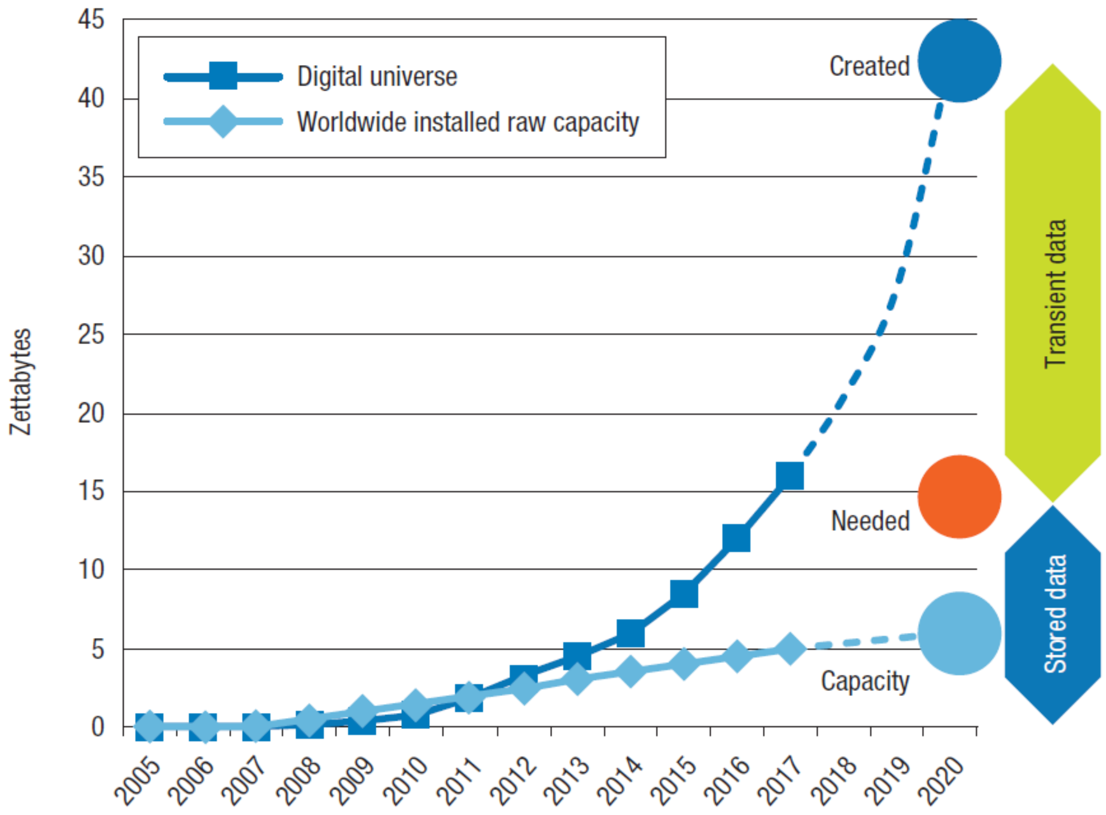
SES-dedup: a Case for ECC-based SSD Deduplication

Zhichao Yan[1,2], Hong Jiang[1], Song Jiang[1], Yujuan Tan[3], Hao Luo[4]
The University of Texas-Arlington[1], Hewlett Packard Enterprise (Nimble Storage)[2], Chongqing University[3], Twitter[4]

MSST 2019

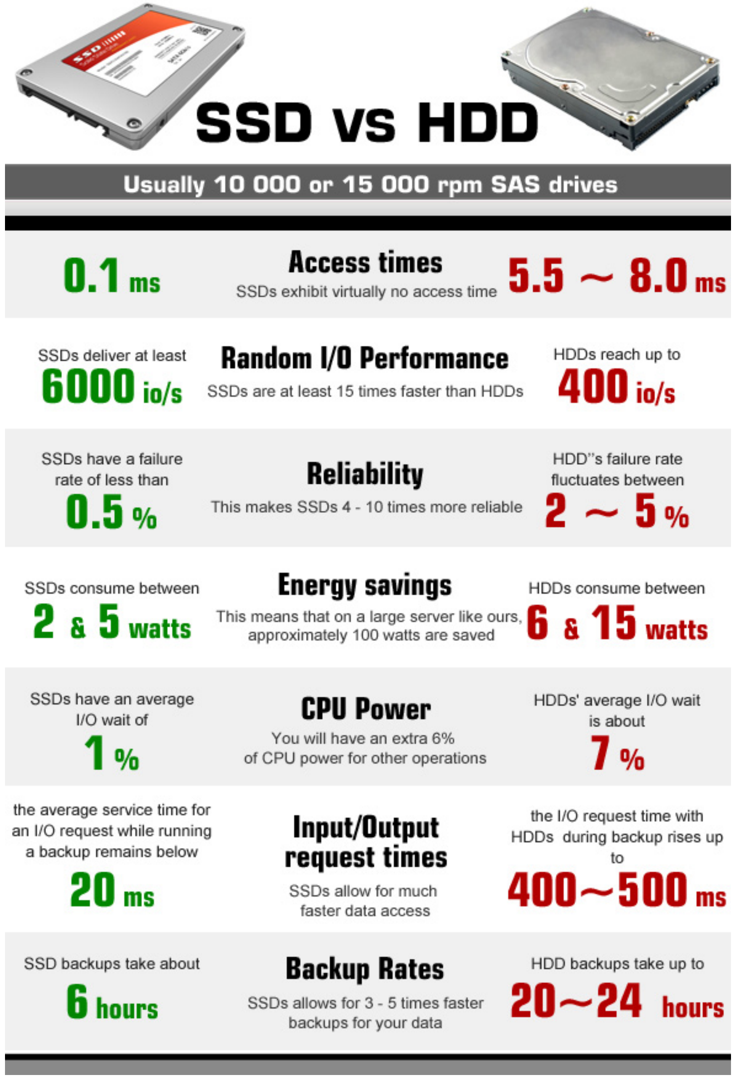
Massive Data Need to Be Stored

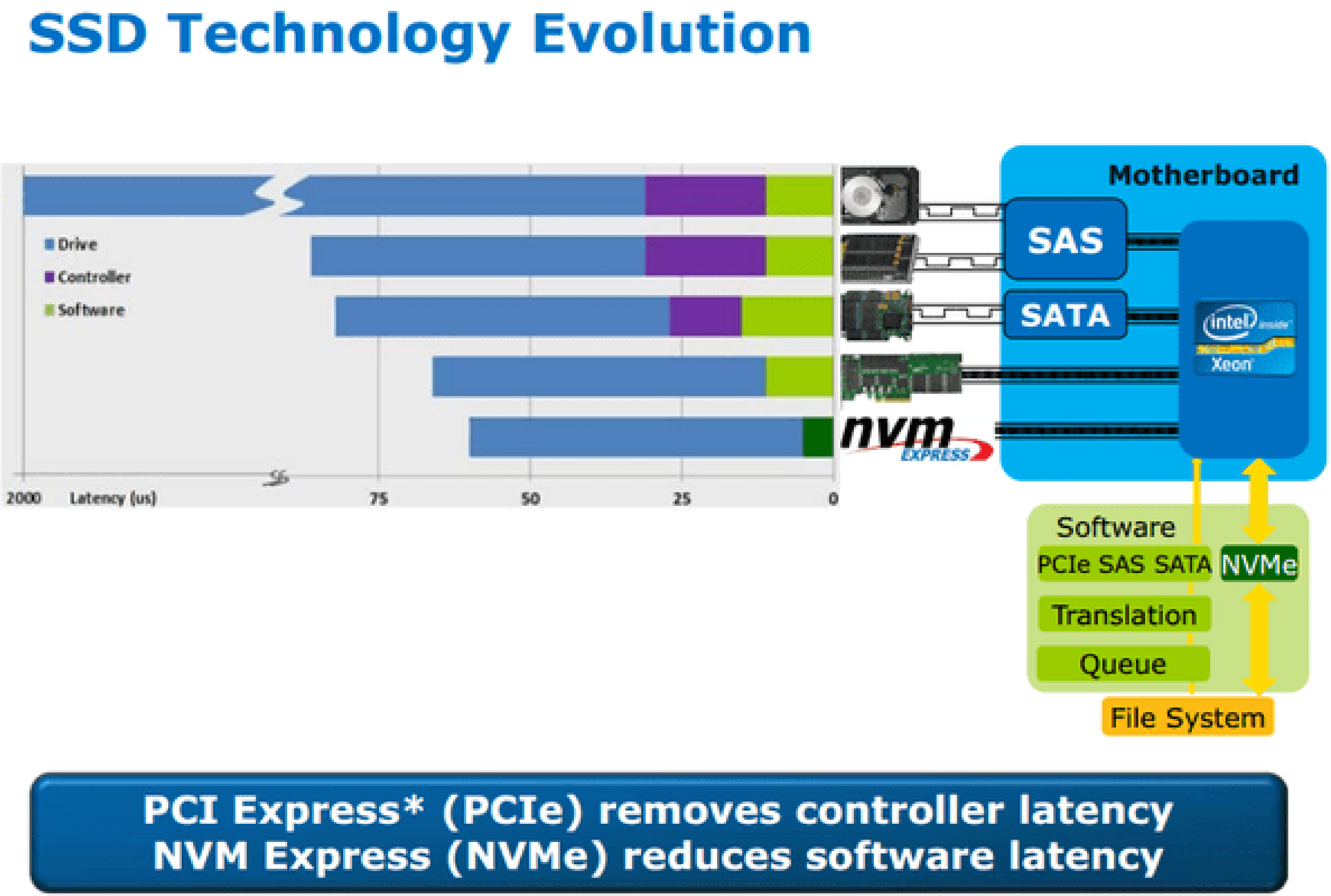
"The world's most valuable resource is no longer oil, but data" The Economist, May 2017



Seagate's projected gap between storage supply and demand

SSDs have taken the primary storage by storm





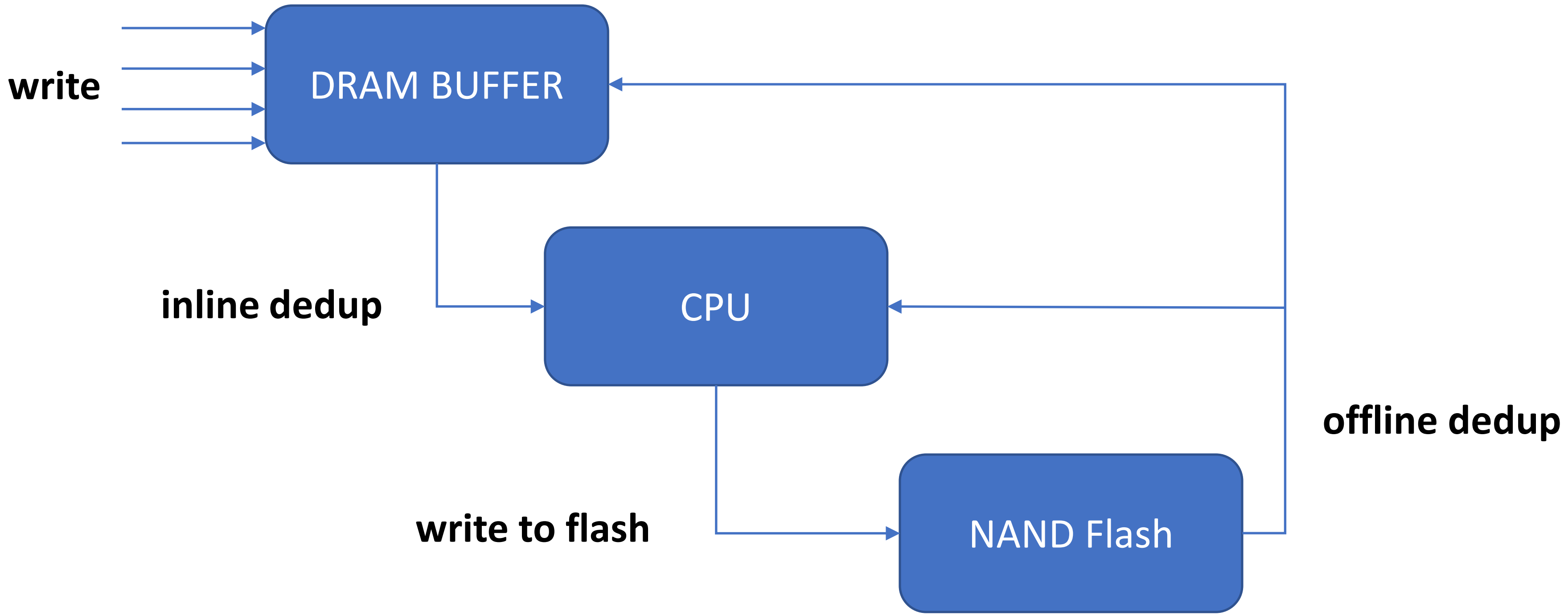
Integrating Deduplication within SSDs

- Avoid duplicated writes to NAND flash chips → lower P/E 😄
- Improve the reliability with lower P/E😄
- Increase the effective capacity 😄
- Help behind-the-scenes maintenance tasks such as WL and GC 😄
- Computation and memory costs 😫
- Data movements 😫
- Existing work:
  CAFTL (FAST 2011), Dedup in SSD (MSST 2012)
  Pearls of wisdom : fixed-size chunking, adopting weak hashing (ECC)

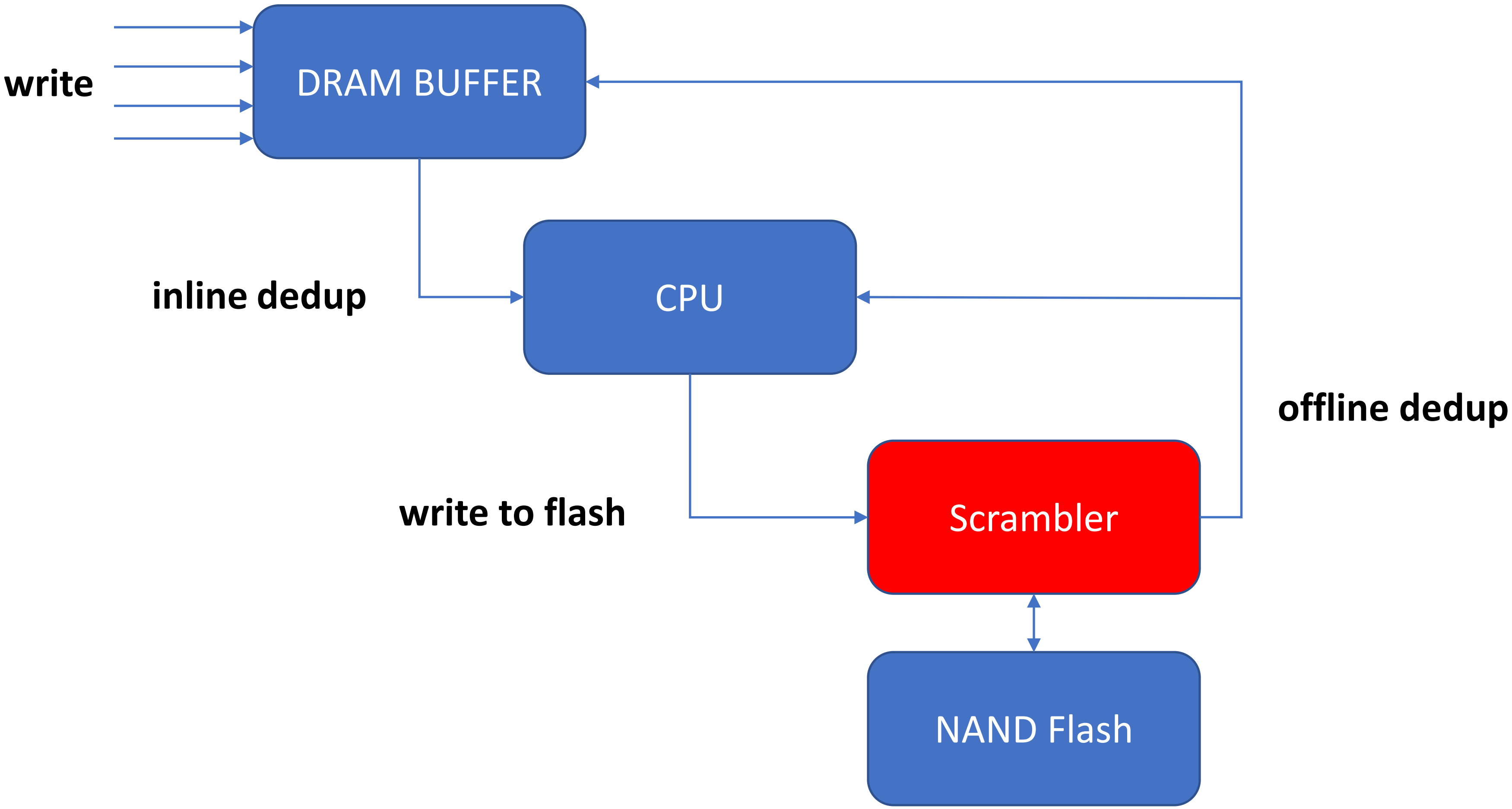
Agenda

- Problem
- SES-dedup
- Evaluation
- Summary

A Typical Work Flow for Existing SSD Deduplication

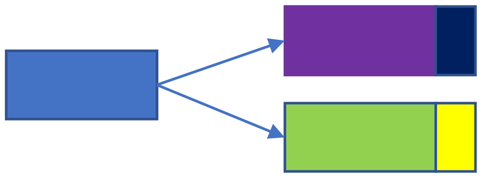


The Ignore Scrambler Module



Problem: The Ignored Scrambler Module

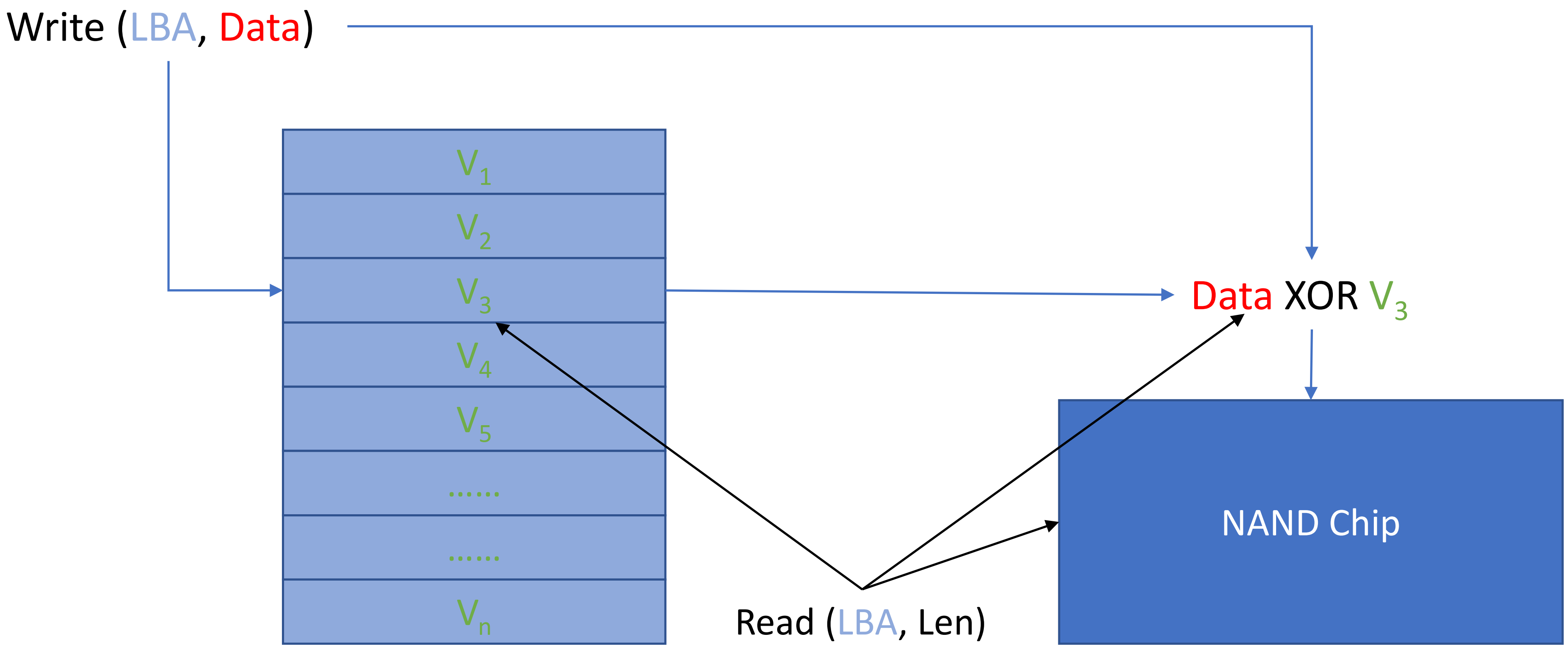
- NAND chip's raw bit error rate will increase when similar patterns are written repeatedly (skewed storage reliability).
- As a result, a randomized module (scrambler) is added to randomized the data before storing to the NAND chips



Different ECC value ➔ different data on NAND flash but might be the same content

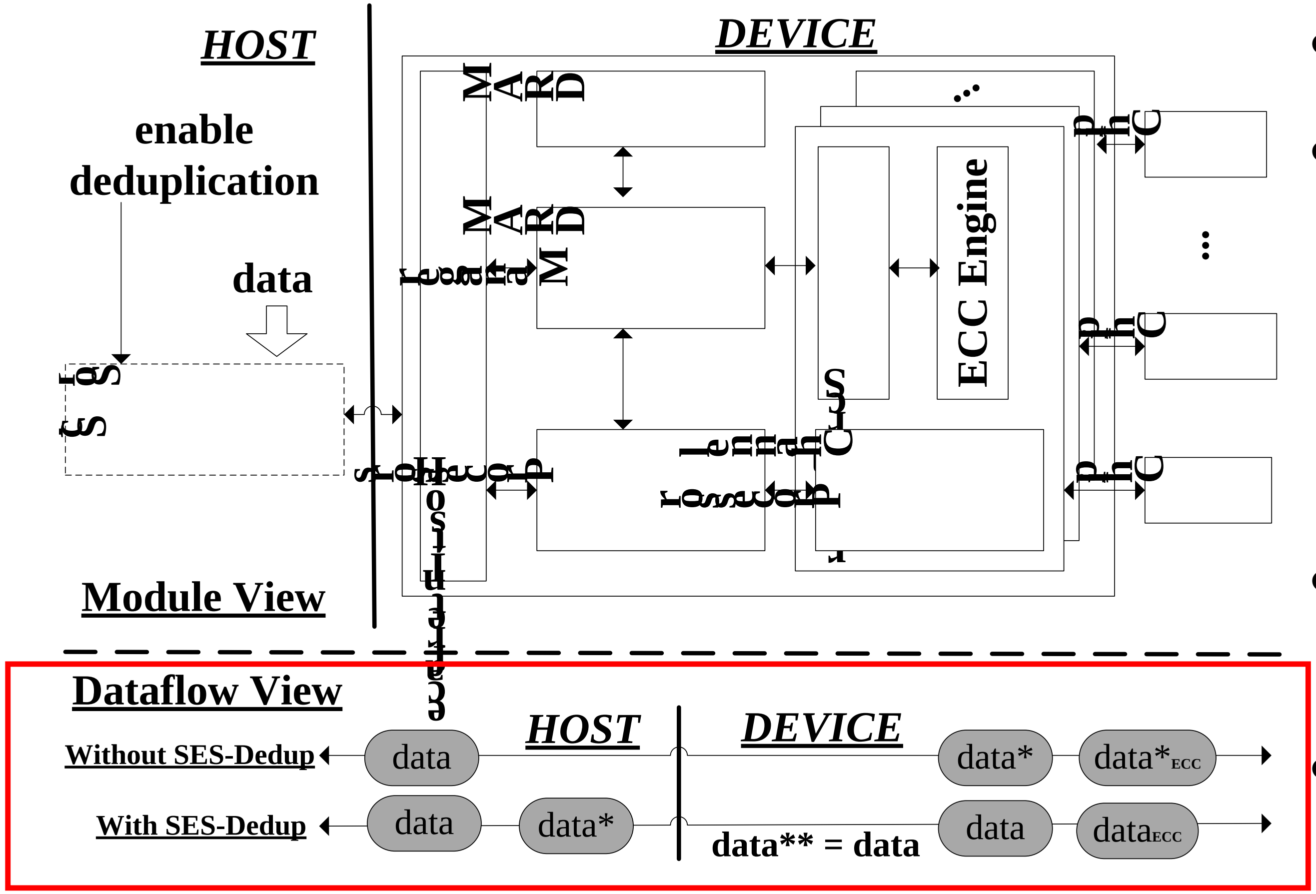
- ECC is calculated by data written to NAND chip, so the randomized data will render ECCs useless as the deduplication fingerprints
- Need to reconsider the deduplication workflow in SSD

LBA-based Scrambler



Linear Feedback Shift Register (LFSR）

Scrambler-resistant ECC-based SSD deduplication: A Host-side Design



- fixed-size chunking
- weak hashing(ECC) plus byte-to-byte comparison by exploiting the asymmetric feature of the read and write operation
- Reconstruct a software scrambler at the host
- Selectively bypassing the hardware scrambler

More suitable for personal usage that provides a flexible on-demand interface to enable the deduplication feature on SSDs.

Device-side SES-dedup

- $([V_{data}] \oplus [V_{scrambler}]) \times [M_{encoding}] = [ECC]$
- $\left([V_{data}] \times [M_{encoding}]\right) \oplus \left([V_{scrambler}] \times [M_{encoding}]\right) = [ECC]$
- $[V_{data}] \times [M_{encoding}] = [ECC] \oplus \left([V_{scrambler}] \times [M_{encoding}]\right)$
- Store $\left([V_{scrambler}] \times [M_{encoding}]\right)$ in a lookup table
- All identical input data’s encodings can be recalculated, which can be used for deduplication
- Extra lookup table plus trivial computation
- Suitable for data center with lots of SSDs

Evaluation

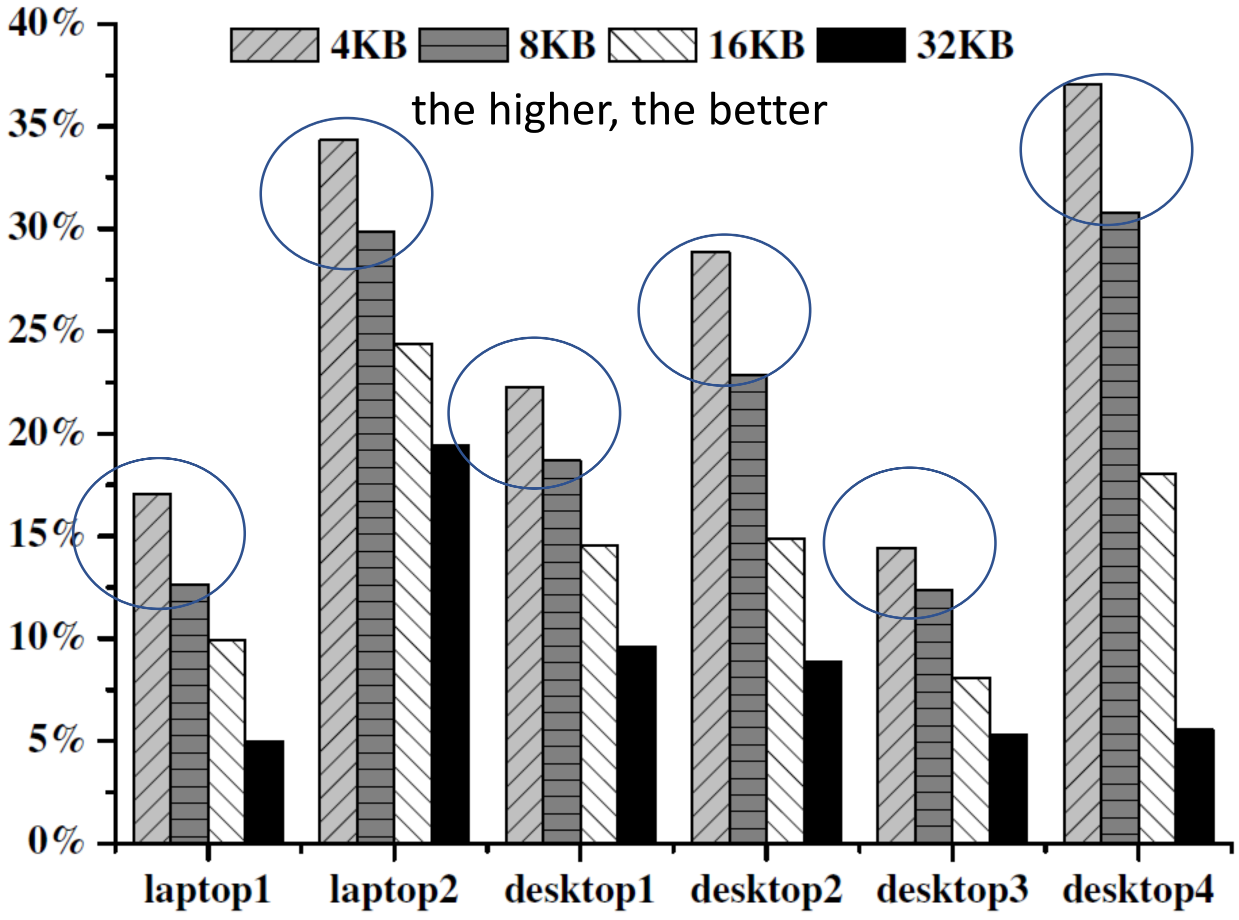
- GEM5 full system simulator (A 1.6 GHz X86 CPU plus an eight-bank 8 GB DDR3-1600 DRAM) + FlashSim SSD model with ECC-based deduplication functions.

Description	Configuration
Flash Page Size	8 KB
Pages per Block	256
Block per Plane	256
Plane per Package	8
Number of Packages	8
Garbage Collection Threshold	5%
Flash Erase Latency	1.8 ms

NAND Type	Read	Write	SHA-256
SLC	23.4 us	262.6 us	226.5 us
MLC-1	33.5 us	390.0 us	
MLC-2	43.3 us	1084.4 us	

Shrink stimulated SSD size to 32 GB with 64 MB DRAM to make our collected data easily saturate its capacity. Each codeword of 1 KB is protected by a code rate of 32/33 LDPC code

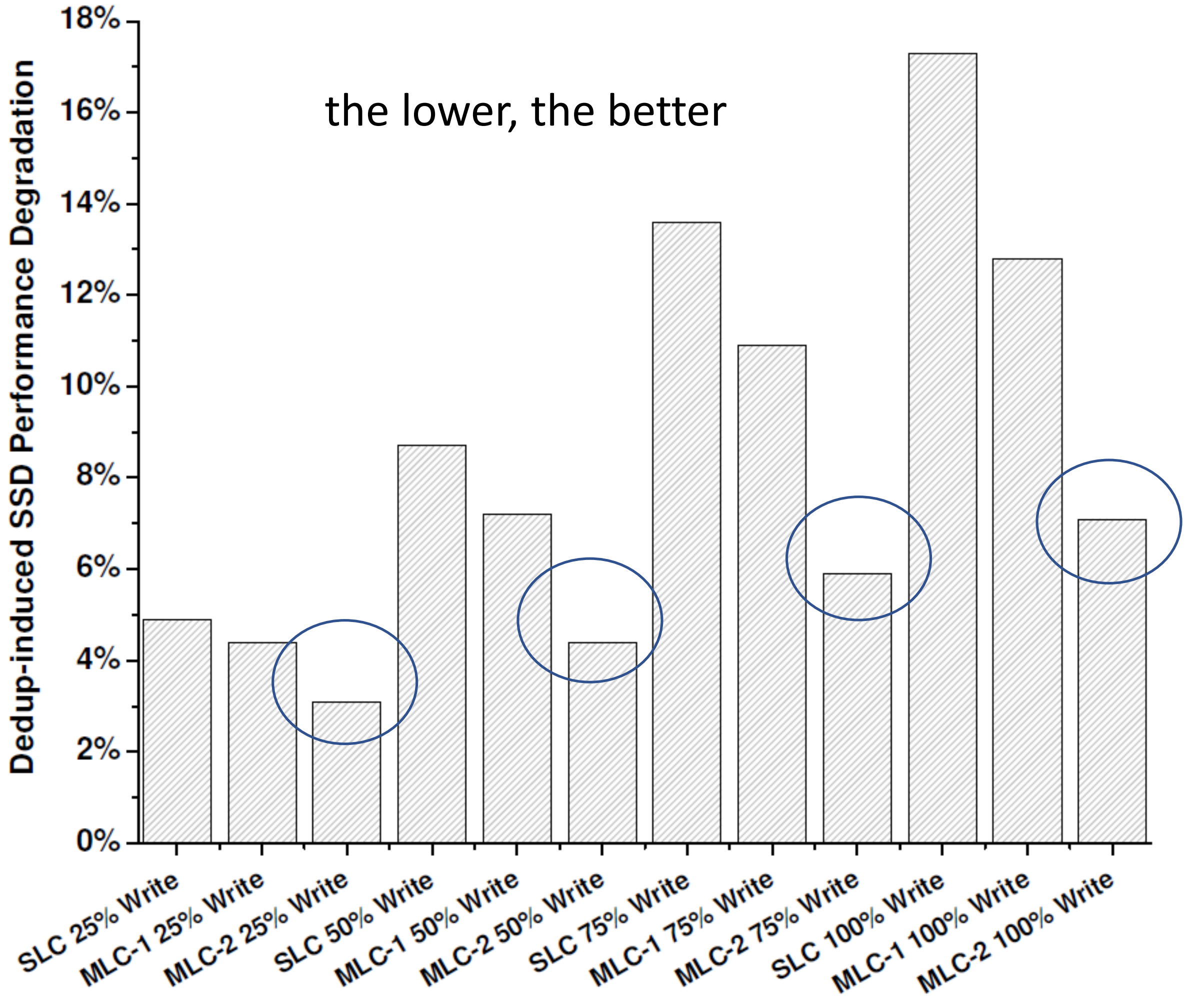
Redundancy with Chunking Granularity Study



Data redundancy rates of fixed-size chunking

- Exist a lot of redundant data in these datasets, which is up to 37.0% on Desktop 4.
- Most redundant data can be found in 8 KB chunks comparing to 4 KB chunking, whose size is close to modern SSD's page size.
- Plan to explore the sub-page ECC dedup in the future

Performance Overheads on SHA-256



Mixed R/W workloads to process a data set without any deduplicatable pages to learn its overheads caused by SHA-256

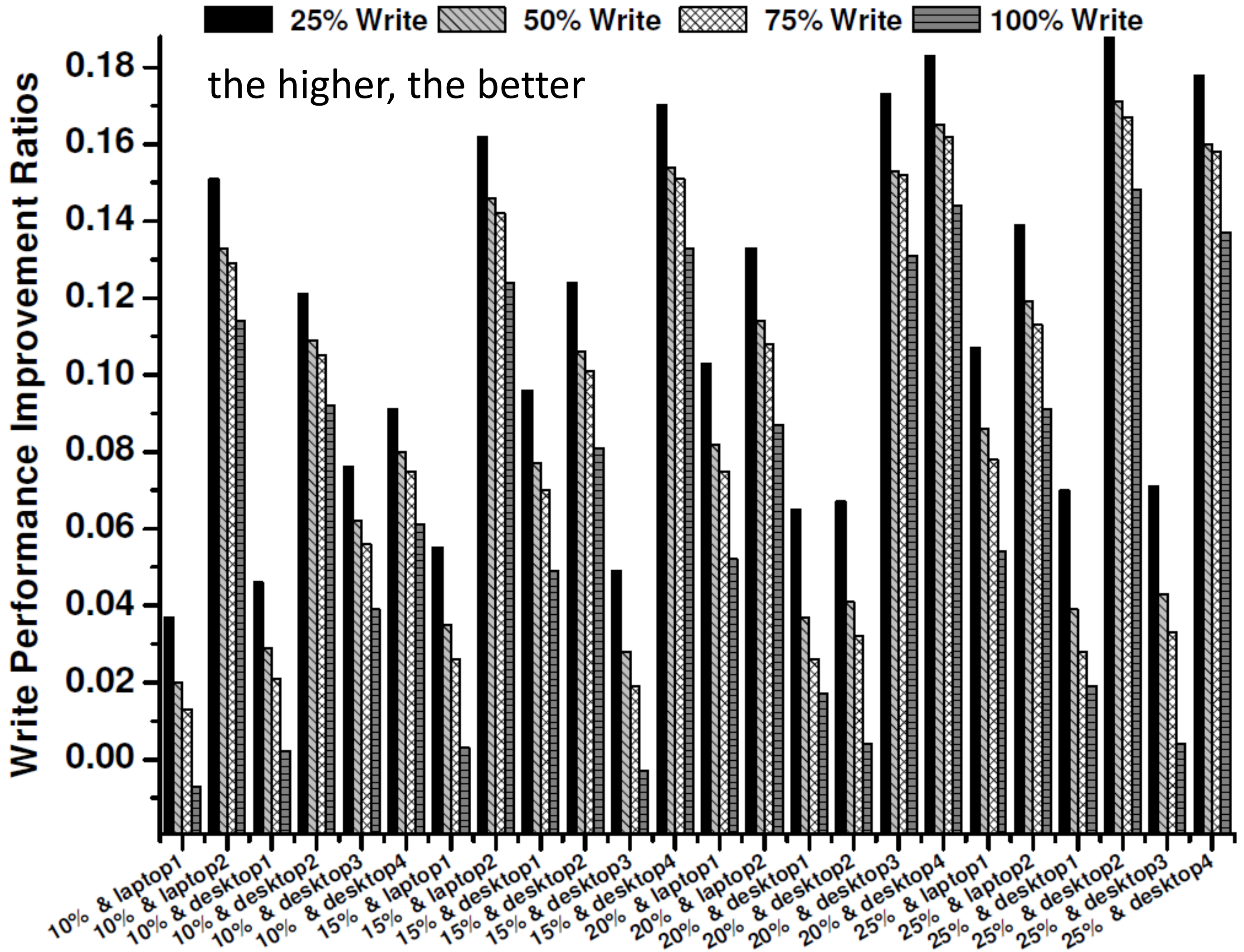
SSD performance degrades on different types of NAND flash chips with different mixed random read-and-write workloads on fixed chunking of size 8 KB

Skew-distributed Duplicated Pages

	Hot FP Ratio	Ratio in Redundant data
laptop 1	17.6%	74.1%
laptop 2	13.8%	86.3%
desktop 1	15.8%	79.8%
desktop 2	14.9%	81.1%
desktop 3	18.8%	72.1%
desktop 4	12.7%	89.3%

- Hot FP: reference count > 2
- Small portion of hot FPs occupy most redundant data
- Put the hot FPs in the memory, and further store partial ECC to reduce the FP's memory footprint
- Replace high-cost write operations with low-cost read operations to exploit the asymmetric latencies of read and write operations
- 4.8 MB out of 64 MB extra DRAM space (7.5%)

Performance Improvements on Different Sizes of Fingerprint Table under Simulated SLC SSD



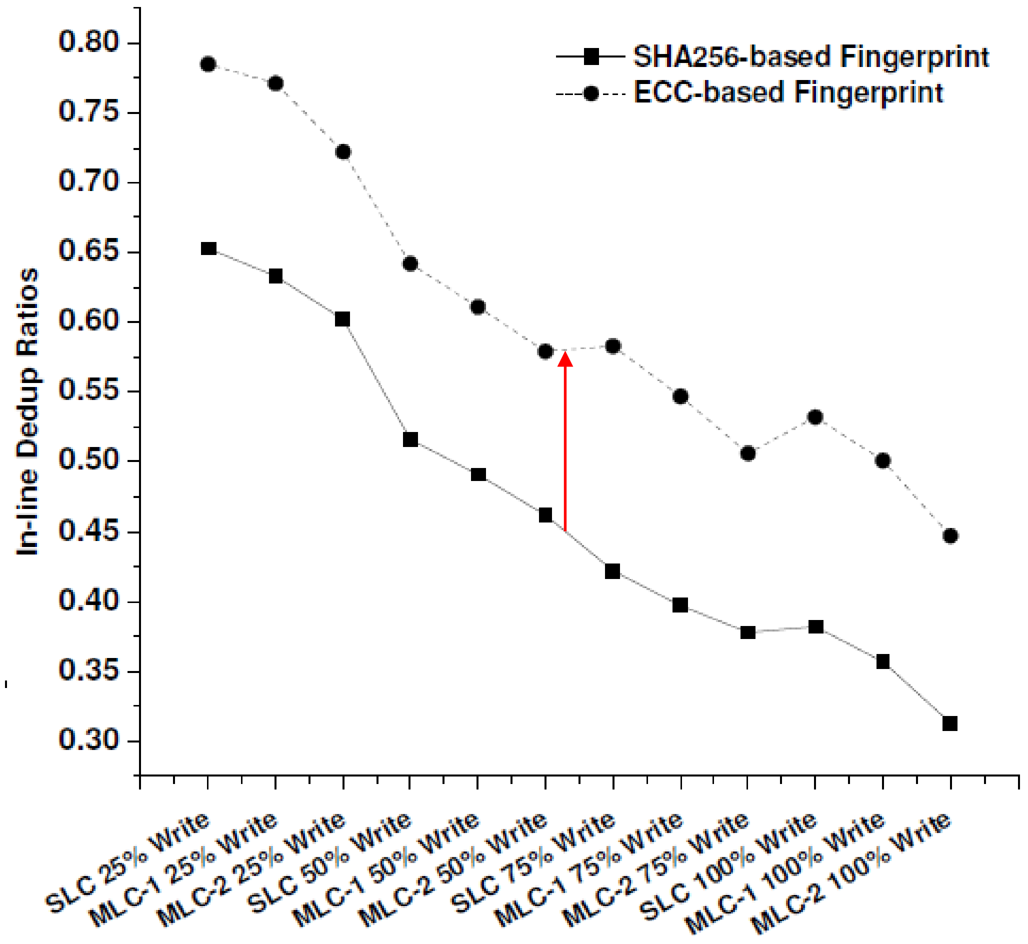
- different data sets → different data distributions → different random write perf improvements
- 15% of max table size can obtain the best price/performance ratio
- SES-dedup get 17.0% random write performance under this setting.

Inline and Offline Dedup: Host-side SES-dedup

Inline and offline deduplication processing redundancy data ratios on the host-side SES-Dedup system with 100% random write workload

Data Set	In-line Dedup			Off-line Dedup			Duplicate Ratio
	SLC	MLC-1	MLC-2	SLC	MLC-1	MLC-2	
laptop1	7.1%	6.5%	5.4%	5.5%	6.1%	7.2%	12.6%
laptop2	17.4%	16.1%	12.9%	12.5%	13.8%	17.0%	29.9%
desktop1	11.0%	9.9%	8.1%	7.7%	8.8%	10.6%	18.7%
desktop2	13.7%	12.1%	9.9%	9.2%	10.8%	13.0%	22.9%
desktop3	6.5%	6.1%	5.2%	5.8%	6.2%	7.1%	12.3%
desktop4	18.2%	16.9%	13.6%	12.6%	13.9%	17.2%	30.8%

Inline and Offline Dedup: Device-side SES-dedup



- Different from the host-side approach, the device-side SES-Dedup system will add the ECC processing latency to support its deduplication function
- Majority of duplicated pages can be detected and removed inline while leaving some pages to be processed offline in the ECC-based SES-Dedup approach
- Process 19.9% to 42.8% more duplicated data inline than SHA256-based approach, avoiding more P/E operations

Summary

- Revisit the ECC-based SSD deduplication
- Consider the impacts of randomization module
- Propose two SES-dedup designs to bypass the scrambler module
- Verified their effectives on the simulated platform
- SES-dedup approach can remove up to 30.8% redundant data with up to 17.0% performance improvement by replaying our collected data traces in the SSD simulator.

Q&A

Thanks!