Extending the Lifetime of NVM: Challenges and Opportunities

Saeed Kargar, Faisal Nawab*
University of California Santa Cruz, University of California Irvine*
VLDB 2021
August 19, 2021

Baskin Engineering UC SANTA CRUZ

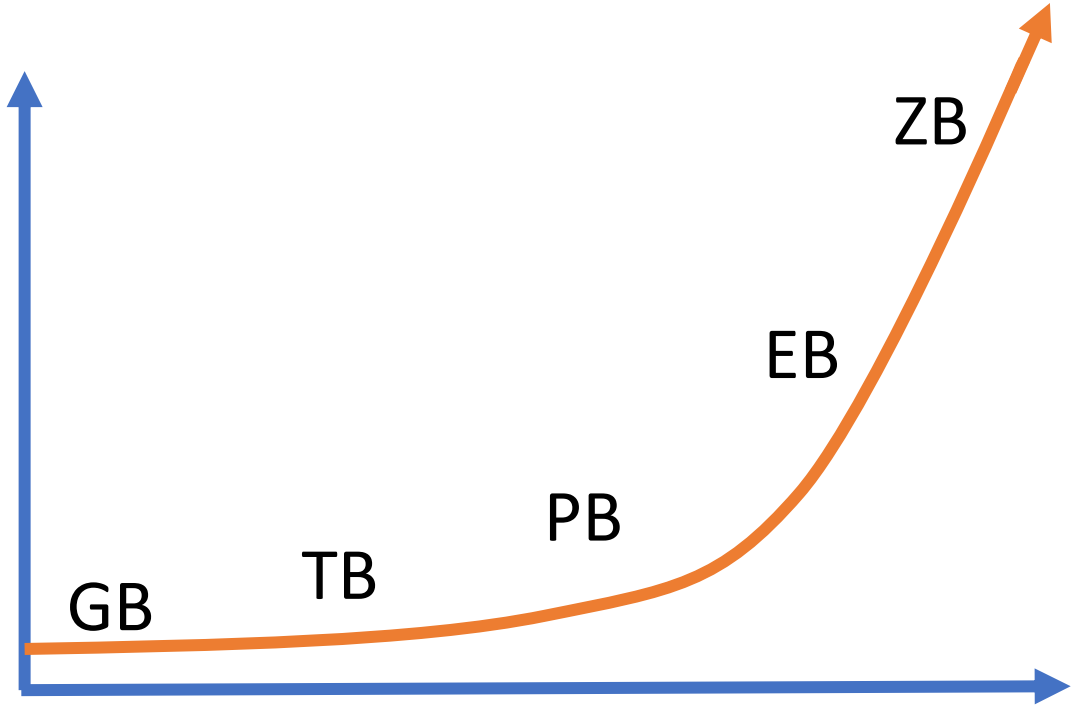
EDGE LAB

Outline

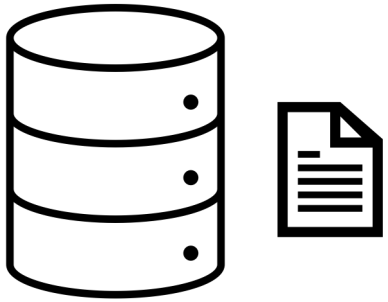
- Background
- Local write optimizations (reducing bit flips)
- Reducing Write amplification
- Memory-awareness
- Conclusion

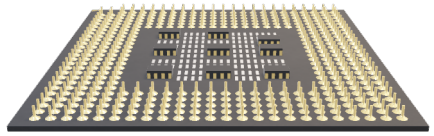
Background

- data is growing exponentially
- doubling every two years
- We need massive storage



- Massive storage is crucial for computing because computing is done on data that is stored on storage
- To have a faster computing, we need to get as much of the data as close to CPU as possibly can, so the CPU waits less for the data





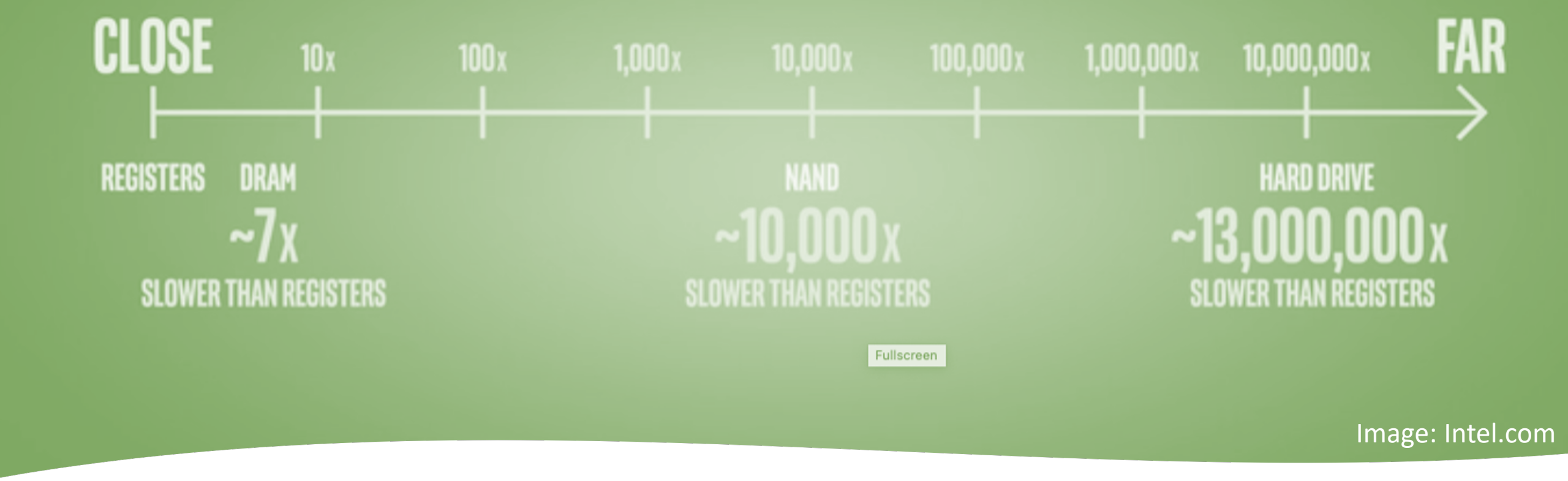
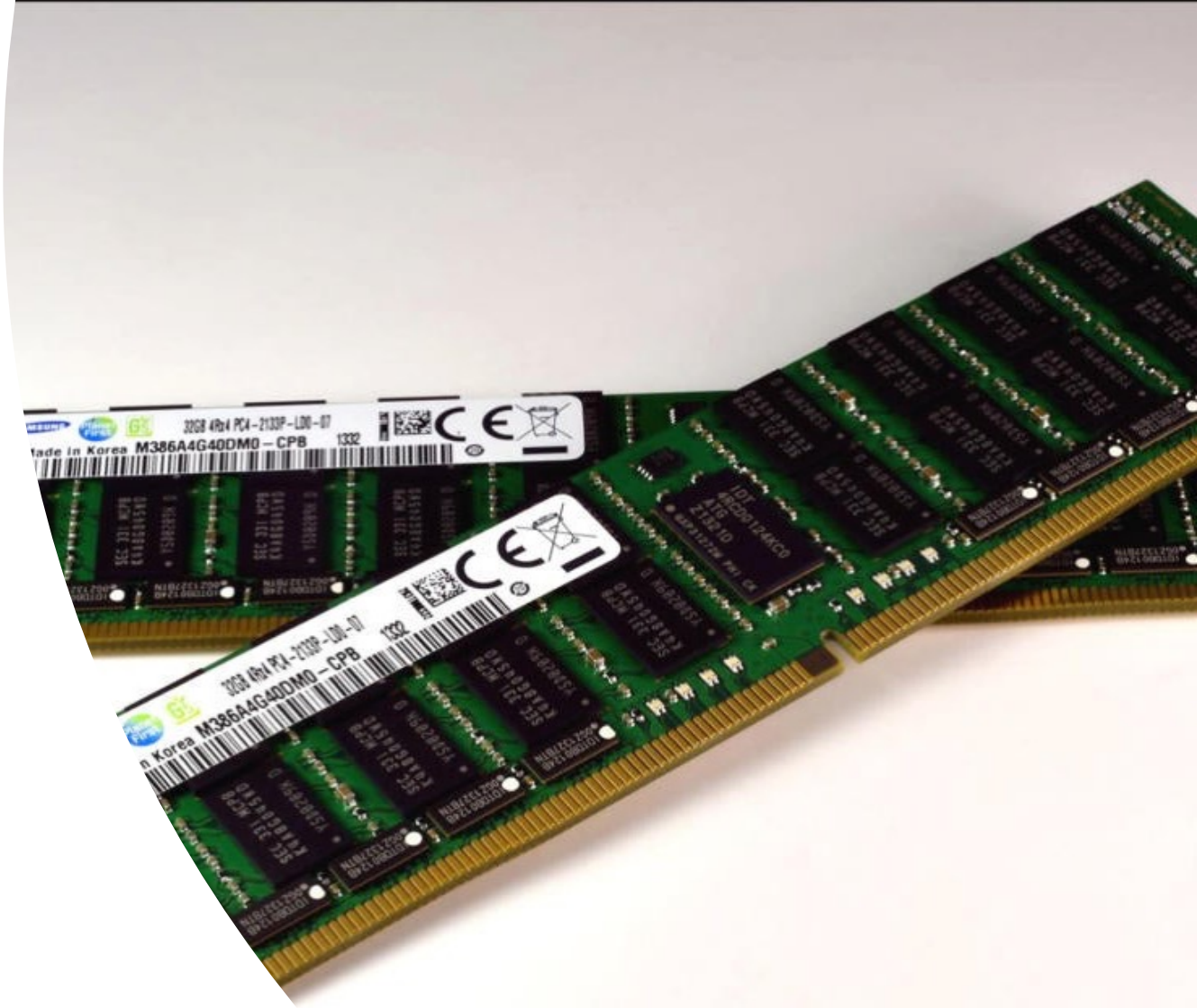


Image: Intel.com

- However, the fastest storage devices (NAND SSDs) are tens of thousands of times slower than the fastest storage memory, which results is performance degradation
- We need faster storage

DRAM

- 1966
- Read/Write Latency: ~10 – 15 ns
- Endurance (Writes/bit): ~10^{18}

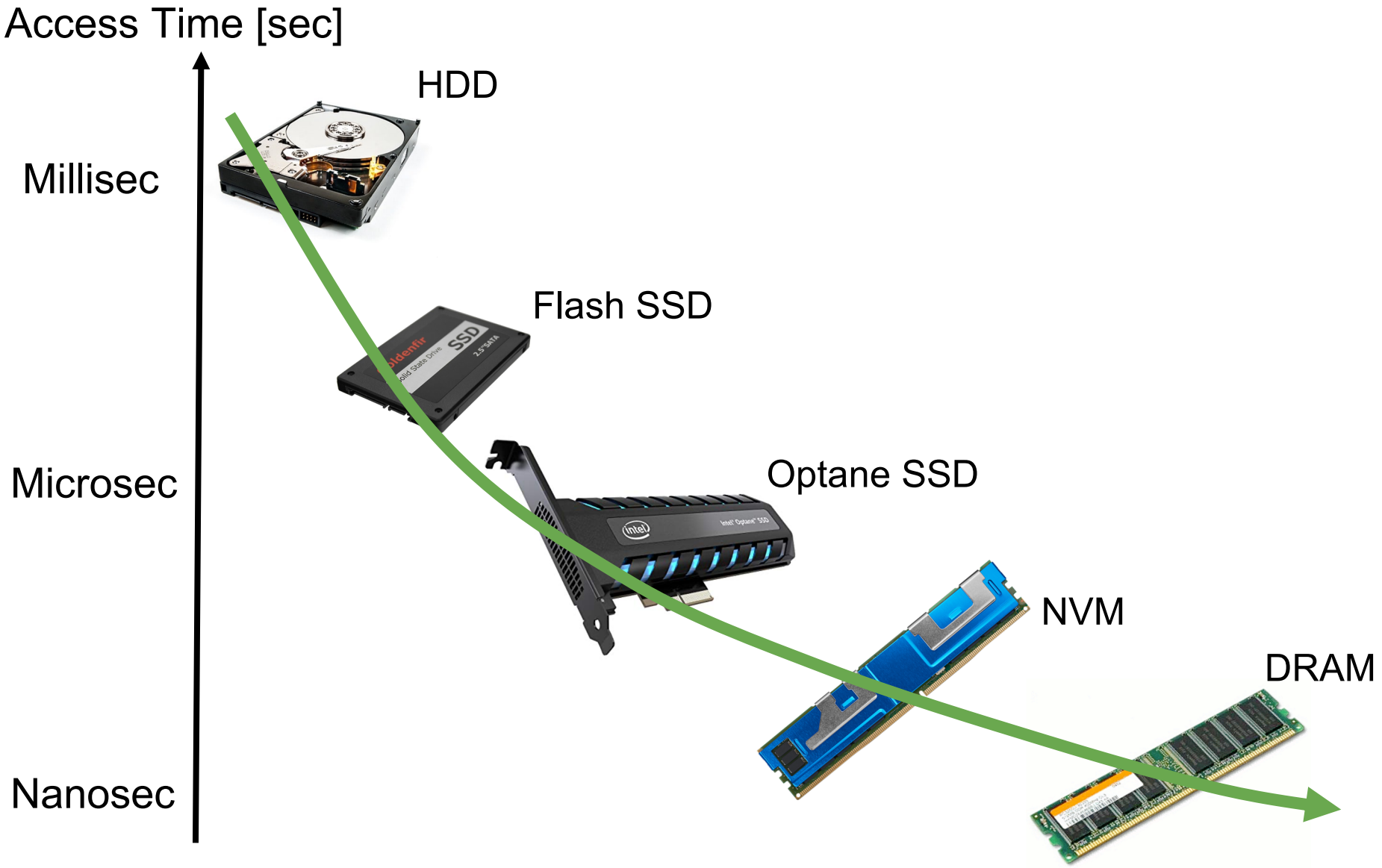


However,

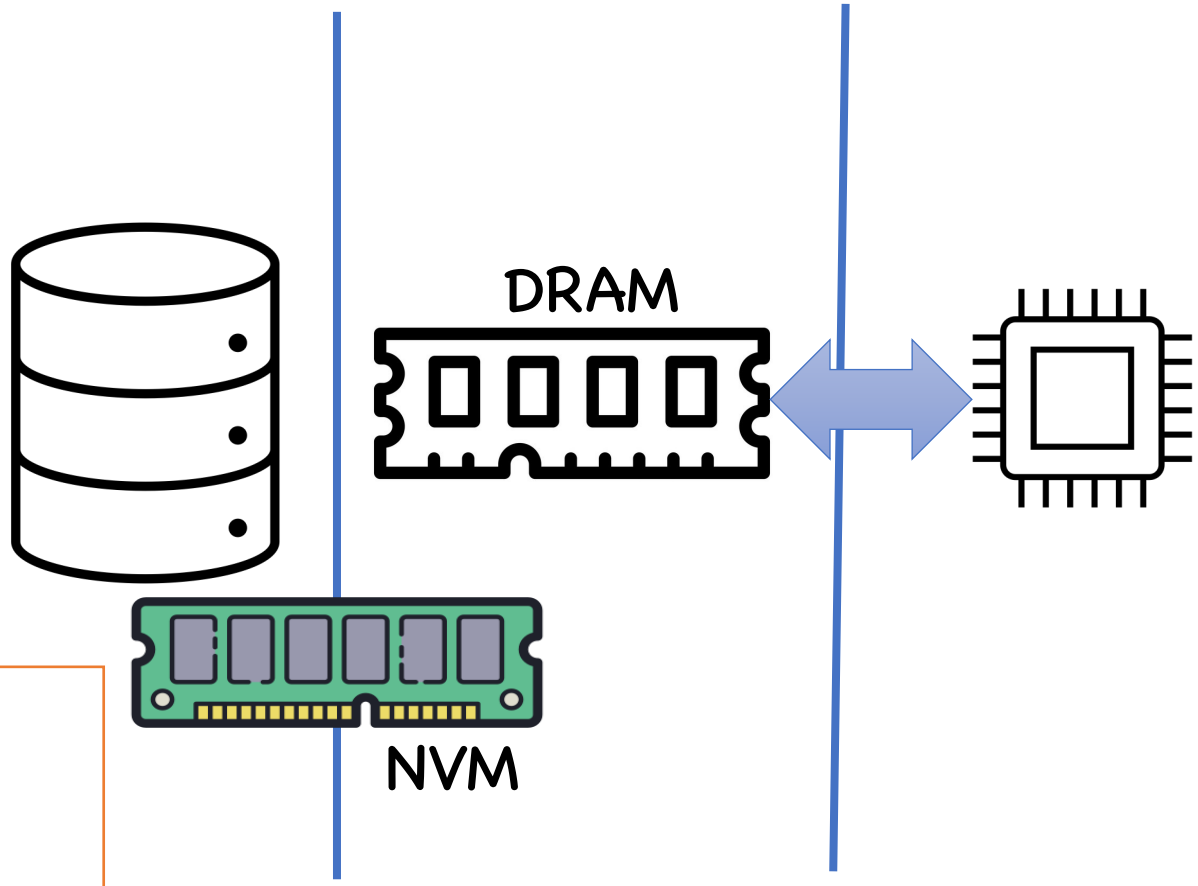
- Expensive (cost per GB basis)
- Volatile
- Leakage power dissipation
- Relatively small capacities
- …..

Non-volatile Memory (NVM)

Moving Towards Non-Volatile Memory (NVM) Era



NVM



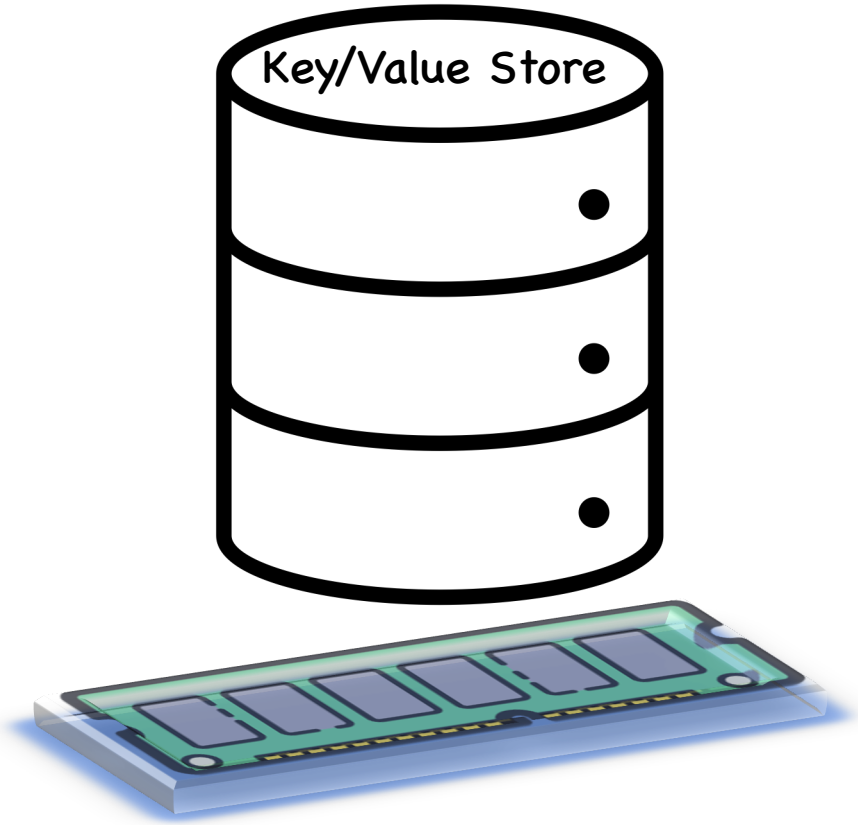
Higher capacity

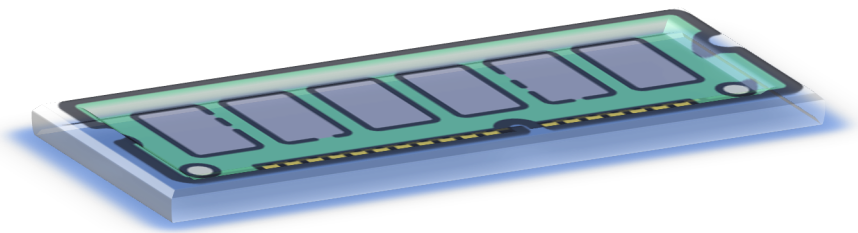
Cheaper

Faster

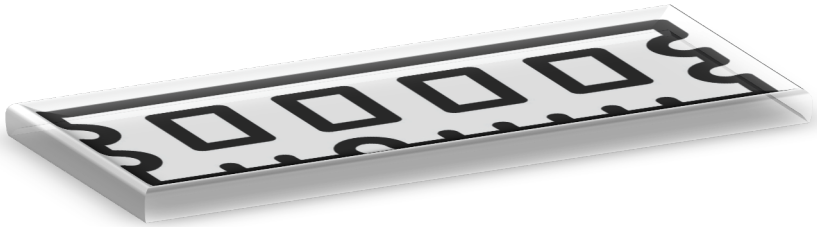
Non-volatile

- These non-volatile memories give us new opportunities.
  - they provide a unique combination of affordable larger capacity and support for data persistence.
  - They give us access to more data in memory enabling you to process large amounts of data faster.


Key/Value Store
NVM



≠



NVM

DRAM

Limitations

Write endurance

PCM: 10^8 - 10^9

DRAM: > 10^{17}

the number of writes that can be applied to a segment of storage media before it becomes unreliable

Write latency

PCM: **~ 80 - 500 ns**

DRAM: **~10 ns**

Asymmetric

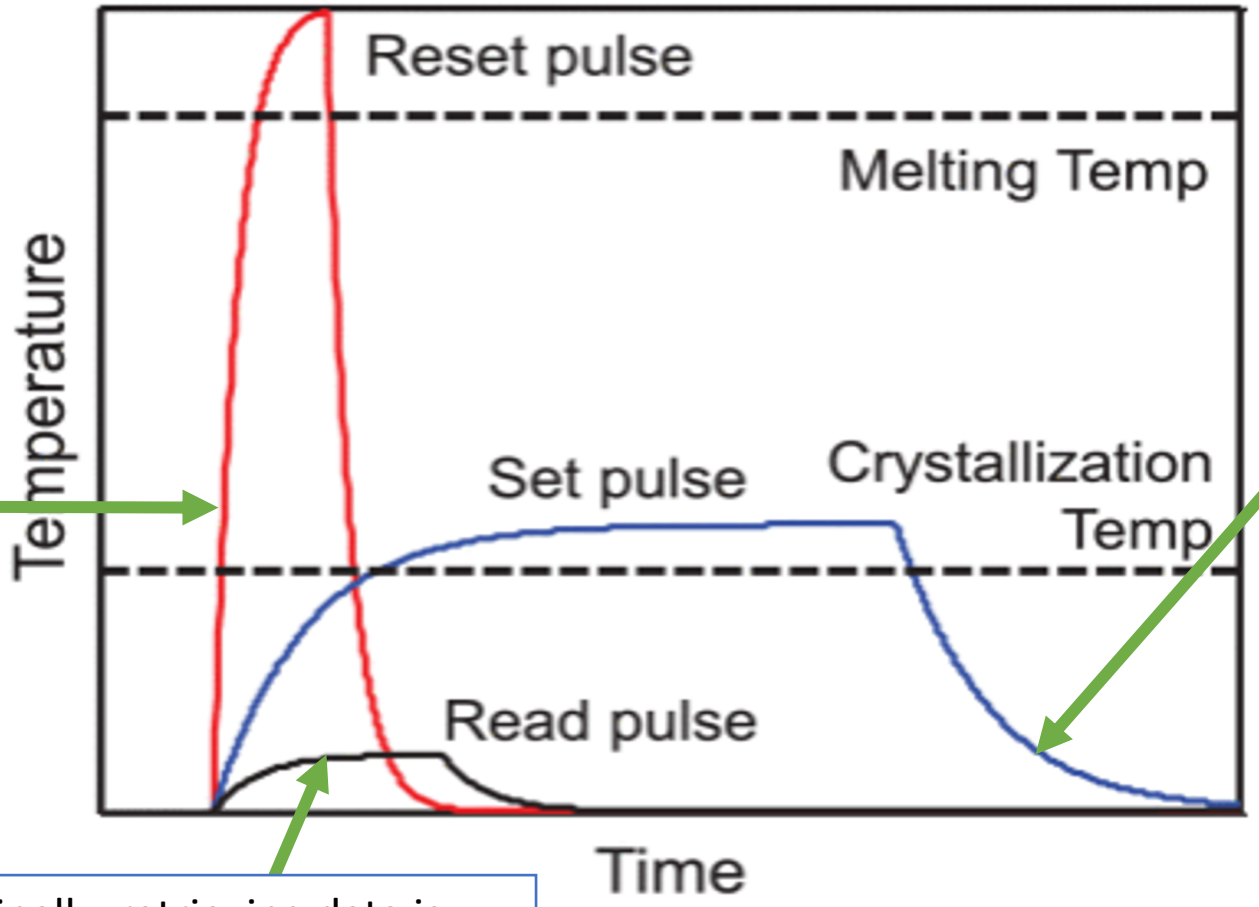
(read vs. write)

80 - 500 ns write latency VS. **20 - 50 ns** read latency

write consumes significantly higher energy than read

- For PCM, to write an individual bit (~20/100 pJ/b) compared to writing a DRAM page (~0.02 pJ/b),
- power consumption is 15.7 to 22.5x more than that for reading a bit

Temperature/Current and Time required to program or read PCM cell

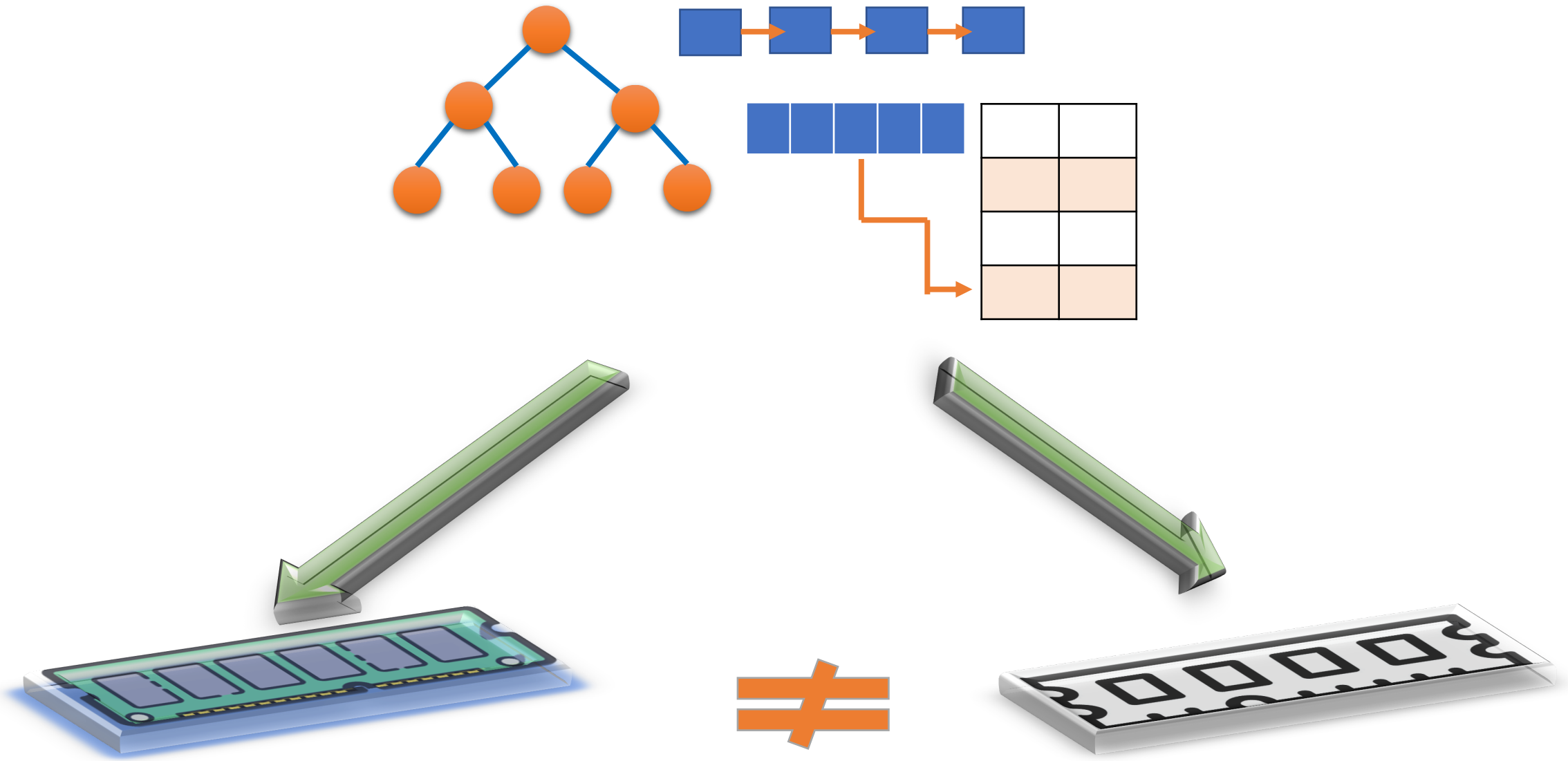


High-power pulses for the RESET operation that places the memory cell into the high-resistance RESET state

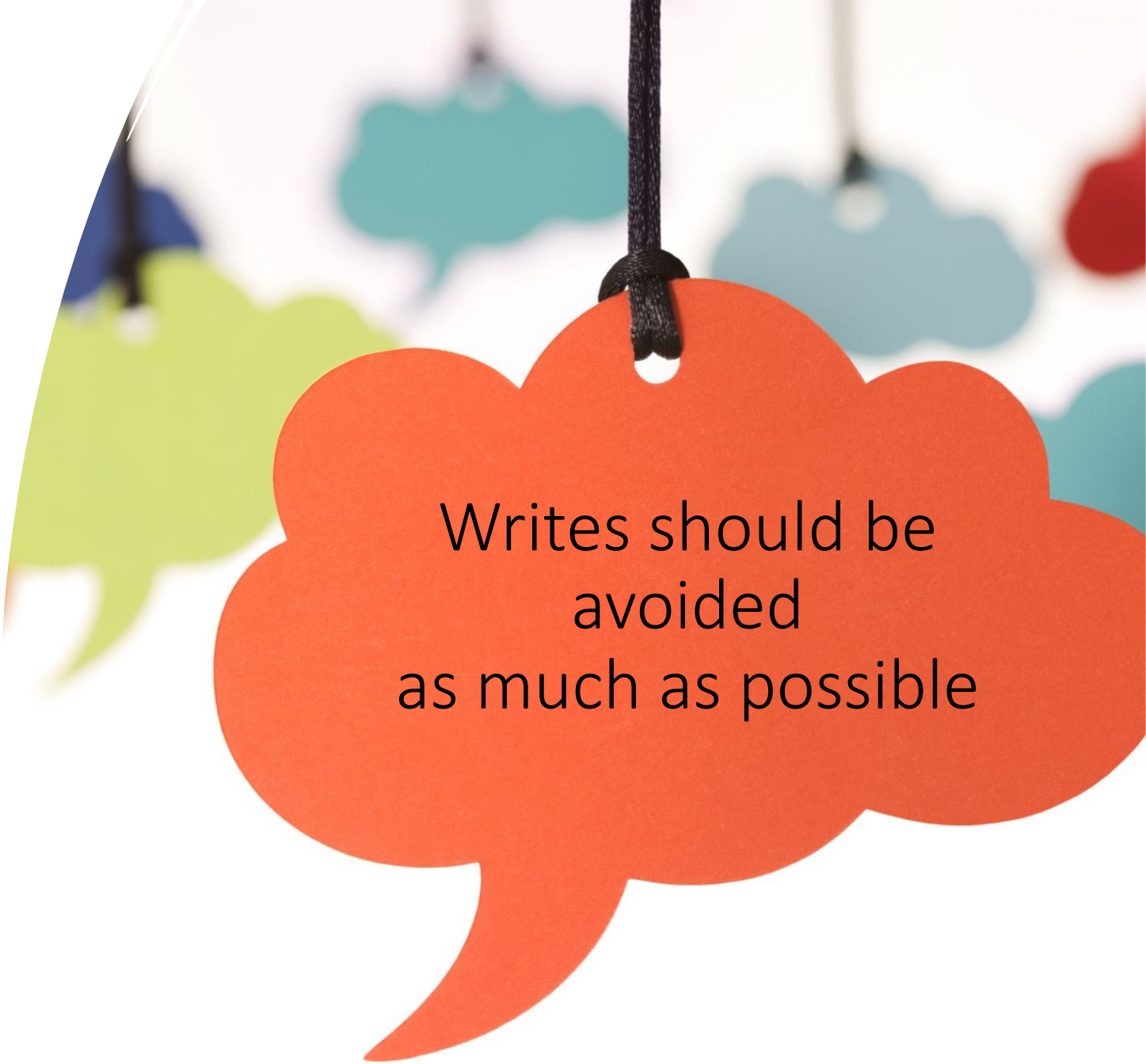
moderate power but longer duration pulses for the SET operation returning the cell to the low-resistance SET state.

Finally, retrieving data is done with very low power by sensing the device resistivity.

Image by: Fong, Scott W, et al; 2017


≠
NVM
DRAM

In the NVM-based systems and data structures



How to mitigate the NVM's write limitation

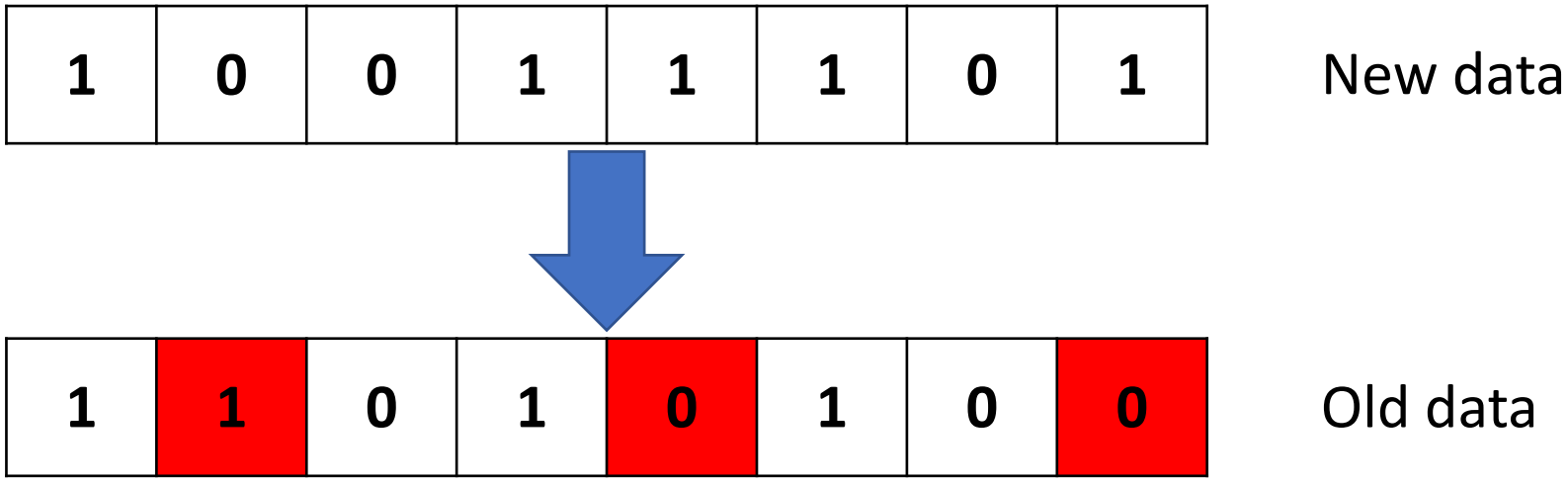
- **Local write optimizations** [Bittman+, FAST'2019; Cho+, MICRO'2009; Dgien+, NANOARCH'2014; Dong+, HPCC'2015; Jalili+, DATE'2016; Palangappa+, GLSVLSI'2015]

- **Reducing write amplification** [Shimin+, VLDB'2015; Dai+, OSDI'2020; Huang+, DATE'2020; Kannan+, ATC'2018; Lu+, TOC'2017; Ma+, FGCS'2020; Xia+, ACT'2017; Zuo+, MSST'2017; Oukid+, SIGMOD'2016]

- **Memory-awareness** [Kargar, S; Nawab, F, ICDE'2021]

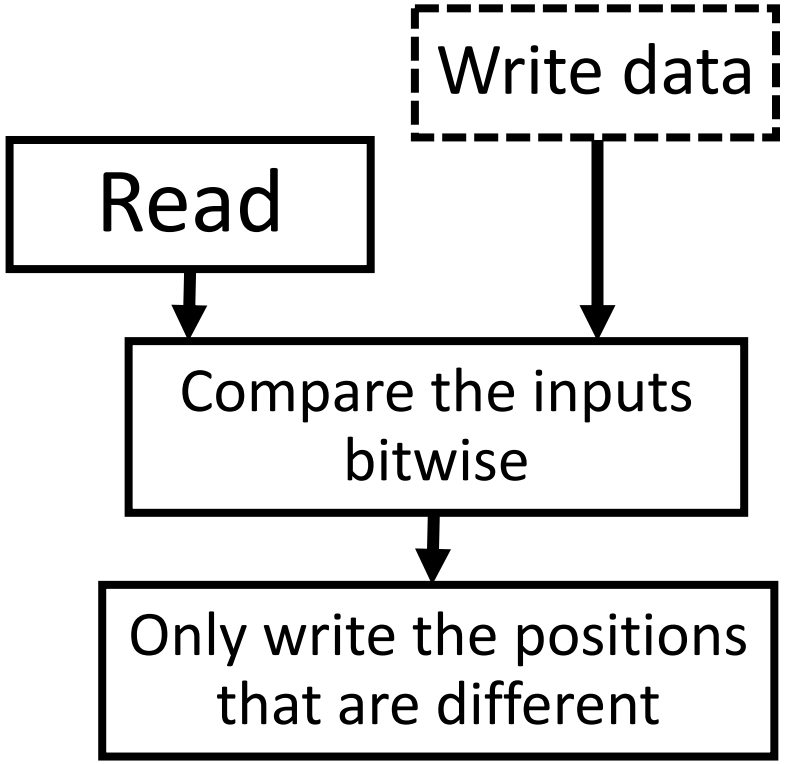
Local write optimizations

- This group of methods focuses on reducing the number of bit flips through the Read-Before-Write (RBW) technique.


1 0 0 1 1 1 0 1
New data
1 1 0 1 0 1 0 0
Old data

Read-Before-Write

- This technique replaces each NVM write operation with a more efficient read-modify-write operation



Data-Comparison Write (DCW) [Yang, Byung-Do, et al; 2007]

- The DCW scheme performs the read operation before the write operation to know the previously stored data in the selected cell.
- If the input data and the previously stored data are the same, no write operation performs.
- If not, the write operation is the same as the conventional write scheme.

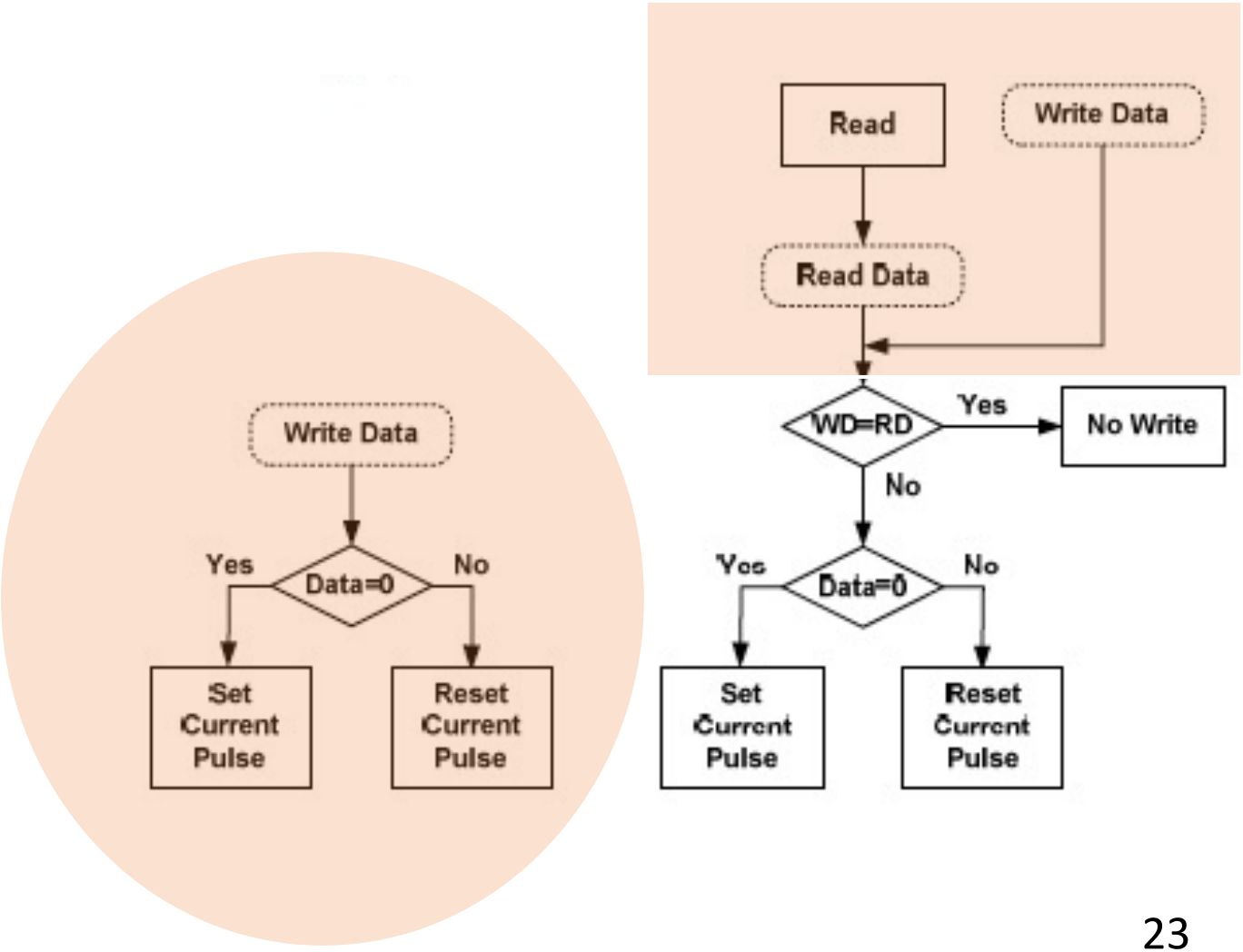
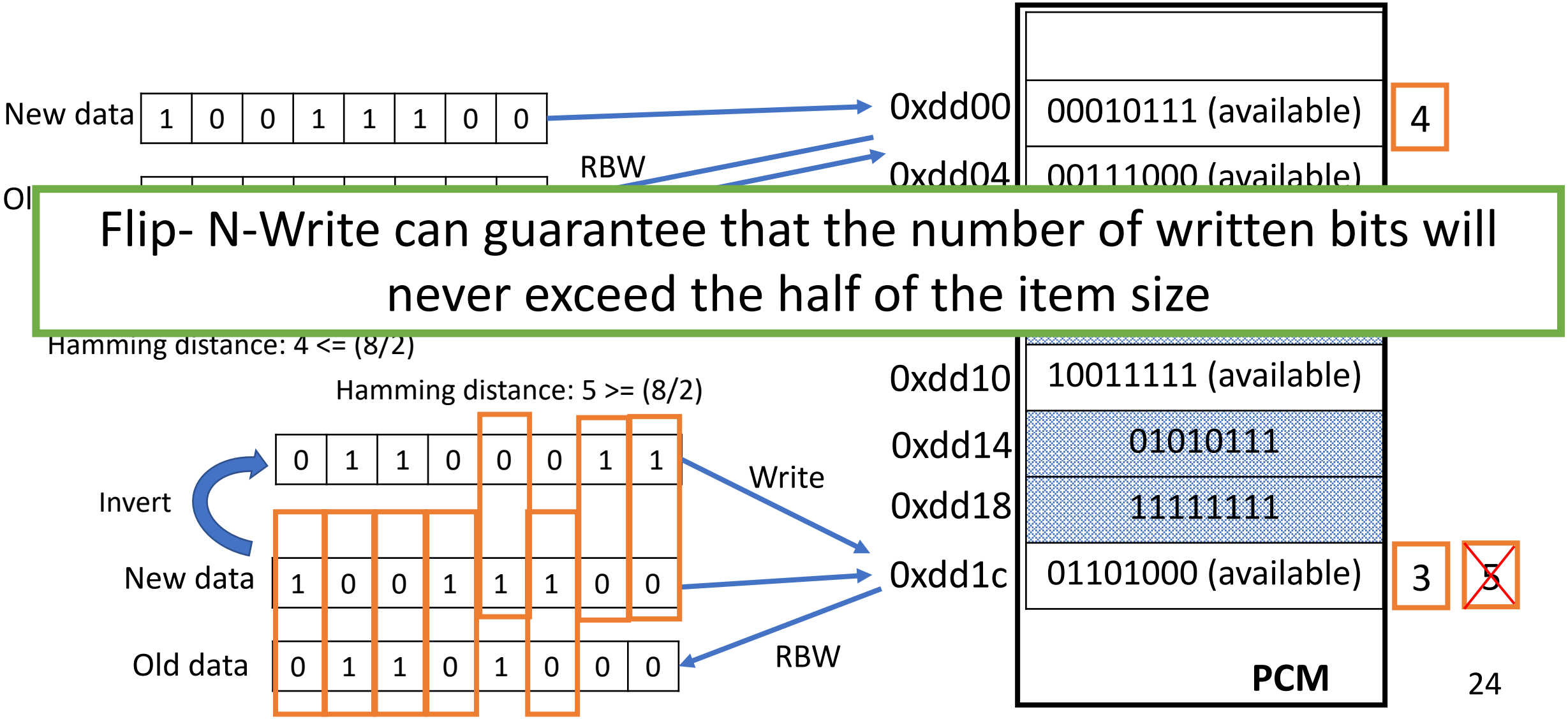


Image by: Yang, Byung-Do, et al; 2007

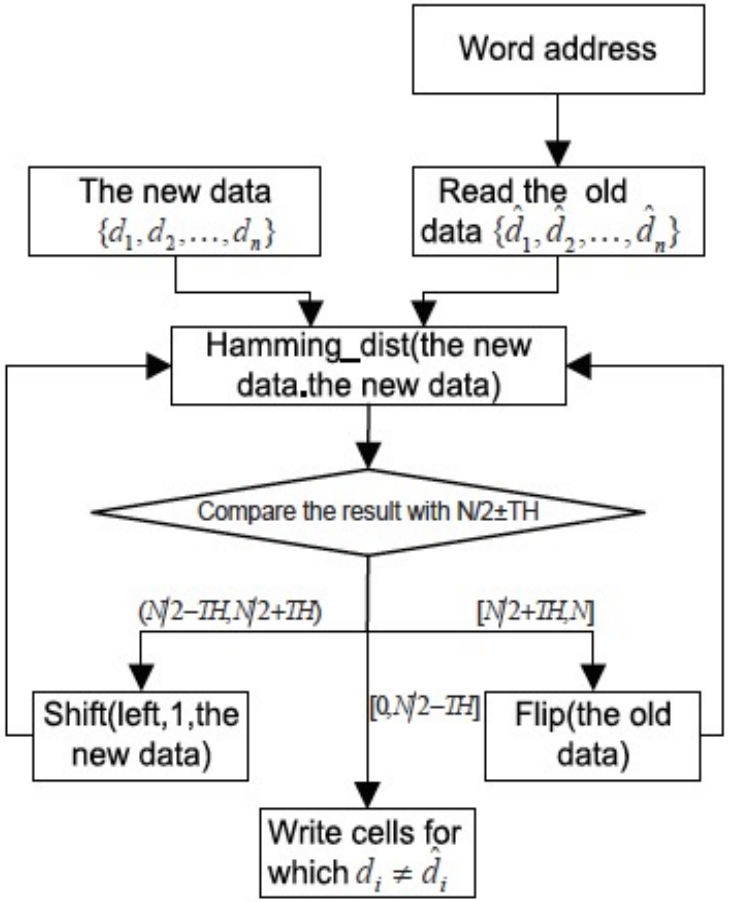
Flip-N-Write [Cho, Sangyeun, and Hyunjin Lee; 2009]



Flip- N-Write can guarantee that the number of written bits will never exceed the half of the item size

Min-Shift

[Luo, Xianlu, et al; 2017]



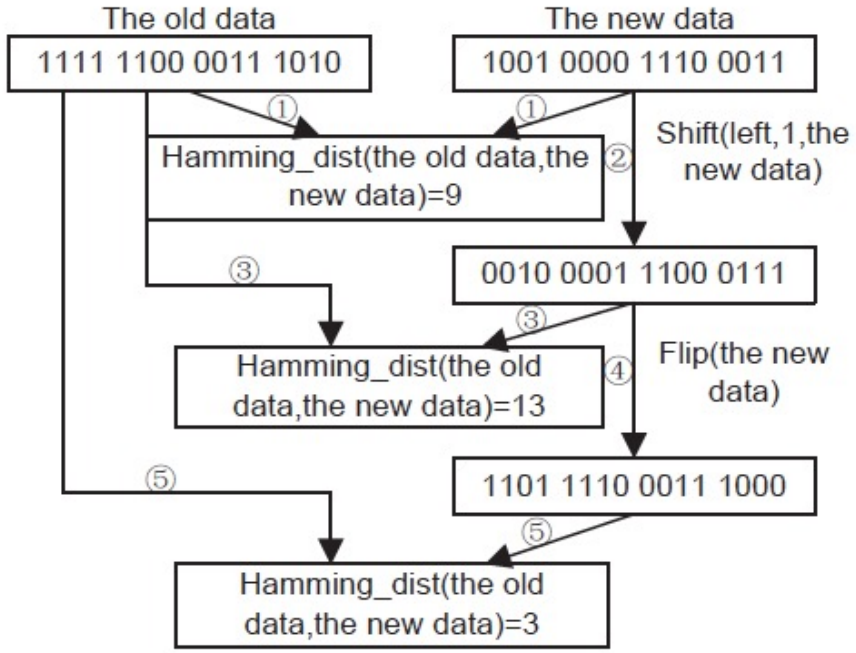
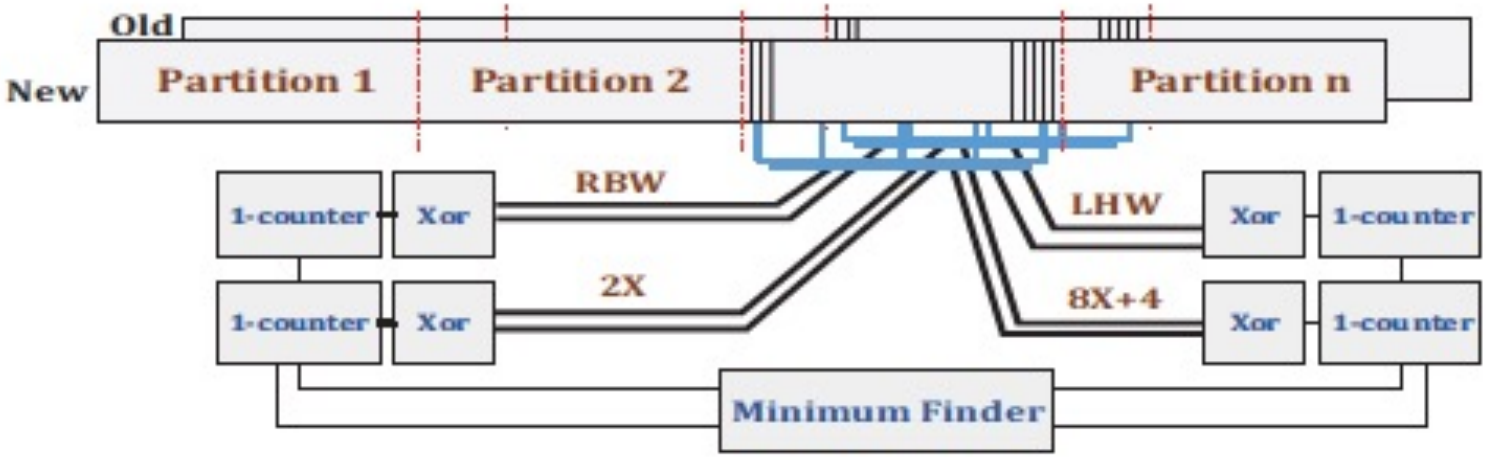


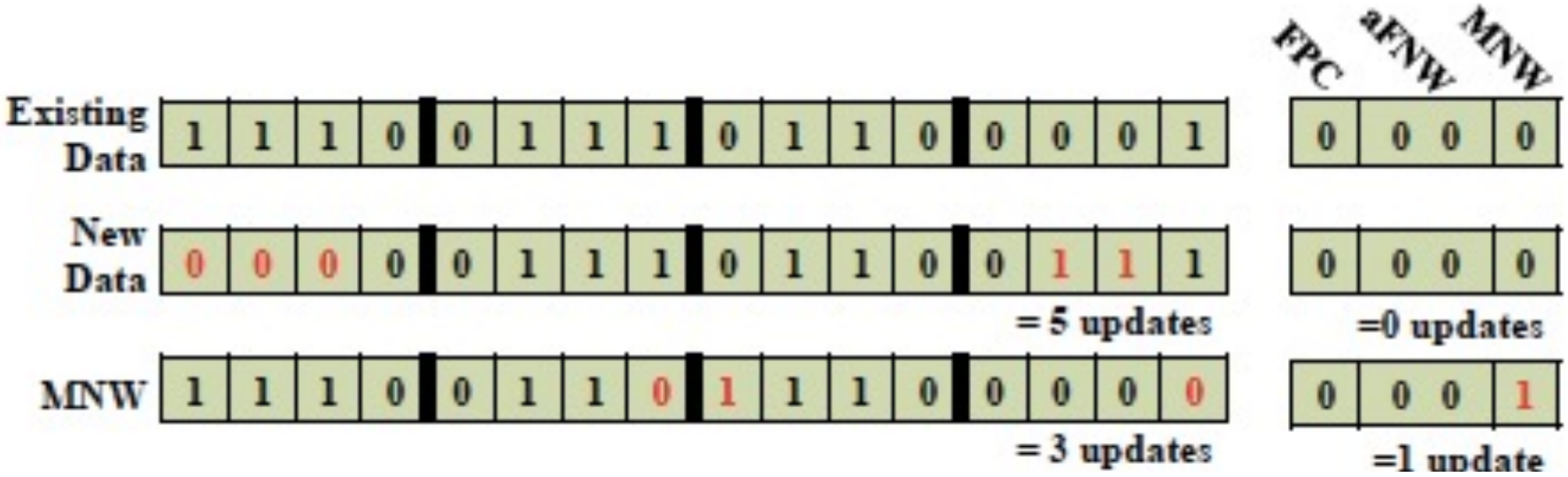
Image by: Luo, Xianlu, et al; 2017



- The block is partitioned into some parts (words) and then by considering the possible locations for hot bits and inverted form of corresponding cells, the number of bit flips using each pattern is counted to find the minimum number of bit flips.

Image by: Jalili, Majid, and Hamid Sarbazi-Azad; 2016

Captopril [Jalili, Majid, and Hamid Sarbazi-Azad; 2016]



Flip-Mirror-Rotate [Palangappa, Poovaiah M, et al; 2015]

FMR comprises three components:
adaptive Flip-N-Write (aFNW), Mirror-N-Write (MNW), and Rotate-N-Write (RNW)

Image by: Palangappa, Poovaiah M, et al; 2015

MinFS

[Luo, Xianlu, et al; 2014]

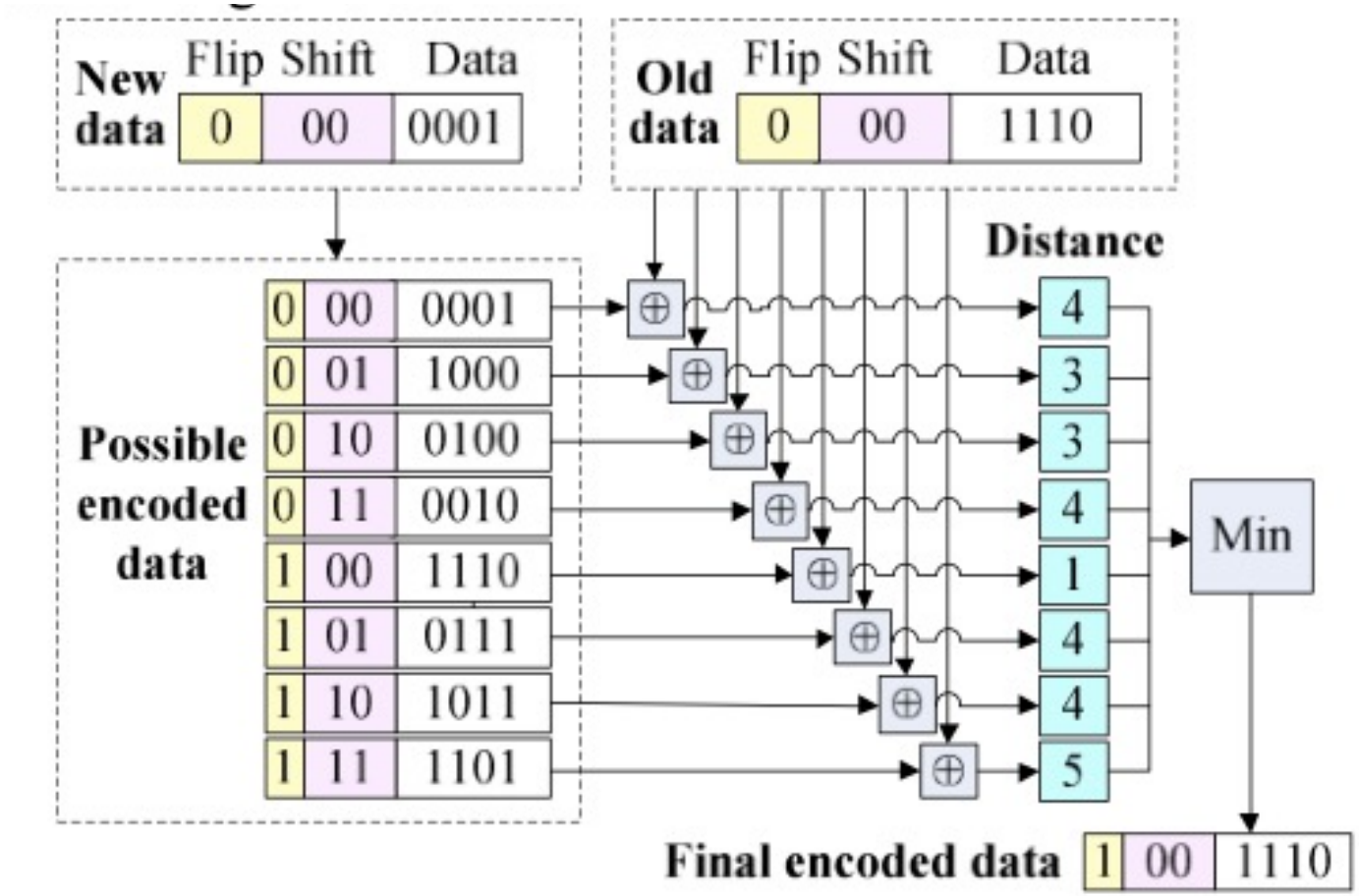
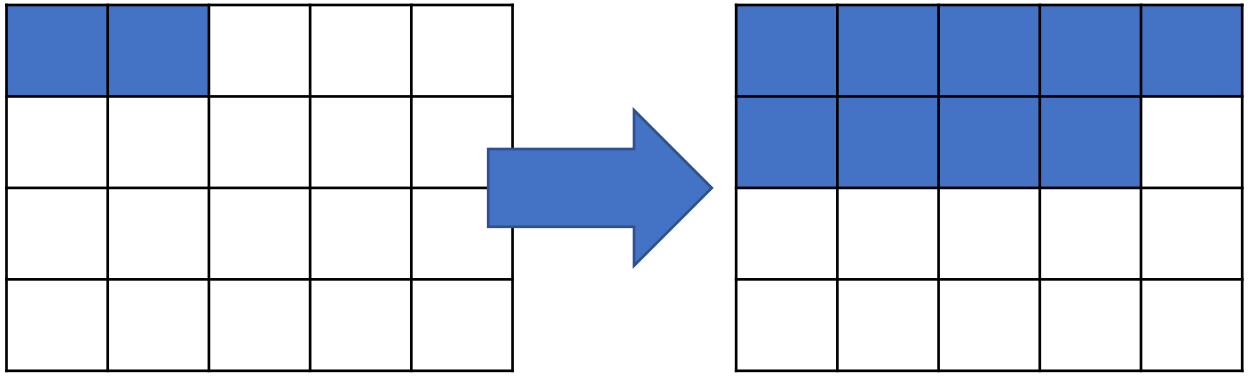


Image by: Luo, Xianlu, et al; 2014

Reducing write amplification

Write amplification

- Write amplification is an undesirable phenomenon associated with flash memory and solid-state drives where the actual amount of information physically written to the storage media is a multiple of the logical amount intended to be written.



Example: B-Tree

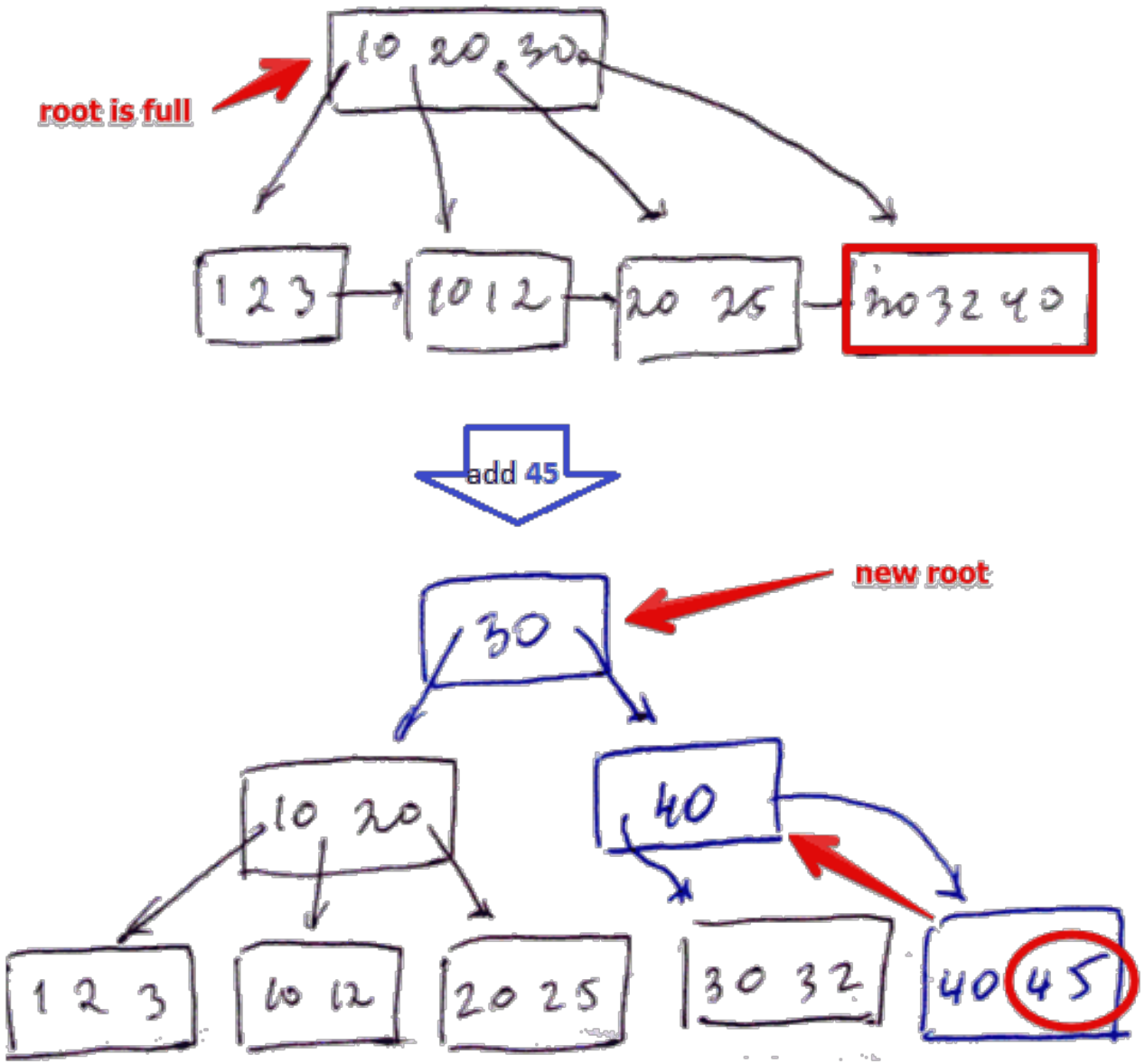
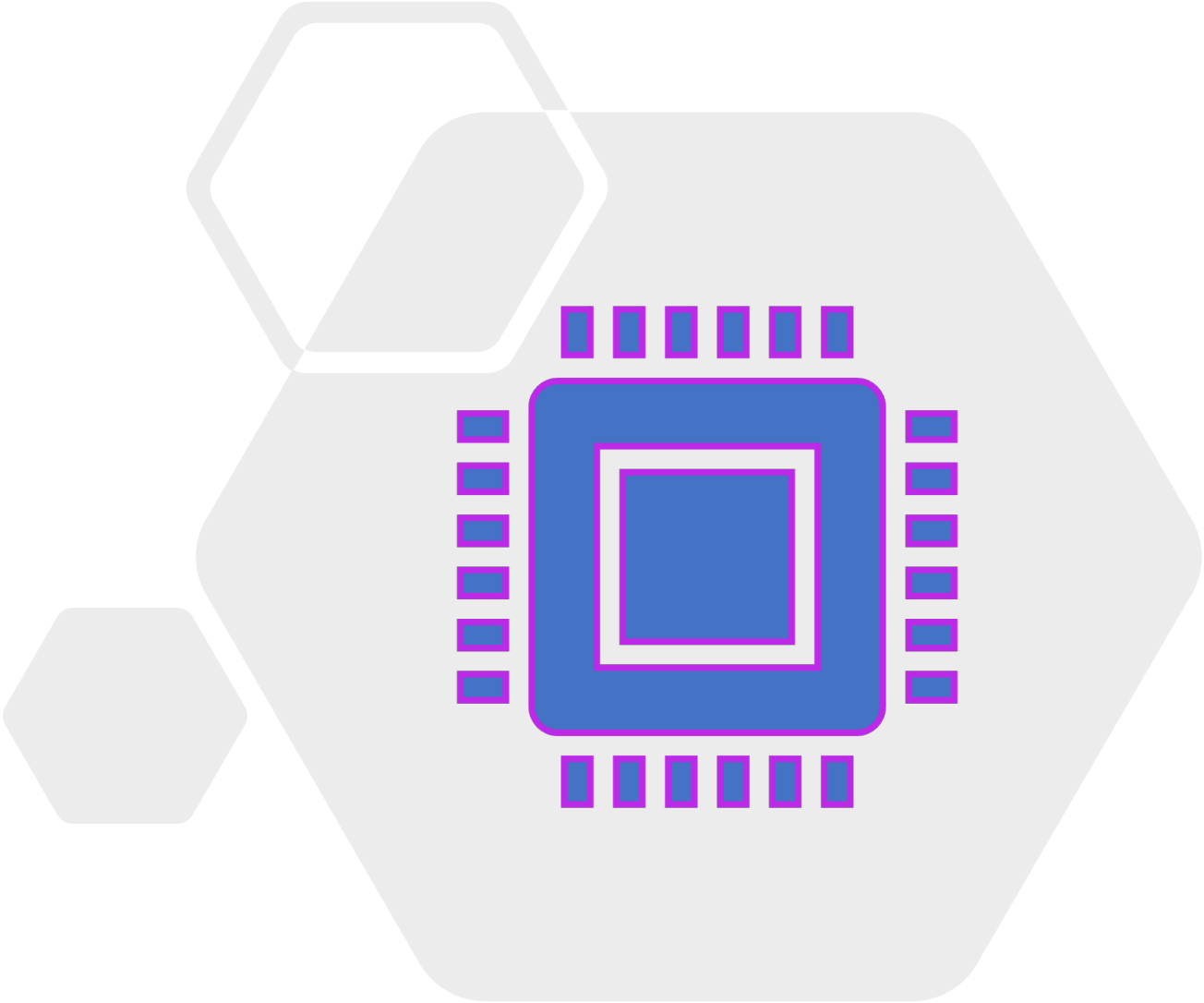


Image by: mlwiki.org/index.php/B-Tree

- Reducing write amplification can have the positive side-effect of increasing NVM write endurance since less data is written.
- However, this is not an easy task to do due to the fact that all the existing data structures and database systems have been designed for DRAMs and HDDs, where the challenges of the lifespan of memory segments and the energy consumption of writes are not as significant in DRAM/HDD as they are in NVM/SSDs.

Examples

- SSD optimized LSMs
  - ✓ WiscKey (FAST ‘16), VT-tree (FAST ‘13)
- NVM optimized B+Tree
  - ✓ NV-Tree (FAST ‘15), FP-Tree (ICMD ‘16)
- PCM optimized Hashing data structure
  - ✓ LibreKV (IEEE-TETC ‘17), Hi-KV (ATC ‘17), Path-Hashing (TPDS’18)



Path-hashing [Zuo, Pengfei, and Yu Hua; 2017]

- Path hashing leverages position sharing to allocate several standby cells for each addressable cell in the hash table to deal with hash collisions.
- The addressable cells in the hash table are addressable by the hash functions and the standby cells are not addressable.
- When the hash collisions occur in an addressable cell in the hash table, the conflicting items can be stored in its standby cells.

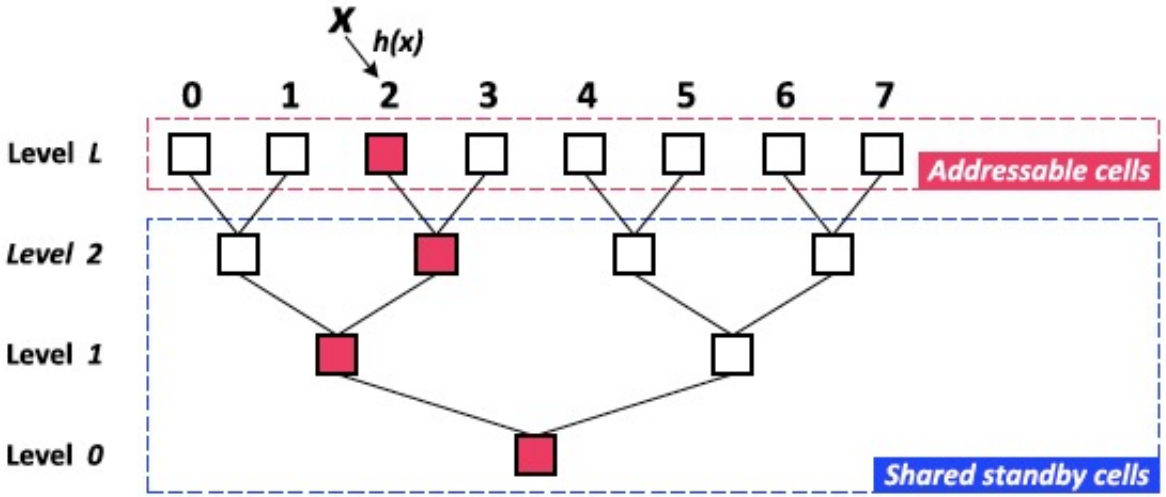
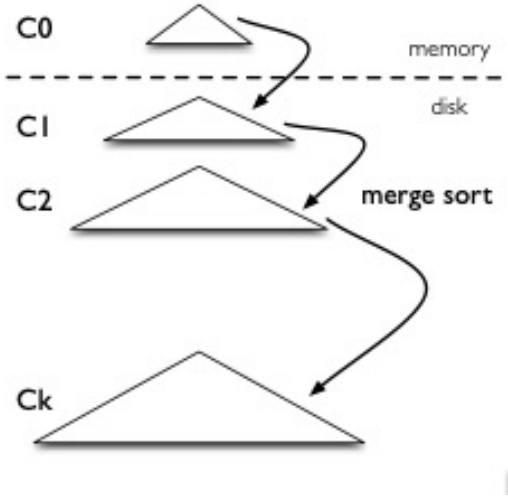
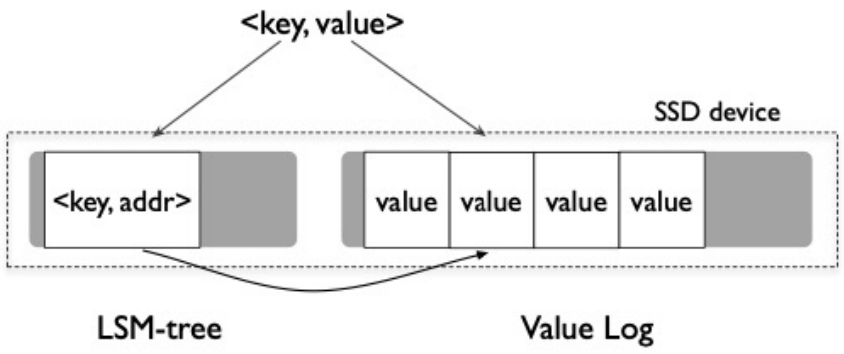


Image by: Zuo, Pengfei, and Yu Hua; 2017

WiscKey [Lu, Lanyue, et al; 2017]

Image by: Lu, Lanyue, et al; 2017



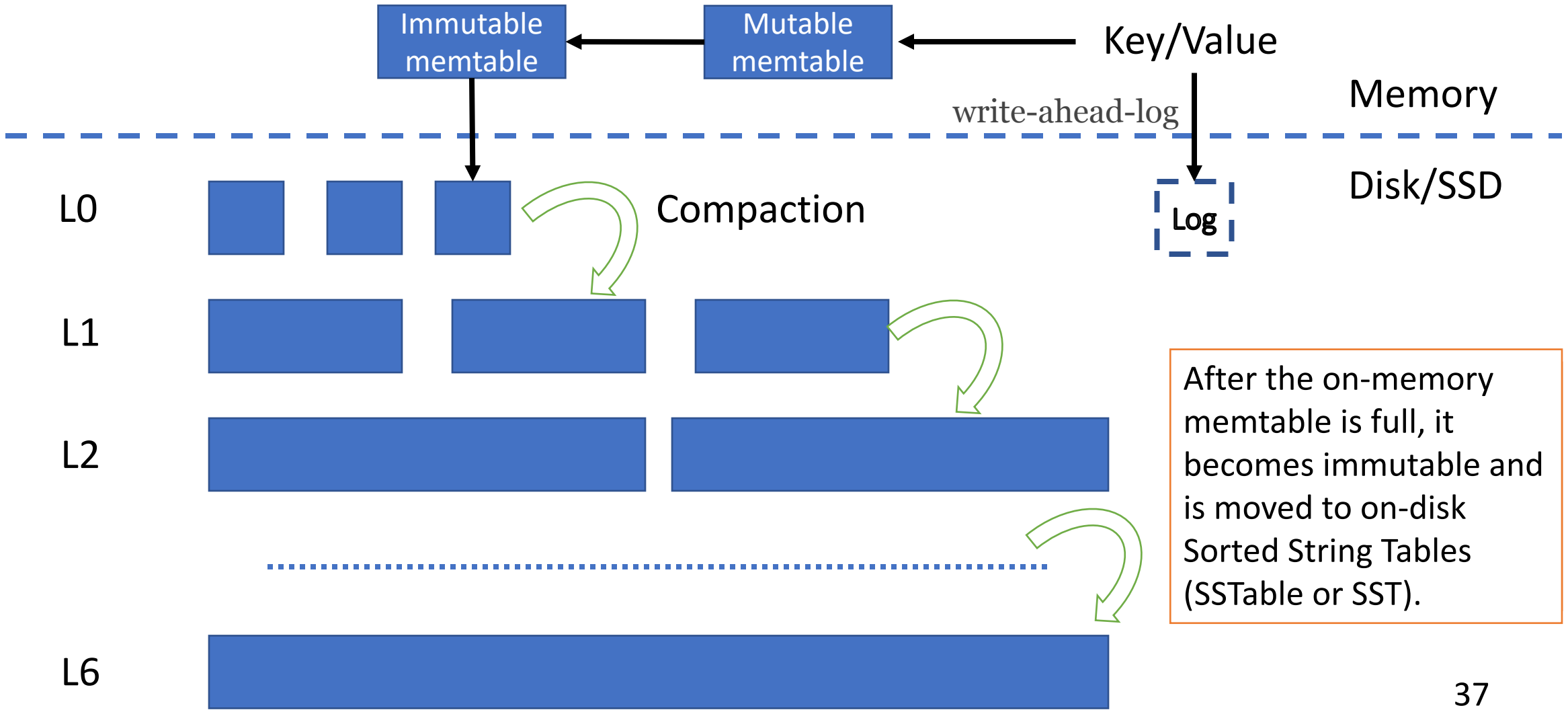


- LSM-tree-based key-value store that separates keys and values to minimize write and read amplification.
- WiscKey stores key-value pairs into an append-only log and the LSM-tree simply serves as a primary index that maps each key to its location in the log.
- While this can greatly reduce the write cost by only merging keys.

NoveLSM

This method is a persistent LSM-based key-value storage system designed to take advantage of having a non-volatile memory in its design.

LevelDB



After the on-memory memtable is full, it becomes immutable and is moved to on-disk Sorted String Tables (SSTable or SST).

NoveLSM

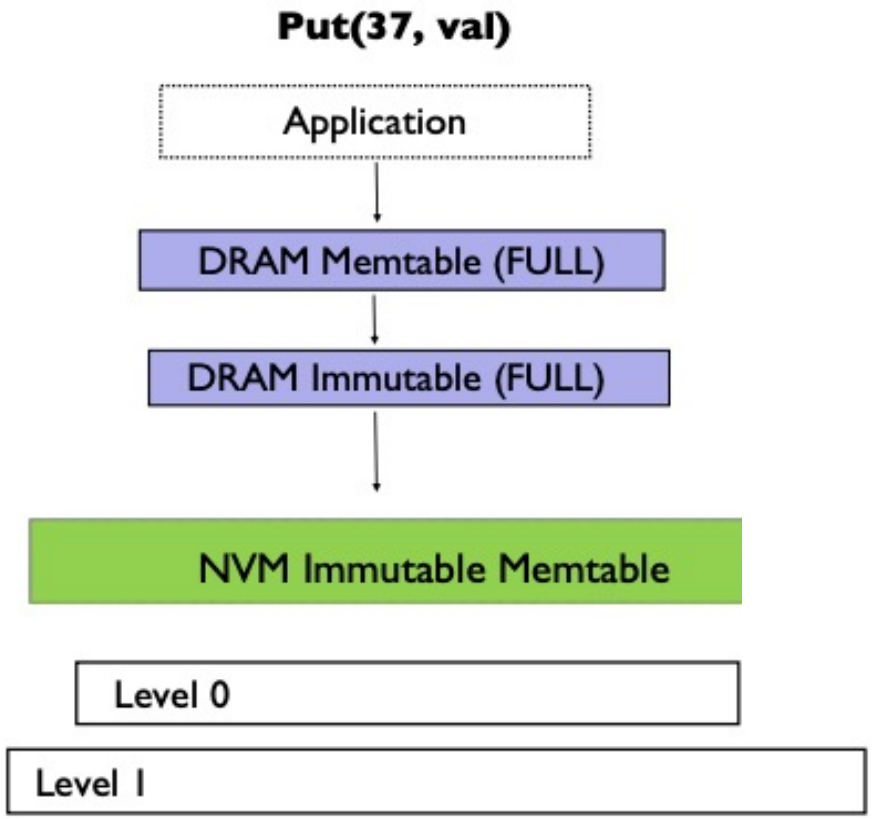
[Kannan, Sudarsun, et al; 2018]

- To tackle NVM's limited write endurance, NoveLSM comes up with a new design, where only the parts of the key/value store that do not need to be changed frequently, such as immutable memtables, are handled by NVM.
- On the other hand, other parts, such as mutable memtables, which need constant updates and data movements, are placed on DRAM, which do not have any restrictions on write operation.

NoveLSM

Immutable NVM Design

Reduce serialization with a immutable persistent skip list



Copy data to large NVM memtable w/o serialization

Reads avoid deserialization

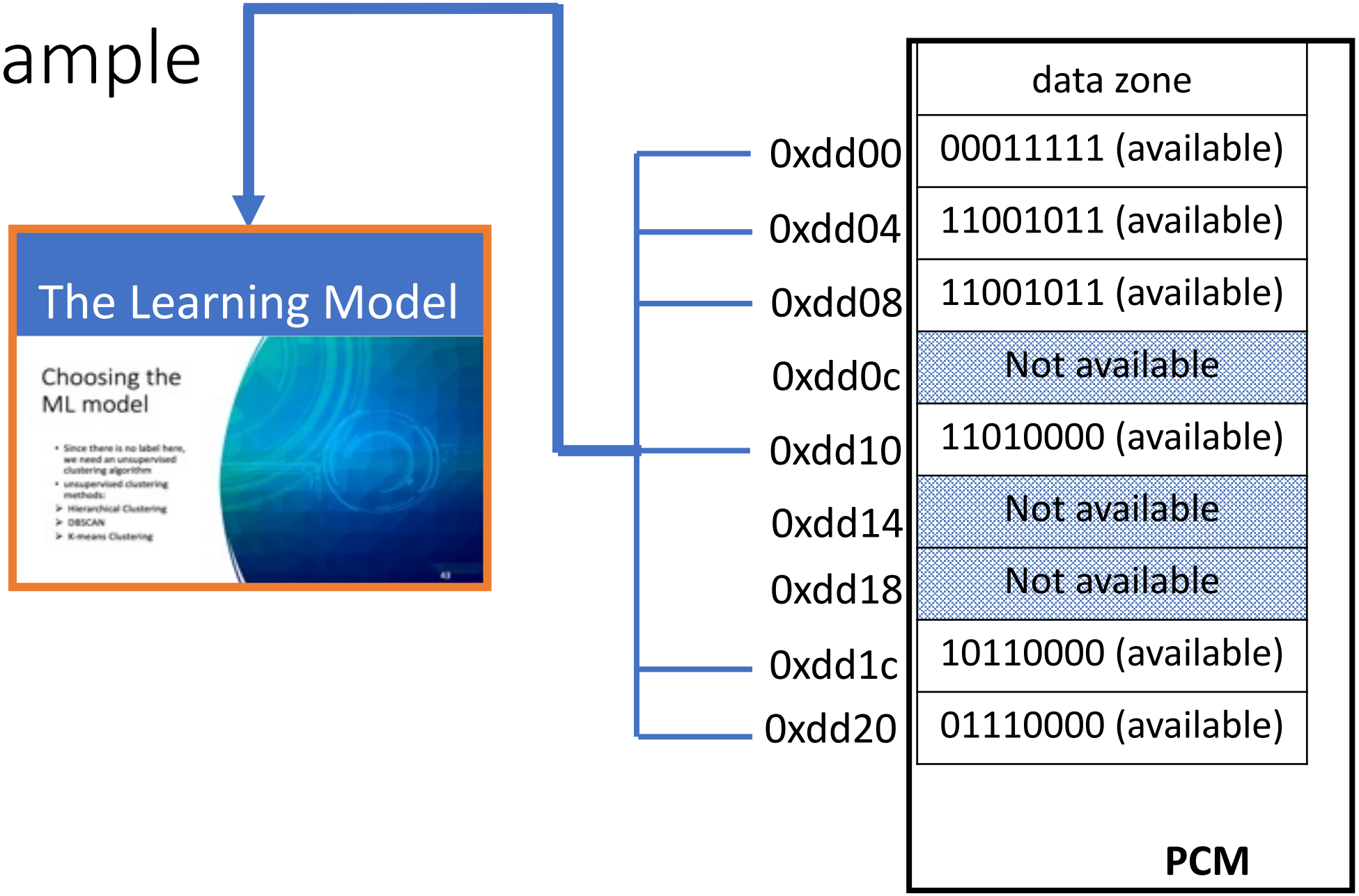
Image by: https://www.usenix.org/sites/default/files/conference/protected-files/atc18_slides_kannan.pdf

Memory-awareness

- The methods in this group tries to take advantage of existing patterns in real-world data.

- One way is to use a machine learning method to learn the existing data distribution among real-world workloads to decrease the number of bit flips in write operations.

Example


The Learning Model
Choosing the ML model
Since there is no label here, we need an unsupervised clustering algorithm
unsupervised clustering methods:
Hierarchical Clustering
DBSCAN
K-means Clustering
0xdd00
0xdd04
0xdd08
0xdd0c
0xdd10
0xdd14
0xdd18
0xdd1c
0xdd20
data zone
00011111 (available)
11001011 (available)
11001011 (available)
Not available
11010000 (available)
Not available
Not available
10110000 (available)
01110000 (available)
PCM

Choosing the ML model

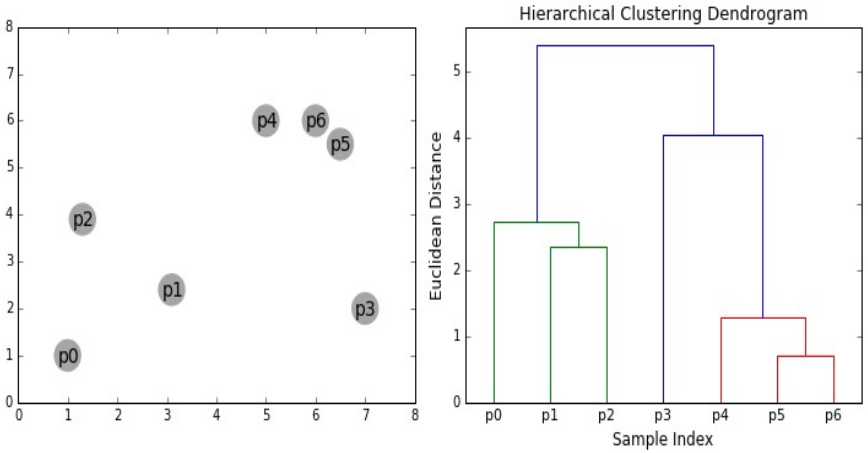
- Since there is no label here, we need an unsupervised clustering algorithm
- unsupervised clustering methods:

➢ Hierarchical Clustering

➢ DBSCAN

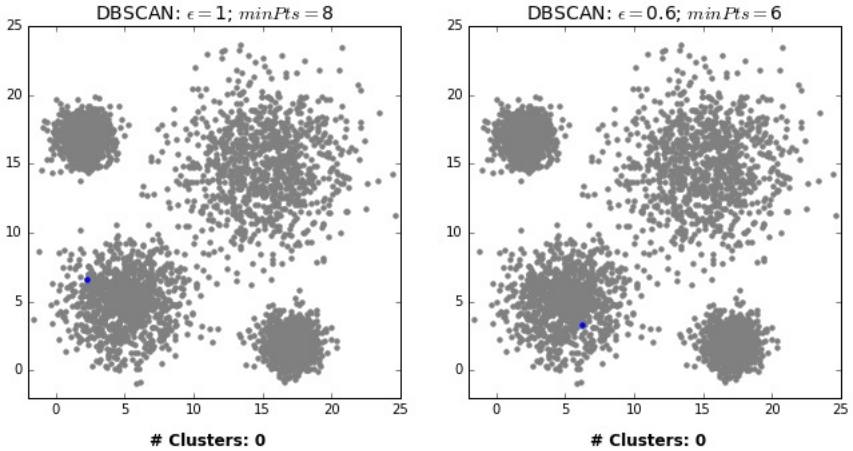
➢ K-means Clustering

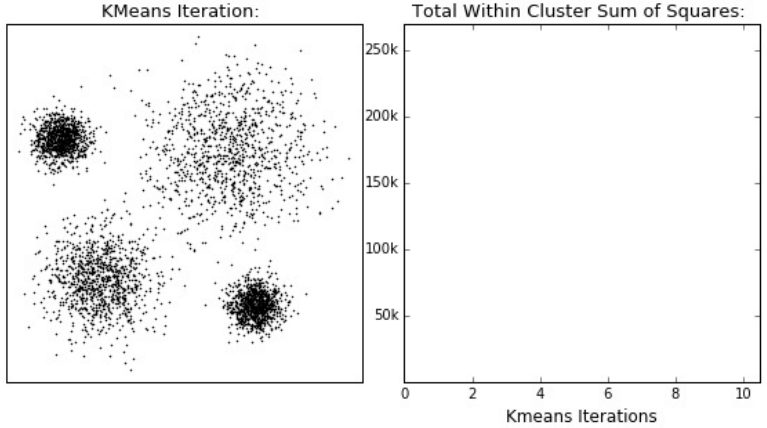
Unsupervised clustering algorithms



Hierarchical Clustering

Density-Based Clustering

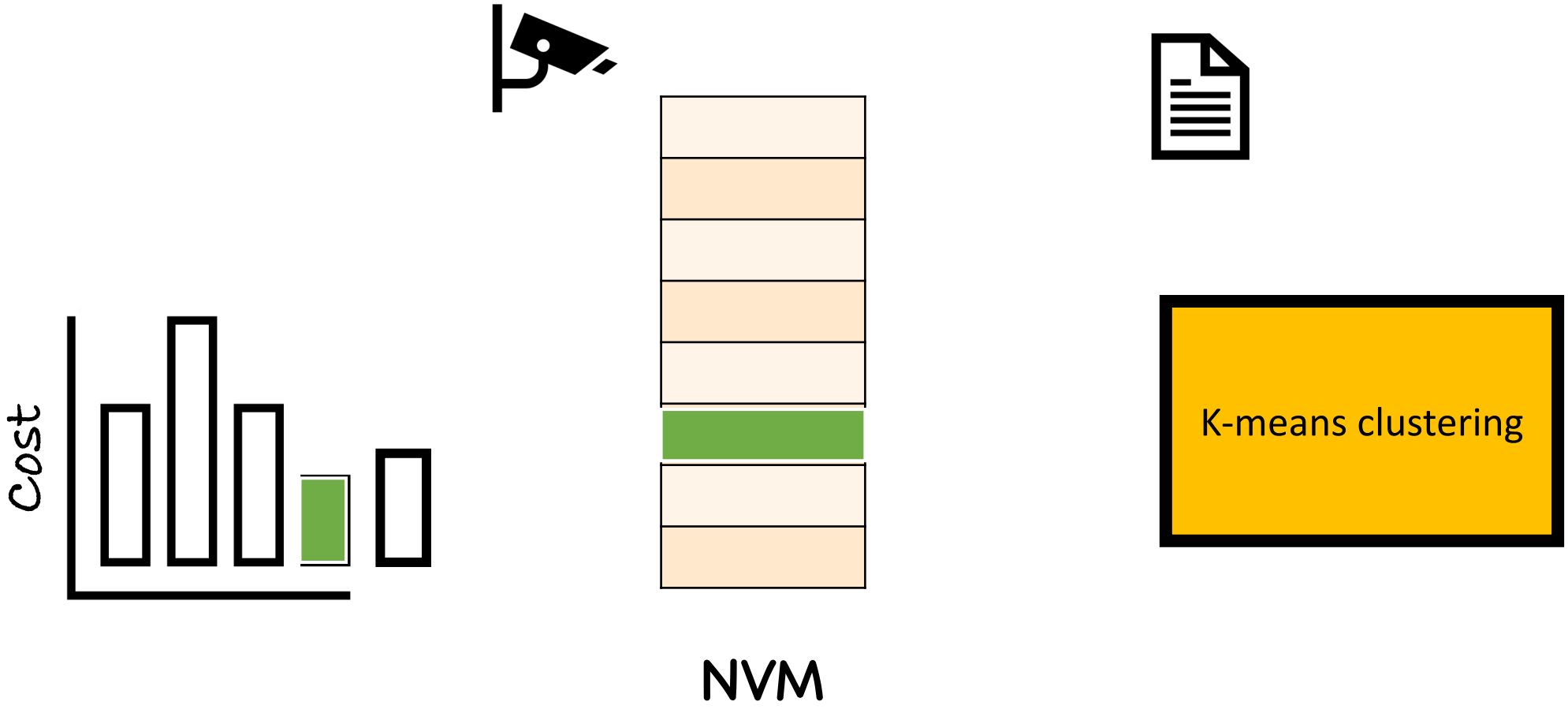




K-means Clustering

Images by: https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/

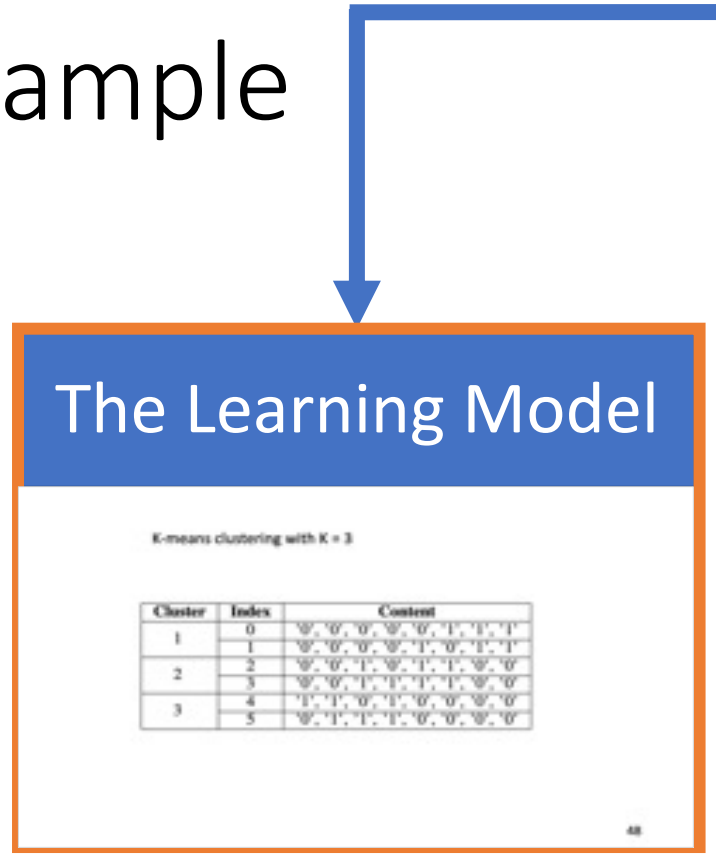
Predict and Write: Using K-Means Clustering to Extend the Lifetime of NVM Storage (ICDE2021) (Kargar, Saeed, et al; 2021)



ML model

- PNW utilizes K-means clustering to cluster data elements into a number of clusters based on their similarity.
- In PNW, each memory location is encoded as a vector of bits, each of which is used as a feature/dimension.
- The entire data zone can be encoded as a 2D tensor (that is, an array of vectors) of shape (n, m), where the first axis (n) represents the samples (old data) and the second axis (m) represents the features.

Example

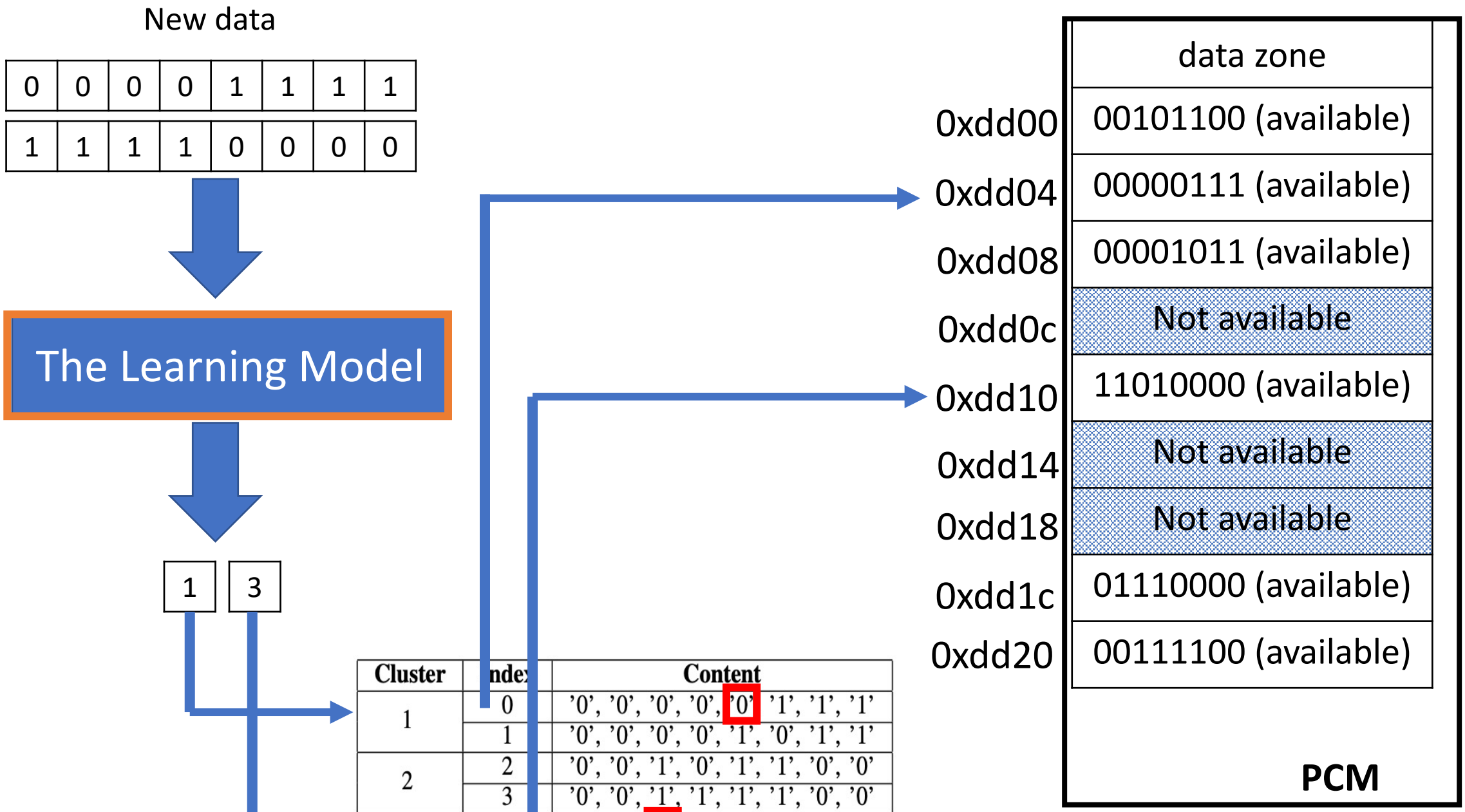


	data zone
0xdd00	00000111 (available)
0xdd04	00101100 (available)
0xdd08	00001011 (available)
0xdd0c	Not available
0xdd10	11010000 (available)
0xdd14	Not available
0xdd18	Not available
0xdd1c	01110000 (available)
0xdd20	00111100 (available)

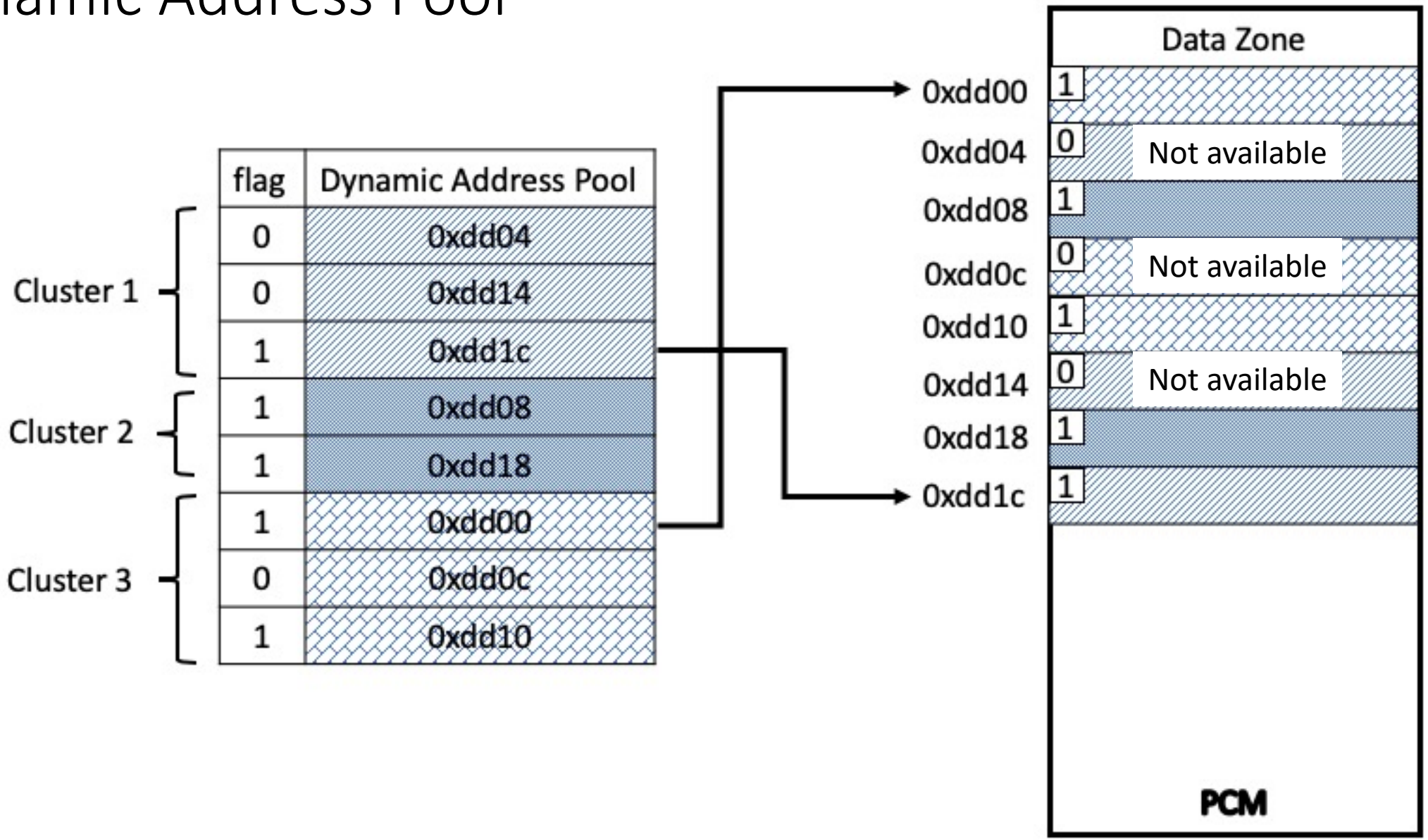
PCM

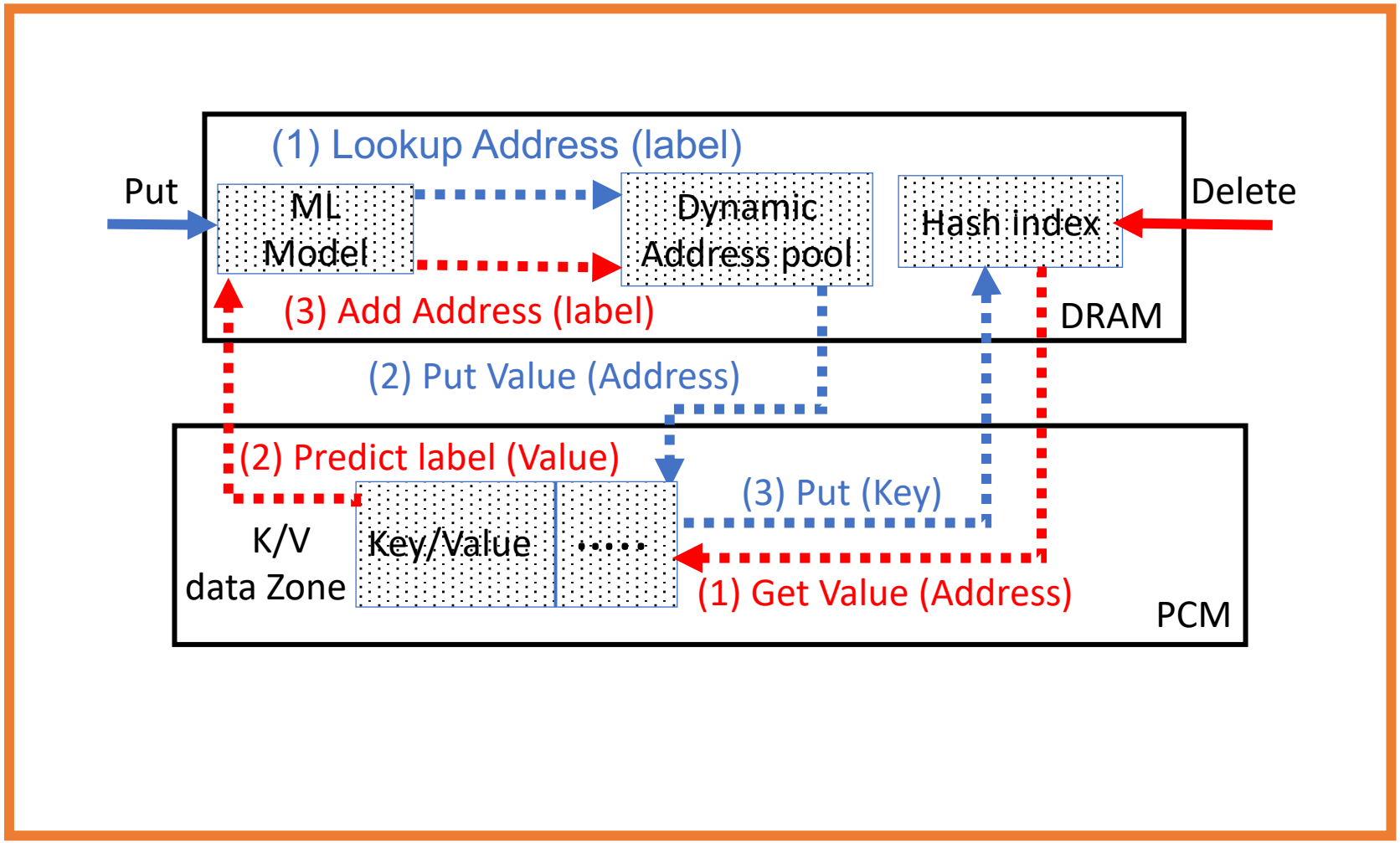
K-means clustering with K = 3

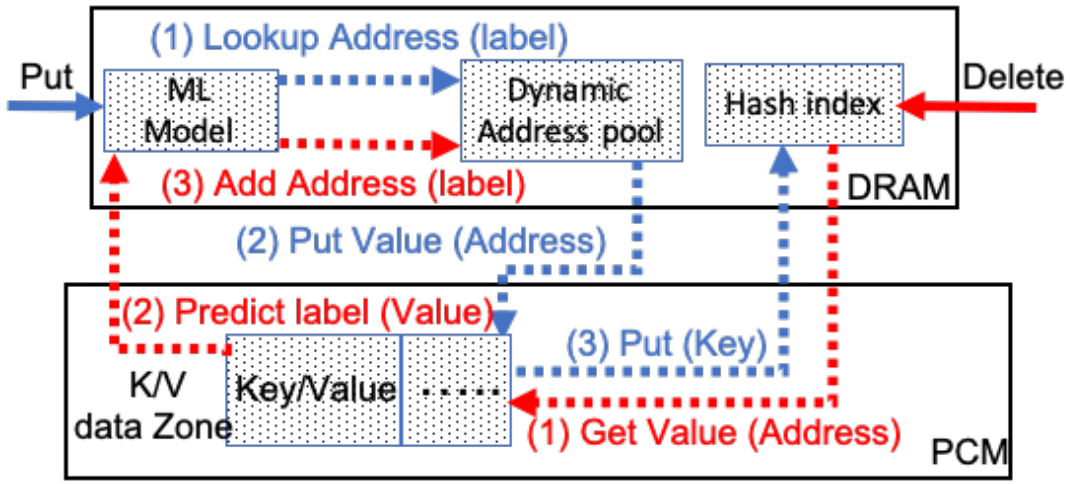
Cluster	Index	Content
1	0	’0’, ’0’, ’0’, ’0’, ’0’, ’1’, ’1’, ’1’
	1	’0’, ’0’, ’0’, ’0’, ’1’, ’0’, ’1’, ’1’
2	2	’0’, ’0’, ’1’, ’0’, ’1’, ’1’, ’0’, ’0’
	3	’0’, ’0’, ’1’, ’1’, ’1’, ’1’, ’0’, ’0’
3	4	’1’, ’1’, ’0’, ’1’, ’0’, ’0’, ’0’, ’0’
	5	’0’, ’1’, ’1’, ’1’, ’0’, ’0’, ’0’, ’0’

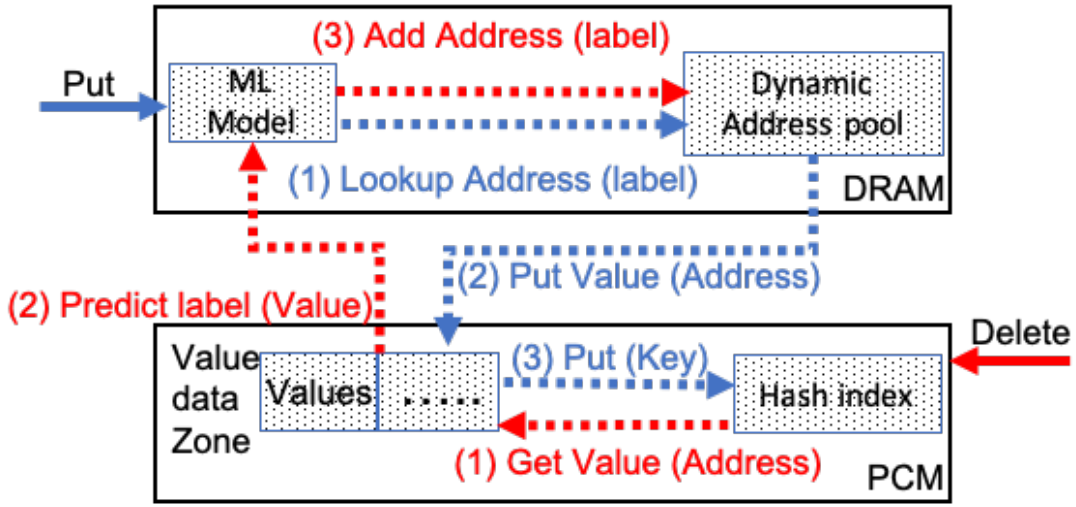

New data
0 0 0 0 1 1 1 1
1 1 1 1 0 0 0 0
The Learning Model
1
3
Cluster
Index
Content
1
0
'0', '0', '0', '0', '0', '1', '1', '1'
1
'0', '0', '0', '0', '1', '0', '1', '1'
2
2
'0', '0', '1', '0', '1', '1', '0', '0'
3
'0', '0', '1', '1', '1', '1', '0', '0'
3
4
'1', '1', '0', '1', '0', '0', '0', '0'
5
'0', '1', '1', '1', '0', '0', '0', '0'
0xdd00
0xdd04
0xdd08
0xdd0c
0xdd10
0xdd14
0xdd18
0xdd1c
0xdd20
data zone
00101100 (available)
00000111 (available)
00001011 (available)
Not available
11010000 (available)
Not available
Not available
01110000 (available)
00111100 (available)
PCM

Dynamic Address Pool




(1) Lookup Address (label)
Put
ML
Model
Dynamic
Address pool
Hash index
Delete
(3) Add Address (label)
DRAM
(2) Put Value (Address)
(2) Predict label (Value)
(3) Put (Key)
K/V
data Zone
Key/Value
.....
(1) Get Value (Address)
PCM


(1) Lookup Address (label)
Put
ML Model
Dynamic Address pool
Hash index
Delete
(3) Add Address (label)
DRAM
(2) Put Value (Address)
(2) Predict label (Value)
(3) Put (Key)
K/V data Zone
Key/Value
(1) Get Value (Address)
PCM


(3) Add Address (label)
Put
ML Model
Dynamic Address pool
(1) Lookup Address (label)
DRAM
(2) Put Value (Address)
(2) Predict label (Value)
Delete
Value data Zone
Values
(3) Put (Key)
Hash index
(1) Get Value (Address)
PCM

Challenges

- Dimensionality problem (the curse of dimensionality)
  - **Principal Component Analysis** (**PCA**)

- Determining the Number of Clusters
  - **Sum of Square Error graph to find the optimal K and "elbow method"**

Conclusion

- NVMs are becoming an integral part of the future computer systems
- Overcome they limitation, especially their low write endurance
- There is an opportunity now for researchers in data management systems to adopt solutions to overcome these limitations of NVMs that would be essential for their adoption and success.

skargar@ucsc.edu

nawabf@uci.edu

References

- Yang, Byung-Do, et al. "A low power phase-change random access memory using a data-comparison write scheme." 2007 IEEE International Symposium on Circuits and Systems. IEEE, 2007.
- Fong, Scott W., Christopher M. Neumann, and H-S. Philip Wong. "Phase-change memory—Towards a storage-class memory." IEEE Transactions on Electron Devices 64.11 (2017): 4374-4385.
- Cho, Sangyeun, and Hyunjin Lee. "Flip-N-Write: A simple deterministic technique to improve PRAM write performance, energy and endurance." Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture. 2009.
- Luo, Xianlu, et al. "Enhancing lifetime of NVM-based main memory with bit shifting and flipping." *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*. IEEE, 2014.
- Jalili, Majid, and Hamid Sarbazi-Azad. "Captopril: Reducing the pressure of bit flips on hot locations in non-volatile main memories." *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2016.
- Palangappa, Poovaiah M., and Kartik Mohanram. "Flip-Mirror-Rotate: An architecture for bit-write reduction and wear leveling in non-volatile memories." *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*. 2015.
- Luo, Xianlu, et al. "Enhancing lifetime of NVM-based main memory with bit shifting and flipping." *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*. IEEE, 2014.
- Zuo, Pengfei, and Yu Hua. "A write-friendly and cache-optimized hashing scheme for non-volatile memory systems." *IEEE Transactions on Parallel and Distributed Systems* 29.5 (2017): 985-998.

- Lu, Lanyue, et al. "Wisckey: Separating keys from values in ssd-conscious storage." ACM Transactions on Storage (TOS)13.1 (2017): 1-28.
- Kannan, Sudarsun, et al. "Redesigning LSMs for nonvolatile memory with NoveLSM." 2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18). 2018.
- Kargar, Saeed, Heiner Litz, and Faisal Nawab. "Predict and Write: Using K-Means Clustering to Extend the Lifetime of NVM Storage." 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021.