USENIX ATC '24

SimEnc

A High-Performance Similarity-Preserving Encryption Approach for Deduplication of Encrypted Docker Images

Tong Sun[1], Bowen Jiang[1], Borui Li[2], Jiamei Lv[1], Yi Gao[1], and Wei Dong[1]

[1]The State Key Laboratory of Blockchain and Data Security,

College of Computer Science & School of Software Technology, Zhejiang University, China

[2]School of Computer Science and Engineering, Southeast University, China

浙江大學 区块链与数据安全全国重点实验室

STATE KEY LABORATORY OF BLOCKCHAIN AND DATA SECURITY

ZHEJIANG UNIVERSITY

東南大學 SOUTHEAST UNIVERSITY

Background

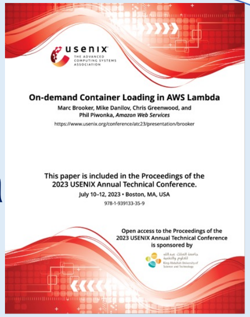
- **Encrypted container images are becoming increasingly popular in registries for privacy**

Encrypting images for content confidentiality in Container Registry

Last updated 2024-05-02

IBM

[ATC'23]
AWS Lambda


On-demand Container Loading in AWS Lambda
Marc Brooker, Mike Danilov, Chris Greenwood, and Phil Piwonka, *Amazon Web Services*
https://www.usenix.org/conference/atc23/presentation/brooker

This paper is included in the Proceedings of the 2023 USENIX Annual Technical Conference.
July 10–12, 2023 • Boston, MA, USA
978-1-939133-35-9

Open access to the Proceedings of the 2023 USENIX Annual Technical Conference is sponsored by

aws

Background

- **Huge storage cost**

Stored over 15PB of container images in 2020

Background

- **97% of Docker files across different layers are duplicated**[1]

[1] Large-scale Analysis of the Docker Hub Dataset. In Proc. of IEEE CLUSTER, 2019.

Background

- **97%** of Docker files across different layers are duplicated[1]

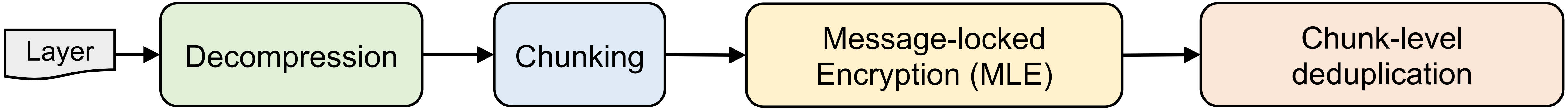
deduplication to save space

[1] Large-scale Analysis of the Docker Hub Dataset. In Proc. of IEEE CLUSTER, 2019.

Challenge

- **Deduplication for encrypted images is difficult**
  - Deduplication exploits identical content
  - Encryption makes all contents look random

Previous work

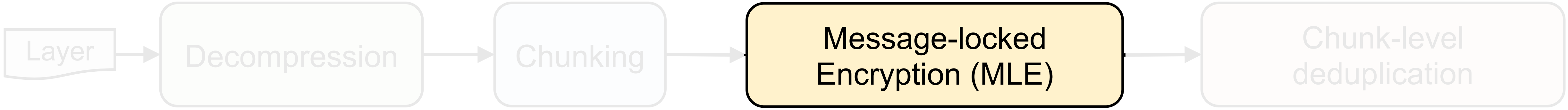


Previous work

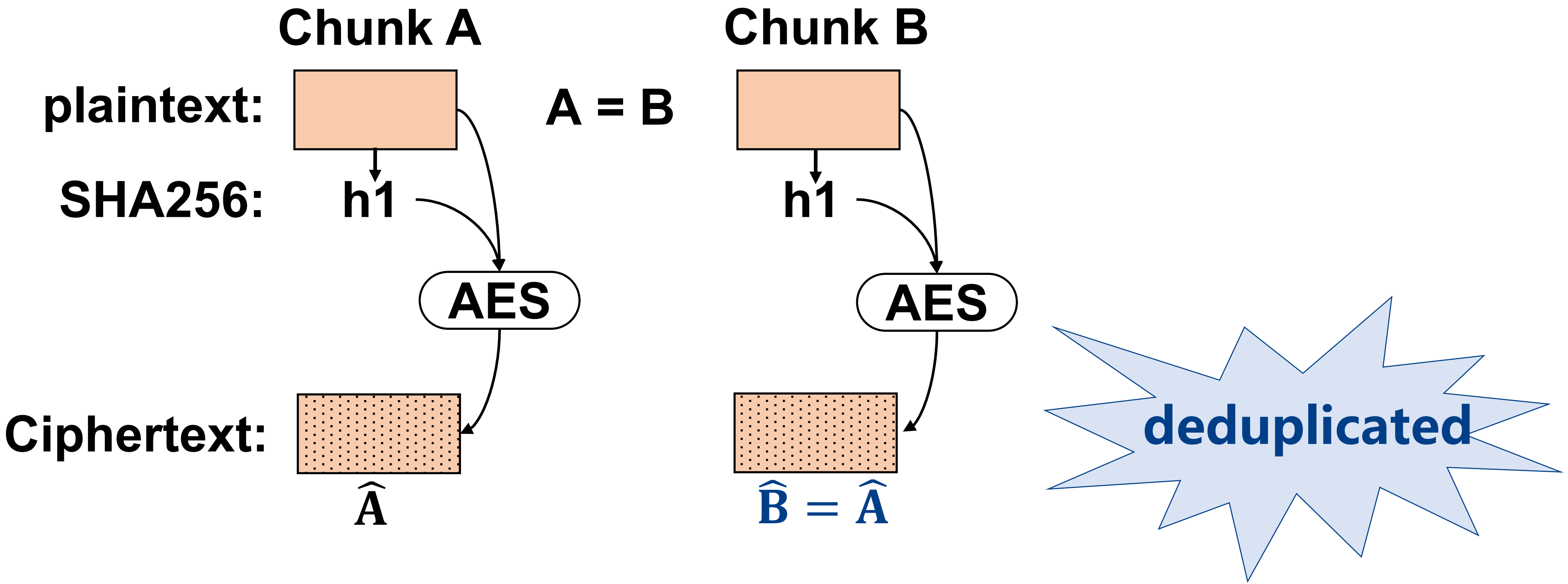
Layer → Decompression → Chunking → Message-locked Encryption (MLE) → Chunk-level deduplication

- **Message-locked Encryption (MLE)**
  - generates encryption keys from the content

Previous work

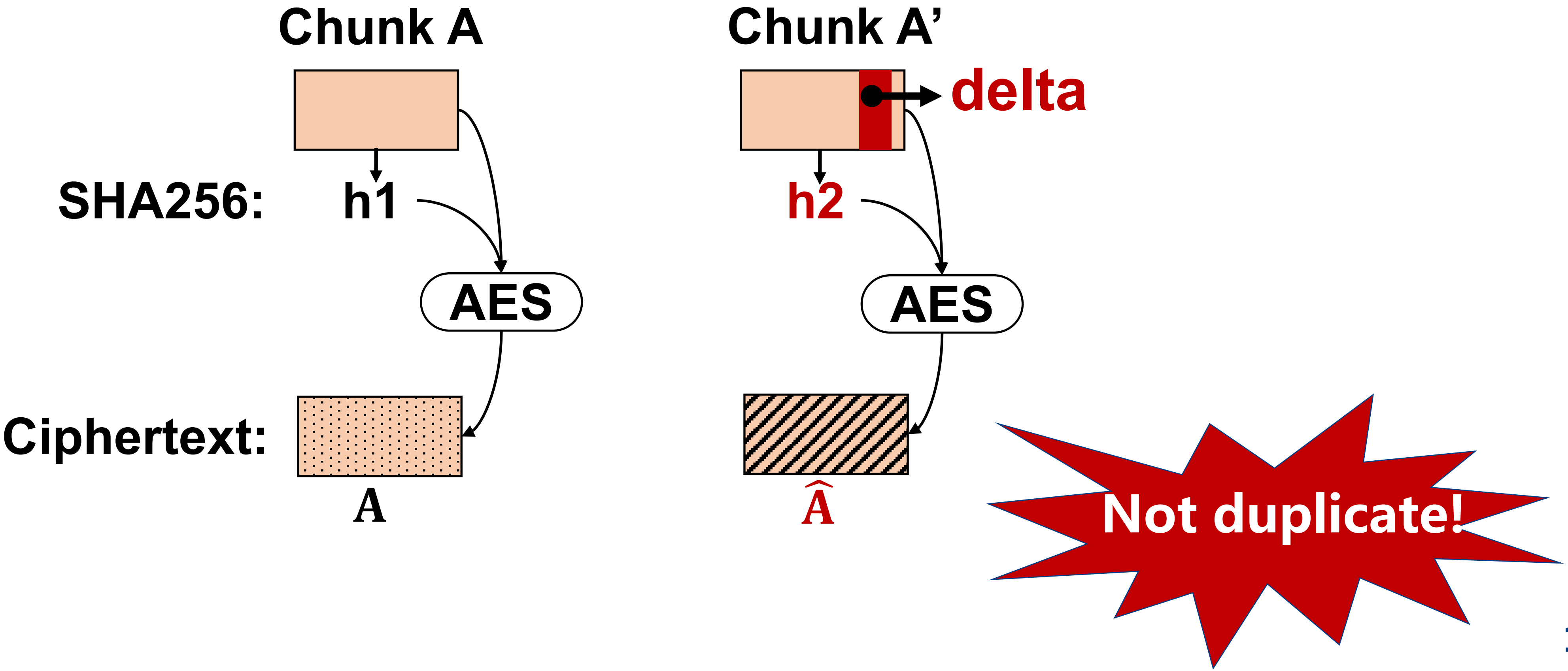


- **Message-locked Encryption (MLE)**
  - generates encryption keys from the content



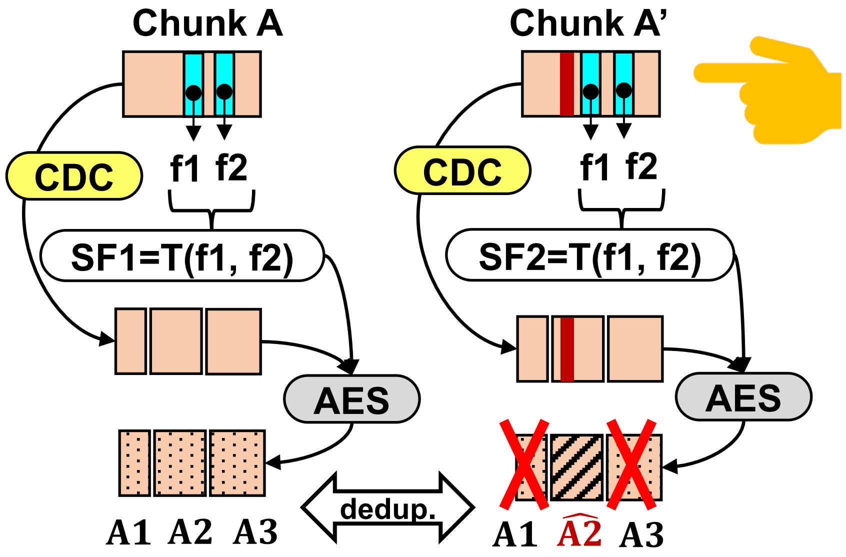
Observation #1

- **Even minor modifications to the image content can hinder MLE deduplication**



Observation #1

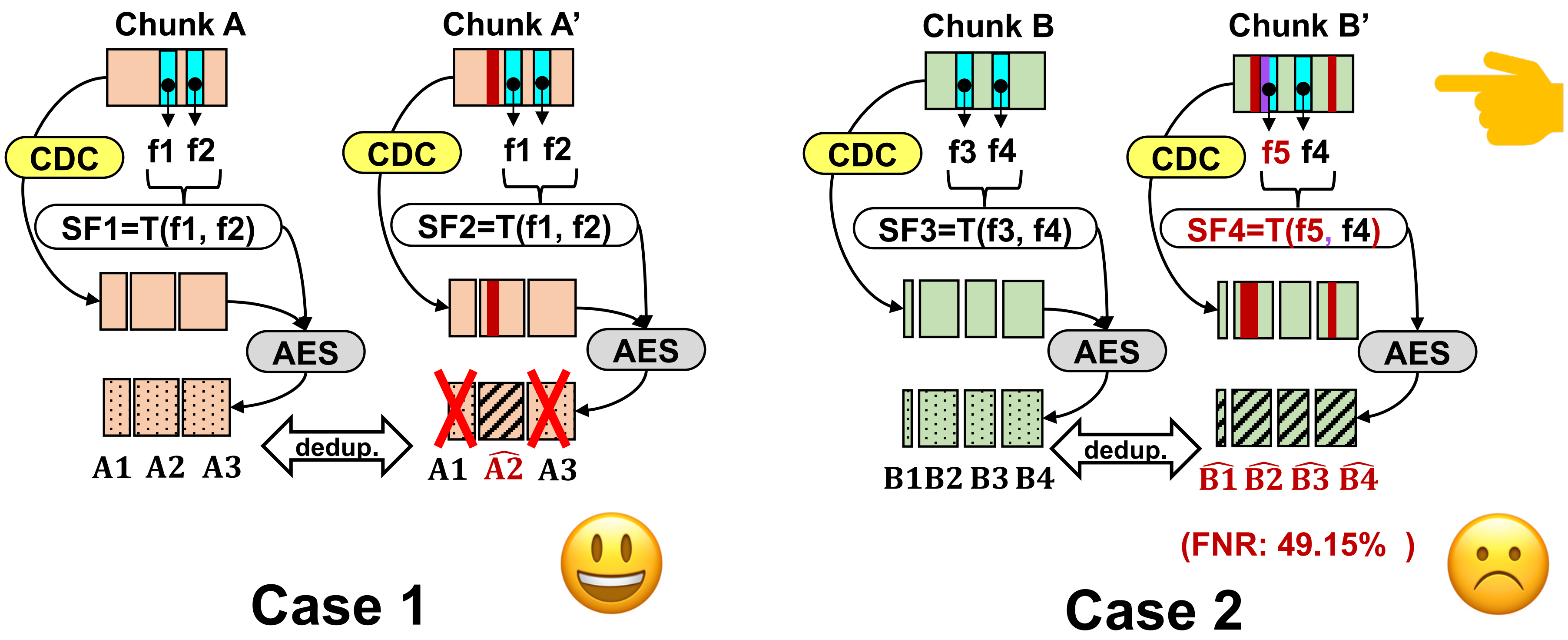
- **Locality-sensitive hashing (LSH)-based MLE (SOTA)**



Case 1

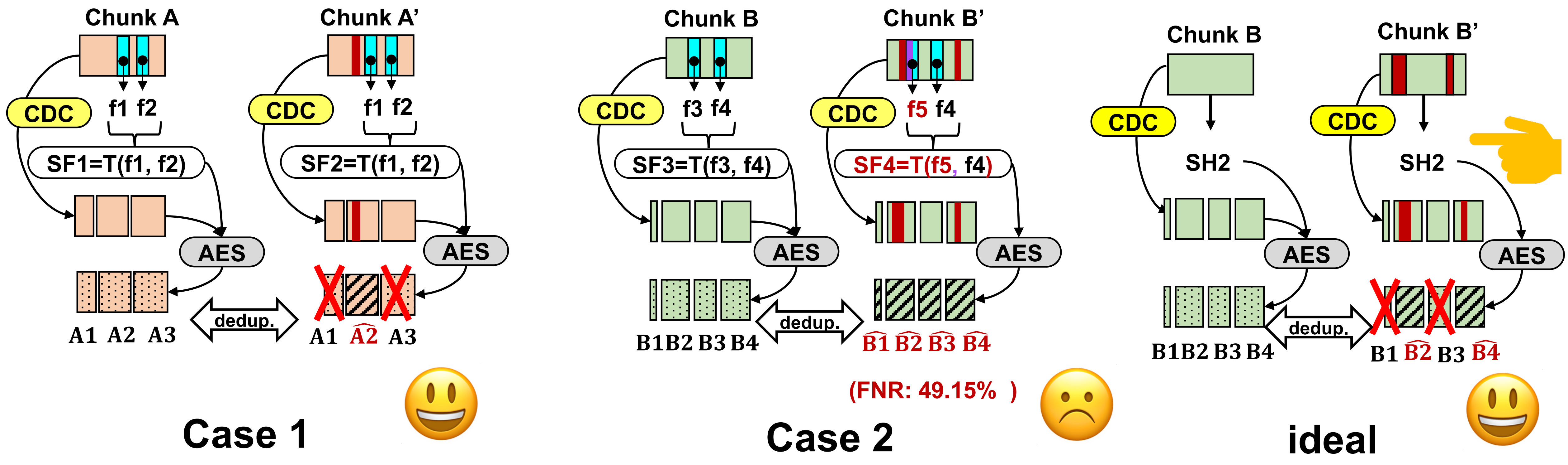
Observation #1

- **Locality-sensitive hashing (LSH)-based MLE (SOTA)**



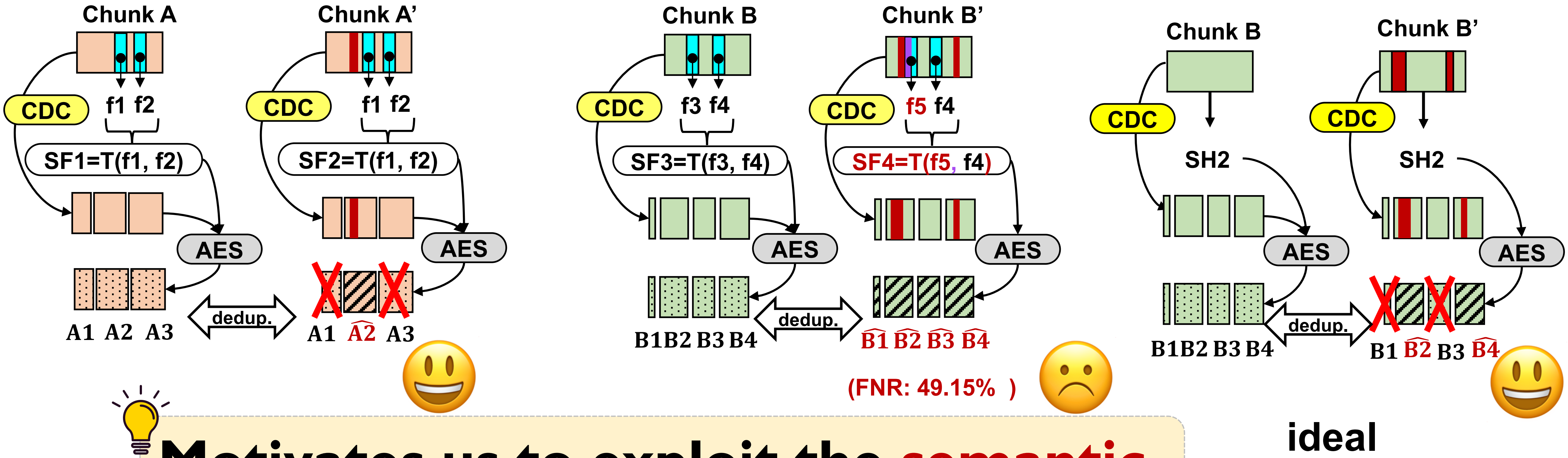
Observation #1

- **Locality-sensitive hashing (LSH)-based MLE (SOTA)**



Observation #1

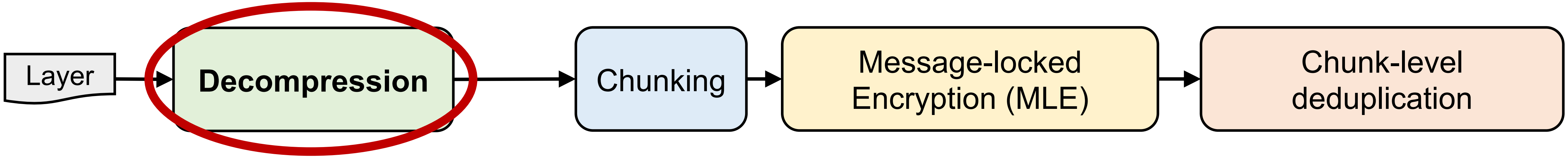
- **Locality-sensitive hashing (LSH)-based MLE**



Motivates us to exploit the semantic hashing technique

Observation #2

Deduplication stage

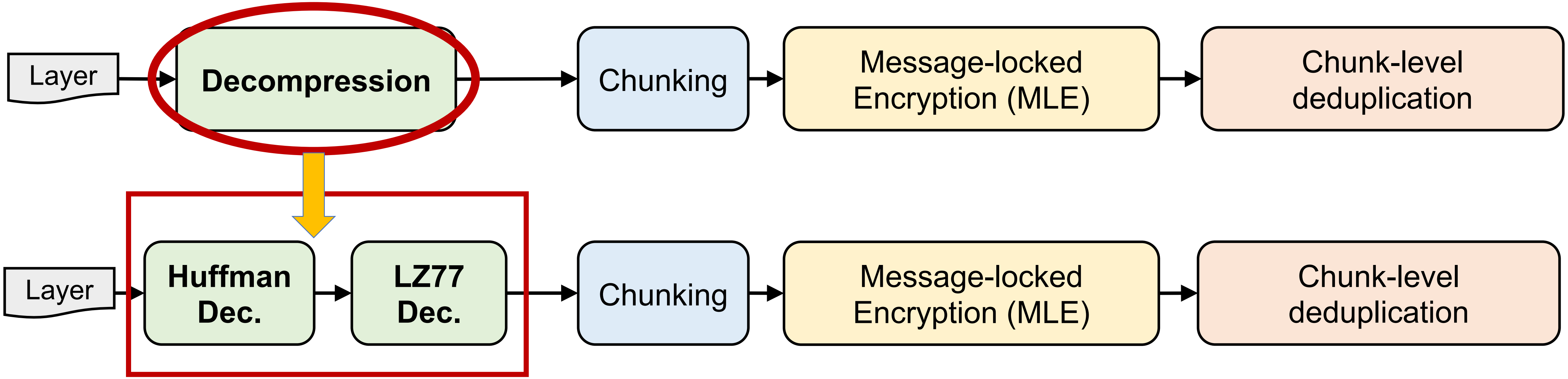


Two drawbacks

- Re-compression is needed to restore images, **increasing pull latency**
- Decompressing before deduplication **reduces throughput**

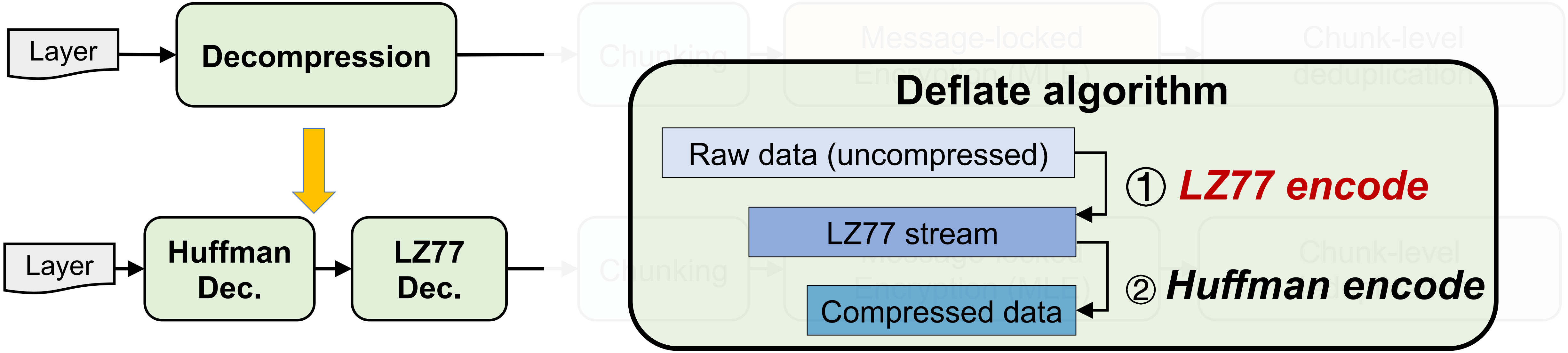
Observation #2

Deduplication stage

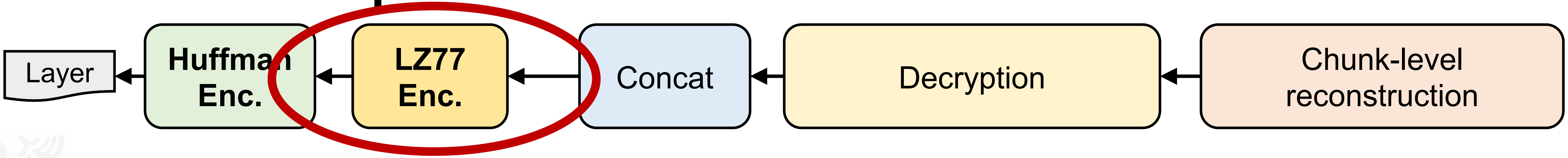


Observation #2

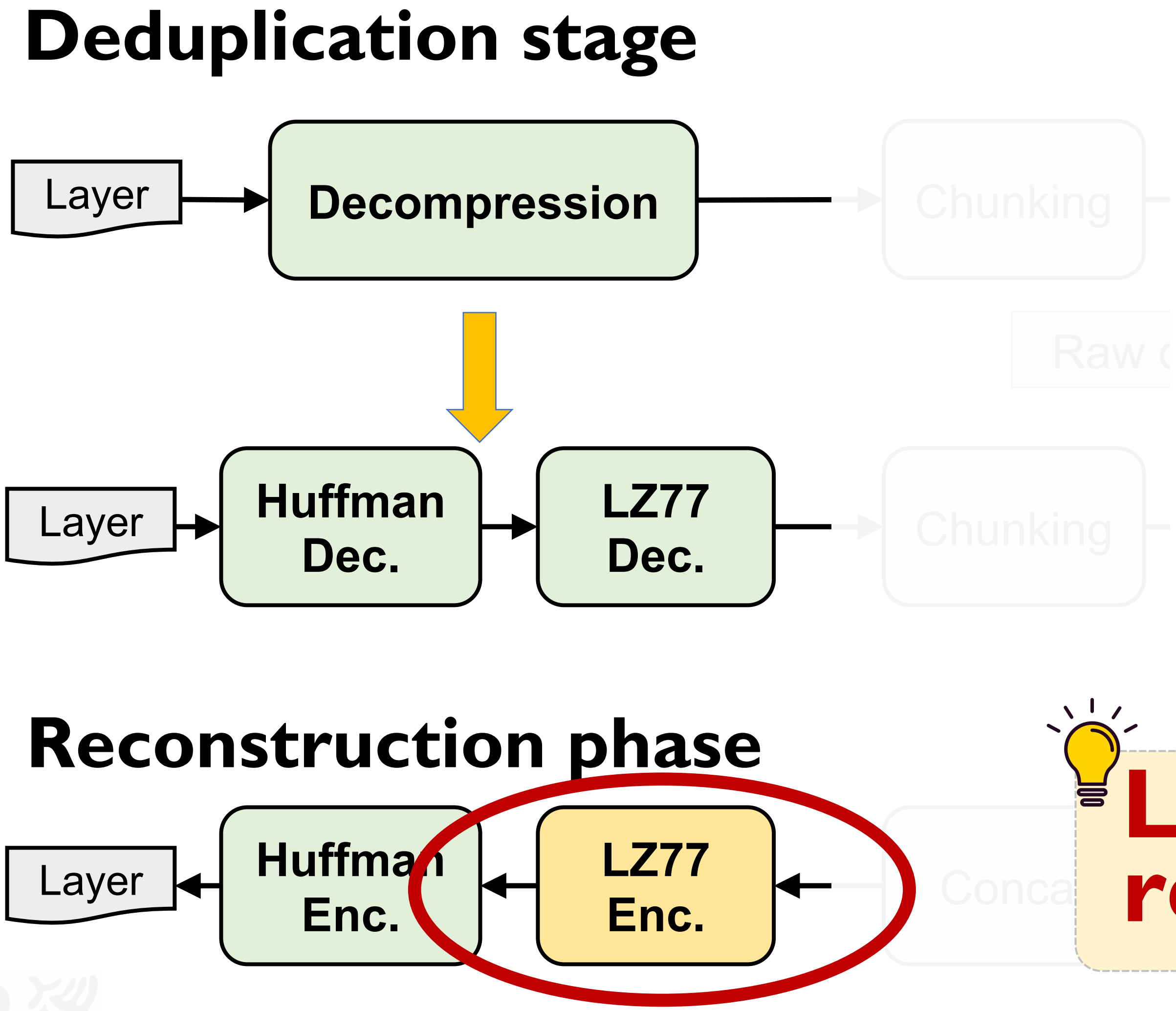
Deduplication stage

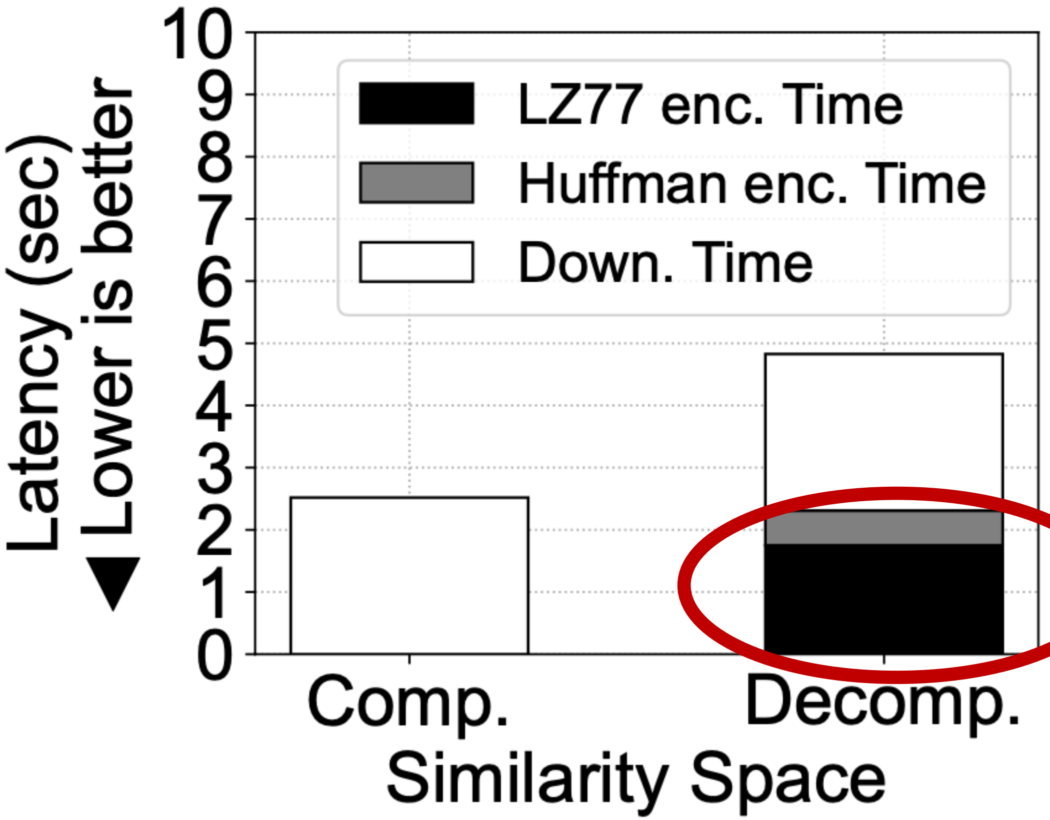


Reconstruction phase



Observation #2





LZ77 encoding dominates recompression time

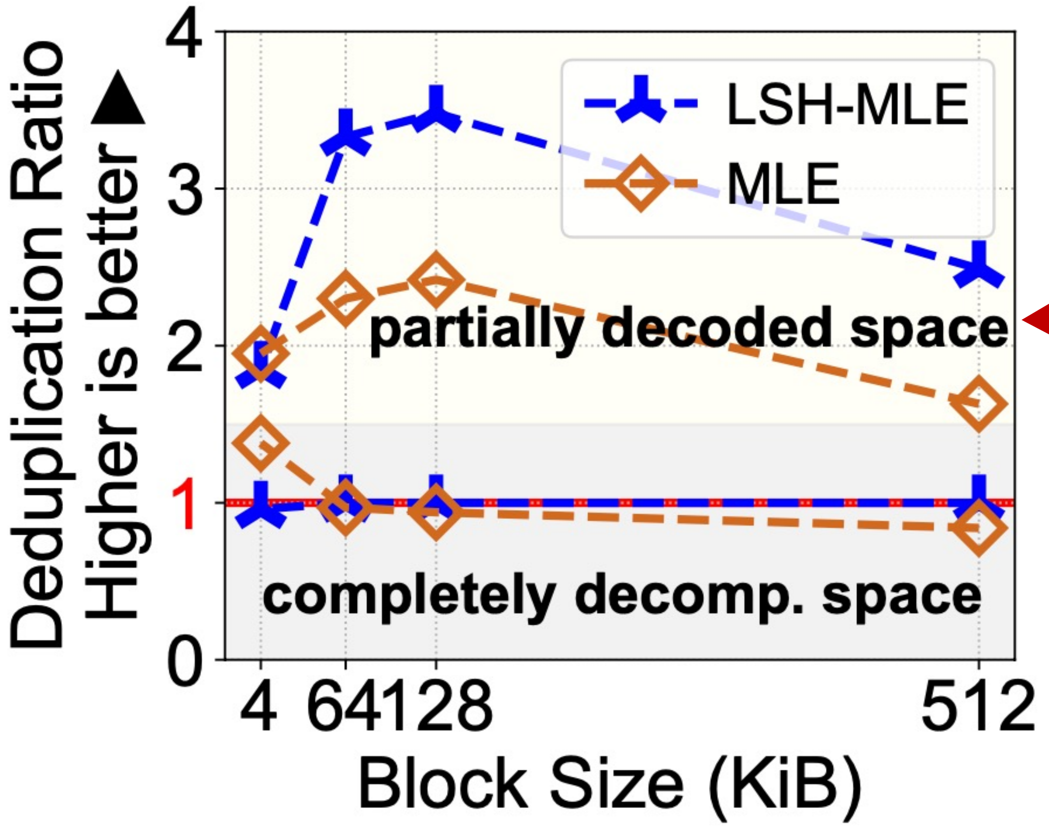
Observation #2

- **Can Docker images be deduplicated after Huffman decoding instead of completely decompression (Huffman dec.+ LZ77 dec.)?**

- **If yes, it can potentially reduce pull latency.**

Observation #2

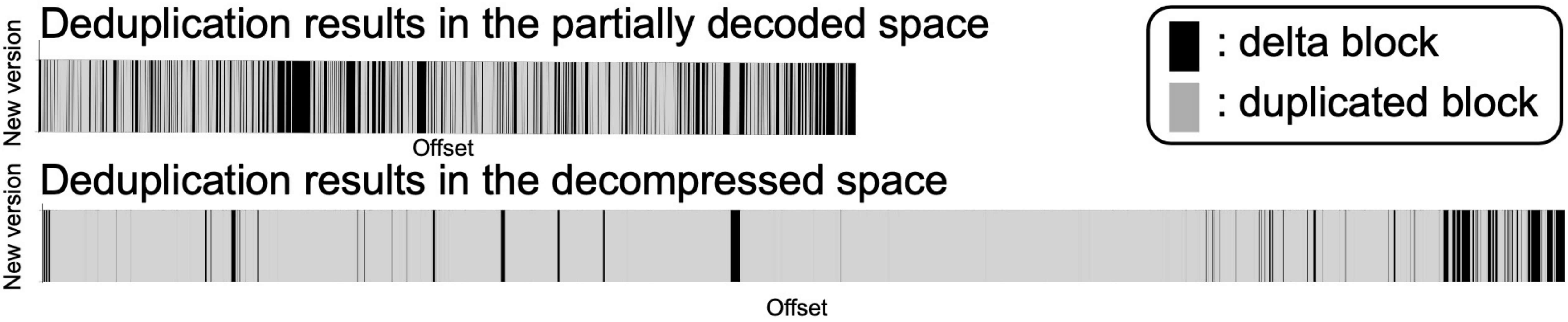
- **Can Docker images be deduplicated after Huffman decoding instead of completely decompression?**



Why?

Observation #2

- **Deduplication results**



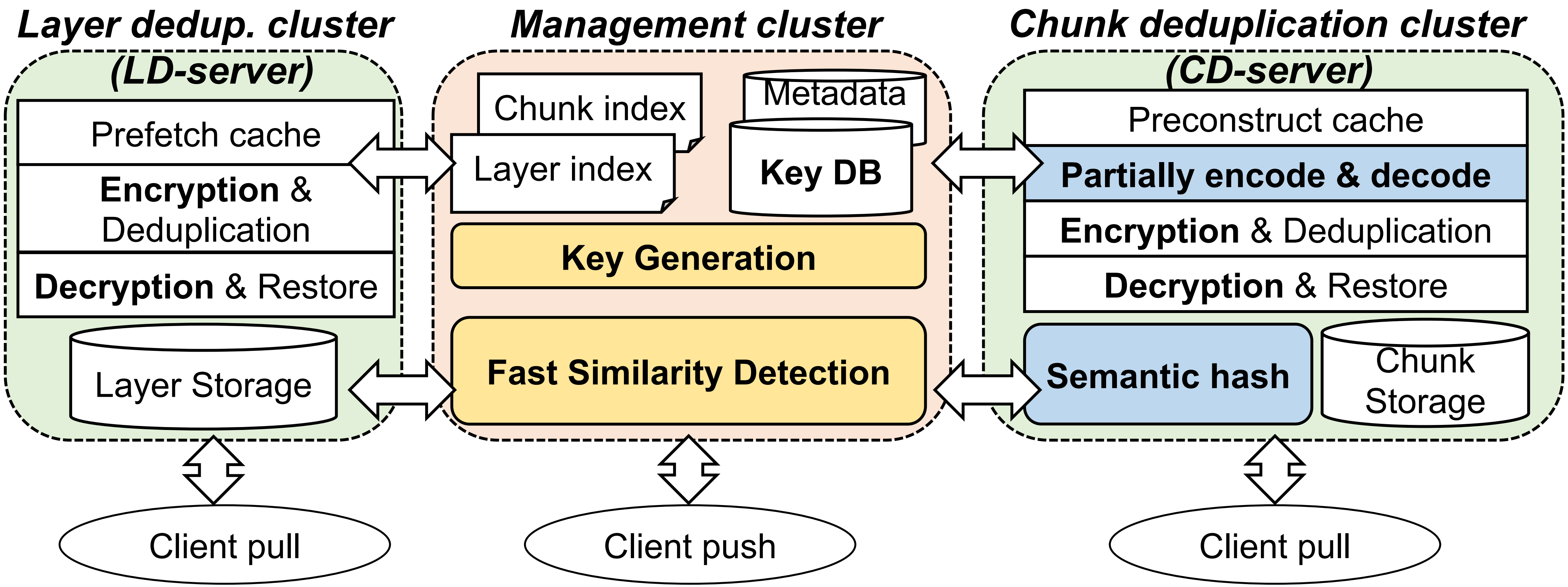
After decompression, the layer exhibits data bloat, resulting in significantly larger duplicated bytes than in the partially decoded space.

Contributions

- We explore a new similarity space in Docker images by only using Huffman decoding
- We propose a fast similarity space selection mechanism that leverages the Huffman tree located at the header of each layer for similarity assessment.
- We propose a semantic-aware MLE technique, which is the first work to introduce semantic hashing in encrypted deduplication for improving the deduplication ratio.

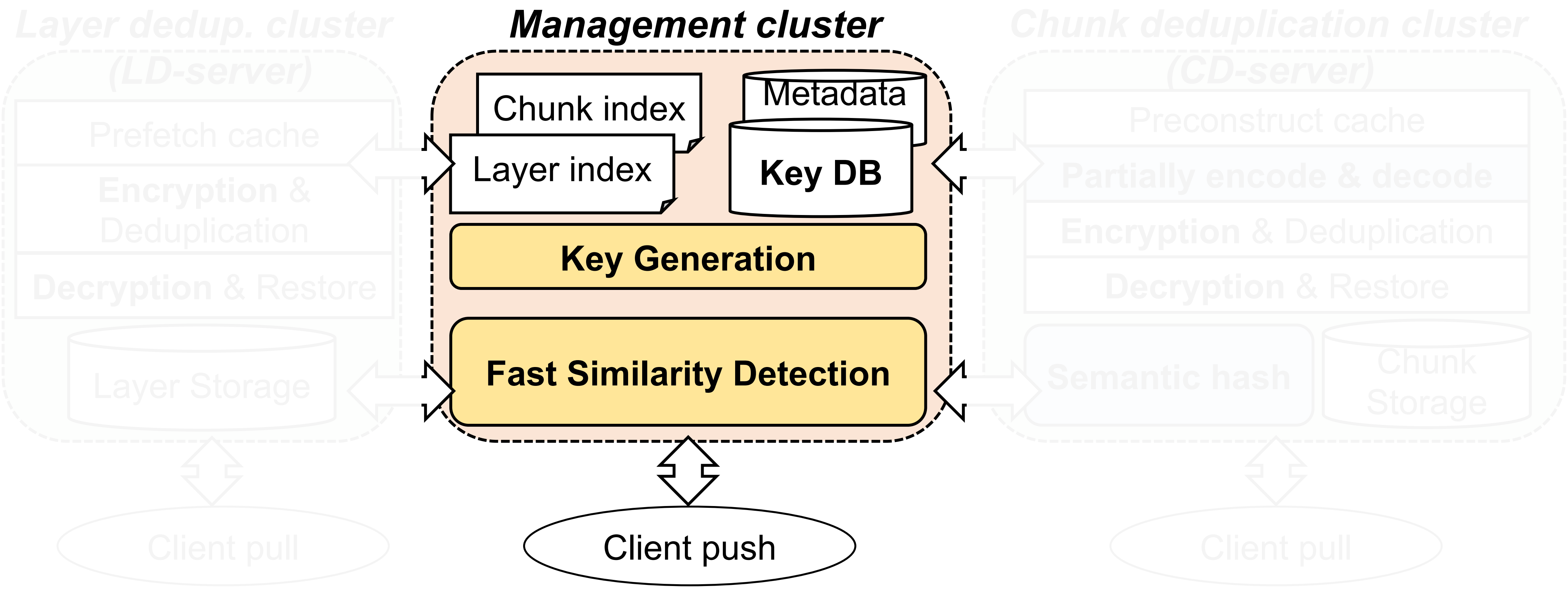
SimEnc Design

- **Overview**



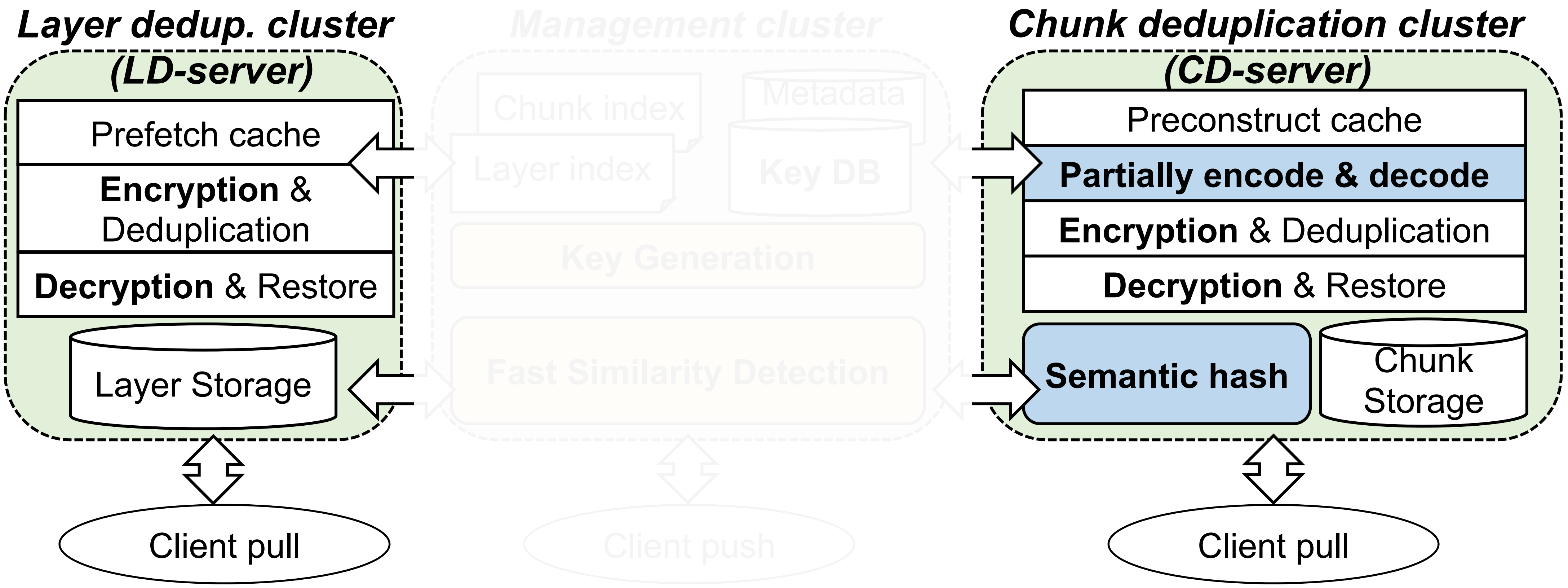
SimEnc Design

- **Overview**



SimEnc Design

- **Overview**



SimEnc Design

Three modes

- balance the trade-off between deduplication ratio and pull latency

(1) Basic deduplication mode n (B-mode n)

- first n layers (layer-level deduplication),
- the remaining layers (chunk-level deduplication)

(2) High deduplication mode (H-mode)

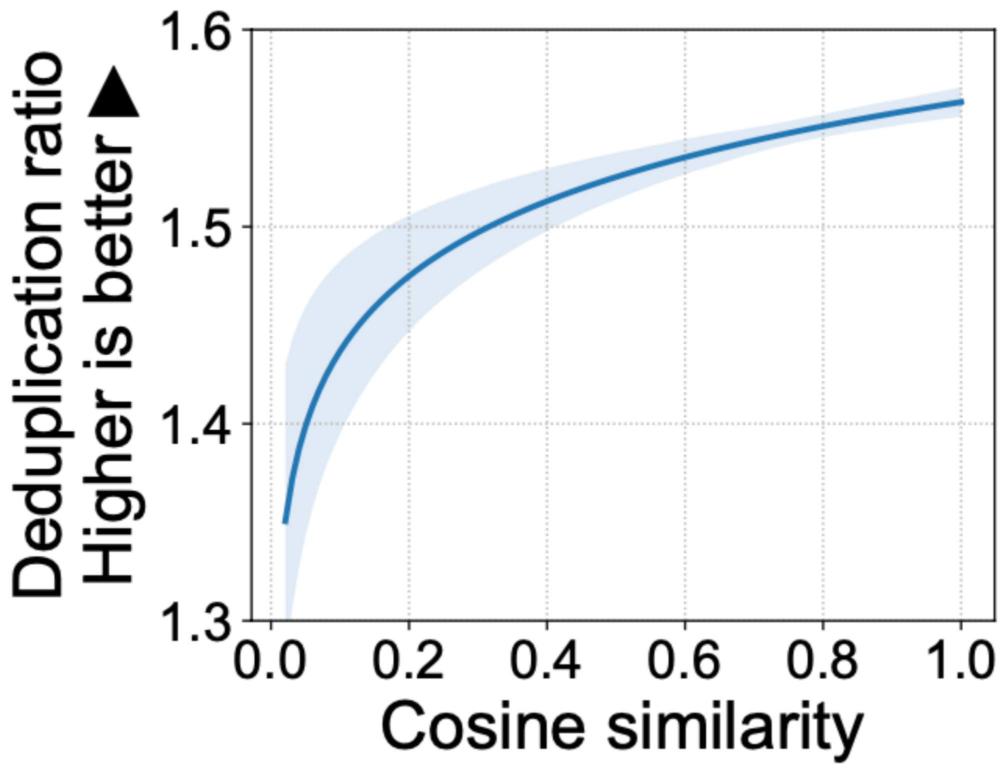
- all layers (chunk-level deduplication)

(3) Flexible deduplication mode (F-mode)

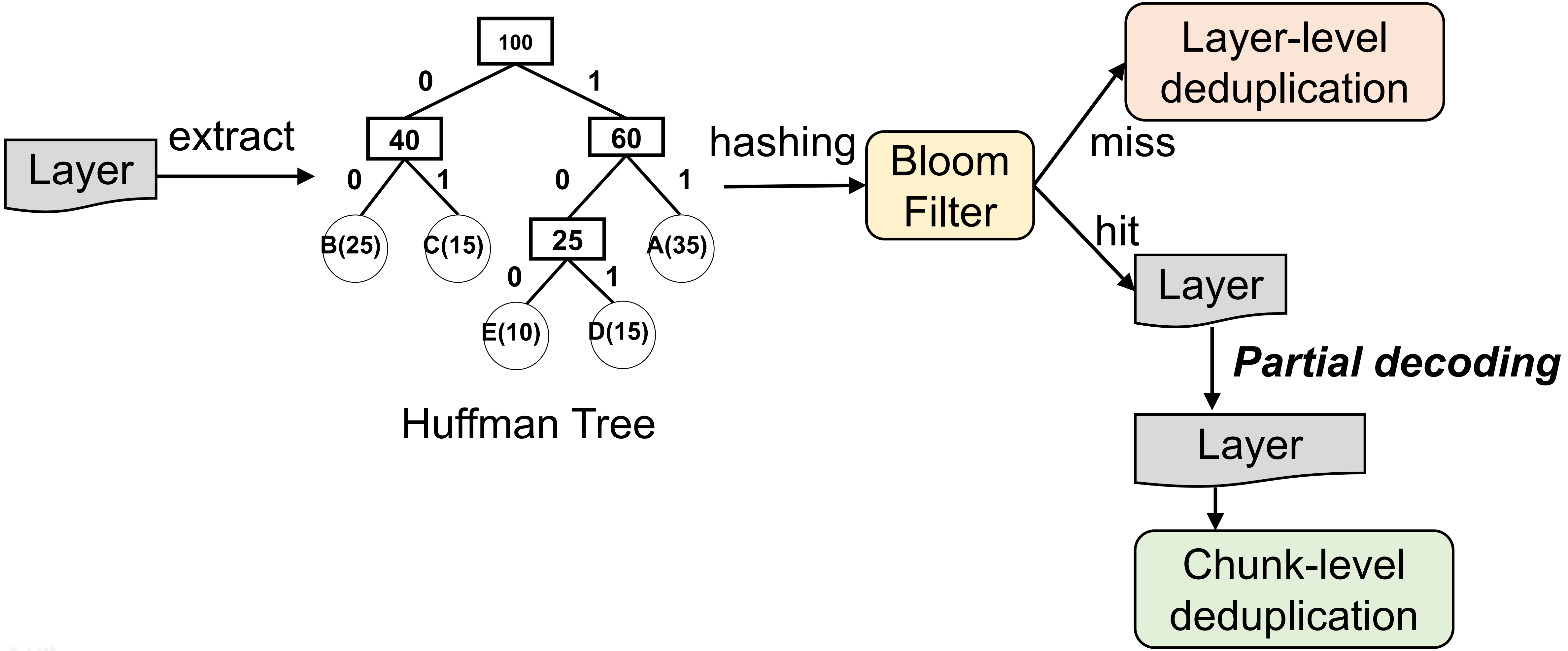
- utilizes image similarity to select the similarity space for deduplication

Fast Similarity Space Selection

- **F-mode**
  - rapidly determines the deduplication space
- **Insight:** exploit the Huffman tree similarity of each layer head



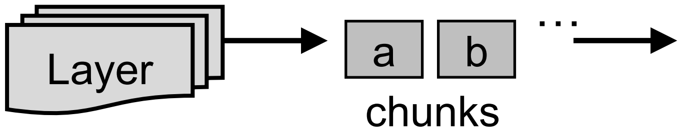
Fast Similarity Space Selection



Semantic-aware MLE

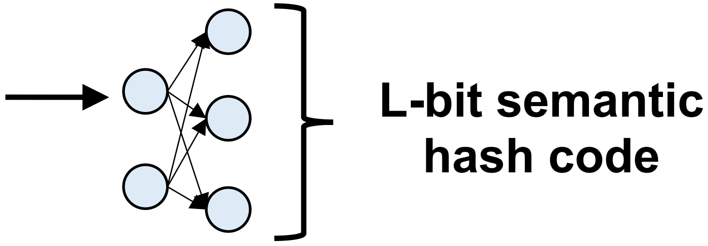
- **We introduce the semantic hash technique**
  - Generate similar hash values for similar chunks

- **Training stage**



Chunk/Label	a	b	c	…
a	1	-1	1	…
b	-1	1	-1	…
c	1	-1	1	…
…	…	…	…	…

Pseudo-label

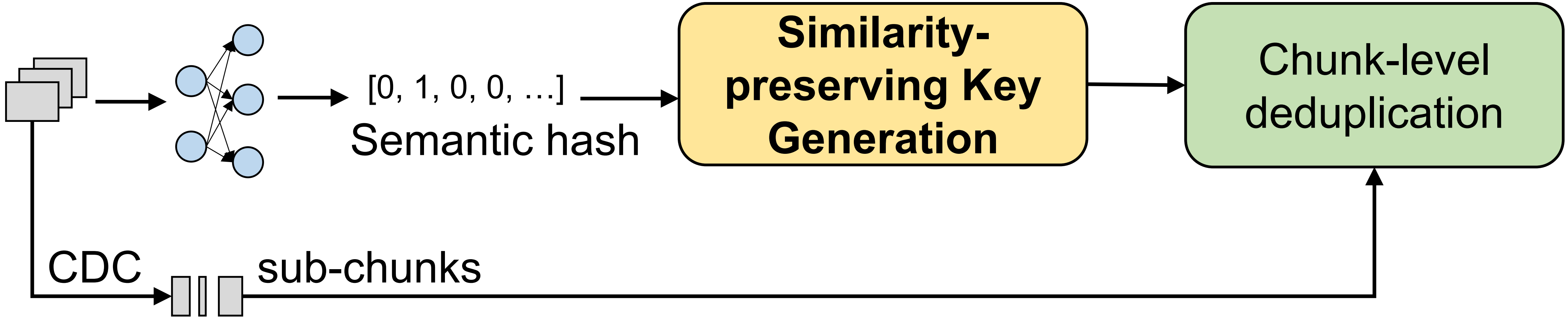


Hash network training

Similarity-preserving Key Generation

- **Semantic hashing is not directly applicable in MLE**

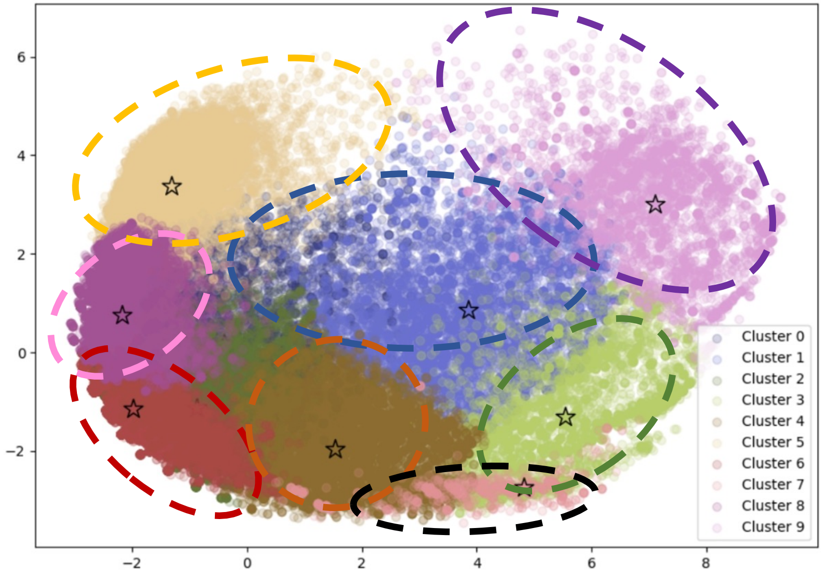
- **Inference Stage**
  - Similarity-preserving key generation



Similarity-preserving Key Generation

- **It is difficult to generate identical keys for similar chunks**

- **Our key idea is clustering semantic hashes to assign keys to the same class**
  - e.g., using the representative chunk key of the class



Similarity-preserving Key Generation

- **It is difficult to generate identical keys for similar chunks**
- **Our key idea is clustering semantic hashes to assign keys to the same class**
  - e.g., using the representative block key of the class
- **We choose the DBSCAN clustering algorithm**

> - ***is it possible to design an adaptive clustering to set hyperparameters automatically?***

Similarity-preserving Key Generation

- **Automatically select the distance ε**

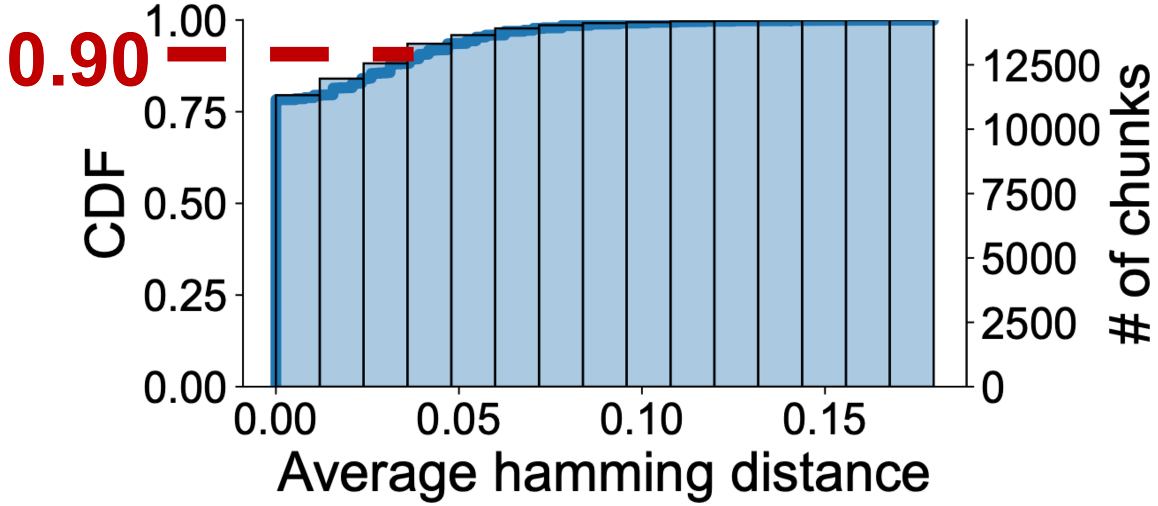
$$N_{\varepsilon}(p) = \{q \in D \mid distance(p,q) \leq \varepsilon\}$$

Similarity-preserving Key Generation

- **Automatically select the distance ε**

$$N_\varepsilon(p) = \{q \in D \mid distance(p,q) \le \varepsilon\}$$

- **Our insight**
  - utilize the LSH method to guide the measurement of semantic hash code distribution



Implementation

- **We have implemented a prototype of SimEnc in Go and C/C++. Our code is open-sourced.**

  https://github.com/suntong30/SimEnc

- **CNN architecture for training the semantic hash**
  - e.g., eight conv layers for 512KiB chunks

Evaluation

- **Platform**
  - Three PCs, each equipped with a 20-core Intel i9-10900K CPU (@3.70 GHz), 128GB DDR4 DRAM, and a 4TB S690MQ SSD
  - 800Mbps Network
  - One GeForce RTX 3090 Ti for training and inference

Evaluation

- **Baselines**
  - **[ATC'20] DupHunter**, the state-of-the-art Docker registry for **plaintext** deduplication.
  - **[ATC'23] AWS Lambda**, the state-of-the-art serverless platform for **encrypted** Docker image deduplication using MLE.
  - **Improved AWS Lambda**, we integrate LSH-based MLE in AWS Lambda with Finesse[1], to generate identical keys for similar chunks.

[1] Finesse: Fine-Grained Fea- ture Locality based Fast Resemblance Detection for Post-Deduplication Delta Compression. In *Proc. of USENIX FAST*, 2019.

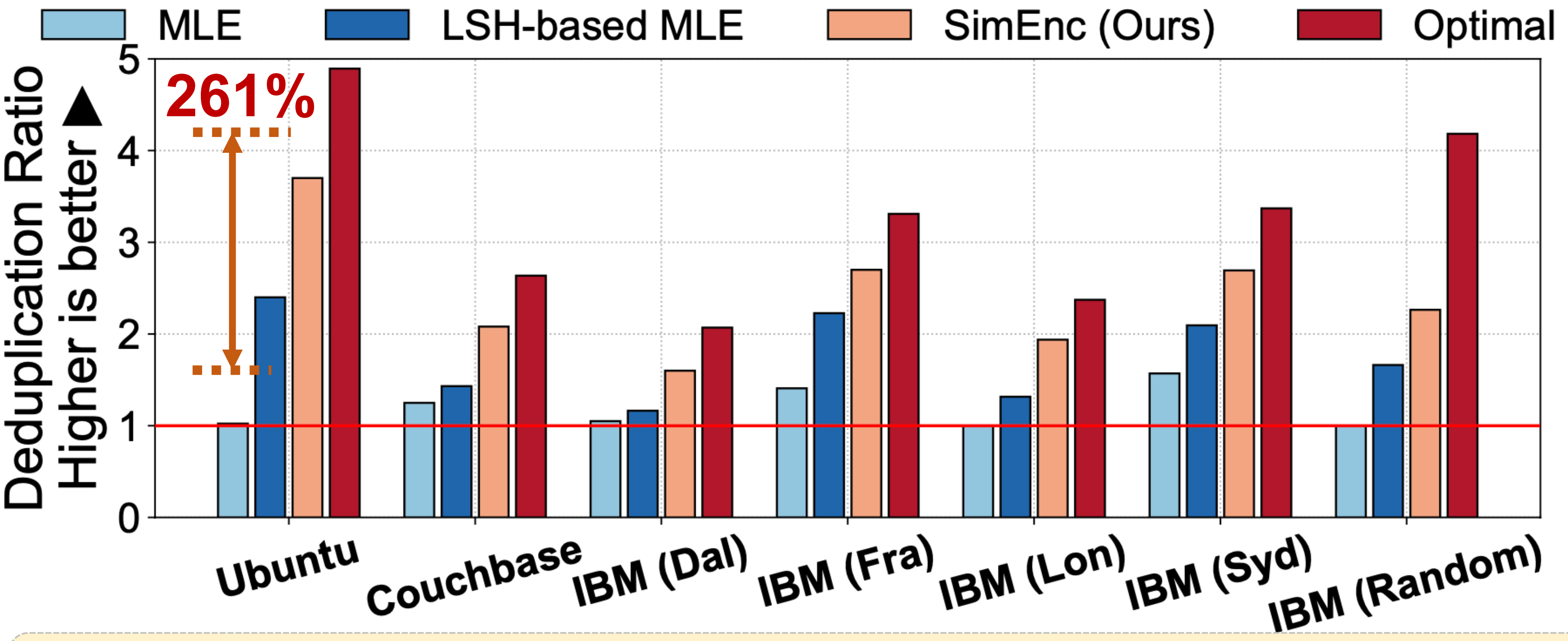
Evaluation

- **Datasets**

Dataset/Workload	#Layer	#Unique Layer	Comp. size	Partially decoded size	Decomp. size
Ubuntu [24]	46	46	1.18 GiB	1.67 GiB	3.24 GiB
Couchbase [20]	516	263	17.74 GiB	35.85 GiB	41.29 GiB
IBM (Dal) [35]	2000	758	11.23 GiB	15.36 GiB	28.97 GiB
IBM (Fra) [35]	2000	700	10.77 GiB	14.57 GiB	27.88 GiB
IBM (Lon) [35]	2000	710	9.49 GiB	13.11GiB	25.11 GiB
IBM (Syd) [35]	2000	503	19.01 GiB	25.73 GiB	48.48 GiB
IBM (Random) [35]	13619	7521	263.13 GiB	318.8GiB	643.95 GiB

Evaluation

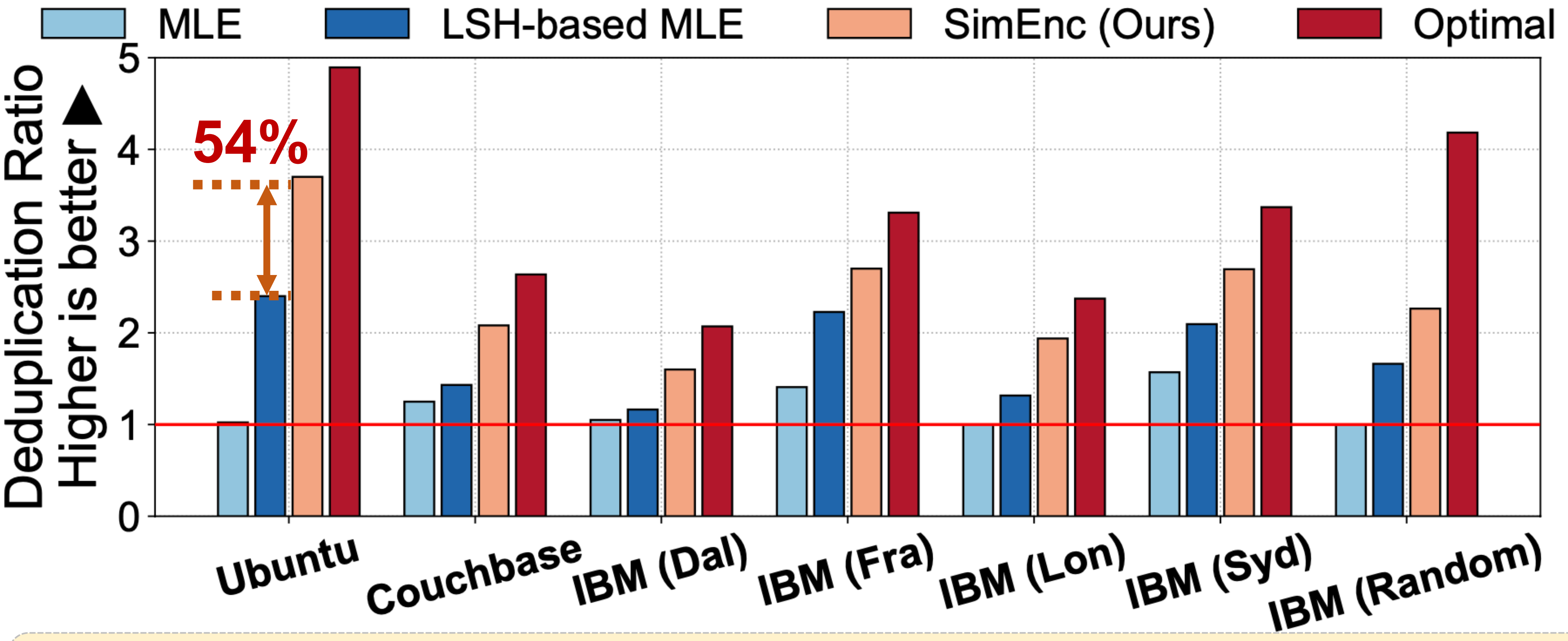
- **Deduplication ratio**



SimEnc achieves an average deduplication ratio that is **38.6%** higher than the **LSH-based MLE** and **109.2%** higher on average compared to the **MLE**

Evaluation

- **Deduplication ratio**



SimEnc achieves an average deduplication ratio that is 38.6% higher than the **LSH-based MLE** and 109.2% higher on average compared to the **MLE**

Evaluation

- **Comparison with DupHunter (IBM Fra)**

High deduplication mode (H-mode)		
Docker Registry	Deduplication ratio	Latency (s)
DupHunter [83]	1.866	0.285
SimEnc (Ours)	2.710 **45.2%↑**	0.206 **27.7%↓**
Flexible mode (F-mode)		
Docker Registry	Deduplication ratio	Latency (s)
DupHunter [83]	1.45	0.124
SimEnc with DupHunter'selective method	1.49	0.117
SimEnc (Ours)	2.70	0.133

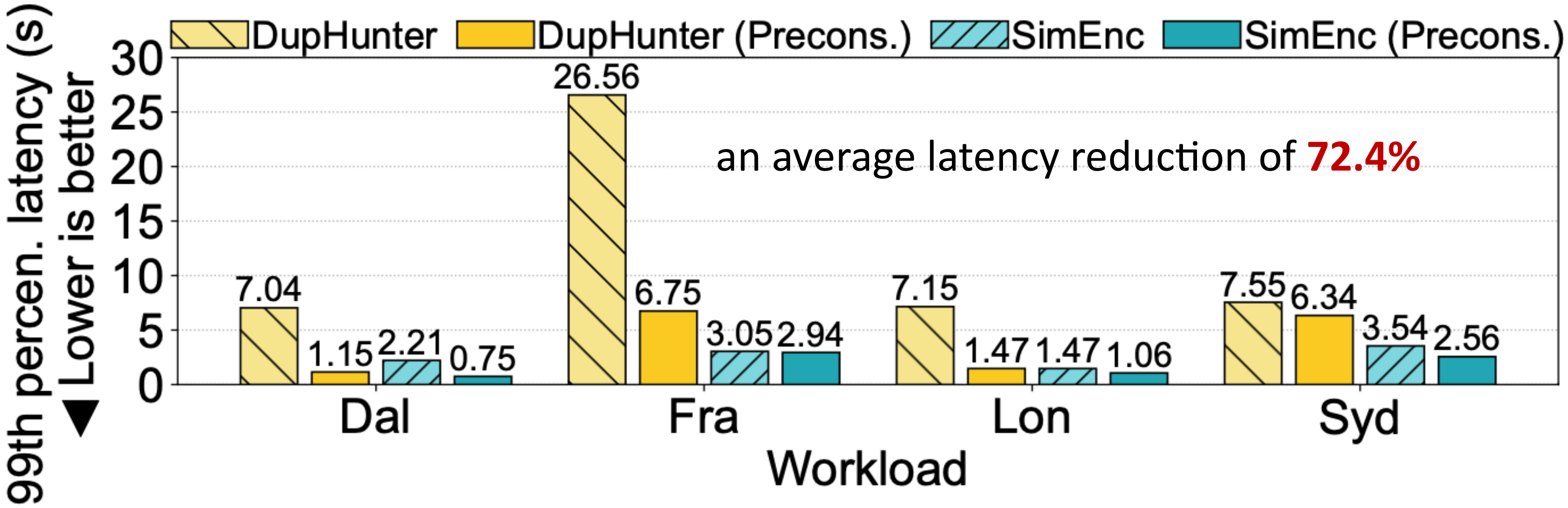
Evaluation

- **Comparison with DupHunter**

High deduplication mode (H-mode)		
Docker Registry	Deduplication ratio	Latency (s)
DupHunter [83]	1.866	0.285
SimEnc (Ours)	2.710	0.206
Flexible mode (F-mode)		
Docker Registry	Deduplication ratio	Latency (s)
DupHunter [83]	1.45	0.124
SimEnc with DupHunter'selective method	1.49	0.117
SimEnc (Ours)	2.70 **86.2%↑**	0.133

Evaluation

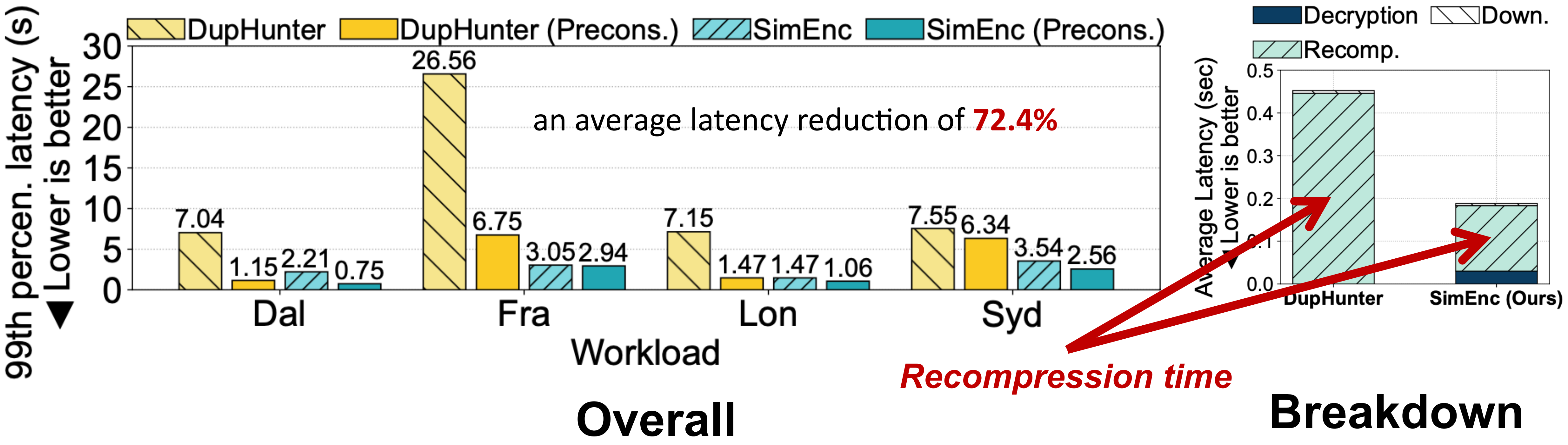
- **Pull latency**



Overall

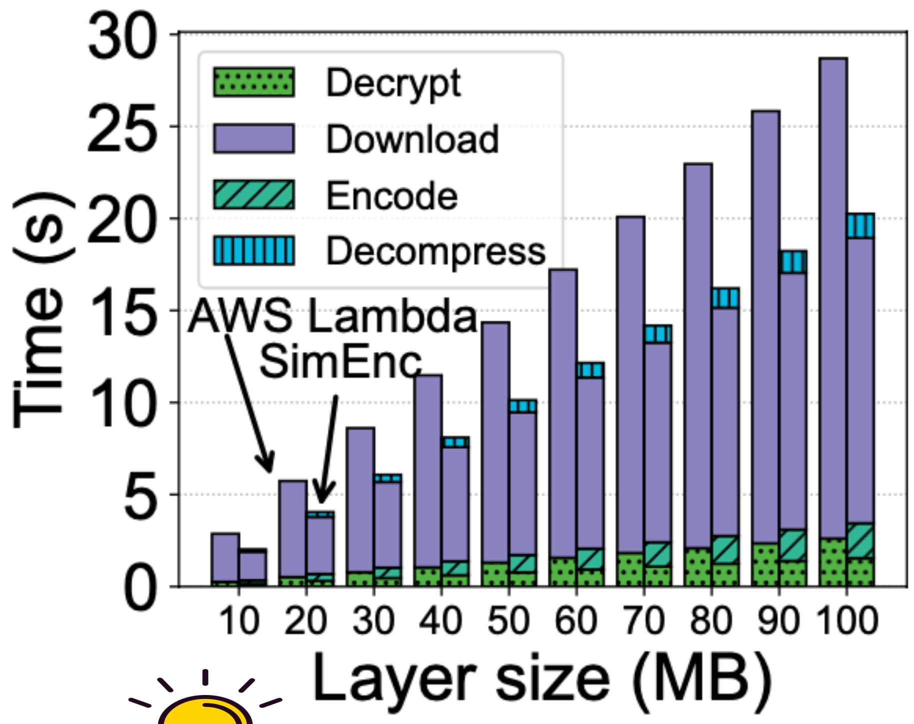
Evaluation

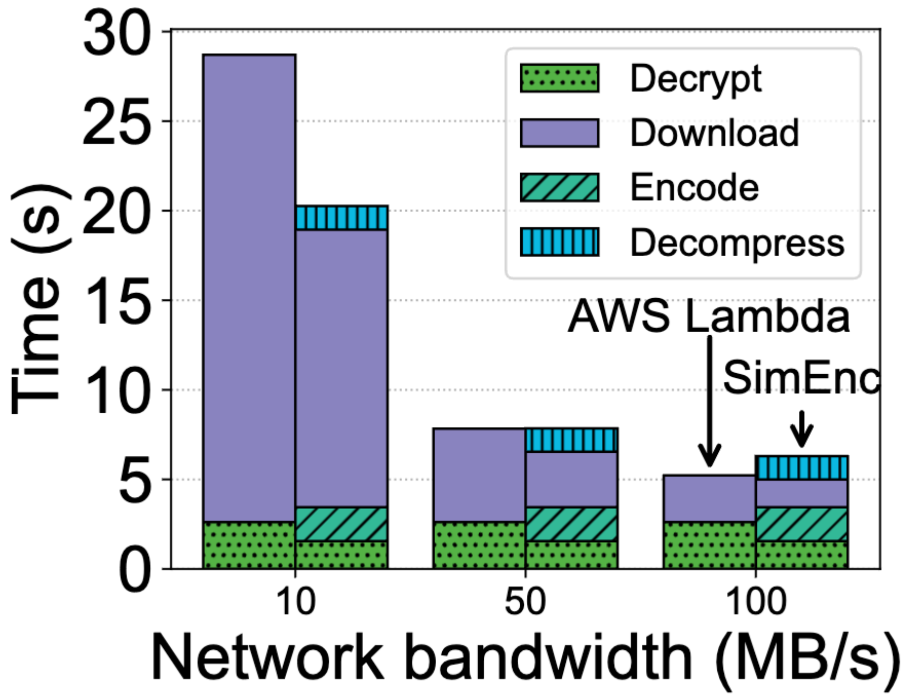
- **Pull latency**



Evaluation

- **Comparison of end-to-end latency with AWS Lambda**



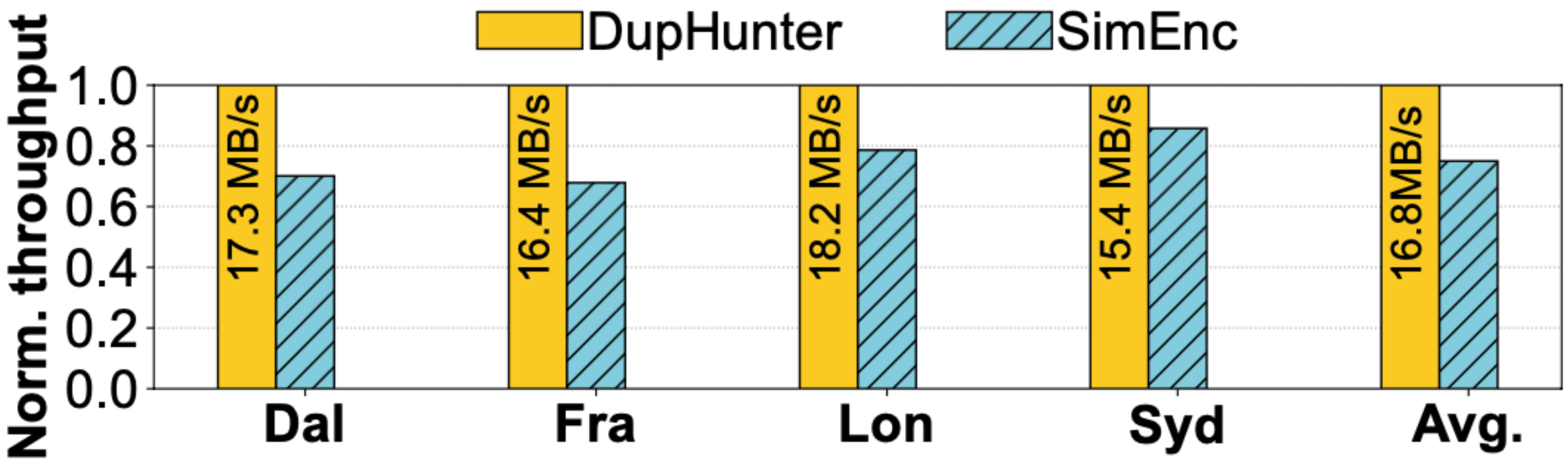


In low-bandwidth scenarios (e.g., <400Mbps), SimEnc achieves lower end-to-end latency because it transmits original compressed data.

Evaluation

Deduplication throughput

- **Partially decoding: 135MB/s**
- **Semantic hash: 16.8MB/s (One GPU) Bottleneck**



Utilizing multiple GPUs for parallel inference could improve throughput

SimEnc

A High-Performance Similarity-Preserving Encryption Approach for Deduplication of Encrypted Docker Images

- **Partial decoding**
- **Fast similarity space selection**
- **semantic-MLE**

Thank you for your attention!

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

Tong Sun, Bowen Jiang, Borui Li, Jiamei Lv, Yi Gao, and Wei Dong

If you have any questions, please contact **tongsun@zju.edu.cn**

浙江大學 区块链与数据安全全国重点实验室
STATE KEY LABORATORY OF BLOCKCHAIN AND DATA SECURITY
ZHEJIANG UNIVERSITY

東南大學
SOUTHEAST UNIVERSITY