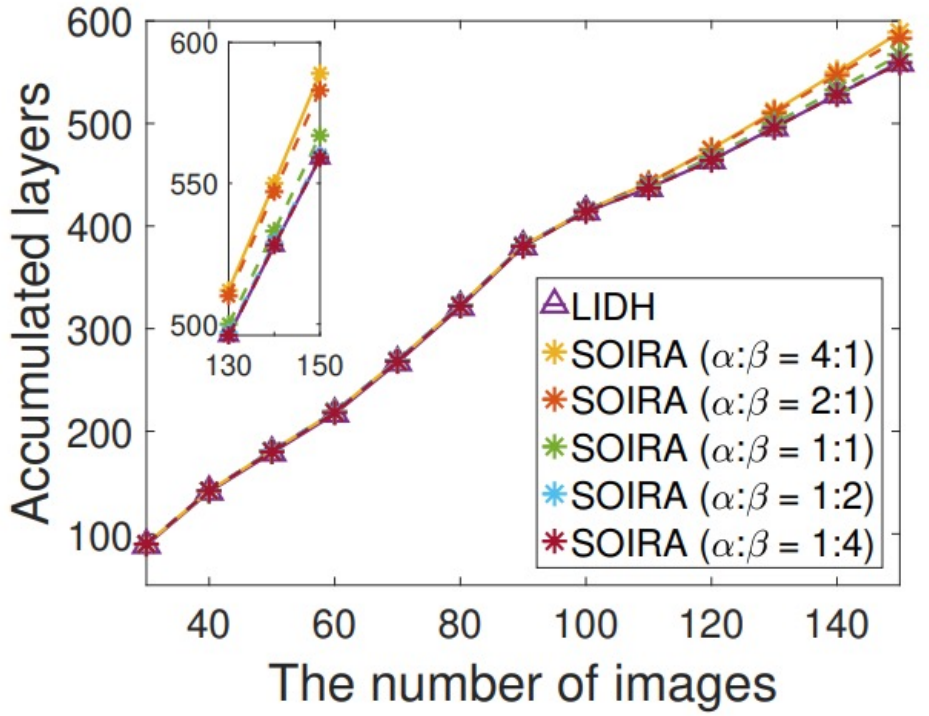




- **OFPL: The least layer storage, the worst performance in layer number.**
- **GOIRA: 1.3% storage saving, 7.3% increase of layer number.**
- **SOIRA: 10% storage saving, single-digit growth of layer number.**

Storage and Operation Overhead



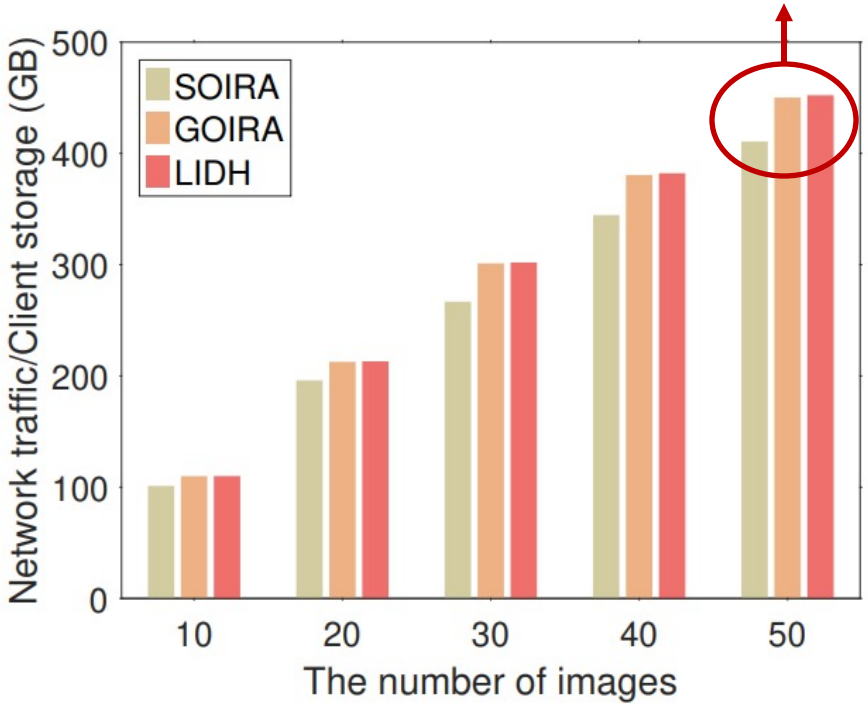


- With **higher α:β** , SOIRA prefers **lower storage consumption**.
- Different storage saving can be achieved by adjusting α:β.

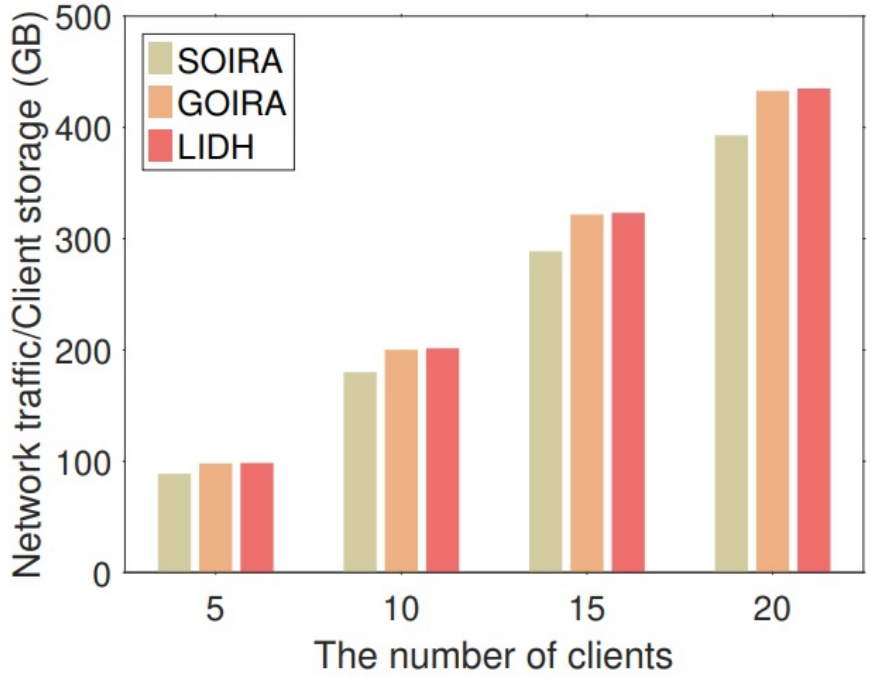
Network Traffic and Client Storage

8.3% saving compared with GOIRA,

8.8% saving compared with LIDH.



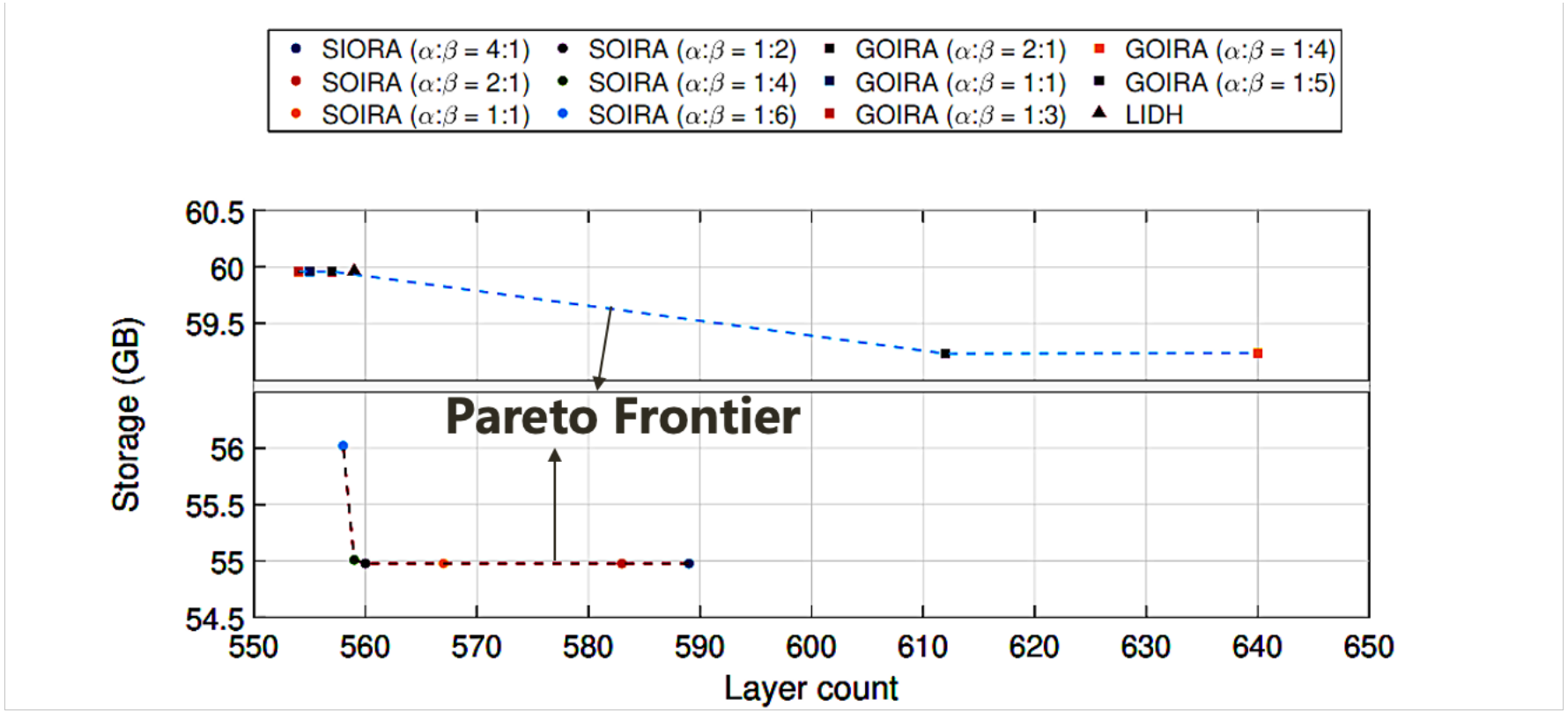
Client number is 10.



Each client downloads 20 images.

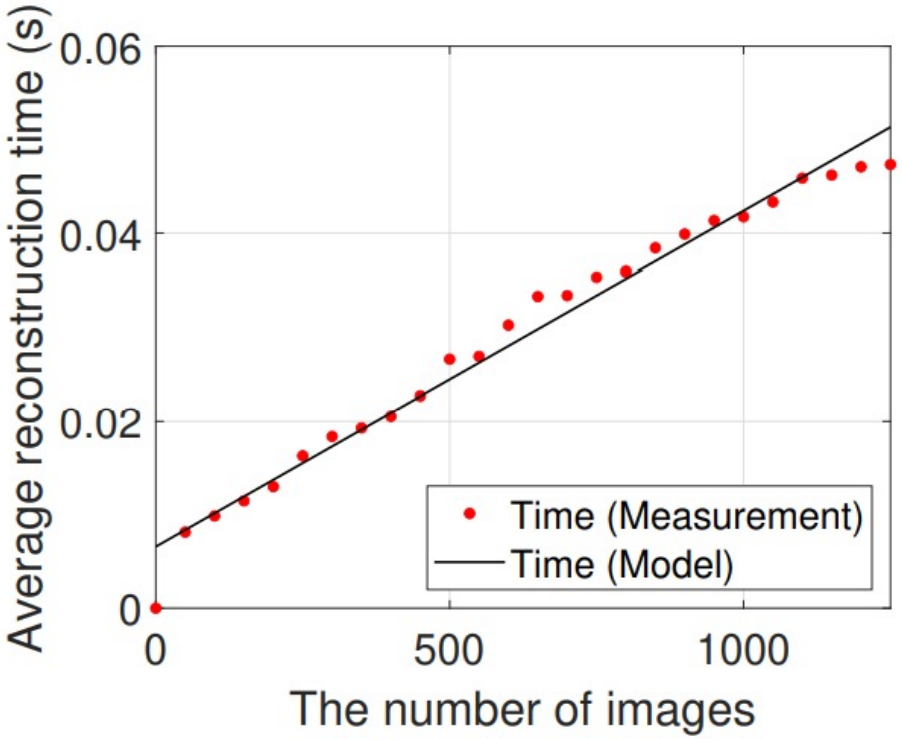
- **More clients or downloaded images, more traffic saving.**

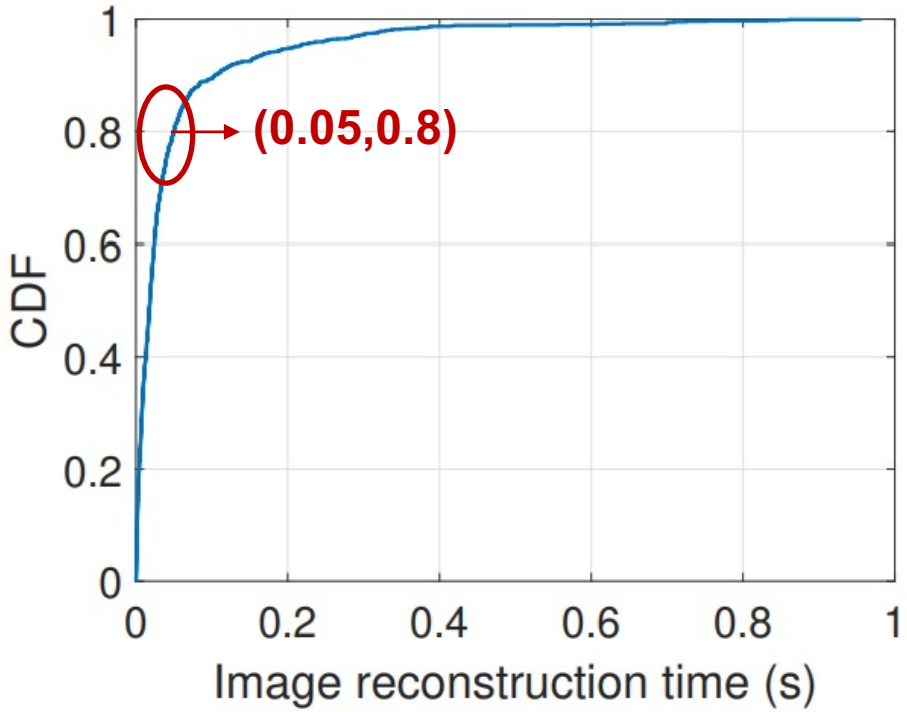
Storage and Operation Overhead Trade-off



- The layer count and storage run in the opposite direction.
- An **improved Pareto frontier** and better performance is achieved by our SOIRA.

Image Reconstruction Time





- **The time goes through a slight and tolerable time increase with the expansion of the image number.**
- **Around 80% images consume reconstruction time less than 0.05s.**

Summary

- Evaluation of content addressable storage to reveal the file redundancy.
- Quantification and measurement of layer similarity.
- Image reconstruction towards effective layer sharing to save both storage and network resource consumption.

Thanks for your listening!

Q&A

Sisi Li (sisili@bupt.edu.cn)