Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23)
Workshop on Accelerating Analytics and Data Management Systems (ADMS'23)

Towards MRAM Byte-Addressable Persistent Memory in Edge Database Systems

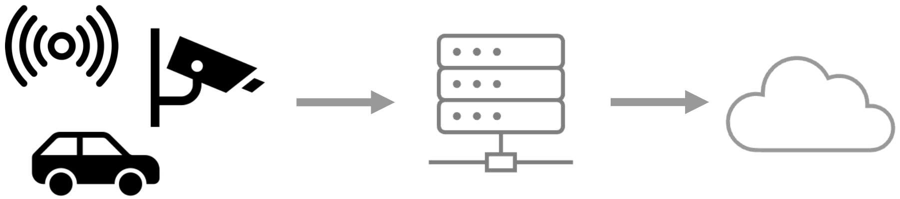
Luís Ferreira, Fábio Coelho, José Pereira
INESC TEC & University of Minho

August 28, 2023

Towards MRAM Byte-Addressable Persistent Memory in Edge Database Systems

Endpoints

80% of generated data by 2025



E.g., IoT devices

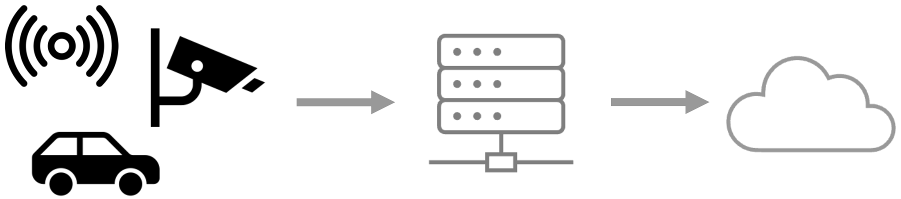
Why place data at the endpoints?

- Connectivity
- Privacy
- Latency
- Energy
- Scalability

Towards MRAM Byte-Addressable Persistent Memory in Edge Database Systems

Endpoints

80% of generated data by 2025

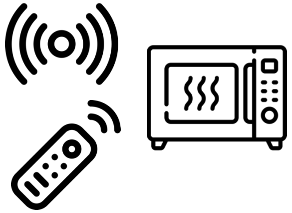


E.g., IoT devices

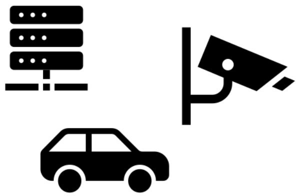
Why place data at the endpoints?

- Connectivity
- Privacy
- Latency
- Energy
- Scalability

Microcontrollers (MCUs)



Microprocessors (MPUs)

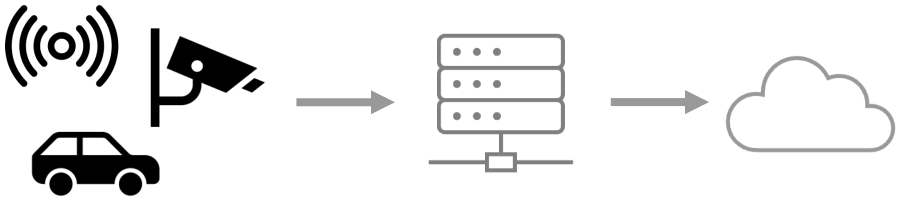


Very **limited resources**

Towards MRAM Byte-Addressable Persistent Memory in Edge Database Systems

Endpoints

80% of generated data by 2025



E.g., IoT devices

Why place data at the endpoints?

- Connectivity
- Privacy
- Latency
- Energy
- Scalability

Microcontrollers (MCUs)

Microprocessors (MPUs)

Very **limited resources**

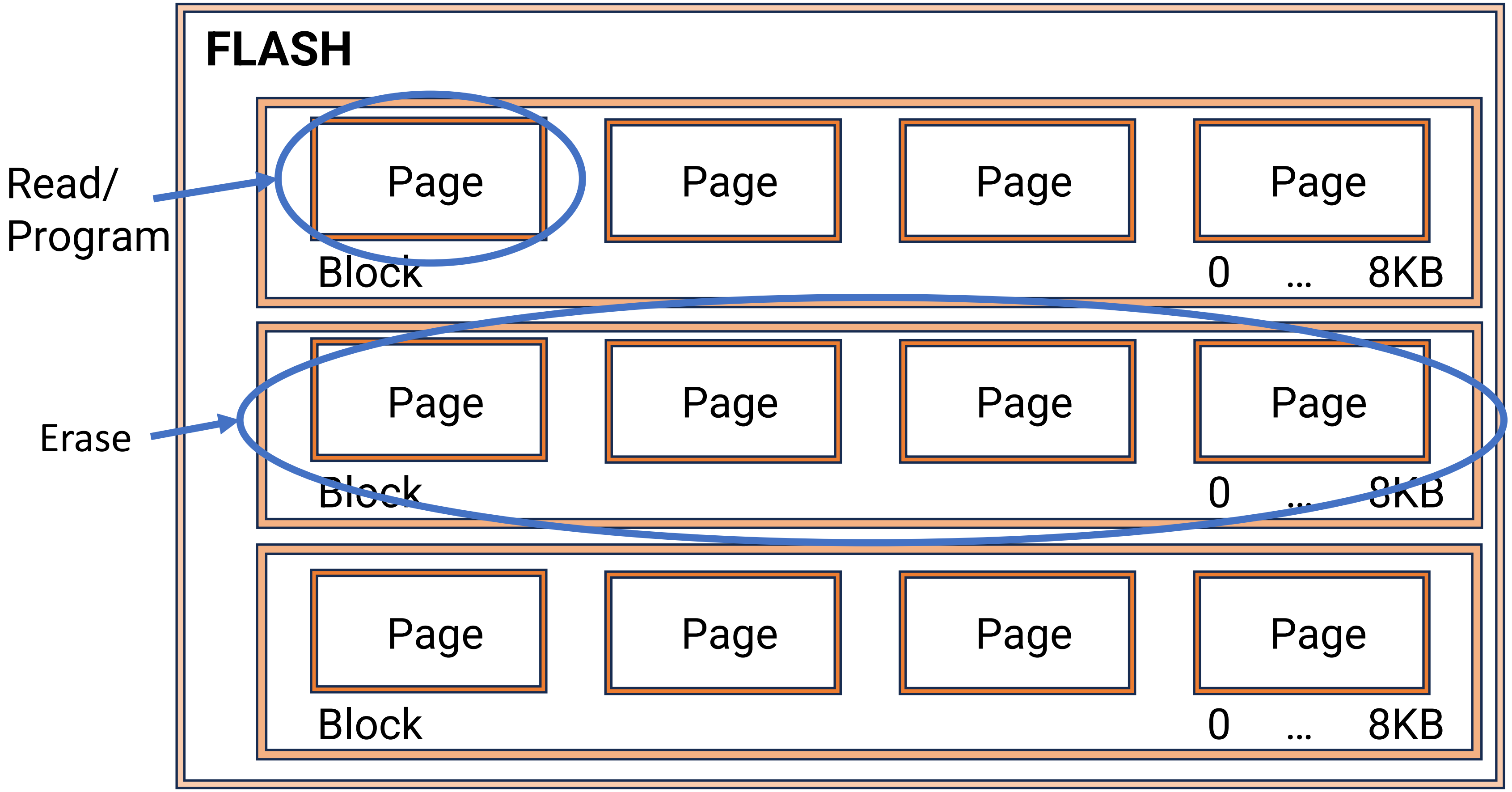
Storage medium

~~FLASH~~
~~E.g., SD Card,~~
~~embedded~~

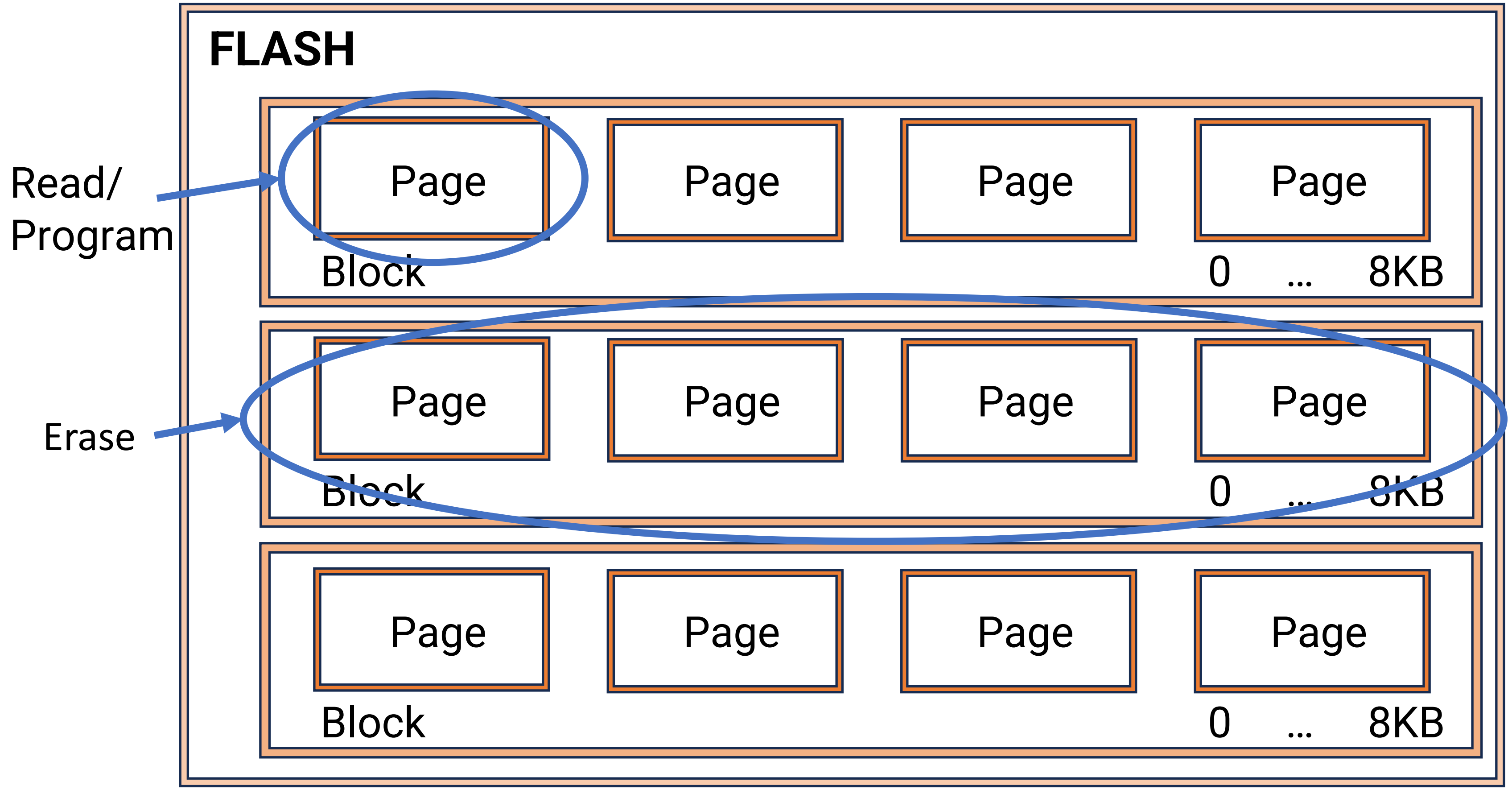
→

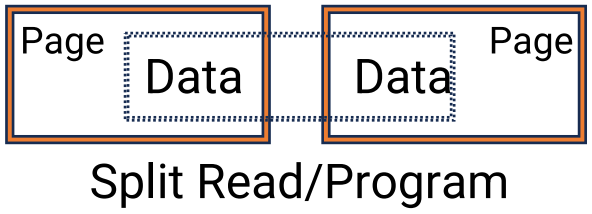
MRAM
Persistent Memory

Background - FLASH

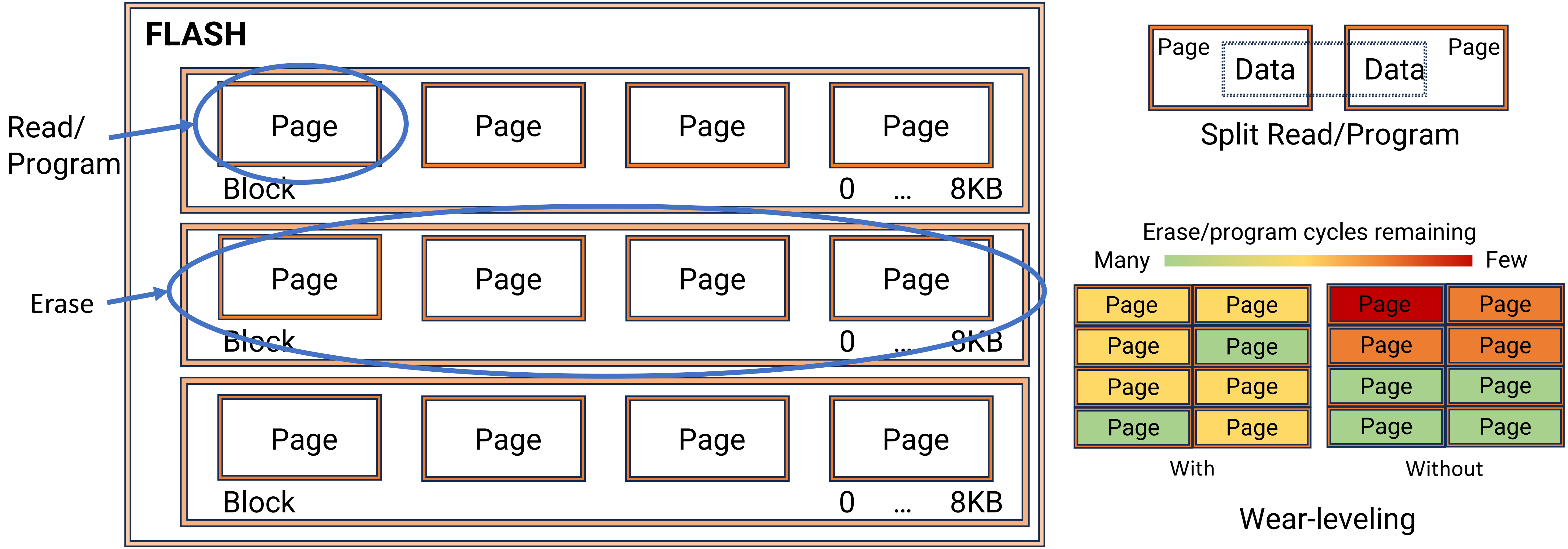


Background - FLASH

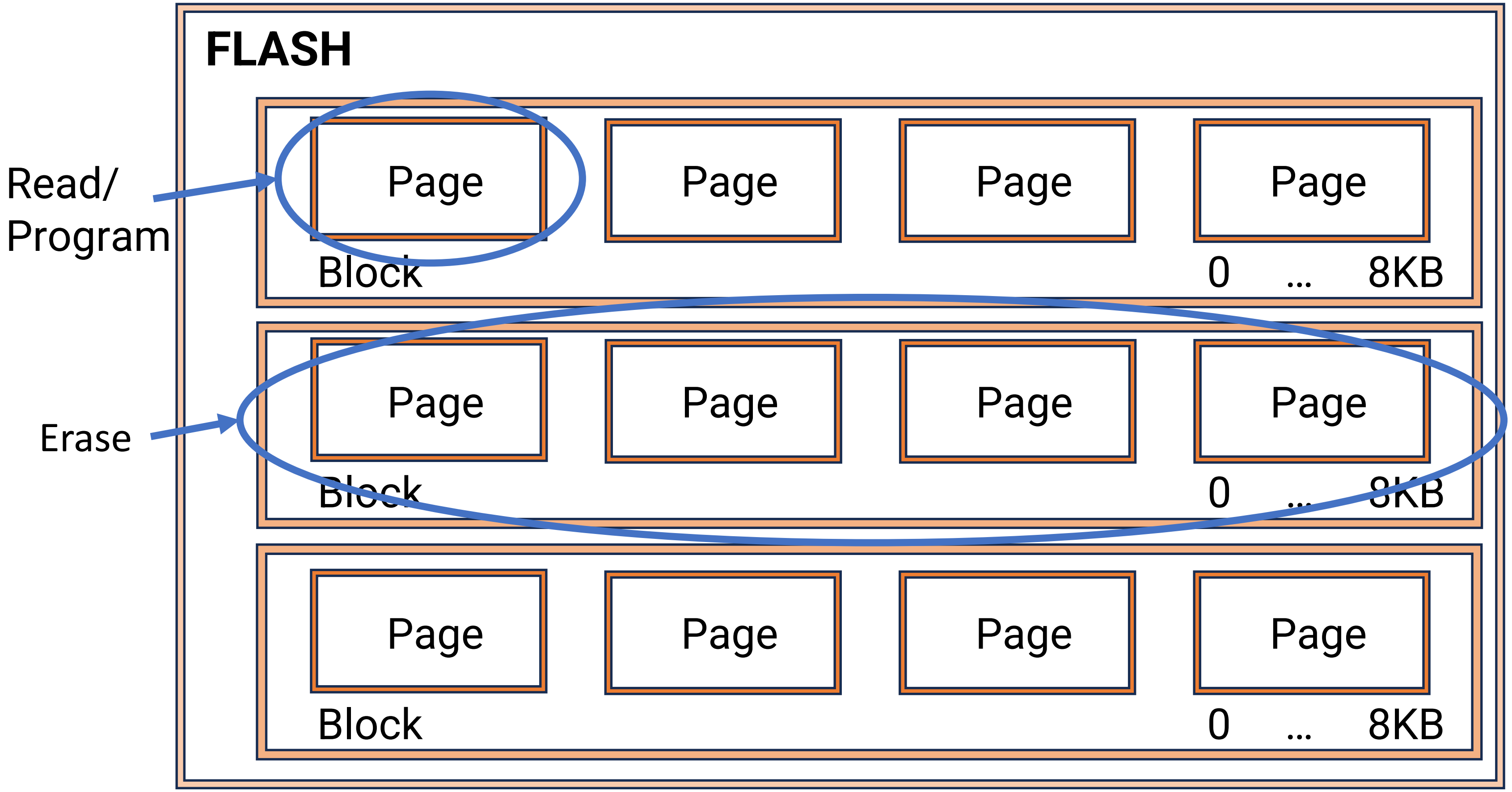


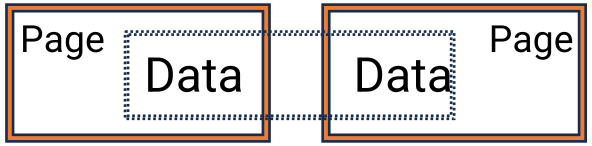


Background - FLASH

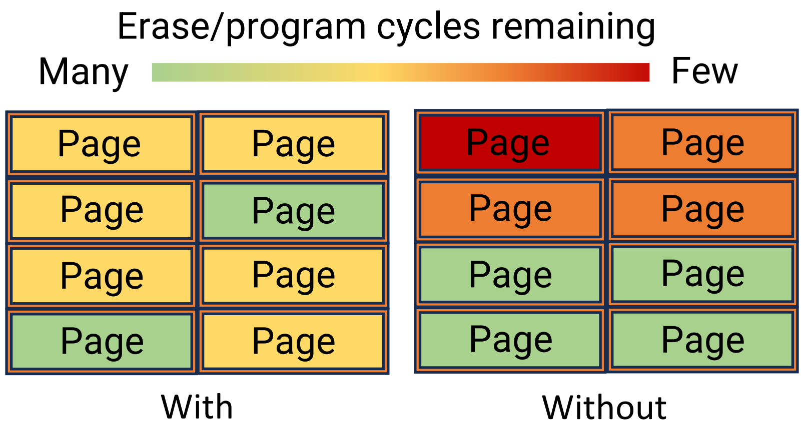


Background - FLASH





Split Read/Program



Wear-leveling

NAND vs NOR

Problem

FLASH physical limitations:

- I/O operations incur extra computational overhead
- Small I/O operations are slow
- Limited endurance

- Asymmetric performance for random and sequential accesses

Specification comparison: MRAM vs FLASH

Table 1
Storage devices' characteristics comparison.

	Read	Write	Erase	Capacity	Endurance	Energy	Cost
MRAM							
$M1$) AS3004316 [24]	57 MB/s	57 MB/s	N/A	4 Mb	100T	1.58 nJ/B	6.63 €/Mb
$M2$) MR4A16BMA35 [25]	57 MB/s	57 MB/s	N/A	32 Mb	Inf	1.58 nJ/B	1.99 €/Mb
$M3$) EMxxLx [26]	400 MB/s	400 MB/s	N/A	64 Mb	Inf	0.895 nJ/B	0.84 €/Mb
$M4$) EMD4E001GAS2 [27]	2.6 GB/s	2.6 GB/s	N/A	1 Gb	0.01T	0.523 nJ/B	0.098 €/Mb
FLASH							
NAND FLASH [21]	235 MB/s	23.5 MB/s	737 MB/s	128 Gb	60K	7.02 nJ/B	0.0012 €/Mb
NOR FLASH [28]	337 MB/s	2.5 MB/s	0.65 MB/s	512 Mb	100K	66.8 nJ/B	0.023 €/Mb

Background - MRAM

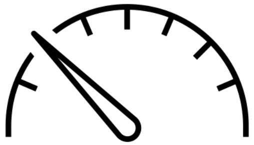
MRAM Persistent Memory

- Byte–addressable
- Maximum throughput with small I/O operations
- No need for erase before write
- No wear-leveling

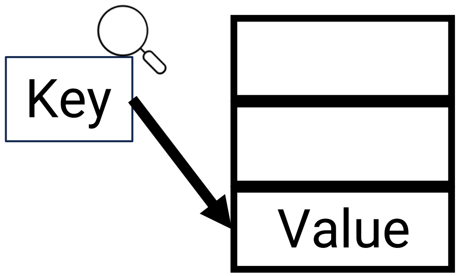
- Symmetric performance for sequential and random accesses

Where does MRAM stand out?

Experiments

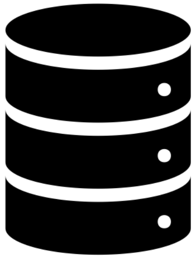


Raw performance



Key-value stores

- LPHT
- CLHT[1]
- Control: RocksDB



Relational databases

- SQLite

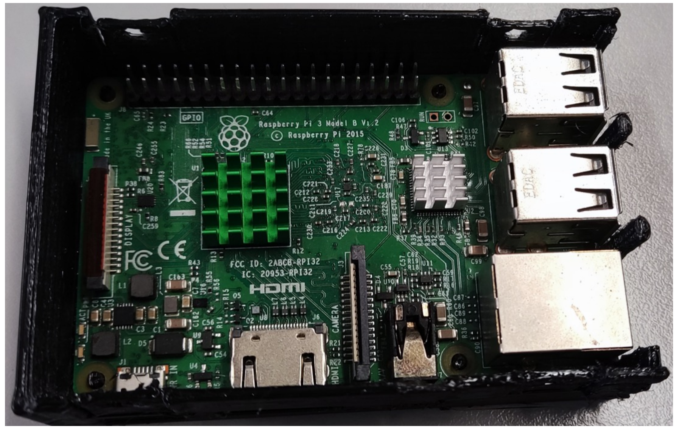
[1] T. David, R. Guerraoui, V. Trigonakis, Asynchronized concurrency: The secret to scaling concurrent search data structures

Experimental Setup

Table 2
Hardware specifications.

	STM32	RPi
CPU Model	STM32H743ZI	BCM2837
CPU frequency	480 MHz	1.2 GHz
CPU cores	1	4
RAM	1MB	1GB
Storage class	MRAM	SD Card
Storage device	Avalanche AS3004316 [24]	SanDisk Extreme [35]
Storage size	4Mb	32GB
Peak energy	66 mW	360-1440 mW[a]
Max. write cycles	10^{14}	$10^3 - 10^4$[b]
Cost (Euros)	60	50

STM32 with MRAM



Raspberry Pi 3B with SD card

Raw Performance Comparison: MRAM vs FLASH

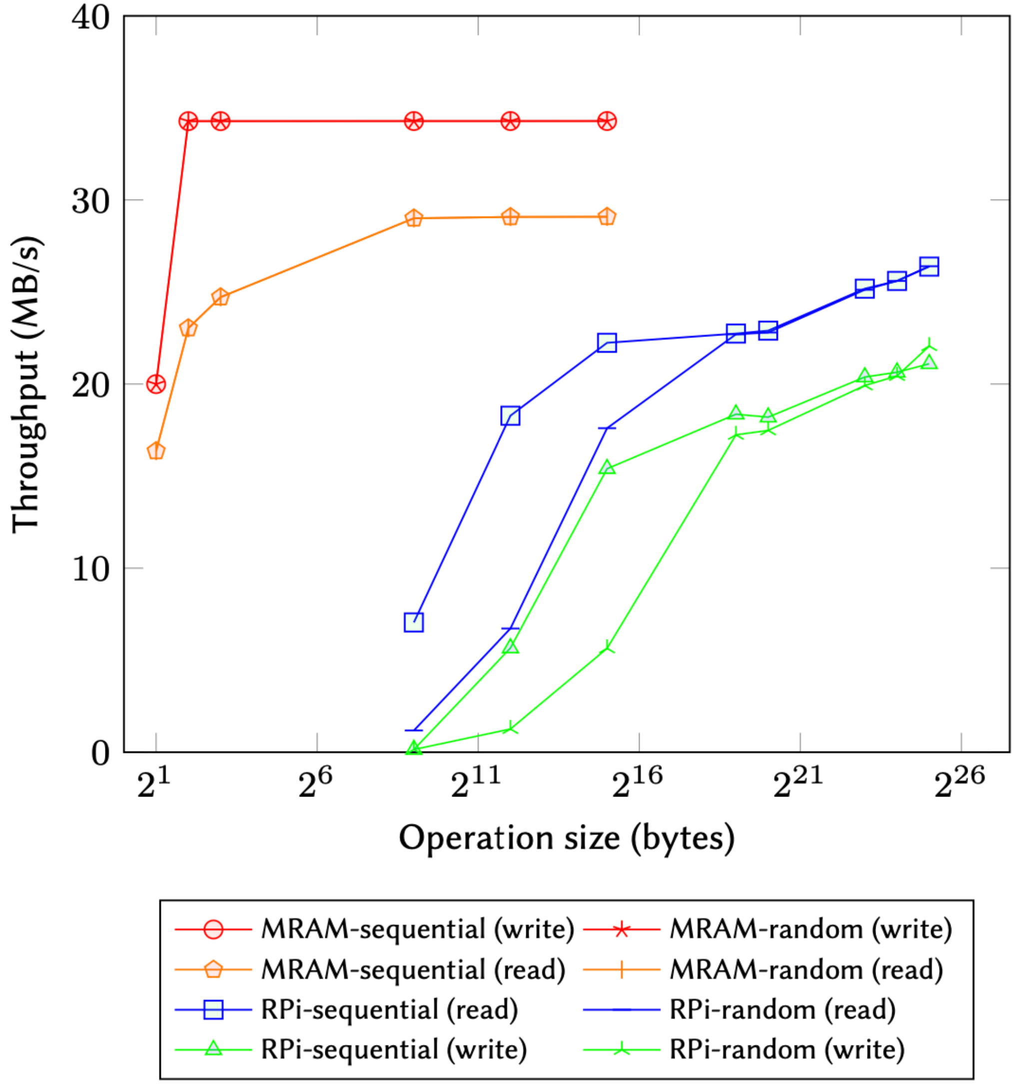
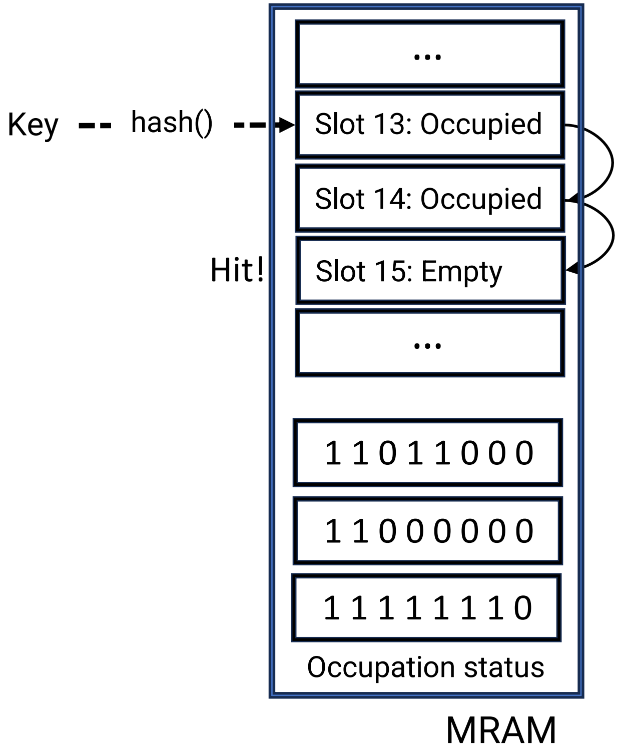


Figure 1: Storage medium read and write throughput.

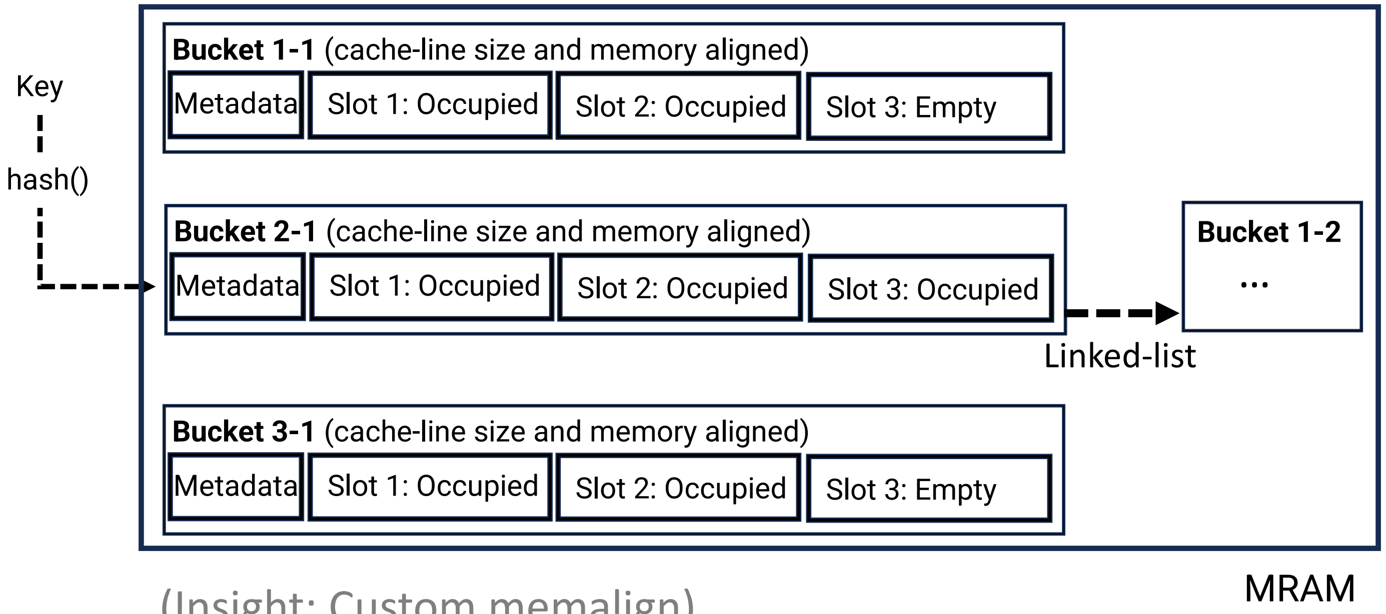
- Overall better performance
- Maximum throughput achieved at much smaller I/O operations.
- Symmetric random and sequential access performance.

Data Systems: Key-value stores

LPHT



CLHT[1]



(Insight: Custom memalign)

Code available at: https://github.com/luismeruje/Hashtables-STM32

[1] T. David, R. Guerraoui, V. Trigonakis, Asynchronized concurrency: The secret to scaling concurrent search data structures

LPHT vs RocksDB

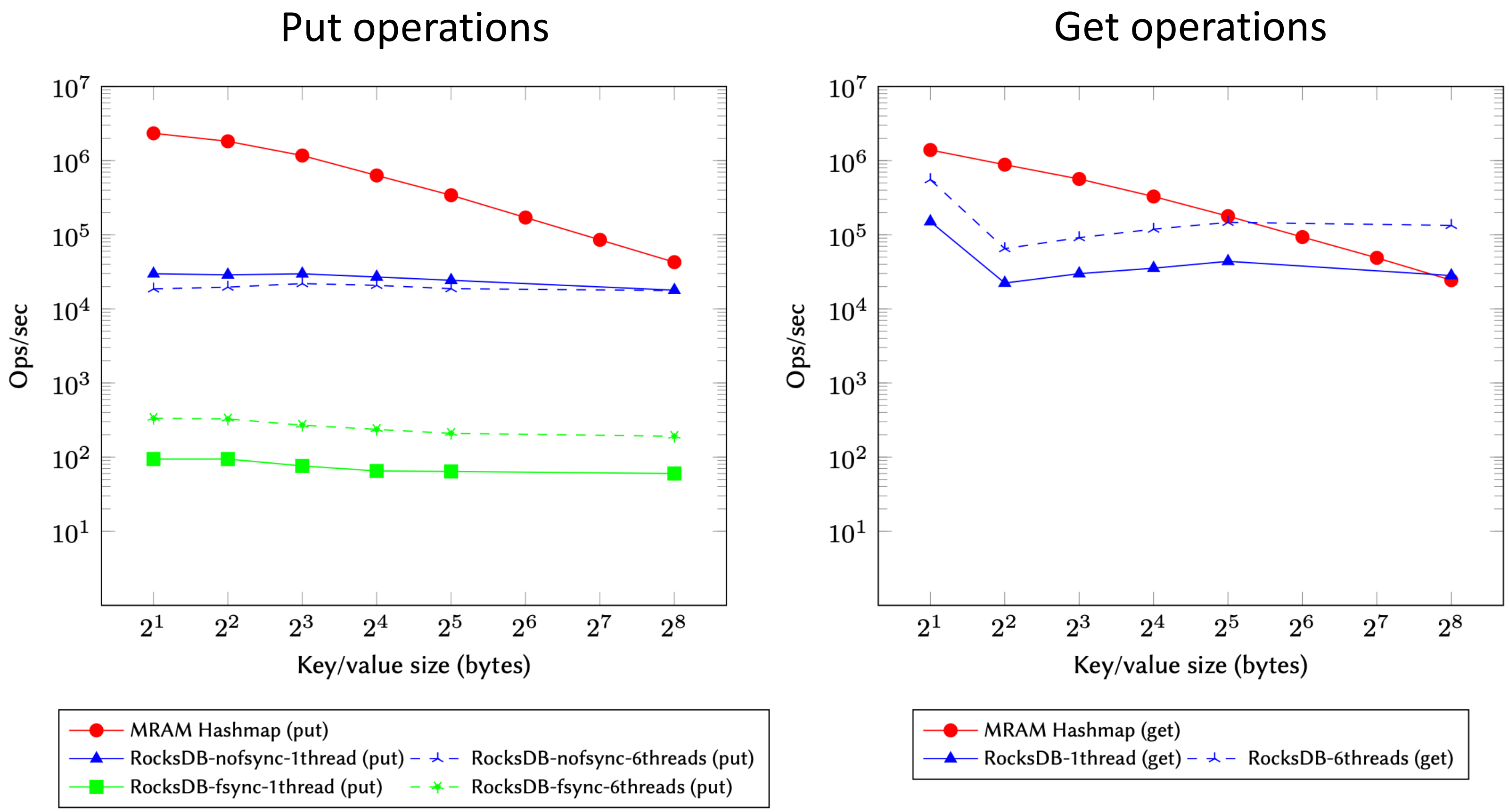


Figure 2: RocksDB (NAND FLASH) vs Linear Probing Hashmap (MRAM) with varying key/value size. Results for write and read operations show on the left side, and right side, respectively.

- Big advantage in scenarios with small key/values (most common scenario)
- 134x to 3837x better for write scenario with fsync
- 1.4x to 35x better for write scenario without fsync
- 1.64x to 6.69x better for read scenario (below 32 bytes)

CLHT vs RocksDB

- Between 11x and 1827x more put operations per second
- Around 9x more get operations per second

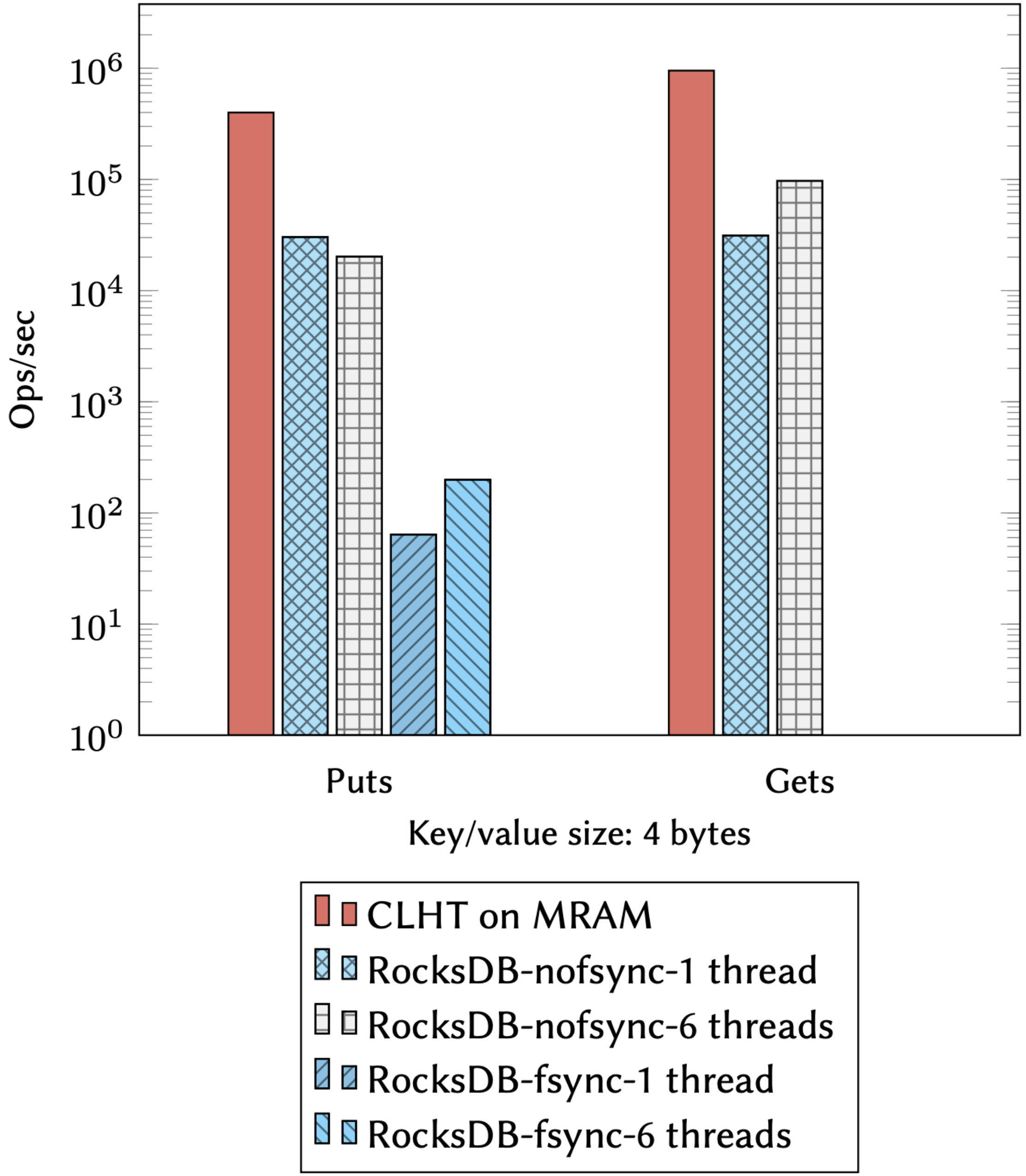
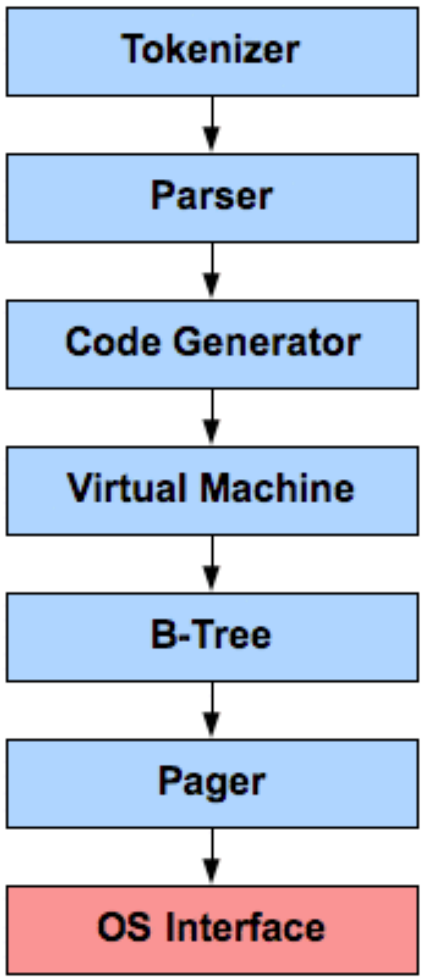
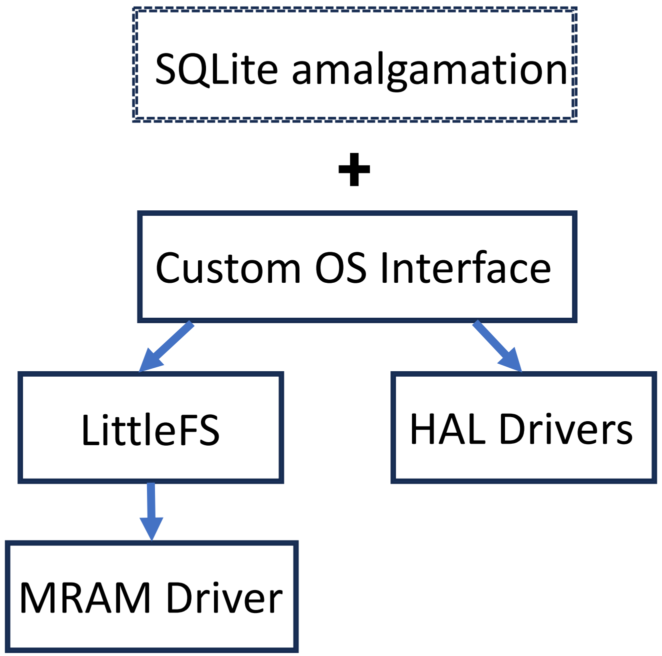


Figure 3: Comparison between CLHT running on MRAM and RocksDB running on NAND FLASH in RPi.

Data Systems: SQLite



Source: https://www.sqlite.org/vfs.html



Code available at: https://github.com/luismeruje/SQLite-STM32

Data Systems: SQLite

STM32 with MRAM loses on almost all scenarios.
Only outperforms RPi with FLASH in insert scenario with 2 rows/transaction.

Increased storage performance does not compensate lower computation capabilities.

Possible improvements:

- Optimized file system (remove overhead)
- Use hashing controllers, DMAs, and others to take load off the CPU.
- MRAM memory with better performance.

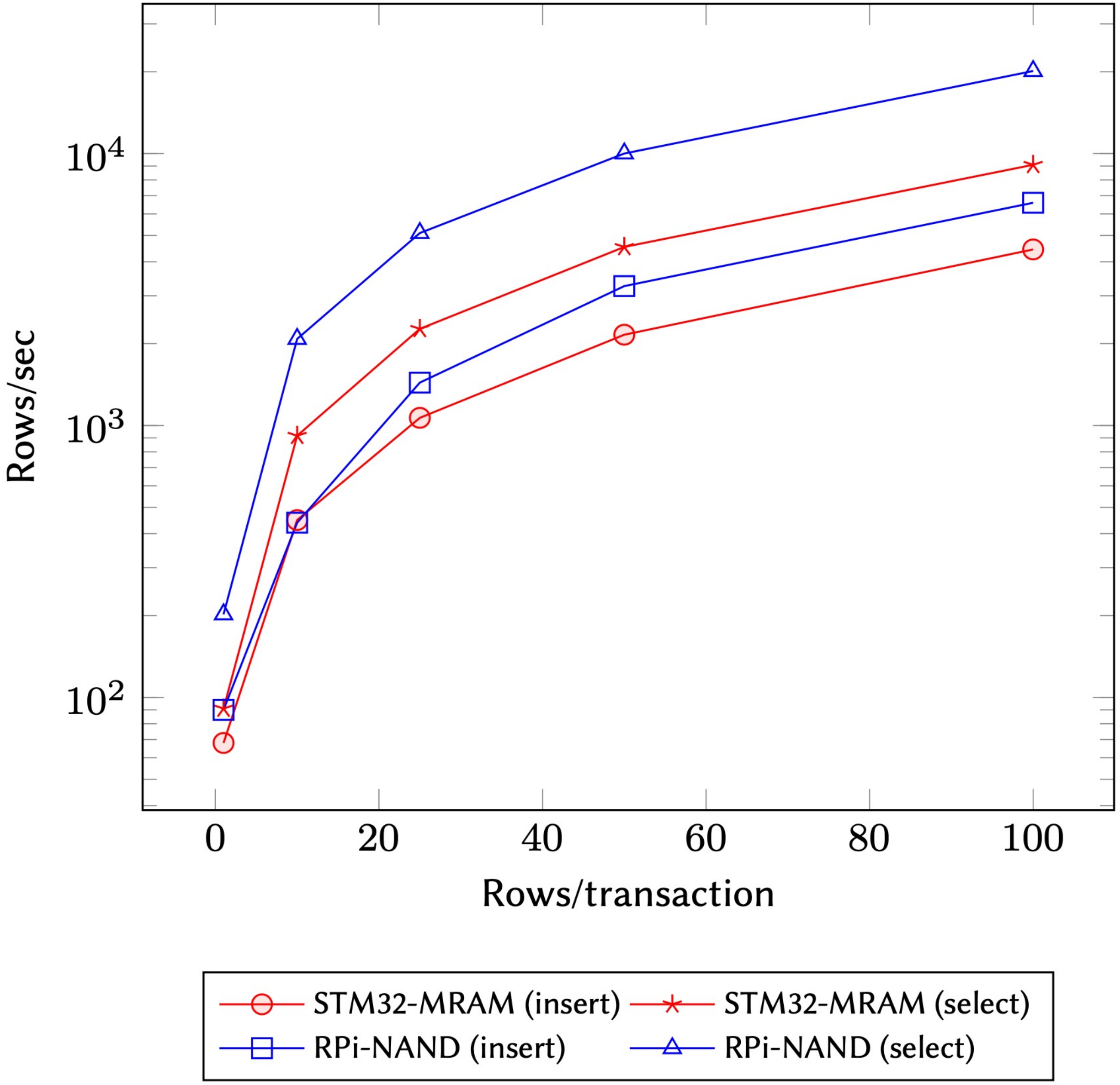


Figure 4: Comparison between SQLite running on: STM32's MRAM and RPI's NAND FLASH.

Discussion

- Big advantage in small I/O operations.
- Better endurance and performance.
- Less energy consumption.
- Less computational overhead.
- Possibly replace MPUs with MCUs.

Discussion

- Big advantage in small I/O operations.
- Better endurance and performance.
- Less energy consumption.
- Less computational overhead.
- Possibly replace MPUs with MCUs.
- Downsides: low capacity, high cost.

Discussion

- Big advantage in small I/O operations.
- Better endurance and performance.
- Less energy consumption.
- Less computational overhead.
- Possibly replace MPUs with MCUs.
- Downsides: low capacity, high cost.

- Hybrid approach could be the best solution for this moment.
- Barely grasping at the capabilities of MRAM, M3 and M4 would likely give much better results.

Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23)
Workshop on Accelerating Analytics and Data Management Systems (ADMS'23)

Towards MRAM Byte-Addressable Persistent Memory in Edge Database Systems

Luís Ferreira, Fábio Coelho, José Pereira
INESC TEC & University of Minho

August 28, 2023

Code repositories:
https://github.com/luismeruje/SQLite-STM32 (SQLite adaptation)
https://github.com/luismeruje/Hashtables-STM32 (LPHT and CLHT adaptations)

Universidade do Minho

FCT
Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

INESCTEC