SmartDedup: Optimizing Deduplication for Resource-constrained Devices

Qirui Yang Runyu Jin Ming Zhao

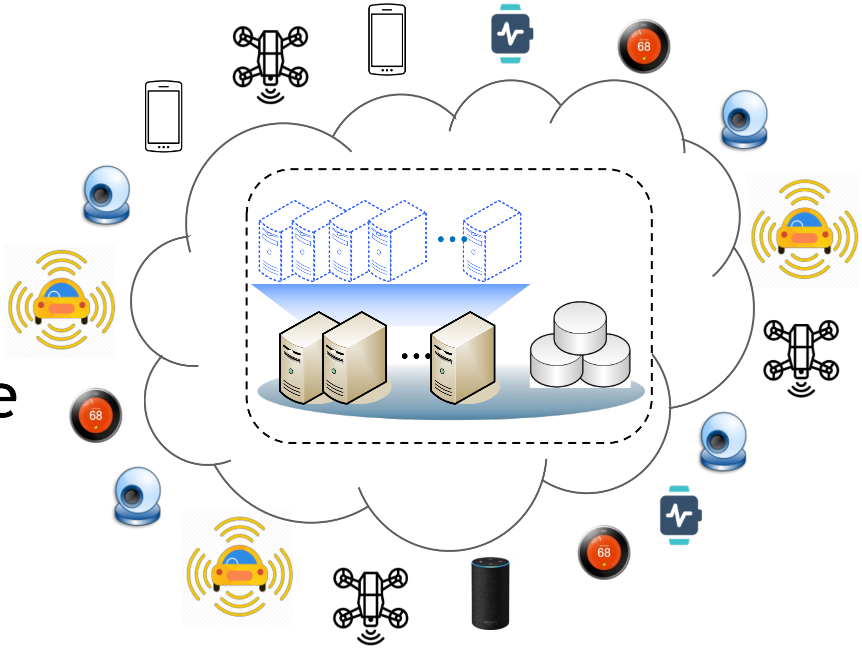
qirui.yang@asu.edu, runyu.jin@asu.edu, mingzhao@asu.edu

Arizona State University

http://visa.lab.asu.edu

Resource Management on Edge and IoT

- Data-intensive workloads
  - Multitude of sensors
  - Data-rich applications
  - Multitasking systems
- Limited on-device storage
  - Limited I/O performance
  - Limited capacity
  - Limited endurance



Deduplication to the Rescue?

- Well-known benefits
  - Eliminate redundant I/Os → improve performance
  - Reduce flash writes → improve endurance
  - Remove redundant data → improve utilization
- Widely used in the cloud
  - Backup storage, primary storage, cache storage, …
- But
  - Is there enough data duplication in device workloads?
  - How to exploit it using limited resources on the device?

Smartphone Trace Analysis

- Real-world device workloads are indeed I/O intensive
- All workloads have a good level of duplicates

File system level traces collected from users from different countries

ID	Daily I/Os (GB)	Daily write (GB)	# of months	Duplication ratio (%)
1	12.1	1.4	6	21.9
2	9.1	2.1	3	45.3
3	5.9	1	2.5	23.2
4	16.5	2.4	5	47.5
5	12.2	2.8	3	41.6
6	10.6	2.4	2.1	28.3

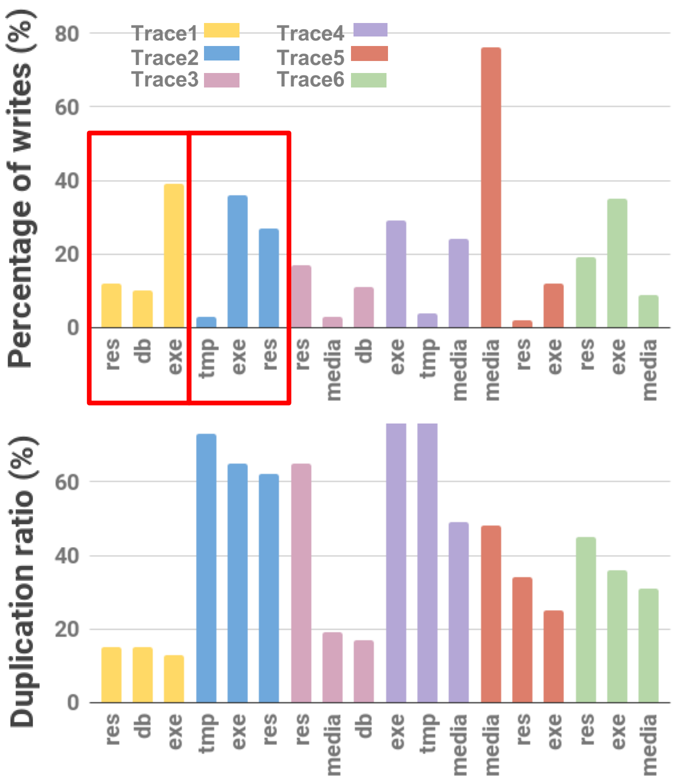
Trace: http://visa.lab.asu.edu/traces

More In-depth Analysis

Five categories:

- Resource files (res)
- Database files (db)
- Executables (exe)
- Temporary files (tmp)
- Multimedia (media)

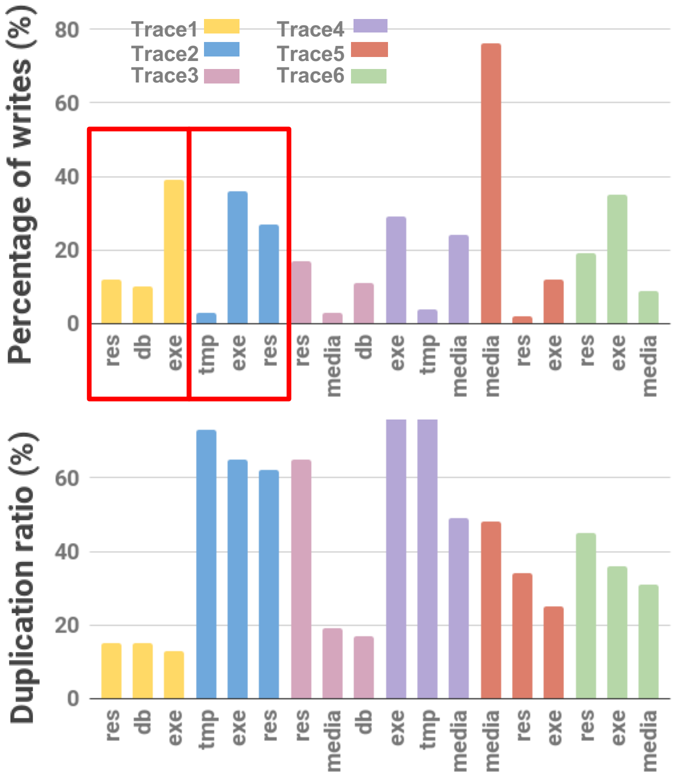
Top 3 duplicate contributors for the traces



More In-depth Analysis

- Duplicate sources differ by devices
- 80% of duplicates are from files that are not entirely duplicate
- System-level deduplication is necessary

Top 3 duplicate contributors for the traces



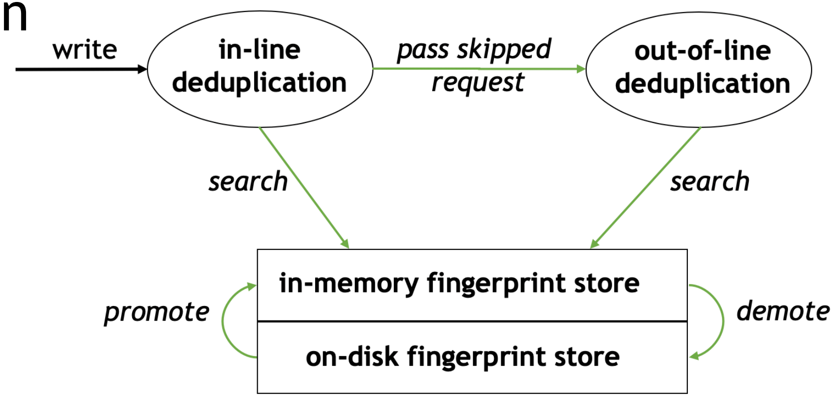
![VISA logo]

Challenges to Device Deduplication

- Limited memory
    - Deduplication relies on the in-memory fingerprint store for quickly detecting duplicates
- Limited storage performance and endurance
    - Deduplication incurs additional metadata operations
- Limited power and energy capacity
    - Fingerprinting is CPU intensive and draws quite a bit of power

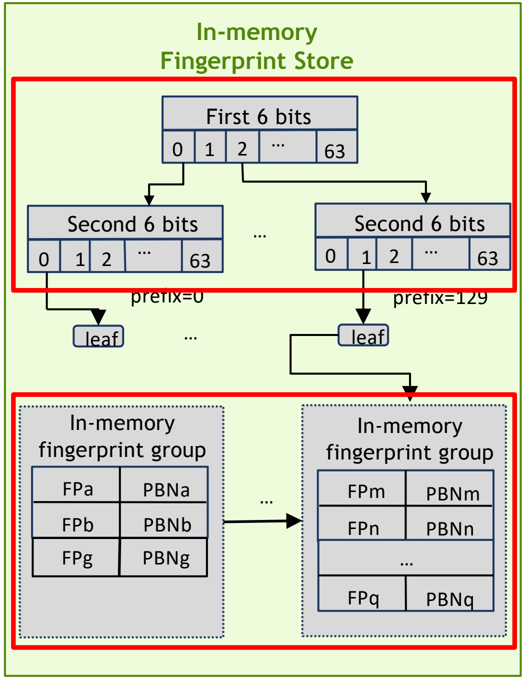
SmartDedup—A Smart Deduplication Solution for Smart Devices

- Cohesively designed two-level fingerprint stores
  - On-disk store complements the small in-memory store
- Synergistically integrated hybrid deduplication
  - Out-of-line deduplicates data skipped by in-line
- Dynamically adapted deduplication
  - According to resource availability and workload characteristics



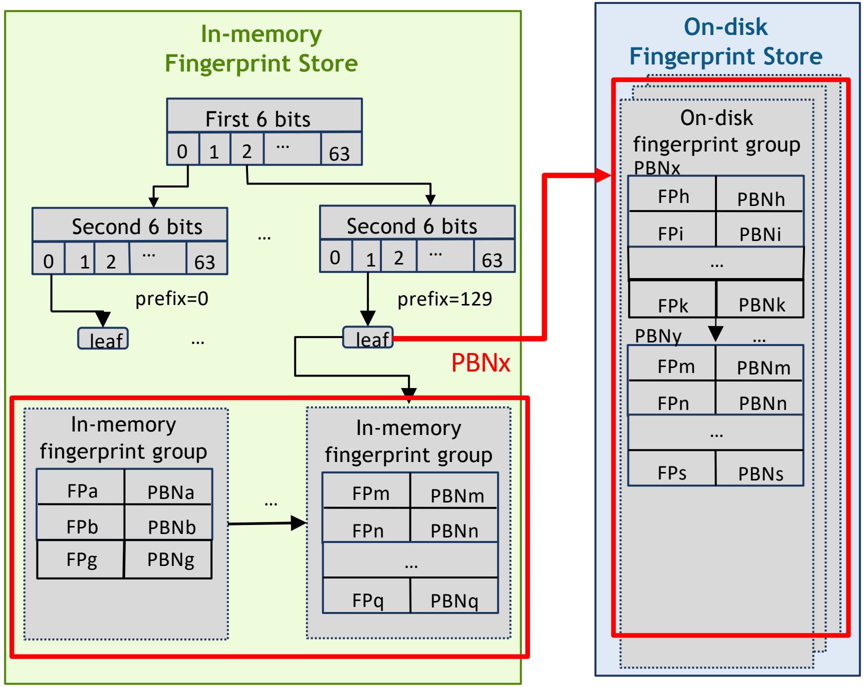
In-memory Fingerprint Store

- Store only important fingerprints
- Organized by a prefix tree
  - Index groups of fingerprints – small memory footprint
  - Support the two-level fingerprint store



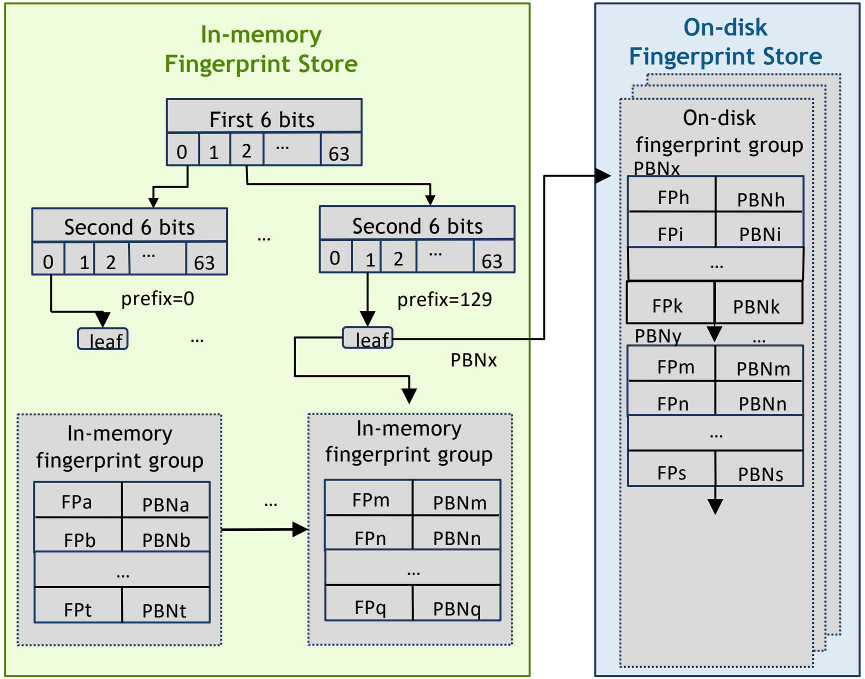
On-disk Fingerprint Store

- Maintain fingerprints that are not in the in-memory store
- Share the same index with in-memory store
  - Save memory
  - Facilitate the fingerprint migration



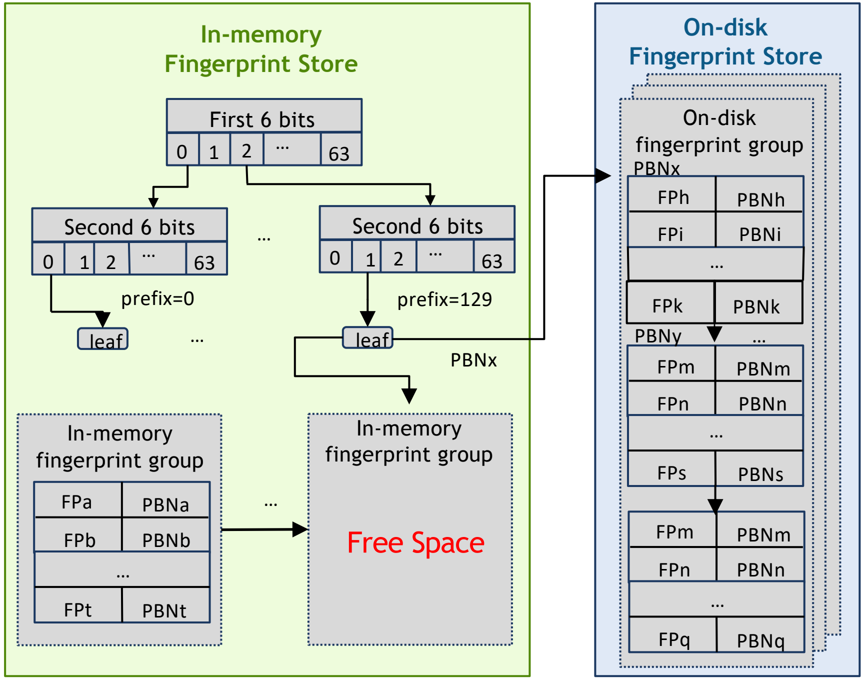
Fingerprints Migration

- Fingerprints migrate between memory and disk on demand
- In-memory store keeps the most recently used fingerprints
- Fingerprints evicted by groups to save I/Os



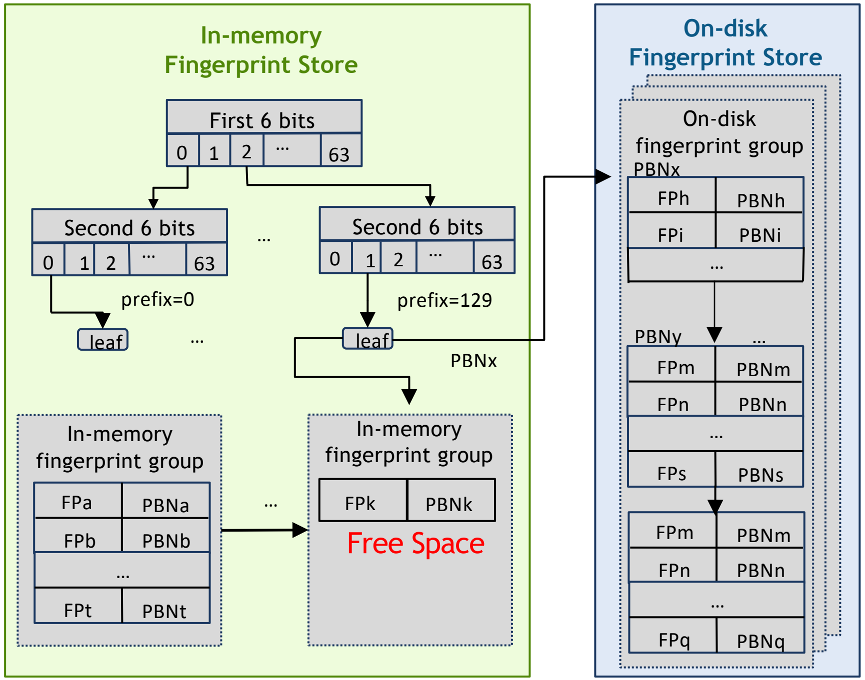
Fingerprints Migration

- Fingerprints migrate between memory and disk on demand
- In-memory store keeps the most recently used fingerprints
- Fingerprints evicted by groups to save I/Os



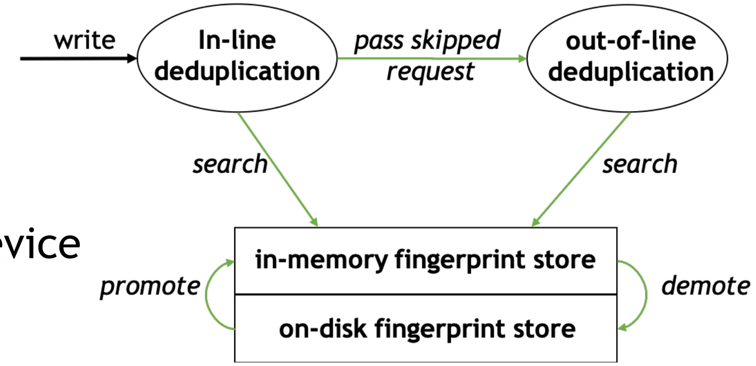
Fingerprints Migration

- Fingerprints migrate between memory and disk on demand
- In-memory store keeps the most recently used fingerprints
- Fingerprints evicted by groups to save I/Os

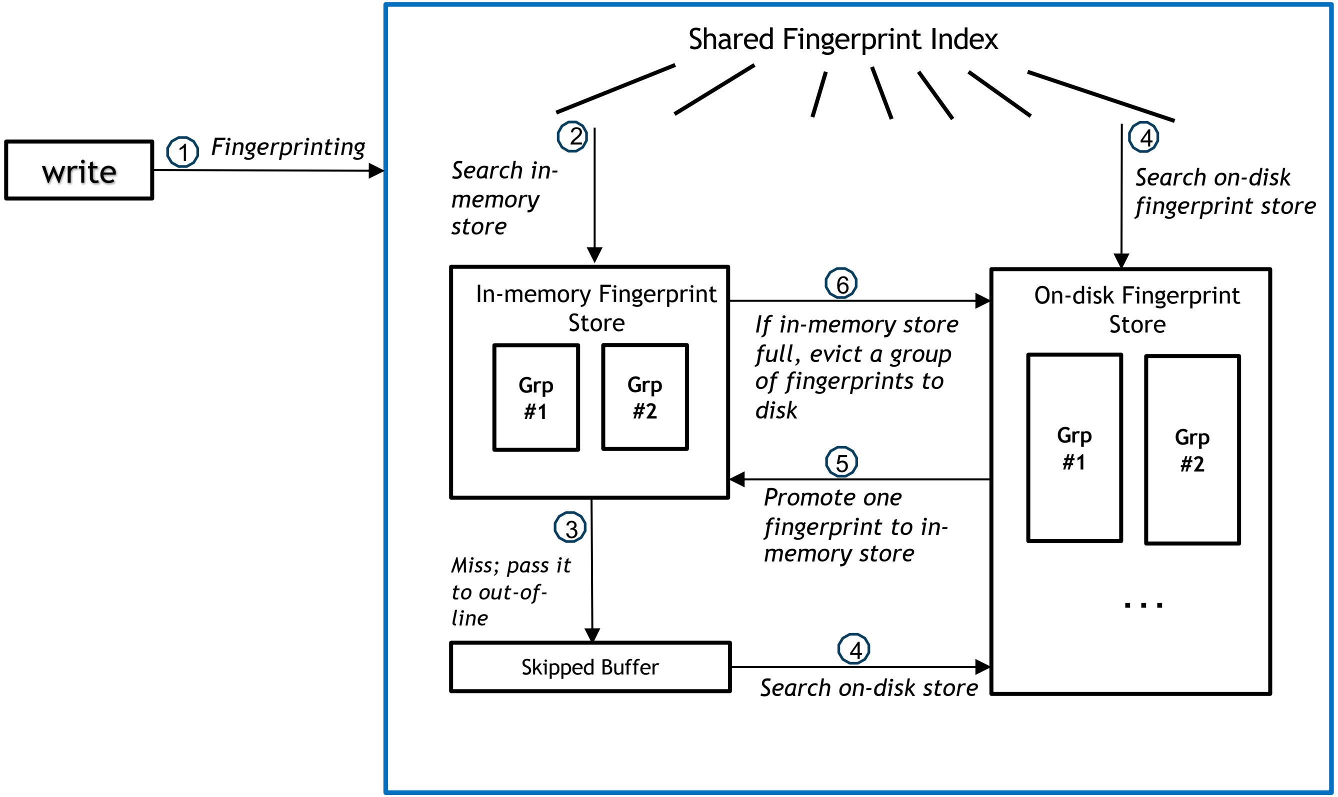


Hybrid Deduplication

- In-line Deduplication
  - Work in the I/O path
  - Immediately eliminate duplicate writes
  - Limited by the available resources on the device
- Out-of-line Deduplication
  - Work in background
  - Slowly and thoroughly eliminate duplicate data
  - Limited improvement on performance and endurance



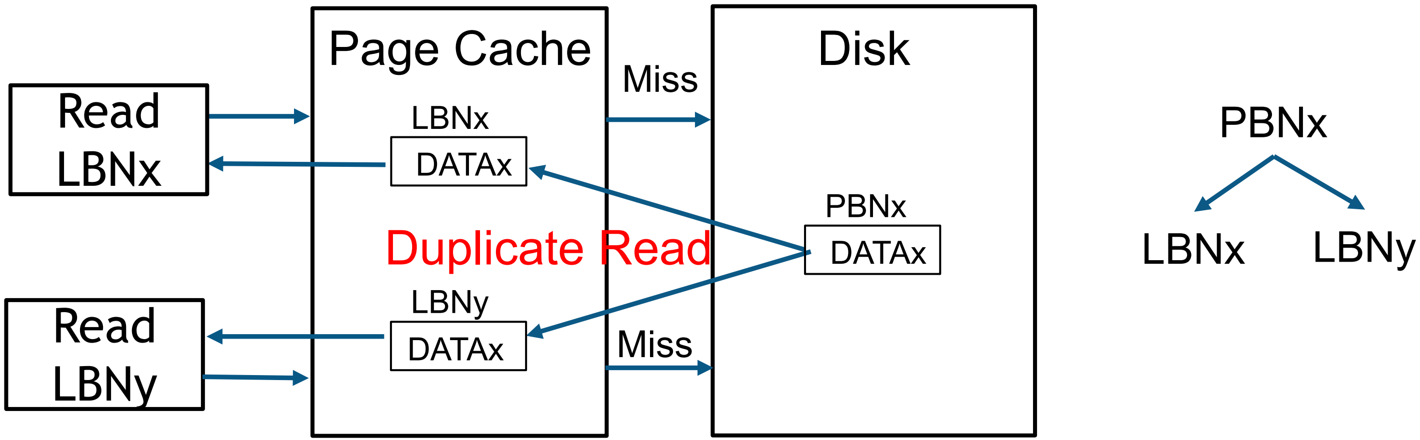
Write Path



write
1 Fingerprinting
Shared Fingerprint Index
2 Search in-memory store
4 Search on-disk fingerprint store
In-memory Fingerprint Store
Grp #1
Grp #2
6 If in-memory store full, evict a group of fingerprints to disk
On-disk Fingerprint Store
Grp #1
Grp #2
...
5 Promote one fingerprint to in-memory store
3 Miss; pass it to out-of-line
Skipped Buffer
4 Search on-disk store

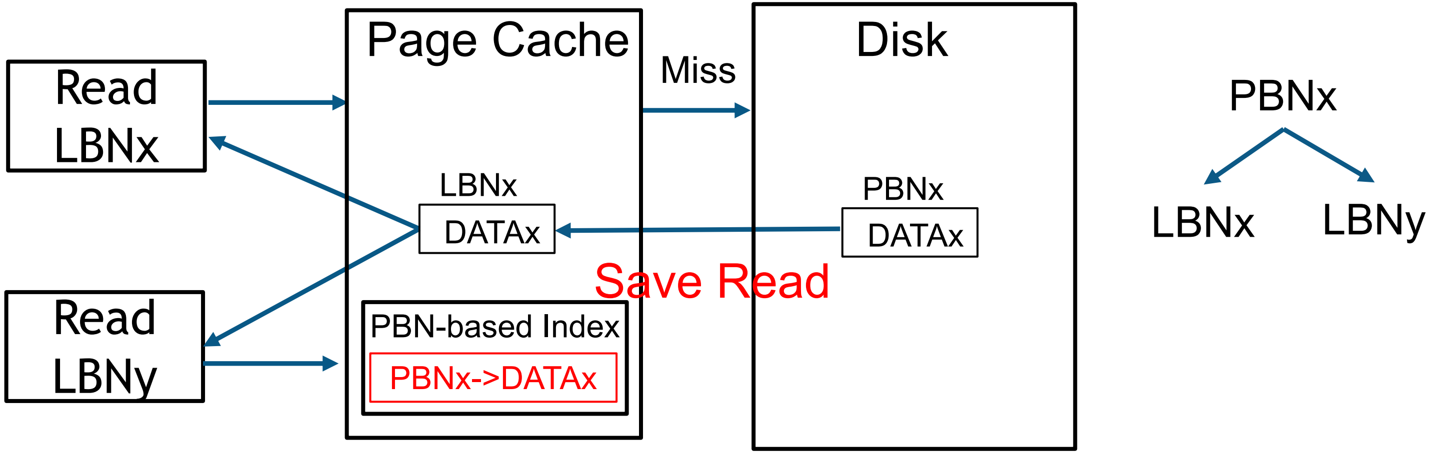
Read Path

- Native read path cannot utilize duplicate data in the page cache
  - Page cache is indexed by logical block numbers (LBNs)
  - Cannot avoid reads to different LBNs if the same data is already in the page cache



Optimized Read Path

- Page Cache Index
  - Map from PBNs to the corresponding pages in page cache
  - Use this index to find data for reads with different LBNs



Adaptive Deduplication

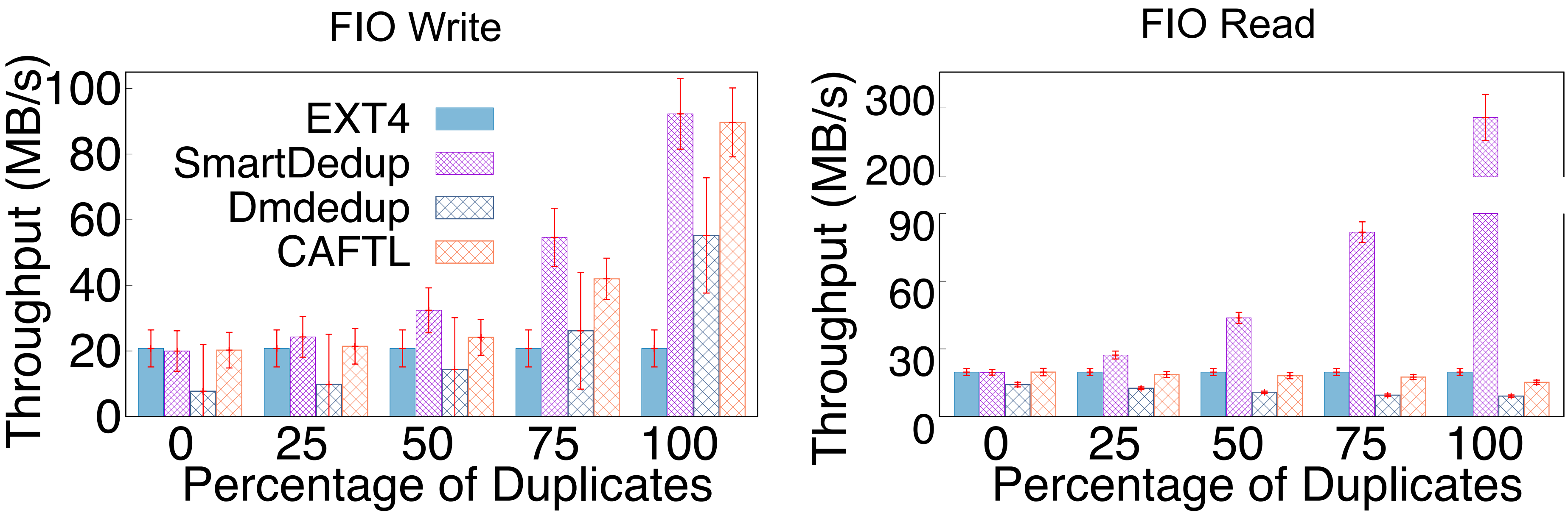
- Based on resource availability
  - Keep CPU and disk utilization under 100%
  - Proportional to available battery life
  - Completely disable deduplication in low battery state
- Based on the current duplication level
  - Gradually reduce the rate if the duplication level is dropping
  - Quickly increase the rate if the duplication level is growing

Evaluation

- Prototypes
  - EXT4 and F2FS (4KB fixed-size chunking)
- Testing devices
  - Nexus 5X and Raspberry Pi 3
- Benchmarks
  - FIO—intensive I/O benchmark
  - DEDISbench—
    workloads from real-world device images
  - Real-world device traces
- Baselines
  - Native EXT4 and F2FS
  - Dmdedup: block-level deduplication, flexible metadata backends [OLS 14']
  - CAFTL: deduplication for resource-constrained FTL layer [FAST 11']

	Nexus 5X	Raspberry Pi 3
CPU	Qualcomm Snapdragon 808	Broadcom BCM2837
RAM	2GB	1GB
Storage	32GB eMMC	16GB SDHC UHS-1
OS	Android Nougat	Raspbian Stretch Lite
Kernel	Linux 3.10	Linux 4.4
File System	EXT4	F2FS

FIO (on Nexus)



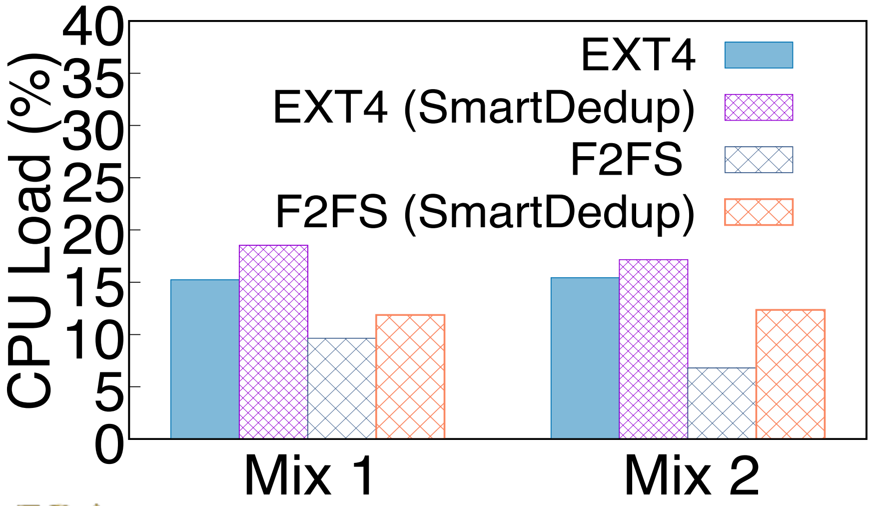
E.g., SmartDedup achieves 16.9% write speedup and 38.2% read speedup vs EXT4 with 25% of duplicates

FIO (Resource Usage)

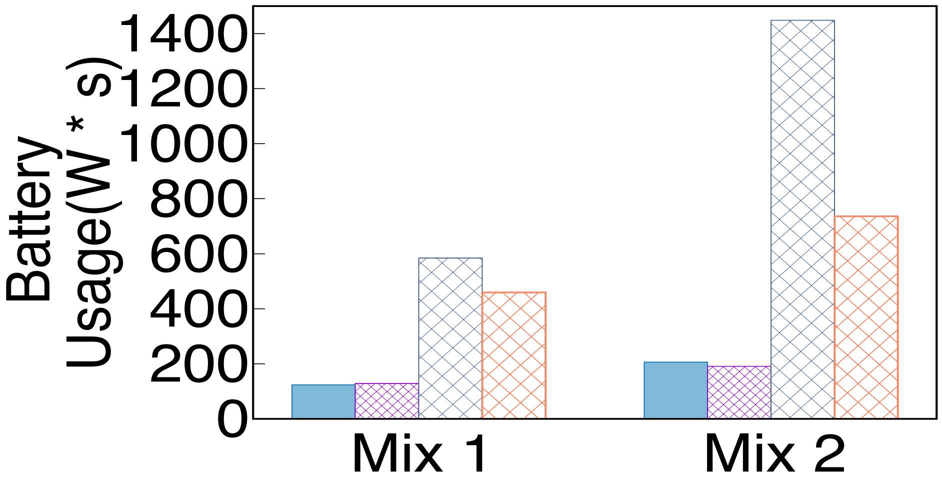
- Small CPU overhead (3.3%) vs. EXT4
- Save battery (49.2%) vs. F2FS
- Small memory footprint: 3.5MB

	I/O (GB)	Read/write ratio	Duplication ratio (%)
Mix 1	4	1	25
Mix 2	6	2	25

CPU Load

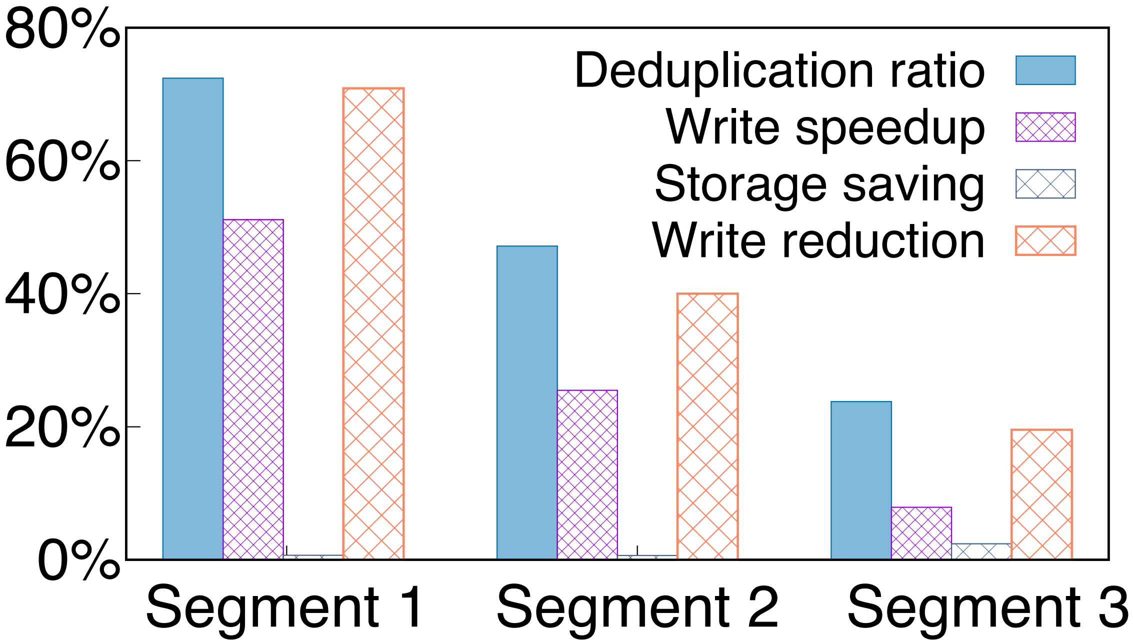


Battery usage



VISA

Trace Replay (on Nexus)



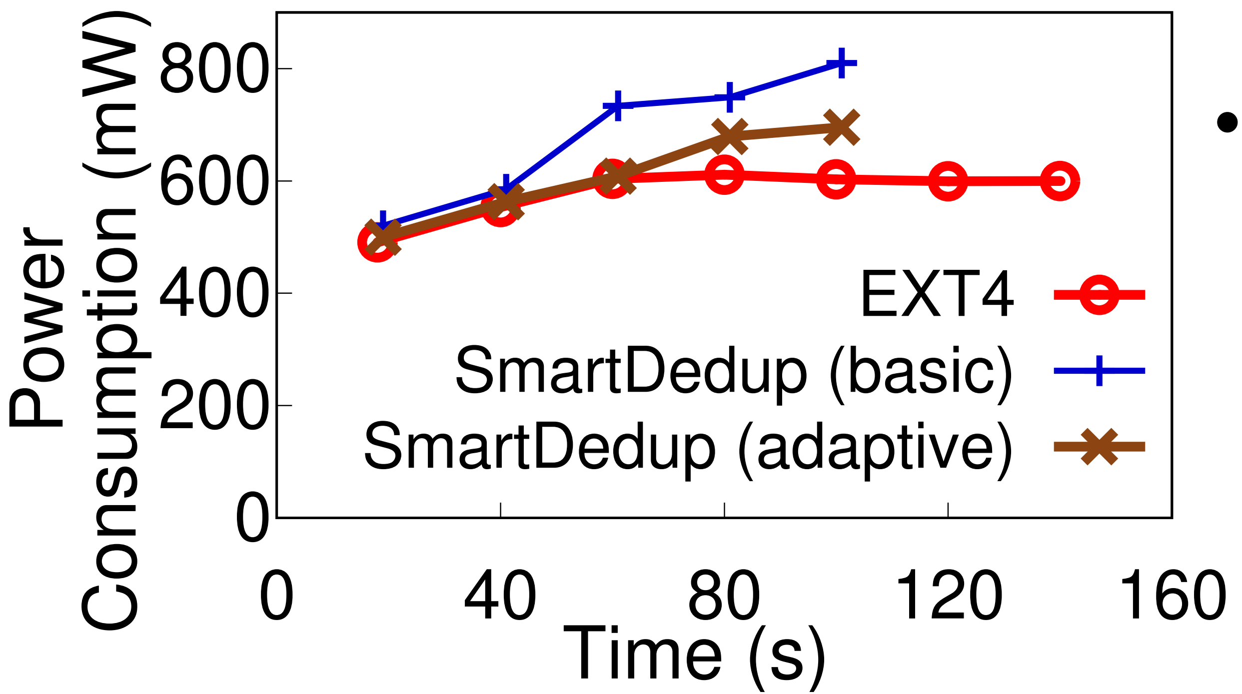
ID	Trace src	Write (GB)	Duplication ratio (%)	Read/write ratio
1	4	17.2	75.8	1.5
2	6	12.4	47.9	2.2
3	2	9.1	26.4	6.8

- SmartDedup achieves
  - Up to 51.1% write speedup
  - Up to 70.9% write reduction

VISA

Adaptive Deduplication (on Nexus)

Adapting to duplication level of current workload



- SmartDedup (adaptive)
  - Saves 14% power overhead
  - With only 8% loss in deduplication ratio

VISA

Conclusions

- Deduplication is important to smart devices—performance, utilization, endurance
- SmartDedup achieves significant performance and endurance improvement with low resource cost
- Trace: http://visa.lab.asu.edu/traces

- Acknowledgements
  - Colleagues @ the ASU VISA Lab
  - National Science Foundation

- Welcome to SmartDedup's poster tonight 6:30pm