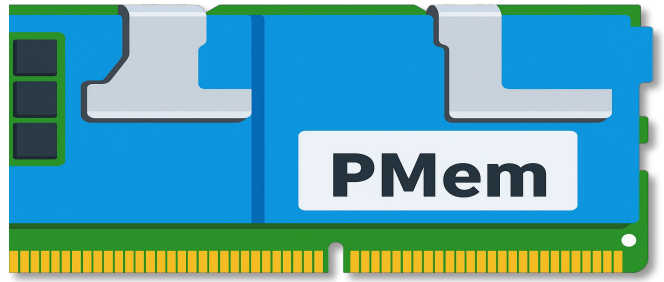
Workshop on Reliable Large-scale Data Management
29th September 2025

Speed Kills: Revisiting Data Deduplication for Modern Storage Devices

Rui Pedro Oliveira, Tânia Esteves, João Paulo
INESC TEC & University of Minho

Storage Landscape



The storage hardware landscape is continuously evolving..

Storage Landscape

- Spinning magnetic disks
- Poor random I/O performance
- Susceptible to fragmentation
- No read/write asymmetry

Storage Landscape

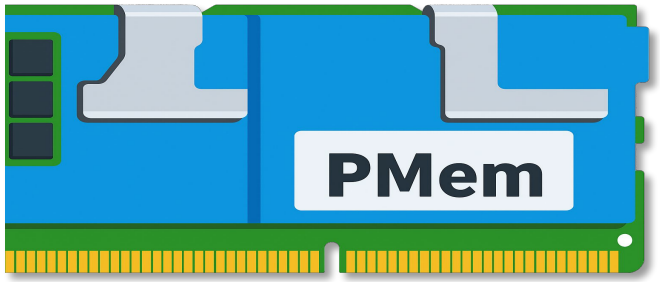
- No moving parts
- Better peak sequential write performance than random
- Small read/write asymmetry (read bandwidth is up to 1.06x the write bandwidth)

Storage Landscape

- Faster transport method (PCIe vs AHCI)
- More parallelism and lower latency than SATA SSDs
- Some read/write asymmetry (read bandwidth is up to 2.1x the write bandwidth)

Storage Landscape

- Byte addressable
- Usable as main memory or storage device
- Similar sequential and write random performance
- Better sequential read performance
- Large read/write asymmetry (read bandwidth is up to 3x the write bandwidth)



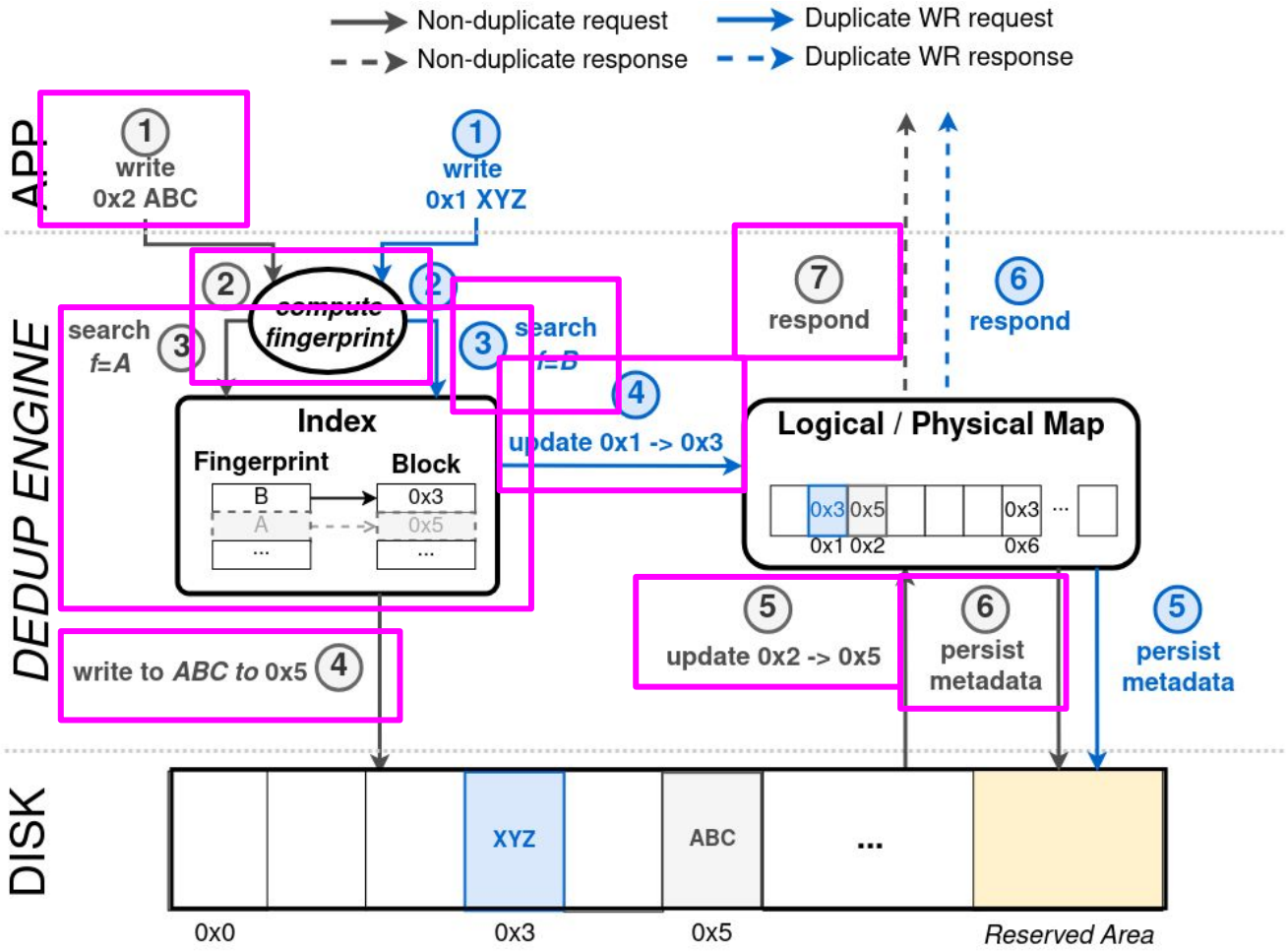
Challenges and Contributions

- Weak link between deduplication solutions and their target hardware
- No systematic study of performance impact of different deduplication mechanisms

- This paper, on top of a literature consolidation, studies:
  - How different characteristics of storage devices (e.g. HDD, SSD, PMem) impact deduplication design, namely fingerprinting
  - How compact index metadata structures can effectively deal with large volumes of data

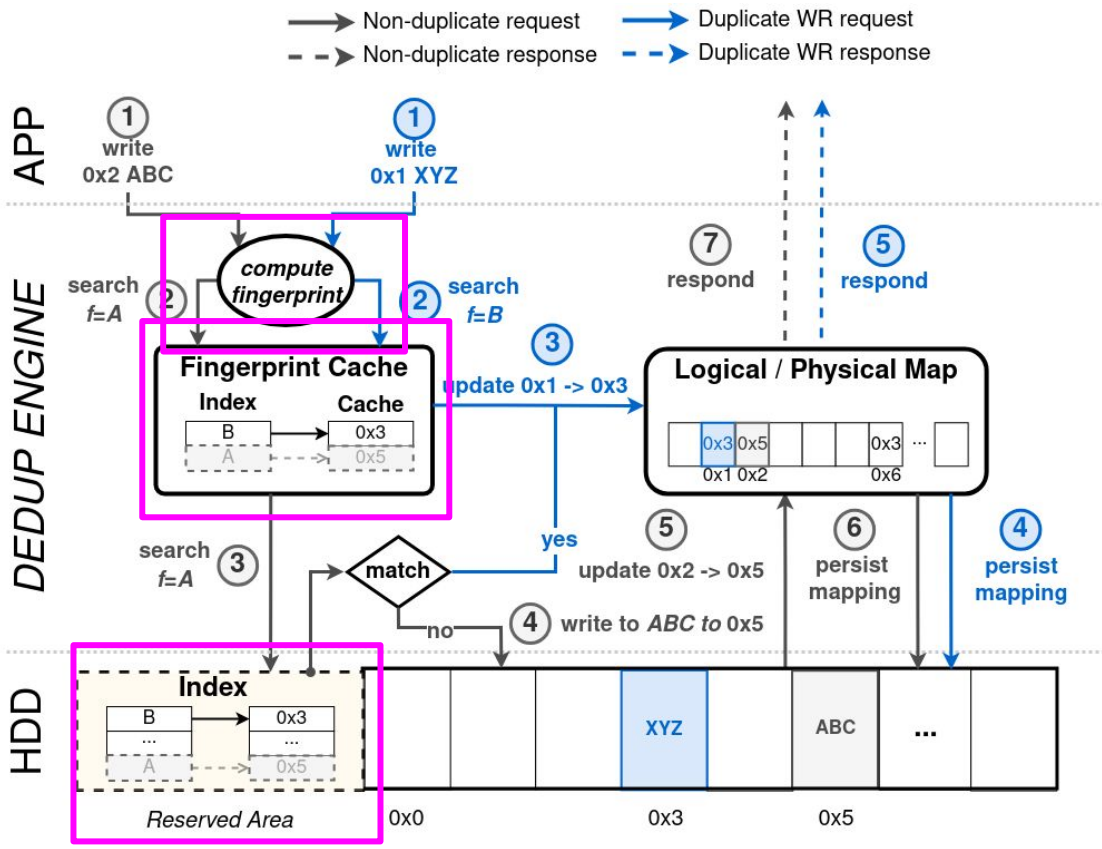
Data Deduplication

1. Chunking
2. Fingerprinting
3. Indexing
4. Writing to disk (if required)
5. Update logical/physical mapping
6. Persist metadata



Data Deduplication: HDD

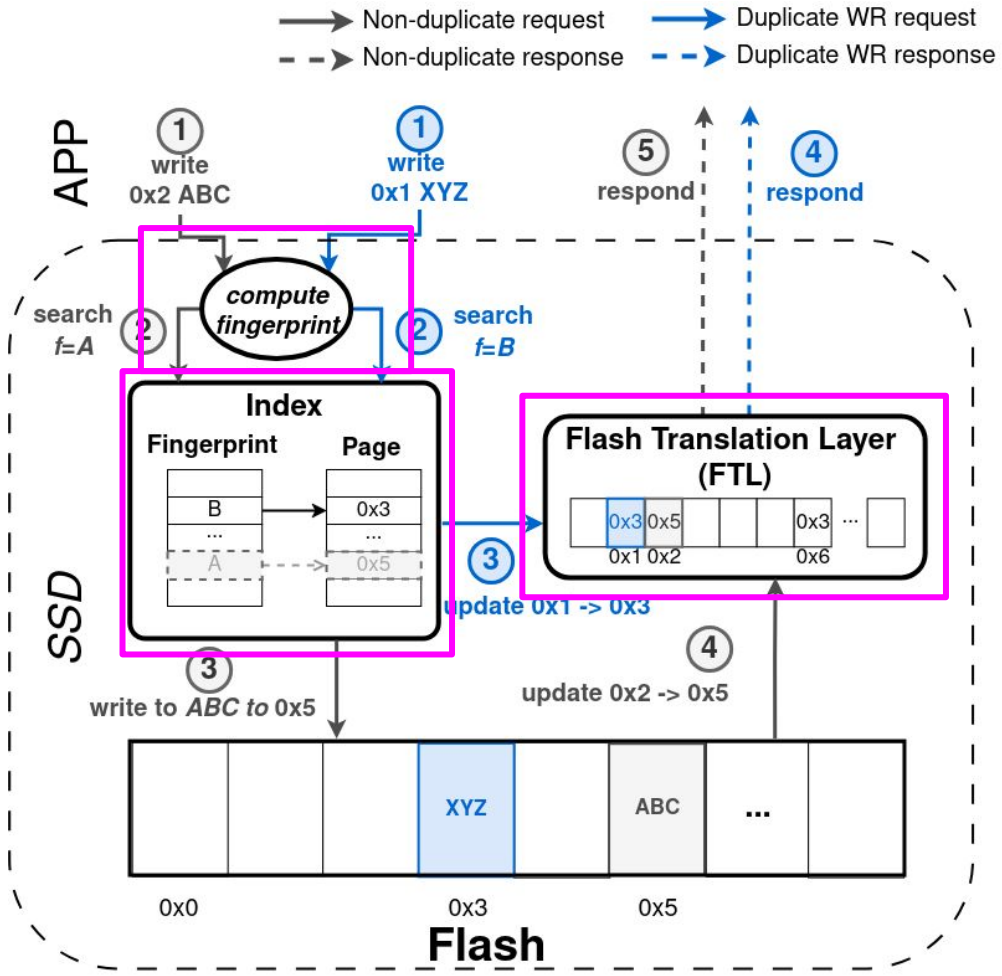
- Cryptographic hash functions [1,2,3]
- Fingerprint caches [1], Bloom filters [2]
- Efforts to preserve on-disk layout [2,3]



1 - Quinlan, Sean, and Sean Dorward. "Venti: A new approach to archival data storage.", 2002.
2 - Zhu, Benjamin, Kai Li, and R. Hugo Patterson. "Avoiding the disk bottleneck in the data domain deduplication file system.", 2008.
3 -Srinivasan, Kiran, et al. "iDedup: Latency-aware, inline data deduplication for primary storage.", 2012.

Data Deduplication: SSD

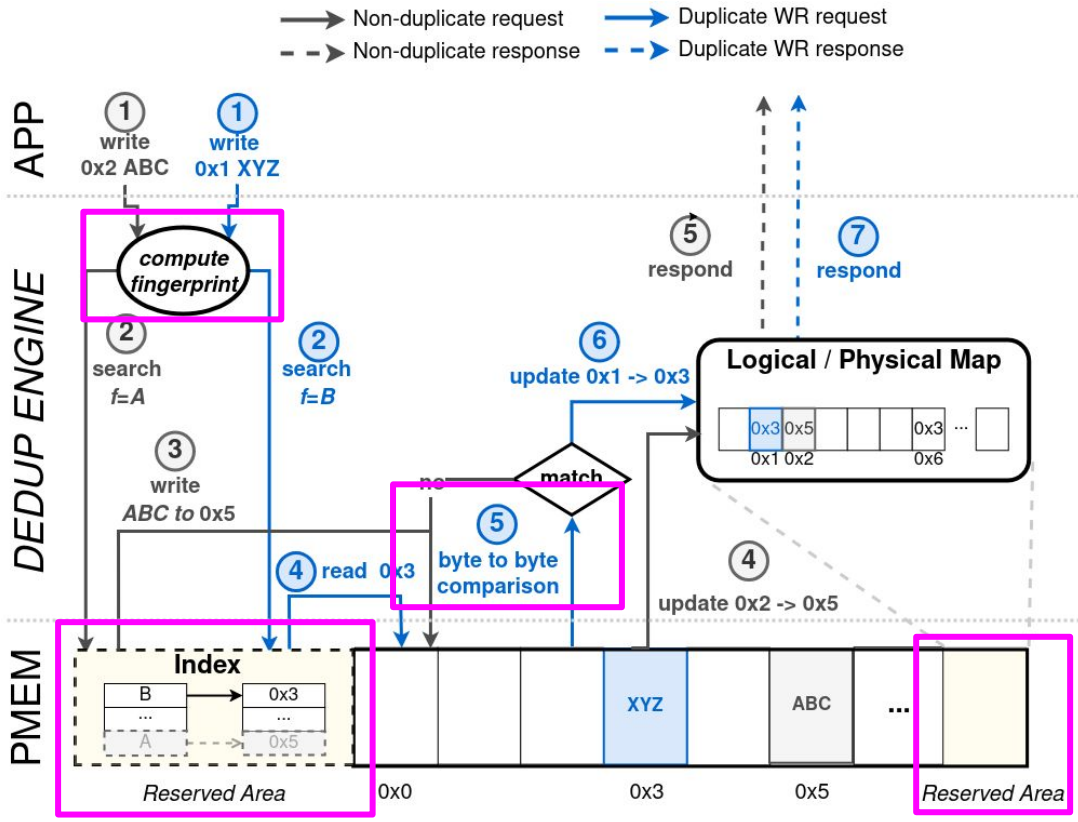
- Exploits FTL, GC and wear leveling mechanisms [1,2,3]
- Some systems offer a best-effort approach [1,2,3]
- Metadata can be stored on device[1], DRAM [3,4], or PMem [2]



1 - Chen, Feng, Tian Luo, and Xiaodong Zhang. "CAFTL: A Content-Aware flash translation layer enhancing the lifespan of flash memory based solid state drives.", 2011.
2 - Zhou, You, et al. "Remap-SSD: Safely and efficiently exploiting SSD address remapping to eliminate duplicate writes.", 2021.
3 - Wen, Yuhong, et al. "Eliminating storage management overhead of deduplication over ssd arrays through a hardware/software co-design.", 2024.
4 - Miranda, Mariana, et al. "S2Dedup: SGX-enabled secure deduplication.", 2021.

Data Deduplication: PMem

- Byte addressable
- Non-cryptographic hash functions [1,2,3]
- Byte-to-byte comparison handles collisions [1,2,3]



1 - Qiu, Jiansheng, et al. "Light-Dedup: A Light-weight Inline Deduplication Framework for Non-Volatile Memory File Systems." 2023.
2 - Zuo, Pengfei, et al. "Improving the performance and endurance of encrypted non-volatile main memory through deduplicating writes.", 2018.
3 - Du, Chunfeng, et al. "FSDedup: Feature-Aware and Selective Deduplication for Improving Performance of Encrypted Non-Volatile Main Memory.", 2024

Study on Deduplication

Fingerprinting

Fingerprinting

There are many different hash functions:

- SHA-2 family
- Blake3
- xxHash
- Cyclic Redundancy Check (CRC)
- …

How to choose the appropriate hash function?

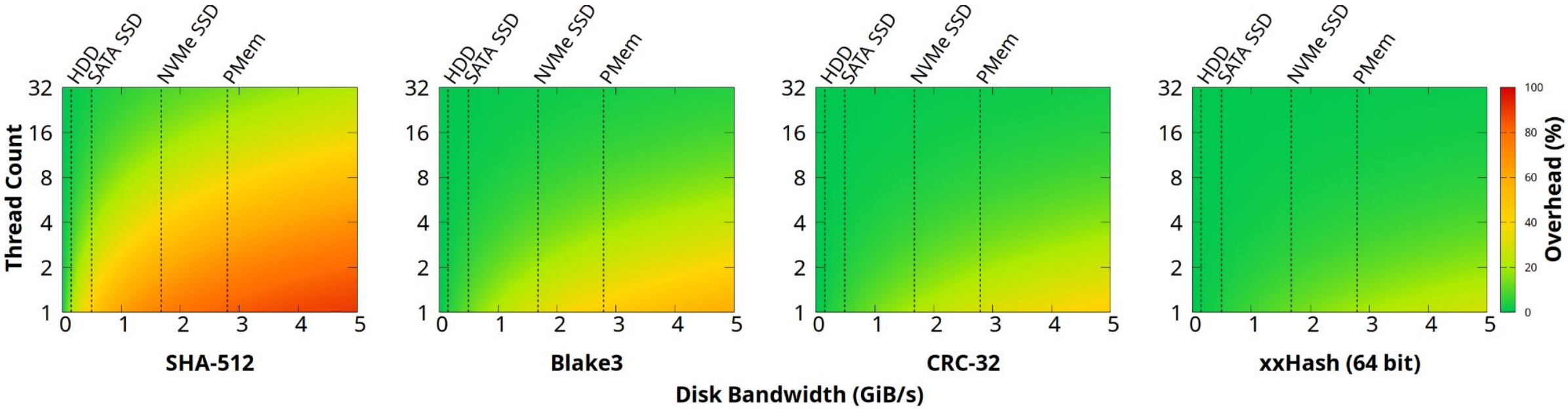
Fingerprinting

	SHA-512	Blake3	CRC-32	xxHash
Cryptographic	Yes	Yes	No	No
Latency (ns)	6535 ± 1579	1095 ± 700	603 ± 510	325 ± 429
Throughput (MiB/s)	598 ± 7	3569 ± 58	6473 ± 60	12001 ± 145
Digest Size (B)	64	32	4	8
Hashes needed for 50% chance of collisions	1.4×10^{77}	4.0×10^{38}	7.7×10^{4}	5.1×10^{9}

How does hashing performance impact deduplication overhead?

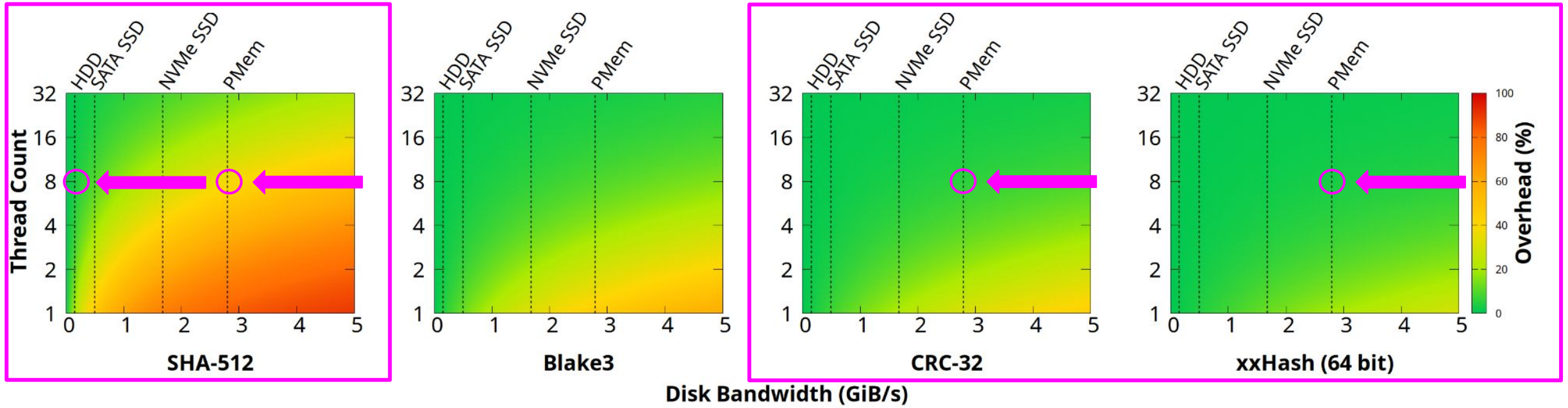
Fingerprinting

- We plot how hash performance impacts deduplication overhead based on:
  - Disk Bandwidth
  - Thread Count



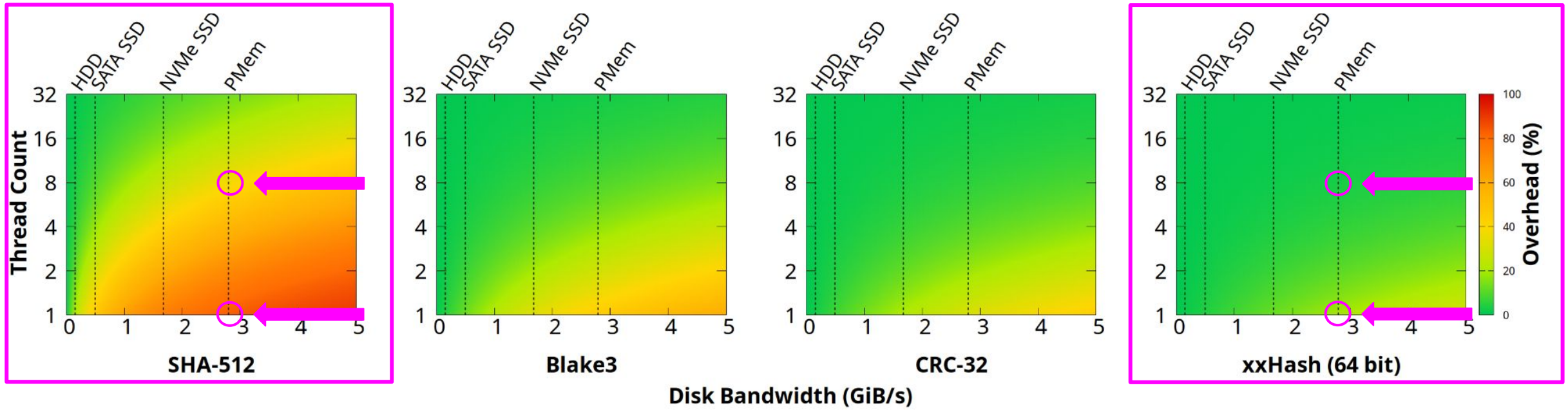
Fingerprinting

- Hash function matters more for fast devices
  - e.g. w/ 8 threads, SHA-512 on HDD introduces 3% overhead vs. 37.4% on PMem
  - xxHash and CRC-32 can still introduce low overhead on fast PMem devices (2.89% and 5.23% w/ 8 threads)



Fingerprinting

- More threads reduce overhead, and slower hashes benefit more
  - SHA-512 on PMem drops 45 percentage points going from 1 to 8 threads
  - xxHash on PMem drops only 16 percentage points going from 1 to 8 threads
- However, they add contention to managing metadata structures



Fingerprinting

But this analysis is not enough!

Fingerprinting

The fingerprint method to use depends on:

- **I/O Workload**

More Duplicates → More False Collisions → More Byte Comparisons

- **Collision Resistance of Hash Function**

More Collision Resistant → Fewer True Collisions → Fewer Byte Comparisons

- **Read/Write Asymmetry of Device**

More Read Bandwidth → Cheaper Byte Comparisons

Study on Deduplication

Indexing

Indexing

- Deduplication indexes can grow very large
- Storing on disk carries heavy performance penalty

Do we need to keep track of the full index?

Indexing

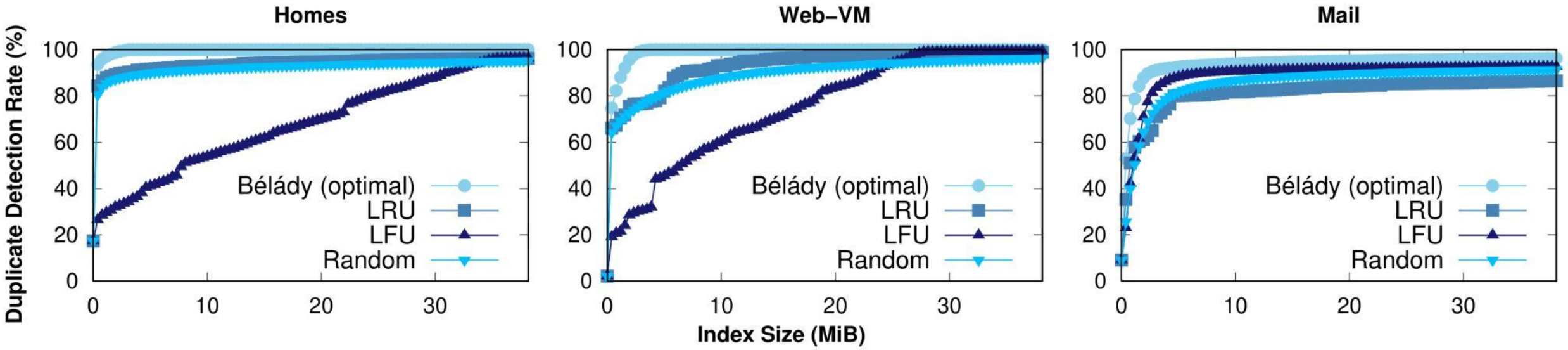
The answer is no!

- Some systems employ partial deduplication
- Partial deduplication systems must choose an eviction policy

But how do you choose the appropriate eviction policy?

Indexing

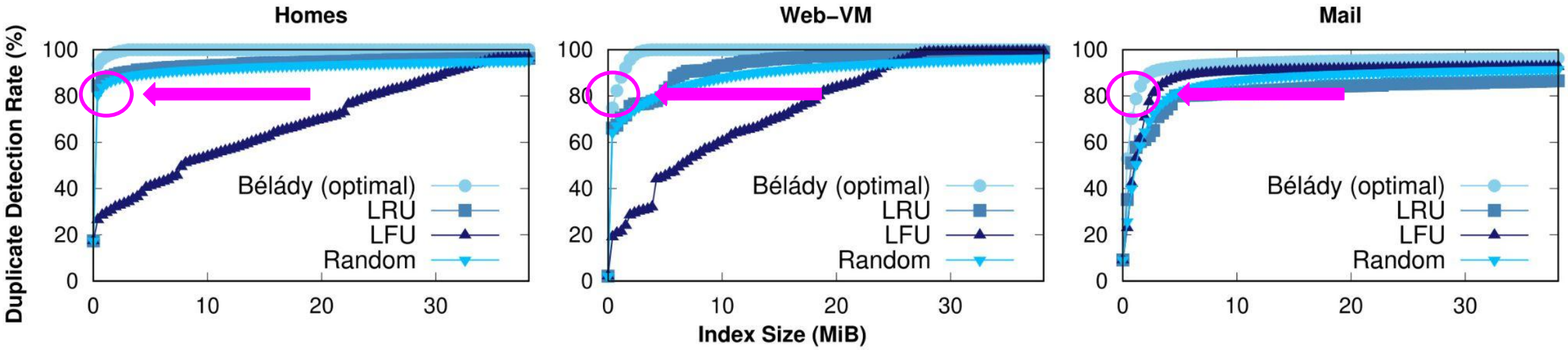
- We compare the efficiency of 4 eviction policies for the FIU IODedup traces:
  - Bélády (the optimal cache replacement algorithm)
  - Least Recently Used (LRU)
  - Least Frequently Used (LFU)
  - Random Eviction (averaged over 10 tests)



*20B per entry

Indexing

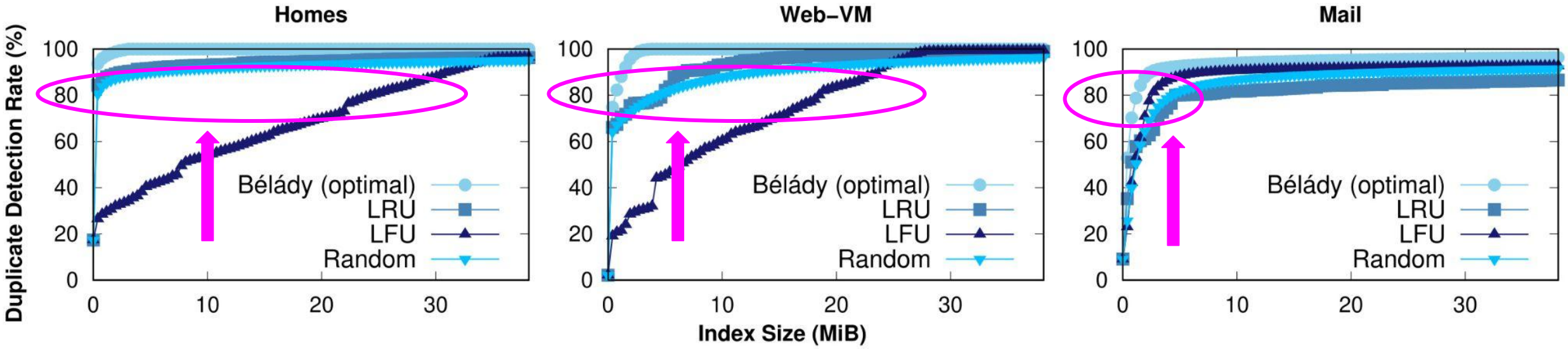
- Small index sizes are, in theory, enough to identify 80% of duplicates
  - 0.03 MiB (Homes)
  - 0.65 MiB (Web-VM)
  - 1.23 MiB (Mail)



*20B per entry

Indexing

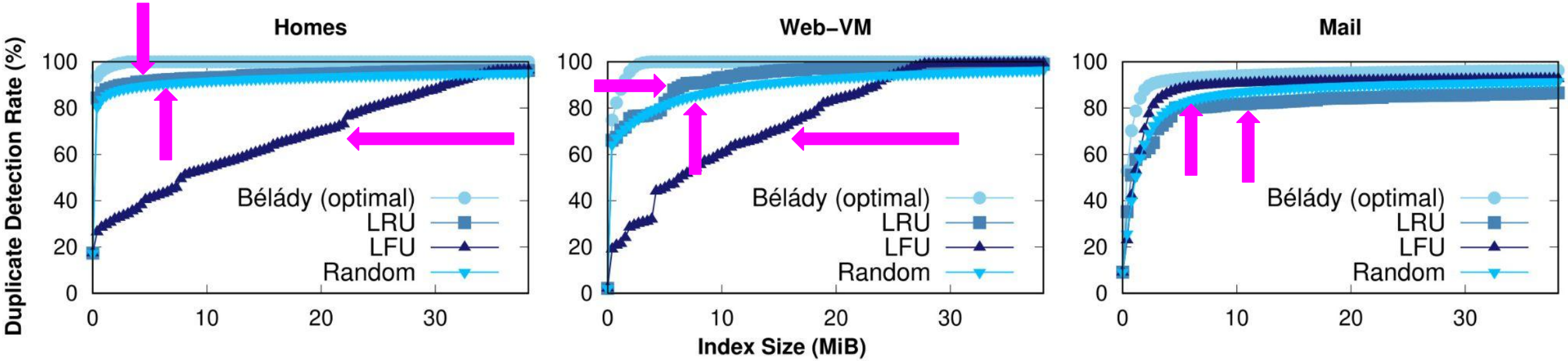
- In practice, indexes need to be slightly larger
  - LRU: 0.20 MiB (Homes), 4.80 MiB (Web-VM), 2.54 MiB (Mail)
  - Random: 0.35 MiB (Homes), 4.38 MiB (Web-VM), 4.29 MiB (Mail)
  - LFU: 24.35 MiB (Homes), 18.50 MiB (Web-VM), 4.57 MiB (Mail)



*20B per entry

Indexing

- Policy matters
  - LRU performs closest to the optimal policy (except in Mail)
  - Random is the second closest eviction policy
  - LFU performs poorly in Homes, Web-VM due to **cache solidification**



*20B per entry

Takeaways

1. Cryptographic hash function are too slow for high-performance mediums, but using non-cryptographic functions require byte-to-byte comparisons
2. The trade-off of byte-to-byte comparisons is not trivial to evaluate, it depends on many factors
3. Systems must choose a fingerprinting method based on the workload and device characteristics
4. Partial indexes can achieve good deduplication performance, but their efficiency depends on the eviction policy and the I/O workload

*20B per entry

Future Directions

- **Computation on the critical I/O path**
  - Computation will become more expensive
  - Inline deduplication exclusively with non-cryptographic hash functions
- **Indexing and Metadata Management**
  - Indexes will still not fit in RAM
  - Penalty for on-disk indexing will be less
  - Reduced RAM fingerprint

Future Directions

- **New Devices and Protocols**
  - NVMe-oF, CXL will change improve remote storage speeds
- **Computational Storage**
  - CSD devices will allow deduplication to run exclusively on device

Workshop on Reliable Large-scale Data Management
29th September 2025

Speed Kills: Revisiting Data Deduplication for Modern Storage Devices

Rui Pedro Oliveira, Tânia Esteves, João Paulo
INESC TEC & University of Minho

rui.v.oliveira@inesctec.pt

dsr-haslab.github.io

INESC TEC

Universidade do Minho