Research Summary

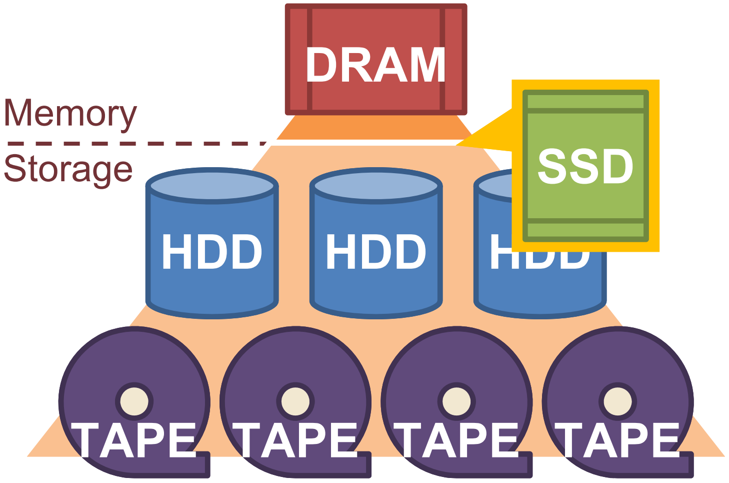
Yuan-Hao Chang
Deputy Director / Research Fellow / Professor
Institute of Information Science,
Academia Sinica

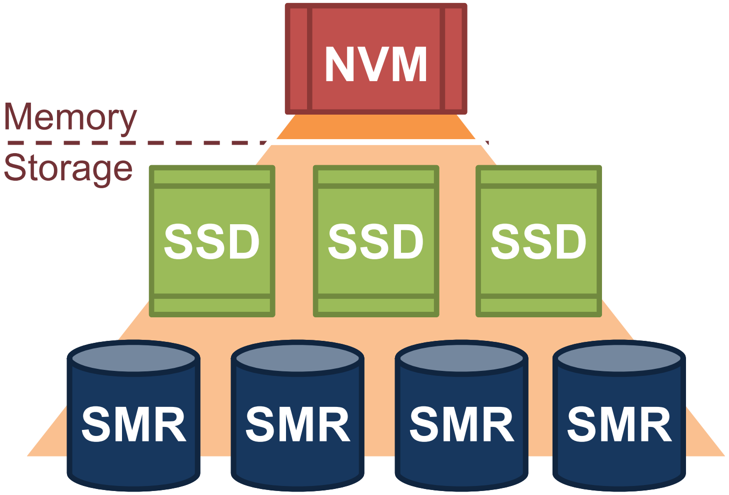
Research Interests

- Emerging Memory Technologies
- Non-volatile Memories
- Memory/Storage Systems
- Embedded Systems
- Operating Systems

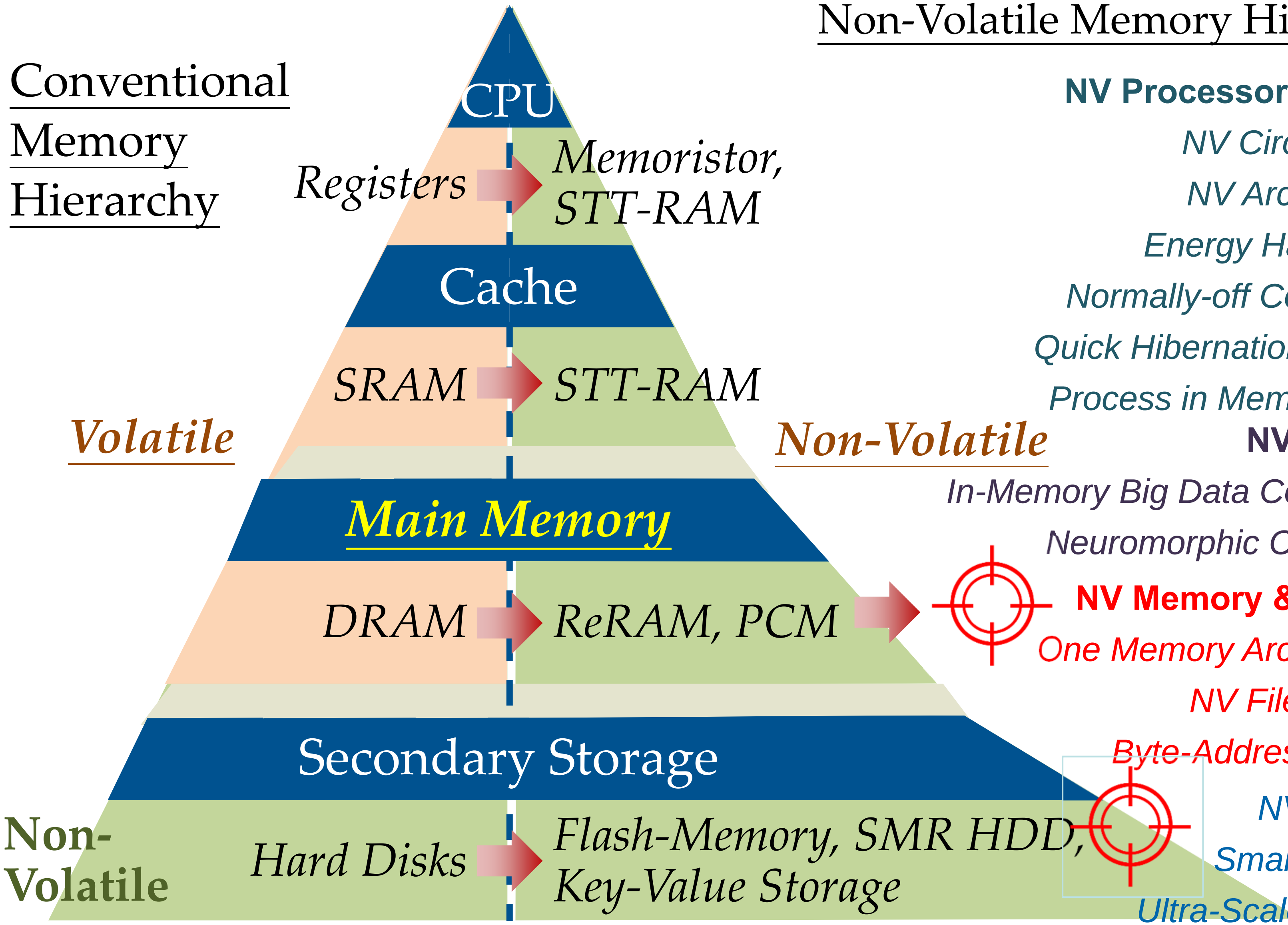
Next-generation Memory/Storage Systems

- ***Present***:
  - **DRAM:** Main memory
  - **Solid-state Drive (SSD):** Cache/buffer for high performance environments
  - **Hard Disk (HDD):** Main storage media.
  - **Magnetic tapes:** Deep data archiving
- ***Foreseeable Future***:
  - **Non-volatile memory (NVM):** Enlarge the scalability of in-memory computing
    - Optane DIMM, NVDIMM, Process-in-Memory
  - **SSDs:** Take over the main storage media
    - Reasons: Density ↑ and cost ↓
    - OCSSD, ZNS-SSD, Z-NAND, Optane
  - **HDDs** & **Tapes:** Replaced by new magnetic recording technologies
    - Objectives: Density ↑ and cost ↓
    - Promising Candidate: **SMR**





Non-volatile Computing - Key Technology for Future IoTs



Non-Volatile Memory Hierarchy

NV Processor & Cache

NV Circuit/Chip,

NV Architecture,

Energy Harvesting,

Normally-off Computing,

Quick Hibernation/Restore

Process in Memory (PIM)

NV Memory

In-Memory Big Data Computing,

Neuromorphic Computing

NV Memory & Storage

One Memory Architecture,

NV File System,

Byte-Addressable FS

NV Storage

Smart Storage

Ultra-Scale Storage

Research Directions

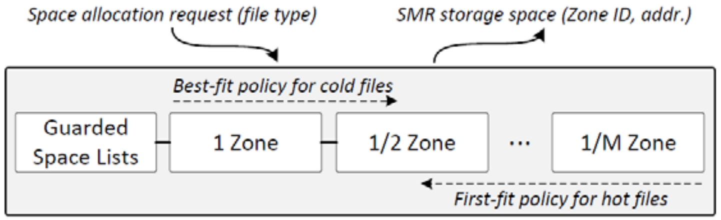
- Software: **Non-Volatile System Software**
  - Most researches are based on the existing operating system designs.
    - Memory Space: Managed by the memory management.
    - Storage Space: Managed by the file systems.
  - *New non-volatile operating system and system software are* needed by the non-volatile computing environment.
    - The objective is to get rid of unnecessary software stacks (e.g., page cache, swap).
- System: **Non-Volatile Systems**
  - System in a Chip (SiC): Control units + memory + storage are all in a chip.
  - **Process-in/near-memory (PIM)**: Computing + Memory + Storage
  - **Self-sustainable Sensor Nodes** (Green IoT)
    - Instant backup & restore capabilities of NV processors.
    - Low standby power of NV memory (& storage).
    - Intermittent systems (Energy harvesting systems).
- Storage (including file systems):
  - Approximate storage
  - Smart storage (CXL)
  - UPMAN
- Application (supported by Memory/Storage):
  - E.g., DNA sequencing, graph processing, random forest

Research Summary 2022

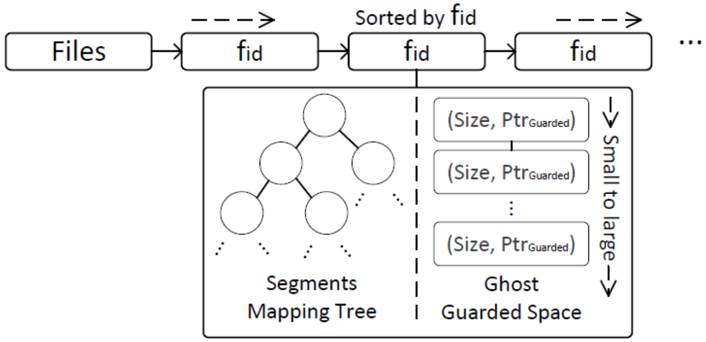
1. Storage Systems - Flash Drives and SMR Disks

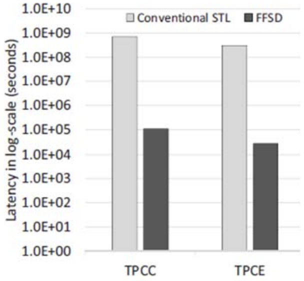
File-Oriented Fast Secure Deletion for SMR Drives (IEEE TCAD'22, DAC 2019)

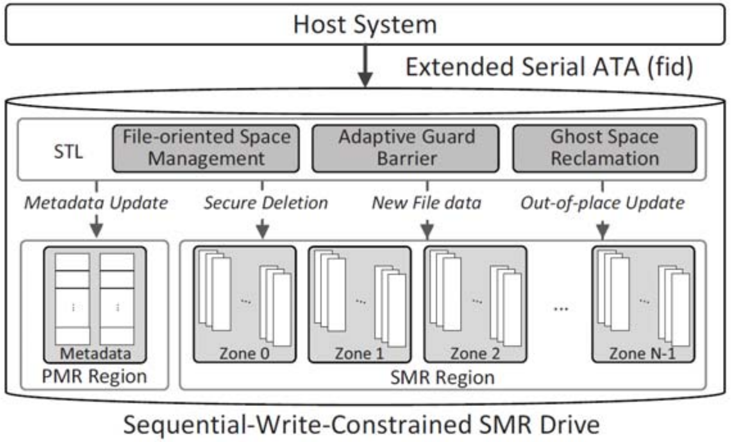
- Observation
  - Existing secure deletion approaches are inefficient
    - File systems have no knowledge of the data layout on the storage device
    - Storage devices are not aware of the file information in file systems
  - This inefficiency is exaggerated on shingled magnetic recording (SMR) drive
- Goal: Enable file-oriented fast secure deletion on sequential-write-constraint SMR drives
- Main Idea
  - Minimize secure deletion overhead
    - Adaptive guard barrier mechanism
  - Manage storage space with file awareness
    - File-oriented space management design
  - Enhance the space utilization
    - Free space reclamation scheme

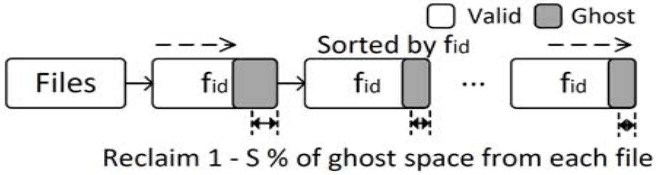

Space allocation request (file type)
SMR storage space (Zone ID, addr.)
Best-fit policy for cold files
Guarded Space Lists
1 Zone
1/2 Zone
···
1/M Zone
First-fit policy for hot files

Adaptive guard barrier mechanism









Secure Deletion Latency Reduction: 8660x

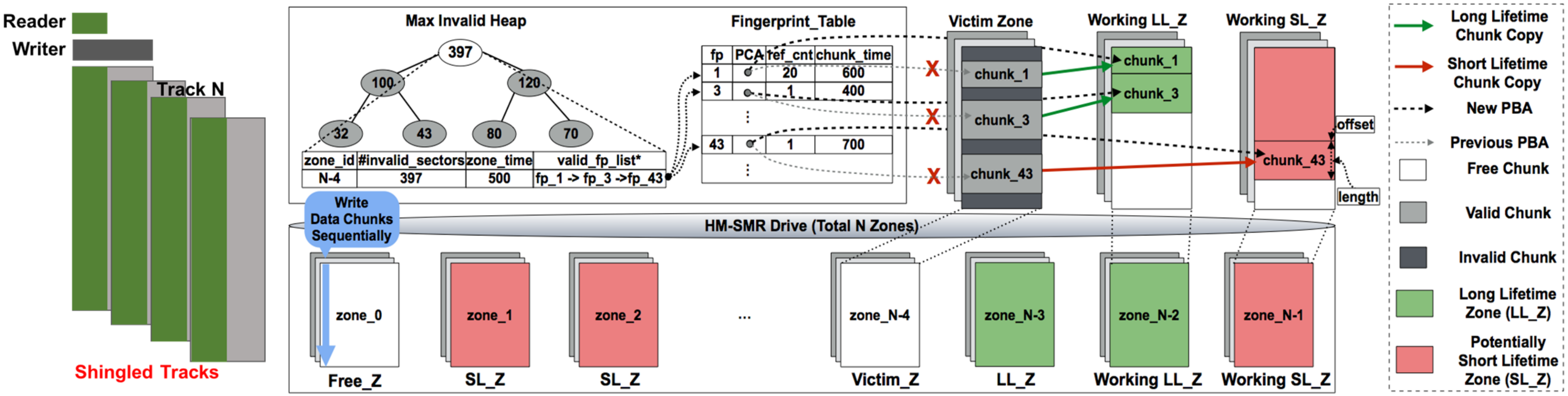
Space Overhead: 14.21%

- Shuo-Han Chen, Chun-Feng Wu, Ming-Chang Yang, and Yuan-Hao Chang, "A File-Oriented Fast Secure Deletion Strategy for Shingled Magnetic Recording Drives," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 41, no. 8, pp. 2463-2476, Aug. 2022.
- Shuo-Han Chen, Ming-Chang Yang, Yuan-Hao Chang, and Chun-Feng Wu, "Enabling File-Oriented Fast Secure Deletion on Shingled Magnetic Recording Drives," ACM/IEEE Design Automation Conference (DAC), Las Vegas, Nevada, USA, Jun. 2-6, 2019. **(Top Conference)**

SMR-Aware Deduplication

- We are among the pioneers to propose an *SMR-aware deduplication design* to improve the run time performance for SMR disks with considering (1) the SMR write constraint and (2) the deduped chunk lifetime/behavior. (IEEE TCAD’22, DAC’18)
  – This design advocates a vertical integration solution by managing the host-managed SMR drives with deduplication system.
  – The idea is to predict and separate “chunk lifetime” based on the semantic information.
- The proposed design was evaluated by the famous Skylight disk emulation tool and proved to improve performance for 82% .

	Low Reference ref_cnt==1	High Reference ref_cnt>1
Elder Chunks chunk_time < zone_time	Long Lifetime	Long Lifetime
Younger Chunks chunk_time ≥ zone_time	Potentially Short Lifetime	Long Lifetime

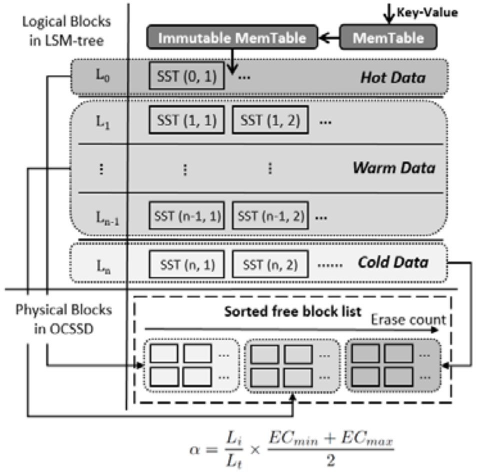


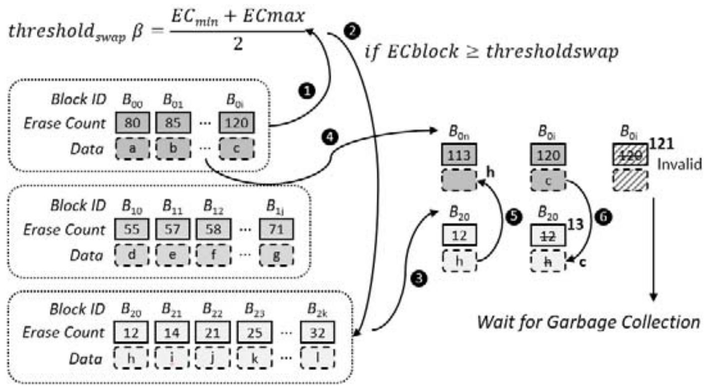
- Chun-Feng Wu, Martin Kuo, Ming-Chang Yang, and Yuan-Hao Chang, "Performance Enhancement of SMR-based Deduplication SystemsIEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 41, no. 9, pp. 2835-2848, Sep. 2022.
- Chun-Feng Wu, Ming-Chang Yang, and Yuan-Hao Chang, "Improving Runtime Performance of Deduplication System with Host-Managed SMR Storage Drives," ACM/IEEE Design Automation Conference (DAC), San Francisco, USA, Jun. 24-28, 2018. **(Top Conference)**

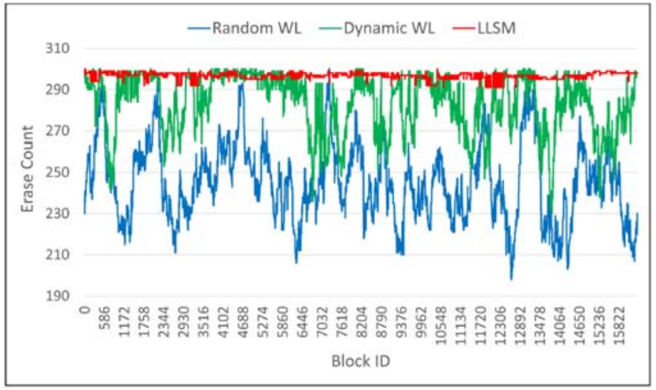
LLSM: A Lifetime-Aware Wear-Leveling for LSM-Tree on NAND Flash

(IEEE TCAD’22, CASES’22)

- Lifetime is a critical issue for flash memory-based SSDs. However, the existing wear-leveling strategies fail to solve the endurance issues caused by LSM-tree based key-value stores on SSDs
- We design a lifetime-aware wear-leveling for LSM-tree on flash memory-based SSDs to prolong the SSD lifetime via OCSSD
  - Level-Aware Block Allocation: properly allocate the data in the LSM-tree to the memory blocks according to the data hotness based on the LSM-tree levels
  - Proactive Block Swapping: intelligently swap the old and young blocks to prevent uneven wear-out among the memory blocks in terms of long-term usage







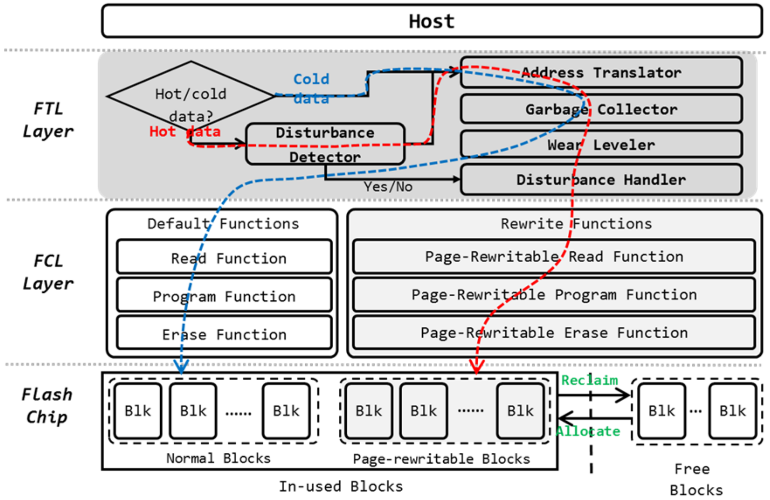
- Extensive results show that the proposed LLSM improves the SSD lifetime up to 21.7%

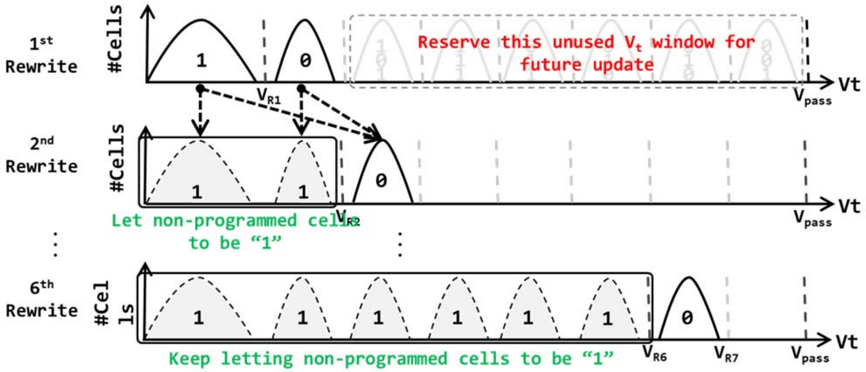
- Dharamjeet, Yi-Shen Chen, Tseng-Yi Chen, Yuan-Hung Kuan, and Yuan-Hao Chang, "LLSM: A Lifetime-Aware Wear-Leveling for LSM-Tree on NAND Flash Memory," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 41, no. 11, pp. 3946-3956, Nov. 2022. (Integrated with ACM/IEEE CASES’22)
- Dharamjeet, Yi-Shen Chen, Tseng-Yi Chen, Yuan-Hung Kuan, and Yuan-Hao Chang, "LLSM: A Lifetime-Aware Wear-Leveling for LSM-Tree on NAND Flash Memory," ACM/IEEE International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES), Hybrid-Shanghai, China, Oct. 7-14, 2022. (Journal Track, Integrated with IEEE TCAD) **(Top Conference)**

Enduring More Data through Enabling Page Rewrite Capability on MLC Flash

(SAC 2022)

- Goal
  - To sustain more and more data in flash-based storage system
  - To minimize the adverse effect of effective disturbance while taking advantage of page-rewrite-programming
    - Effective disturbance - a page with a lower number of rewrites was disturbed because of rewriting an adjacent page with a higher number of rewrite
- Main Idea
  - The in-used blocks are divided into normal blocks and page-rewritable blocks and the idea is to increase the likelihood of generating invalid data as much as possible on those page-rewritable blocks that enable page rewrite
    - Storing **write-cold** data to normal blocks and storing **write-hot** data to page-rewritable blocks
  - To address the disturbance problem
    - in-place rewrite - rewriting the same data to the same page can increase its Vt and make it less susceptible to the disturbance
    - page migration – moving data to other pages that does not contain less valid data



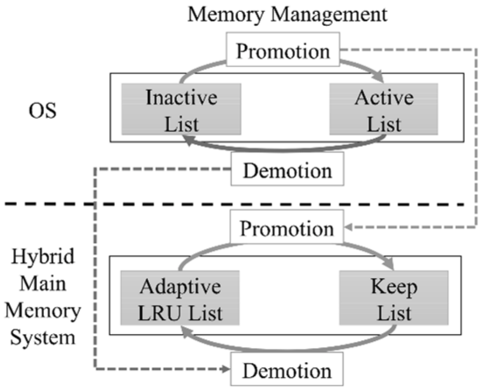

1st Rewrite
#Cells
1
0
Reserve this unused Vt window for future update
Vt
V_R1
V_pass
2nd Rewrite
#Cells
1
1
0
V_R2
Let non-programmed cells to be "1"
Vt
V_pass
6th Rewrite
#Cells
1
1
1
1
1
1
0
Keep letting non-programmed cells to be "1"
V_R6
V_R7
V_pass
Vt

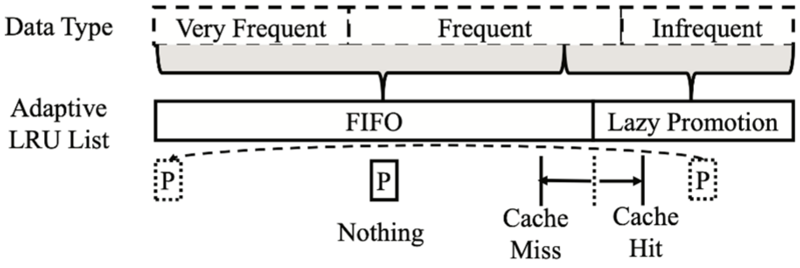
- Yu-Ming Chang, Chien-Chung Ho, Che-Wei Tsao, Shu-Hsien Liao, Wei-Chen Wang, Tei-Wei Kuo, and Yuan-Hao Chang, "On Enduring More Data Through Enabling Page Rewrite Capability on Multi-level-cell Flash Memory," ACM Symposium on Applied Computing (SAC), Virtual Conference, Apr. 25-29, 2022.

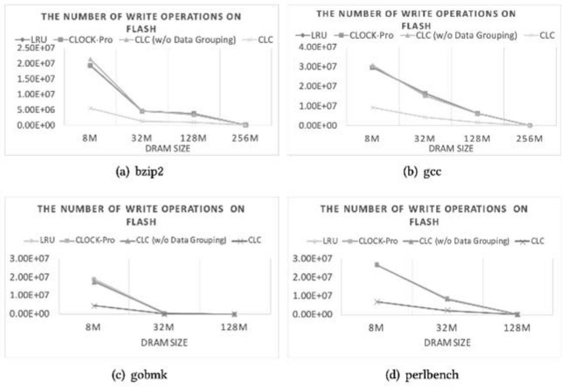
2. NVM Main Memory and Storage

Rethinking the Interactivity of OS and Device Layers in Memory Management (ACM TECS'22)

- Observation:
  - The hybrid main-memory module uses DRAM as the cache of NVMs to enhance its performance and lifetime. However, it also introduces new design challenges in both the OS and the memory module.
- Goal:
  - Rethinking the interactivity of OS and hybrid main-memory module, and propose a cross-layer cache design to optimize the DRAM cache's hit ratio and minimize the time overhead on the data movement between DRAM and NVM.
- Main Idea:
  - Keep List maintains the recent frequently accessed data information from the operating system to optimize the hit ratio of the DRAM cache.
  - Adaptive LRU List dynamically adjusts the length of the FIFO list and the lazy promotion list based on recent access behavior to avoid unnecessary management operations.
  - Data Grouper takes advantage of NVM's block-size read/write feature to reduce the writes to flash memory without significantly sacrificing the DRAM cache hit rate.







(a) bzip2

(b) gcc

(c) gobmk

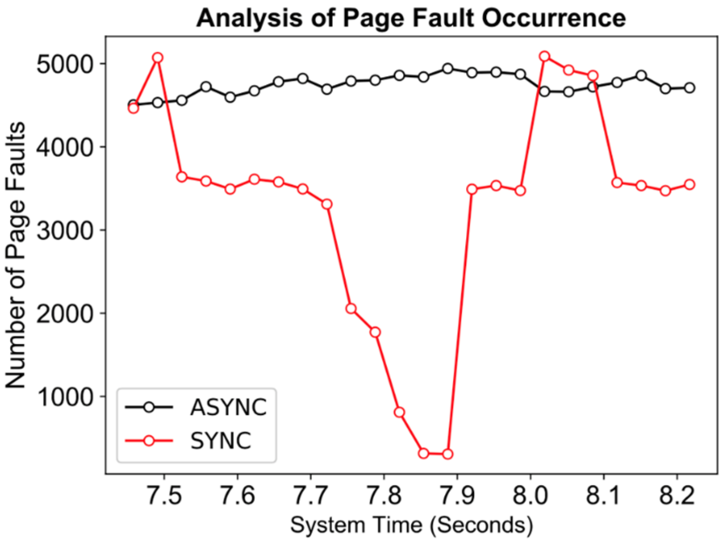
(d) perlbench

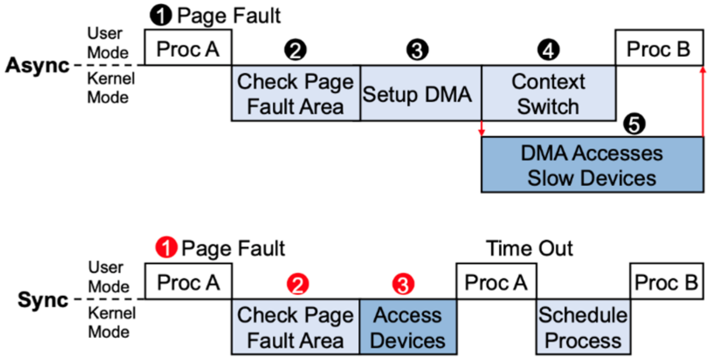
- Tse-Yuan Wang, Chun-Feng Wu, Che-Wei Tsao, Yuan-Hao Chang, Tei-Wei Kuo, and Xue Liu, "Rethinking the Interactivity of OS and Device Layers in Memory Management," ACM Transactions on Embedded Computing Systems (TECS), vol. 21, no. 4, pp. 42:1-42:21, July 2022.

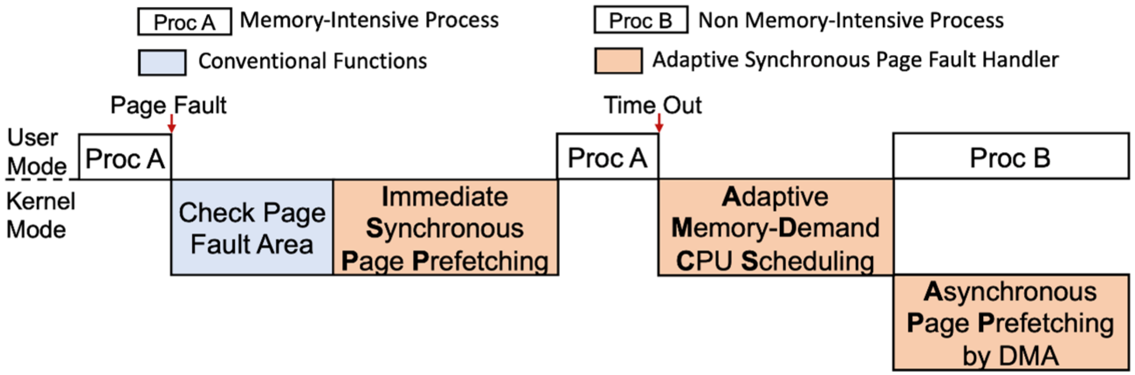
Exploring Synchronous Page Fault Handling

[IEEE TCAD’22, CODES’22 (Best Paper Award)]

- Observation
  - Async page fault handler: Stable results.
  - Sync page fault handler: Number of page faults fluctuates.
    - High Peak: Reconstructing working set.
    - Low Bottom: After reconstructing working set.
- Goal
  - Deal with the working set contention issue so as able to remove the peak of page faults and reduce the occurrence of page faults.
- Main Idea
  - Adjust the designs of CFS CPU scheduler to fit sync page fault handling.
  - Prefetch and reconstruct the working set.



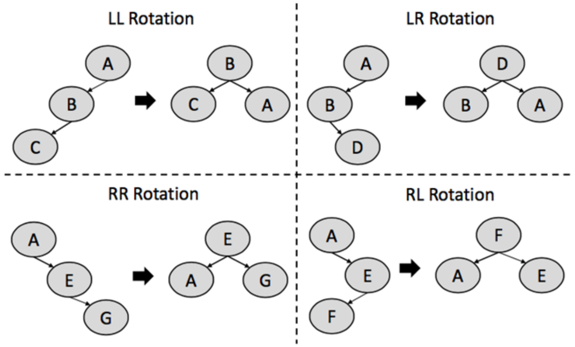


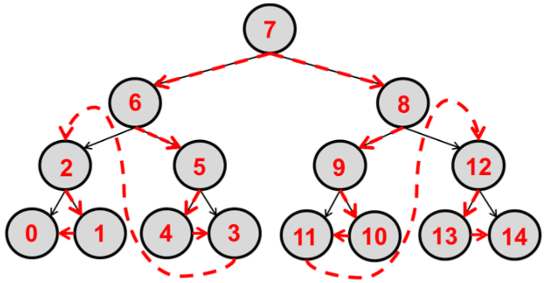

Proc A Memory-Intensive Process
Proc B Non Memory-Intensive Process
Conventional Functions
Adaptive Synchronous Page Fault Handler
Page Fault
Time Out
User Mode
Kernel Mode
Proc A
Check Page Fault Area
Immediate Synchronous Page Prefetching
Proc A
Adaptive Memory-Demand CPU Scheduling
Proc B
Asynchronous Page Prefetching by DMA

- Yin-Chiuan Chen, Chun-Feng Wu, Yuan-Hao Chang, and Tei-Wei Kuo, "Exploring Synchronous Page Fault HandlingIEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 41, no. 11, pp. 3791-3802, Nov. 2022. (Integrated with ACM/IEEE CODES+ISSS'22)
- Yin-Chiuan Chen, Chun-Feng Wu, Yuan-Hao Chang, and Tei-Wei Kuo, "Exploring Synchronous Page Fault Handling," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Hybrid-Shanghai, Oct. 7-14, 2022. (Journal Track, Integrated with IEEE TCAD) (Top Conference)

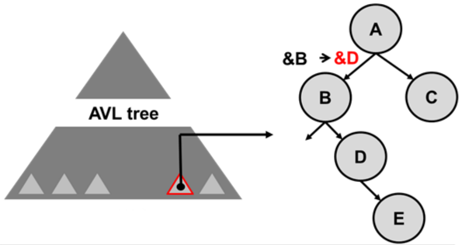
Search Tree Redesign on Phase Change Memory
- Rethinking Self-balancing Binary Tree over PCM

- We exploit the *write heterogeneity of phase change memory* to improve the performance of a self-balancing binary search tree by reducing the rotation overheads of tree balancing (IEEE TCAD in 2022, ASP-DAC 2018)
  - Observations:
    - Rotations of tree balancing are not arbitrary: Some nodes have stronger relation
    - Data-comparison writes eliminate unnecessary bit flips: Reducing the number of bit flips can reduce the write energy and/or latency
  - A write-heterogeneity-aware management scheme of the AVL tree structure is developed for binding nodes with considerations of their relation
    - The relation among nodes of an AVL tree is analyzed by examining possible rotations
    - Our depth-first alternating traversal (DFAT) algorithm is then designed for tree indexing
    - Gray code is leveraged by DFAT to reduce the average number of bit flips per write





Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gray Code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110



Key value	Binary code address	Gray code address
A	0111111111111101	0100000000000011
C	0111111111111110	0100000000000001
B	*0111111111111111*	*0100000000000000*
D	*1000000000000000*	*1100000000000000*
E	1000000000000001	1100000000000001

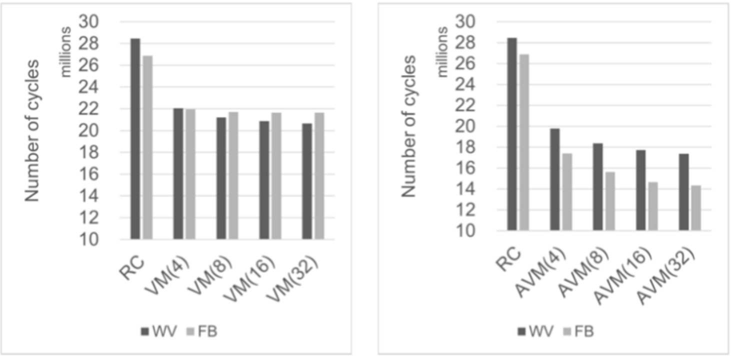
- Che-Wei Chang, Chun-Feng Wu, Yuan-Hao Chang, Ming-Chang Yang, and Chieh-Fu Chang, "Leveraging Write Heterogeneity of Phase Change Memory on Supporting Self-balancing Binary Tree," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 41, no. 6, pp. 1757-1770, Jun. 2022.
- Chieh-Fu Chang, Che-Wei Chang, Yuan-Hao Chang, and Ming-Chang Yang, “Rethinking Self-balancing Binary Search Tree over Phase Change Memory with Write Asymmetry,” in ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC), Jeju Island, Korea, Jan. 22-25, 2018.

GraphRC: Accelerating Graph Processing on Dual-addressing Memory with Vertex Merging

(ICCAD 2022)

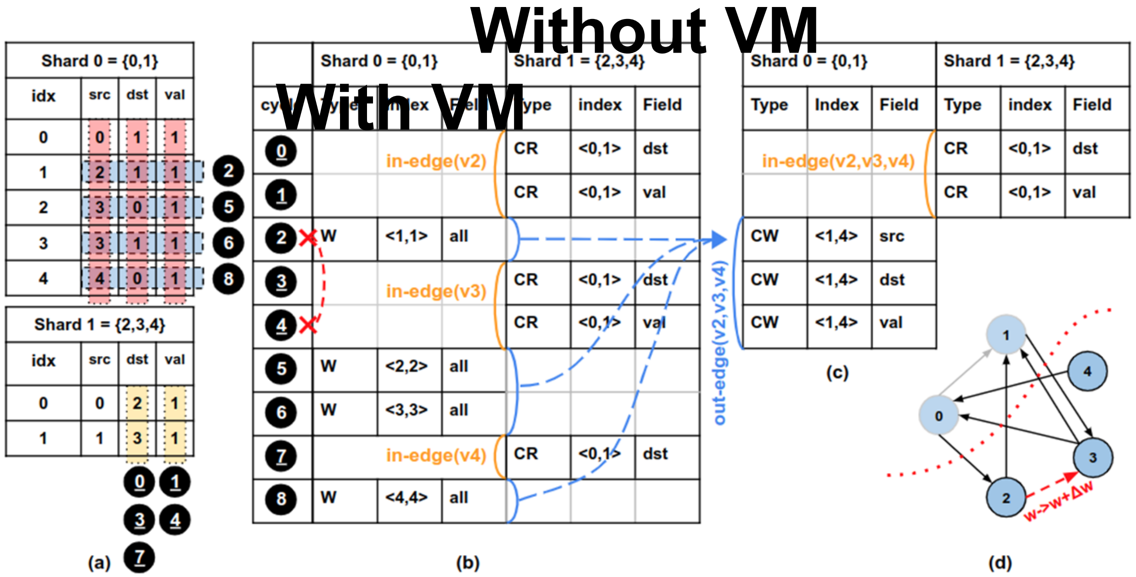
- Observation
  - Graph processing task can be accelerated on dual-addressing memory.
  - Memory accesses suffer from a low utilization rate due to the size mismatching between graph data and the cache block size.
- Solutions
  - Propose vertex merging (VM) that improves the cache block utilization rate by merging memory requests from consecutive vertices.
  - Then, identify data dependencies inherent in a graph that limits the effectiveness of VM. Propose aggressive vertex merging (AVM) that merges vertices based its importance not its data dependency.

AVM reduces execution time by *73.27%* while **VM** offers *22.51%* reduction.



(a) VM applied to PR algo. on WV and FB datasets.

(b) AVM applied to PR algo. on WV and FB datasets.



Shard 0 = {0,1}				
idx	src	dst	val	
0	0	1	1	
1	2	1	1	2
2	3	0	1	5
3	3	1	1	6
4	4	0	1	8

Shard 1 = {2,3,4}			
idx	src	dst	val
0	0	2	1
1	1	3	1
		0 3 7	1 4

(a)

	Shard 0 = {0,1}			Shard 1 = {2,3,4}		
cycle	Type	Index	Field	Type	index	Field
0	in-edge(v2)			CR	<0,1>	dst
1				CR	<0,1>	val
2	W	<1,1>	all			
3	in-edge(v3)			CR	<0,1>	dst
4				CR	<0,1>	val
5	W	<2,2>	all			
6	W	<3,3>	all			
7	in-edge(v4)			CR	<0,1>	dst
8	W	<4,4>	all			

(b)

Shard 0 = {0,1}			Shard 1 = {2,3,4}		
Type	Index	Field	Type	index	Field
in-edge(v2,v3,v4)			CR	<0,1>	dst
			CR	<0,1>	val
CW	<1,4>	src			
CW	<1,4>	dst			
CW	<1,4>	val			

(c)

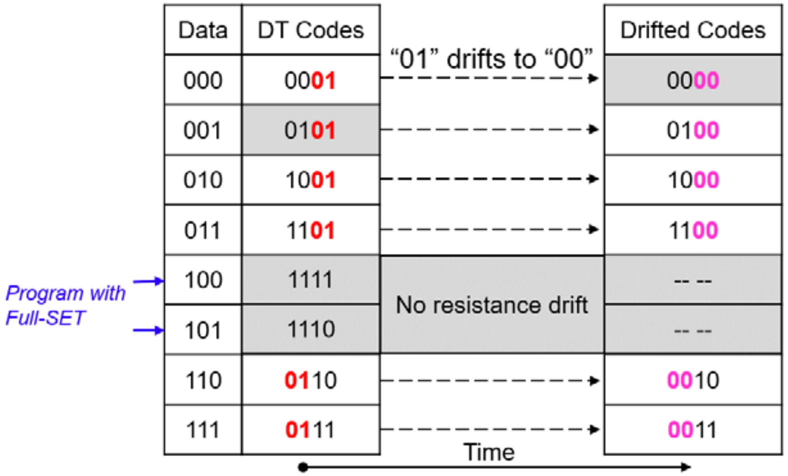
(d)

- Wei Cheng, Chun-Feng Wu, Yuan-Hao Chang, and Ing-Chao Lin, "GraphRC: Accelerating Graph Processing on Dual-addressing Memory with Vertex Merging," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), San Diego, California, USA, Oct. 30 - Nov. 3, 2022. (Acceptance rate: 22.5%(132/586)) (Top Conference)

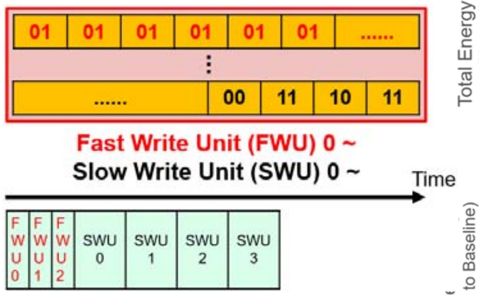
Drift-tolerant Coding to Enhance the Energy Efficiency of MLC PCM

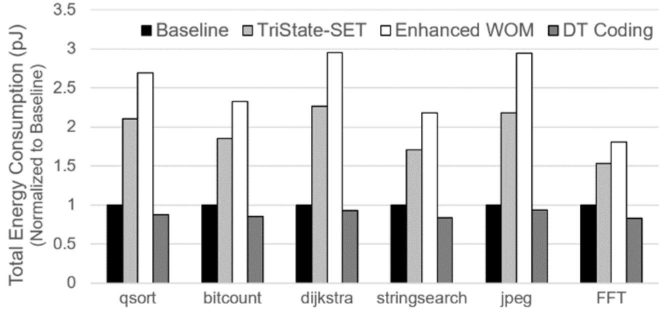
(ISLPED 2022)

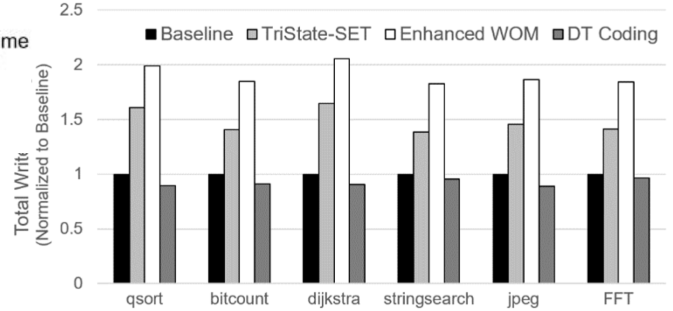
- Resistance drift is a critical challenge of PCM. It may become more severe in MLC PCM because more states are represented by a cell.
- We design a drift-tolerant coding to improve the energy efficiency and resolve the resistance drift errors of MLC PCM without sacrificing the data accuracy
  - Drift-tolerant Code: remap 3-bit datawords into 4-bit drift-tolerant codes to tolerate the resistance drift errors of MLC PCM when applies Partial-SET to accelerate PCM writes
  - Write Process Segmentation: divide the write process into fast and slow write units to improve the write performance of drift-tolerant coding



Data	DT Codes		Drifted Codes
000	0001	"01" drifts to "00"	0000
001	0101		0100
010	1001		1000
011	1101		1100
100	1111	No resistance drift	-- --
101	1110		-- --
110	0110		0010
111	0111		0011







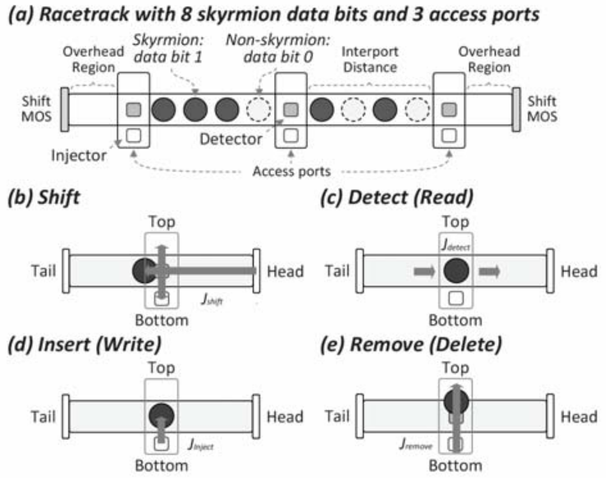
- Extensive results show that the drift-tolerant coding reduces 6.2─17.1% energy consumption and 3.2─11.3% on MLC PCM

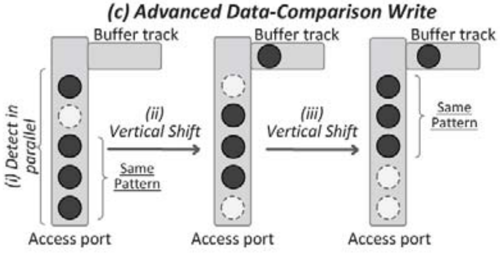
- Yi-Shen Chen, Yuan-Hao Chang, and Tei-Wei Kuo, "Drift-tolerant Coding to Enhance the Energy Efficiency of Multi-Level-Cell Phase-Change Memory" ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA, Aug. 1-3, 2022. **(Top Conference)**

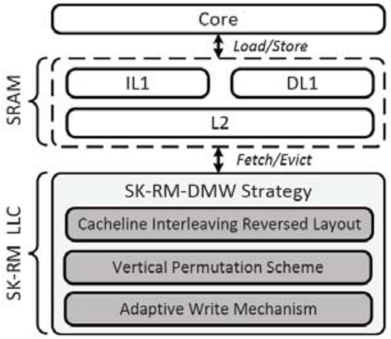
Evolving Skyrmion Racetrack Memory as Energy-Efficient Last-Level Cache Devices

(ISLPED 2022)

- Observation:
  - Skyrmion racetrack memory (SK-RM) stores data bits on the racetrack via skyrmions and relies on shift operations to read/write data bits.
  - However, shift operations lead to unpredictable data access performance or excessive energy consumption
- Goal:
  - Drawing benefits from both the bit-interleaved and word-based mapping approaches
    - Word-based mapping: Lower energy, but higher latency
    - Bit-interleaved mapping: Lower latency, but higher energy
- Main idea:
  - Dual Write Mode (DWM) strategy
    - Combine data write method of bit-interleaved and word-based mapping approaches
    - Utilize the feature of shifting skyrmions across racetracks
    - Include buffer tracks at each access ports







22.89% & 44.62% reduction in latency and energy consumption.

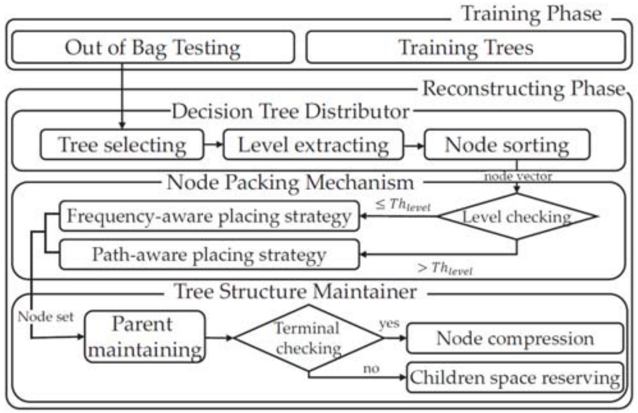
- Ya-Hui Yang, Shuo-Han Chen, and Yuan-Hao Chang, "Evolving Skyrmion Racetrack Memory as Energy-Efficient Last-Level Cache Devices" ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA, Aug. 1-3, 2022. (Top Conference)

3. Machine Learning Techniques with NVM

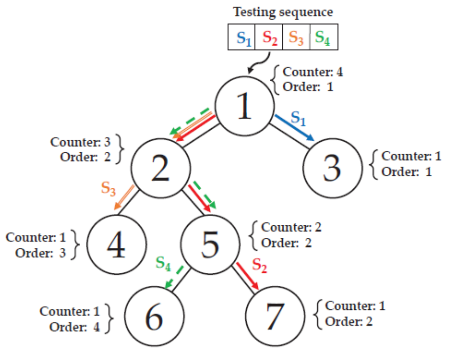
Minimizing the Read Latency of Flash Memory to Preserve Inter-tree Locality in Random Forest

(ICCAD 2022)

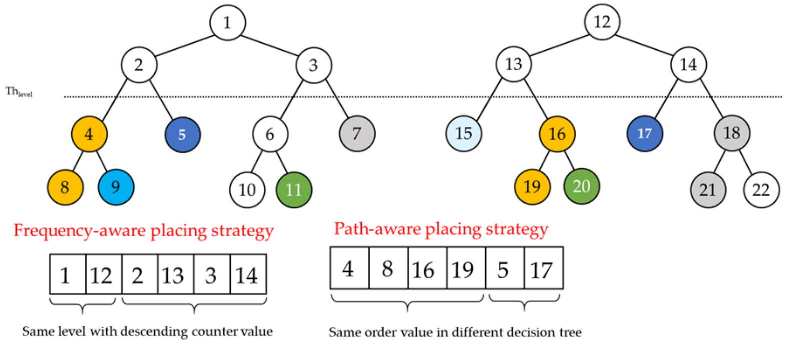
- Motivation:
  - Data move between the limited main memory space and the flash storage frequently while inferencing in memory-limited embedded system.
- Goal:
  - Reduce the swapping operations between DRAM and flash memory during inference
- Design philosophy
  - Reconstruct the trained RF model after training phase to pack the to-be-used-together nodes together in flash pages.
    - Considering both intra- inter- locality to reconstruct the trained RF model
    - Collecting the locality information in normal training process (OOB testing)



Overview of LaRF



Collecting locality information


1
2
3
12
13
14
Th_{level}
4
5
6
7
15
16
17
18
8
9
10
11
19
20
21
22
Frequency-aware placing strategy
Path-aware placing strategy
1 12 2 13 3 14
4 8 16 19 5 17
Same level with descending counter value
Same order value in different decision tree

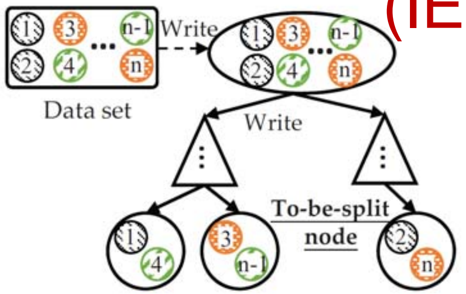
Node packing mechanism

- Yu-Cheng Lin, Yu-Pei Liang, Tseng-Yi Chen, Yuan-Hao Chang, Shuo-Han Chen, and Wei-Kuan Shih, "On Minimizing the Read Latency of Flash Memory to Preserve Inter-tree Locality in Random Forest," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), San Diego, California, USA, Oct. 30 - Nov. 3, 2022. (Acceptance rate: 22.5%(132/586)) (Top Conference)

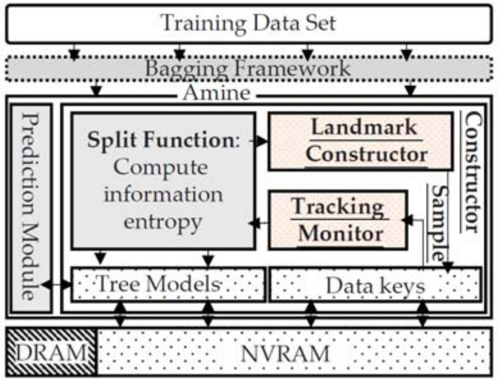
Planting Fast-growing Forest with NVM (AMINE)

(IEEE TCAD’22)

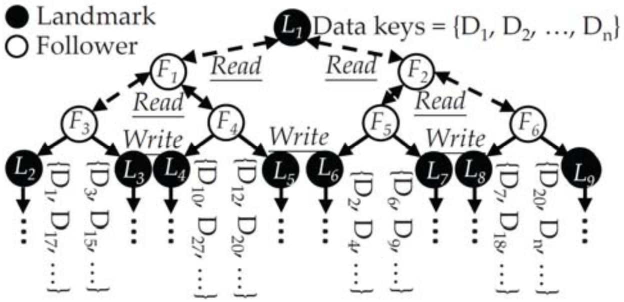
- Motivation:
  - Heavy write traffic in random forest
  - Write traffic may hurt both the performance and lifetime of NVRAM



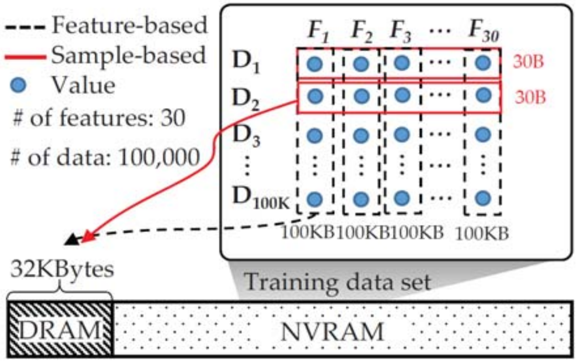
- Goal:
  - Enable a faster random forest constructing method, and prolong the lifespan of NVRAM
- Design philosophy
  - Replace partial writing operations by reading operation while construction a RF model
    - Only keep the data index in landmark nodes, and the follower nodes read their data by the index stored in their previous landmark node.
  - Change the manner to read data from dataset to limited DRAM
    - Using sample-based access manner instead of feature-based access manner



Overview of Amine.



The design philosophy of Amine.



The difference between sample-based and feature-based readers.

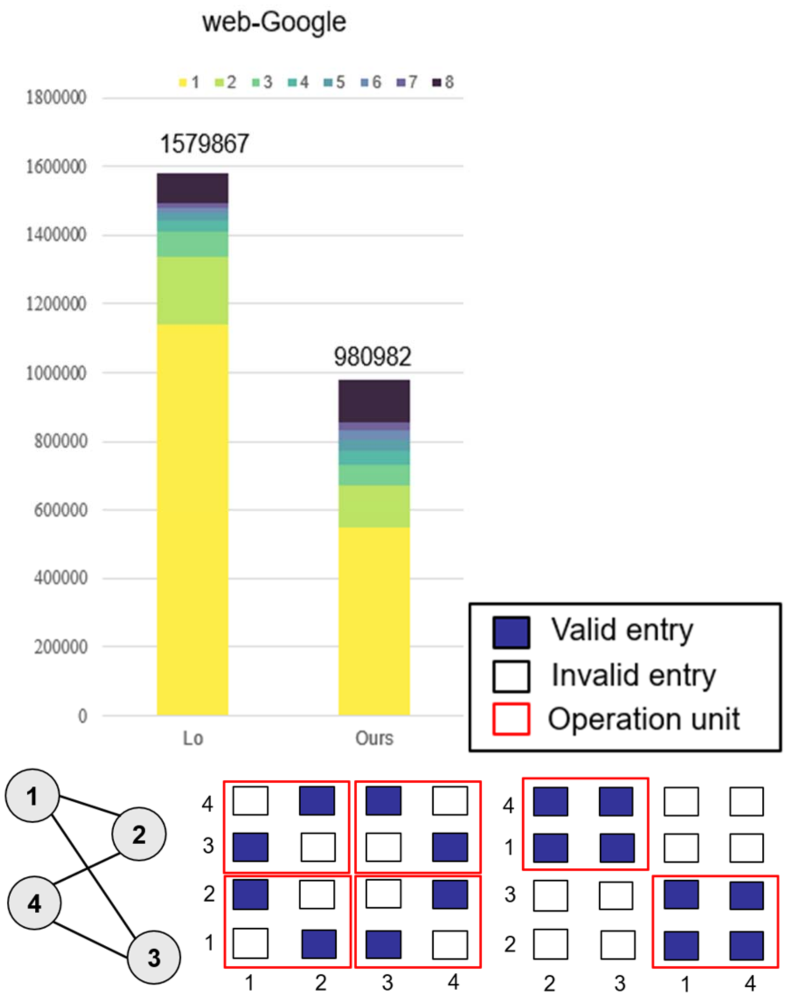
- Yu-Pei Liang, Tseng-Yi Chen, Yuan-Hao Chang, Yi-Da Huang, and Wei-Kuan Shih, "Planting Fast-growing Forest by Leveraging the Asymmetric Read/Write Latency of NVRAM-based Systems," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 41, no. 10, pp. 3304-3317, Oct. 2022.

4. In/Near-Memory Processing with NVM

SGIRR: Sparse Graph Index Remapping for ReRAM Crossbar Operation Unit

(ICCAD 2022)

- Observation
  - Placing an adjacency matrix on the ReRAM crossbar array with operation units for accelerating matrix multiplication may lead to unnecessary operation units cost and undesirable energy dissipation.
  - Reason: Most the real-world graphs are too sparse and discrete for the ReRAM crossbar array to utilize effectively, suffering from sparse matrix-vector multiplication (SpMV) problem.
- Goal
  - Design an effective index remapping algorithm to minimize the total number of operation units in ReRAM crossbar arrays and energy consumption.
- Contribution
  - We propose a two-stage spatial-aware algorithm and an operation-unit-aware column filtering approach to derive the graph order permutation, achieving better crossbar operation unit usage and energy consumption than the existing work.



(a) Original graph (b) Cost 4 operation units (c) Cost 2 operation units

- Cheng-Yuan Wang, Yao-Wen Chang, and Yuan-Hao Chang, "SGIRR: Sparse Graph Index Remapping for ReRAM Crossbar Operation Unit and Power Optimization," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), San Diego, California, USA, Oct. 30 - Nov. 3, 2022. (Acceptance rate: 22.5%(132/586)) (Top Conference)

512Gb In-Memory-Computing 3D NAND Flash Supporting Similar Vector Matching Operations on AI Edge Devices

(ISSCC 2022)

- Observation
  - Existing vector similarity search (VSS) on edge devices is not inefficient
    - Long search latency and large search energy due to large invalid data movement
  - Exploiting 3D NAND with in-memory computing (IMC) for VSS will face two major challenges:
    - A low-readout accuracy by using the wide range Vt-level of cells
    - The large-readout power consumption for the possible data-patterns.
- Goal: Enable 3D NAND-based IMC for similar vector matching to boost the VSS performance
- Main Idea

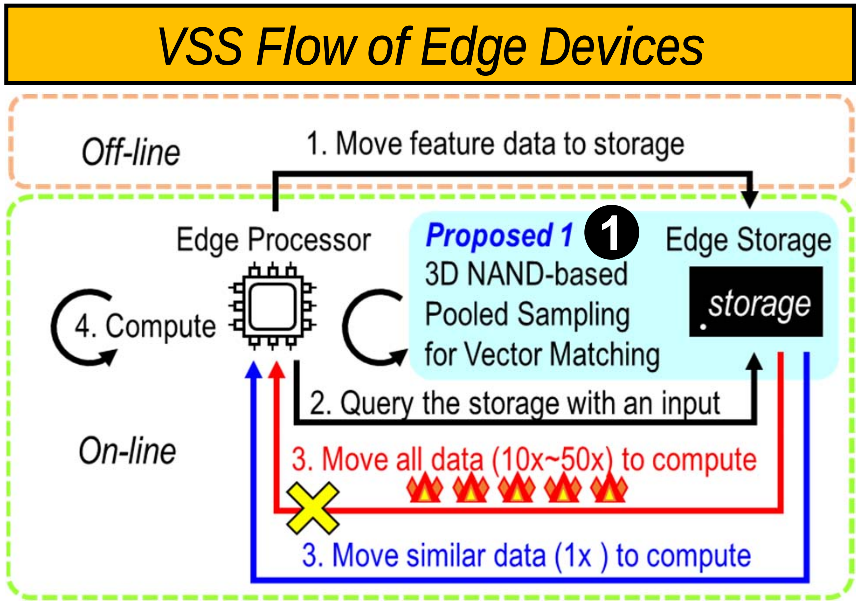
❶ Adopted "pool sampling" as the major search algorithm

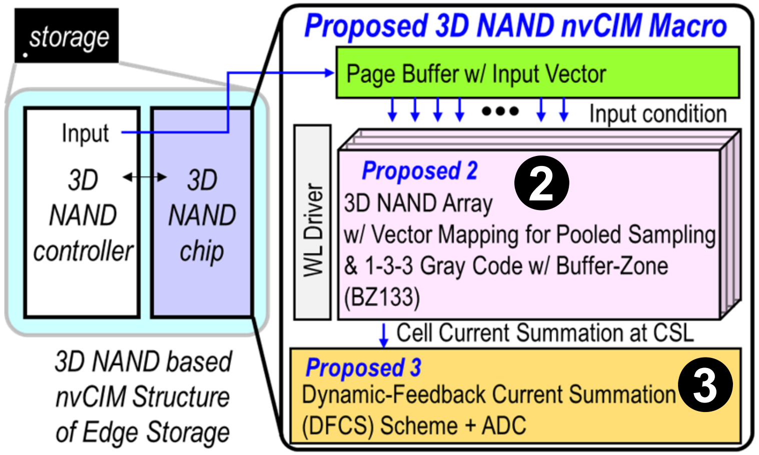
- $\vec{V}_{INPUT}\cdot\vec{V}_{INDEX_0}+\cdots+\vec{V}_{INPUT}\cdot\vec{V}_{INDEX_K}$ $=\vec{V}_{INPUT}\cdot(\vec{V}_{INDEX\_0}+\cdots+\vec{V}_{INDEX\_K})$

– Reuse the selective-BL read function on page buffer with unary data format [HTLue'19:IEDM]

❷ A 1-3-3 Gray code with buffer zone (BZ133) for TLC cells, guarding against a low readout accuracy for VVM operation

❸ Dynamic-feedback-based current-summation (DFCS) scheme to guard against the wide summation current range of VVM operations.



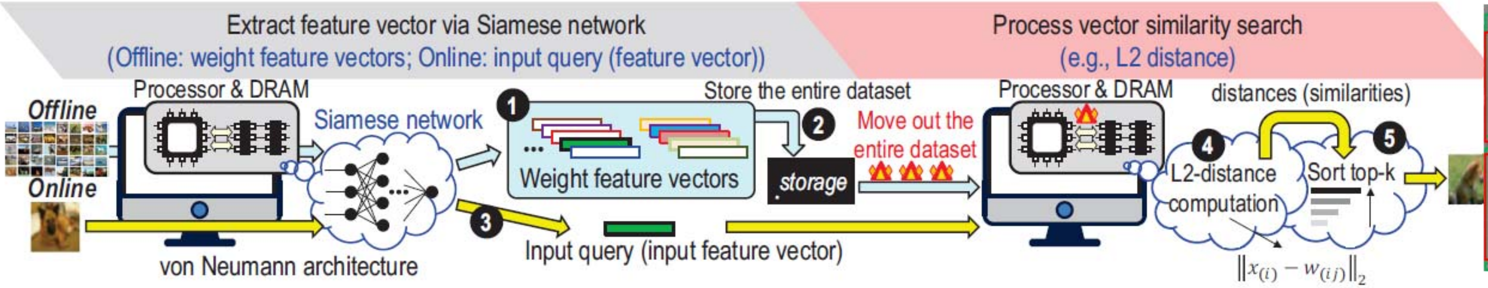

storage
Input
3D
NAND
controller
3D
NAND
chip
3D NAND based
nvCIM Structure
of Edge Storage
Proposed 3D NAND nvCIM Macro
Page Buffer w/ Input Vector
Input condition
WL Driver
Proposed 2 ❷
3D NAND Array
w/ Vector Mapping for Pooled Sampling
& 1-3-3 Gray Code w/ Buffer-Zone
(BZ133)
Cell Current Summation at CSL
Proposed 3 ❸
Dynamic-Feedback Current Summation
(DFCS) Scheme + ADC

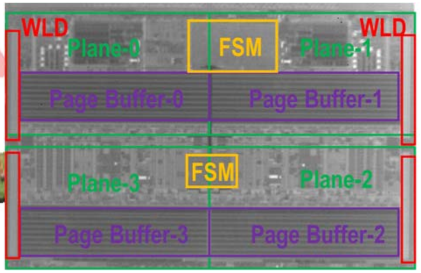
- Han Wen Hu, Wei-Chen Wang, Chung-Kuang Chen, Yung-Chun Lee, Bo-Rong Lin, Huai-Mu Wang, Yen-Bo Lin, Yu-Chao Lin, Chih-Chang Hsieh, Chia-Ming Hu, Yi-Ting Lai, Yuan-Hao Chang, Hsiang-Pang Li, Han-Sung Chen, Tei-Wei Kuo, Keh-Chung Wang, Meng-Fan Chang, Chun-Hsiung Hung, and Chih-Yuan Lu, "A 512Gb In-Memory-Computing 3D NAND Flash Supporting Similar Vector Matching Operations on AI Edge Devices," IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, California, USA, Feb. 20-24, 2022. (Top Conference)

ICE: An Intelligent Cognition Engine with NAND Processing-in-Memory for Vector Similarity Search

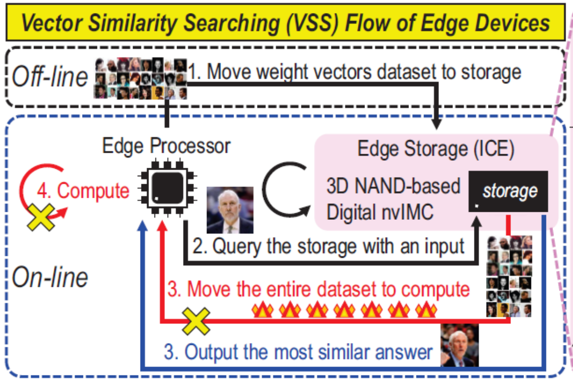
(MICRO 2022)

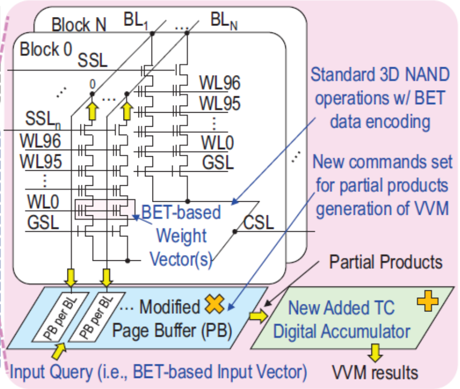
- Observation
  - Existing vector similarity search (VSS) on edge devices is not inefficient
    - Long search latency and large search energy due to large invalid data movement
  - Exploiting 3D NAND with nonvolatile in-memory computing (nvIMC) for VSS will face two major challenges:
    - Digital-based solution: ECC is critical to the nvIMC design for VSS applications since it guarantees data reliability
    - Analog-based solution: numerous ADCs and DACs increases the chip size


Extract feature vector via Siamese network
(Offline: weight feature vectors; Online: input query (feature vector))
Process vector similarity search
(e.g., L2 distance)
Offline
Online
Processor & DRAM
Siamese network
1
Weight feature vectors
Store the entire dataset
2
storage
Move out the entire dataset
Processor & DRAM
distances (similarities)
4
L2-distance computation
Sort top-k
5
3
Input query (input feature vector)
von Neumann architecture
$\|x_{(i)} - w_{(ij)}\|_2$



- Goal: Enable 3D NAND-based digital nvIMC to accelerate the VSS applications
- Main Idea
  - Exploit bit-error tolerant data encoding to mitigate the bit-error influence
  - Adopt modified page buffer to achieve single bit multiplication after computation unfolding
  - Add a new two's complement accumulator to achieve sign-bit computations in accumulation state
  - Propose a hierarchical top-n search to filter invalid data and output the most similar answer during conducting VSS applications


Vector Similarity Searching (VSS) Flow of Edge Devices
Off-line
1. Move weight vectors dataset to storage
Edge Processor
Edge Storage (ICE)
4. Compute
3D NAND-based Digital nvIMC
storage
2. Query the storage with an input
On-line
3. Move the entire dataset to compute
3. Output the most similar answer


Block N
BL_1
BL_N
Block 0
SSL
SSL_n
WL96
WL95
WL0
GSL
WL96
WL95
WL0
GSL
BET-based Weight Vector(s)
CSL
Standard 3D NAND operations w/ BET data encoding
New commands set for partial products generation of VVM
Partial Products
PB per BL
PB per BL
Modified Page Buffer (PB)
New Added TC Digital Accumulator
Input Query (i.e., BET-based Input Vector)
VVM results

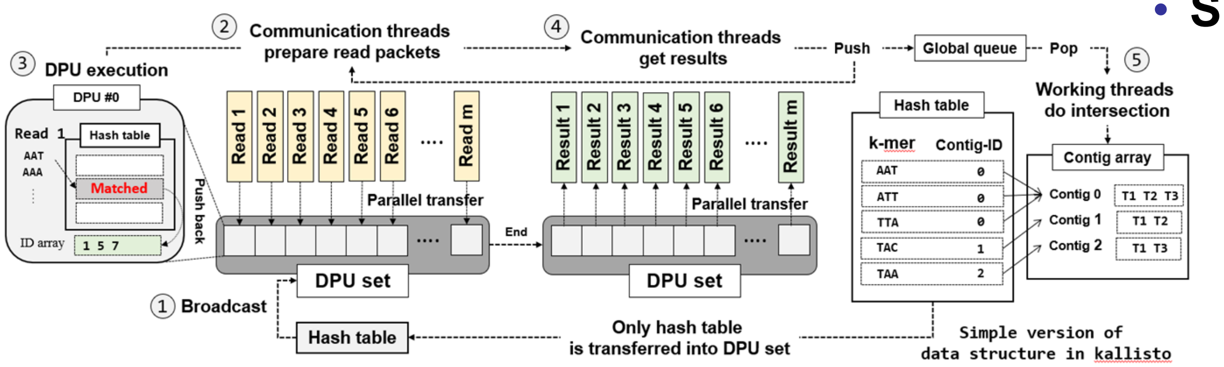
- Han-Wen Hu, Wei-Chen Wang, Yuan-Hao Chang, Yung-Chun Lee, Bo-Rong Lin, Huai-Mu Wang, Yen-Po Lin, Yu-Ming Huang, Chong-Ying Lee, Tzu-Hsiang Su, Chih-Chang Hsieh, Chia-Ming Hu, Yi-Ting Lai, Chung-Kuang Chen, Han-Sung Chen, Hsiang-Pang Li, Tei-Wei Kuo, Meng-Fan Chang, Keh-Chung Wang, Chun-Hsiung Hung, and Chih-Yuan Lu, "ICE: An Intelligent Cognition Engine with 3D NAND-based In-Memory Computing for Vector Similarity Search Acceleration," ACM/IEEE International Symposium on Microarchitecture (MICRO), Chiago, Illinois, USA, Oct. 1-5, 2022. (Top Conference)

RNA-seq Quantification on Processing in memory Architecture (DPU) (NVMSA 2022)

- **Goal** : We choose RNA-seq quantification to be a case study, *understanding the characteristics of sequencing on UPMEM DPU*.

- **Concerns**
  - DPU constrains: e.g. no data sharing between DPUs
  - Frequent data movement between CPU and DPU
  - DPU-based software design

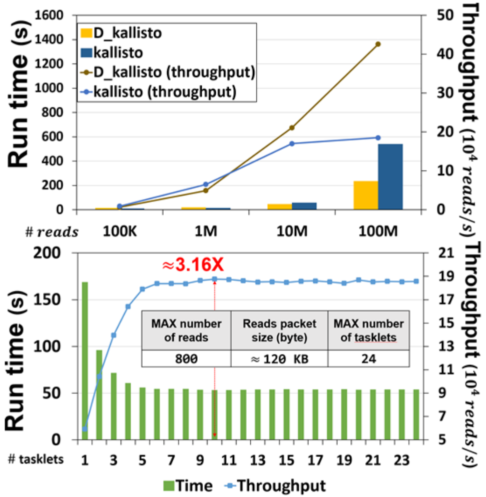
- **Implementation**
  1. Host CPU broadcasts the hash table to all DPUs
  2. Communication threads send read packets to DPU
  3. Host CPU launches the DPU program
  4. Communication threads get result back to host
  5. Working thread will get a ID from queue and do intersection.



k-mer	Contig-ID
AAT	0
ATT	0
TTA	0
TAC	1
TAA	2

Contig array	
Contig 0	T1 T2 T3
Contig 1	T1 T2
Contig 2	T1 T3

- **Implementation Result**



MAX number of reads	Reads packet size (byte)	MAX number of tasklets
800	≈ 120 KB	24

High throughput in large data size

High efficiency DPU program

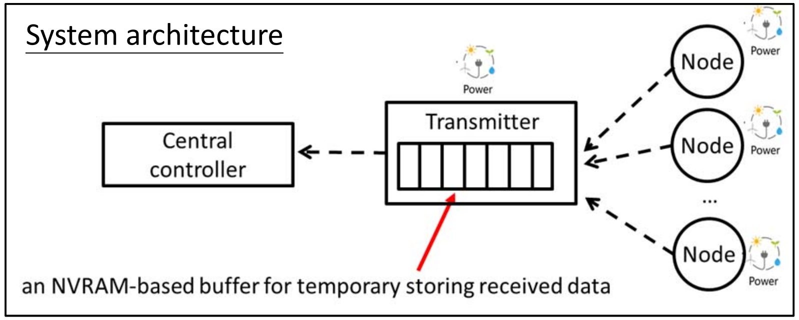
- **Suggestions / Observations**
  - **UPMEM DPU system is a material *suitable for large data size*.**
  - ***Frequent data transfers is an overhead* in DPU system.**
  - ***WRAM is expensive*, we have to design the data flow carefully during DPU-based programming.**

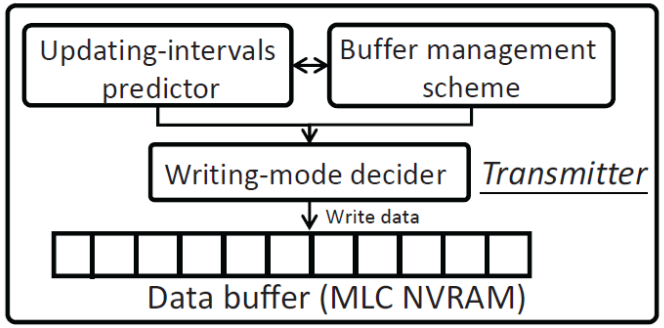
- Liang-Chi Chen, Shu-Qi Yu, Chien-Chung Ho, Yuan-Hao Chang, Da-Wei Chang, Wei-Chen Wang, and Yu-Ming Chang, "RNA-seq Quantification on Processing in memory Architecture: Observation and Characterization," IEEE Nonvolatile Memory Systems and Applications Symposium (NVMSA), Taipei, Taiwan, Aug. 23-25, 2022.

5. Intermittent Systems

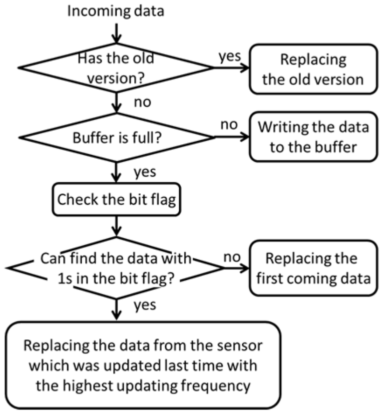
Minimizing Age-of-Information of NVRAM-based Intermittent Systems (NVMSA 2022)

- Motivation:
  - Traditional replacement policy cannot minimize the AoI
  - Even adopting the multi-level write, the energy still may be waste without considering the buffer replacement policy
- Goal:
  - Minimize the average AoI of the generated data of the sensor nodes.
  - Design a buffer-replacement-policy-aware writing-mode decider
- Main ideas
  - Predict the node behavior
    - Approximate average the updating interval
  - Balance the update frequency of the sensor nodes to prevent the soaring of AoI.
    - 1.Duplication Check - 2. Updating-Frequency Balance
  - Use the proper writing mode to write the data to NVRAM for further reducing the energy consumption
    - using light write if the data frequently update and tends to be replaced





Overview of the proposed method

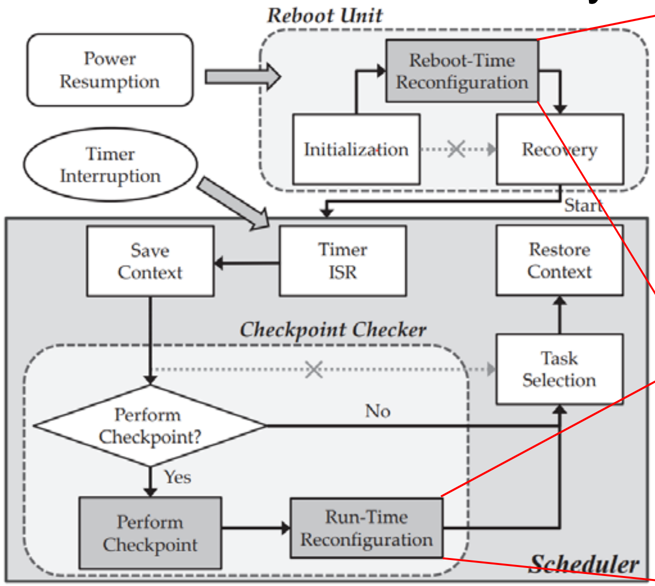

Incoming data
Has the old version?
yes
Replacing the old version
no
Buffer is full?
no
Writing the data to the buffer
yes
Check the bit flag
Can find the data with 1s in the bit flag?
no
Replacing the first coming data
yes
Replacing the data from the sensor which was updated last time with the highest updating frequency

Buffer management scheme

- Hung-Yu Lin, Yu-Pei Liang, Shuo-Han Chen, Yuan-Hao Chang, Tseng-Yi Chen, and Wei-Kuan Shih, "Minimizing Age-of-Information of NVRAM-based Intermittent Systems," IEEE Nonvolatile Memory Systems and Applications Symposium (NVMSA), Taipei, Taiwan, Aug. 23-25, 2022.

SACS: A Self-Adaptive Checkpointing Strategy for Microkernel-Based Intermittent Systems

- In this work, we propose a new self-adaptive checkpointing strategy for improving the performance of microkernel-based intermittent systems. (ISLPED'22, Best Paper Nomination)
  - By observing the number of performed context switches in each run time, our design adaptively adjusts the checkpointing interval to achieve a good balance between the execution progress (performance) and the number of performed checkpoints.
  - At runtime, we design a checkpoint checker that (1) determines the necessity of performing checkpoints and (2) conservatively enlarges the checkpointing interval.
  - At reboot time, we design a reboot-time reconfiguration procedure that approaches the suitable checkpointing interval according to the execution status learned from previous run times.
- Compared to a state-of-the-art design called ELASTIN, our approach could reduce the execution time by 50.6% under unstable harvesting condition.



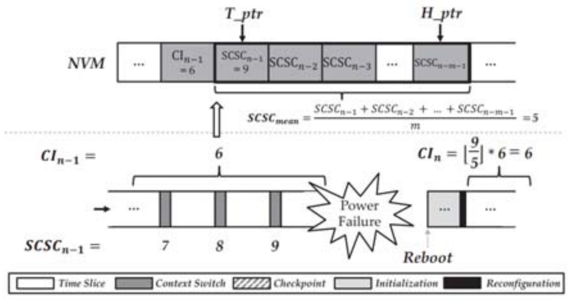


Fig. 5: Example of Reboot-Time Reconfiguration

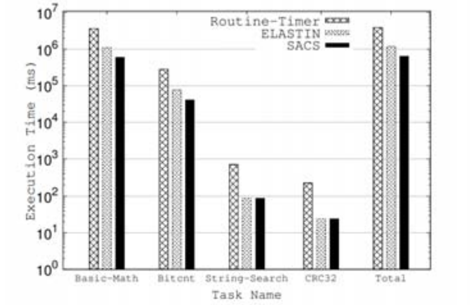
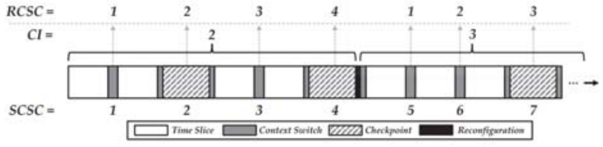


Fig. 7: Execution Time Under Stable Harvesting Condition



$$CI_{new} = \begin{cases} \lfloor\frac{CI_{old}}{2}\rfloor & \text{if } SCSC_{old} < CI_{old} \\ RTF * CI_{old} & \text{if } SCSC_{old} \geq CI_{old} \end{cases} \quad (1)$$

$$RunTimeFactor(RTF) = \lfloor\frac{SCSC_{old}}{SCSC_{mean}}\rfloor \quad (2)$$

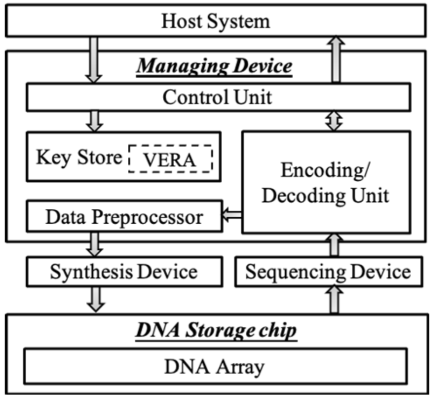
Fig. 4: Example of Run-Time Reconfiguration

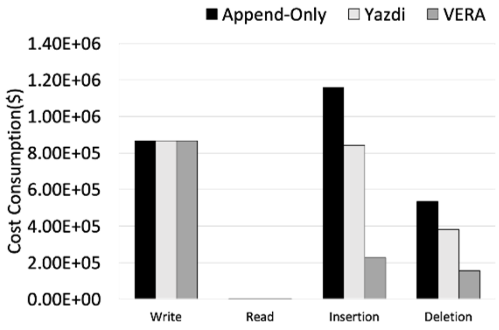
- Yen-Ting Chen, Han-Xiang Liu, Yuan-Hao Chang, Yu-Pei Liang, and Wei-Kuan Shih, "SACS: A Self-Adaptive Checkpointing Strategy for Microkernel-Based Intermittent Systems" ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA, Aug. 1-3, 2022. (Best Paper Nomination - Top Conference)

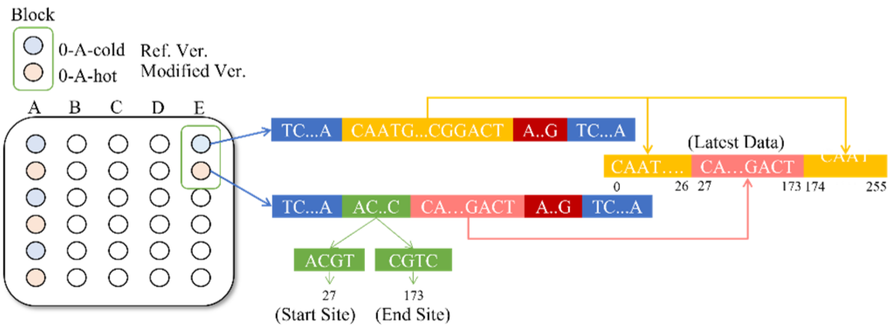
6. Others

Write-Cost Reduction for DNA Storage

- Observation (ACM TECS in 2022)
  - Accessing digital data over DNA requires a series of extremely time-consuming processes, and the rewriting cost is the predominant cost of DNA data storage system.
  - DNA-based storage systems are usually lacking management.
- This work is a pioneer in implementing a management scheme for the DNA-based system and further reducing unnecessary laboratory operations. (ACM TECS in 2022)
  - This design advocates a complete management scheme by comparing the latest data and incoming data to know the updated part of the data with minimized reading overhead.
  - The idea is to manage the version when the number of versions exceeds the bits provided by the primer.
  - lower the total writing bits in each updated operation (i.e., insertion and deletion).
- Compared with the baseline solution, VERA can reduce the writing cost of the DNA storage system by 77%.







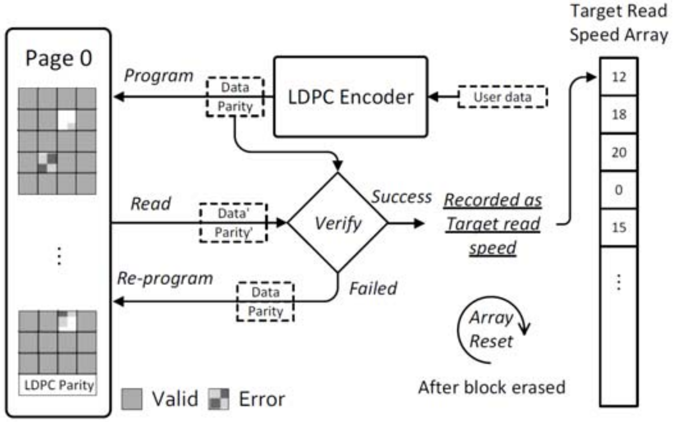
- Yi-Syuan Lin, Yu-Pei Liang, Tseng-Yi Chen, Yuan-Hao Chang, Shuo-Han Chen, Hsin-Wen Wei, and Wei-Kuan Shih, "How to Enable Index Scheme for Reducing the Writing Cost of DNA Storage on Insertion and Deletion," ACM Transactions on Embedded Computing Systems (TECS), vol. 21, no. 3, pp. 30:1-30:25, May 2022.

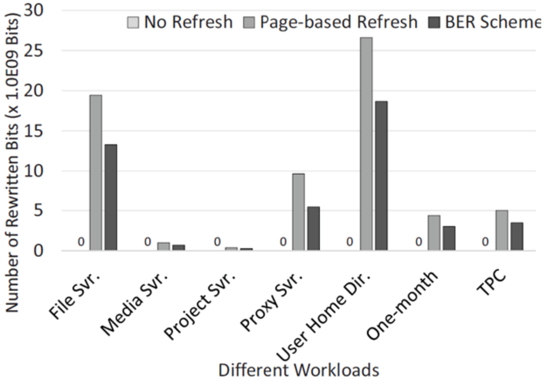
Research Summary 2021

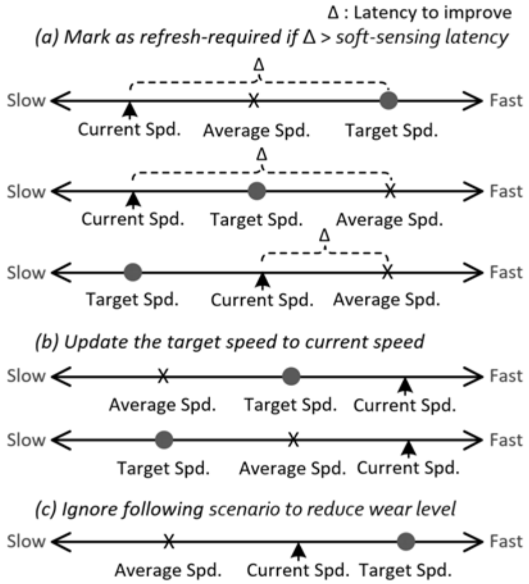
1. Storage Systems -
Flash Drives and SMR Disks

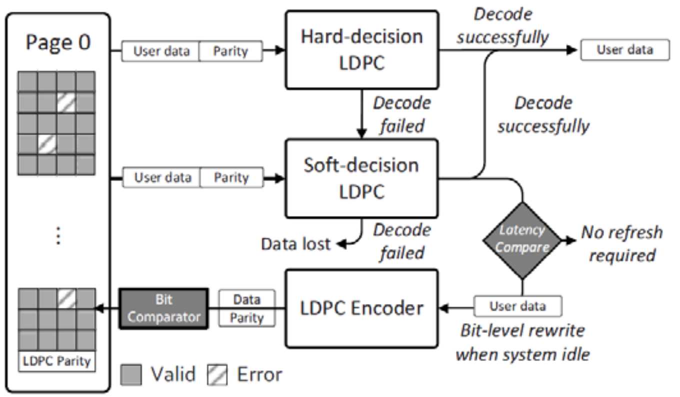
Lifetime & Read Optimization for Bit-alterable Flash (IEEE TCAD in 2021, DAC 2019)

- Observation
  - Bit-alterable NAND flash has become a reality to offer the possibility of removing error bits without page-based rewrites.
  - However, bit-level rewrites has the similar latency as page-based rewrites
- Goal
  - Maximize the read performance with minimal lifetime degradation.
- Main Idea
  - Identify different type of bit errors
    - *Read speed tracking mechanism*
  - Avoid unnecessary rewrites without lowering throughput
    - *Sensing latency reduction strategy*









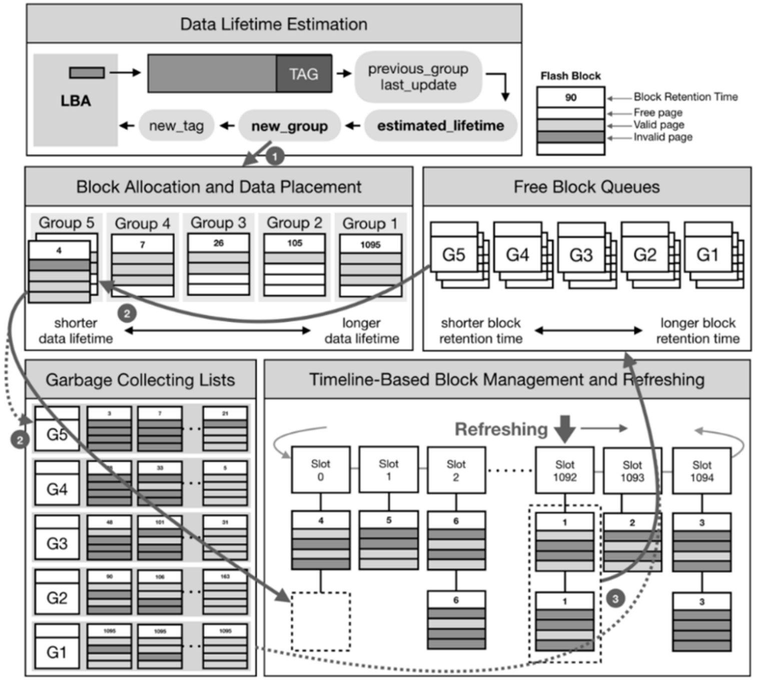
Rewritten Bits Reduction: 40.39%

Read Speed Improvement: 25.22%

- Shuo-Han Chen, Ming-Chang Yang, and Yuan-Hao Chang, "Optimizing Lifetime Capacity and Read Performance of Bit-Alterable 3D NAND Flash," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 40, no. 2, pp. 218-231, Feb. 2021.
- Shuo-Han Chen, Ming-Chang Yang, and Yuan-Hao Chang, "The Best of Both Worlds: On Exploiting Bit-Alterable NAND Flash for Lifetime and Read Performance Optimization," ACM/IEEE Design Automation Conference (DAC), Las Vegas, Nevada, USA, Jun. 2-6, 2019. **(Top Conference)**

Harmonization for Data Lifetime and Block Retention time

- We propose a *Time Harmonization Strategy* to harmonize the “retention capability” of flash blocks with different “data lifetime requirement” of the written data (IEEE TC in 2021)
  - The mismatch of data lifetime requirement and flash block retention capability could induce additional data migration overhead.
  - Goal: (1) Estimate the data lifetime, and efficiently (2) allocate suitable flash blocks to accommodate data in accordance with the block retention capability and the estimated data lifetime.



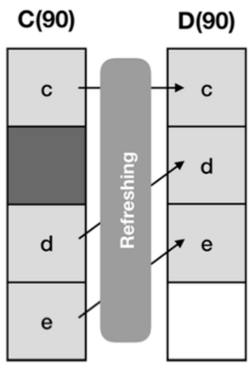


Fig. 1. Example of refreshing operations.

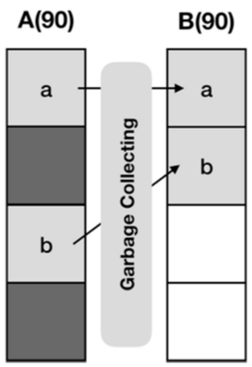


Fig. 2. Example of garbage collection operations.

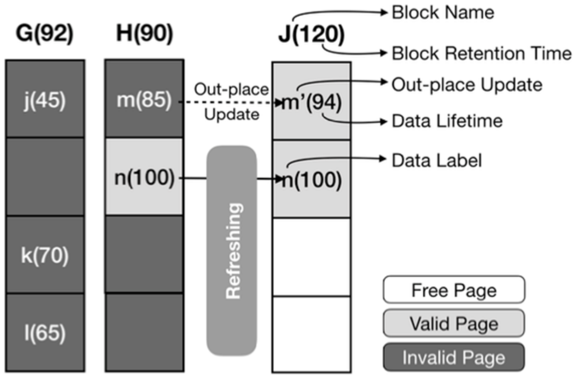


Fig. 3. Considering retention capability and data lifetime during refreshing.

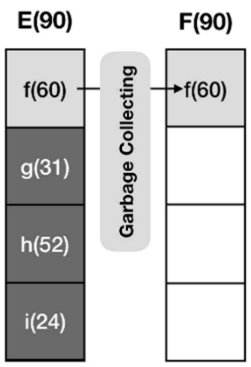


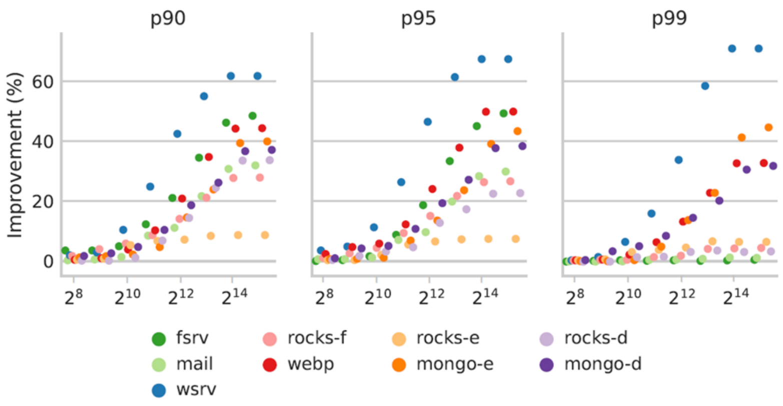
Fig. 4. Considering retention capability and data lifetime during garbage collection.

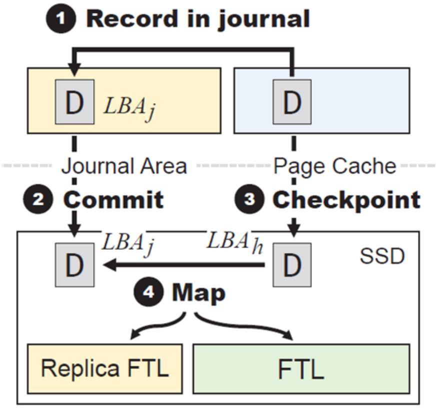
- Ming-Chang Yang, Chun-Feng Wu, Shuo-Han Chen, Yi-Ling Lin, Che-Wei Chang, and Yuan-Hao Chang, "On Minimizing Internal Data Migrations of Flash Devices via Lifetime-Retention Harmonization," IEEE Transactions on Computers (TC), vol. 70, no. 3, pp. 428-439, Mar. 2021.

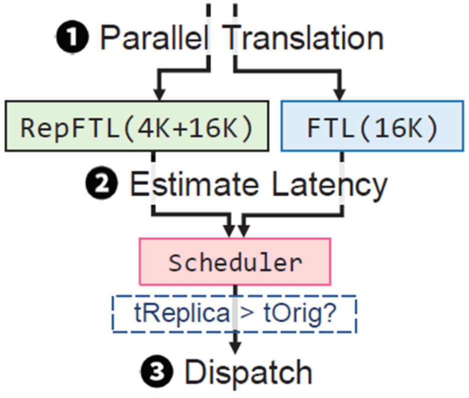
Reptail – Cutting Storage Tail Latency with Inherent Redundacncy

(DAC 2021)

- Observation
  - Increasing SSD density raises read tail latency.
  - Data-consistency protection creates data redundancy that is otherwise unused.
- Goal
  - Achieve good performance, consistency and high density simultaneously on edge storage
- Main Idea
  - Expose SSD to the data identicalness semantics
    - *Redundancy mapping & Journal defragmentation*
  - SSD has multiple replicas to fulfill a read request
    - *Low-overhead table*





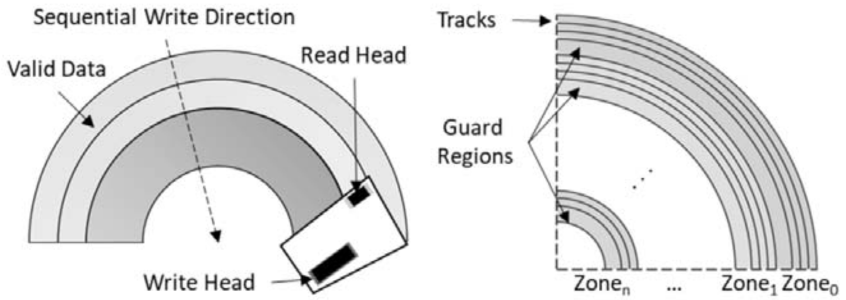


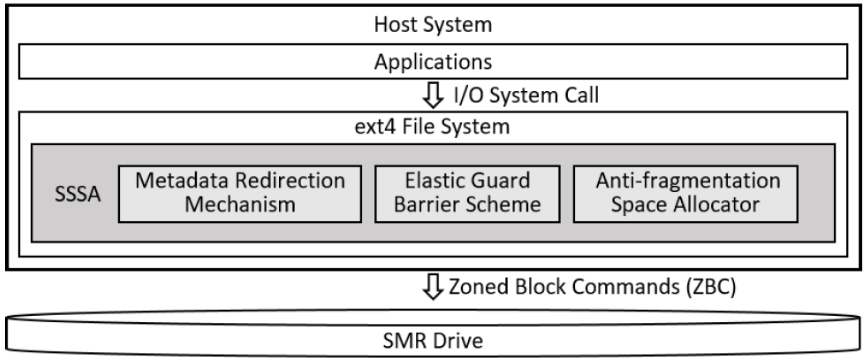
95th Read Tail Latency Reduction: 20%

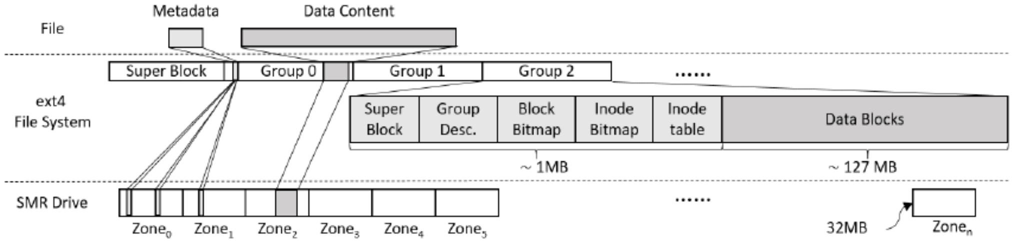
- Yun-Chih Chen, Chun-Feng Wu, Yuan-Hao Chang, and Tei-Wei Kuo, "Reptail: Cutting Storage Tail Latency with Inherent Redundancy," ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, Dec. 5-9, 2021. (Acceptance rate: 23%) (Top Conference)

Secure Deletion for EXT4 File Data and Metadata on SMR-based (ASP-DAC 2021)

- Observation
  - The efficiency of secure deletion is highly dependent on the data layout.
  - The sequential-write constraint of SMR drive hinder the efficiency of secure deletion.
  - However, the small-size nature of file system metadata aggravates the efficiency.
- Goal
  - Facilitate the process of securely erasing both the deleted files and their metadata simultaneously.
- Main Idea
  - Metadata Redirection Mechanism
    - Store file inode along with the file data
  - Elastic Guard Barrier Scheme
    - Divide SMR zones into smaller segments
  - Anti-Fragmentation Space Allocator
    - Pack small files into smallest segments




Host System
Applications
I/O System Call
ext4 File System
SSSA
Metadata Redirection Mechanism
Elastic Guard Barrier Scheme
Anti-fragmentation Space Allocator
Zoned Block Commands (ZBC)
SMR Drive



Secure Deletion Latency Reduction: 91.3%

- Ping-Xiang Chen, Shuo-Han Chen, Yuan-Hao Chang, Yu-Pei Liang, and Wei-Kuan Shih, "Facilitating the Efficiency of Secure File Data and Metadata Deletion on SMR-based Ext4 File System," ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, Jan. 18-21, 2021.

2. NVM Main Memory and Storage

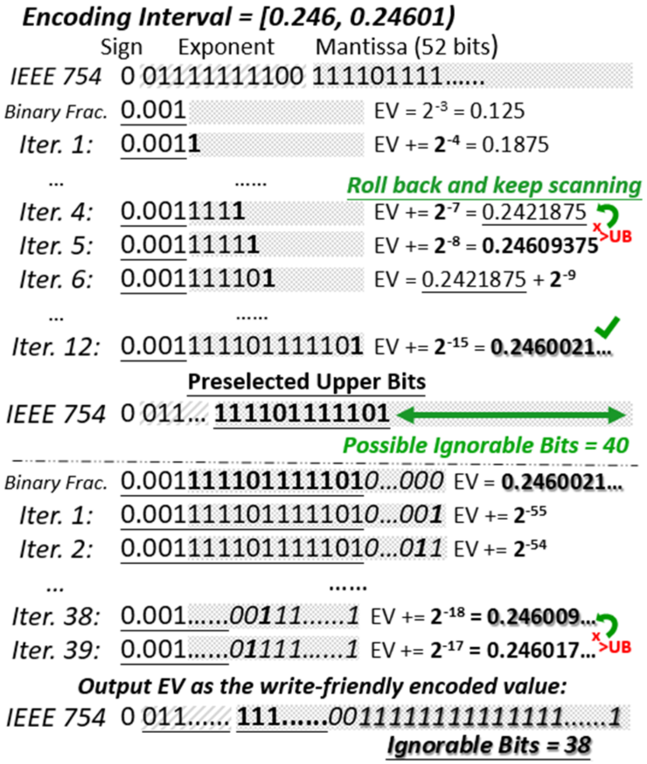
Write-friendly Arithmetic Coding for NVM

(ASP-DAC 2021)

- Observation
  - Storage-Class Memory technologies and data compression techniques can be used to alleviate the energy consumption of wearable IoT devices
  - However, the information gap between the PCM devices and data compression techniques hinders the cooperation among the two techniques for achieving further performance optimization
- Goal
  - Design an energy-aware and write-friendly arithmetic coding (AC) to improve energy efficiency of PCM
- Main Idea
  - Exploit the property of encoding interval in arithmetic coding to smartly choose an ideal encoded value consists of most ignorable bits, so as to reduce the number of write operations during the compression on PCM

    - Upper Bits Preselecting

    - Ignorable Bits Determining



Encoding Interval = [0.246, 0.24601)

Sign Exponent Mantissa (52 bits)

IEEE 754 0 01111111100 111101111......

Binary Frac. 0.001 EV = 2^{-3} = 0.125

Iter. 1: 0.001**1** EV += $\mathbf{2^{-4}}$ = 0.1875

... .

Roll back and keep scanning

Iter. 4: 0.001111**1** EV += $\mathbf{2^{-7}}$ = 0.2421875

Iter. 5: 0.0011111**1** EV += $\mathbf{2^{-8}}$ = **0.24609375** >UB

Iter. 6: 0.00111110**1** EV = 0.2421875 + $\mathbf{2^{-9}}$

... .

Iter. 12: 0.00111110111110**1** EV += $\mathbf{2^{-15}}$ = **0.2460021...** ✓

Preselected Upper Bits

IEEE 754 0 011... **111101111101**

Possible Ignorable Bits = 40

Binary Frac. 0.001**111101111101***0...000* EV = **0.2460021...**

Iter. 1: 0.001111101111101*0...00***1** EV += $\mathbf{2^{-55}}$

Iter. 2: 0.001111101111101*0...0***1***1* EV += $\mathbf{2^{-54}}$

... .

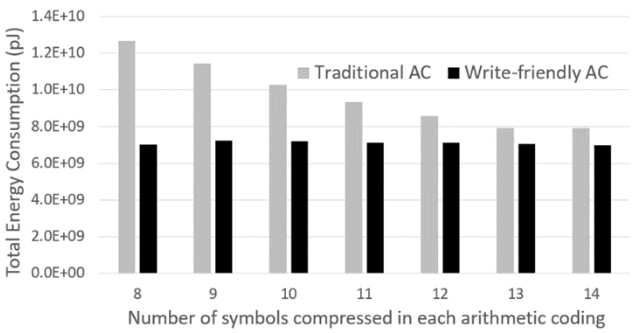
Iter. 38: 0.001......*00***1***11......1* EV += $\mathbf{2^{-18}}$ **= 0.246009...**

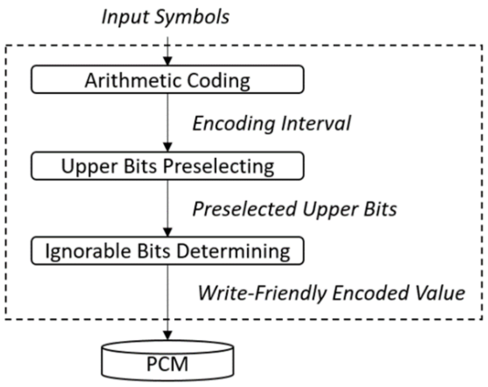
Iter. 39: 0.001......*0***1***111......1* EV += $\mathbf{2^{-17}}$ **= 0.246017...** >UB

Output EV as the write-friendly encoded value:

IEEE 754 0 011...... **111......***0011111111111111......1*

Ignorable Bits = 38





Input Symbols

Arithmetic Coding

Encoding Interval

Upper Bits Preselecting

Preselected Upper Bits

Ignorable Bits Determining

Write-Friendly Encoded Value

PCM

Reduce 10.6-44.6% energy, compared to the traditional AC

- Yi-Shen Chen, Chun-Feng Wu, Yuan-Hao Chang, and Tei-Wei Kuo, "A Write-friendly Arithmetic Coding Scheme for Achieving Energy-Efficient Non-Volatile Memory Systems," ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, Jan. 18-21, 2021.

Wear-leveling-enabled B+-tree over NVM

(IEEE TCAD in 2021)

- Observations:
  - The frequent B$^+$-tree key operations result in inter- and intra-node wear-leveling issue.
- Goal:
  - To balance the amount of *write traffic within a node space* but considers the *global wear-leveling issue over the whole NVRAM space* as well
- Solution:
  - Circular node structure:
    - *To move the start point of insertion in a tree node* for evenly distributing the amount of write traffic to each entry space within a node.
  - Global wear-leveling strategy:
    - ***A cursor*** points a candidate node for swapping in the current B$^+$-tree structure.
    - The amount of write traffic of the (to-be-allocated) node ≥ *Threshold*

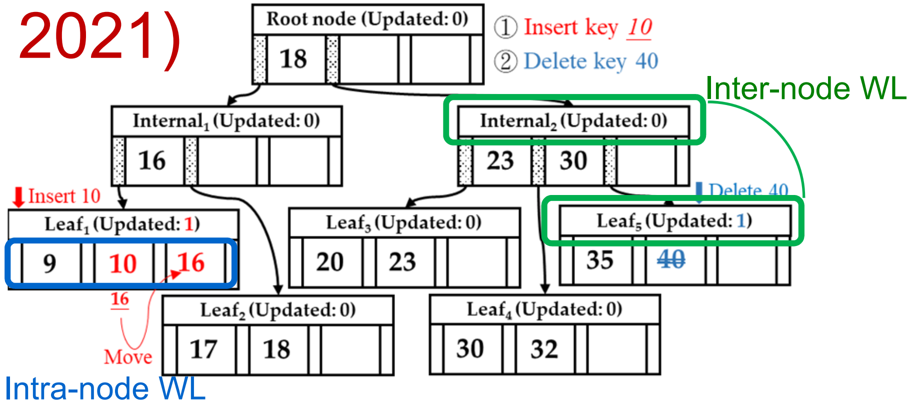
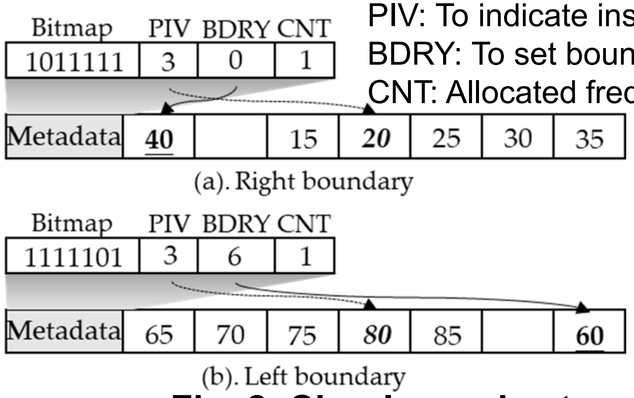


Fig. 1. Our observations



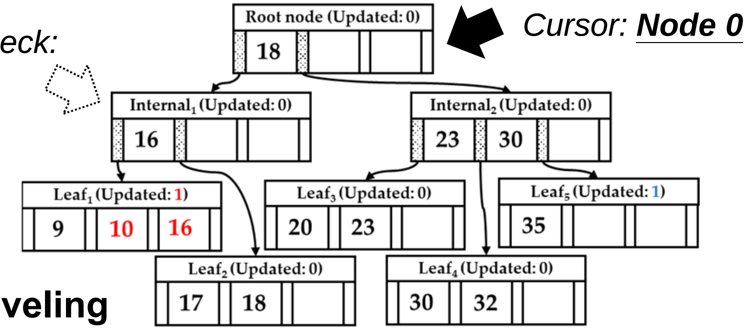
(a). Right boundary

(b). Left boundary

PIV: To indicate insertion entry
BDRY: To set boundary
CNT: Allocated frequency

Fig. 2. Circular node structure

Next check:



Cursor: ***Node 0***

Fig. 3. Global wear-leveling

- Dharamjeet, Tseng-Yi Chen, Yuan-Hao Chang, Chun-Feng Wu, Chi-Heng Lee, and Wei-Kuan Shih, "Beyond Write-reduction Consideration: A Wear-leveling-enabled B+-tree Indexing Scheme over an NVRAM-based Architecture," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 40, no. 12, pp. 2455-2466, Dec. 2021.

Write-Reduction Multiversion Scheme to Support Dual-range Query over NVM

(IEEE TVLSI in 2021)

- Observation
  - multi-version indexing scheme will rewrite duplicate information for achieving an efficient multi-versioned data query, and it will generate a massive amount of write traffics to an NVRAM space
  - all multi-version indexing schemes cannot simultaneously provide efficient key and version-range queries
- Goal
  - To propose a write-reduction multi-version indexing scheme with efficient dual-range-query operations over NVRAM-based storage
- Main Idea
  - Efficient key-range-query
    - Preserves the partial nature of the MVBT (multi-version tree), but reduces the amount of write traffic to NVRAM by a pointer-based split mechanism.
  - Efficient version-range-query
    - Aggregates the version data belonging to the same index together (version forest)

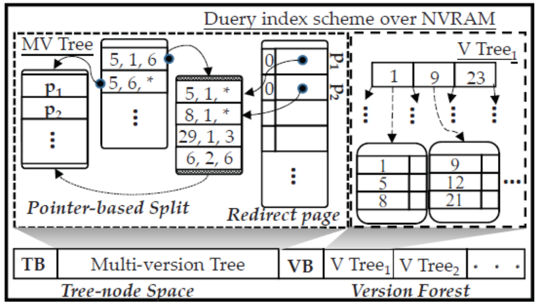


Fig. 1. Duery indexing scheme

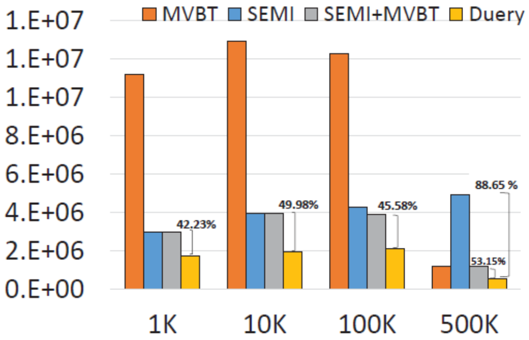


Fig. 2. Key-range performance (read bytes)

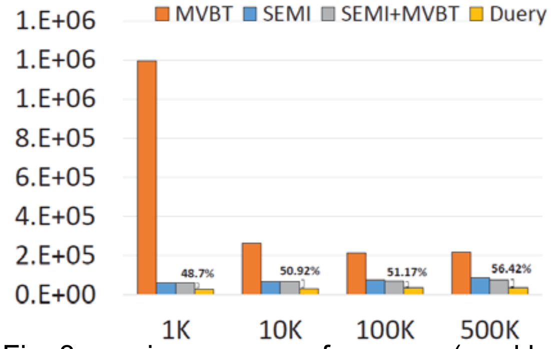
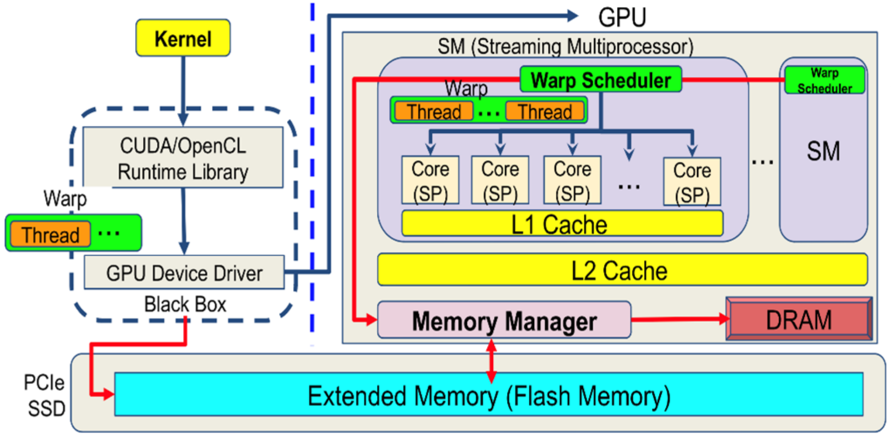


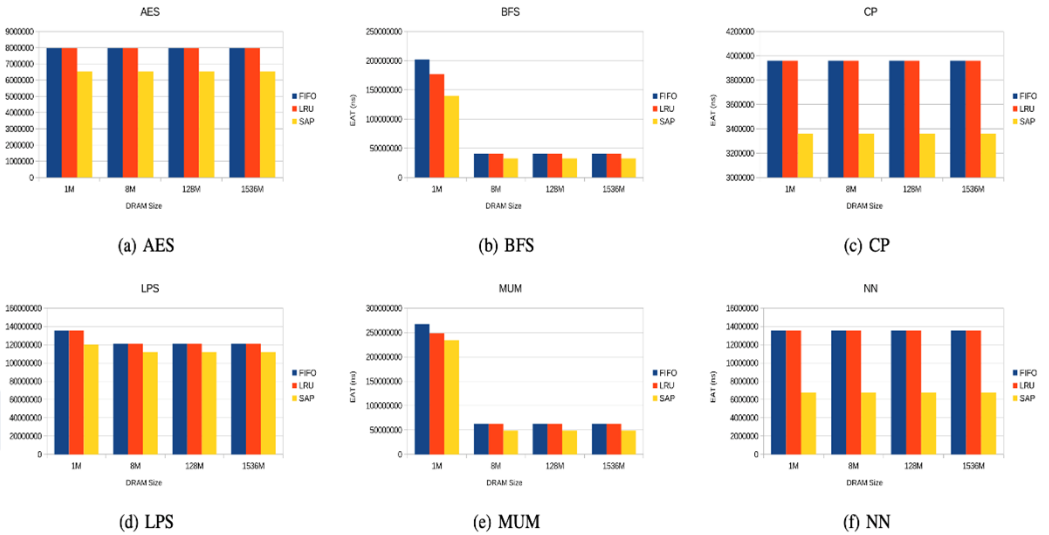
Fig. 3. version-range performance (read bytes)

- I-Ju Wang, Yu-Pei Liang, Tseng-Yi Chen, Yuan-Hao Chang, Bo-Jun Chen, Hsin-Wen Wei, and Wei-Kuan Shih, "Enabling Write-reduction Multiversion Scheme with Efficient Dual-range Query over NVRAM," IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 29, no. 6, pp. 1244-1256, Jun. 2021.

Scheduling-aware Prefetching (NVMSA 2021)

- Observation:
  - GPU tasks usually show weak temporal locality because the relation between two warps is independent.
  - Applying page replacement approaches on GPU tasks usually suffers from the fast locality saturation issue. Thus, even when larger DRAM is equipped in the GPU, the DRAM hit ratio cannot be further improved.
- Goal:
  - Aiming to provide low-cost energy-efficient GPU memory extension systems, this work proposes a scheduler-aware prefetching design to improve system performance.
- Main Idea:
  - Memory Manager fully utilizes the information of the internal hardware warp scheduler inside GPU device to perform data prefetching without changing the writing way of the GPU program.





(a) AES (b) BFS (c) CP

(d) LPS (e) MUM (f) NN

Effective Access Time by Different Cache Algorithms

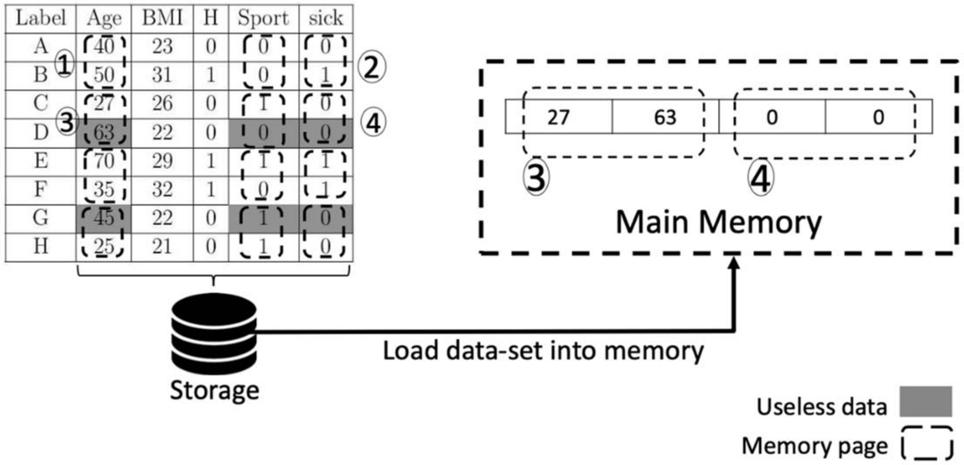
- Tse-Yuan Wang, Chun-Feng Wu, Che-Wei Tsao, Yuan-Hao Chang, and Tei-Wei Kuo, "Scheduling-Aware Prefetching: Enabling the PCIe SSD to Extent the Global Memory of GPU Device," IEEE Nonvolatile Memory Systems and Applications Symposium (NVMSA), August 2021.

3. Machine Learning Techniques and Processing-in-Memory with NVM

Random Forest I/O-aware Algorithm

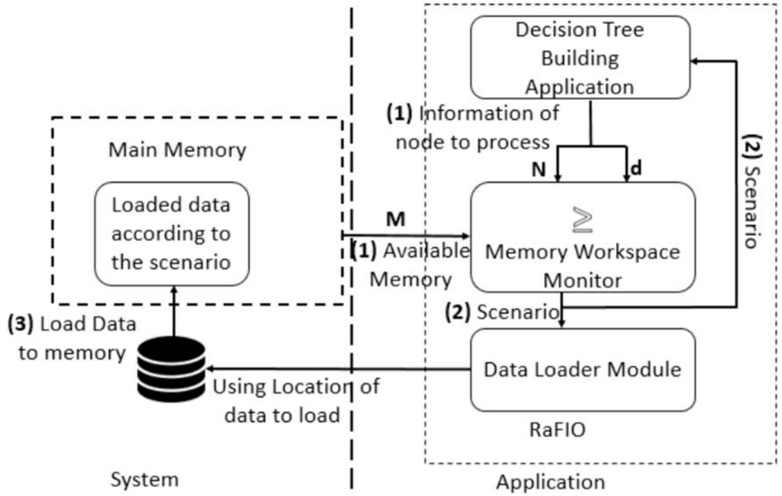
(SAC 2021)

- Observation
  - During training random forest, performance drops significantly when the dataset size is larger than the available memory size.
  - Reasons: Randomly bagging data causes unnecessary data movements.
- Goal
  - Reduce unnecessary data movement by avoiding loading useless data and smartly selecting the data according to their reuse pattern in the following tree building steps
- Main Idea
  - Decision Tree Building Module: Perform on-demand data loading according to the available memory space.
  - Data Loader Module: Pre-process data to easily locate useful data without reading them multiple times during data loading.



Label	Age	BMI	H	Sport	sick
A	40	23	0	0	0
B	50	31	1	0	1
C	27	26	0	1	0
D	63	22	0	0	0
E	70	29	1	1	1
F	35	32	1	0	1
G	45	22	0	1	0
H	25	21	0	1	0

Unnecessary Data Movements

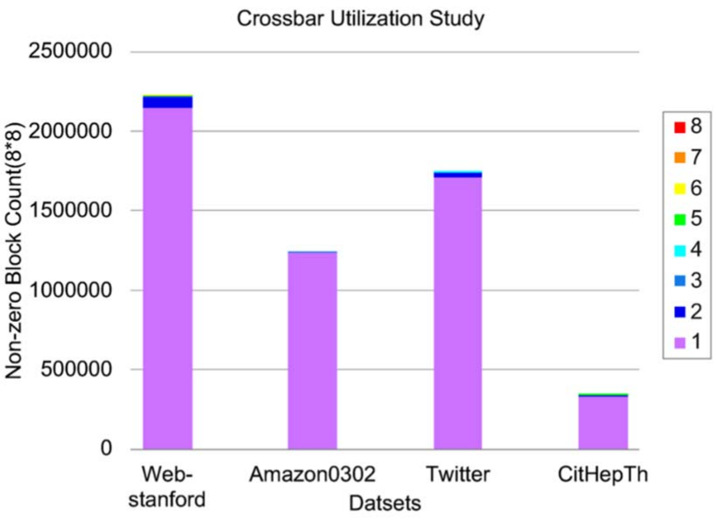


Unnecessary Data Movements

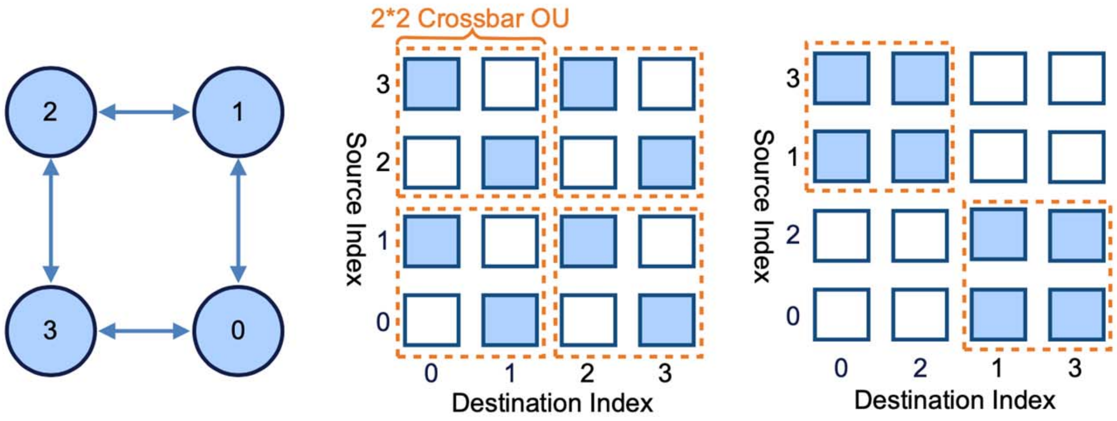
- Camelia Slimani, Chun-Feng Wu, Yuan-Hao Chang, Stephane Rubini, and Jalil Boukhobza, "RaFIO: A Random Forest I/O-Aware Algorithm," ACM Symposium on Applied Computing (SAC), Gwangju, South Korea, Mar. 22-26, 2021.

Space-efficient Graph Placement on ReRAM Crossbar (ISLPED 2021)

- Observation
  - Placing an adjacency matrix on the crossbar array for accelerating matrix multiplication may lead to unnecessary energy wasting.
  - Reason: Elements in the graph adjacency matrix are usually sparse and discrete, and thus extra crossbar OUs are required for processing because of the low-utilization.
- Goal
  - Interested in proposing a hardware/software co-design solution to solve the sparse and discrete issues by clustering graph nodes on the crossbar accelerators.
- Main Idea
  - Remap and shuffle the original adjacency matrix with being aware of the graph localities.



Low Utilization on Crossbar Accelerator



Original Graph Original Adjacency Matrix Shuffled Adjacency Matrix

Design Concept: Remapping and Reshuffling

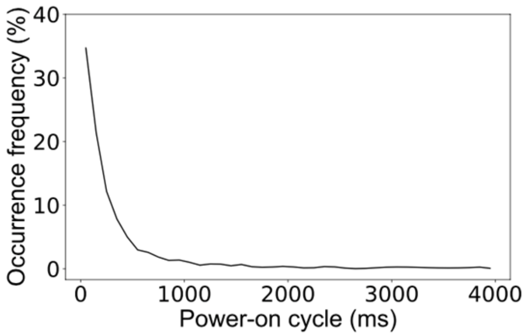
- Ting-Hsuan Lo, Chun-Feng Wu, Yuan-Hao Chang, Tei-Wei Kuo, and Wei-Chen Wang, "Space-efficient Graph Data Placement to Save Energy of ReRAM Crossbar," ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), Virtual Conference, Jul. 26-28, 2021. (Top Conference)

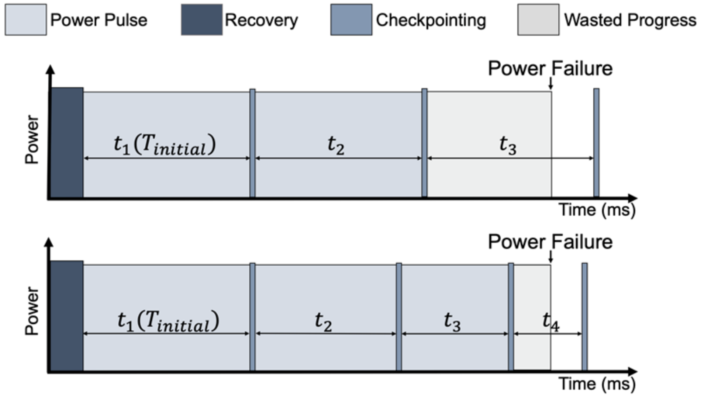
4. Intermittent Systems

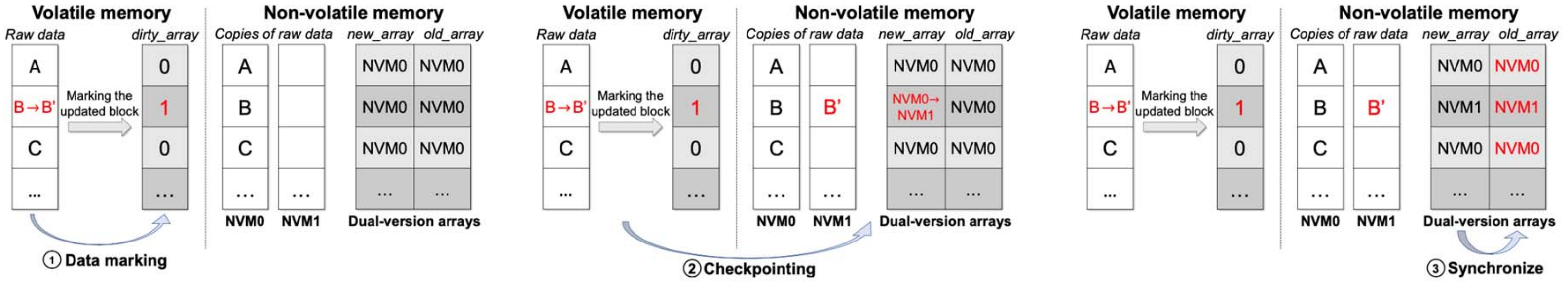
iCheck – Progressive Checkpointing for Intermittent Systems

(IEEE TCAD in 2021)

- Observation:
  - Prior checkpointing approaches rely on capacitor measurement (i.e., for checkpoint timing).
  - The power failure rate of intermittent devices is often lower nearby the start of a power-on cycle.
- Goal:
  - To eliminate the dependency of capacitor since it has relatively shorter lifetime compared with other hardware components in embedded devices.
- Main idea:
  - Power-failure-aware checkpoint triggering that trigger the device to checkpoint progressively, i.e., the checkpointing frequency increases as the time progresses to maximize forward progress.
  - Recoverable incremental-state data checkpointing to avoid data inconsistency issue caused by incomplete checkpointing with dual-version data buffering strategy.







(a) Mark the modified data blocks.

(b) Back up the modified memory blocks.

(c) Synchronize the contents of *new_array* and *old_array*.

- Wen Sheng Lim, Chia-Heng Tu, Chun-Feng Wu, and Yuan-Hao Chang, "iCheck: Progressive Checkpointing for Intermittent Systems," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 40, no. 11, pp. 2224-2236, Nov. 2021.

Energy-Reduction On-chip Memory Management on Intermittent Systems

(RTAS 2021)

Motivation

Existing intermittent systems rarely considered the energy consumed during moving data and may waste precious power resource over data movement.

Goal

Adopting a STT-RAM-based scratchpad memory and a small volatile cache to be the on-chip memory to enable an energy-reduction intermittent system.

Main Challenge

- How to identify the data access pattern in the run time and further distinguish the read/write behavior.
- How to monitor the lifetime of STT-RAM-based SPM

Main Idea

- Maximizing the space utilization of both SPM and cache.
- Avoiding moving the write-intensive data into SPM.

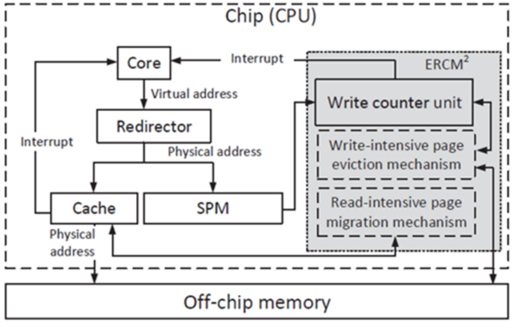


Fig. 1 Architecture of hybrid on-chip memory architecture and ERCM² scheme.

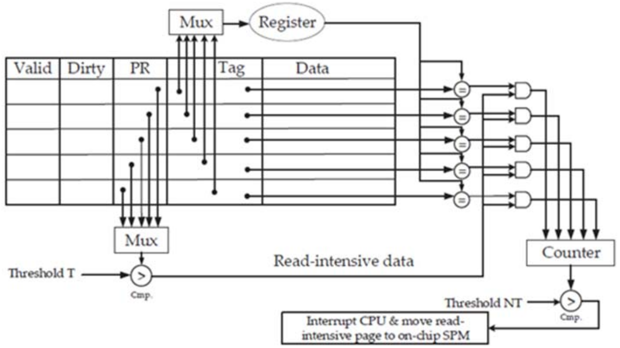


Fig. 2 augmented cache design

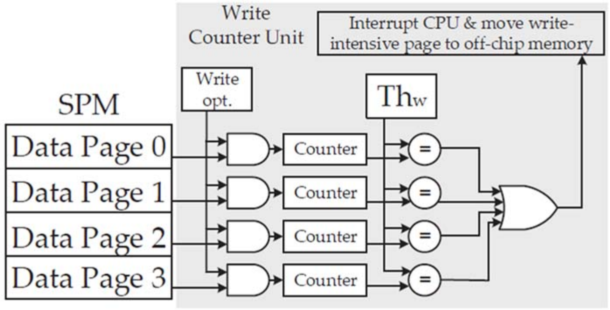


Fig. 3 The proposed write counter unit.

- Yu-Pei Liang, Yu-Ting Fang, Shuo-Han Chen, Yen-Ting Chen, Tseng-Yi Chen, Wei-Lin Wang, Wei-Kuan Shih, and Yuan-Hao Chang, "Brief Industry Paper: An Energy-Reduction On-Chip Memory Management for Intermittent Systems," IEEE Real-time and Embedded Technology and Application Symposium (RTAS), Virtual Conference, May 18-21, 2021. (Top Conference)

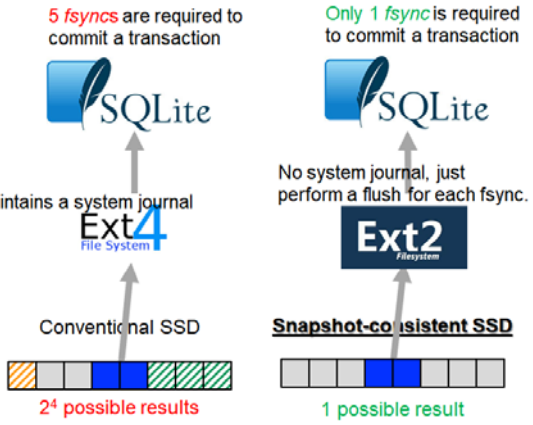
Research Summary 2020

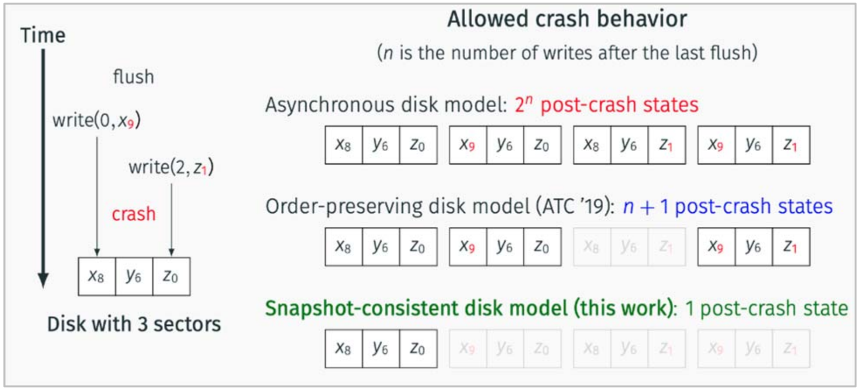
1. Storage Systems - Flash Drives and SMR Disks

Crash Recovery Support from the Storage Level

(OSDI 2020)

- This is a verified Snapshot-Consistent Flash Translation Layer (SCFTL) to guarantee determinized time on recovering a flash drive to the state right before the last flush.
- This is the first attempt to leverage formal verification techniques to ensure the correctness of a complex FTL implementation.
- SCFTL is the first work providing a determinized storage crash recovery mechanism to enable a more efficient design of upper layers in the storage stack (e.g., the file system or database system).
- SCFTL is accepted and published in OSDI 2020 (17.6%).
  - The first OSDI from Taiwan in the past 26 years.
  - SCFTL is available at: https://github.com/yunshengtw/scftl
  - 50% of OSDI’20 papers come from MIT, Berkley, and CMU.
  - More than 25% of OSDI paper comes from huge companies, e.g., Google, Amazon, Facebook


5 fsyncs are required to commit a transaction
SQLite
Maintains a system journal
Ext4 File System
Conventional SSD
2⁴ possible results
Only 1 fsync is required to commit a transaction
SQLite
No system journal, just perform a flush for each fsync.
Ext2 Filesystem
Snapshot-consistent SSD
1 possible result


Allowed crash behavior
(n is the number of writes after the last flush)
Time
flush
write(0, x9)
write(2, z1)
crash
x8 y6 z0
Disk with 3 sectors
Asynchronous disk model: 2ⁿ post-crash states
x8 y6 z0 | x9 y6 z0 | x8 y6 z1 | x9 y6 z1
Order-preserving disk model (ATC ’19): n + 1 post-crash states
x8 y6 z0 | x9 y6 z0 | x8 y6 z1 | x9 y6 z1
Snapshot-consistent disk model (this work): 1 post-crash state
x8 y6 z0 | x9 y6 z0 | x8 y6 z1 | x9 y6 z1

Existing non-determinizied work vs. SCFTL

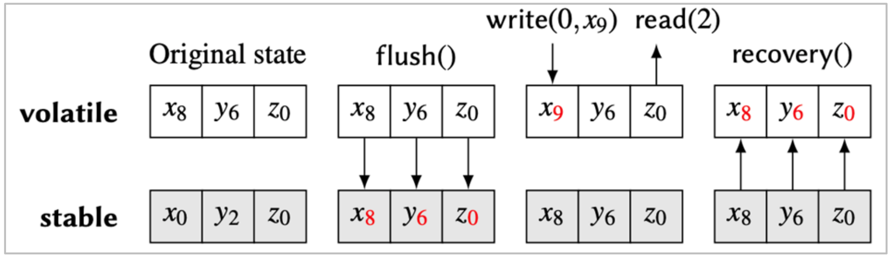
系統界的奧斯卡 -- OSDI 與 SOSP

02-03

Google 的三駕馬車（MapReduce，GFS，Bigtable），Dynamo，Raft，如果你對系統領域感興趣，那這些系統領域鼎鼎大名論文一定聽說過，甚至仔細閱讀過。這些論文通常都發表在系統領域頂級的會議上，其中最著名的是 OSDI 和 SOSP 這兩個會議，可以說是系統界的奧斯卡。

OSDI

OSDI 全稱叫做 USENIX conference on Operating Systems Design and Implementation，但實際上所覆蓋的領域已經遠遠超過 Operation System。OSDI 是系統領域和 SOSP 並駕齊驅的兩個頂級會議之一，大家熟悉的 MapReduce，BigTable，Chubby，Spanner，DryadLINQ，KLEE，Ceph，TensorFlow 等等，都是發表在 OSDI。所以想深入系統領域的同學，OSDI 的論文是必讀的。

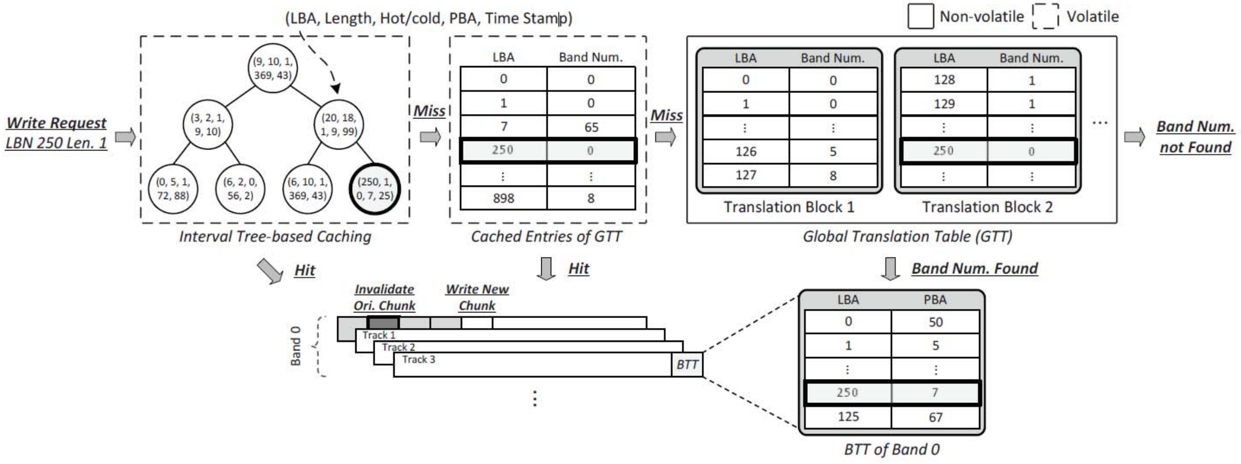


Design Concept of the proposed SCFTL

- Yun-Sheng Chang, Yao Hsiao, Tzu-Chi Lin, Che-Wei Tsao, Chun-Feng Wu, Yuan-Hao Chang, Hsiang-Shang Ko, and Yu-Fang Chen, "Determinizing Crash Behavior with a Verified Snapshot-Consistent Flash Translation Layer" USENIX Symposium on Operating Systems Design and Implementation (OSDI), Banff, Alberta, Canada, Nov. 4-6, 2020. (Acceptance rate: 17.6% (70/398)) (Top Conference)

A Demand-based Shingled Translation Layer for SMR Disks (ACM TECS in 2020)

- Observation
  - Drive-managed SMR (DM-SMR) employs a shingled translation layer (STL) to hide its inherent sequential-write constraint from the host software.
  - However, the access pattern and the data update frequency of incoming workloads are not considered during managing STL
- Goal
  - Enhance the access performance of DM-SMR via considering the access pattern and update frequency at the same time.
- Main Idea
  - Lower the overhead during updating STL
    - Two-layer mapping scheme & Interval tree-based caching
  - Separate hot/cold data to lower overhead during space reclaim
    - *Hotness-aware band allocator & Neighboring invalid space reclamation*



Cached Entries of GTT

LBA	Band Num.
0	0
1	0
7	65
250	0
⋮	⋮
898	8

Translation Block 1

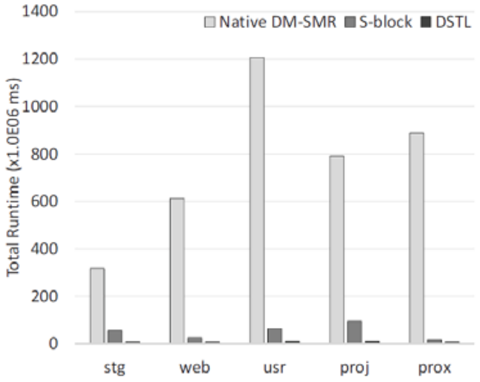
LBA	Band Num.
0	0
1	0
⋮	⋮
126	5
127	8

Translation Block 2

LBA	Band Num.
128	1
129	1
⋮	⋮
250	0
⋮	⋮

BTT of Band 0

LBA	PBA
0	50
1	5
⋮	⋮
250	7
125	67

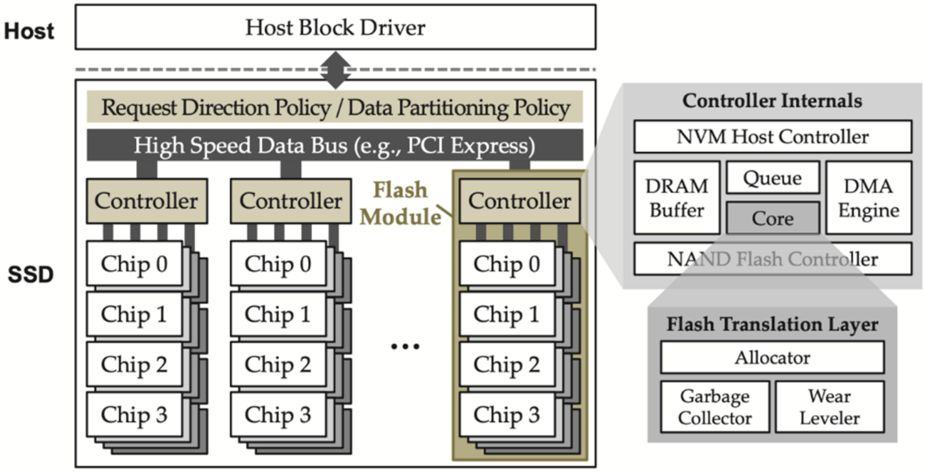


Total Runtime Reduction: 98.72%

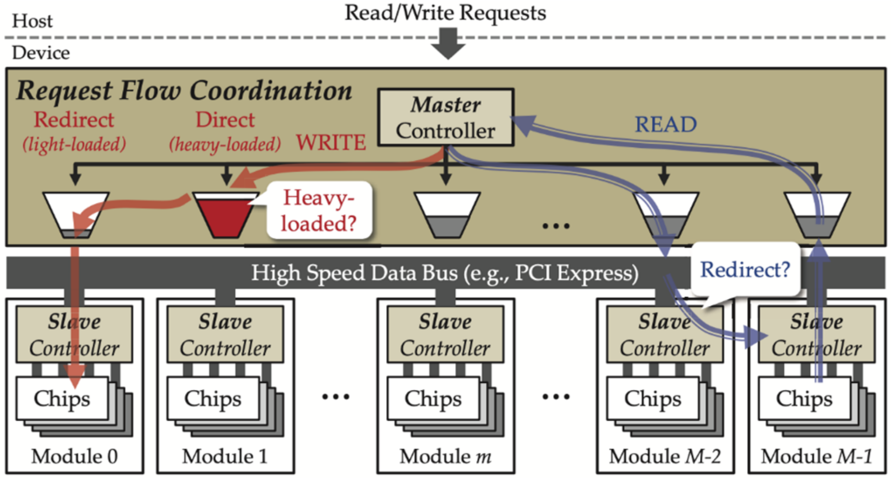
- Yi-Jing Chuang, Shuo-Han Chen, Yuan-Hao Chang, Yu-Pei Liang, Hsin-Wen Wei, and Wei-Kuan Shih, "DSTL: A Demand-based Shingled Translation Layer for Enabling Adaptive Address Mapping on SMR Drives," ACM Transactions on Embedded Computing Systems (TECS), vol. 19, no. 4, pp. 25:1-25:21, Jul. 2020.

Request Flow Coordination for Large-scale Flash Storage

- We propose a *Request Floe Coordination Design* to appropriately control and throttle the I/O request flows over the increasingly complicated SSD internal organization with scalable and manageable coordination overhead. (IEEE TC in 2020)
  - Due to the interface and architecture changes, we aim to achieve good design scalability to facilitate the device development when the scale of SSDs keeps growing. This work tends to embrace the new many-chip and many-core SSD architecture
  - Avoid overloading any sub-module of the SSD by making a good use of the abundant internal resources, so that the request response time can be effectively reduced and the overall SSD performance can be significantly improved.



Architecture of Many Chips & Cores

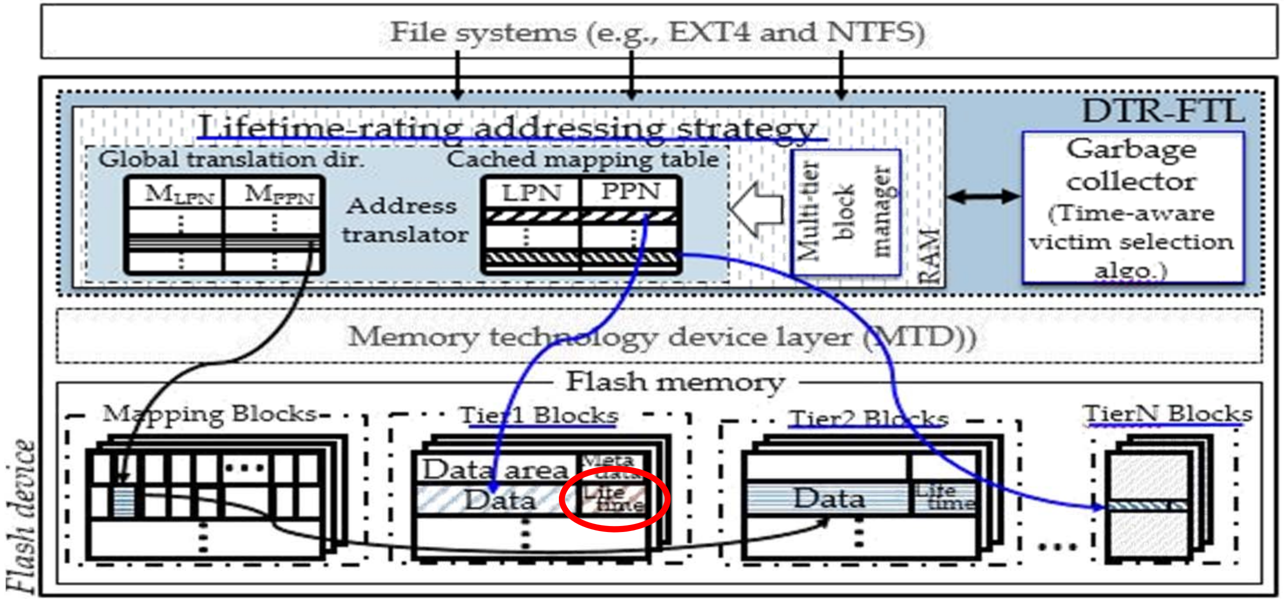


Request Flow Coordination Design

- Ming-Chang Yang, Yuan-Hao Chang, Tei-Wei Kuo, and Chun-Feng Wu, "Request Flow Coordination for Growing-Scale Solid-State Drives," IEEE Transactions on Computers (TC), vol. 69, no. 6, pp. 832-843, Jun. 2020.

Cutting Expired Data with Zero Overhead in Flash Storage (DAC 2020)

- Observation
  - There are many disused (or expired) data in the flash storage device but still be treated as valid data. The unnecessary page copying during GC process cause huge extra overhead.
  - Once the host system can provide the data lifetime information, we can directly identify data is expired or not.
- Goal and Method
  - We proposed a dual-time referencing FTL design (DTR-FTL). Our DTR-FTL will aggregate data together based on the data lifetime information and block retention time.
  - Automatically cut out expired via lifetime information.
  - Using those low retention time blocks (high P/E cycles blocks) to record short lifetime data.

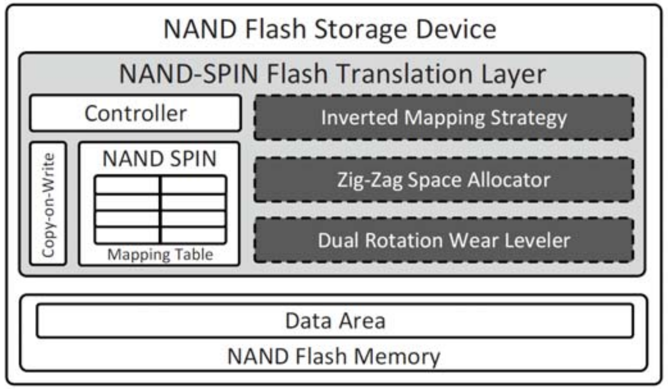


Wei-Lin Wang, Tseng-Yi Chen, Yuan-Hao Chang, Hsin-Wen Wei, and Wei-Kuan Shih, "How to Cut Out Expired Data with Nearly Zero Overhead for Solid-State Drives," ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, Jul. 19-23, 2020. (Acceptance rate: 23%(228/984)) **(Top Conference)**

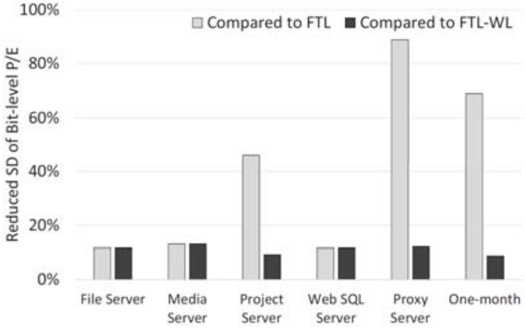
Bit-Level Wearing for NAND-SPIN

(NVMSA 2020)

- Observation
  - NAND-SPIN memory resolved the high program latency & inefficient power consumption of 1T1MTJ STT-MRAM
  - Unique characteristic: A string of NAND-SPIN cells can be updated multiple times before being erased
- Goal
  - Alleviate the bit-level uneven wearing issue by allowing more bits to be programed before being erased
- Main Idea
  - Avoid both uneven intra-entry wearing and excessive string-based erases
    - Inverted mapping strategy & Zig-Zag space allocator
  - Perform wear leveling at both bit and entry level
    - *Dual rotation wear leveler*



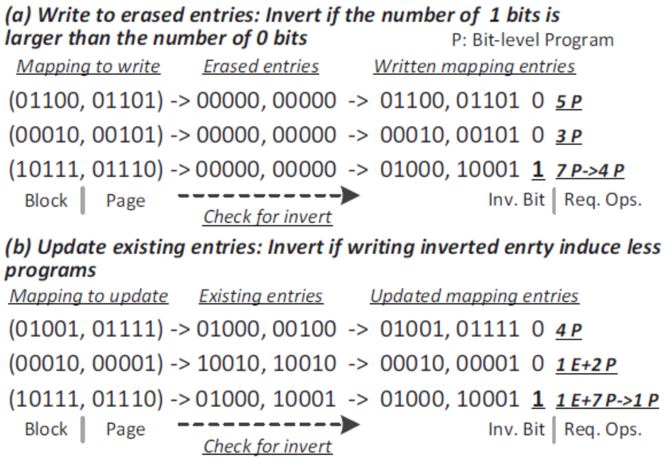
NAND Flash Storage Device
NAND-SPIN Flash Translation Layer
Controller
Inverted Mapping Strategy
Copy-on-Write
NAND SPIN
Mapping Table
Zig-Zag Space Allocator
Dual Rotation Wear Leveler
Data Area
NAND Flash Memory



Bit-level P/E

Std. Reduction : 40.11%

Mean Reduction : 9.86%



(a) Write to erased entries: Invert if the number of 1 bits is larger than the number of 0 bits

P: Bit-level Program

Mapping to write *Erased entries* *Written mapping entries*

(01100, 01101) -> 00000, 00000 -> 01100, 01101 0 *5 P*

(00010, 00101) -> 00000, 00000 -> 00010, 00101 0 *3 P*

(10111, 01110) -> 00000, 00000 -> 01000, 10001 **1** *7 P->4 P*

Block | Page *Check for invert* Inv. Bit | Req. Ops.

(b) Update existing entries: Invert if writing inverted enrty induce less programs

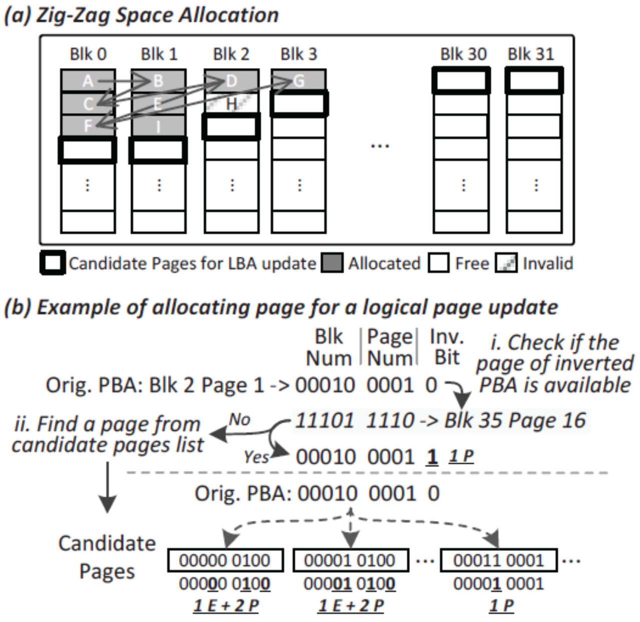
Mapping to update *Existing entries* *Updated mapping entries*

(01001, 01111) -> 01000, 00100 -> 01001, 01111 0 *4 P*

(00010, 00001) -> 10010, 10010 -> 00010, 00001 0 *1 E+2 P*

(10111, 01110) -> 01000, 10001 -> 01000, 10001 **1** *1 E+7 P->1 P*

Block | Page *Check for invert* Inv. Bit | Req. Ops.



(a) Zig-Zag Space Allocation

(b) Example of allocating page for a logical page update

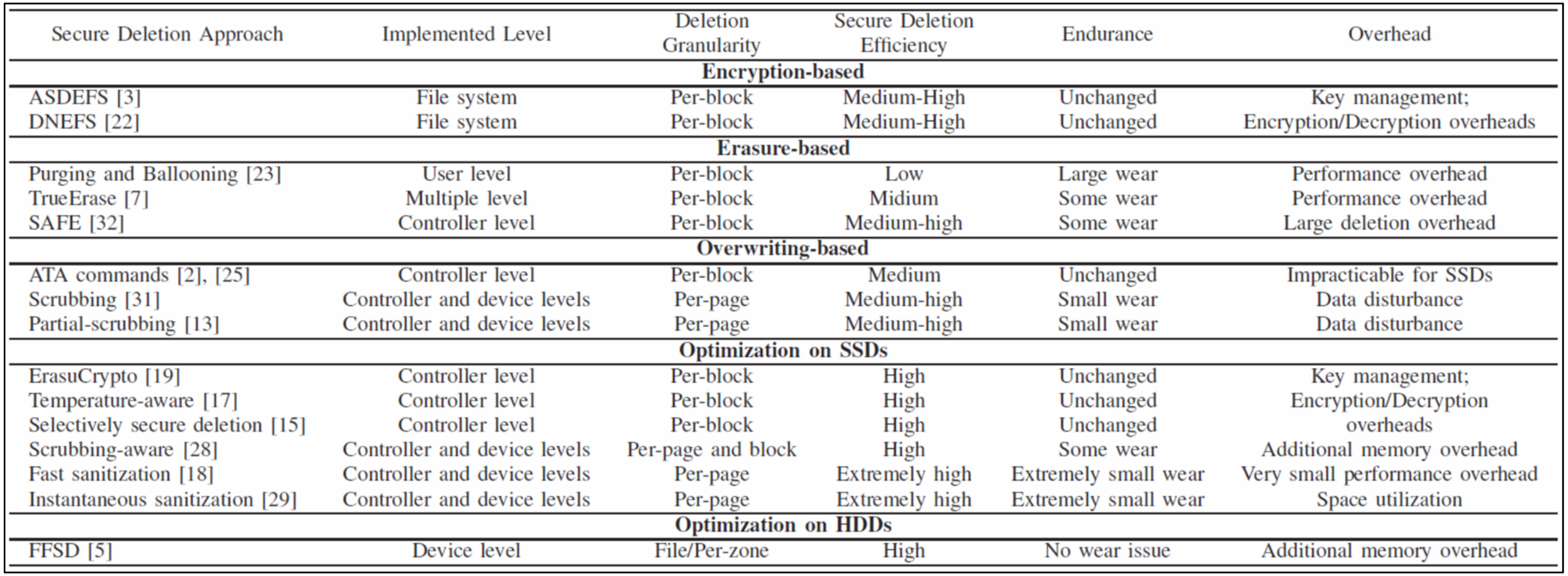
Wei-Chun Cheng, Shuo-Han Chen, Yuan-Hao Chang, Kuan-Hsun Chen, Jian-Jia Chen, Tseng-Yi Chen, Ming-Chang Yang, and Wei-Kuan Shih, "Alleviating the Uneven Bit-Level Wearing of NVRAM-based FTL via NAND-SPIN," IEEE Nonvolatile Memory Systems and Applications Symposium (NVMSA), Korea, Aug. 19-21, 2020.

Challenges of Secure Deletion in Storages

Objectives: (ICAN 2020)

This work aims at evaluating and comparing the implementation challenges of secure data deletion and sanitization techniques.

- ➢ State-of-the-art designs that have been paid to pursue better efficiency, verifiability, and portability for both HDDs and SSDs are summarized
- ➢ The pros and cons on implementing “secure deletion” of different techniques are discussed
  - ● E.g., encryption-based, erasure-based, overwriting-based, and mixed techniques



Secure Deletion Approach	Implemented Level	Deletion Granularity	Secure Deletion Efficiency	Endurance	Overhead
Encryption-based					
ASDEFS [3]	File system	Per-block	Medium-High	Unchanged	Key management;
DNEFS [22]	File system	Per-block	Medium-High	Unchanged	Encryption/Decryption overheads
Erasure-based					
Purging and Ballooning [23]	User level	Per-block	Low	Large wear	Performance overhead
TrueErase [7]	Multiple level	Per-block	Midium	Some wear	Performance overhead
SAFE [32]	Controller level	Per-block	Medium-high	Some wear	Large deletion overhead
Overwriting-based					
ATA commands [2], [25]	Controller level	Per-block	Medium	Unchanged	Impracticable for SSDs
Scrubbing [31]	Controller and device levels	Per-page	Medium-high	Small wear	Data disturbance
Partial-scrubbing [13]	Controller and device levels	Per-page	Medium-high	Small wear	Data disturbance
Optimization on SSDs					
ErasuCrypto [19]	Controller level	Per-block	High	Unchanged	Key management;
Temperature-aware [17]	Controller level	Per-block	High	Unchanged	Encryption/Decryption
Selectively secure deletion [15]	Controller level	Per-block	High	Unchanged	overheads
Scrubbing-aware [28]	Controller and device levels	Per-page and block	High	Some wear	Additional memory overhead
Fast sanitization [18]	Controller and device levels	Per-page	Extremely high	Extremely small wear	Very small performance overhead
Instantaneous sanitization [29]	Controller and device levels	Per-page	Extremely high	Extremely small wear	Space utilization
Optimization on HDDs					
FFSD [5]	Device level	File/Per-zone	High	No wear issue	Additional memory overhead

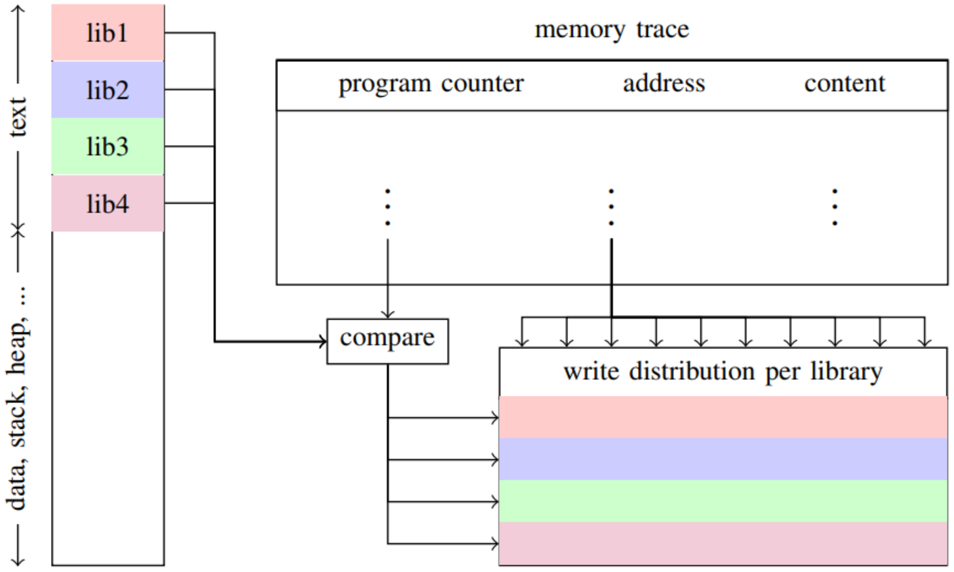
- Wei-Chen Wang, Chien-Chung Ho, Yu-Ming Chang, and Yuan-Hao Chang, "Challenges and Designs for Secure Deletion in Storage Systems," IEEE International Conference on Computing, Analytics and Networks (ICAN), Chiayi, Taiwan, Feb. 7-8, 2020.

2. NVM Main Memory and Storage

Split’n Trace NVM by Leveraging Library

(NVMSA 2020)

- Observation
  - NVM suffers from (1) reduced cell endurance and (2) asymmetric read/write latency
  - Memory access analysis is becoming an increasingly important topic.
  - However, existing analyzers usually neglect the need of without investigating the application semantics of different memory regions.
- Goal
  - Enriches the simulation result with semantics from the analyzed application and splits the main memory into semantic regions and
- Approach
  - Leverage Unikraft by ascribing memory regions of the simulation to the relevant OS libraries
- Result
  - Derive a detailed analysis of which libraries and thus functionalities are responsible for which memory access patterns.

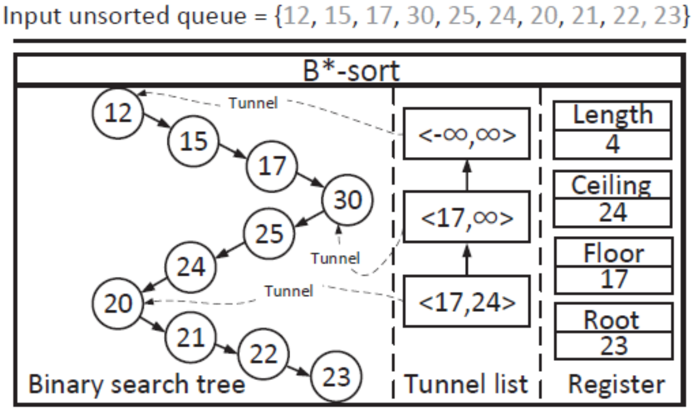


- Christian Hakert, Kuan-Hsun Chen, Simon Kuenzer, Sharan Santhanam, Shuo-Han Chen, Yuan-Hao Chang, Felipe Huici, and Jian-Jia Chen, "Split'n Trace NVM: Leveraging Library OSes for Semantic Memory Tracing," IEEE Nonvolatile Memory Systems and Applications Symposium (NVMSA), Korea, Aug. 19-21, 2020.

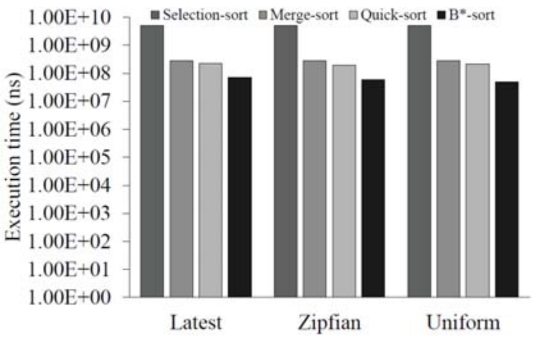
B*-sort: Enabling Write-once Sorting on NVM

(IEEE TCAD in 2020)

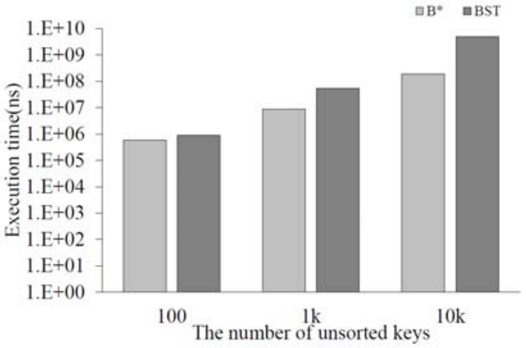
- Observation
  - Most NVM devices have read/write asymmetric issue
  - Sorting algorithms are commonly-used algorithm
  - However, the conventional array-based sorting algorithms are unfriendly for the NVM-based systems because of the heavy write traffic
- Goal
  - Mitigate the influence cause by read/write asymmetric on NVM-based systems
- Main Idea
  - Use the nature property of Binary search tree
    - *Write-once property*
    - *To get sorted result by in-order traversal*
  - Use a linked-list (tunnel list) to avoid the worst case
    - *List of shortcut to the root nodes of subtrees*



Pwfswjfx!pg!C+.tpsu



Fyfdvujpo!ujnf!
gps!nbjo!nfnpsz!bddftt

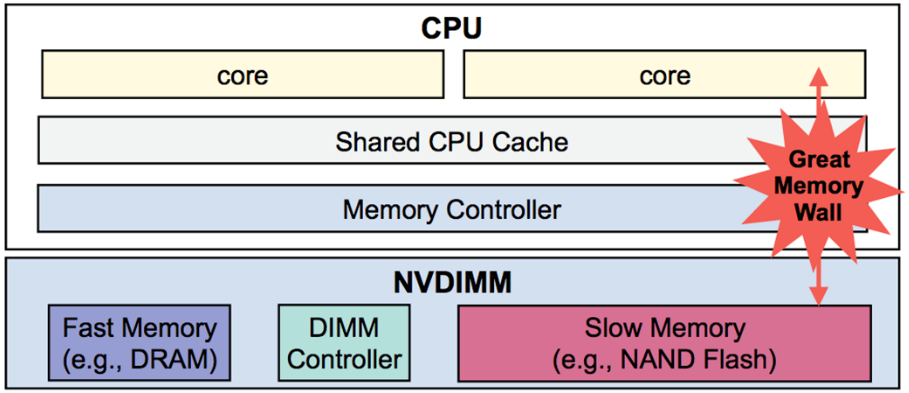


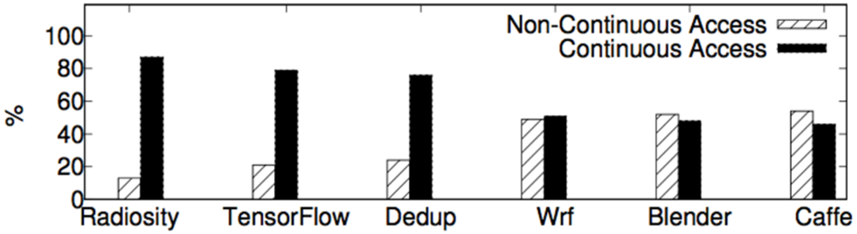
Xpstu.dbtf!fyfdvujpo!ujnf!
)C+.tpsu!wt/!CTU*

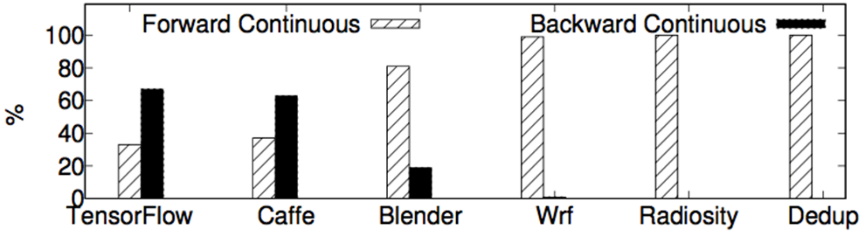
- Yu-Pei Liang, Tseng-Yi Chen, Yuan-Hao Chang, Shuo-Han Chen, Hsin-Wen Wei, and Wei-Kuan Shih, "B*-sort: Enabling Write-once Sorting for Non-volatile Memory," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 39, no. 12, pp. 4549-4562, Dec. 2020.

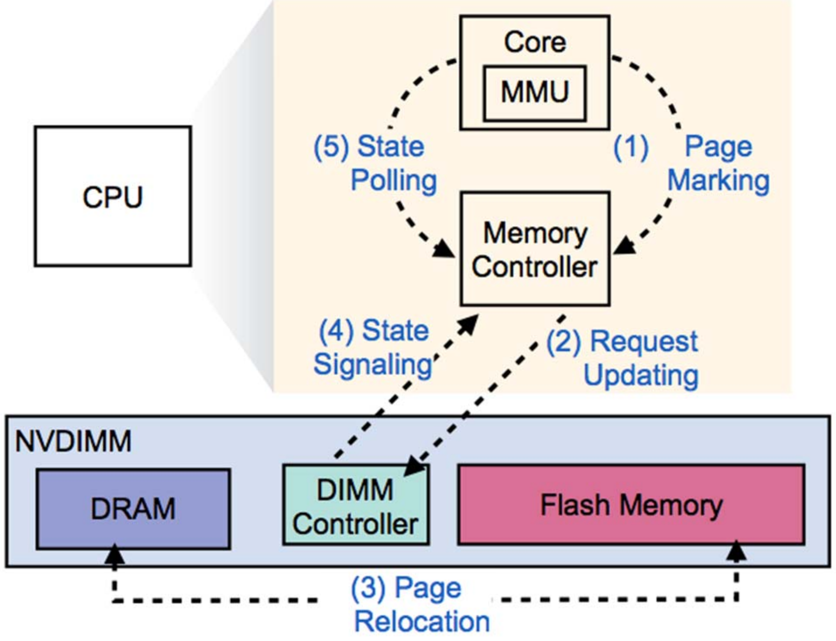
Joint Management of CPU and NVDIMM

- We propose a *joint management framework* to efficiently relieving the impact of great memory wall. (IEEE TC in 2020)
  - Investigate the potential challenges by applying NVDIMM for expanding the main memory. Also, investigate process access behaviors on memory.
  - Propose a page semantic-aware strategy to precisely predict, mark, and relocate memory pages to the fast DRAM from the slow flash memory in advance.





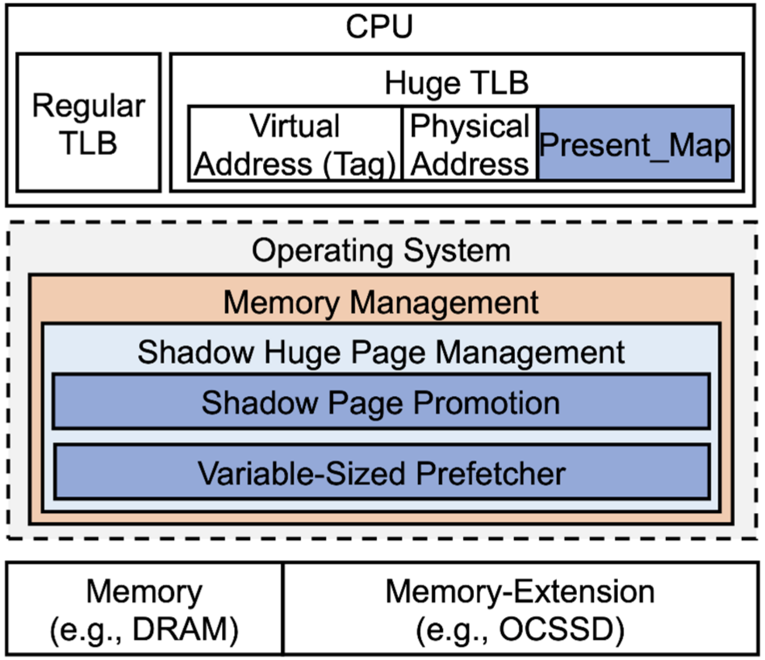


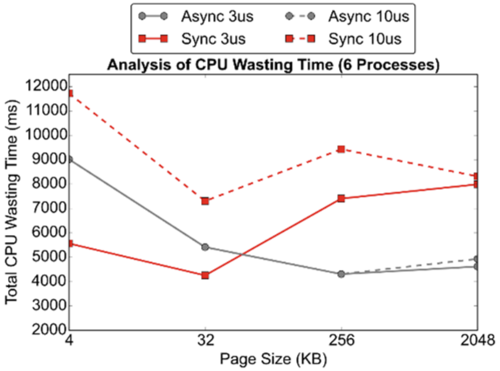


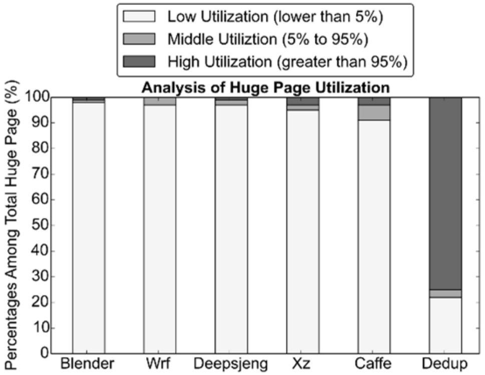
- Chun-Feng Wu, Yuan-Hao Chang, Ming-Chang Yang, and Tei-Wei Kuo, "Joint Management of CPU and NVDIMM for Breaking Down the Great Memory Wall," IEEE Transactions on Computers (TC), vol. 69, no. 5, pp. 722-733, May 2020.

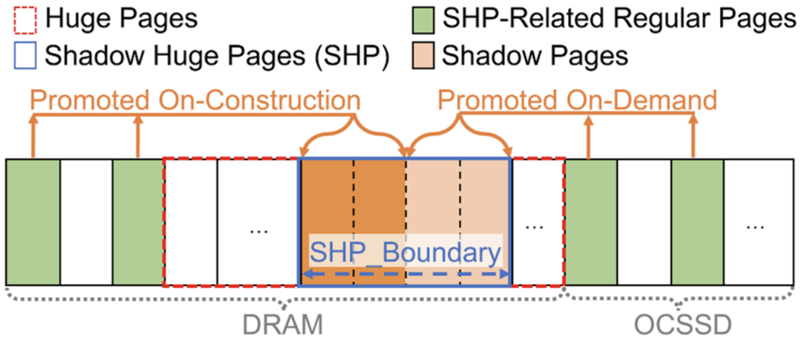
When Storage Response Time Catches Up with Overall Context Switch Overhead

- We propose the shadow huge page management to minimize the total CPU wasting time caused by performing data movements and context switch. (IEEE TCAD in 2020, COES+ISSS 2020)
  - With placing ULL devices as memory extension, it is more efficient to keep the CPU core busy waiting without triggering context switch for serving small-size data movements.
  - For serving big-size data movements, the shadow page promotion is designed to minimize the data movements while moving or constructing a huge page
  - The variable-sized prefetcher can avoid unnecessary data movements by adaptively changing the amount of prefetched data with being aware of the spatial locality.







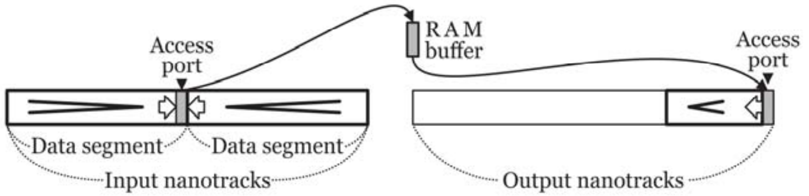


- Chun-Feng Wu, Yuan-Hao Chang, Ming-Chang Yang, and Tei-Wei Kuo, "When Storage Response Time Catches Up with Overall Context Switch Overhead, What is Next?," accepted and to appear in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD). (Integrated with ACM/IEEE CODES+ISSS'20)
- Chun-Feng Wu, Yuan-Hao Chang, Ming-Chang Yang, and Tei-Wei Kuo, "When Storage Response Time Catches Up with Overall Context Switch Overhead, What is Next?," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Germany, Sep. 20 - 25, 2020. (Journal Track, Integrated with IEEE TCAD) (Acceptance rate: 21.9%(28/128)) **(Top Conference)**

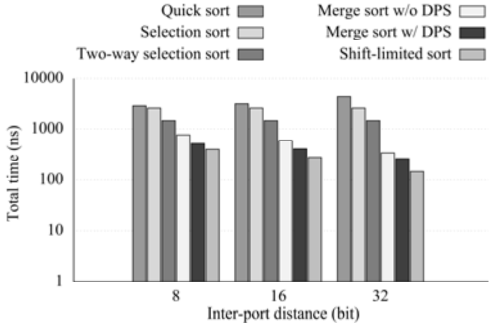
Shift-limited Sort on Skyrmion Memory based Systems

(IEEE TCAD in 2020, CODES+ISSS 2020)

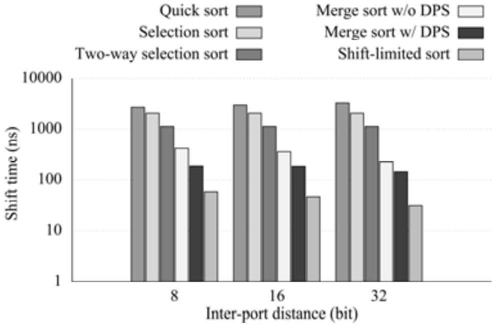
- Observation
  - Skyrmion Racetrack Memory (SK-RM) requires time-costly *shift operations* to align the to-be-accessed data bits with access ports.
- Goal
  - Eliminate unnecessary shift operations.
- Main Idea
  - Propose a *back-to-back* data placement strategy to boost merging.



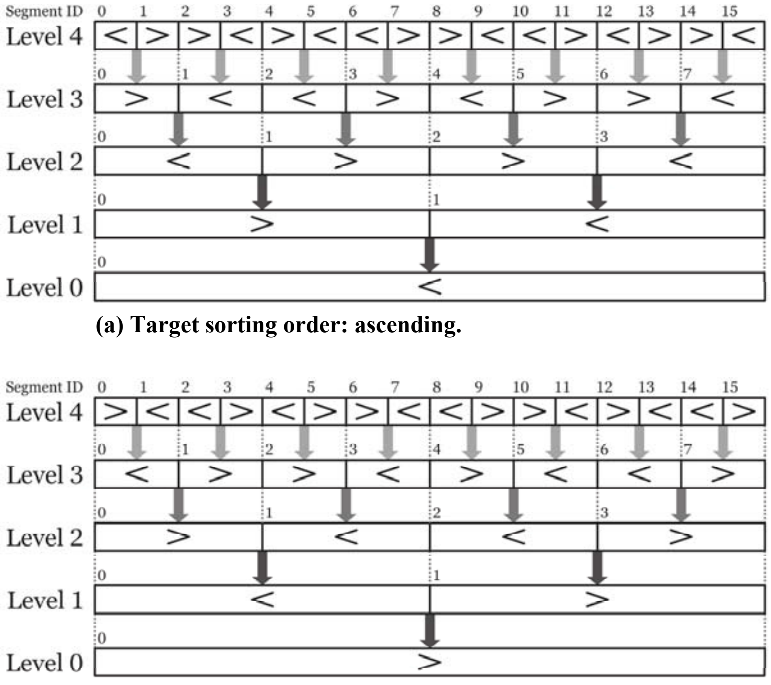
Concepts of the segmental-merging algorithm.



The overall I/O time of SK-RM. (Track size: 64 bits)



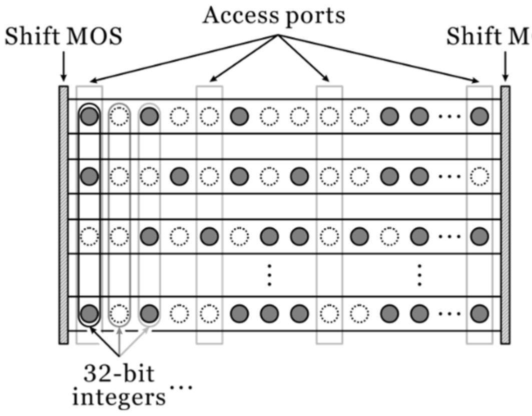
Shift overheads. (Track size: 64 bits)



(a) Target sorting order: ascending.

(b) Target sorting order: descending.

Determining the initial sorting order of each data segment.

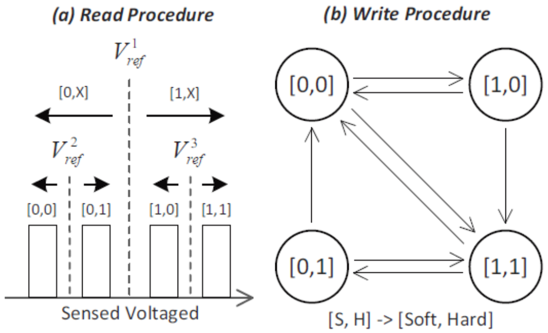


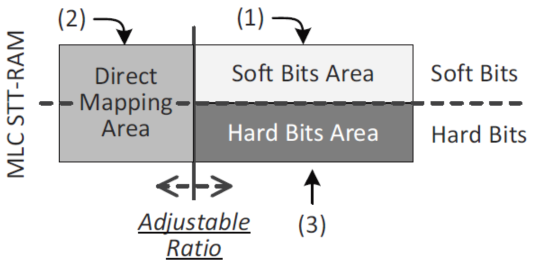
Bit-interleaved and block cluster architecture.

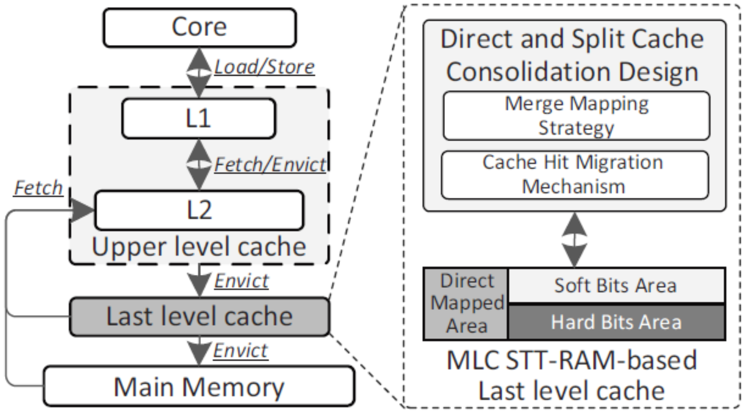
- Yun-Shan Hsieh, Po-Chun Huang, Ping-Xiang Chen, Yuan-Hao Chang, Wang Kang, Ming-Chang Yang, and Wei-Kuan Shih, "Shift-limited Sort: Optimizing Sorting Performance on Skyrmion Memory based Systems," accepted and to appear in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD). (Integrated with ACM/IEEE CODES+ISSS'20)
- Yun-Shan Hsieh, Po-Chun Huang, Ping-Xiang Chen, Yuan-Hao Chang, Wang Kang, Ming-Chang Yang, and Wei-Kuan Shih, "Shift-limited Sort: Optimizing Sorting Performance on Skyrmion Memory based Systems," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Germany, Sep. 20 - 25, 2020. (Journal Track, Integrated with IEEE TCAD) (Acceptance rate: 21.9%(28/128)) **(Top Conference)**

Direct and Split STT-MRAM Cache (SAC 2020)

- Observation
  - On MLC STT-RAM, the write disturbance issue emerged due to its connected structure
    - Larger MTJ -> Hard Bit, Smaller MTJ -> Soft Bit
    - Writes to hard bit always destroy the content of soft bit
    - Extra energy and latency are required for restoring soft bit
- Goal
  - A simple and effective solution to improve the energy efficiency of MLC STT-RAM
- Main Idea
  - Incorporate the direct and cell split mapping methods for enhancing the energy efficiency
  - Avoid frequent swap between SBL and HBL






Core

Load/Store

L1

Fetch/Envict

Fetch

L2

Upper level cache

Envict

Last level cache

Envict

Main Memory

Direct and Split Cache Consolidation Design

Merge Mapping Strategy

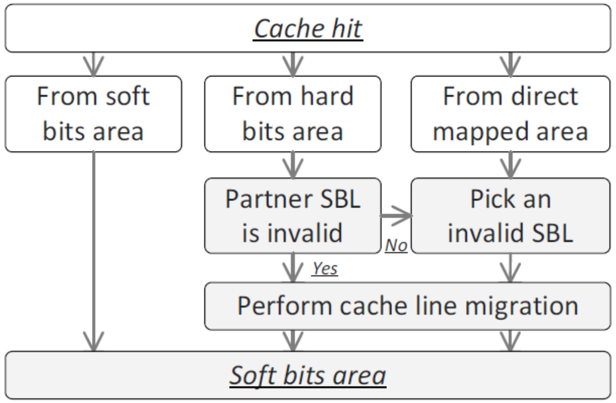
Cache Hit Migration Mechanism

Direct Mapped Area

Soft Bits Area

Hard Bits Area

MLC STT-RAM-based Last level cache


Cache hit

From soft bits area

From hard bits area

From direct mapped area

Partner SBL is invalid

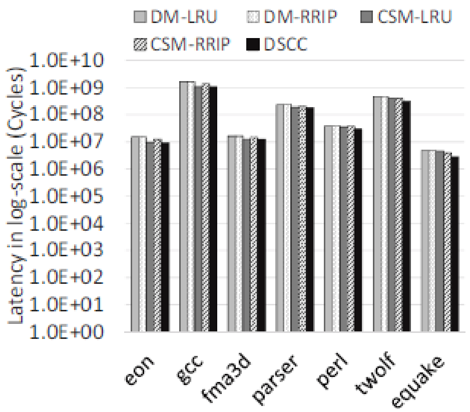
No

Pick an invalid SBL

Yes

Perform cache line migration

Soft bits area



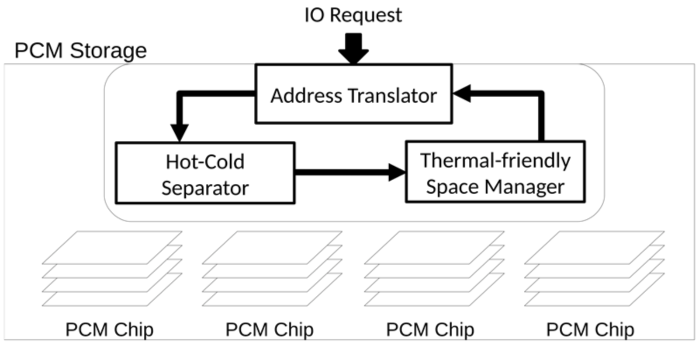
Latency Reduction : 39.5%

- Shuo-Han Chen, Yu-Pei Liang, Yuan-Hao Chang, Yun-Fei Liu, Chun-Feng Wu, Hsin-Wen Wei, and Wei-Kuan Shih, "Reinforcing the Energy Efficiency of Cyber-Physical Systems via Direct and Split Cache Consolidation on MLC STT-RAM," ACM Symposium on Applied Computing (SAC), Brno, Czech Republic, Mar. 30 - Apr. 3, 2020.

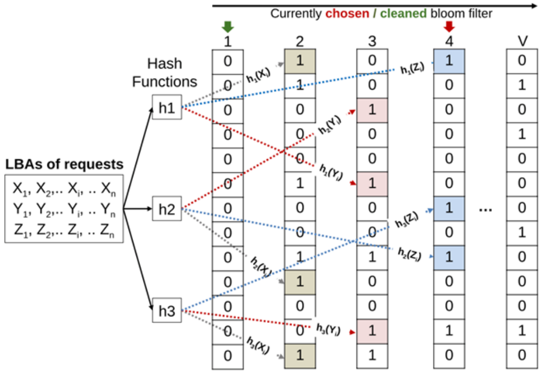
Overheating-avoidance for 3D PCM Storage

(RACS 2020)

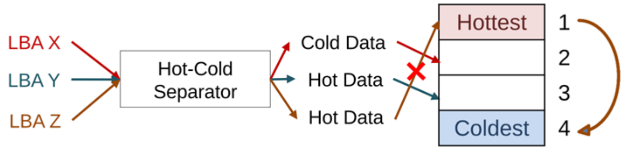
- Observation:
  - The ambient temperatures of many layers is higher than the crystallization transition temperature
  - The programmed times of some cells exceed the limited programmed times
- We proposed an *overheating-avoidance remapping approach* to avoid the overheated ambient temperatures of PCM layers and to achieve the wear leveling at the same time.
  - **Hot-Cold Separator**
    - To record the recency and frequency of data accesses and to determine data access behavior
  - **Thermal-friendly Space Manager**
    - To allocate data to the suitable PCM layers according to the data access behavior and the ambient temperatures of PCM layers



OARS Architecture



1	2	3	4	V
0	1	0	1	0
0	1	0	0	1
0	0	1	0	0
0	0	0	0	1
0	0	0	0	0
0	1	1	0	0
0	0	0	1	0
0	0	0	0	1
0	1	1	1	0
0	1	0	0	0
0	0	0	0	0
0	0	1	1	1
0	1	1	0	0



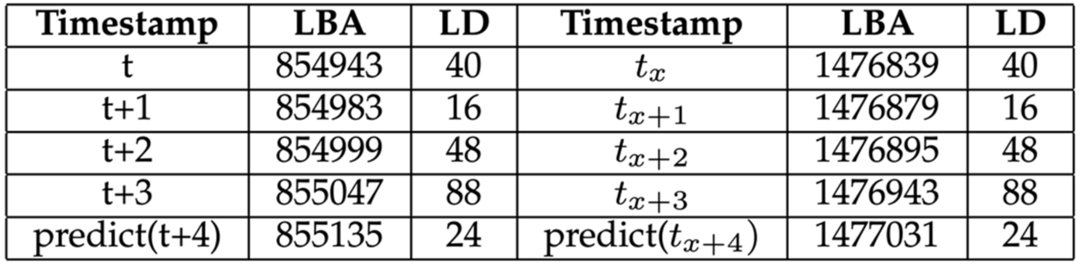
- Yu-Chen Lin, Tse-Yuan Wang, Che-Wei Tsao, Yuan-Hao Chang, Jian-Jia Chen, Xue Liu, and Tei-Wei Kuo, "Overheating-Avoidance Remapping Scheme for Reliability Enhancement of 3D PCM Storage Systems," ACM Research in Adaptive and Convergent Systems (RACS), October 2020.

3. Machine Learning Techniques with NVM

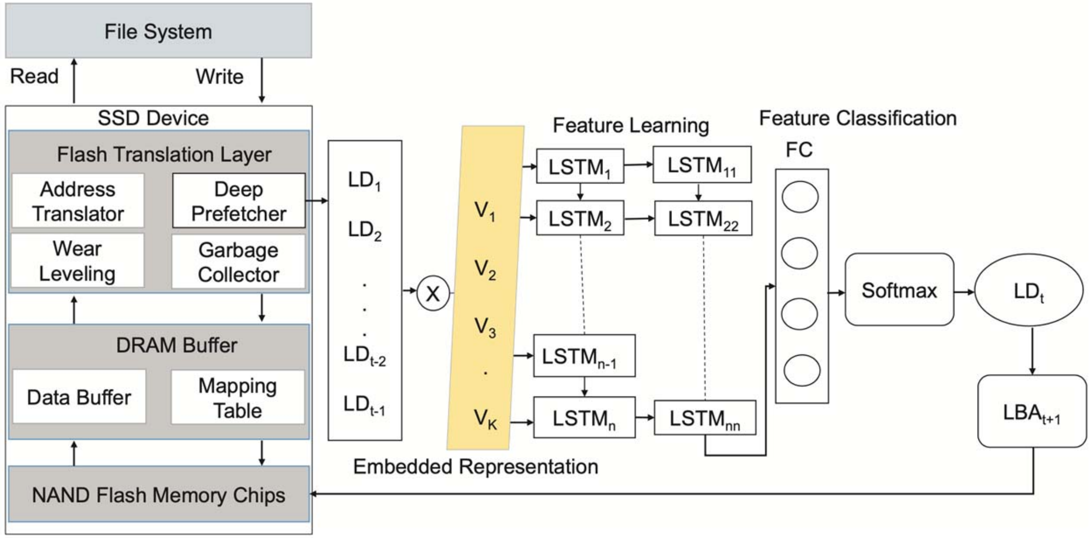
DeepPrefetcher: Data Prefetching in SSD with Deep Learning

- We propose the DeepPrefethcer to reduce the information complexity for improving the prediction accuracy. (IEEE TCAD in 2020, CASES 2020)
  - Data prefetching is one of the potential solutions to predict and move the data from the NAND flash chip to the SSD buffer.
  - New Features: Introduced the LD feature which indicates the difference between successive LBA requests. (LD$_t$ = LBA$_t$+1 – LBA$_t$)
  - Provides more expressive representation by capturing contextual resemblance and semantic sequence of data.

Example of LD feature



Timestamp	LBA	LD	Timestamp	LBA	LD
t	854943	40	t_x	1476839	40
t+1	854983	16	t_{x+1}	1476879	16
t+2	854999	48	t_{x+2}	1476895	48
t+3	855047	88	t_{x+3}	1476943	88
predict(t+4)	855135	24	predict(t_{x+4})	1477031	24

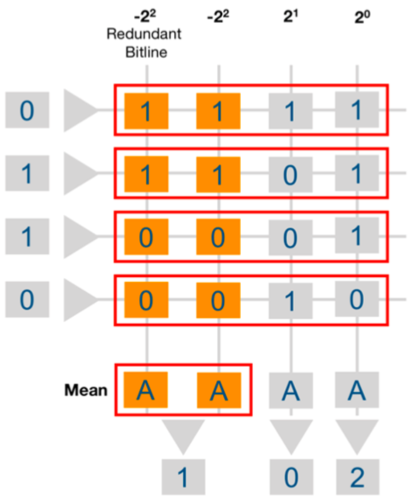


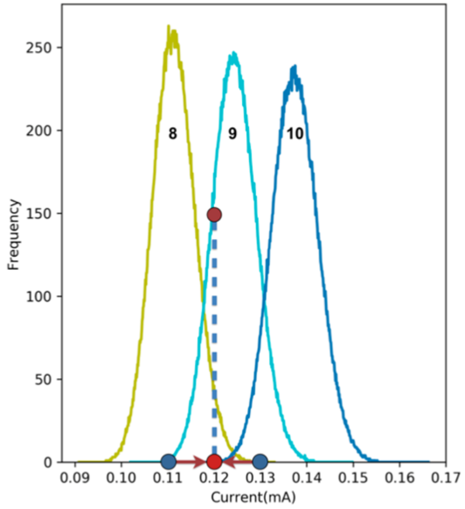
An Overview of DeepPrefetcher

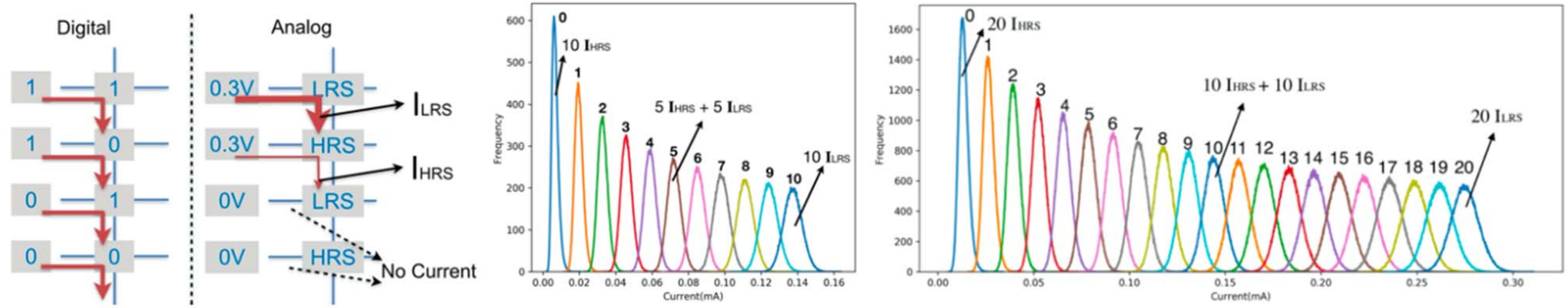
- Gaddisa Olani Ganfure, Chun-Feng Wu, Yuan-Hao Chang, and Wei-Kuan Shih, "DeepPrefetcher: A Deep Learning Framework for Data Prefetching in Flash Storage Devices," accepted and to appear in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD). (Integrated with ACM/IEEE CASES'20)
- Gaddisa Olani Ganfure, Chun-Feng Wu, Yuan-Hao Chang, and Wei-Kuan Shih, "DeepPrefetcher: A Deep Learning Framework for Data Prefetching in Flash Storage Devices," ACM/IEEE International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES), Germany, Sep. 20 - 25, 2020. (Journal Track, Integrated with IEEE TCAD) **(Top Conference)**

Minimizing Analog Variation Errors of ReRAM Crossbar

- We propose an *Adaptive Data Manipulation Strategy* to significantly reduce the occurrence of the overlapping variation error. (IEEE TCAD in 2020, EMSOFT 2020)
  - Overlapping variation error: Current distributions becomes wider while more ReRAM cells in the LRS state are involved; and wider distribution overlaps with neighbors.
  - Proposed designs aim to amortize the sensing results retrieved from redundant bit-lines so that the magnitude of the overlapping variation error can be alleviated







(a) Configuration Conversion from Digital to Analog Aspect

(b) Accumulated Current Distribution of 10 Valid Wordlines

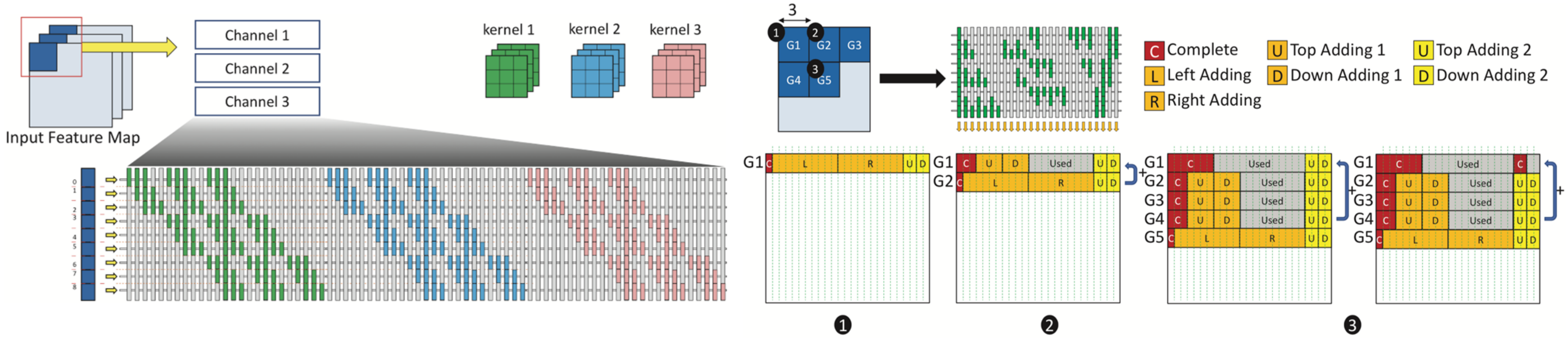
(c) Accumulated Current Distribution of 20 Valid Wordlines

- Yao-Wen Kang, Chun-Feng Wu, Yuan-Hao Chang, Tei-Wei Kuo, and Shu-Yin Ho, "On Minimizing Analog Variation Errors to Resolve the Scalability Issue of ReRAM-based Crossbar Accelerator," accepted and to appear in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD). (Integrated with ACM/IEEE EMSOFT'20)
- Yao-Wen Kang, Chun-Feng Wu, Yuan-Hao Chang, Tei-Wei Kuo, and Shu-Yin Ho, "On Minimizing Analog Variation Errors to Resolve the Scalability Issue of ReRAM-based Crossbar Accelerator," ACM/IEEE International Conference on Embedded Software (EMSOFT), Germany, Sep. 20 - 25, 2020. (Journal Track, Integrated with IEEE TCAD) **(Top Conference)**

Kernel Unfolding to Enhance CNN Performance

(NVMSA in 2020)

- Observation:
  - The convolutional neural network brings a large requirement on the MAC operations and results in huge cost on feeding input feature map, especially for high bandwidth but long access latency PIM devices.
- We proposed a *kernel unfolding approach* to trade data movement with computation power.
  - **Kernel unfolding scheme:**
    - To fully utilize the sensing units by spreading the unfolding kernel over all bitlines.
  - **Output buffer management:**
    - To efficiently merge the partial sum into the output feature map and reduce the additional movement cost on the output feature map.

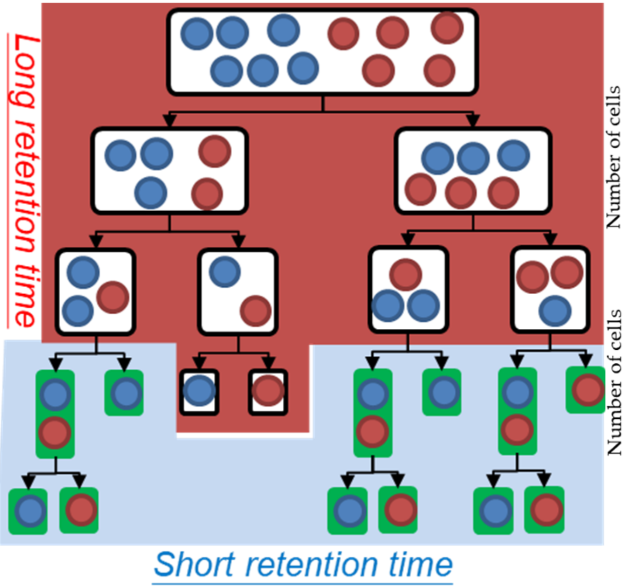


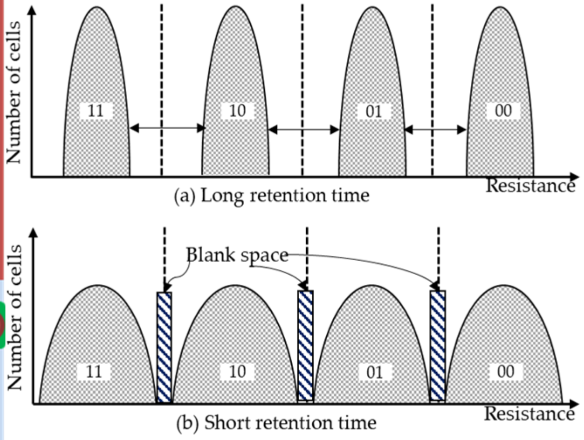
Yueh-Han Wu, Tse-Yuan Wang, Yuan-Hao Chang, Tei-Wei Kuo, and Hung-Sheng Chang, "A Kernel Unfolding Approach to Trade Data Movement with Computation Power for CNN Acceleration," IEEE Nonvolatile Memory Systems and Applications Symposium (NVMSA), Korea, Aug. 19-21, 2020.

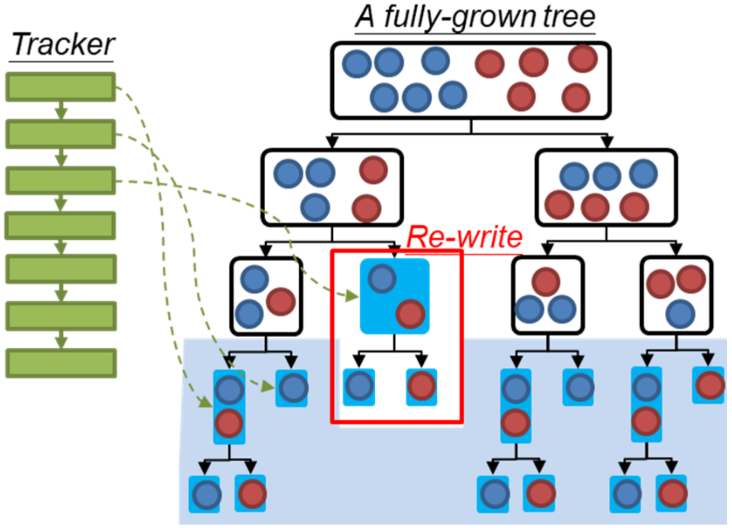
Cultivating Random Forest w/o Loss of Accuracy

(ACM/IEEE ISLPED 2020 – Best Paper Award)

- Observation
  - In a decision tree algorithm, redundant nodes will be pruned by post-pruning strategies because of the overfitting problem.
  - The redundant nodes become unnecessary writes to non-volatile memory device
- Goal
  - We proposed Duo-phase Pruning Framework to minimize the energy consumption of decision tree construction without losing accuracy.
- Main Idea
  - To write nodes marked by a pre-pruning strategy in the low-cost writing mode.
  - To keep high accuracy through rewriting the misclassified nodes in the normal mode.
- Energy efficiency is improved for 30%.
- The first ISLPED best paper award from Taiwan in the past 25 years.







Tseng-Yi Chen, Yuan-Hao Chang, Ming-Chang Yang, and Huang-Wei Chen, "How to Cultivate a Green Decision Tree without Loss of Accuracy?" ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA, Aug. 10-12, 2020. **(Best Paper Award, Top Conference)**

Beyond Address Mapping: User-Oriented Multi-Regional Management for 3D Flash

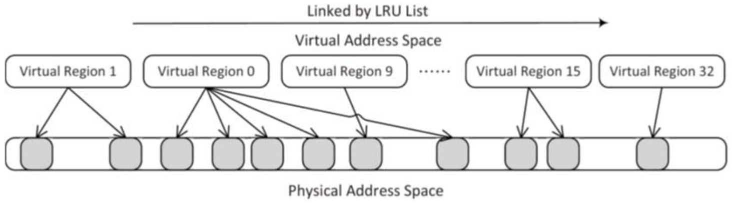
Observation: (IEEE TCAD in 2020)

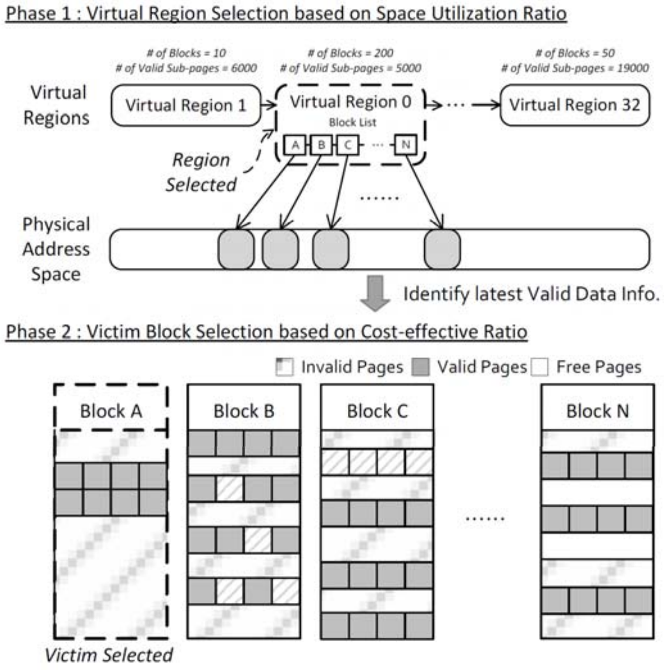
Access performance of 3D NAND flash could be degraded because of the trend of the growing number of access/erase units.

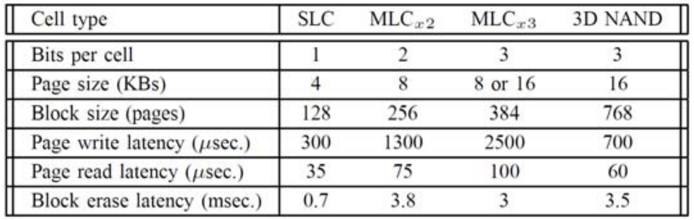
We proposed a multi-regional space management design

- A multi-regional mapping scheme
  - To enable subpage-level address mapping
  - To adaptively adjust the mapping granularity
- A user-oriented buffering approach
  - To dynamically allocate storage space for absorbing update requests
- A two-step victim block selection mechanism
  - To enhance the garbage collection efficiency via the maintained mapping information

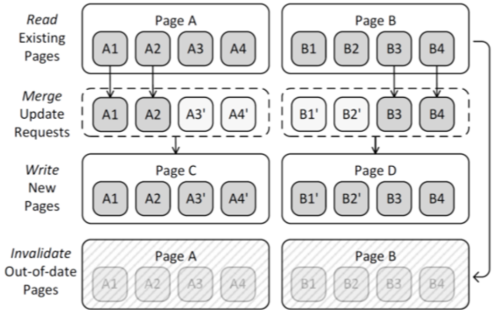
With the proposed design, the access performance of 3D NAND flash storage devices can be improved by an average of 61% when compared with the conventional DFTL design.







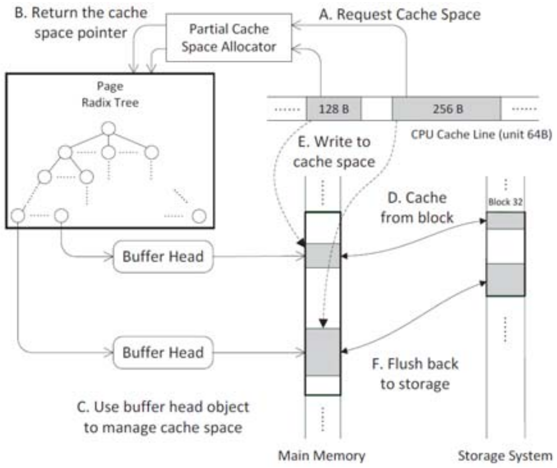
Cell type	SLC	MLC$_{x2}$	MLC$_{x3}$	3D NAND
Bits per cell	1	2	3	3
Page size (KBs)	4	8	8 or 16	16
Block size (pages)	128	256	384	768
Page write latency (μsec.)	300	1300	2500	700
Page read latency (μsec.)	35	75	100	60
Block erase latency (msec.)	0.7	3.8	3	3.5

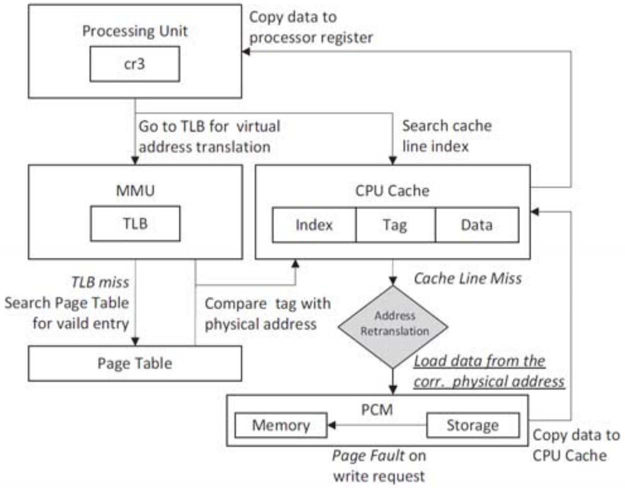


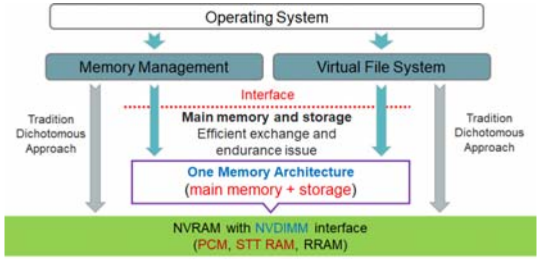
Shuo-Han Chen, Che-Wei Tsao, and Yuan-Hao Chang, "Beyond Address Mapping: A User-Oriented Multi-Regional Space Management Design for 3D NAND Flash Memory," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 36, no. 6, pp. 1286-1299, Jun. 2020.

Union Page Cache for NVM-based Storage

- Observation:
  Existing page cache mechanisms introduce too many unnecessary data movements when NVM is used as both main memory and storage.
- We design a *union page cache* to reduce unnecessary data movement by using both memory and storage space as the page cache space. (IEEE TCAD in 2020, DAC 2018)
  - A partial cache policy to only keep the modified data in a partial page and an in-place read policy to avoid caching data in page cache.
  - An address re-translator is designed to resolve the file mapping issue.
  - With evaluation on ZC702 platform, the energy consumption is reduced by 17%-90% and average read/write latency is reduced by 17%-85%.




Processing Unit
cr3
Copy data to processor register
Go to TLB for virtual address translation
Search cache line index
MMU
TLB
CPU Cache
Index
Tag
Data
TLB miss
Search Page Table for vaild entry
Compare tag with physical address
Cache Line Miss
Address Retranslation
Page Table
Load data from the corr. physical address
PCM
Memory
Storage
Copy data to CPU Cache
Page Fault on write request

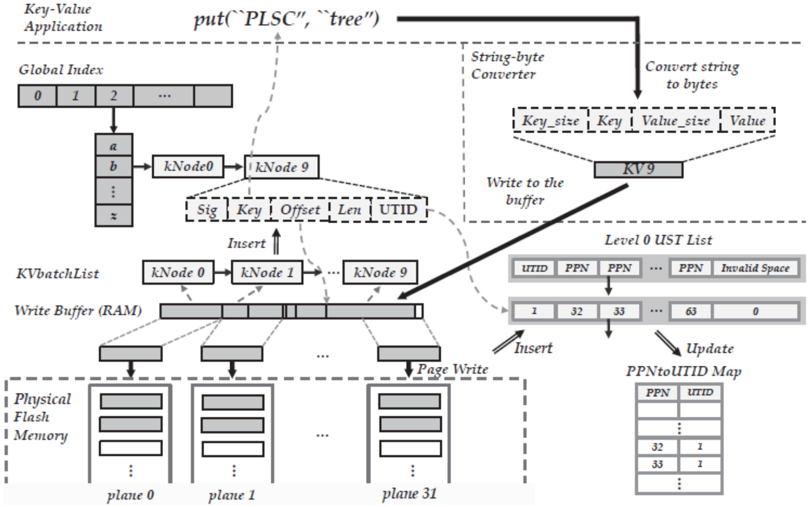


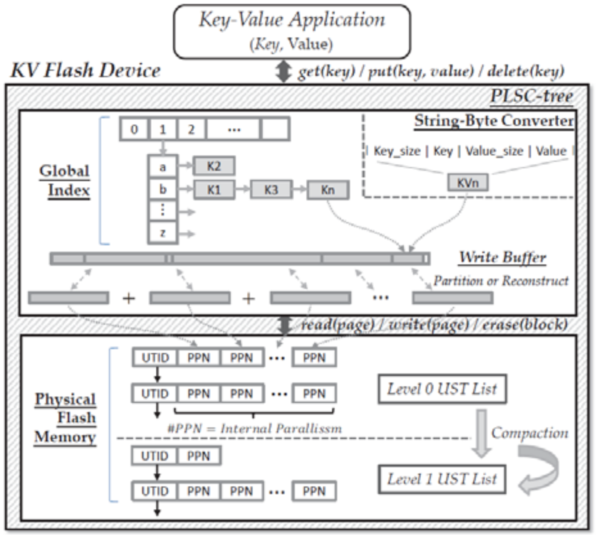
Zynq-7000 ZC702

- Shuo-Han Chen, Tseng-Yi Chen, Yuan-Hao Chang, Hsin-Wen Wei, and Wei-Kuan Shih, "A Partial Page Cache Strategy for NVRAM-Based Storage Devices," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 39, no. 2, pp. 373-386, Feb. 2020.
- Shuo-Han Chen, Tseng-Yi Chen, Yuan-Hao Chang, Hsin-Wen Wei, and Wei-Kuan Shih, "Enabling Union Page Cache to Boost File Access Performance of NVRAM-Based Storage Devices," ACM/IEEE Design Automation Conference (DAC), San Francisco, USA, Jun. 24-28, 2018. **(Top Conference)**

Parallel-Log-Single-Compaction Tree for KVSSD

- Observation: Without re-designing the management strategy of LSM-tree based key-value applications, true potential of SSDs can not be well exploited.
- We proposed a *two-level and flash-friendly key-value management strategy*, specially tailored for key-value solid state drives (KVSSDs) to achieve high device performance. (ASP-DAC in 2020)
  - **First level:** Maximize the write performance by leveraging the internal parallelism of SSDs.
  - **Second level:** Alleviate the internal recycling overheads of SSDs by reorganizing and storing key-value pairs with a smaller storage unit.



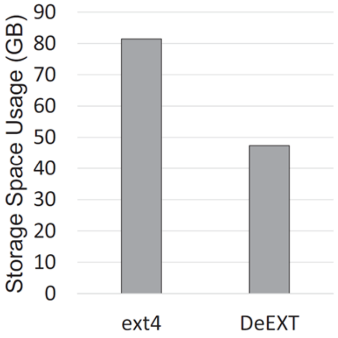


Yen-Ting Chen, Ming-Chang Yang, Yuan-Hao Chang, and Wei-Kuan Shih, "Parallel-Log-Single-Compaction-Tree: Flash-Friendly Two-Level Key-Value Management in KVSSDs," ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC), Beijing, China, Jan. 13-16, 2020.

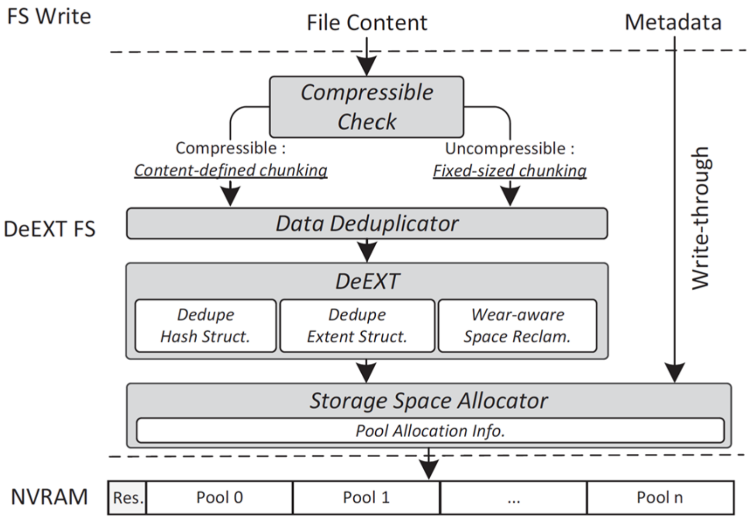
Dual-Chunking Deduplication for NVM-based Storage

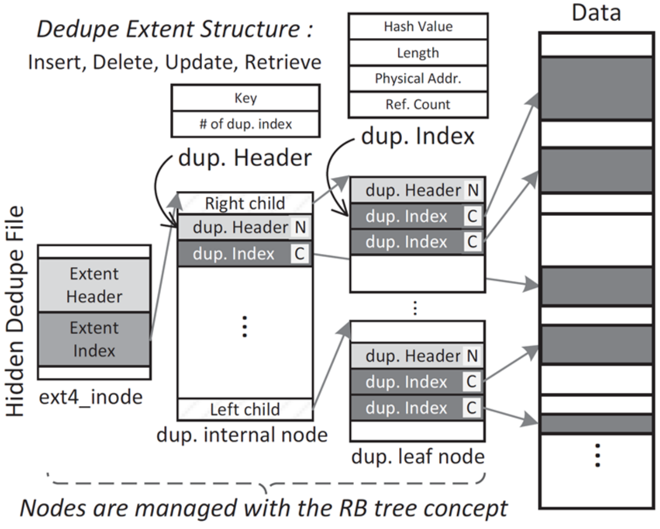
(ASP-DAC 2020)

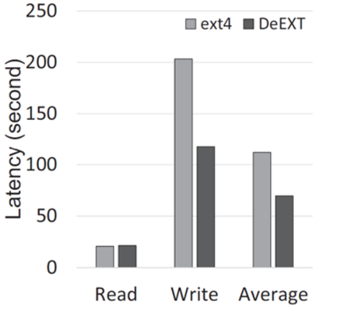
- Observation
  - The high manufacturing cost of NVRAM greatly lowers the incentive of deploying NVRAM in consumer electronics due to the consideration of profitability.
- Goal
  - We proposed to resolve the profitability issue via avoiding storing duplicate data on NVRAM becomes a crucial task for lowering the deployment cost of NVRAM.
- Main Idea
  - Integrating the deduplication into the design of ext4, DeEXT
    - *Dedupe Extent Structure*
  - Proposing the concept of dual-chunking data deduplication
    - *For data with high deduplication ratio -> Fixed-size chunking*
    - *For data with low deduplication ratio -> Content defined chunking*



Space Usage Reduction: 41.84%







Access Performance Improvement: 37.76%

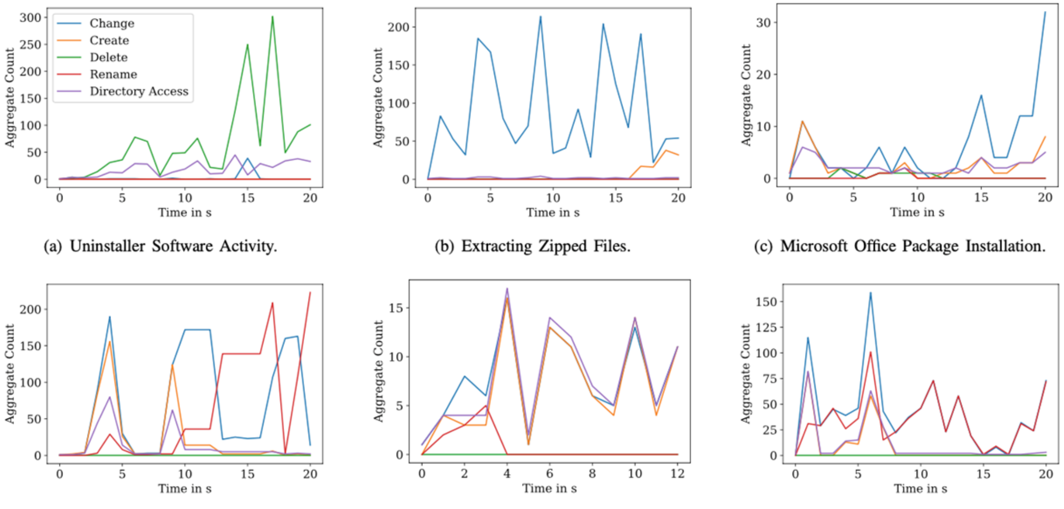
Shuo-Han Chen, Yu-Pei Liang, Yuan-Hao Chang, Hsin-Wen Wei and Wei-Kuan Shih, "Boosting the Profitability of NVRAM-based Storage Devices via the Concept of Dual-Chunking Data Deduplication," ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC), Beijing, China, Jan. 13-16, 2020.

4. Others

DeepGuard – User-behavior Analysis for Ransomware Detection

(ISI 2020)

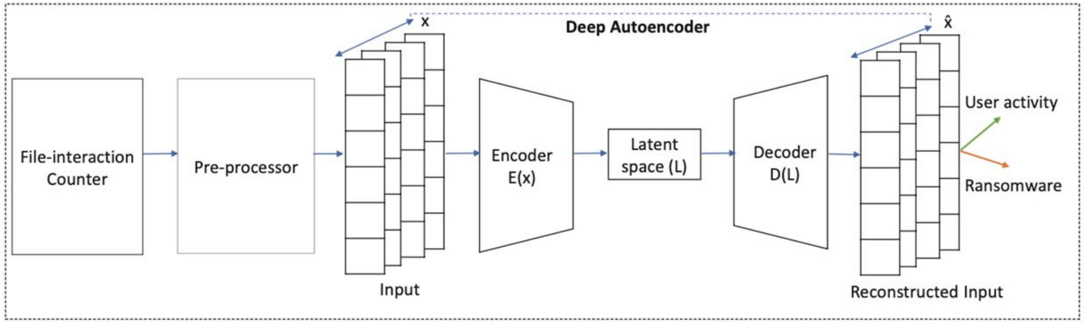
- Observation
  - Normal Processes: User activity barely changes specific file-related operations at a time.
  - Ransomware: Most of the file-related operations are present with a high aggregate count in all cases for ransomware.
- Goal
  - DeepGuard: Deep Generative User-behaviors Analytics for Ransomware Detection
- Goal & Main Idea
  - Log the file-interaction pattern of typical user activity and pass it through deep generative autoencoder architecture to recreate the input.
  - Train the proposed deep generative autoencoder architecture to learn the file access behaviors.



(a) Uninstaller Software Activity. (b) Extracting Zipped Files. (c) Microsoft Office Package Installation.

(d) Ryuk Ransomware Activity. (e) Petya Ransomware Activity. (f) Sodinokibi Ransomware Activity.

Normal Processes vs Ransomware



Architecture of DeepGuard

- Gaddisa Olani Ganfure, Chun-Feng Wu, Yuan-Hao Chang, and Wei-Kuan Shih, "DeepGuard: Deep Generative User-behavior Analytics for Ransomware Detection,"IEEE International Conference on Intelligence and Security Informatics (ISI), Nov. 9-10, 2020.

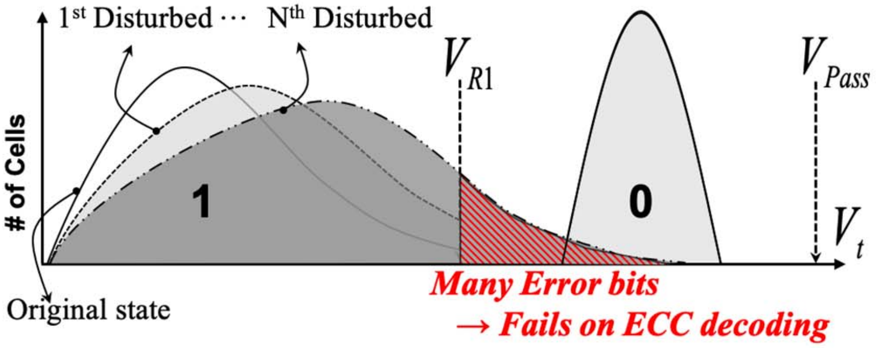
Research Summary 2019

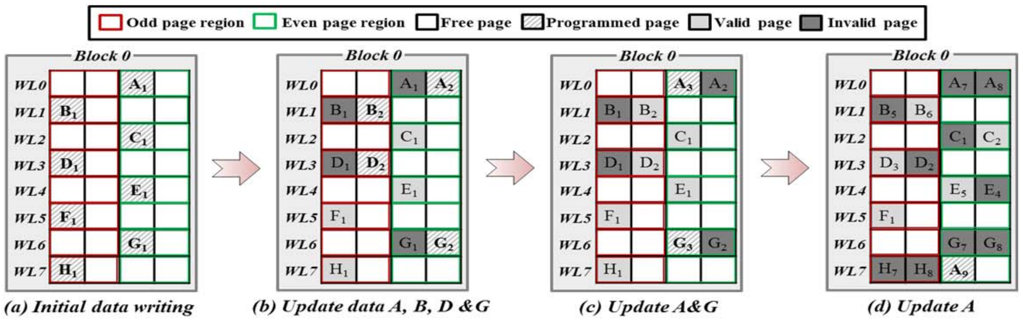
1. Storage Systems - Flash Drives and SMR Disks

Instantaneous Sanitization for Flash

(ICCAD 2019)

- Observation:
  - In the past, lots of studies and works tell us to keep the Vt distribution of cells with the different data status as wide as possible for improving the reliability
  - However, there exists the possibility to realize the efficient sanitization if we properly control and utilize cells' Vt distribution and disturbance properties
- We propose *Instantaneous Sanitization and Recycling Programming Designs* to achieve efficient sanitization by smartly exploiting cells' Vt distribution and disturbance. Instantaneous Sanitization Design: With the purposely generated overlapping, the page data can be instantaneously sanitized by being disturbed exactly once.
  - Recycling Programming Design: To reduce the needs of frequently invoking GC process by reusing the sanitized pages.
  - The evaluation on proposed scheme is based on real flash chips. The device-level evaluation shows the write response time improves excessively by 65.81% to 86.91%.



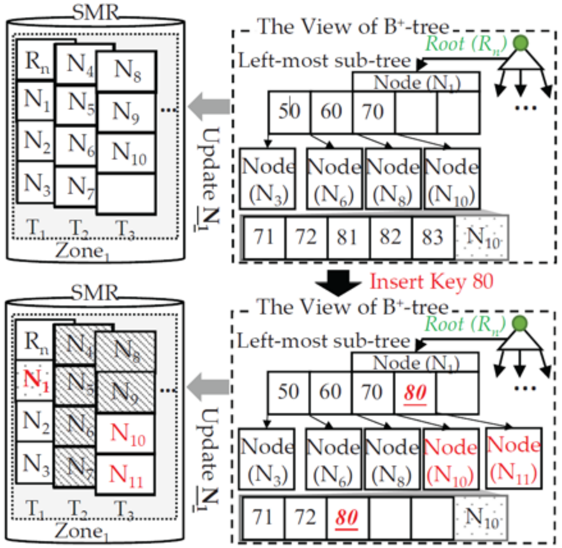

Odd page region
Even page region
Free page
Programmed page
Valid page
Invalid page
Block 0
WL0 WL1 WL2 WL3 WL4 WL5 WL6 WL7
A1 B1 C1 D1 E1 F1 G1 H1
A1 A2 B1 B2 C1 D1 D2 E1 F1 G1 G2 H1
A3 A2 B1 B2 C1 D1 D2 E1 F1 G3 G2 H1
A7 A8 B5 B6 C1 C2 D3 D2 E5 E4 F1 G7 G8 H7 H8 A9

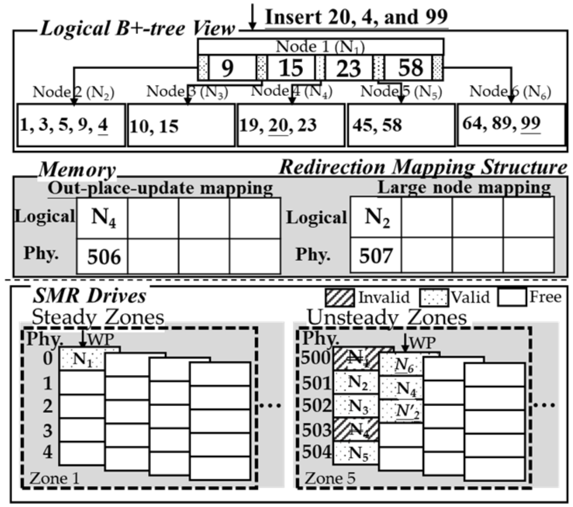
(a) Initial data writing (b) Update data A, B, D &G (c) Update A&G (d) Update A

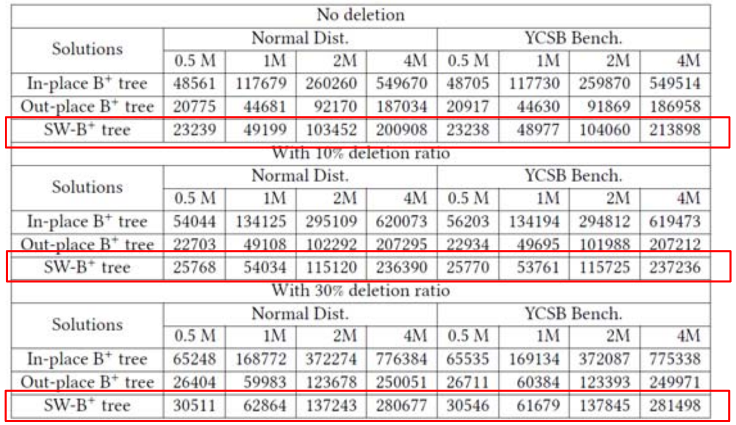
Wei-Chen Wang, Ping-Hsien Lin, Yung-Chun Li, Chien-Chung Ho, Yu-Ming Chang, and Yuan-Hao Chang, "Toward Instantaneous Sanitization through Disturbance-induced Errors and Recycling Programming over 3D Flash Memory," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), Westminster, CO, USA, Nov. 4-7, 2019. (Acceptance rate: 23.8%(94/394)) **(Top Conference)**

Sequential-Write B+-tree for SMR Drives

- Observation
  - In B$^+$-tree structure, frequent leaf node updates will hurts SMR storage performance because of the sequential write constraint of SMR technology.
- Goal
  - We proposed Sequential-write B+-tree Structure to update a B+-tree node over host-managed SMR drives without losing storage performance. (ACM TECS in 2019, CODES+ISSS 2019)
- Main Idea
  - To separately manage leaf and non-leaf nodes in non-steady and steady zones.
  - To enlarge a leaf node size for reducing the frequency of garbage collection process.







Solutions	Normal Dist.				YCSB Bench.			
No deletion								
	0.5 M	1M	2M	4M	0.5 M	1M	2M	4M
In-place B$^+$ tree	48561	117679	260260	549670	48705	117730	259870	549514
Out-place B$^+$ tree	20775	44681	92170	187034	20917	44630	91869	186958
SW-B$^+$ tree	23239	49199	103452	200908	23238	48977	104060	213898
With 10% deletion ratio								
	0.5 M	1M	2M	4M	0.5 M	1M	2M	4M
In-place B$^+$ tree	54044	134125	295109	620073	56203	134194	294812	619473
Out-place B$^+$ tree	22703	49108	102292	207295	22934	49695	101988	207212
SW-B$^+$ tree	25768	54034	115120	236390	25770	53761	115725	237236
With 30% deletion ratio								
	0.5 M	1M	2M	4M	0.5 M	1M	2M	4M
In-place B$^+$ tree	65248	168772	372274	776384	65535	169134	372087	775338
Out-place B$^+$ tree	26404	59983	123678	250051	26711	60384	123393	249971
SW-B$^+$ tree	30511	62864	137243	280677	30546	61679	137845	281498

Performance improvement: 55% on average

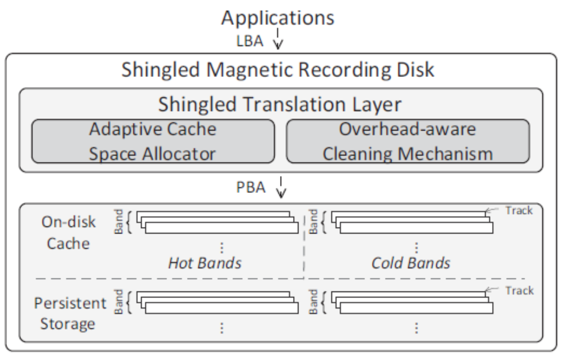
- Yu-Pei Liang, Tseng-Yi Chen, Yuan-Hao Chang, Shuo-Han Chen, Kam-Yiu Lam, Wei-Hsin Li, and Wei-Kuan Shih, "Enabling Sequential-write-constrained B+-tree Index Scheme to Upgrade Shingled Magnetic Recording Storage Performance ACM Transactions on Embedded Computing Systems (TECS), vol. 18, no. 5s, pp. 66:1-66:20, Oct. 2019. (Integrated with ACM/IEEE CODES+ISSS'19)

- Yu-Pei Liang, Tseng-Yi Chen, Yuan-Hao Chang, Shuo-Han Chen, Kam-Yiu Lam, Wei-Hsin Li, and Wei-Kuan Shih, "Enabling Sequential-write-constrained B+-tree Index Scheme to Upgrade Shingled Magnetic Recording Storage Performance," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), New York, NY, USA, Oct. 13-18, 2019. (Journal Track, Integrated with ACM TECS) **(Top Conference)**

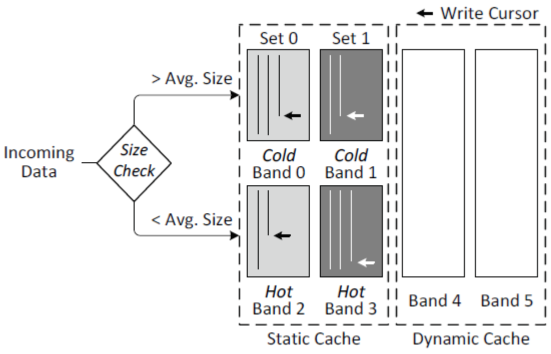
Sequential-Write-Constrained Cache Management for SMR Drives

(Elsevier JSA in 2019, SAC 2019)

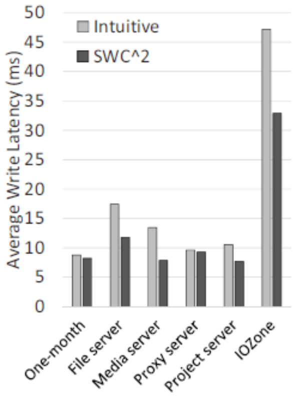
- Observation
  - In SMR drives, different input data sequence may influence the usage of log-based on-disk cache
- Goal
  - We propose a sequential-write-constrained cache (SWC^2) management to mitigate the write amplification issue of SMR drives with the cached data hotness consideration.
- Main Idea
  - Managing the on-disk cache by dynamic space allocation through the set-associative method
  - Separating the hot/cold data to further increase the usage of on-disk cache


Applications
LBA
Shingled Magnetic Recording Disk
Shingled Translation Layer
Adaptive Cache Space Allocator
Overhead-aware Cleaning Mechanism
PBA
On-disk Cache
Band
Track
Hot Bands
Cold Bands
Persistent Storage

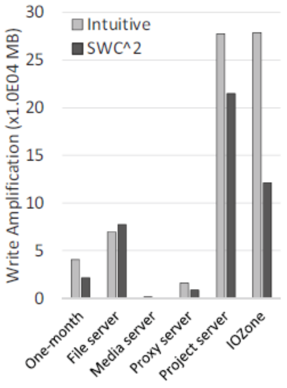
System architecture of SWC^2



Cache Band allocation of SWC^2



Average Write Latency



Write amplification

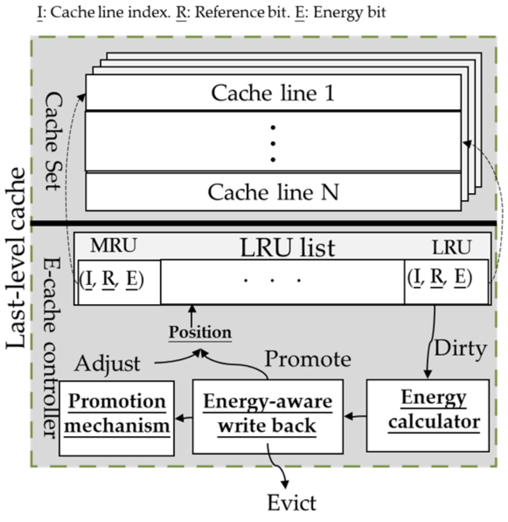
- Yu-Pei Liang, Shuo-Han Chen, Yuan-Hao Chang, Yong-Chin Lin, Hsin-Wen Wein, and Wei-Kuan Shih, "Mitigating Write Amplification Issue for SMR Drives via the Design of Sequential-Write-Constrained Cache Management," Elsevier Journal of Systems Architecture (JSA), vol. 99, pp. 101634, Oct. 2019.
- Shuo-Han Chen, Yong-Ching Lin, Yuan-Hao Chang, Ming-Chang Yang, Tseng-Yi Chen, Hsin-Wei Wei, and Wei-Kuan Shih, "A New Sequential-Write-Constrained Cache Management to Mitigate Write Amplification for SMR Drives," ACM Symposium on Applied Computing (SAC), Limassol, Cyprus, Apr. 8-12, 2019.

2. One/Unified Memory System – NVM Main Memory and Storage

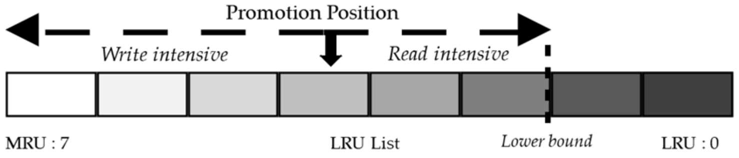
Energy-aware Write-back for STT-MRAM Main Memory

(ISLPED 2019)

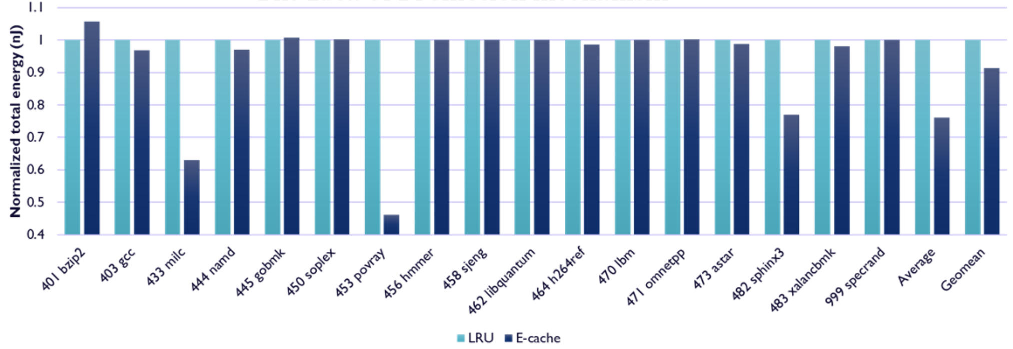
- Observation
  - Traditional cache replacement algorithm does not consider the characteristics of MLC STT-RAM.
  - MLC STT-RAM has different power consumption while different written bits patterns are written into MLC STT-RAM.
- Goal
  - Propose an energy-aware cache replacement policy with the following considerations:
    - Asymmetric write energy of STT-RAM
    - Low-complexity and lightweight resource footprint.
- Main Idea
  - Predict the write back energy consumption by considering the data pattern in the cache
  - A promotion mechanism was proposed to avoid the high write-back energy data will be checked again shortly



Energy-aware Write Back Cache replacement



The Idea of Promotion mechanism

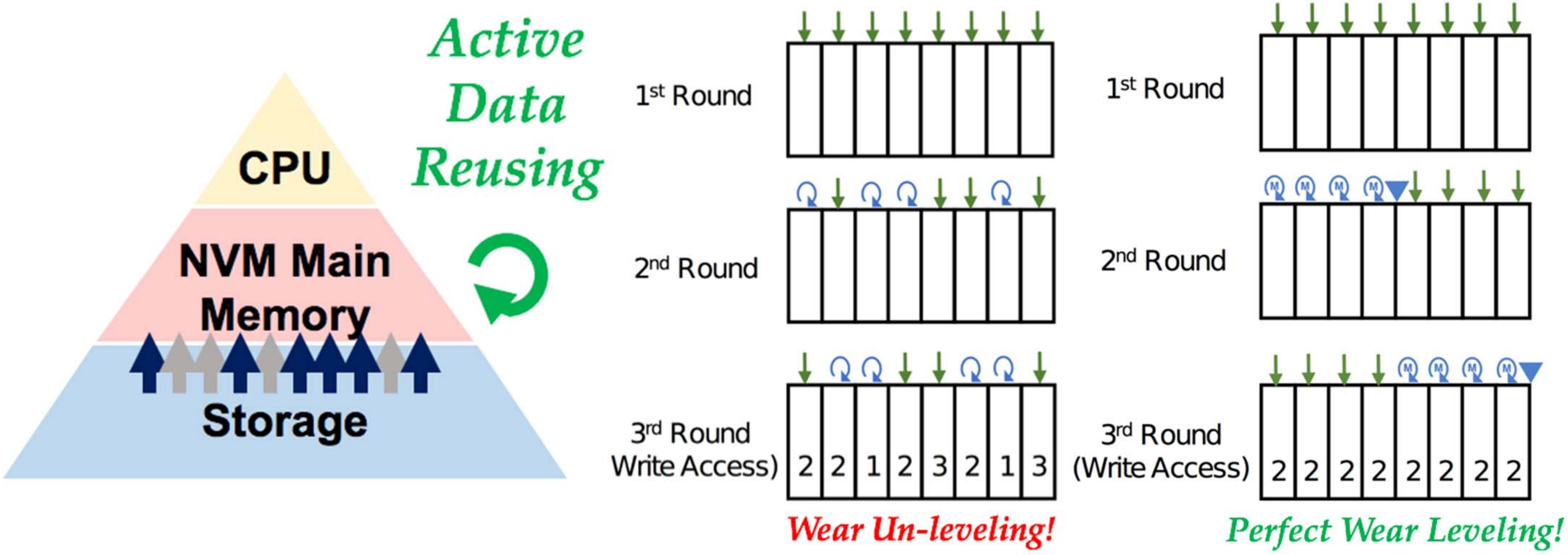


Result of Energy consumption

Yu-Pei Liang, Tseng-Yi Chen, Yuan-Hao Chang, Shuo-Han Chen, Pei-Yu Chen, and Wei-Kuan Shih, "Rethinking Last-level-cache Write-back Strategy for MLC STT-RAM Main Memory with Asymmetric Write Energy," ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), Lausanne, Switzerland, Jul. 29-31, 2019. **(Top Conference)**

NVM-friendly Bagging for Random Forest

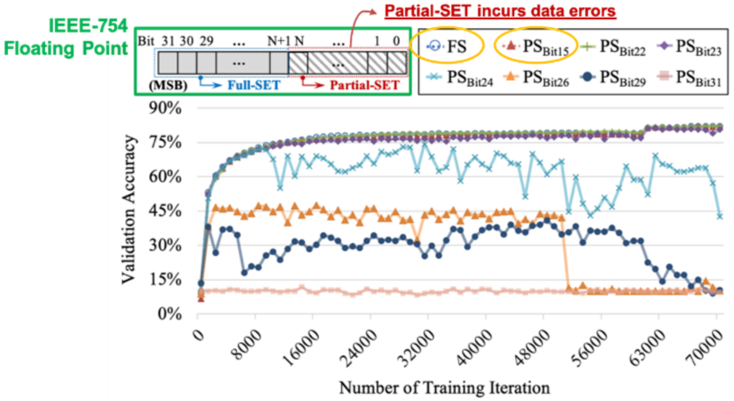
- We propose an *NVM-friendly bagging strategy* to exploit the concept of "reusing the sampled data" to trade the "***randomness***" of the sampled data for the ***reduced data movement*** between different layers in the memory hierarchy.
  - Goal is to trade the "randomness" of a random forest for endurance and performance improvements for NVM main memory.
  - Observe that during building (or training) a random forest, some data may be resampled (i.e., reused) to form decision trees in any two consecutive rounds of bagging.
  - Reduce 72% of writes during random forest construction.

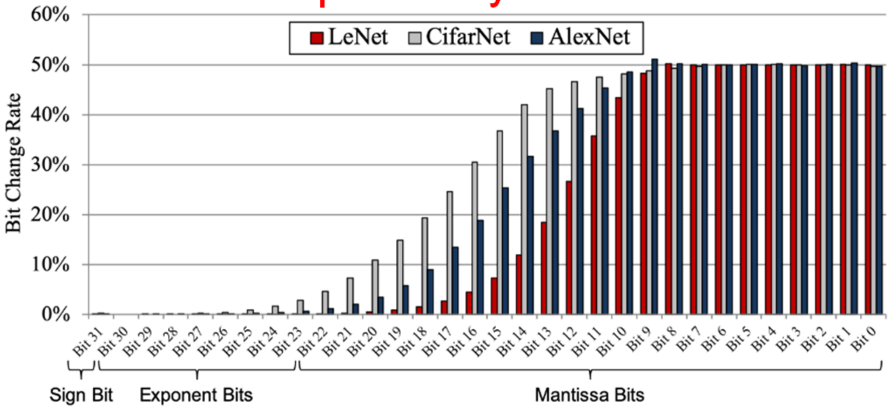


Yu Ting Ho, Chun-Feng Wu, Ming-Chang Yang, Tseng-Yi Chen, and Yuan-Hao Chang, "Replanting Your Forest: NVM-friendly Bagging Strategy for Random Forest," IEEE Nonvolatile Memory Systems and Applications Symposium (NVMSA), Hangzhou, China, Aug. 18-21, 2019. **(Best Paper Award)**

Achieving Lossless Accuracy with Lossy Programming for Neural Network Training

- Observation: (ACM TECS in 2019, CODES+ISSS 2019 – **Best Paper Award**)
  Combining lossy-SET operations of NVM and approximate computing of neural network (NN) seems to be a great solution, but we also find that it is very challenging to combine them if we take performance, endurance and NN accuracy into the consideration simultaneously.
- We propose a *Data-Aware Programming Design* to exploit Dual-SET operations to program NN data from the unique viewpoints of data contents and data flow.
  - The layer-aware SET policy and the bit-aware dual-SET policy are proposed to efficiently program intermediate data, weights and biases respectively.
  - The buffered marching-based wear leveling policy is proposed to balance the asymmetric damages of different data on NVM.
  - The experiment results show that the proposed design could improve the average memory access latency up to 4.3x and enhance the lifetime up to 3.4x.
- The first CODES best paper award from Taiwan in the past 28 years.





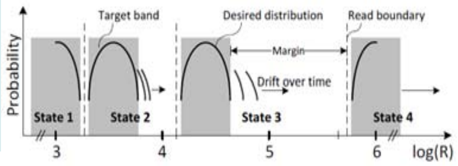
- Wei-Chen Wang, Yuan-Hao Chang, Tei-Wei Kuo, Chien-Chung Ho, Yu-Ming Chang, and Hung-Sheng Chang, "Achieving Lossless Accuracy with Lossy Programming for Efficient Neural-Network Training on NVM-Based Systems," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), New York, NY, USA, Oct. 13-18, 2019. (Journal Track, Integrated with **ACM TECS**) **(Best Paper Award - Top Conference)**

Multi-Write Modes for NVM-based File System

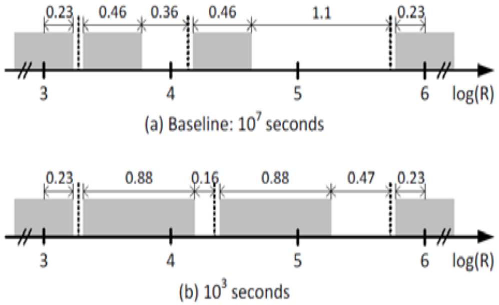
- We propose a *multi-write-mode strategy* to boost performance of journaling FS on NVM-based storage without expiring data retention. (IEEE TVLSI in 2019, ISLPED 2018)
  - Retention Monitoring to track the data's retention time
  - Incremental Retention to gradually prolong the data's retention time
- The proposed strategy is integrated into EXT3 and the results show that the write latency can be reduced by an avg. of 45% with 50% of compressible data.

Light write : Less wear & shorter retention time
Heavy write : Higher wear & longer retention time

Metadata & hot data are with light writes and checkpoint back to data area.
Cold data are with heavy writes and checkpoint in commit place.

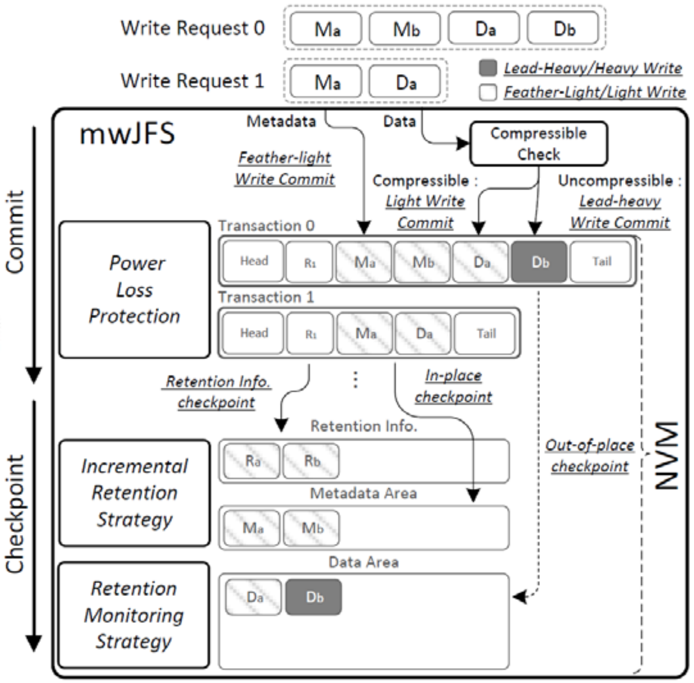


Non-Volatility (s)	Target Band log(R)	Write Speedup
10^7	0.46	Baseline
10^6	0.56	1.2 ×
10^5	0.67	1.5 ×
10^4	0.77	1.7 ×
10^3	0.88	1.9 ×
10^2	0.98	2.1 ×



(a) Baseline: 10^7 seconds

(b) 10^3 seconds

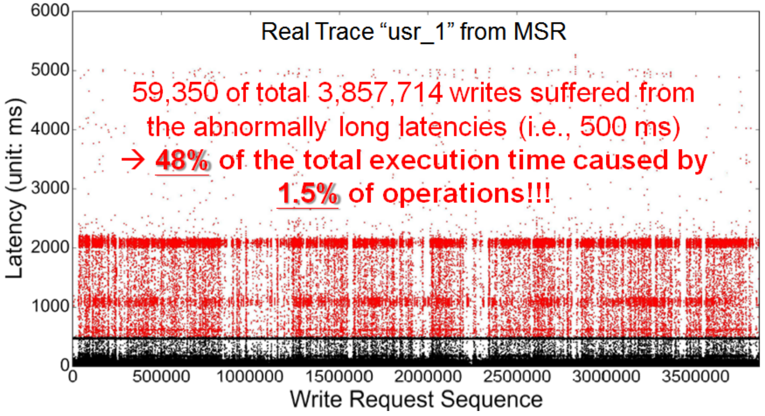


- Shuo-Han Chen, Yuan-Hao Chang, Yu-Ming Chang, and Wei-Kuan Shih, "mwJFS: A Multi-Write-Mode Journaling File System for MLC NVRAM Storages," IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 27, no. 9, pp. 2060-2073, Sep. 2019.
- Shuo-Han Chen, Yuan-Hao Chang, Tseng-Yi Chen, Yu-Ming Chang, Pei-Wen Hsiao, Hsin-Wen Wei, and Wei-Kuan Shih, "Enhancing the Energy Efficiency of Journaling File System via Exploiting Multi-Write Modes on MLC NVRAM," ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), Bellevue, USA, Jul. 23-25, 2018. **(Top Conference)**

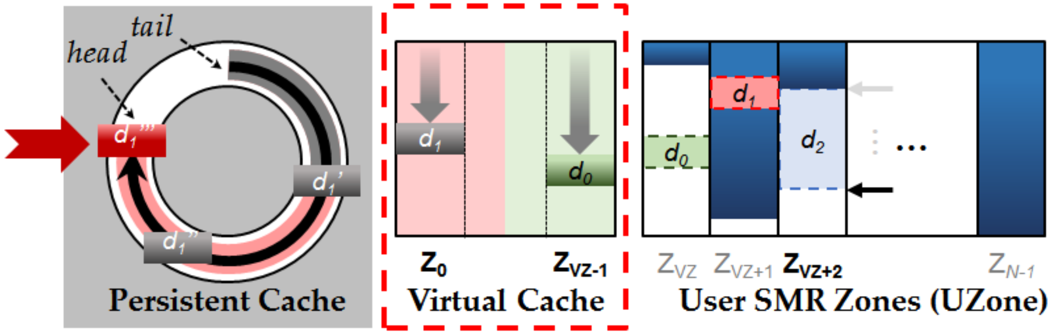
Virtual Persistent Cache for SMR Drives

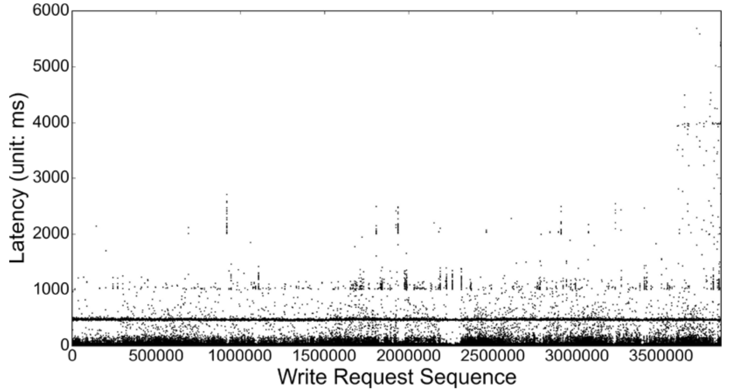
with Seagate & David Du

- We propose a *virtual persistent cache* to emulate an SMR as a HDD-like drive. (IEEE TC in 2019, ICCAD 2017)
  - Avoid storing data of different update frequencies in Persistent Cache
  - Read-merge-write operations are minimized
- Implemented over Seagate's Shingled-Magnetic-Recording Drives.
- Impact: Make low-cost SMR have the performance of HHDs


Real Trace "usr_1" from MSR

59,350 of total 3,857,714 writes suffered from the abnormally long latencies (i.e., 500 ms) → 48% of the total execution time caused by 1.5% of operations!!!

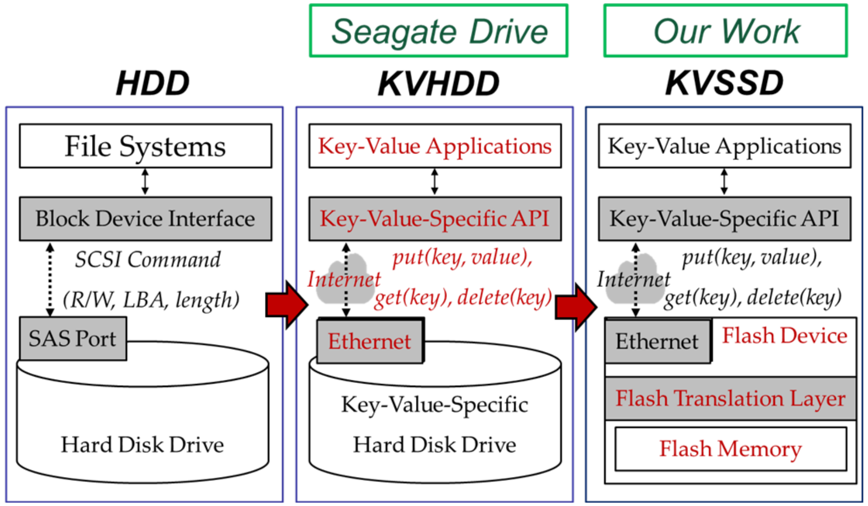


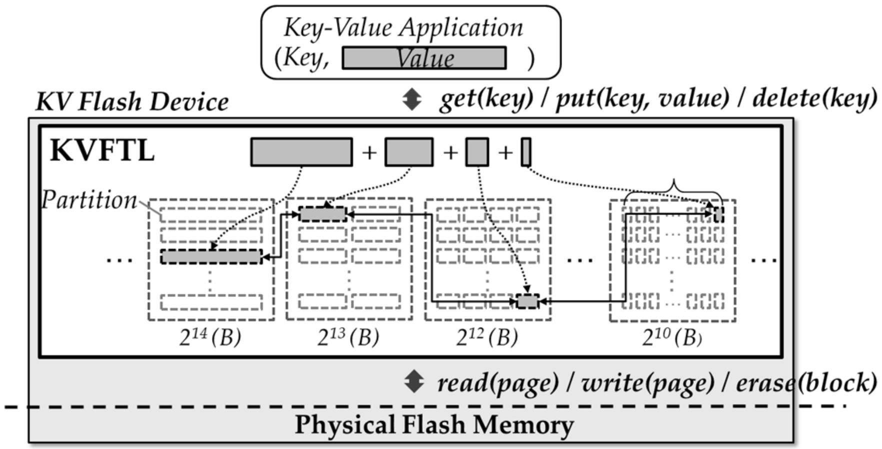


- Ming-Chang Yang, Yuan-Hao Chang, Fenggang Wu, Tei-Wei Kuo, and David Hung-Chang Du, "On Improving the Write Responsiveness for Host-Aware SMR Drives," IEEE Transactions on Computers (TC), vol. 68, no. 1, pp. 111-124, Jan. 2019.
- Ming-Chang Yang, Yuan-Hao Chang, Fenggang Wu, Tei-Wei Kuo, and David Hung-Chang Du, "Virtual Persistent Cache: Remedy the Long Latency Behavior of Host-Aware Shingled Magnetic Recording Drives," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), Irvine, California, USA, Nov. 13-16, 2017. **(Top Conference)**

KVFTL for KV-SSD

- KV-store has gained popularity in large-scale and data-driven applications.
- Observation: Variable-sized KV objects are incompatible with fixed-sized flash pages, resulting in low space utilization and write amplification.
- We are the first team to propose a *key-value FTL* to improve the performance and space utilization of the key-value solid state drives (KVSSDs). (IEEE TCAD in 2019)
  - The main idea is to *partition* KV objects into various fixed-sized chunks, and chunks of the same size are packed into the same page.





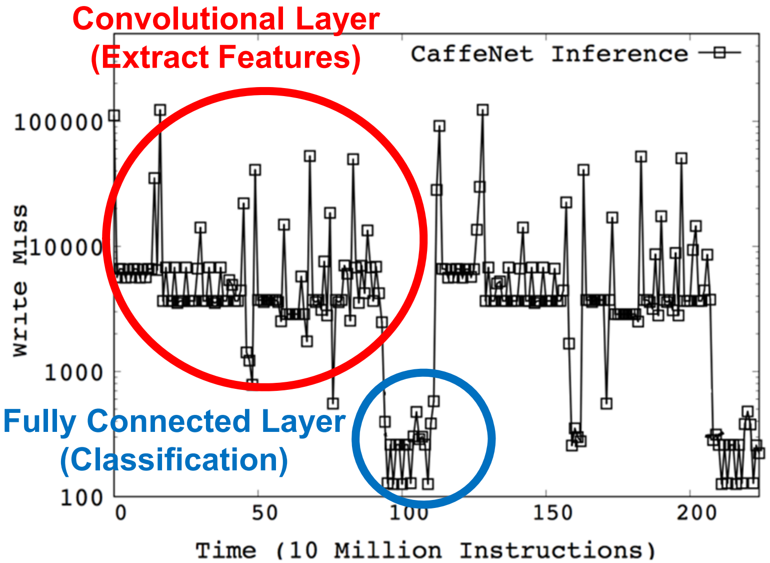
Yen-Ting Chen, Ming-Chang Yang, Yuan-Hao Chang, Tseng-Yi Chen, Hsin-Wen Wei, and Wei-Kuan Shih, "Co-optimizing Storage Space Utilization and Performance for Key-Value Solid State Drives," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 38, no. 1, pp. 29-42, Jan. 2019.

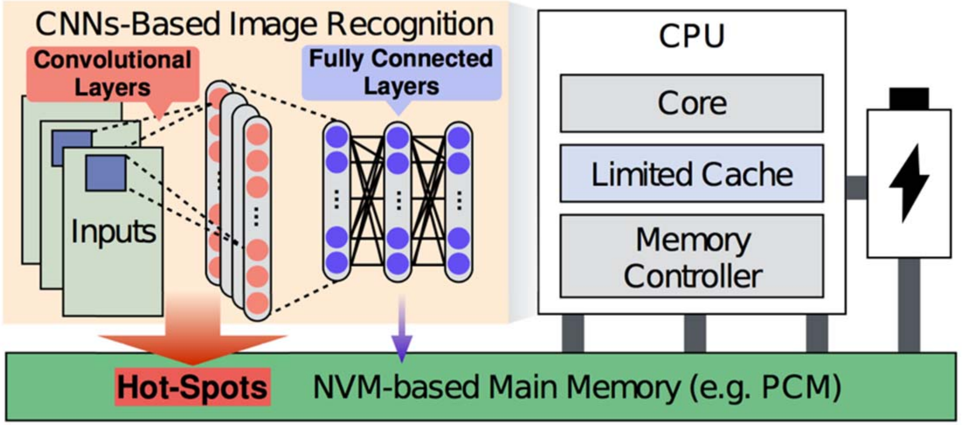
Research Summary 2018

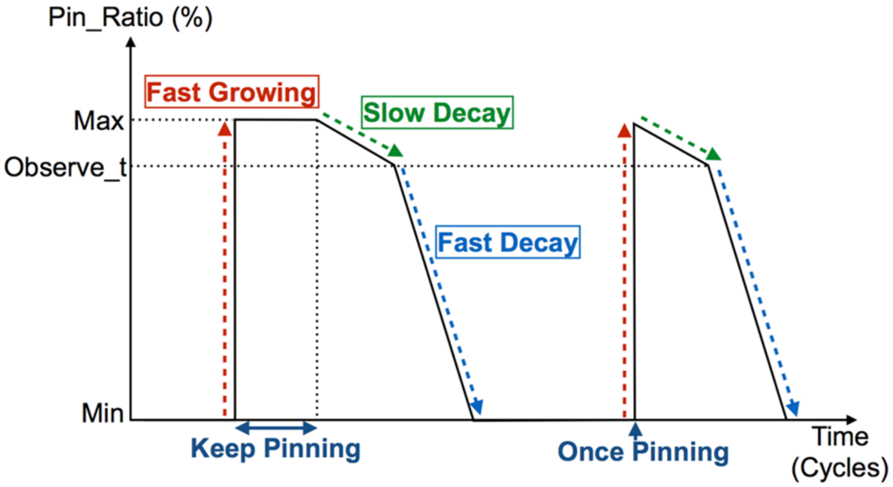
1. One/Unified Memory System: Using NVM as both Main Memory and Storage

Self-Bouncing to Suppress CNN Hot-Spots

- We proposes a *CNN-aware self-bouncing pinning strategy* to efficiently suppress the write hot-spots. (IEEE TCAD in 2018, EMSOFT 2018)
  - Investigate the memory access pattern induced by CNN computing and observe the "write hot-spot" pattern.
  - This is among the first to use the existing CPU cache pinning function to improve the NVM lifetime
  - Propose a self-bouncing pinning strategy by leveraging the iterative access pattern of CNN.



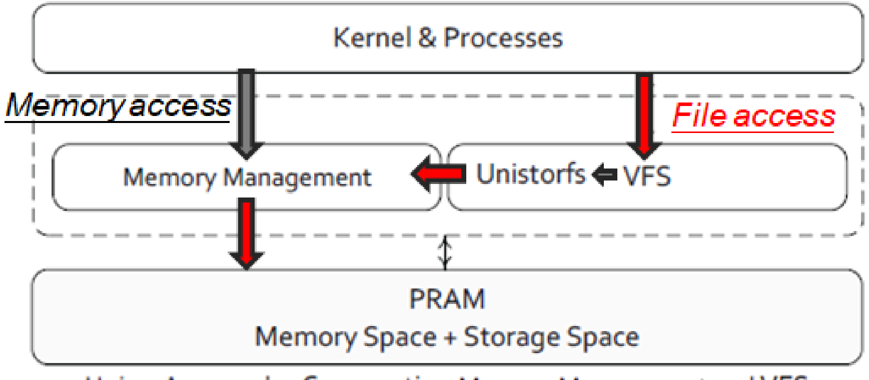




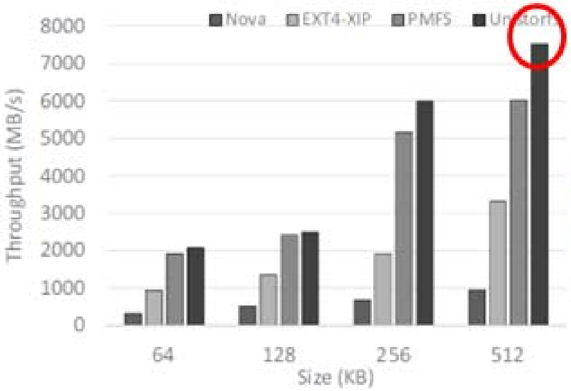
- Chun-Feng Wu, Ming-Chang Yang, Yuan-Hao Chang, and Tei-Wei Kuo, "Hot-Spot Suppression for Resource-Constrained Image Recognition Devices with Non-Volatile Memory," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 37, no. 11, pp. 2567-2577, Nov. 2018. (Integrated with ACM/IEEE EMSOFT'18)
- Chun-Feng Wu, Ming-Chang Yang, Yuan-Hao Chang, and Tei-Wei Kuo, "Hot-Spot Suppression for Resource-Constrained Image Recognition Devices with Non-Volatile Memory," ACM/IEEE International Conference on Embedded Software (EMSOFT), Torino, Italy, Sep. 30 - Oct. 5, 2018. (Journal Track, Integrated with IEEE TCAD) **(Top Conference)**

Union Storage File System (UnistorFS)

- We propose a *union storage file system (UnistoreFS)* to fully exploit the benefit of NVM as both main memory and storage. (ACM TOS in 2018)
- Storage space actually resides NVM (i.e., memory):
  - (1) There is no need to let file system control its own storage space.
  - (2) We propose to let memory manager manage memory (i.e., NVM) directly and let file system allocate space from memory manager.
- UnistorFS realizes the “resource sharing” between main memory and storage without partition nor logical boundary.
- UnistorFS eliminates unnecessary memory accesses and outperform other PRAM-based file systems for 0.2-8.7 times.



Union Approach : Cooperating Memory Management and VFS



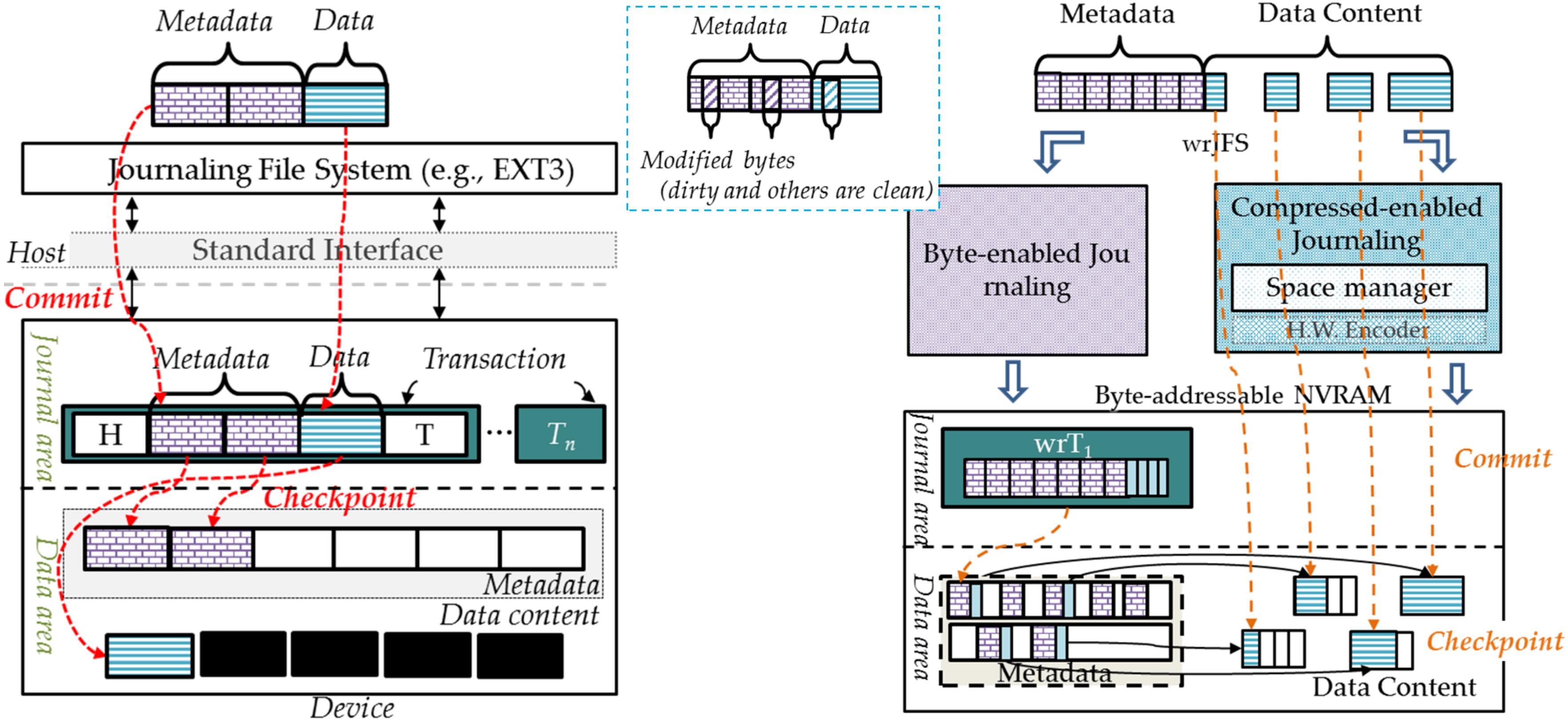
Random write

Demo: https://www.youtube.com/watch?v=A8jH8dmYFSE

- Shuo-Han Chen, Tseng-Yi Chen, Yuan-Hao Chang, Hsin-Wen Wei, and Wei-Kuan Shih, "UnistorFS: A Union Storage File System Design for Resource Sharing between Memory and Storage on Persistent RAM based Systems," ACM Transactions on Storage (TOS), vol. 14, no. 1, pp. 3:1-3:22, Feb. 2018.

Write-reduction for Journaling FS

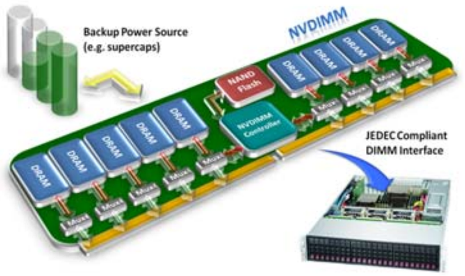
- We propose a write-reduction strategy to lower energy consumption by reducing the amount of writes performed by journaling FS. (IEEE TC in 2018, DAC 2017)
- Partial modification and compression are implemented into EXT4 to reduce writes.
  - Block-aligned commit in journaling is converted into ***byte-aligned commit***.
  - Propose the idea of "***in-place checkpoint***"

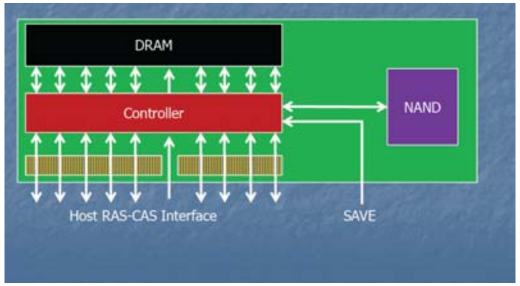


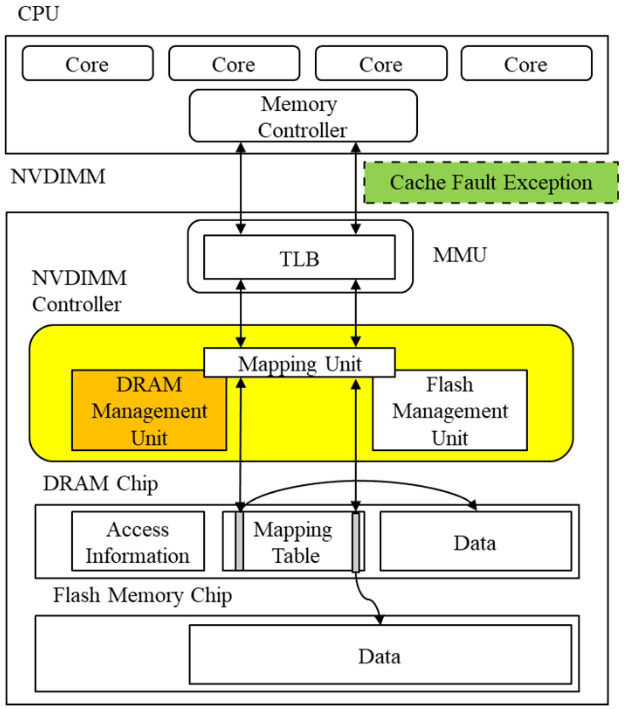
- Tseng-Yi Chen, Yuan-Hao Chang, Shuo-Han Chen, Chih-Ching Kuo, Ming-Chang Yang, Hsin-Wen Wei, and Wei-Kuan Shih, "wrJFS: A Write-Reduction Journaling File System for Byte-addressable NVRAM," IEEE Transactions on Computers (TC), vol. 67, no. 7, pp. 1023-1038, Jul. 2018.
- Tseng-Yi Chen, Yuan-Hao Chang, Shuo-Han Chen, Chih-Ching Kuo, Ming-Chang Yang, Hsin-Wen Wei, and Wei-Kuan Shih, "Enabling Write-Reduction Strategy for Journaling File Systems over Byte-addressable NVRAM," ACM/IEEE Design Automation Conference (DAC), Austin, Texas, USA, Jun. 18-22, 2017. **(Top Conference)**

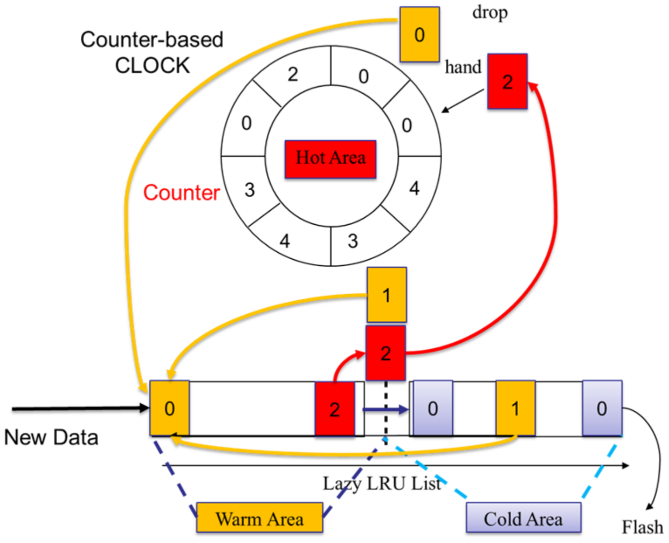
Boosting NVDIMM Performance with ITRI

- A *light-weight caching* to improve the performance of NVDIMM main memory by (1) reducing management overheads and (2) reducing writes to flash memory without sacrificing DRAM hit ratio (IEEE TVLSI in 2018)
- The conception of "*distillation*" is adopted to separate data more precisely, so as to avoid evicting hot data mistakenly. (76% perf. improvement, compared to CLOCK-pro)







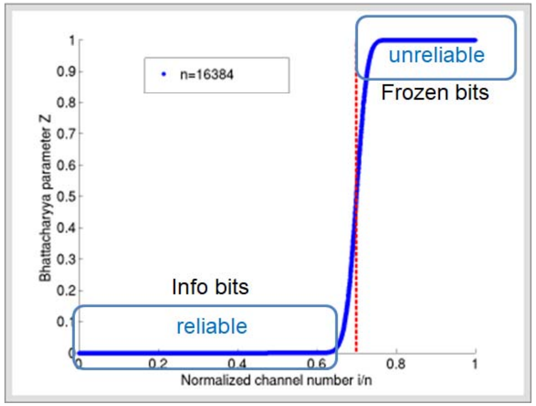


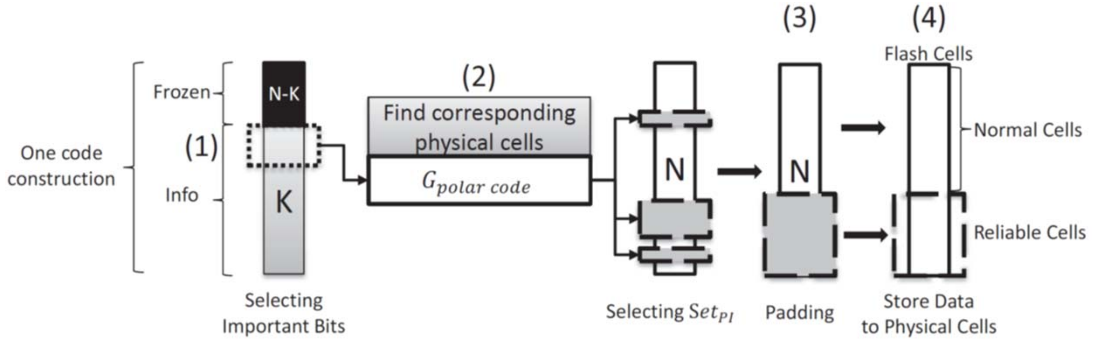
Che-Wei Tsao, Yuan-Hao Chang, and Tei-Wei Kuo, "Boosting NVDIMM Performance with a Light-Weight Caching Algorithm," IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 26, no. 8, pp. 1518-1530, Aug. 2018.

2. Storage Systems - Flash Drives and SMR Disks

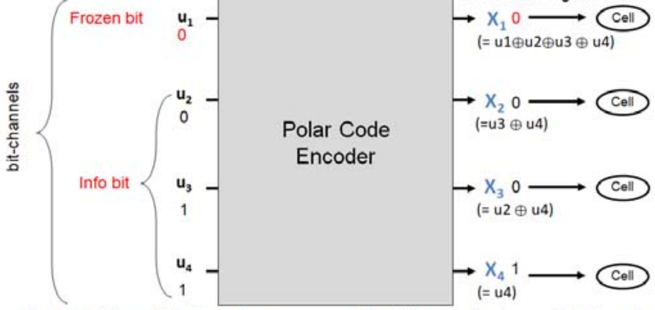
Proactive Channel Adjustment for Polar Code

- We propose a *proactive channel adjustment (PCA)* to enable polar code as the error correction code of flash storages (DAC 2018)
  - Observation:
    1. Polar code can achieve the Shannon limit under a given Channel quality.
    2. Each flash cell can be considered a channel, but each cell has a different quality and each cell's quality is changed after erases.
    → Need a channel ranking for each page and change ranking after erases.
  - We are *the pioneer* that proposes to use a single ranking for all the cells to enable polar code in storage devices.
  - The proposed PCA propose the concept of "*important bits*" and protect these bits to avoid channel re-ranking proactively.





Good ones get better and the bad ones get worse.

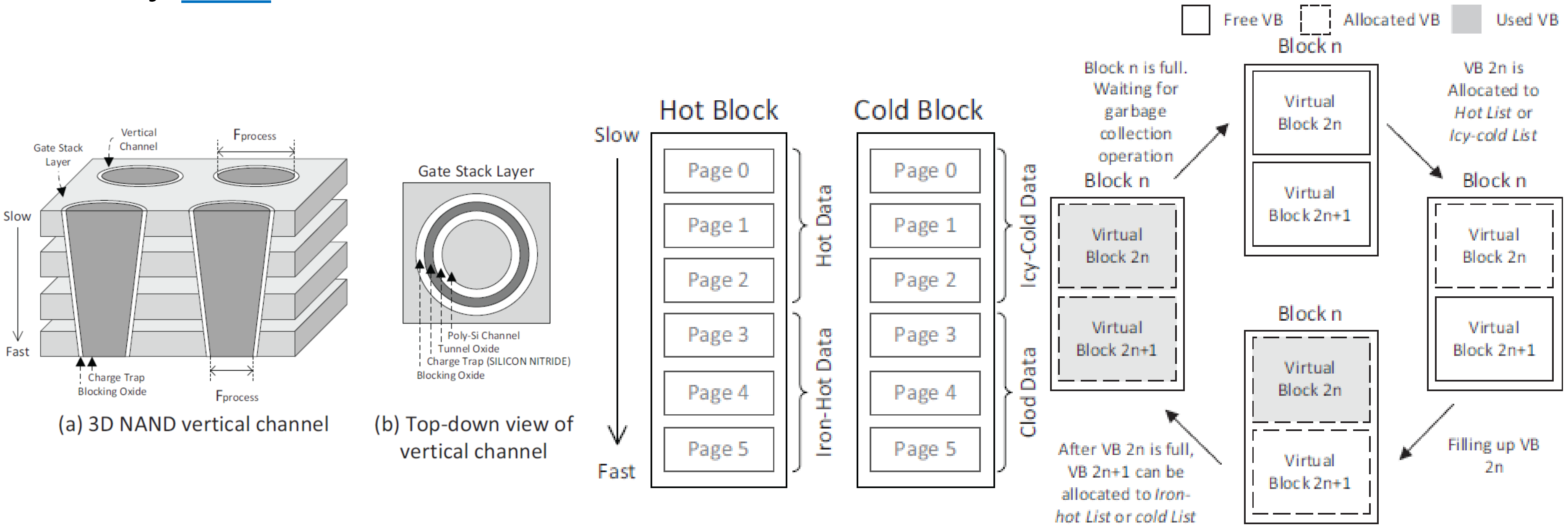


Ranking of these **bit-channels**: u_4 , u_3 , u_2 , u_1. The left most is the strongest bit-channel.

Kun-Cheng Hsu, Che-Wei Tsao, Yuan-Hao Chang, Tei-Wei Kuo, and Yu-Ming Huang, "Proactive Channel Adjustment to Improve Polar Code Capability for Flash Storage Devices," ACM/IEEE Design Automation Conference (DAC), San Francisco, California, USA, Jun 24-28, 2018.**(Top Conference)**

Layer-aware Strategy for 3D Charge-trap Flash

- A *layer-aware strategy* is proposed improve performance of 3D charge-trap flash by utilizing the speed difference among pages in a block. (IEEE TVLSI in 2018)
  - Classify data into different hotness levels, and gradually place data of different hotness to pages with suitable access speed.
  - The performance is improved by 33% and block erase count is reduced by 61%.



(a) 3D NAND vertical channel

(b) Top-down view of vertical channel

Block n is full. Waiting for garbage collection operation

VB 2n is Allocated to *Hot List* or *Icy-cold List*

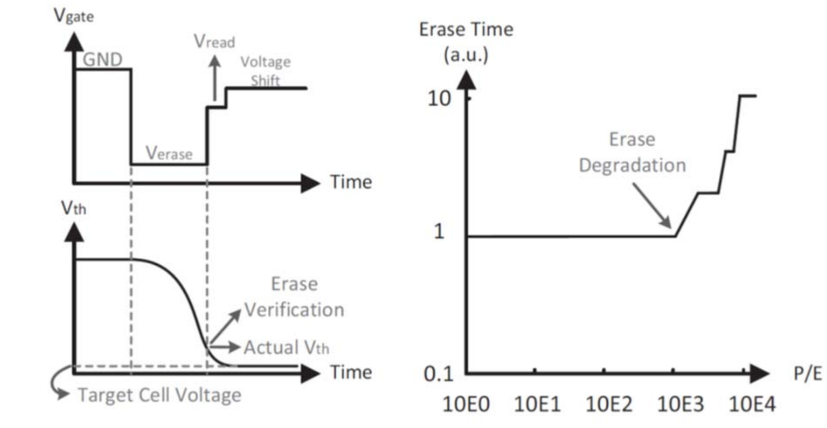
Filling up VB 2n

After VB 2n is full, VB 2n+1 can be allocated to *Iron-hot List* or *cold List*

Shuo-Han Chen, Yen-Ting Chen, Yuan-Hao Chang, Hsin-Wen Wei, and Wei-Kuan Shih, "A Progressive Performance Boosting Strategy for 3D Charge-trap NAND Flash," IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 26, no. 11, pp. 2322-2334, Nov. 2018.

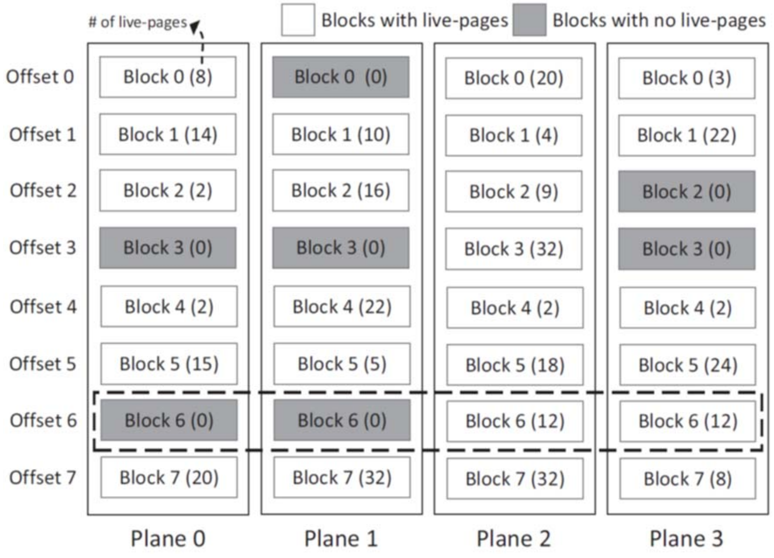
Multi-block Erase for 3D Charge-trap Flash

- Observation: Erase time in 3D charge-trap flash is increased when the number of P/E cycles is increased.
- We propose an *erase efficiency boosting strategy* to improve GC performance by utilizing the "multi-block erase" feature to reduce # of block erases. (IEEE TC in 2018)
  - We introduce the concept of "*erase efficiency*" on selecting a block or a block set for erases.
  - The erase latency is reduced by 75.76% on average.



(a) Transient Vth Shift phenomenon

(b) Erase degradation

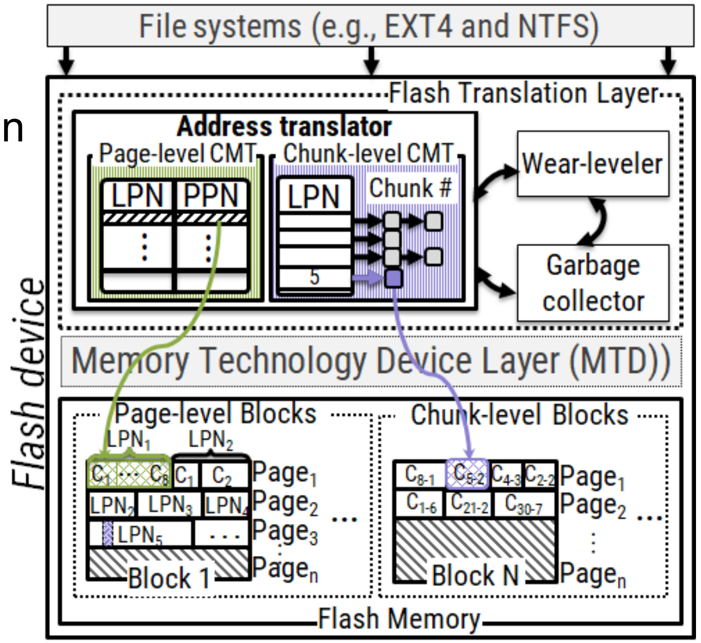


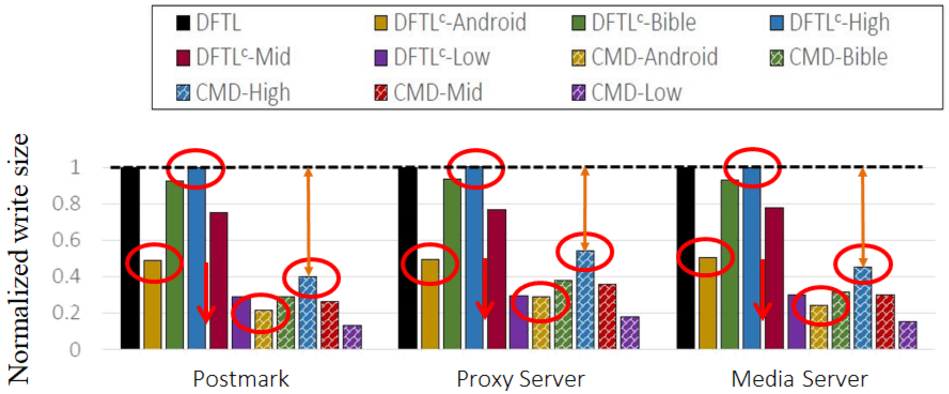
	Plane 0	Plane 1	Plane 2	Plane 3
Offset 0	Block 0 (8)	Block 0 (0)	Block 0 (20)	Block 0 (3)
Offset 1	Block 1 (14)	Block 1 (10)	Block 1 (4)	Block 1 (22)
Offset 2	Block 2 (2)	Block 2 (16)	Block 2 (9)	Block 2 (0)
Offset 3	Block 3 (0)	Block 3 (0)	Block 3 (32)	Block 3 (0)
Offset 4	Block 4 (2)	Block 4 (22)	Block 4 (2)	Block 4 (2)
Offset 5	Block 5 (15)	Block 5 (5)	Block 5 (18)	Block 5 (24)
Offset 6	Block 6 (0)	Block 6 (0)	Block 6 (12)	Block 6 (12)
Offset 7	Block 7 (20)	Block 7 (32)	Block 7 (32)	Block 7 (8)

Shuo-Han Chen, Yuan-Hao Chang, Yu-Pei Liang, Hsin-Wen Wei, and Wei-Kuan Shih, "An Erase Efficiency Boosting Strategy for 3D Charge Trap NAND Flash," IEEE Transactions on Computers (TC), vol. 67, no. 9, pp. 1246-1258, Sep. 2018.

Reducing Write Amplification for Large-Page Flash

- We exploit *data compression with a partial page update* to extend the flash lifetime by reducing write amplification (DAC 2018)
  - Observation:
    Flash page is getting larger to (1) worsen the write amplification and (2) increase read-merge-write overhead.
  - We propose a compression-based management design to pack data of multiple pages in the same flash page with minimal internal space fragmentation.
    - Data are partially updated and merged together out-of-place to reduce write amplification.
    - The proposed design can reduce write amplification for 50%-95%.

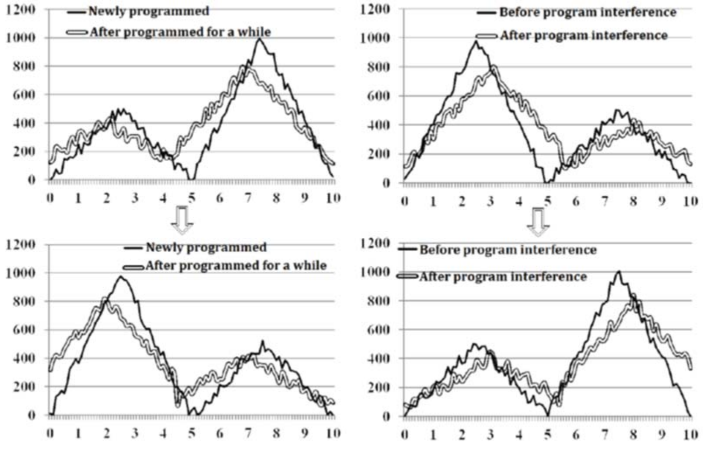




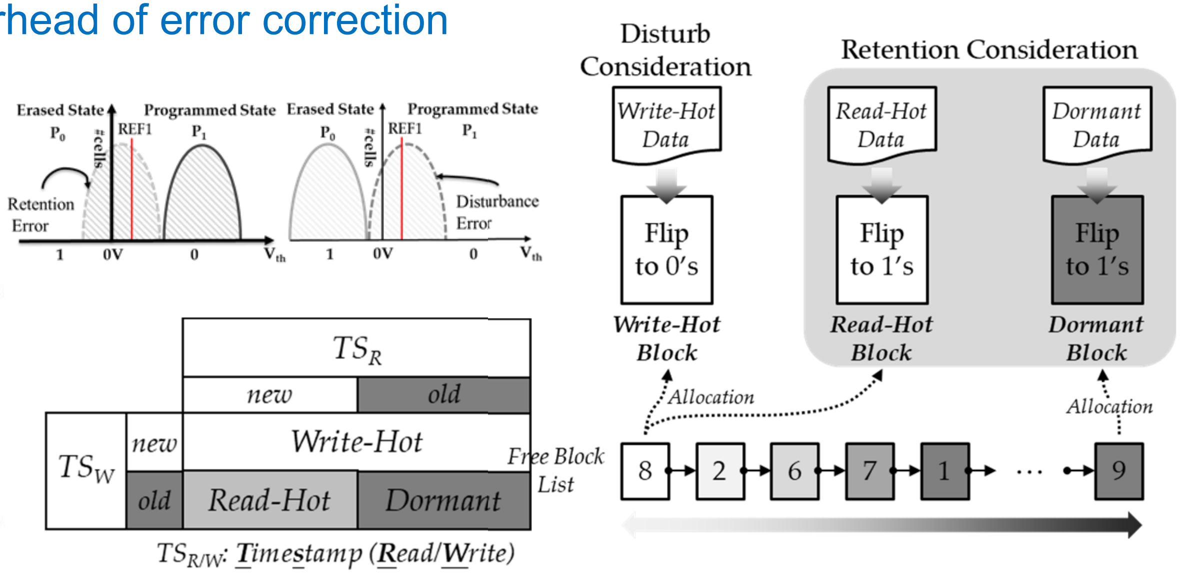
Wei-Lin Wang, Tseng-Yi Chen, Yuan-Hao Chang, Hsin-Wen Wei, and Wei-Kuan Shih, "Minimizing Write Amplification to Enhance Lifetime of Large-page Flash-Memory Storage Devices," ACM/IEEE Design Automation Conference (DAC), San Francisco, USA, Jun. 24-28, 2018. **(Top Conference)**

Pattern-aware Write Strategy for Flash Reliability

- Observation:
  The major bit errors of flash cells come from the *retention error* and the *disturbance error*
- We jointly consider written data pattern and flash block P/E cycle to propose a *pattern-aware write strategy* to enhance flash reliability and lifetime. (ACM TODAES in 2018)
  - Avoid retention error: Flip the bit pattern of dormant data to bit 1 and write the dormant data to vulnerable block
  - Avoid disturbance error: Flip the bit pattern of hot data to bit 0 and write the hot data to strong block
- Achievement: Minimizing the overhead of error correction



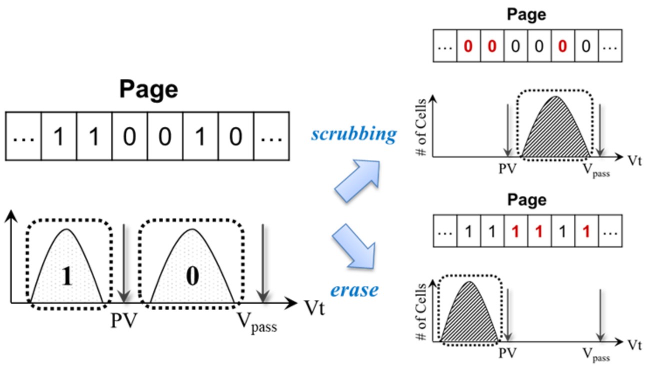
(a) Retention error. (b) Disturbance error.

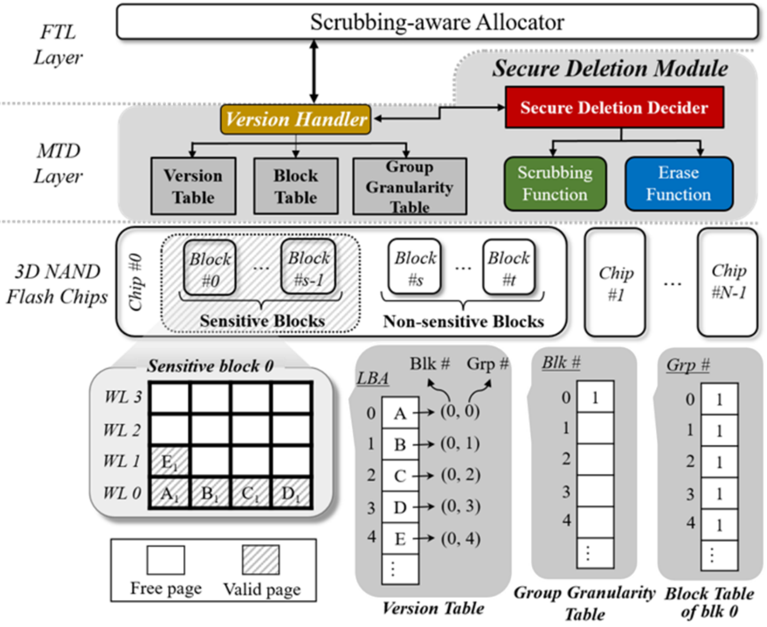


Tseng-Yi Chen, Yuan-Hao Chang, Yuan-Hung Kuan, Ming-Chang Yang, Yu-Ming Chang, and Pi-Cheng Hsiu, "Enhancing Flash Memory Reliability by Jointly Considering Write-back Pattern and Block Endurance," ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 23, no. 5, pp. 64:1-64:24, Aug. 2018.

Scrubbing-aware Secure Deletion for 3D NAND Flash

- We propose a *scrubbing-aware secure deletion module* to improve the secure deletion efficiency and guarantee to delete all versions of data (IEEE TCAD in 2018, CODES 2018)
- Grouping-based write strategy could minimize erase/scrubbing overheads by organizing sensitive data to create the scrubbing-friendly patterns
- A proper operation is chosen by the proposed evaluation equations for each secure deletion command
  - A Disturbance Infection Scrubbing Policy to distribute the scrubbing disturbance to every to-be-deleted pages equally for reducing the secure deletion latency. (Reduce up to 82% of secure deletion time)



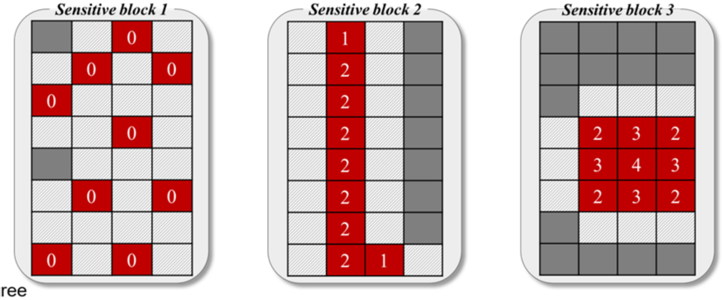


🟥 A to-be-deleted page (TBD page)

▨ A disturbed valid page

⬛ An undisturbed valid page

T_s: scrubbing time
T_e: erase time
T_{lpc}: live-page-copying time
f(): function of concentration degree



Degree of concentration:	$\frac{0*9}{2*9} = 0$	$\frac{1*2+2*7}{2*9} = 0.89$	$\frac{2*4+3*4+1*4}{2*9} = 1.33$
Evaluating scrubbing time:	$9t_s * f(0) + 21 * t_{lpc}$	$9t_s * f(0.89) + 16 * t_{lpc}$	$9t_s * f(1.33) + 9 * t_{lpc}$
Evaluating erase time:	$t_e + 23 * t_{lpc}$	$t_e + 23 * t_{lpc}$	$t_e + 23 * t_{lpc}$

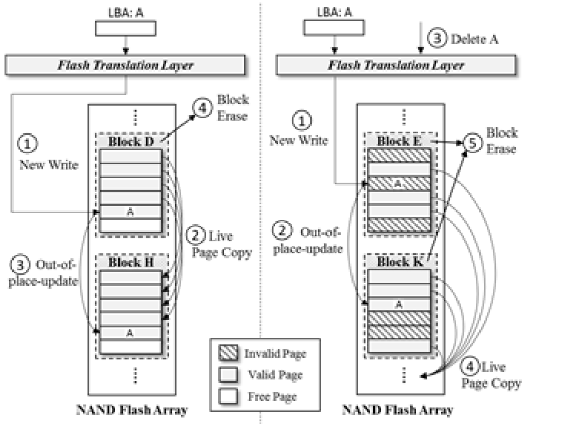
- Wei-Chen Wang, Chien-Chung Ho, Yuan-Hao Chang, Tei-Wei Kuo, and Ping-Hsien Lin, "Scrubbing-aware Secure Deletion for 3D NAND Flash," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 37, no. 11, pp. 2790-2801, Nov. 2018. (Integrated with ACM/IEEE CODES+ISSS'18)
- Wei-Chen Wang, Chien-Chung Ho, Yuan-Hao Chang, Tei-Wei Kuo, and Ping-Hsien Lin, "Scrubbing-aware Secure Deletion for 3D NAND Flash," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Torino, Italy, Sep. 30 - Oct. 5, 2018. (Journal Track, Integrated with IEEE TCAD) **(Top Conference)**

3. Flash Cell Programming Techniques (with Macronix)

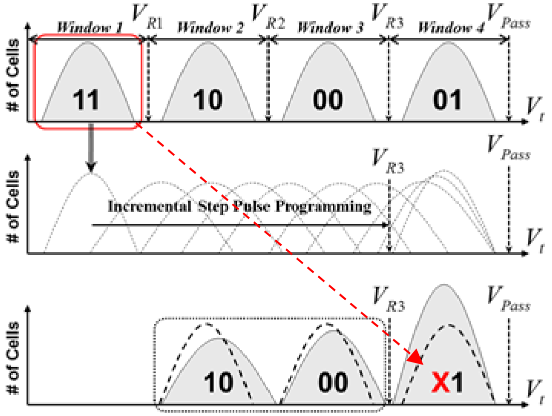
In-situ Sanitization
- Fast Sanitization with Zero Copy

with Macronix

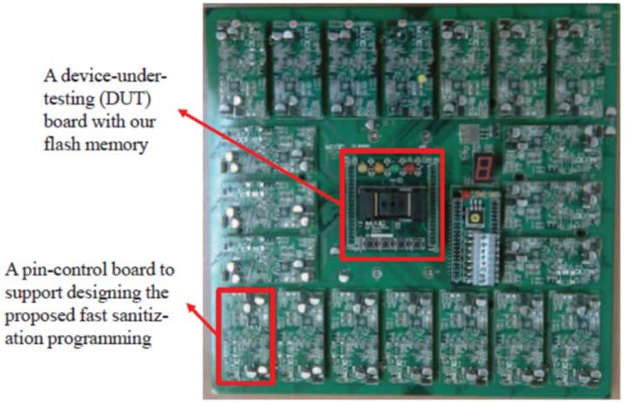
- **In-situ Sanitization** to support physically remove user data without any data moving for MLC flash memories (ICCAD 2018)
  - **Observation**: Typical design managements must actively move out valid data on the block that contains to-be-sanitized pages, which introduces a significant amount of extra writes and harms the performance and endurance.
  - **Programming-assisted Sanitization**:
    - This is the first invention that destroys the upper or lower pages through a programming style with only one voltage required. This totally removes the need of live-data-copy with 1% performance overhead.
    - This design is verified on a testing platform with real chips.



Sanitize data in a typical design management



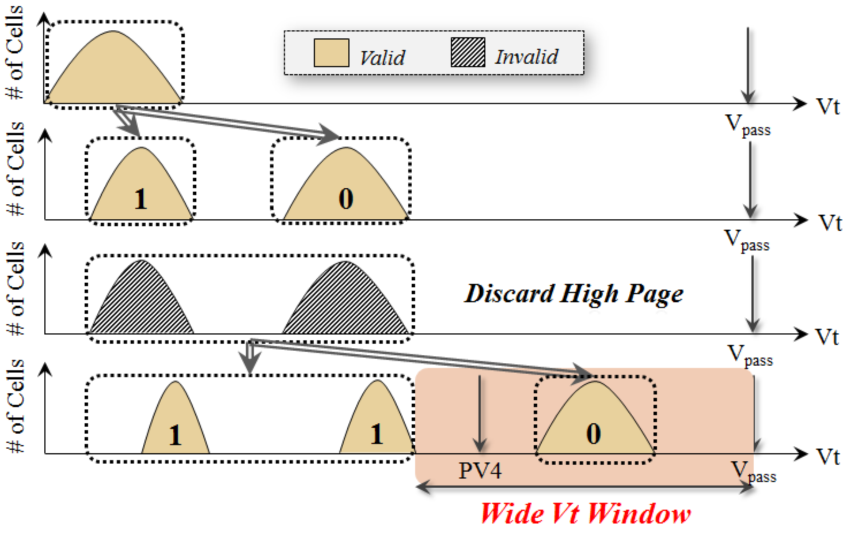
Programming-assisted sanitization

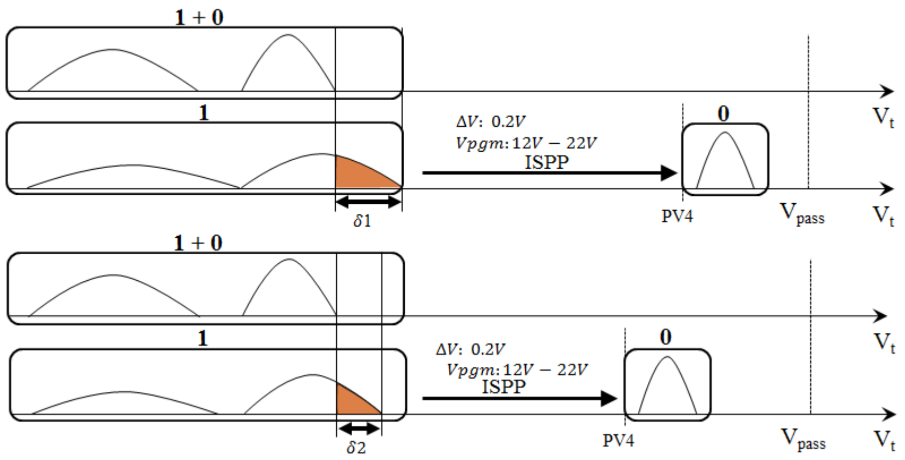

A device-under-testing (DUT) board with our flash memory
A pin-control board to support designing the proposed fast sanitization programming

Ping-Hsien Lin, Yu-Ming Chang, Yung-Chun Li, Wei-Chen Wang, Chien-Chung Ho, and Yuan-Hao Chang, "Achieving Fast Sanitization with Zero Live Data Copy for MLC Flash Memory," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), San Diego, California, USA, Nov. 5-8, 2018. **(Top Conference)**

Achieving SLC Perf. with MLC Flash **with Macronix**

- We propose a ***trim-like program*** to intelligently utilize the knowledge of the *data validity* so as to program low page with the speed of SLC flash. (ACM TOS in 2018, DAC 2015)
- Resolve the fundamental issue of ISPP in programming MLC flash.





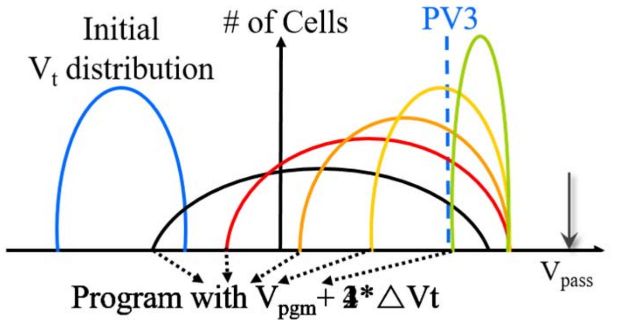
- Chien-Chung Ho, Yu-Ming Chang, Yuan-Hao Chang, and Tei-Wei Kuo, "SLC-Like Programming Scheme for MLC Flash Memory," ACM Transactions on Storage (TOS), vol. 14, no. 1, pp. 11:1-11:26, Mar. 2018.
- Yu-Ming Chang, Yuan-Hao Chang, Tei-Wei Kuo, Yung-Chun Li, and Hsiang-Pang Li, "Achieving SLC Performance with MLC Flash Memory," ACM/IEEE Design Automation Conference (DAC), San Francisco, California, USA, Jun. 7-11, 2015. **(Top Conference)**

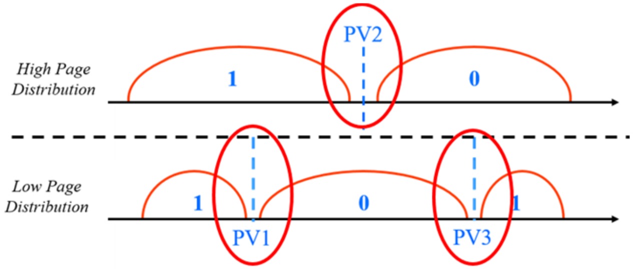
One-Shot Program Design with Macronix

- We propose an ***one-shot programming design*** to achieve the defect-free multilevel 3D flash memories (DAC 2018)
  - Observation:

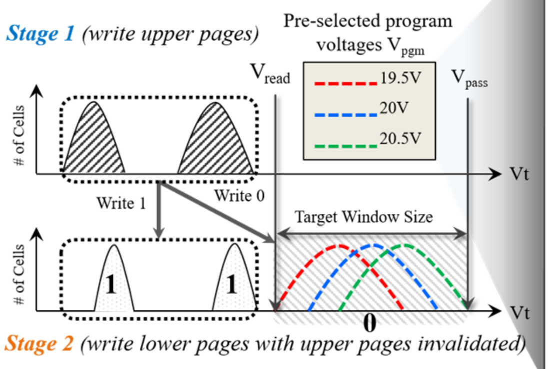
    Conventional programming designs result in the flash defects on low pages

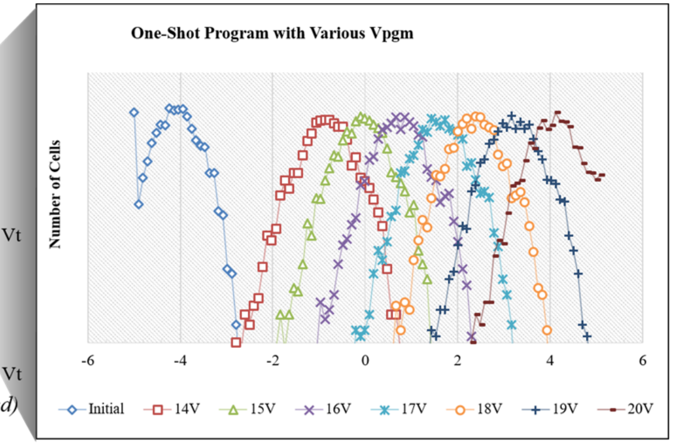
    1. **Time-consuming delay** on programming low page
    2. **More sensing voltages** are required to read a low page data
    3. Narrow V_t window results in the **higher BER** in low pages
  - We propose to only use **one of the pre-selected program voltage** to program a low page
    1. Use one program shot to push cells' V_t distribution to the desire state
    2. Relax the needs of ISPP on programming a low page

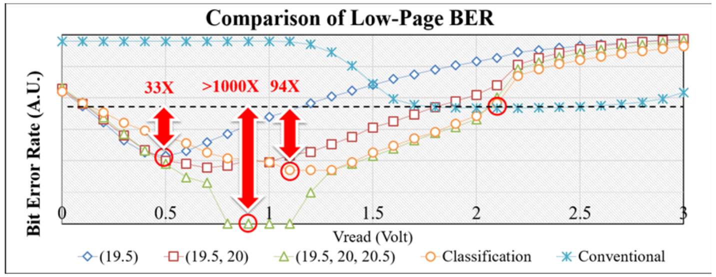




2 Sensing voltage shots are rquired


Stage 1 (write upper pages)
Pre-selected program voltages V_{pgm}
V_{read}
V_{pass}
19.5V
20V
20.5V
# of Cells
Vt
Write 1
Write 0
Target Window Size
1
1
0
Vt
Stage 2 (write lower pages with upper pages invalidated)





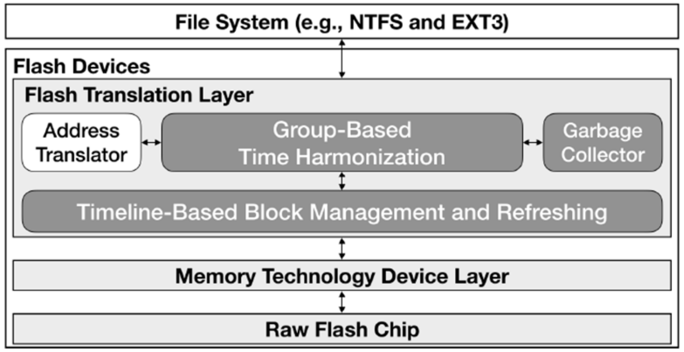
Chien-Chung Ho, Yung-Chun Li, Yuan-Hao Chang, and Yu-Ming Chang, "Achieving Defect-Free Multilevel 3D Flash Memory with One-Shot Program Design," ACM/IEEE Design Automation Conference (DAC), San Francisco, USA, Jun. 24-28, 2018. **(Top Conference)**

Others

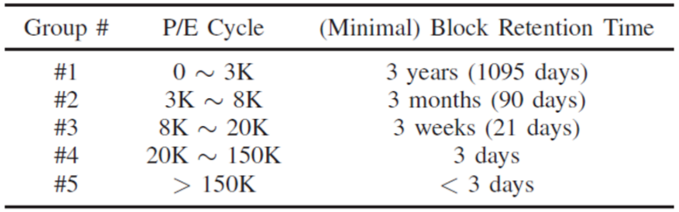
Harmonization for Data Lifetime and Block Retention Time

(NVMSA 2018)

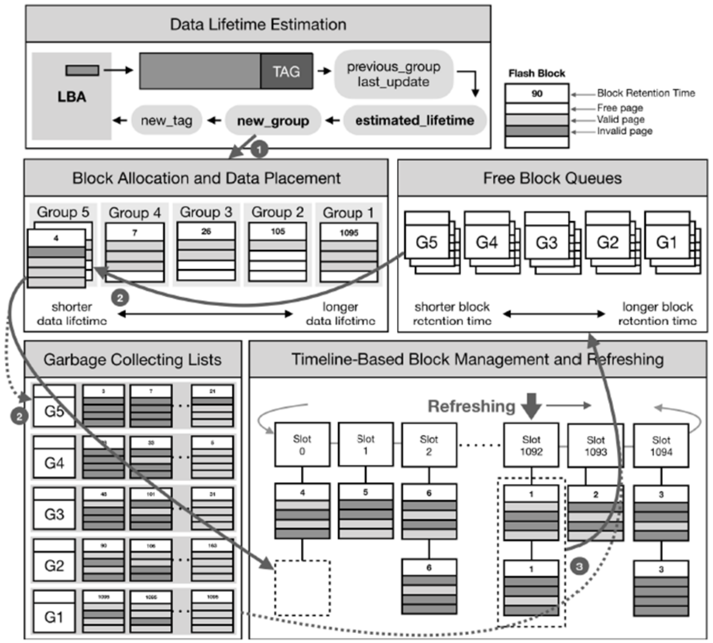
- Observation
  - The access performance and endurance of flash devices are worsened by the mismatch between data lifetime requirement and flash block retention capability.
- Goal
  - We propose a “time harmonization strategy”, which coordinates the flash block retention capability with the data lifetime requirement.
- Main Idea
  - Estimate the data lifetime requirement based on the periods of data update
  - Arrange data and flash blocks into groups in accordance with the estimated data lifetime requirement

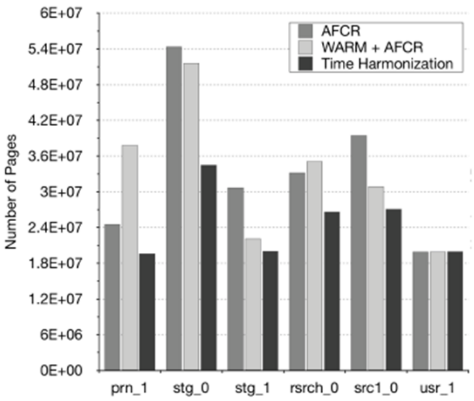

File System (e.g., NTFS and EXT3)
Flash Devices
Flash Translation Layer
Address Translator
Group-Based Time Harmonization
Garbage Collector
Timeline-Based Block Management and Refreshing
Memory Technology Device Layer
Raw Flash Chip

RELATIONSHIP BETWEEN P/E CYCLE AND BLOCK RETENTION CAPABILITY.



Group #	P/E Cycle	(Minimal) Block Retention Time
#1	0 ∼ 3K	3 years (1095 days)
#2	3K ∼ 8K	3 months (90 days)
#3	8K ∼ 20K	3 weeks (21 days)
#4	20K ∼ 150K	3 days
#5	> 150K	< 3 days





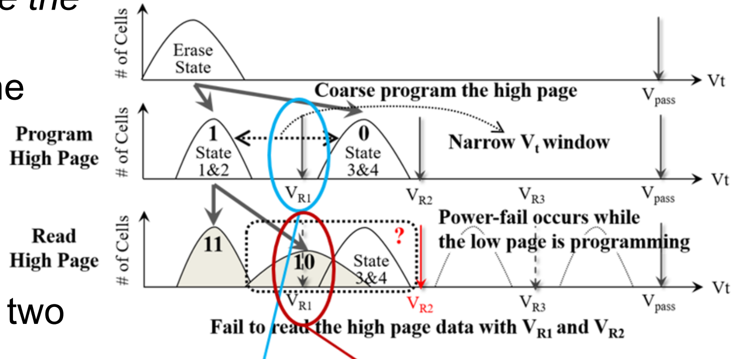
Data Migration Overhead Reduction: 33%

Yi-Ling Lin, Ming-Chang Yang, Yuan-Hao Chang, Che-Wei Chang, and Shuo-Han Chen, "On Harmonizing Data Lifetime and Block Retention Time for Flash Devices," IEEE Nonvolatile Memory Systems and Applications Symposium (NVMSA), Hakodate, Japan, Aug. 28-31, 2018.

Stride-away Programming for Crash Recovery

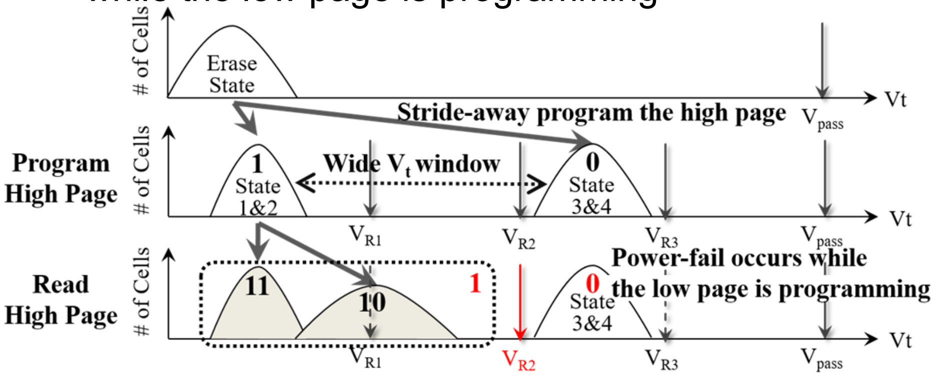
We propose an ***stride-away programming design*** *to optimize the* programming speed and flash reliability by relaxing the programming constraints and eliminating defeats caused by the conventional ISPP strategy. (NVMSA 2018)

- Observation:
  1. Conventional program scheme encounters an embarrassment of data unreadability issue
     - It happens when any power failure occurs before the two pages of the same word line are programmed
     - VR1 can not be used to identify the high page data
  2. The **narrow V_t window** disables the readability of the high page and causes the needs of backup operations as well
- Our proposed stride-away programing proposes **to create the wide fault-tolerable V_t window range for the high page data in the coarse programming stage**
  - Enable the readability while power-fail occurs while the low page is programming

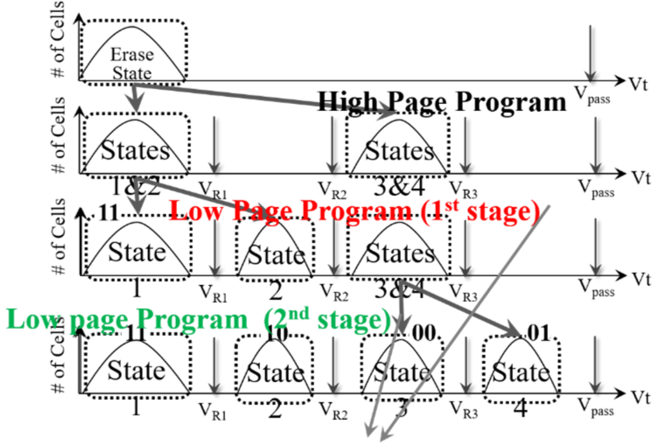


It can be used to identify high page data

It fails!



Success to read the high page data with V_{R2},



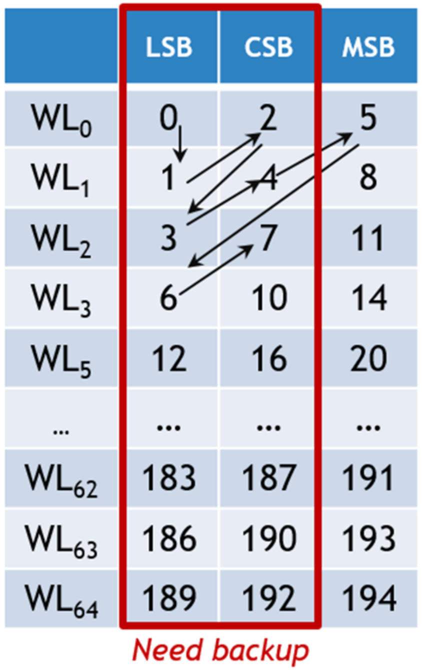
Only 1P1V is needed!

Programming latency is improved

Chien-Chung Ho, Yung-Chun Li, Ping-Hsien Lin, Wei-Chen Wang, and Yuan-Hao Chang, "A Stride-away Programming Scheme to Resolve Crash Recoverability and Data Readability Issues of Multi-level-cell Flash Memory," IEEE Nonvolatile Memory Systems and Applications Symposium (NVMSA), Hakodate, Japan, Aug. 28-31, 2018.

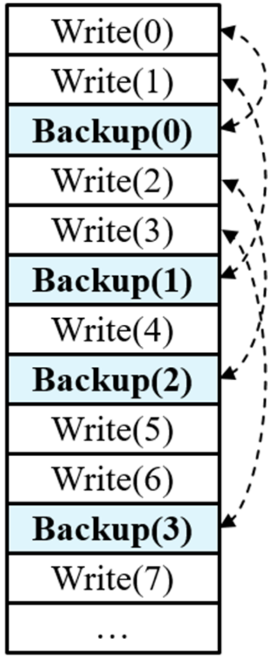
Data-backup-free Programming for TLC-based SSD (SAC 2018)

- Observation: Conventional ISPP strategy fails on while the sudden-power-loss happen when programming
  - Naïve or intuitive solution
    - Backup the data before the disturb occurs
      - Decreasing space utilization
      - Increasing time latency
    - Host-level RAID solution
      - Doesn't work
      - The flash device treats data and parity as just data and there is more data to be backup
- Our proposed data-backup-free programming
  - Chip-level RAID with skipping striping in flash device to avoid any 2 or more parts of the same stripe stored on the same wordline
  - Benefits
    - Make sure the correctness of the programmed data even the sudden-power-loss happens
    - To eliminate the need of data backup

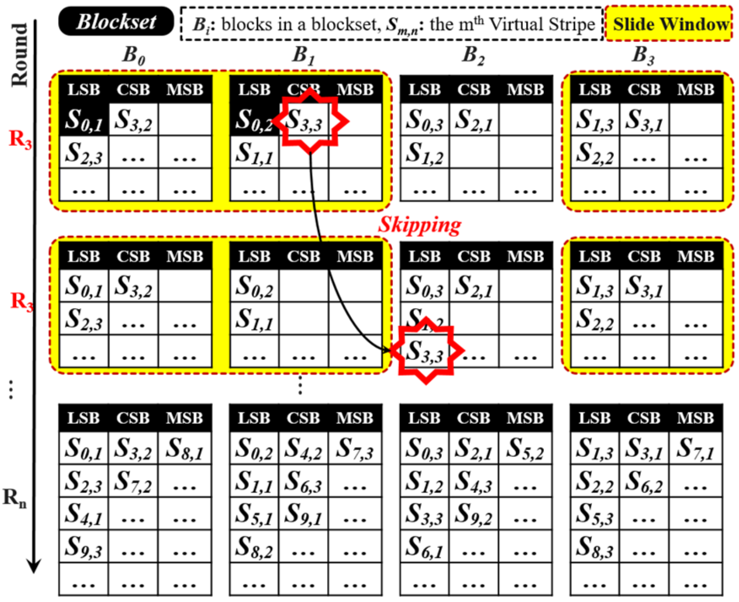


	LSB	CSB	MSB
WL_0	0	2	5
WL_1	1	4	8
WL_2	3	7	11
WL_3	6	10	14
WL_5	12	16	20
…	…	…	…
WL_{62}	183	187	191
WL_{63}	186	190	193
WL_{64}	189	192	194

Need backup



Write(0)
Write(1)
Backup(0)
Write(2)
Write(3)
Backup(1)
Write(4)
Backup(2)
Write(5)
Write(6)
Backup(3)
Write(7)
…



Blockset
B_i: blocks in a blockset, $S_{m,n}$: the mth Virtual Stripe
Slide Window
Round
Skipping

	B_0			B_1			B_2			B_3		
	LSB	CSB	MSB	LSB	CSB	MSB	LSB	CSB	MSB	LSB	CSB	MSB
R_3	$S_{0,1}$	$S_{3,2}$		$S_{0,2}$	$S_{3,3}$		$S_{0,3}$	$S_{2,1}$		$S_{1,3}$	$S_{3,1}$	
	$S_{2,3}$	…	…	$S_{1,1}$			$S_{1,2}$			$S_{2,2}$	…	…
	…	…	…	…	…	…	…	…	…	…	…	…

	B_0			B_1			B_2			B_3		
	LSB	CSB	MSB	LSB	CSB	MSB	LSB	CSB	MSB	LSB	CSB	MSB
R_3	$S_{0,1}$	$S_{3,2}$		$S_{0,2}$			$S_{0,3}$	$S_{2,1}$		$S_{1,3}$	$S_{3,1}$	
	$S_{2,3}$	…	…	$S_{1,1}$			$S_{1,2}$			$S_{2,2}$	…	…
	…	…	…	…	…	…	$S_{3,3}$	…	…	…	…	…

	B_0			B_1			B_2			B_3		
	LSB	CSB	MSB	LSB	CSB	MSB	LSB	CSB	MSB	LSB	CSB	MSB
R_n	$S_{0,1}$	$S_{3,2}$	$S_{8,1}$	$S_{0,2}$	$S_{4,2}$	$S_{7,3}$	$S_{0,3}$	$S_{2,1}$	$S_{5,2}$	$S_{1,3}$	$S_{3,1}$	$S_{7,1}$
	$S_{2,3}$	$S_{7,2}$	…	$S_{1,1}$	$S_{6,3}$	…	$S_{1,2}$	$S_{4,3}$	…	$S_{2,2}$	$S_{6,2}$	…
	$S_{4,1}$	…	…	$S_{5,1}$	$S_{9,1}$	…	$S_{3,3}$	$S_{9,2}$	…	$S_{5,3}$	…	…
	$S_{9,3}$	…	…	$S_{8,2}$	…	…	$S_{6,1}$	…	…	$S_{8,3}$	…	…
	…	…	…	…	…	…	…	…	…	…	…	…

Chin-Chiang Pan, Chien-Chung Ho, Yuan-Hao Chang, Tei-Wei Kuo, Yu-Ming Chang, and Ming-Chang Yang, "Boosting the Performance with a Data-backup-free Programming Scheme for TLC-based SSDs," ACM Symposium on Applied Computing (SAC), Pau, France, Apr. 9-13, 2018.

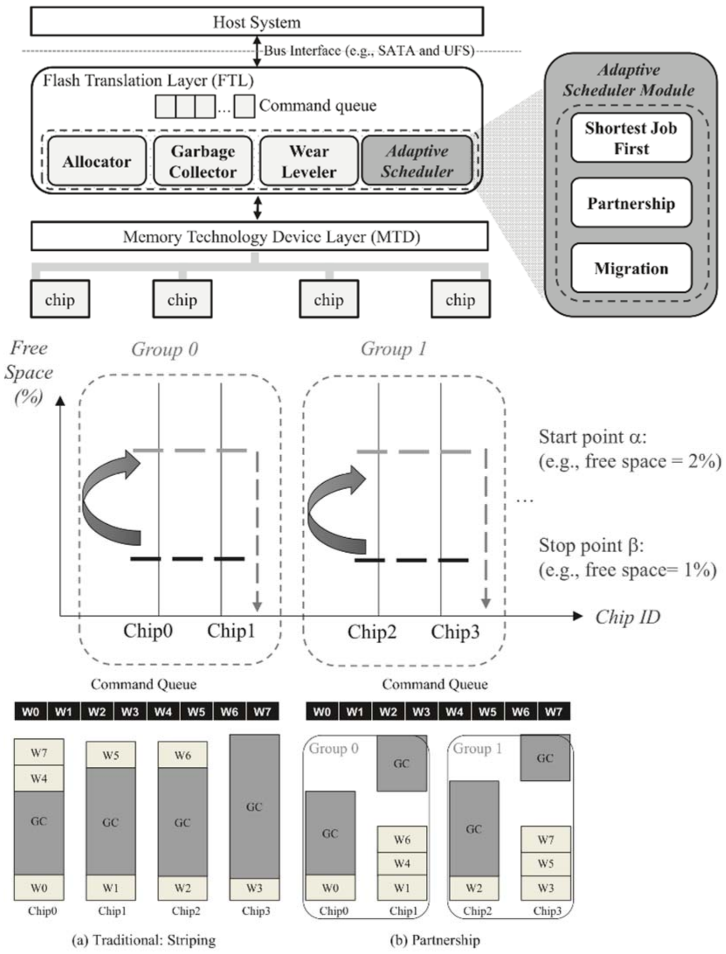
Partnership-based Approach to Minimize Maximal Response Time of Flash Storage

Observation:
Flash memory storage systems have unpredictable maximal response time when all the flash chips are under garbage collection concurrently.

We proposed a partnership-based management design

- A shortest job first strategy
  - To reduce the average waiting time of read / write request in the internal command queue.
- A partnership strategy
  - To partition flash chips into a fixed number of partner groups
  - To guarantee the minimum number of chips that can still serve read / write request
- A migration strategy
  - To adaptively balance the space utilization by redistributing cold data among flash chips

Compared to existing approaches, the proposed approach could greatly reduce the maximal response time for 66.7% and 77/5% under the project server and source server traces respectively.



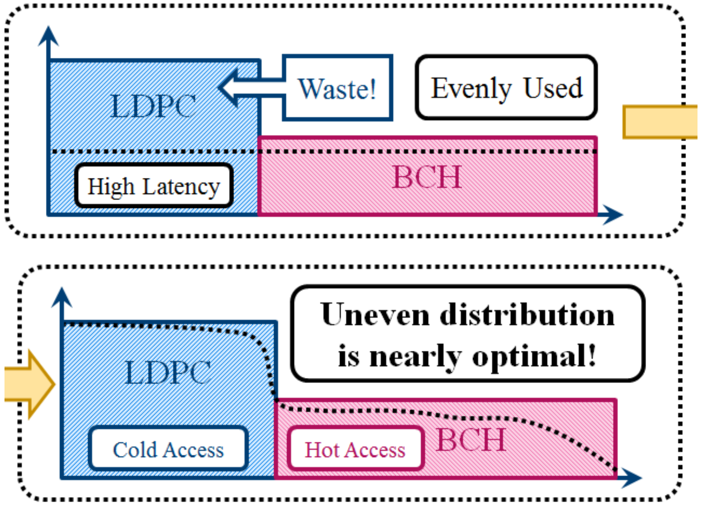
(a) Traditional: Striping
(b) Partnership

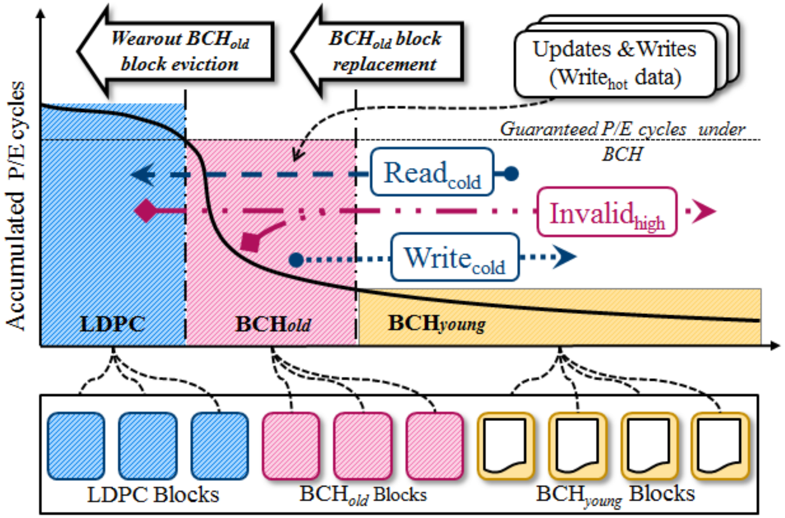
Tse-Yuan Wang, Che-Wei Tsao, Yuan-Hao Chang, Tei-Wei Kuo, and Hsiang-Pang Li, "A Partnership-Based Approach to Minimize the Maximal Response Time of Flash-Memory Storage Systems," ACM Symposium on Applied Computing (SAC), Pau, France, Apr. 9-13, 2018.

Research Summary 2017

Anti-wear Leveling

- **Anti-wear Leveling** for SSDs with Hybrid ECC (IEEE TVLSI in 2017)
  - **Observation**:
    BCH is fast but provides weak correction strength
    LDPC is iterative-pass and provide strong recovering ability.
  - We are the first team to propose the concept of anti-wear leveling to get rid of wear leveling overhead.
  - Anti-wear leveling is to improve performance by avoiding wear leveling but deliberately generating uneven wear leveling distribution over blocks by moving data of different access patterns to different types of blocks protected by either BCH or LDPC.

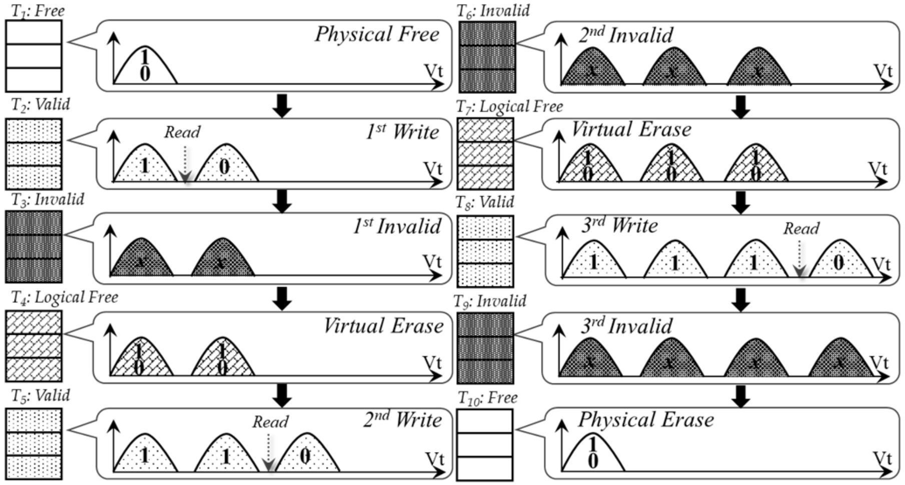


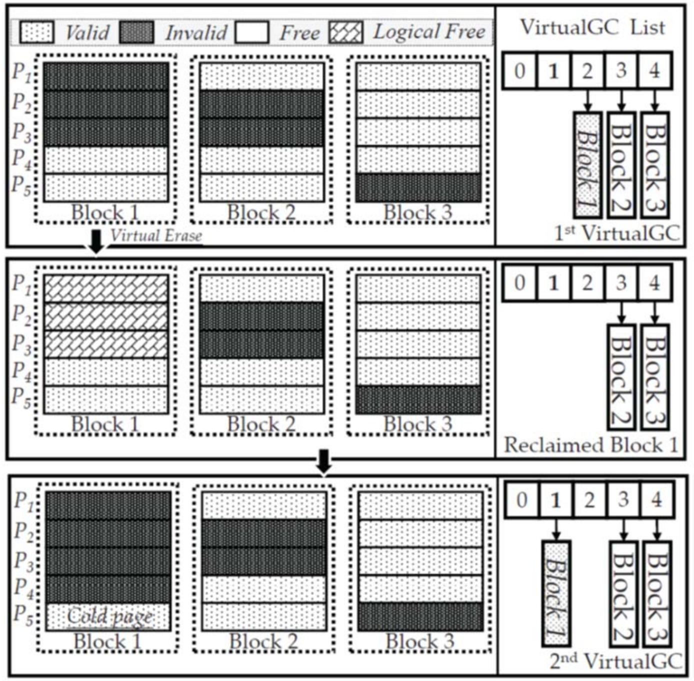

Wearout BCH_old block eviction
BCH_old block replacement
Updates & Writes (Write_hot data)
Guaranteed P/E cycles under BCH
Read_cold
Invalid_high
Write_cold
Accumulated P/E cycles
LDPC
BCHold
BCHyoung
LDPC Blocks
BCH_old Blocks
BCH_young Blocks

Chien-Chung Ho, Yu-Ping Liu, Yuan-Hao Chang, Tei-Wei Kuo, "Anti-wear Leveling Design for SSDs with Hybrid ECC Capability," IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 25, no. 2, pp. 488-501, Feb. 2017.

VirtualGC

- **Enabling erase-free garbage collection** of 3D flash memory (DAC 2017)
  - This is the first work to discuss how to design a performance-efficient flash management design for rewritable 3D flash drives.
  - VirtualGC keeps live pages in the virtually erased blocks for postponing copying live pages as much as possible.

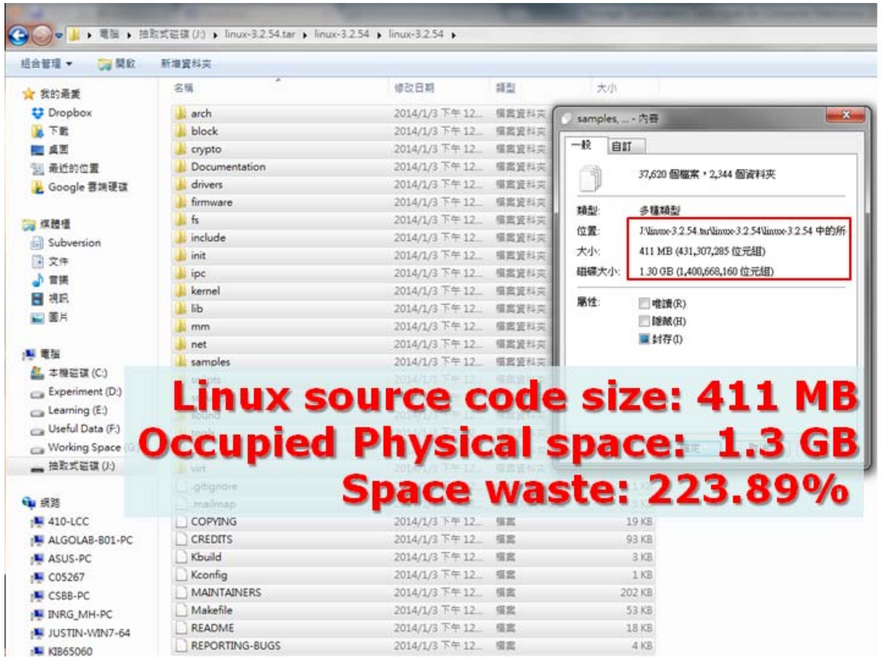




Tseng-Yi Chen, Yuan-Hao Chang, Yuan-Hung Kuan, and Yu-Ming Chang, "VirtualGC: Enabling Erase-free Garbage Collection to Upgrade the Performance of Rewritable SLC NAND Flash Memory," ACM/IEEE Design Automation Conference (DAC), Austin, Texas, USA, Jun. 18-22, 2017. **(Top Conference)**

Dynamic Tail Packing

- Space Utilization Optimization for File Systems (IEEE TECS in 2017)
  - We are the first team to propose the concept of "dynamic tail packing" to optimize the space utilization of embedded file systems .
  - 64% space saving is achieved on storing the source codes of Linux kernel 3.8 over FAT32 (with an implementation on Linux).



samples, ... - 內容

一般 | 自訂

37,620 個檔案，2,344 個資料夾

類型: 多種類型

位置: J:\linux-3.2.54.tar\linux-3.2.54\linux-3.2.54 中的所

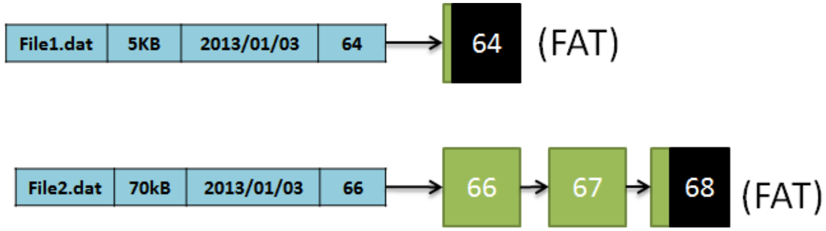
大小: 411 MB (431,307,285 位元組)

磁碟大小: 1.30 GB (1,400,668,160 位元組)

屬性: □ 唯讀(R) □ 隱藏(H) ■ 封存(I)

名稱	修改日期	類型	大小
COPYING	2014/1/3 下午 12...	檔案	19 KB
CREDITS	2014/1/3 下午 12...	檔案	93 KB
Kbuild	2014/1/3 下午 12...	檔案	3 KB
Kconfig	2014/1/3 下午 12...	檔案	1 KB
MAINTAINERS	2014/1/3 下午 12...	檔案	202 KB
Makefile	2014/1/3 下午 12...	檔案	53 KB
README	2014/1/3 下午 12...	檔案	18 KB
REPORTING-BUGS	2014/1/3 下午 12...	檔案	4 KB

Linux source code size: 411 MB
Occupied Physical space: 1.3 GB
Space waste: 223.89%



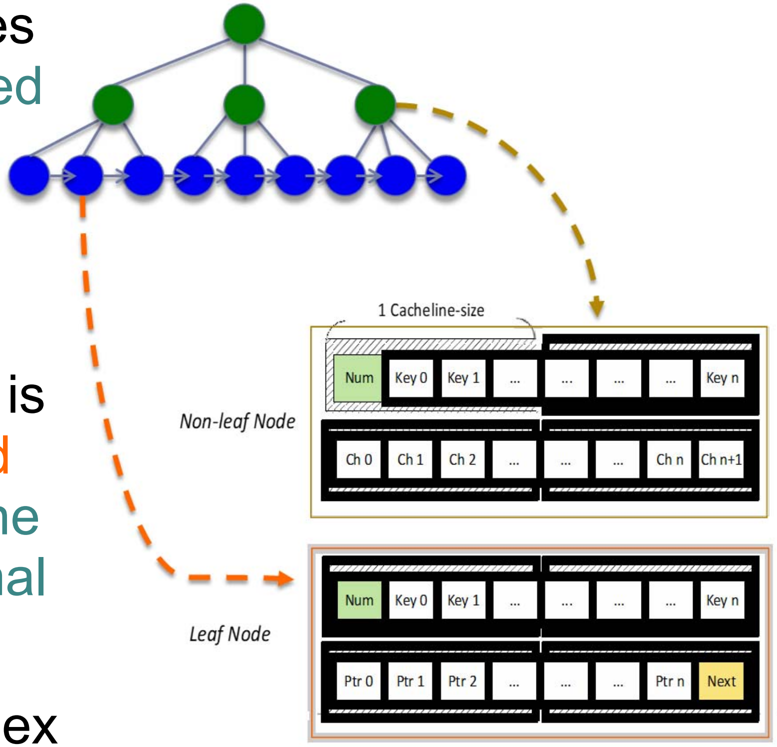
File1.dat	5KB	2013/01/03	64

File2.dat	70kB	2013/01/03	66

Tseng-Yi Chen, Yuan-Hao Chang, Shuo-Han Chen, Nien-I Hsu, Hsin-Wen Wei, and Wei-Kuan Shih, "On Space Utilization Enhancement of File Systems for Embedded Storage Systems," ACM Transactions on Embedded Computing Systems (TECS), vol. 16, no. 3, pp. 83:1-83:28, Apr. 2017.

xB+-Tree: Cache-line-Based Tree

- Observation
  - The existing node structure designs are helpful in reducing the number of writes to NVM but ignore the cache-line-based access behavior in the memory hierarchy!
- Our method
  - We propose a xB+-tree design, which is a simple but efficient cache-line-based tree that tends to keep the keys and the corresponding data items in the minimal number of cache lines.
  - This is the first attempt to consider index design with considering the cache line issue with NVM as main memory



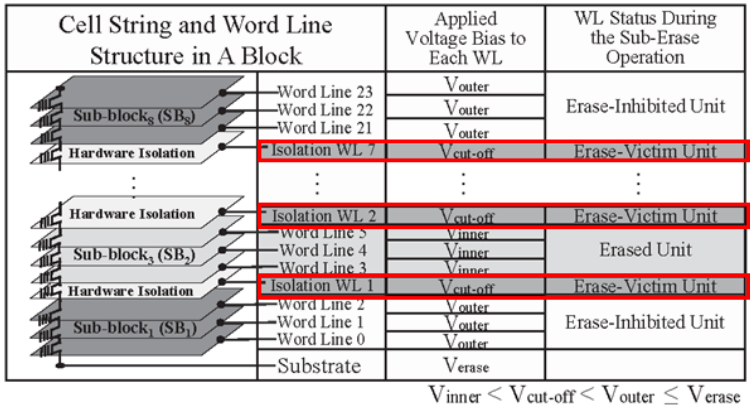
Li-Zheng Liang, Ming-Chang Yang, Yuan-Hao Chang, Tseng-Yi Chen, Shuo-Han Chen, Hsin-Wen Wei, and Wei-Kuan Shih, "xB+-Tree: Access-Pattern-Aware Cache-Line-Based Tree for Non-Volatile Main Memory Architecture," IEEE Computer Software and Applications Conference (COMPSAC), Torino, Italy, Jul. 4-8, 2017.

Research Summary 2016

Enabling Sub-Block Erase for 3D Flash

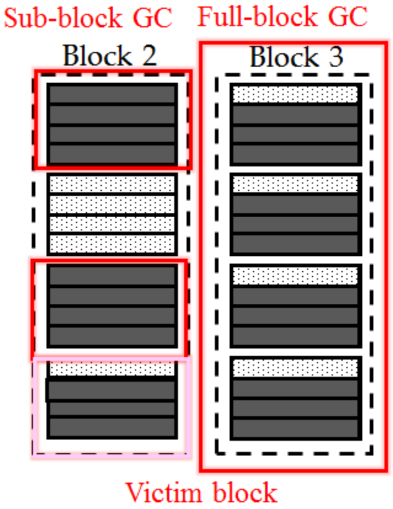
(DAC 2016 – Best Paper Nomination)

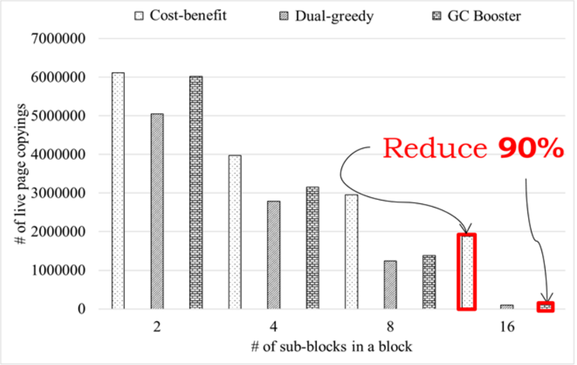
- We are the first team that *utilizes **isolation hardware** to enable sub-block erase* for optimizing the GC performance in large-block 3D flash memory.
- Existing work only considers the live-page copy overhead, but with sub-block erase, both block erase and live-page copy should be considered.
- We propose a new concept of "optimizing the reclaimed space per time unit" to take both *block erase* and *live-page copy* into consideration by concurrently erasing multiple sub-blocks having small enough number of live pages.
- Live-page copy overhead: 90% reduced.
- GC overhead: 48% reduced.

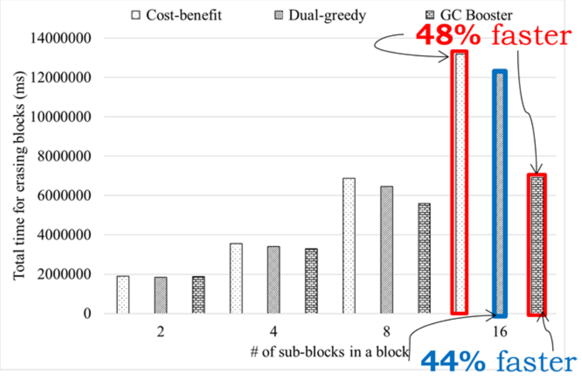


Cell String and Word Line Structure in A Block	Applied Voltage Bias to Each WL	WL Status During the Sub-Erase Operation
Word Line 23	V_{outer}	Erase-Inhibited Unit
Word Line 22	V_{outer}	
Word Line 21	V_{outer}	
Isolation WL 7	$V_{cut\text{-}off}$	Erase-Victim Unit
⋮	⋮	⋮
Isolation WL 2	$V_{cut\text{-}off}$	Erase-Victim Unit
Word Line 5	V_{inner}	Erased Unit
Word Line 4	V_{inner}	
Word Line 3	V_{inner}	
Isolation WL 1	$V_{cut\text{-}off}$	Erase-Victim Unit
Word Line 2	V_{outer}	Erase-Inhibited Unit
Word Line 1	V_{outer}	
Word Line 0	V_{outer}	
Substrate	V_{erase}	

$V_{inner} < V_{cut\text{-}off} < V_{outer} \leq V_{erase}$



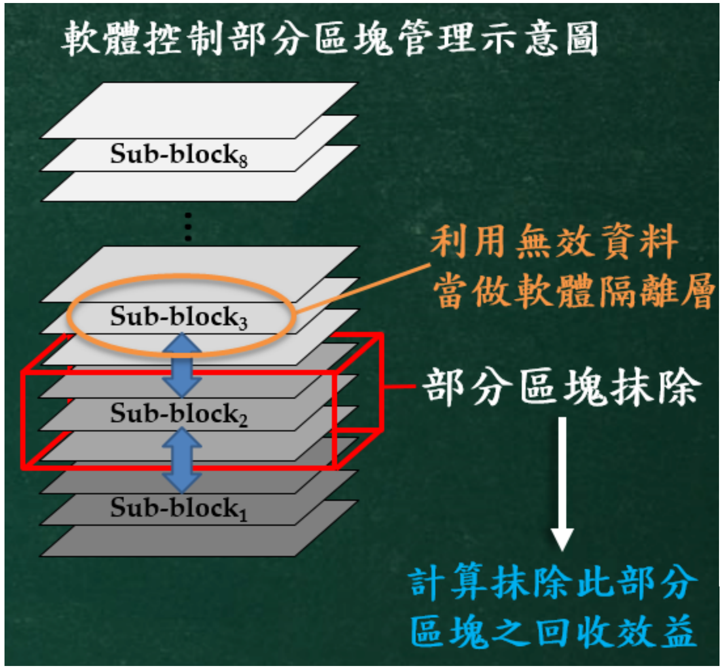


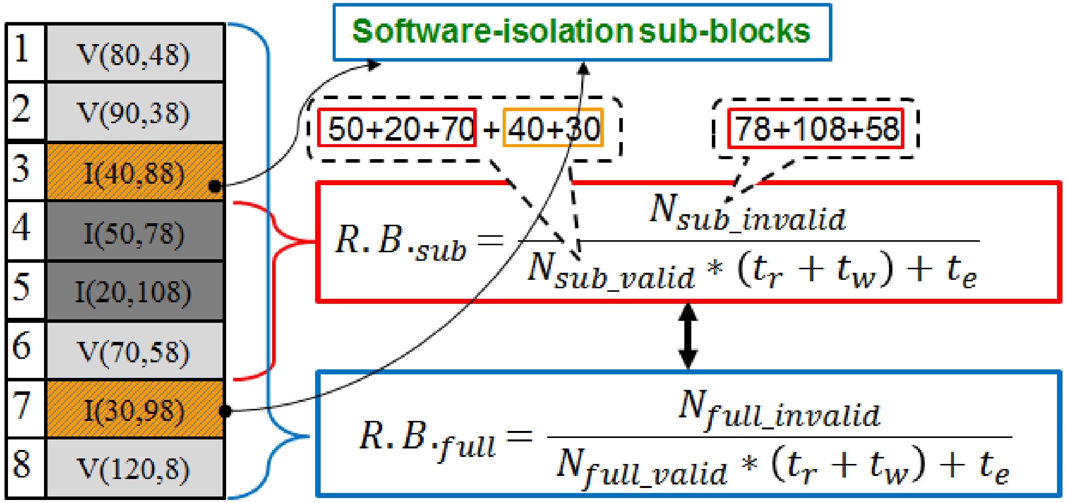


Tseng-Yi Chen, Yuan-Hao Chang, Chien-Chung Ho, and Shuo-Han Chen, "Enabling Sub-blocks Erase Management to Boost the Performance of 3D NAND Flash Memory," ACM/IEEE Design Automation Conference (DAC), Austin, Texas, USA, Jun. 5-9, 2016. **(Best Paper Nomination (16/674) - Top Conference)**

Research Results 2016 (2)

- Enabling **Sub-Block Erase** with Software Isolation for 3D Flash (CODES in 2016)
  - This is the first work that *enables sub-block erase with **software isolation** and without hardware cost* to reduce GC overhead of large-block 3D flash.
  - We propose a new evaluate metric called recycle benefit to evaluate whether the area isolated by the ***software-isolation sub-block*** can be erased.
  - This design reduces at least 20% GC overhead without extra hardware cost.





1	V(80,48)
2	V(90,38)
3	I(40,88)
4	I(50,78)
5	I(20,108)
6	V(70,58)
7	I(30,98)
8	V(120,8)

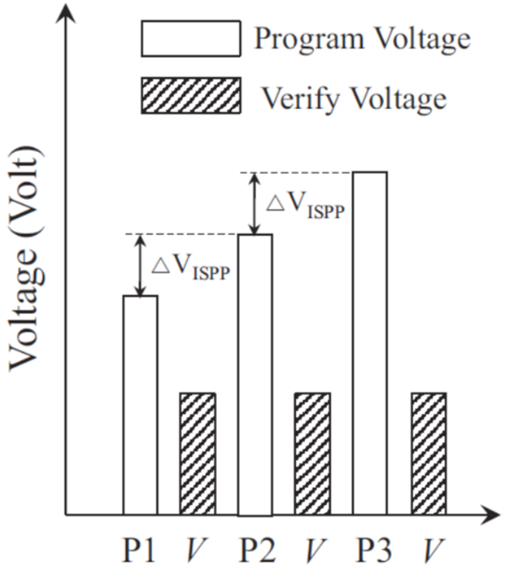
$$R.B._{sub} = \frac{N_{sub\_invalid}}{N_{sub\_valid} * (t_r + t_w) + t_e}$$

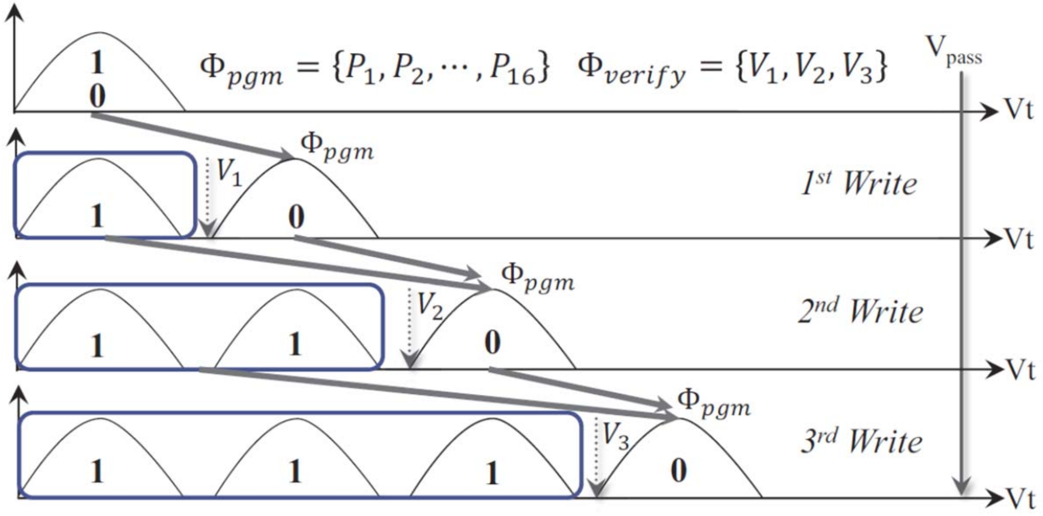
$$R.B._{full} = \frac{N_{full\_invalid}}{N_{full\_valid} * (t_r + t_w) + t_e}$$

Hsin-Yu Chang, Chien-Chung Ho, Yuan-Hao Chang, Yu-Ming Chang, and Tei-Wei Kuo, "How to Enable Software Isolation and Boost System Performance with Sub-block Erase over 3D Flash Memory," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Pittsburgh, Pennsylvania, USA, Oct. 2-7, 2016. **(Top Conference)**

Research Results 2016 (3)

- **Realizing Erase-free SLC Flash Memory** with Rewritable Programming Design (CODES in 2016)
  - The first work proposes a rewriteable usage model to realize the *erase-free concept* through the proposed rewritable programming design. (with Macronix).
  - Write performance is improved by 3.27 times.
  - Based on this rewriteable model, a series of revolutionary research is currently under investigation.

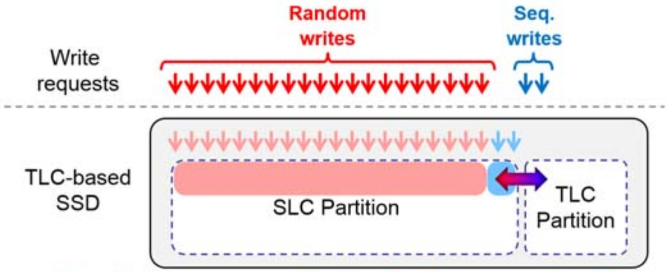




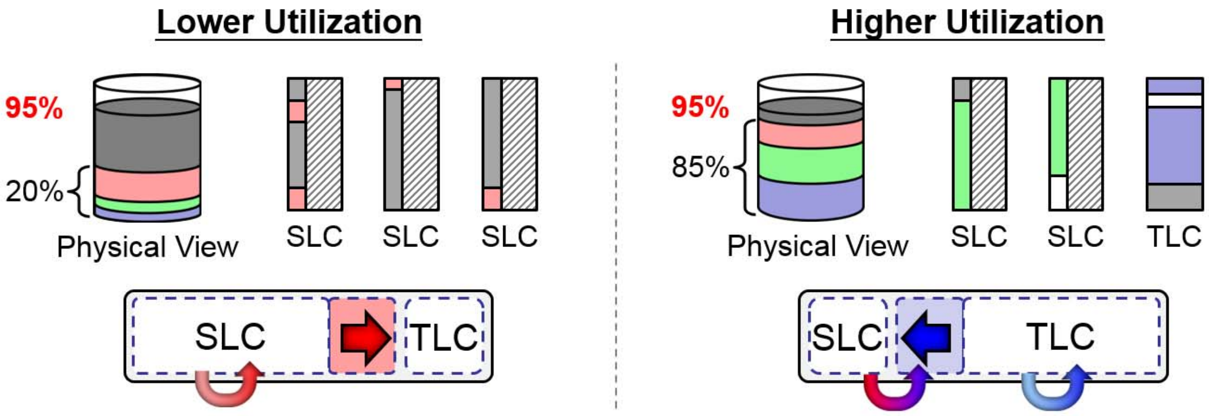
Yu-Ming Chang, Yung-Chun Li, Ping-Hsien Lin, Hsiang-Pang Li, and Yuan-Hao Chang, "Realizing Erase-free SLC Flash Memory with Rewritable Programming Design," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Pittsburgh, Pennsylvania, USA, Oct. 2-7, 2016. **(Top Conference)**

Research Results 2016 (4)

- **Utilization-aware Self-tuning** for TLC Flash (IEEE TVLSI in 2016)
  - We exploit *the mode-switching capability of **TLC** flash block* to dynamically (1) change the partition sizes and (2) decide wh SLC/MLC partitions.
  - A self-tuning design is proposed to improve TLC flash performance by catching working set as much as possible under different space utilizations.



Trading non-used physical space for the performance improvement



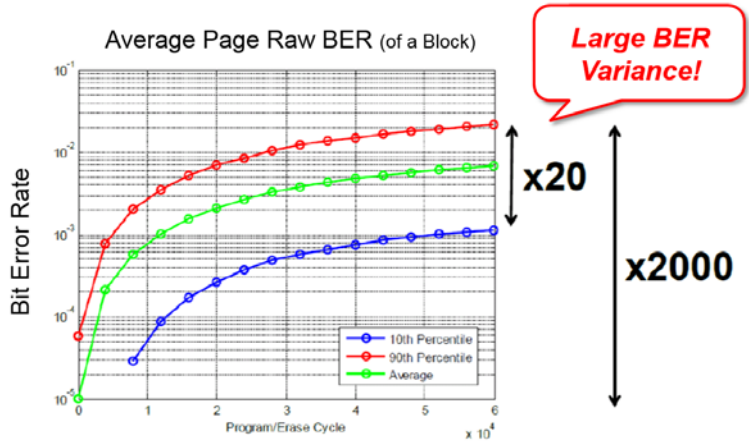
Access pattern		Read	
		Frequently	Infrequently
Write	Frequently	High	High
	Infrequently	Mid	Low

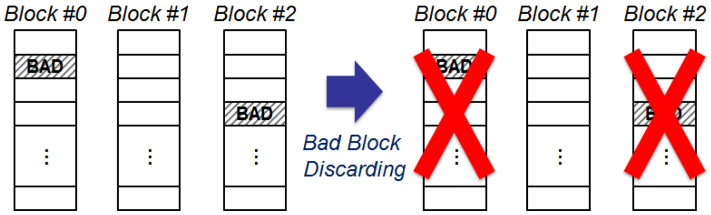
	TLC_{SLC}	TLC_{TLC}	
R (μs)	35~60	90~120	3x
W (ms)	0.25~0.5	2.4	8x

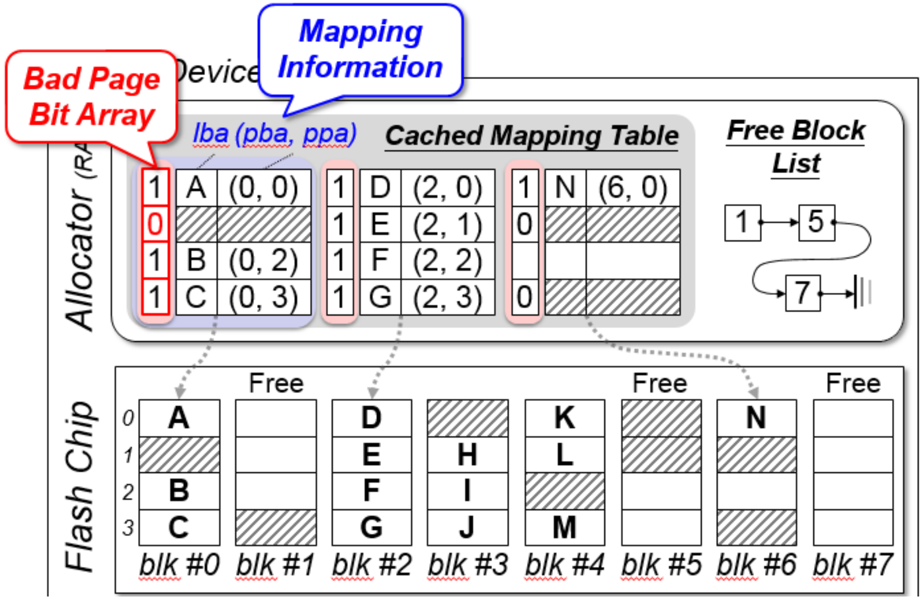
Ming-Chang Yang, Yuan-Hao Chang, Che-Wei Tsao, and Chung-Yu Liu, "Utilization-aware Self-tuning Design for TLC Flash Storage Devices," IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 24, no. 10, pp. 3132-3144, Oct. 2016.

Research Results 2016 (5)

- **Graceful Space Degradation** to Prolong the Lifetime of TLC Flash (IEEE TCAD in 2016)
  - Observation: (1) The bit error rates of pages/blocks are different as they endure more P/E cycles; and (2) low pages have much higher error rate than high pages.
  - **Graceful Space Degradation (or *Bad Block Relaxation*)**: This is the first work that proposes to *mark bad area in the unit of a page* rather than a block to avoid rapid space degradation. (at least 10 times of lifetime extension)



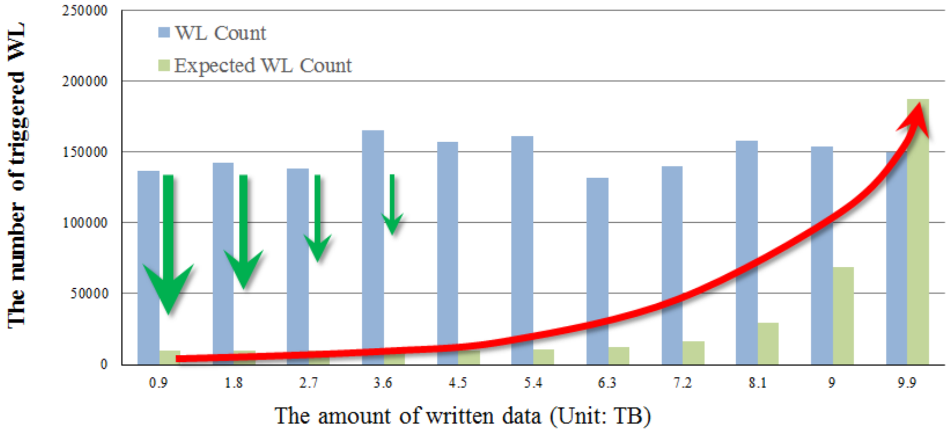




Ming-Chang Yang, Yuan-Hao Chang, Yuan-Hung Kuan, and Che-Wei Tsao, "Graceful Space Degradation: An Uneven Space Management for Flash Storage Devices," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 35, no. 9, pp. 1425-1434, Sep. 2016.

Research Results 2016 (6)

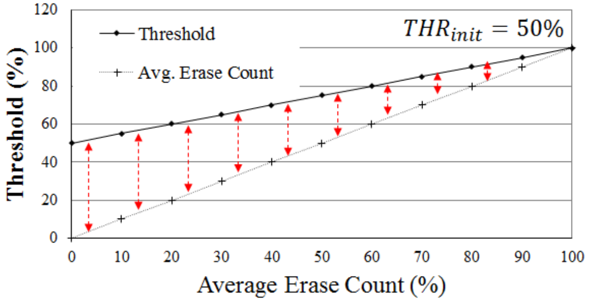
- **Progressive Wear Leveling** for Flash (IEEE TVLSI in 2016)
  - Observation: Existing WL designs control the distribution of block erase counts within a threshold, and such an pro-active approach is unnecessary.
  - Main Idea: Prevent any block from being over-erased instead of controlling the gap of erase counts.
    - Prevent data migration (WL) in the early stages of the device lifetime
    - Progressively increase the WL frequency as the time elapses.
  - WL overheads can be reduced by at least 74%, compared to Rejuvenator.



Average erase count

$$threshold = erase_{avg} * \frac{BLK_{endurance} - THR_{init}}{BLK_{endurance}} + THR_{init}$$

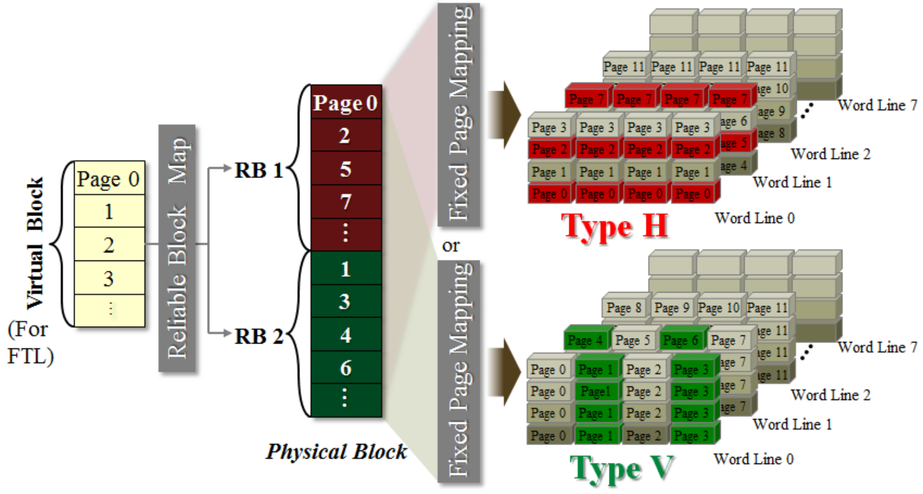
initial threshold: the smallest value to trigger PWL

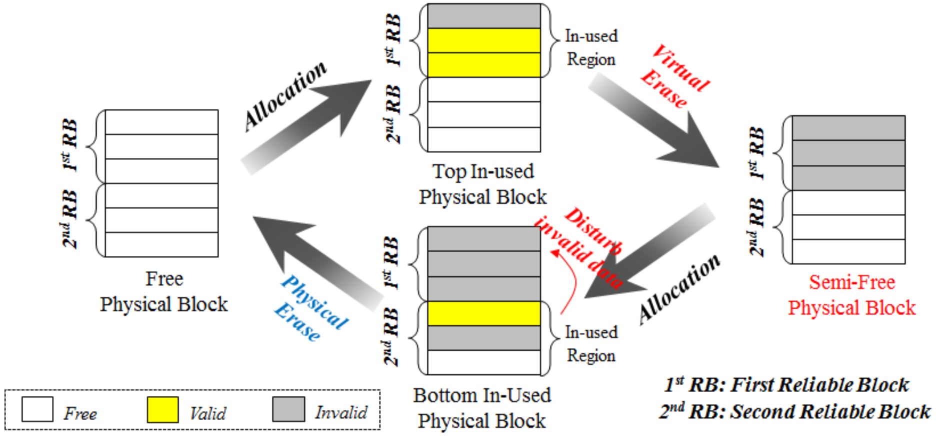


Ming-Chang Yang, Yuan-Hao Chang, Tei-Wei Kuo, and Fu-Hsin Chen, "Reducing Data Migration Overheads of Flash Wear Leveling in a Progressive Way," IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 24, no. 5, pp. 1808-1820, May 2016.

Research Results 2016 (7)

- **Disturbance Alleviation for 3D (MLC) Flash** Memory with Explosive Capacity (IEEE TC in 2016, ICCAD 2015, CODES 2016)
  - The first work that proposes a software solution **with the concept of virtual block and virtual erase** to reduce the disturb bit error rate of **real 3D flash** from 64.03% to 71.13%. (with Macronix).
  - We use software solution to *redirect write disturbs to invalid data*.

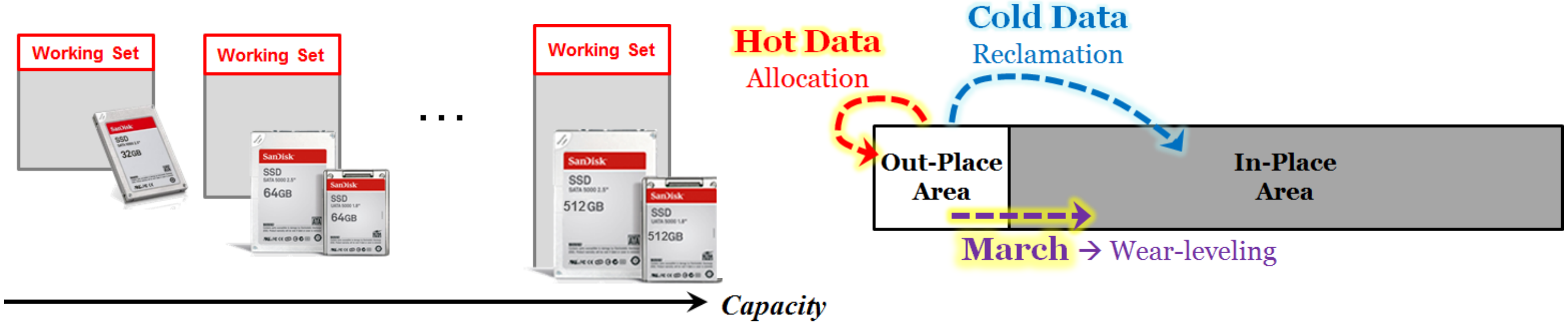




- Yu-Ming Chang, Yuan-Hao Chang, Tei-Wei Kuo, Yung-Chun Li, and Hsiang-Pang Li, "Disturbance Relaxation for 3D Flash Memory," IEEE Transactions on Computers (TC), vol. 65, no. 5, pp. 1467-1483, May 2016.
- Hung-Sheng Chang, Yuan-Hao Chang, Tei-Wei Kuo, Yu-Ming Chang, and Hsiang-Pang Li, "A Disturbance-aware Sub-Block Design to Improve Reliability of 3D MLC Flash Memory," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Pittsburgh, Pennsylvania, USA, Oct. 2-7, 2016. **(Top Conference)**
- Yu-Ming Chang, Yung-Chun Li, Yuan-Hao Chang, Tei-Wei Kuo, Chih-Chang Hsieh, and Hsiang-Pang Li, "On Relaxing Page Program Disturbance over 3D MLC Flash Memory," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), Austin, Texas, USA, Nov. 2-6, 2015. (Acceptance rate: 24.7%(94/381)) **(Top Conference)**

Research Results 2016 (8)

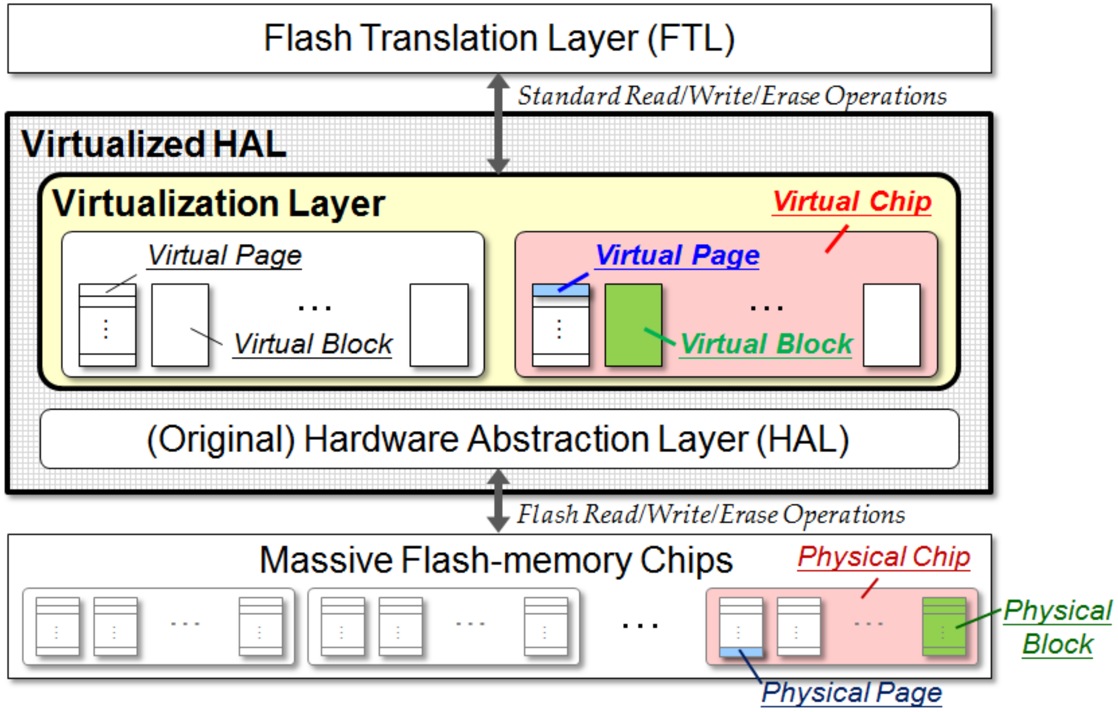
- **Working-Set-Based** Address Mapping for Ultra-Large-Scaled Solid-state Drives (IEEE TC in 2016, CODES in2012)
  - We are the first team to propose a capacity-independent address mapping scheme that only depends on user's access data.
  - The proposed scheme is *at least 1.96 times* of other existing coarse-grained address mapping methods and nearly achieves the performance of the fine-grained address mappings in most cases.



- Ming-Chang Yang, Yuan-Hao Chang, Tei-Wei Kuo, and Po-Chun Huang, "Capacity-independent Address Mapping for Flash Storage Devices with Explosively Growing Capacity," IEEE Transactions on Computers (TC), vol. 65, no. 2, pp. 448-465, Feb. 2016.
- Ming-Chang Yang, Yuan-Hao Chang, Po-Chun Huang, and Tei-Wei Kuo, "Working-Set-Based Address Mapping for Ultra-Large-Scaled Flash Devices," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Tampere, Finland, Oct. 7-12, 2012. **(Top Conference)**

Research Results 2016 (9)

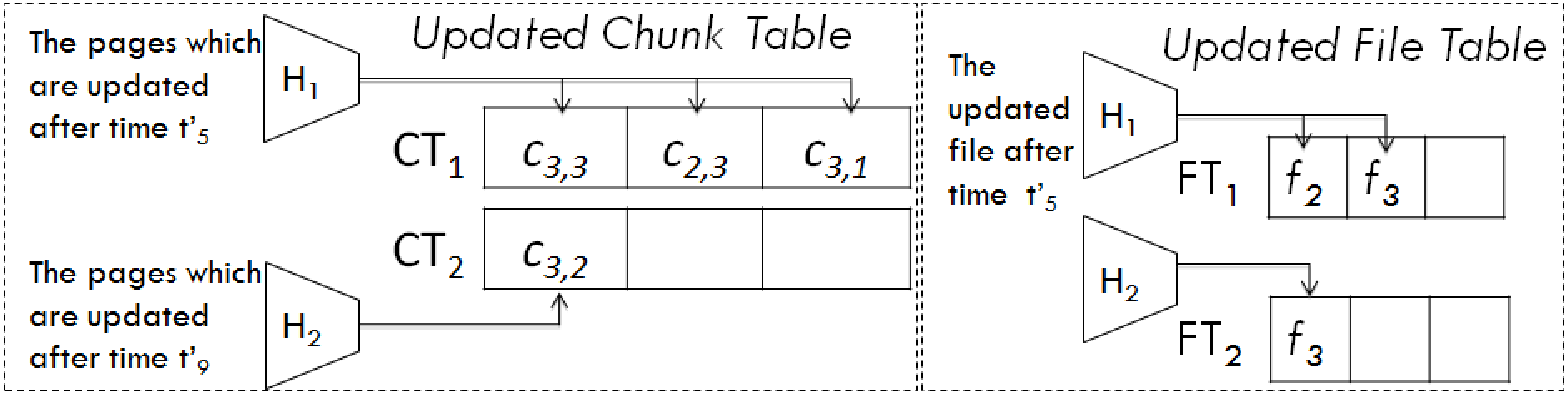
- **Flash Virtualization**: Rethinking Layer Design of Flash Management (IEEE TC in 2016, DAC 2015)
  - We propose to virtualize flash memory called *flash virtualization* to create a uniform virtual flash chip to avoid the redesign of flash management designs due to the fast-changing of 3D flash chips.


Flash Translation Layer (FTL)
Standard Read/Write/Erase Operations
Virtualized HAL
Virtualization Layer
Virtual Chip
Virtual Page
Virtual Page
Virtual Block
Virtual Block
(Original) Hardware Abstraction Layer (HAL)
Flash Read/Write/Erase Operations
Massive Flash-memory Chips
Physical Chip
Physical Block
Physical Page

- Ming-Chang Yang, Yuan-Hao Chang, and Tei-Wei Kuo, "Virtual Flash Chips: Reinforcing the Hardware Abstraction Layer to Improve Data Recoverability of Flash Devices," IEEE Transactions on Computers (TC), vol. 65, no. 9, pp. 2872-2883, Sep. 2016.
- Ming-Chang Yang, Yuan-Hao Chang, and Tei-Wei Kuo, "Virtual Flash Chips: Rethinking the Layer Design of Flash Devices to Improve Data Recoverability," ACM/IEEE Design Automation Conference (DAC), San Francisco, California, USA, Jun. 7-11, 2015. **(Top Conference)**

Research Results 2016 (10)

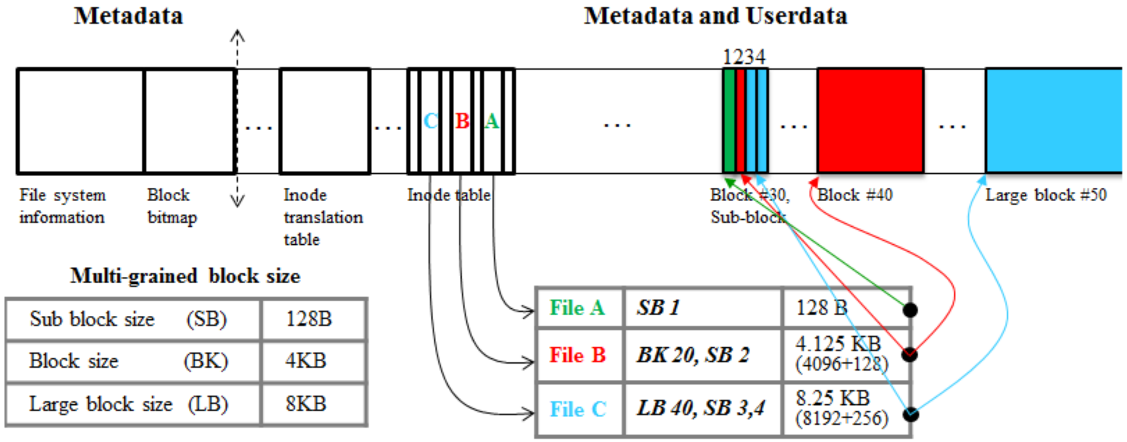
- **Multi-version Checkpointing** for Native Flash File Systems (ASP-DAC – Best Paper Nomination)
  - We utilize the out-place update feature of flash memory to propose a checkpoint-based strategy with optimal space utilization to maintain the consistency among snapshots of a file system for potential recovery needs.



Shih-Chun Chou, Yuan-Hao Chang, Yuan-Hung Kuan, Po-Chun Huang, and Che-Wei Tsao, "Multi-version Checkpointing for Flash File Systems," ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC), Macao, China, Jan. 25-28, 2016. **(Best Paper Nomination)**

Research Results 2016 (11)

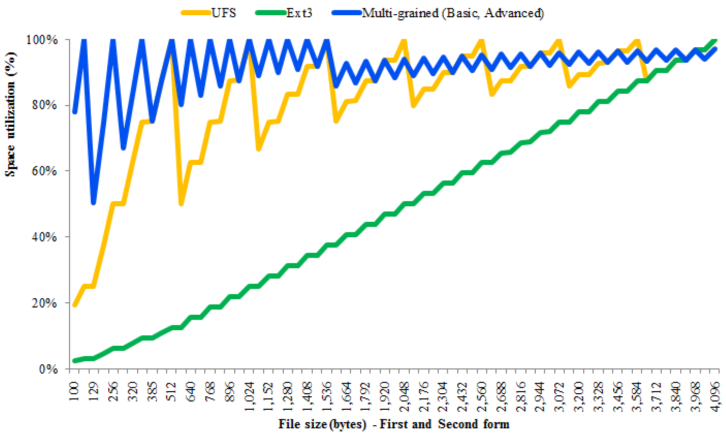
- **Space Utilization for Embedded File Systems** over PCM-based Storage (IEEE TC in 2016)
  - We are among the pioneers to propose “multi-grained” space management, instead of *fixed-sized* one, to utility the byte-addressability of PCM.
  - The proposed scheme can save *up to 80%* (and *18% on average*) of the PCM storage space on storing file data.



Multi-grained block size

Sub block size (SB)	128B
Block size (BK)	4KB
Large block size (LB)	8KB

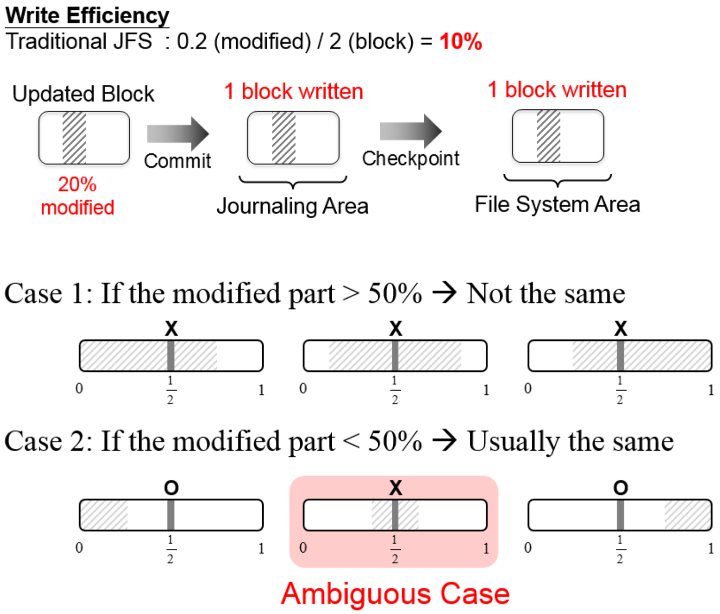
File A	*SB 1*	128 B
File B	*BK 20, SB 2*	4.125 KB (4096+128)
File C	*LB 40, SB 3,4*	8.25 KB (8192+256)

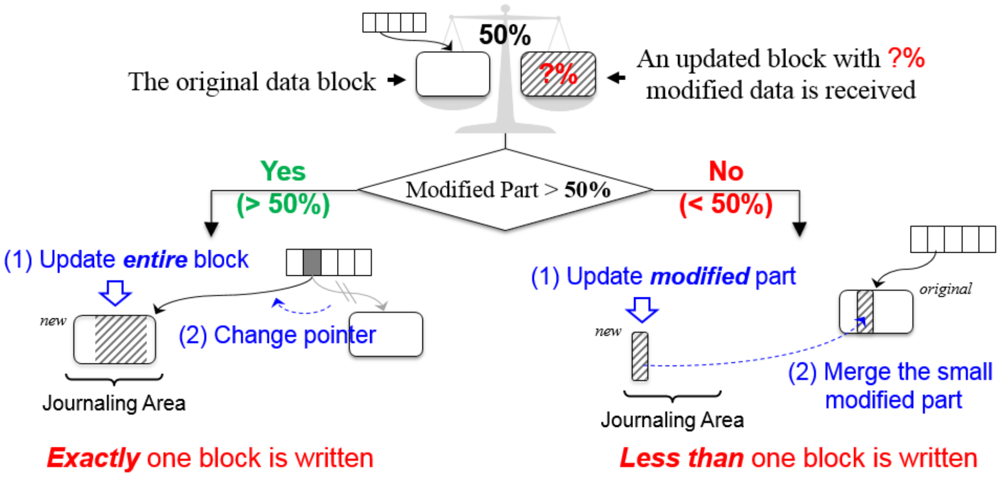


Tseng-Yi Chen, Yuan-Hao Chang, Ming-Chang Yang, Yun-Jhu Chen, Hsin-Wen Wei, and Wei-Kuan Shih, "Multi-grained Block Management to Enhance the Space Utilization of File Systems on PCM Storages," IEEE Transactions on Computers (TC), vol. 65, no. 6, pp. 1831-1845, Jun. 2016.

Research Results 2016 (12)

- **Fifty-percent Rule** to Minimize Write Energy of PCM Storage (ACM TECS in 2016)
  - Observation: File systems usually update data in the unit of a block even when only a small part of the block is modified without taking the byte-addressability of PCM/NVM.
  - We propose a fifty-percent rule to efficiently examine which part and how many data of the updated block are modified to determine which part of data should be updated by utilizing the fact that the updated data are usually consecutive in the same block. (more than 76.8% of energy is saved)

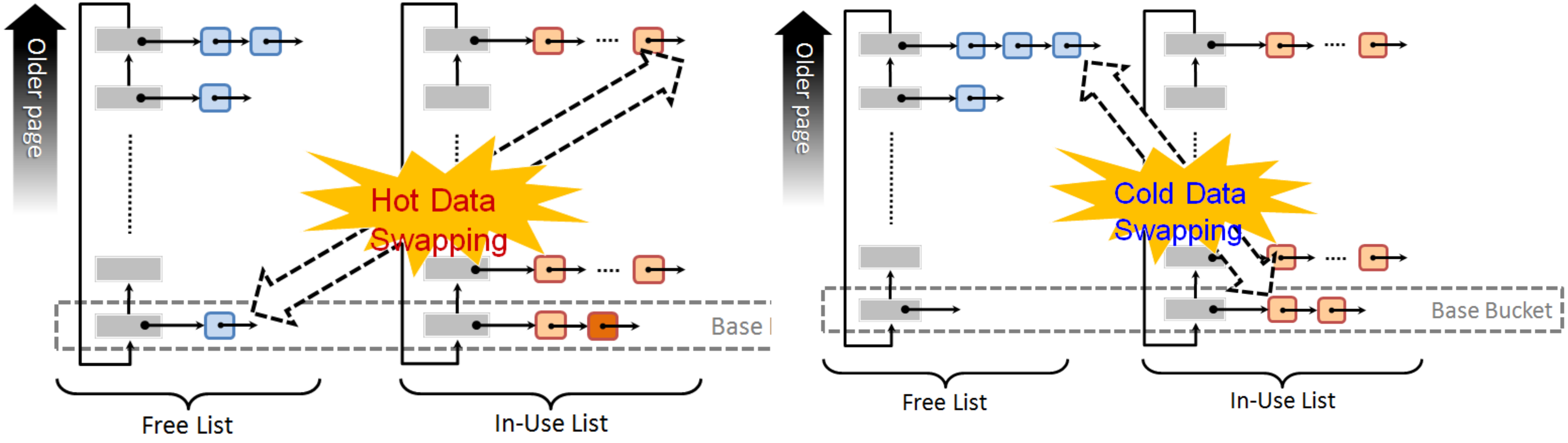

Write Efficiency
Traditional JFS : 0.2 (modified) / 2 (block) = 10%
Updated Block
20% modified
Commit
1 block written
Journaling Area
Checkpoint
1 block written
File System Area
Case 1: If the modified part > 50% → Not the same
X
X
X
0
1/2
1
Case 2: If the modified part < 50% → Usually the same
O
X
O
0
1/2
1
Ambiguous Case


50%
The original data block
?%
An updated block with ?% modified data is received
Modified Part > 50%
Yes (> 50%)
No (< 50%)
(1) Update *entire* block
new
(2) Change pointer
Journaling Area
Exactly one block is written
(1) Update *modified* part
original
new
(2) Merge the small modified part
Journaling Area
Less than one block is written

Ming-Chang Yang, Yuan-Hao Chang, and Che-Wei Tsao, "Byte-addressable Update Scheme to Minimize the Energy Consumption of PCM-based Storage Systems," ACM Transactions on Embedded Computing Systems (TECS), vol. 15, no. 3, pp. 55:1-55:20, Jun. 2016.

Research Results 2016 (13)

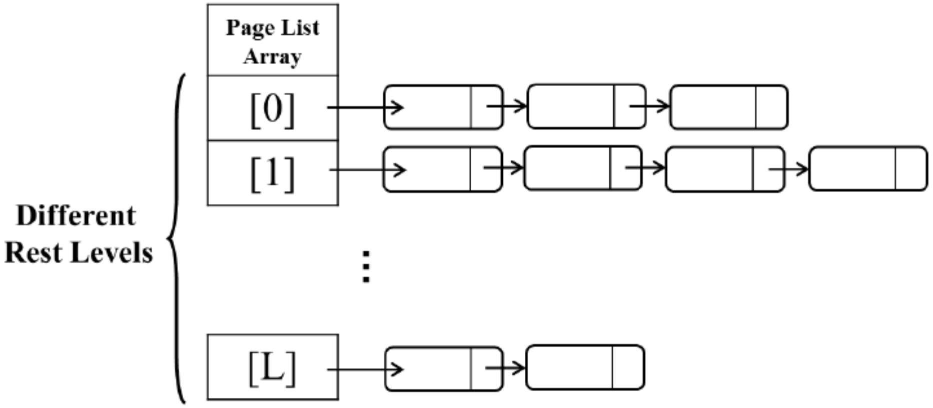
- **Constant-cost PCM Wear Leveling** with Nearly-zero Search Cost (ACM TODAES in 2016)
  - We propose a constant-cost wear leveling design to achieve WL with nearly-zero search cost by realizing the concept of "placing old pages far away so that they are less likely to be used.“
  - The proposed design was implemented in QEMU, and evaluation results show the proposed design can achieve 80% of the lifetime of the ideal case.

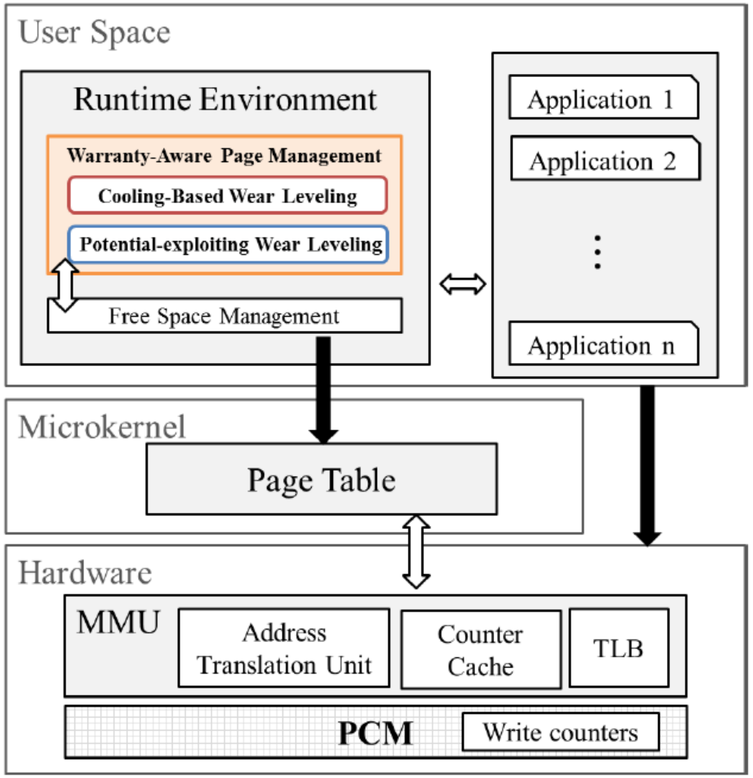


Yu-Ming Chang, Pi-Cheng Hsiu, Yuan-Hao Chang, Chi-Hao Chen, Tei-Wei Kuo, and Cheng-Yuan Michael Wang, "Improving PCM Endurance with a Constant-cost Wear Leveling Design," ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 22, no. 1, pp. 9:1-9:27, Jun. 2016.

Research Results 2016 (14)

- Warranty-Aware Page Management for PCM-Based Embedded Systems (IEEE TVLSI in 2016, ICCAD 2014)
  - Different from existing works using **best efforts** to maximize device lifetime, we are the first team to propose the idea of "**warranty period**" to conduct wear leveling.
  - Wear leveling overhead is less than 20% of existing works with comparable lifetime.

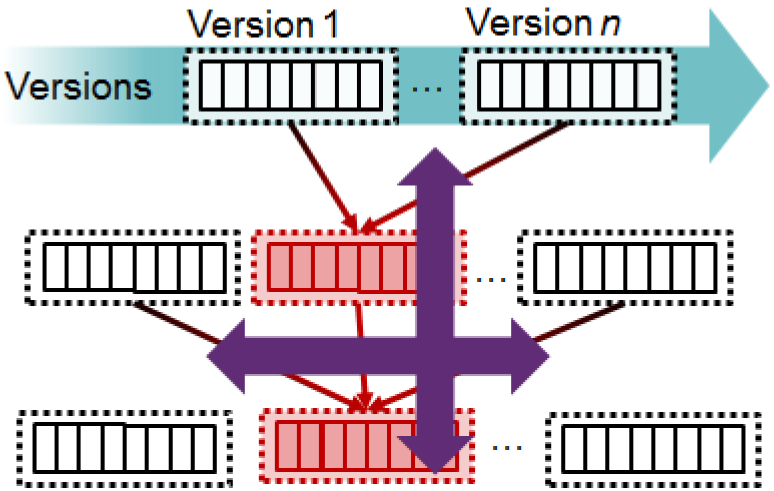


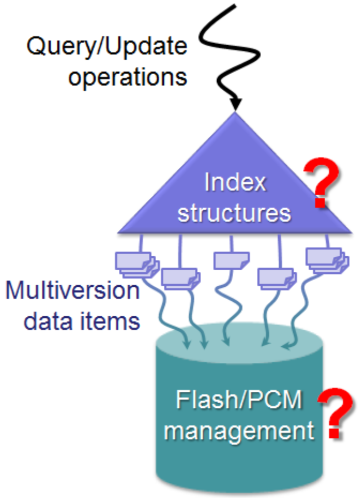


- Sheng-Wei Cheng, Yuan-Hao Chang, Tseng-Yi Chen, Yu-Fen Chang, Hsin-Wen Wei, and Wei-Kuan Shih, "Efficient Warranty-Aware Wear-Leveling for Embedded Systems with PCM Main Memory," IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 24, no. 7, pp. 2535-2547, Jul. 2016.
- Sheng-Wei Cheng, Yu-Fen Chang, Yuan-Hao Chang, Shin-Wen Wei, and Wei-Kuan Shih, "Warranty-Aware Page Management for PCM-Based Embedded Systems," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), San Jose, California, USA, Nov. 3-6, 2014. **(Top Conference)**

Research Results 2016 (15)

- **Embedded Multiversion Database** Designs (ACM TECS in 2016, IEEE TC in 2015, ACM TOADES in 2014, DAC 2014)
  - We are the first team to study how to adopt multiversion databases on *flash storages and PCM storages* as well as to propose solutions to tackle the design issues.
  - The proposed solution can speed up ***range query performance up to several times*** than the existing work.
  - The *space utilization* is more than 80%, more than 2 times of others.



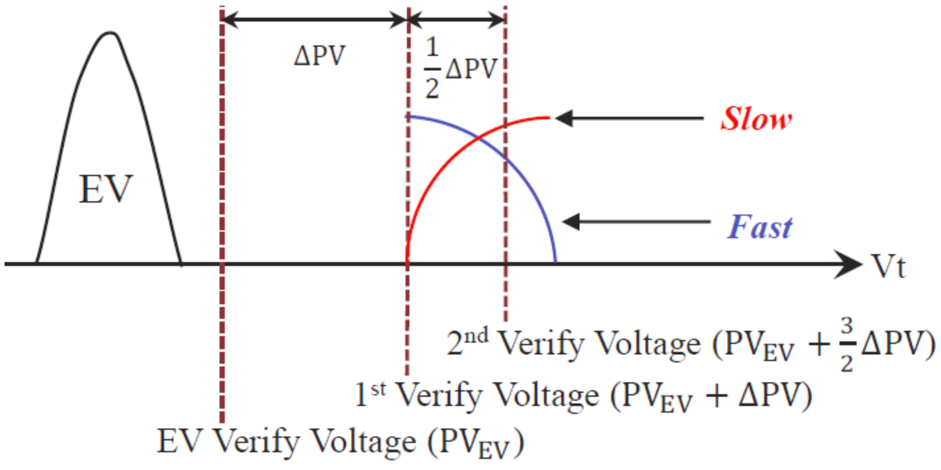


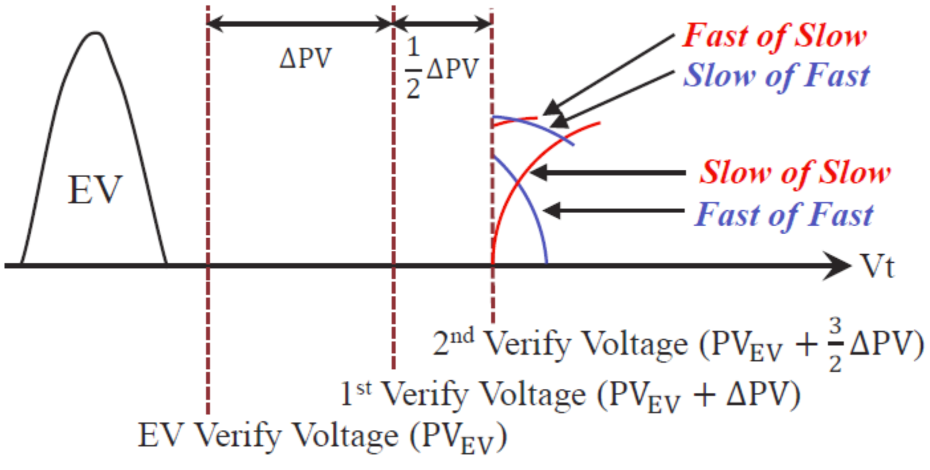
- Yuan-Hung Kuan, Yuan-Hao Chang, Tseng-Yi Chen, Po-Chun Huang, and Kam-Yiu Lam, "Space-Efficient Index Scheme for PCM-based Multiversion Databases in Cyber-Physical Systems," ACM Transactions on Embedded Computing Systems (TECS), vol. 16, no. 1, pp. 21:1-21:26, Oct. 2016.
- Jian-Tao Wang, Kam-Yiu Lam, Yuan-Hao Chang, Jen-Wei Hsieh, and Po-Chun Huang, "Block-based Multi-version B^{+}-Tree for Flash-based Embedded Database Systems," IEEE Transactions on Computers (TC), vol. 64, no. 4, pp. 925-940, April 2015.
- Po-Chun Huang, Yuan-Hao Chang, Kam-Yiu Lam, Jian-Tao Wang, and Chien-Chin Huang, "Garbage Collection for Multiversion Index in Flash-based Embedded Databases," ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 19, no. 3, pp. 25:1-25:27, Jun. 2014.
- Yuan-Hung Kuan, Yuan-Hao Chang, Po-Chun Huang, and Kam-Yiu Lam, "Space-Efficient Multiversion Index Scheme for PCM-based Embedded Database Systems," ACM/IEEE Design Automation Conference (DAC), San Francisco, California, USA, Jun. 1-5, 2014. **(Top Conference)**

Research Summary 2015

Research Results 2015 (1)

- **Relaxing Program Disturb** for 3D MLC Flash (ICCAD)
  - A bi-group programming method is proposed resolve the *slow cell effects* (in ISPP), and can reduce more than *93% of bit errors*.
  - The proposed method is orthogonal to ***wear leveling*** and ***ECC***.
  - Main idea: Assign proper program voltages for cells with different program speeds by means of classifying and programming the cells simultaneously and in a progressive way.

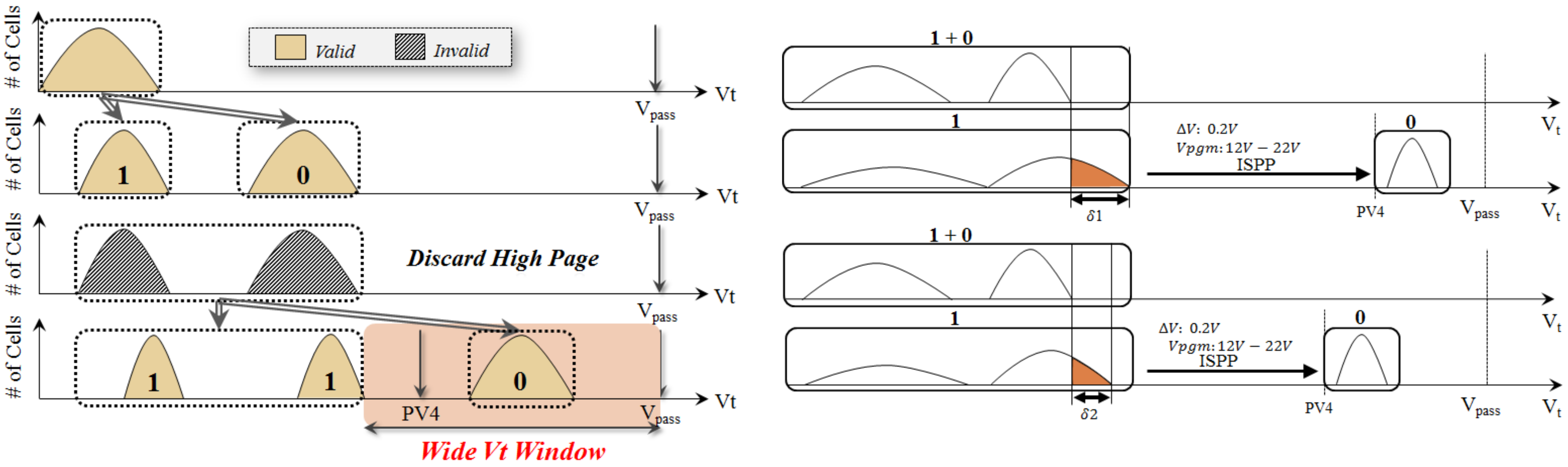

EV
ΔPV
$\frac{1}{2}\Delta PV$
Slow
Fast
Vt
2^{nd} Verify Voltage ($PV_{EV} + \frac{3}{2}\Delta PV$)
1^{st} Verify Voltage ($PV_{EV} + \Delta PV$)
EV Verify Voltage (PV_{EV})


EV
ΔPV
$\frac{1}{2}\Delta PV$
Fast of Slow
Slow of Fast
Slow of Slow
Fast of Fast
Vt
2^{nd} Verify Voltage ($PV_{EV} + \frac{3}{2}\Delta PV$)
1^{st} Verify Voltage ($PV_{EV} + \Delta PV$)
EV Verify Voltage (PV_{EV})

Yu-Ming Chang, Yung-Chun Li, Yuan-Hao Chang, Tei-Wei Kuo, Chih-Chang Hsieh, and Hsiang-Pang Li, "On Relaxing Page Program Disturbance over 3D MLC Flash Memory," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), Austin, Texas, USA, Nov. 2-6, 2015. (Acceptance rate: 24.7%(94/381)) **(Top Conference)**

Research Results 2015 (2)

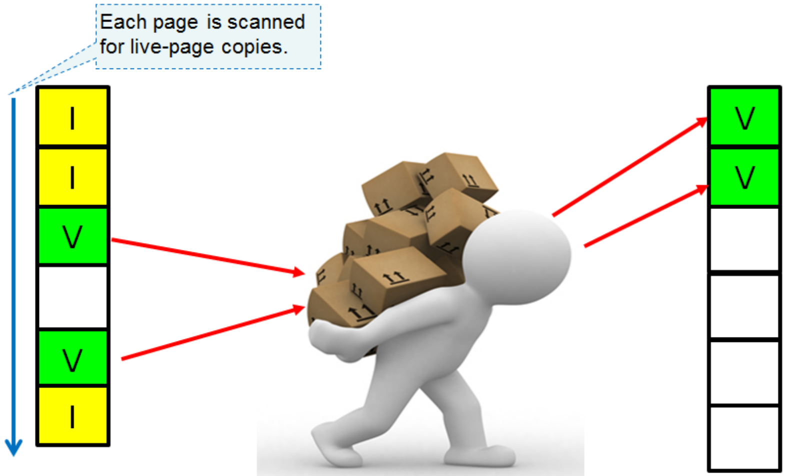
- **Achieving SLC Performance with MLC Flash** (DAC)
  - We propose a ***trim-like program*** to intelligently utilize the knowledge of the *data validity* so as to program low page with the speed of SLC flash.
  - Resolve the fundamental issue of ISPP in programming MLC flash.



Yu-Ming Chang, Yuan-Hao Chang, Tei-Wei Kuo, Yung-Chun Li, and Hsiang-Pang Li, "Achieving SLC Performance with MLC Flash Memory," ACM/IEEE Design Automation Conference (DAC), San Francisco, California, USA, Jun. 7-11, 2015. **(Top Conference)**

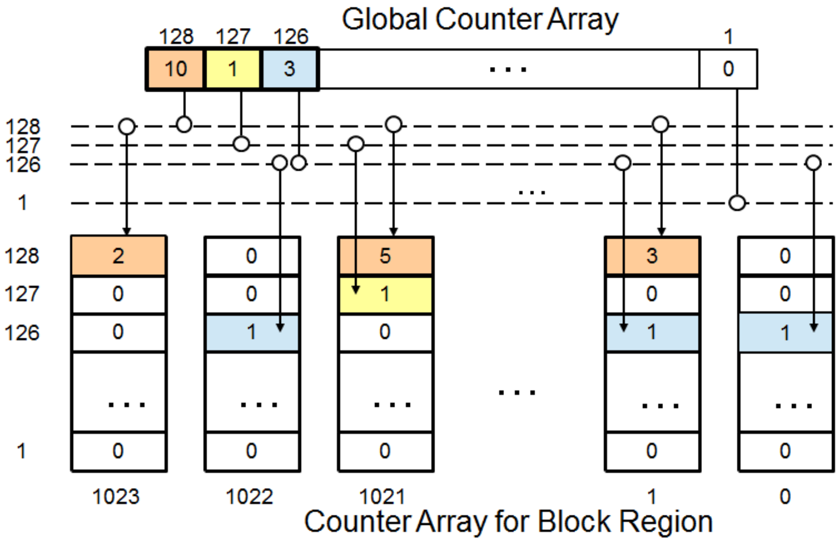
Research Results 2015 (3)

- **Victim Block Selection** Design for the Garbage Collection of Flash Memory (IEEE TC in 2015, DAC 2013)
  - The first work that points out **the ignored enormous time overhead on scanning for victims for space reclamation**, and also proposes a solution to minimize the scanning time overhead.
  - The proposed strategy outperform greedy solutions up to 69.89% and is comparable to the ideal case (<1%).



Each page is scanned for live-page copies.

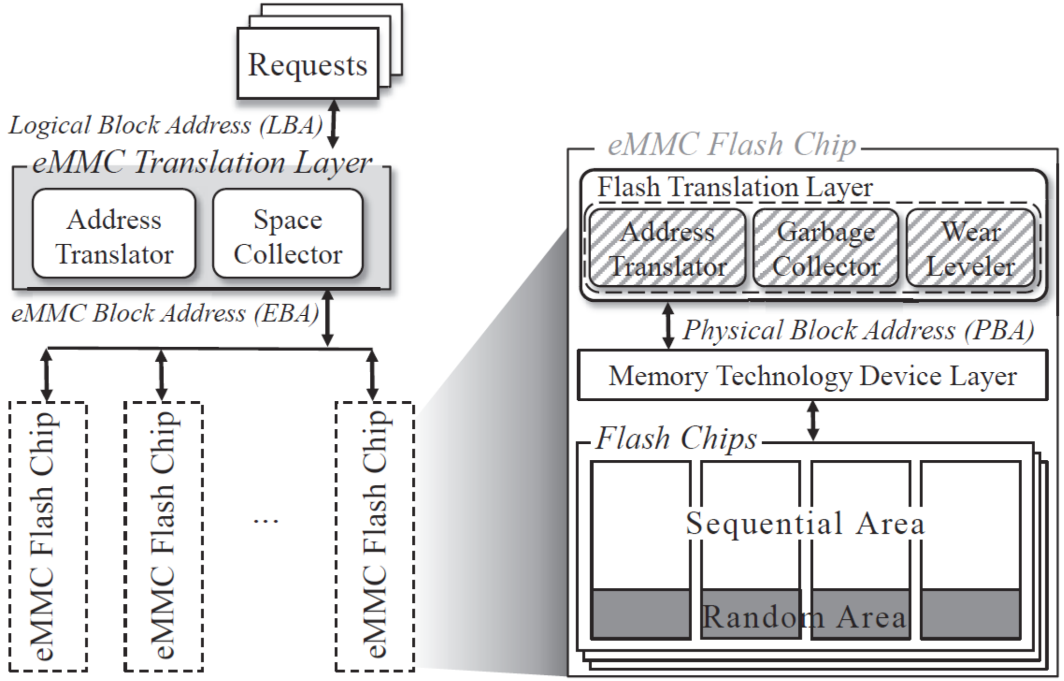
For example, scanning one block needs 60 us X 128 = 7680 us = 7.68 ms.
Suppose that writing one page takes 800us, 7,68ms is enough for writing 7680/800 = 9.6 live pages.

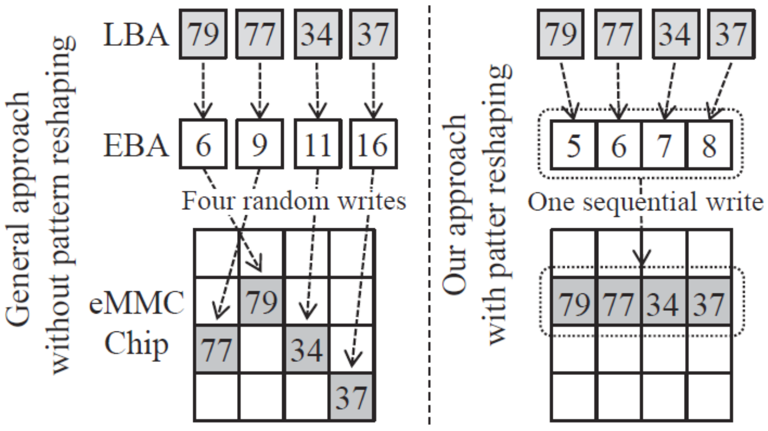


- Che-Wei Tsao, Yuan-Hao Chang, Ming-Chang Yang, and Po-Chun Huang, "Efficient Victim Block Selection for Flash Storage Devices," IEEE Transactions on Computers (TC), vol. 64, no. 12, pp. 3444-3460, Dec. 2015.
- Che-Wei Tsao, Yuan-Hao Chang, and Ming-Chang Yang, "Performance Enhancement of Garbage Collection for Flash Storage Devices: An Efficient Victim Block Selection Design," ACM/IEEE Design Automation Conference (DAC), Austin, Texas, USA, Jun. 2-6, 2013. **(Top Conference)**

Research Results 2015 (4)

- Access Pattern Reshaping for **eMMC-based SSDs** (ICCAD 2015)
  - We propose using ***eMMC*** to replace ***raw flash chips*** in the design of SSDs. (with Genesys)
  - It relaxes the software complexity and scalability of SSDs. We create an eMMC-friendly access pattern to enable eMMC-based SSDs.

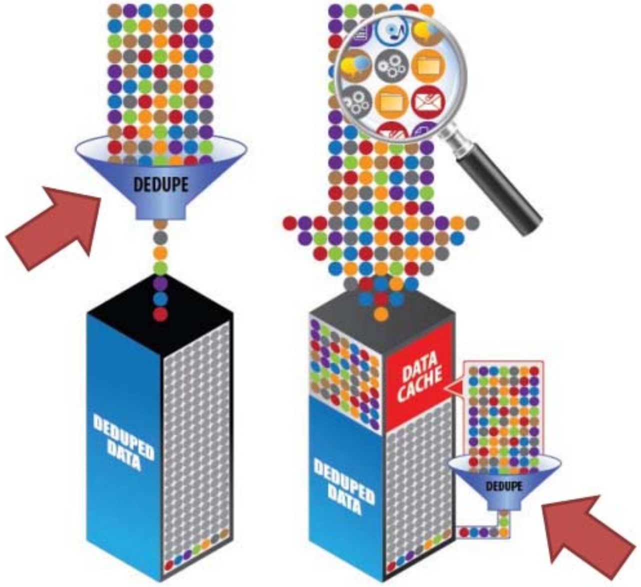


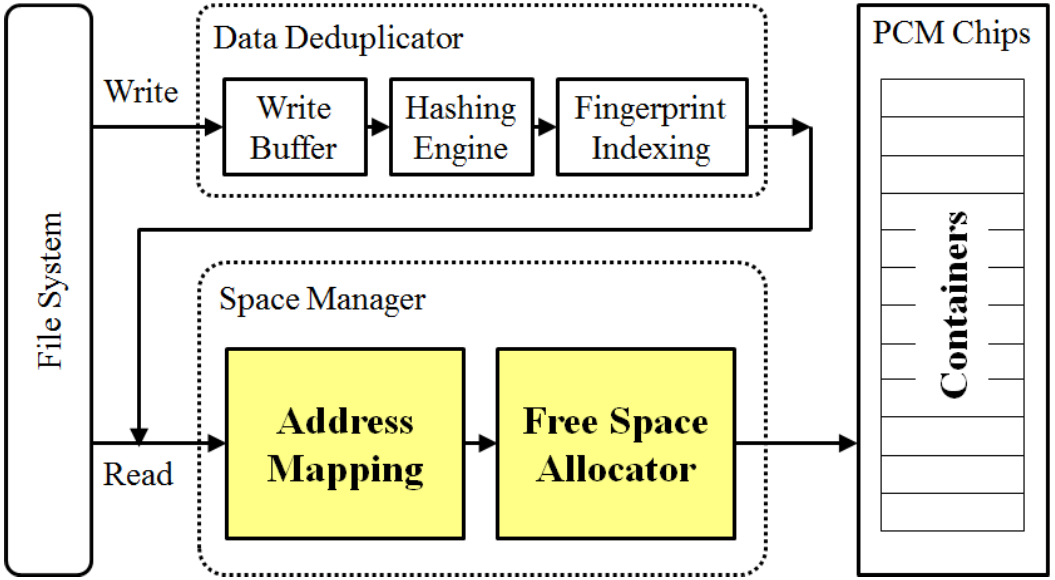


Chien-Chung Ho, Yuan-Hao Chang, and Tei-Wei Kuo, "Access Pattern Reshaping for eMMC-enabled SSDs," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), Austin, Texas, USA, Nov. 2-6, 2015. (Acceptance rate: 24.7%(94/381)) **(Top Conference)**

Research Results 2015 (5)

- **Dedup-based PCM Storage** (CODES 2015)
  - We adopt deduplication to enable PCM storage *storing more data than its capacity*. (PCM suffers from ***high cost per GB!***)
  - A container-based management is proposed to effectively manage variable-sized chunks due to the deduplication.

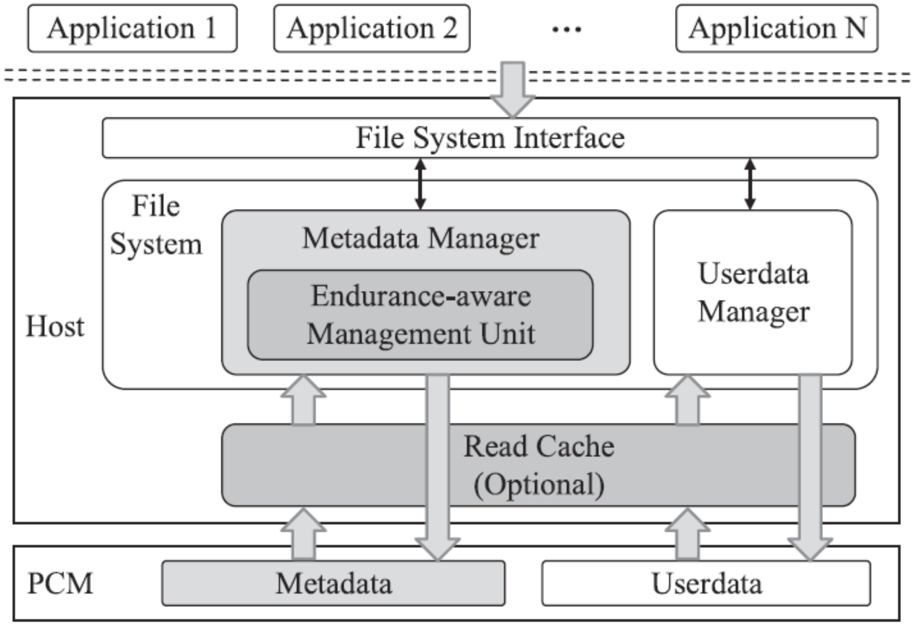


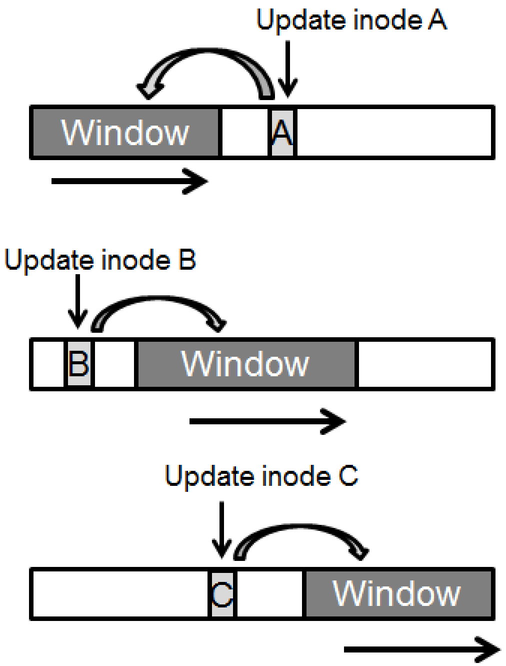


Chun-Ta Lin, Yuan-Hao Chang, Tei-Wei Kuo, Hung-Sheng Chang, and Hsiang-Pang Li, "How to Improve the Space Utilization of Dedup-based PCM Storage Devices," ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Amsterdam, The Netherlands, Oct. 4-9, 2015. **(Top Conference)**

Research Results 2015 (6)

- Marching-based Wear Leveling for PCM Storage (ACM TODAES in 2015)
  - We propose to use PCM as the storage space of file systems.
  - A marching-based idea is proposed to avoid wearing out any PCM cell.
    - Main idea: Collect writes into the marching window, and spread writes by sliding the marching window.

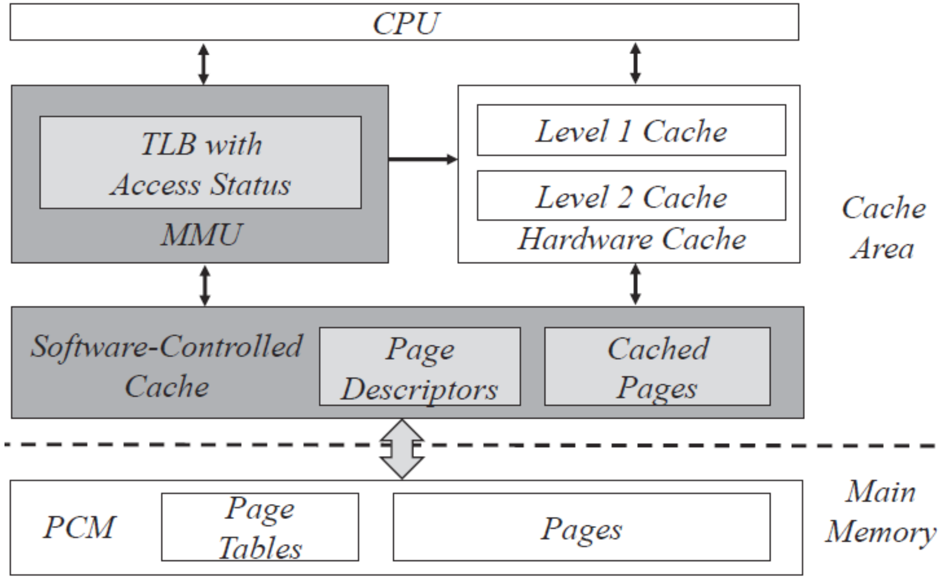


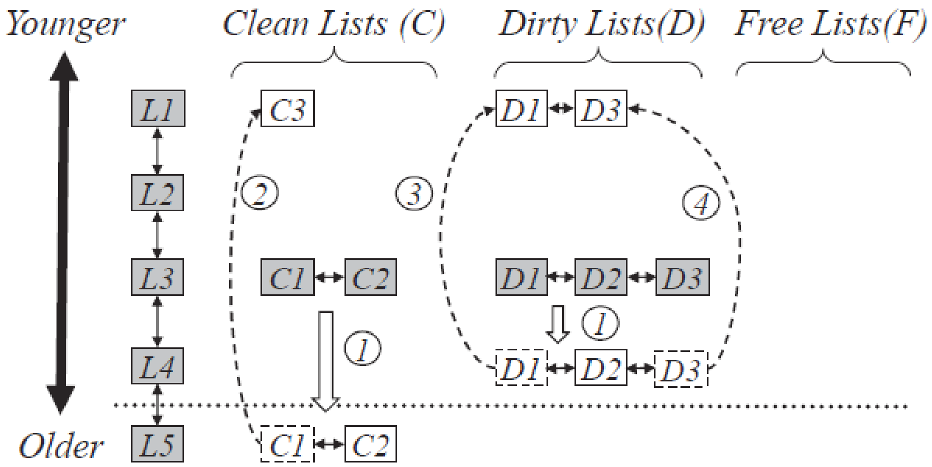


Hung-Sheng Chang, Yuan-Hao Chang, Pi-Cheng Hsiu, Tei-Wei Kuo, and Hsiang-Pang Li, "Marching-based Wear Leveling for PCM-based Storage Systems," ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 20, no. 2, pp. 25:1-25:22, Feb. 2015.

Research Results 2015 (7)

- A Light-weighted Software-Controlled Cache for NVM Main Memory (ICCAD 2015)
  - We consider the model “PCM main memory with SRAM as its cache”.
  - A software-controlled cache design is proposed to prevent updates to the management data structure of the SRAM cache on each update by utilizing the message of *TLB miss* or *cache miss*.



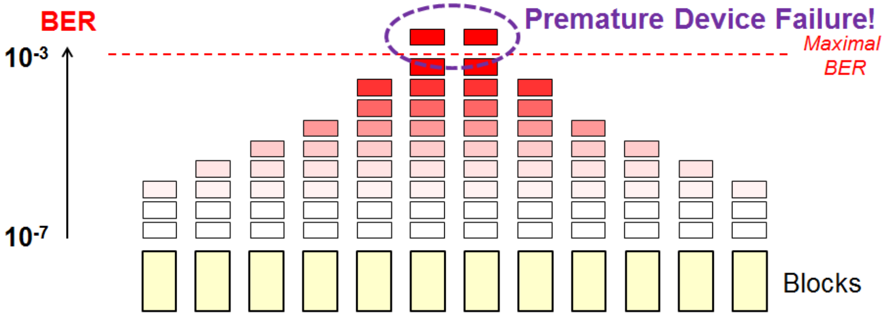


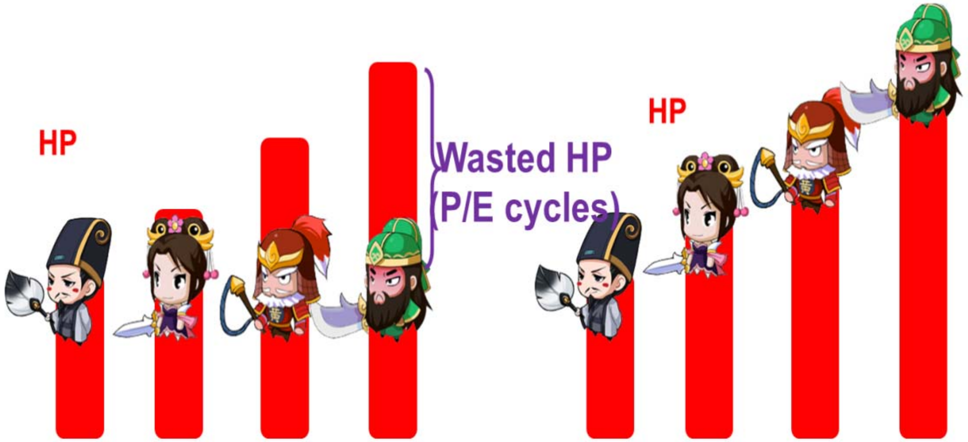
Hung-Sheng Chang, Yuan-Hao Chang, Tei-Wei Kuo, and Hsiang-Pang Li, "A Light-Weighted Software-Controlled Cache for PCM-based Main Memory Systems," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), Austin, Texas, USA, Nov. 2-6, 2015. (Acceptance rate: 24.7%(94/381)) **(Top Conference)**

Research Summary 2014

Research Results 2014 (1)

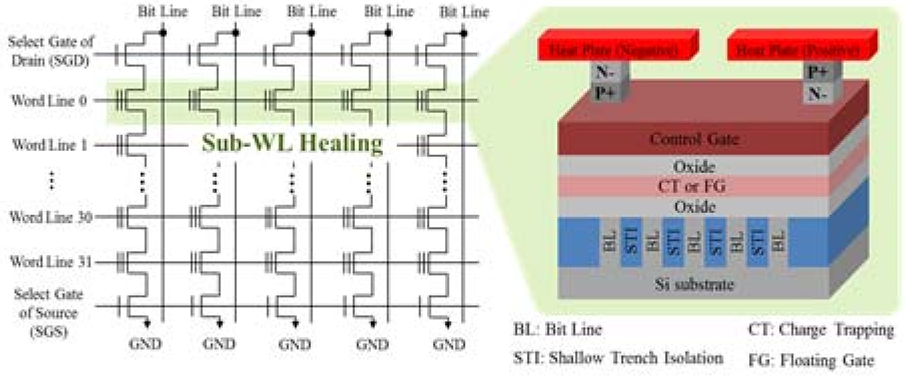
- **New ERA**: New Efficient Reliability-Aware Wear Leveling (DAC)
  - Instead of using the indirect index “***erase count***” to conduct wear leveling, we are the first team to propose using the direct hardware information “***error rate***” to conduct wear leveling.
  - New ERA is *at least 2.5 times* better than the optimal solution with “erase count” as the wear leveling index.



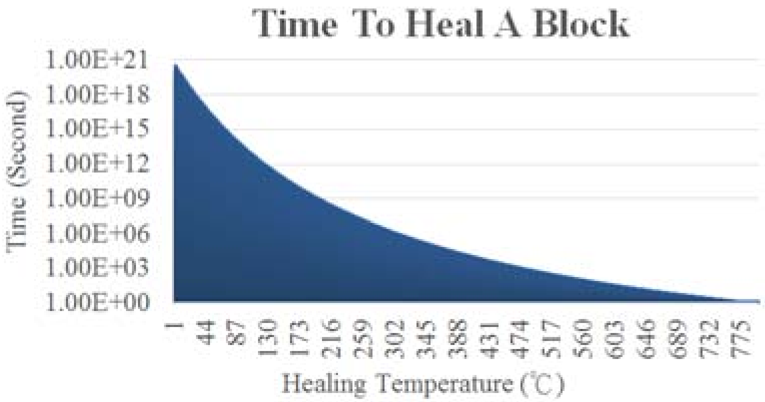


Research Results 2014 (2)

- **Heal-Leveling** for 3D Flash Memory (DAC – Best Paper Nomination (from 787 submissions))
  - If self-healing can be integrated into flash devices, the flash industry will be wholly changed.
  - We are the first team to adopt **heal-leveling** on real 3D flash to significantly enhance 3D flash's lifetime and replace wear leveling. (with Marocnix)

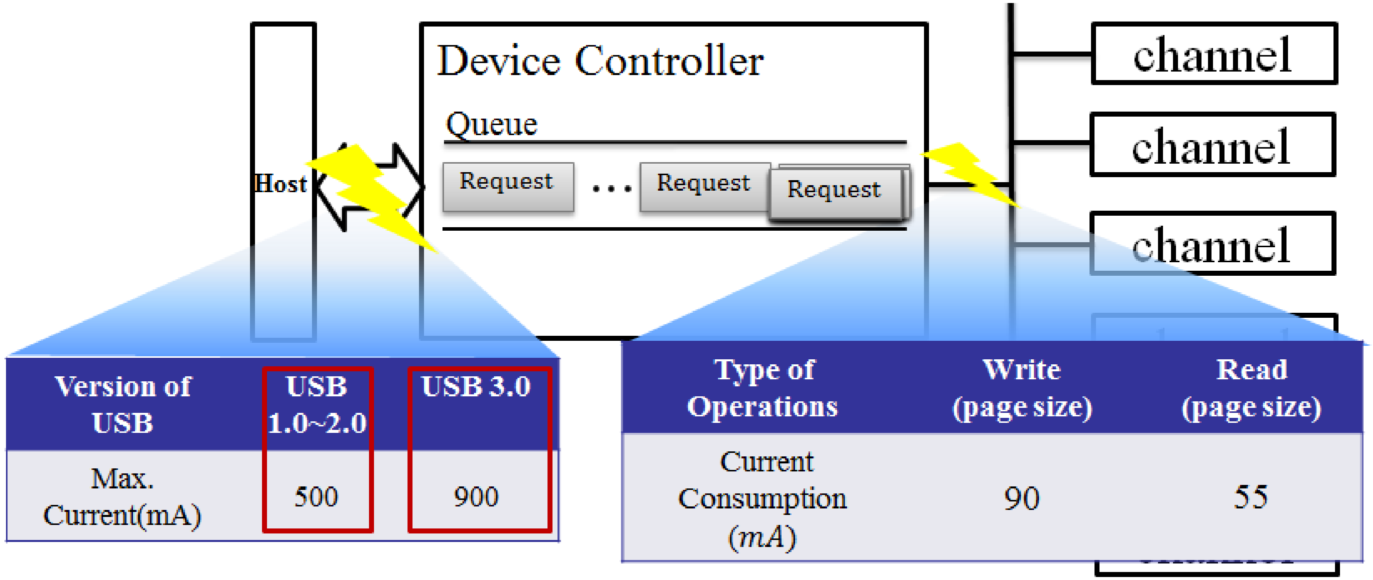


Internal Heating Architecture



Research Results 2014 (3)

- **Current-Aware Scheduling** and Index Design for Low-cost Flash Storage Devices (RTCSA, ACM TODAES)
  - A scheduling algorithm to optimize performance of removable PCIe-based flash devices with considering current constraints.
  - The error bound of the ***read makespan*** between our online scheduler and the optimal scheduler is no longer than $T_{page\_write} - T_{page\_read}$

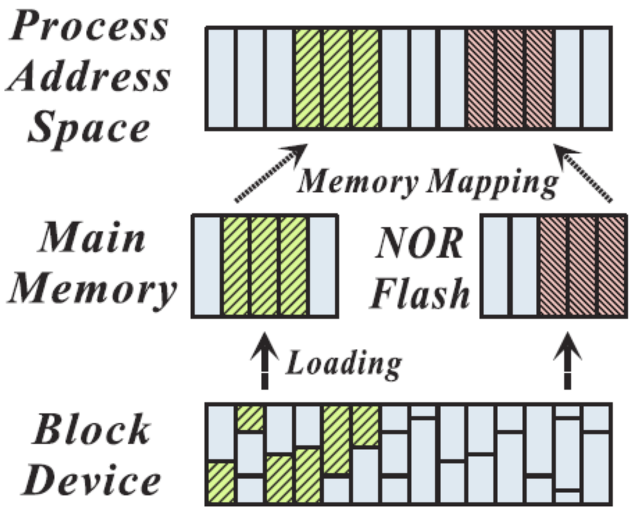


Version of USB	USB 1.0~2.0	USB 3.0
Max. Current(mA)	500	900

Type of Operations	Write (page size)	Read (page size)
Current Consumption (mA)	90	55

Research Results 2014 (4)

- Booting Time Minimization for Real-Time Embedded Systems (IEEE TC)
  - We propose a 0.25-approximation greedy algorithm to guarantee the booting time of a real-time embedded system.

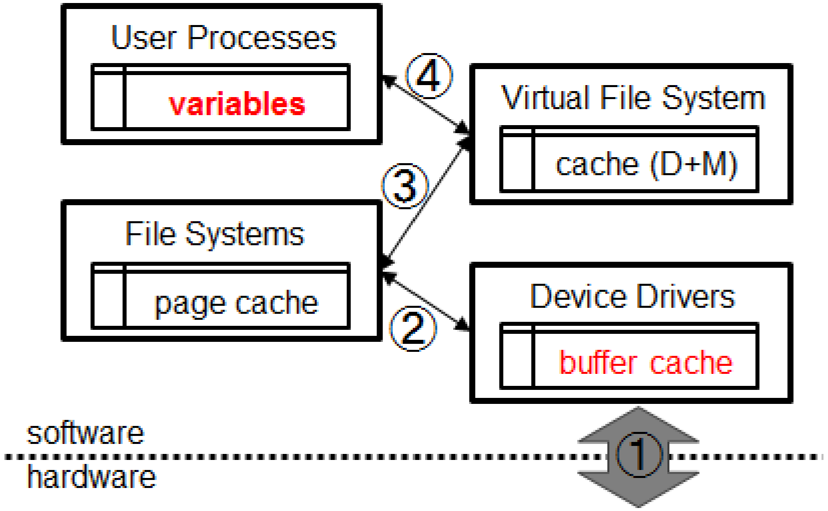


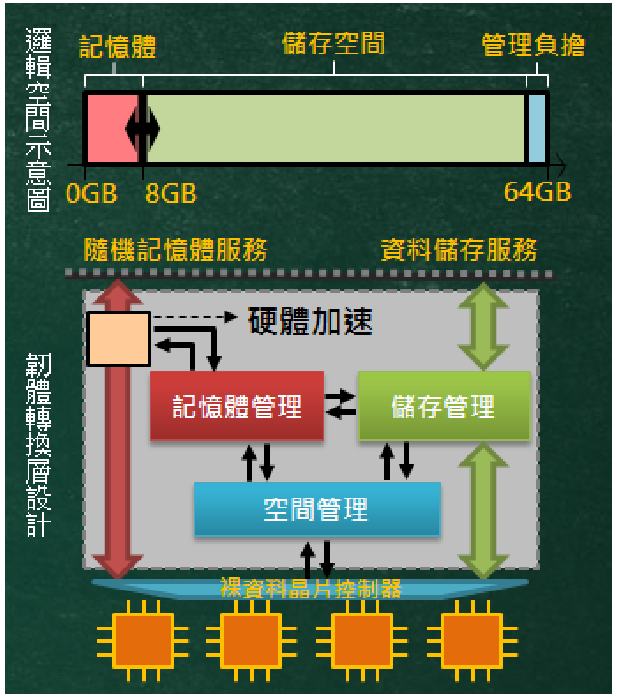
(a) The system architecture

(b) The XIP digital frame

Research Results 2014 (5)

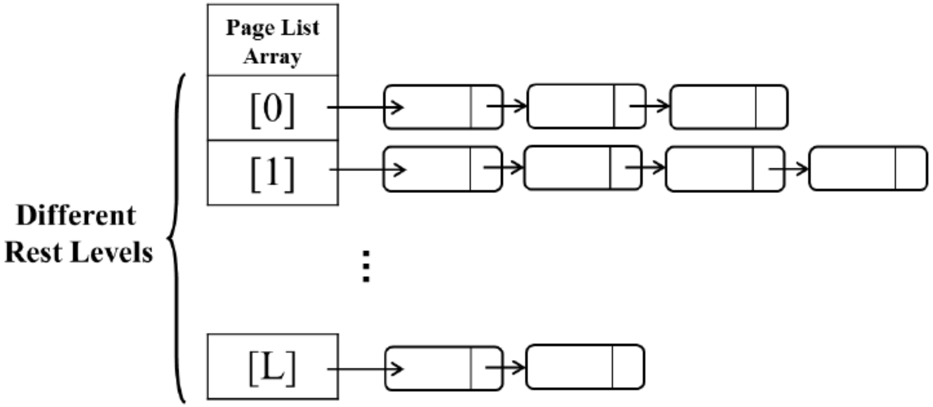
- ***One memory*** – Using NVM as both Memory and Storage (CODES – Best Paper Nomination)
  - – We propose the concept of “one memory” by using NVM as both memory and storage.
  - – We are the **first team** to develop joint management of memory and storage
    - To reduce the data movement overheads.
    - To resolve the lifetime issue of NVM.
      - Stealing the lifetime of the large storage space to rescue the lifetime of the small memory space.

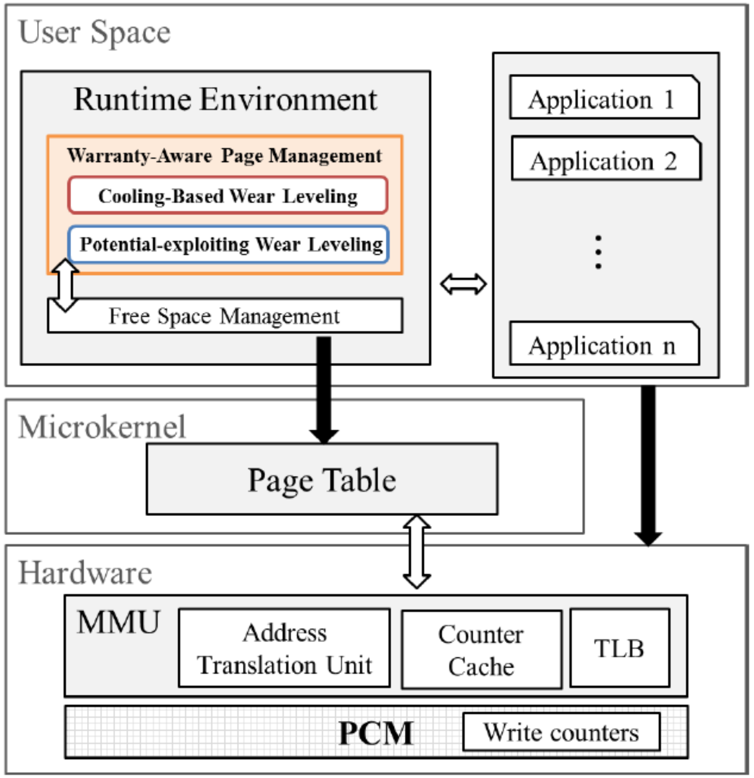




Research Results 2014 (6)

- Warranty-Aware Page Management for PCM-Based Embedded Systems (ICCAD)
  - Different from existing works using **best efforts** to maximize device lifetime, we are the first team to propose the idea of "**warranty period**" to conduct wear leveling.
  - Wear leveling overhead is less than 20% of existing works with comparable lifetime.

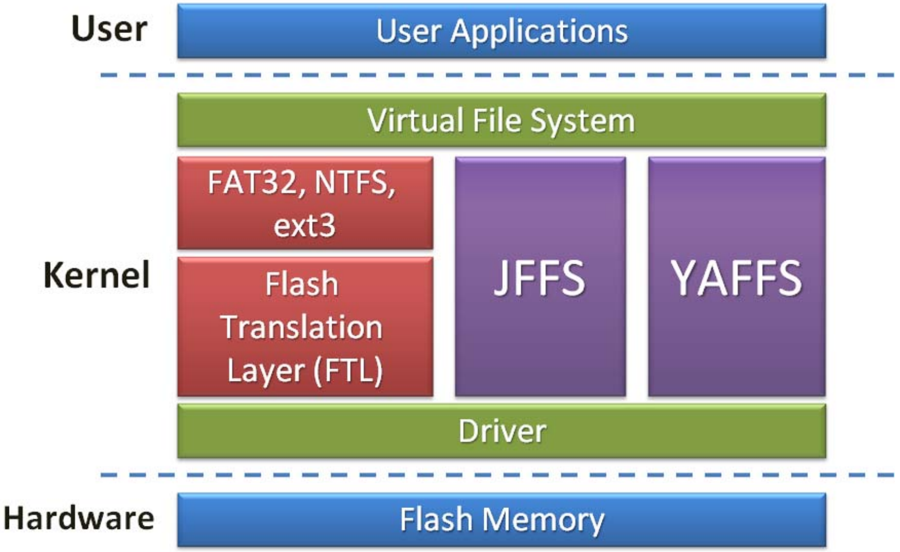


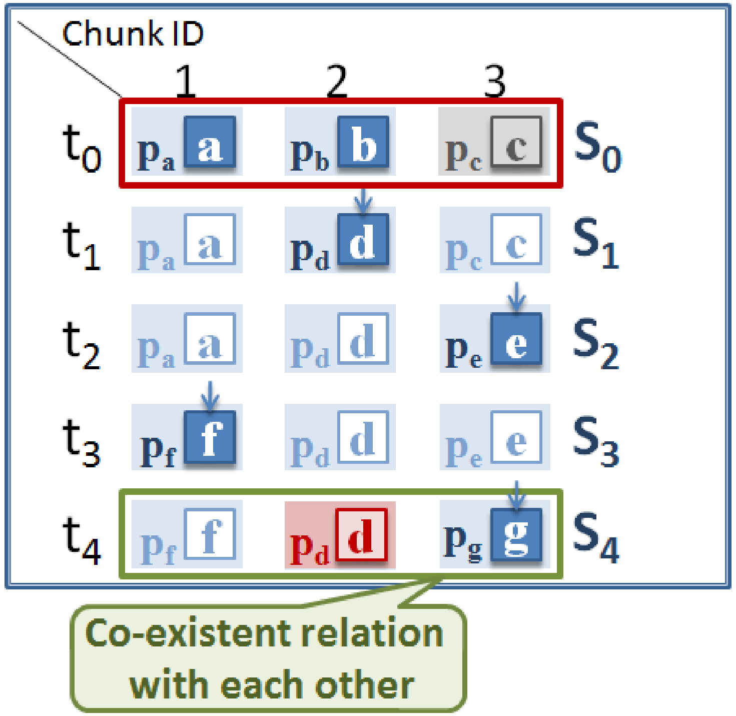


Research Summary 2013

Research Results 2013 (1)

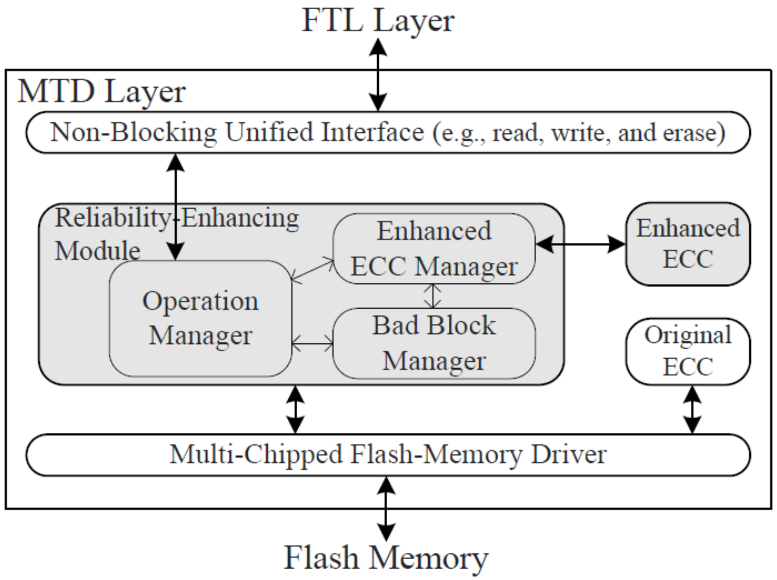
- Version-based Strategy for Native Flash File Systems (IEEE TC)
  - We utilize the out-place update feature of flash memory to propose a version-based strategy with optimal space utilization to maintain the consistency among page versions of a file for potential recovery needs.

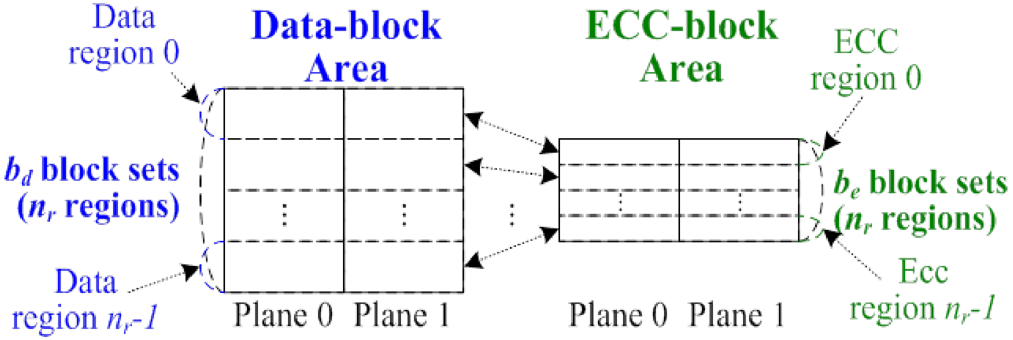



Chunk ID
1
2
3
t_0 p_a a p_b b p_c c S_0
t_1 p_a a p_d d p_c c S_1
t_2 p_a a p_d d p_e e S_2
t_3 p_f f p_d d p_e e S_3
t_4 p_f f p_d d p_g g S_4
Co-existent relation with each other

Research Results 2013 (2)

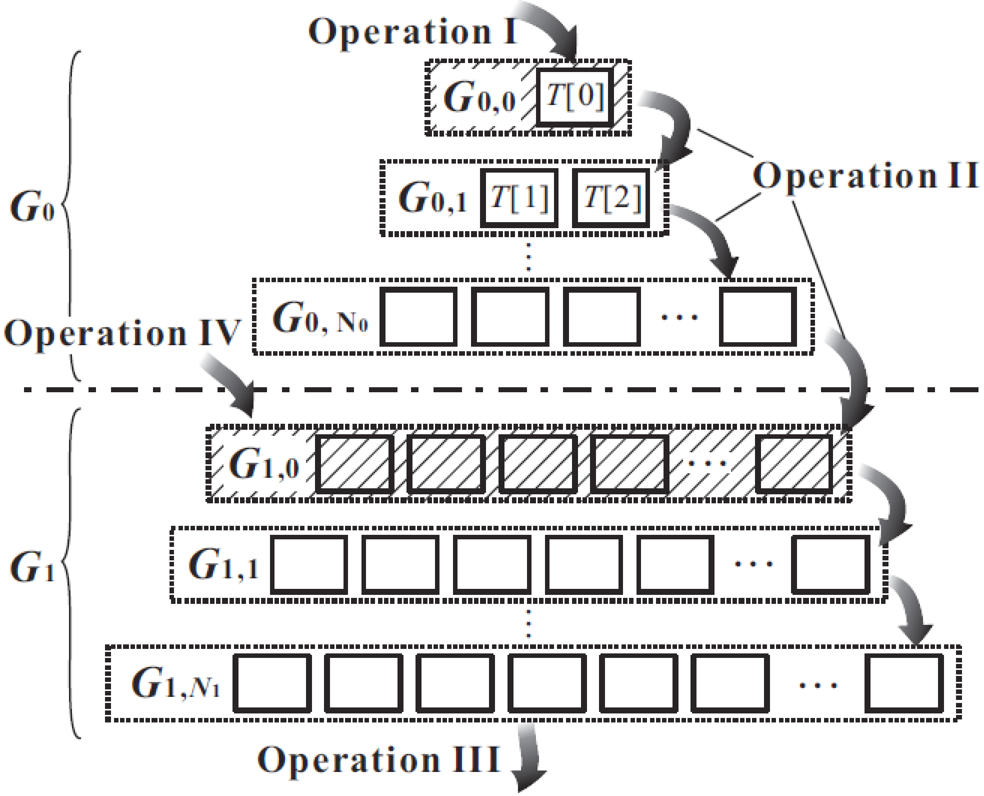
- Reliability Enhancement Design under the Flash Translation Layer (ACM TECS)
  - Different from existing work that proposes solutions at the management layer, we propose to improve flash *reliability* at the ***hardware abstraction layer*** that can leverage the knowledge from both SW and HW ends. (with VIA)


FTL Layer
MTD Layer
Non-Blocking Unified Interface (e.g., read, write, and erase)
Reliability-Enhancing Module
Enhanced ECC Manager
Operation Manager
Bad Block Manager
Enhanced ECC
Original ECC
Multi-Chipped Flash-Memory Driver
Flash Memory



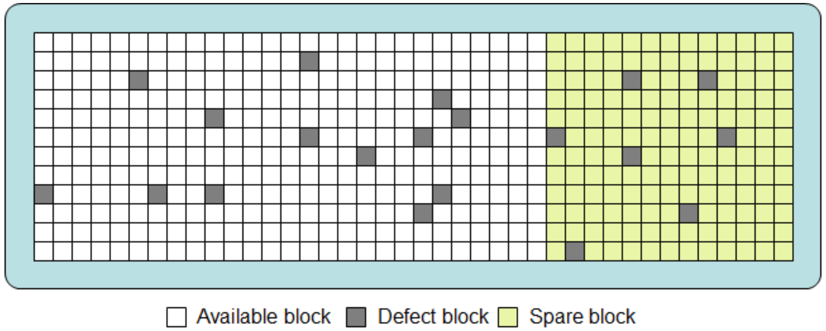
Research Results 2013 (3)

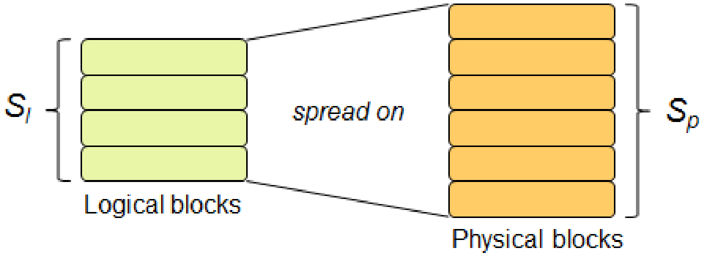
- Joint Management of RAM and Storage (RACS, CODES, DAC, ACM TODAES)
  - By considering RAM as part of storage space, we propose a new design philosophy to jointly manage RAM and storage.



Research Results 2013 (4)

- Downgraded Flash-Memory Management (ACM TECS)
  - We propose to use software solution to enable downgraded flash to be used in low-end storage market. (with Genesys)

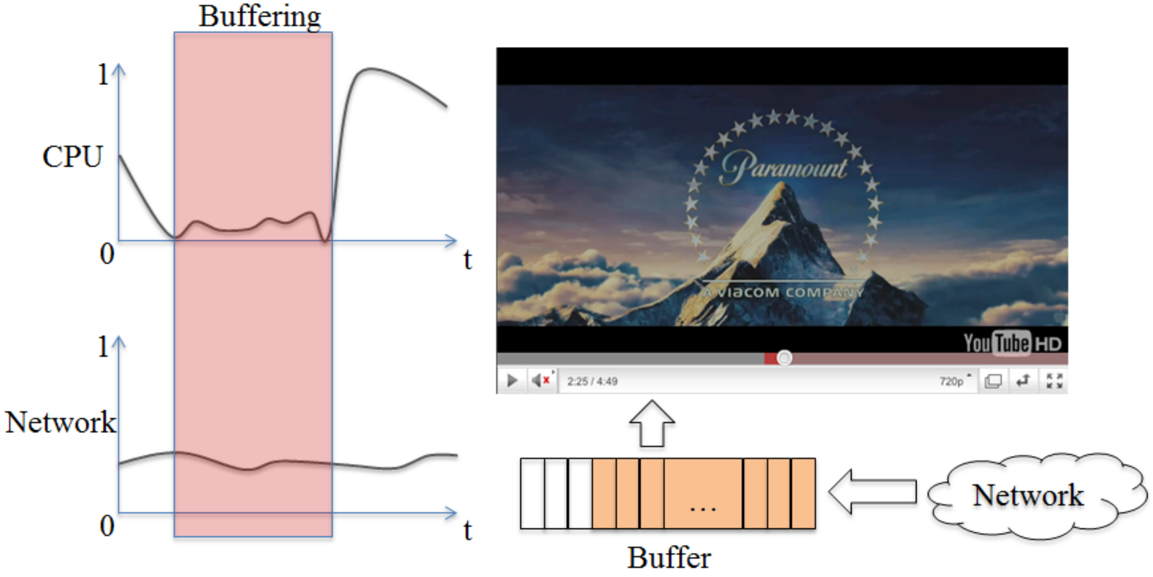




$$S_p \geq 0,\ S_l \geq 0,\ S_p \geq S_l$$

Research Results 2013 (5)

- **Resource-Driven DVFS Scheme** for the Enhancement of Energy-Efficiency of Smart Phones (ACM TECS)
  - Inspired by an observation on the resource usage patterns of mobile applications, we propose a resource-driven DVFS scheme, in which resource state machines are designed to model the resource usage patterns in an online fashion to guide DVFS.



Research Summary 2012 and Before

Research Results 2012

- 17. Management Strategy for Solid-State Drives (ACM TOS)
  - An efficient caching-oriented flash management scheme for multi-chipped SSDs with cache support.
- 18. Working-Set-Based Address Mapping for Ultra-Large-Scaled Solid-state Drives (CODES)
  - We are the first team to propose a capacity-independent address mapping scheme that only depends on user’s access data.
  - The proposed scheme is *at least 1.96 times* of other existing coarse-grained address mapping methods and nearly achieves the performance of the fine-grained address mappings in most cases.
- 19. File System Aware Storage Design (ACM TECS)
  - By analyzing the partition of file systems, the firmware on managing storage media can optimize the performance on writing data to the storage media.

- Yuan-Hao Chang, Cheng-Kang Hsieh, Po-Chun Huang, and Pi-Cheng Hsiu, "A Caching-Oriented Management Design for the Performance Enhancement of Solid-State Drives," ACM Transactions on Storage (TOS), vol. 8, no. 1, pp. 3:1-3:21, Feb. 2012.
- Yuan-Hao Chang, Po-Liang Wu, Tei-Wei Kuo, and Shih-Hao Hung, "An Adaptive File-System-Oriented FTL Mechanism for Flash-Memory Storage Systems," ACM Transactions on Embedded Computing Systems (TECS), vol. 11, no. 1, pp. 9:1-9:19, Mar. 2012.

The Roadmap

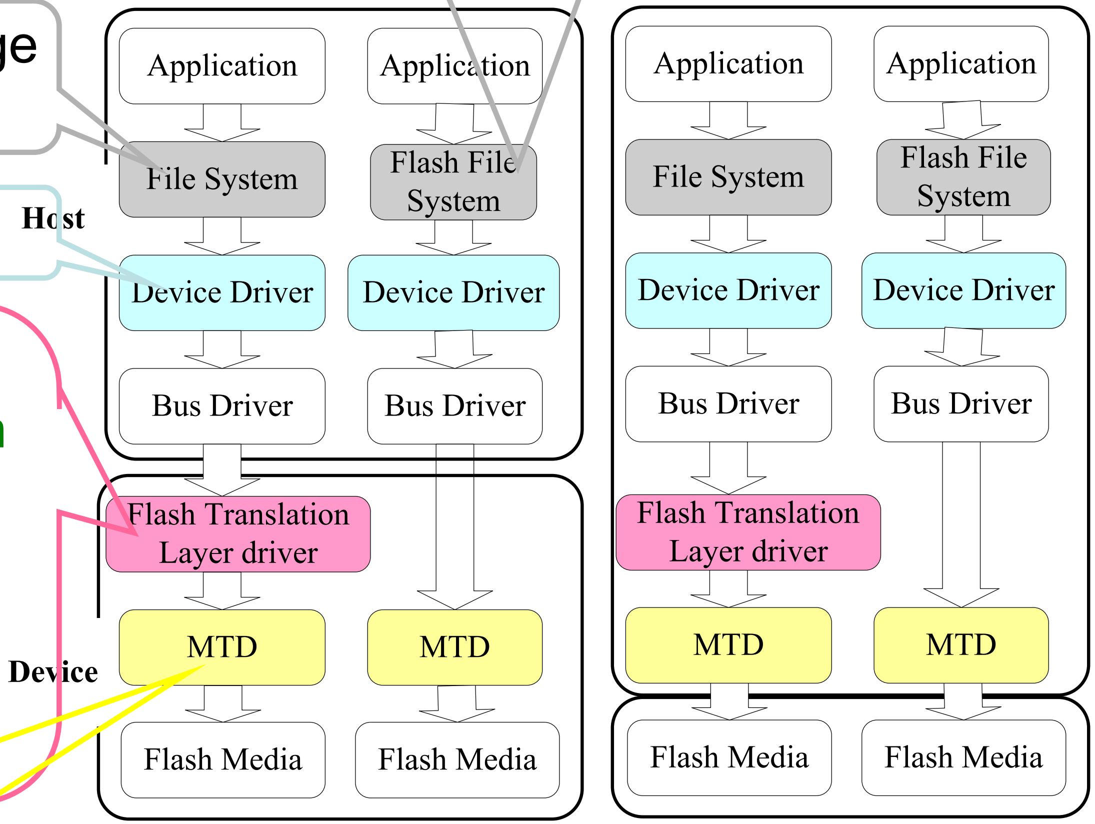
Version-based strategy

Caching design with phase-change memory for file systems

Cohesive caching approach

- Hot-cold data identification
- **File-system-aware FTL design**
- **Static wear leveling design**
- **Commitment-based strategy**
- NOR emulation with NAND
- Set-based mapping strategy
- Caching-oriented FTL for SSD

A reliable MTD design



Past Achievements

- ***Storage Systems for Embedded Systems***
  - Efficient wear leveling design (ACM/IEEE DAC 2007 Best Paper Nomination, IEEE Trans. on Comp. 2010)
  - Enabling XIP with block device (IEEE RTCSA 2007, ACM Trans. on Storage 2010)
  - Access behavior analysis and benchmark design (IEEE RTCSA 2007, ACM/IEEE ICCAD 2008)
  - Set-based design for downgraded devices (ACM/IEEE DATE 2009, ACM Trans. on Embedded Comp. Systems.)
  - Scalable management for flash memory (ACM/IEEE DAC 2009, IEEE Trans. on Comp. 2011)
  - RAID-like design in solid-state drives (IEEE RTCSA 2010, ACM Trans. on Storage 2012)
  - Reliability enhancement for future multi-level-cell flash memory (ACM/IEEE EMSOFT 2010, ACM Trans. on Embedded Comp. Systems)
  - Joint management of memory and storage (ACM/IEEE DAC 2012)

Past Achievements (Cont.)

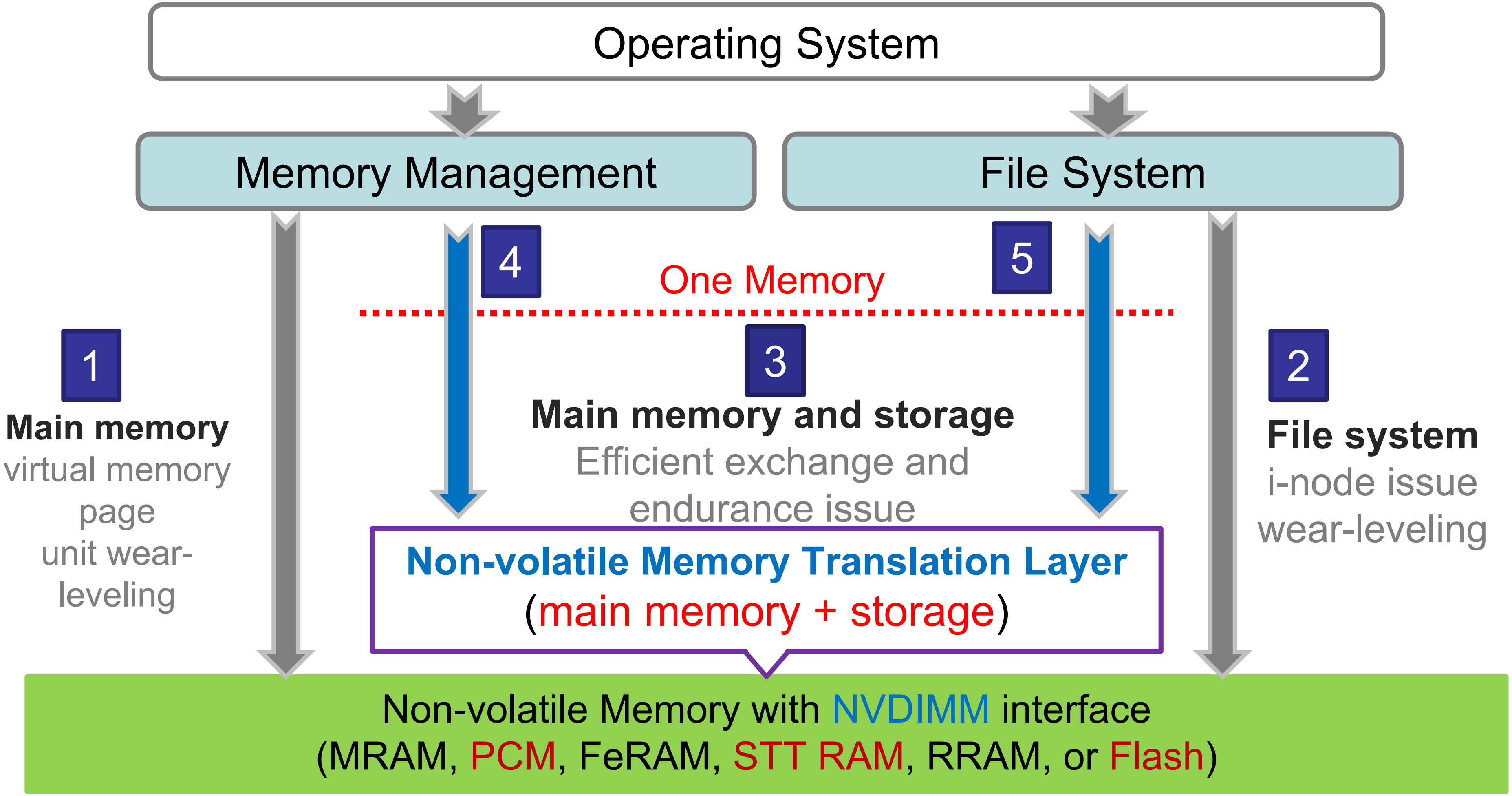
- **File systems and their cache systems for embedded systems**
  - Cohesive cache design with request clustering (ACM Trans. on Storage 2012, technology transferred to Genesys Logic)
  - Integrated cache design with phase-change memory (IWSSPS)
  - Content-aware filtering (ACM/IEEE DATE 2009, ACM Trans. on Embedded Comp. Systems 2012)
  - Version-based design for embedded file systems (ACM/IEEE DAC 2011)
- **(Operating) system designs for embedded systems**
  - Energy-efficient mapping for virtual cores (IEEE ECRTS 2010)
  - Leakage-aware scheduling (ACM/IEEE ASP-DAC 2011)
  - Real-time memory controlling (ACM/IEEE CODES 2011)
  - Efficient fault detection algorithm for memory devices (ACM SIGAPP ACR)

- **Database for embedded systems**
  - Cluster-based management for multi-version B-tree (IEEE RTAS 2012)

рахмат
danke
謝謝
ngiyabonga
spas
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
thank you
gracias
kiitos
dankie
nanni
nandri
dhanyavad
mauruuru
köszönöm
hvala
gràcie
bayarlalaa
bedankt
enkosi
dankon
aciū
akun
хвала
asante
manana
obrigada
murakoze
tenki
djiere dieuf
tau
mochchakkeram
дякую
mamnun
chokrane
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
sobodi
dėkuji
sagolun
sukriya
kop khun krap
arigatô
takk
dakujem
trugarez
obrigado
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
shukriya
merce
мерси
kam sah hamnida
rahmat
terima kasih
rahmet
ευχαριστώ
diolch
dhanyavadagalu
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci

Stretched Goal – NV Computing with One Memory

- Non-volatile computing = Non-volatile processor + one memory
- One memory = Non-volatile memory as both ***main memory*** and ***storage***



Operating System

Memory Management

File System

One Memory

4

5

1

Main memory

virtual memory page unit wear-leveling

3

Main memory and storage

Efficient exchange and endurance issue

2

File system

i-node issue wear-leveling

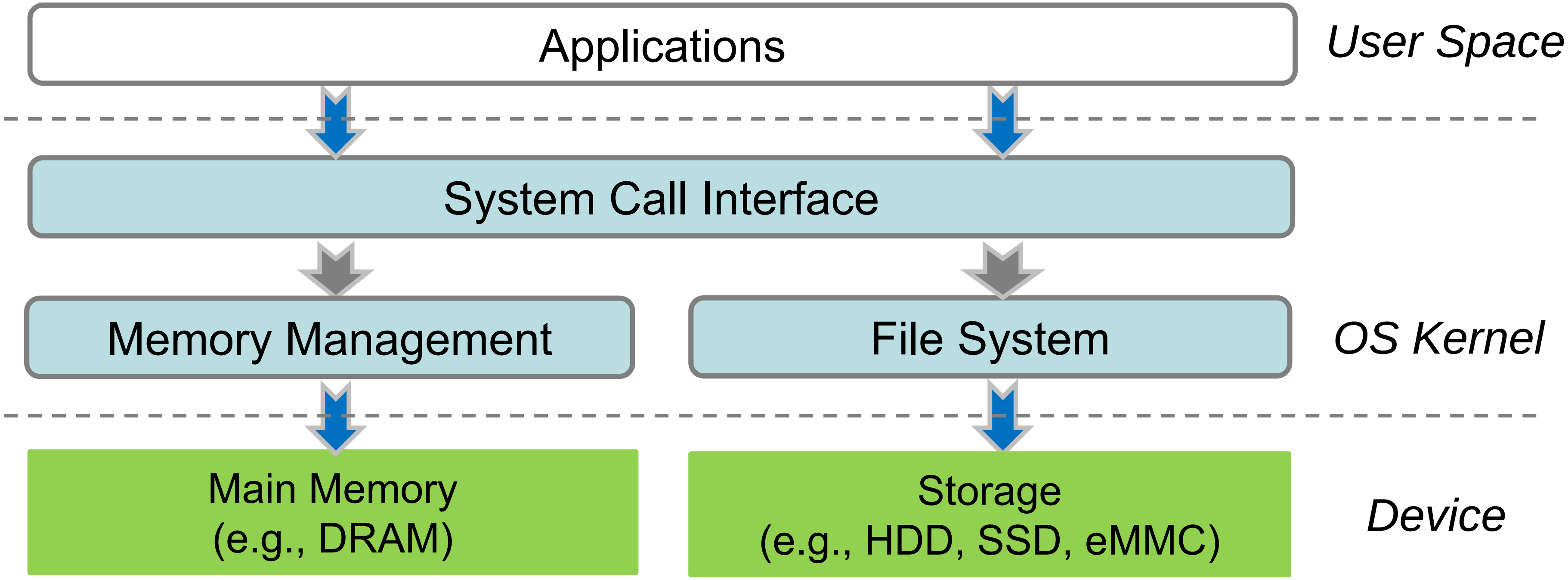
Non-volatile Memory Translation Layer

(main memory + storage)

Non-volatile Memory with NVDIMM interface

(MRAM, PCM, FeRAM, STT RAM, RRAM, or Flash)

Traditional System Architecture



Applications

User Space

System Call Interface

Memory Management

File System

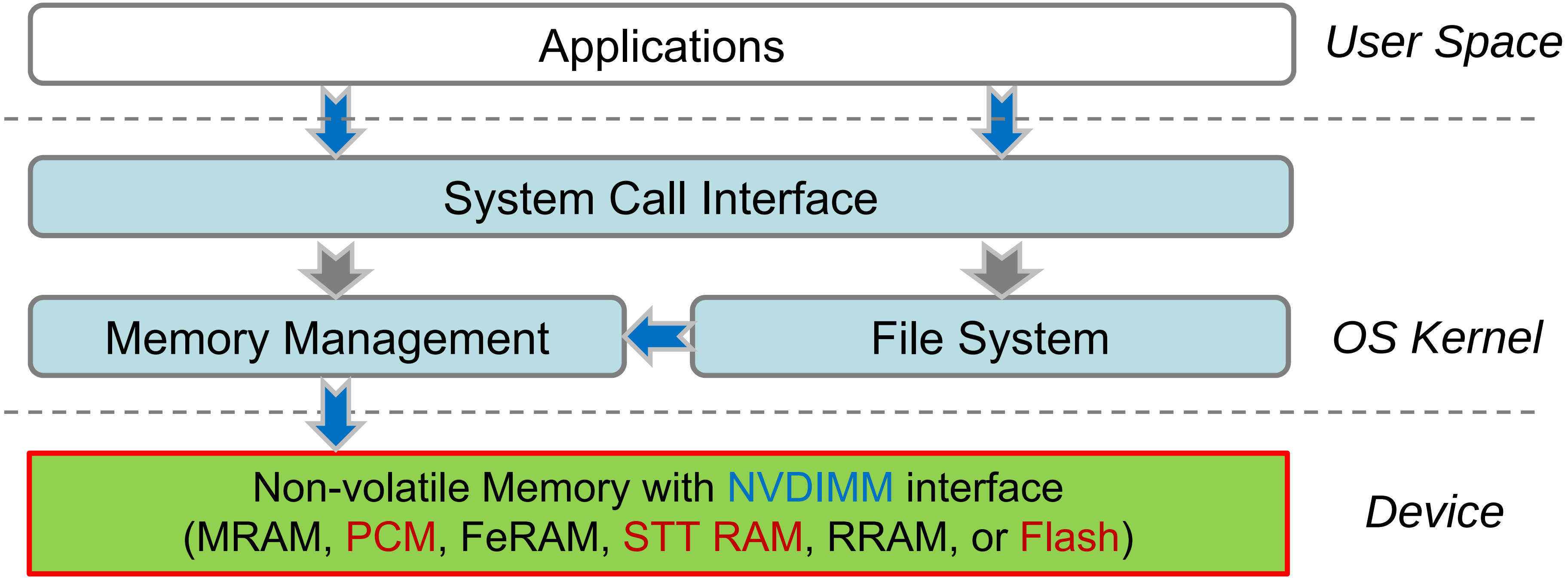
OS Kernel

Main Memory
(e.g., DRAM)

Storage
(e.g., HDD, SSD, eMMC)

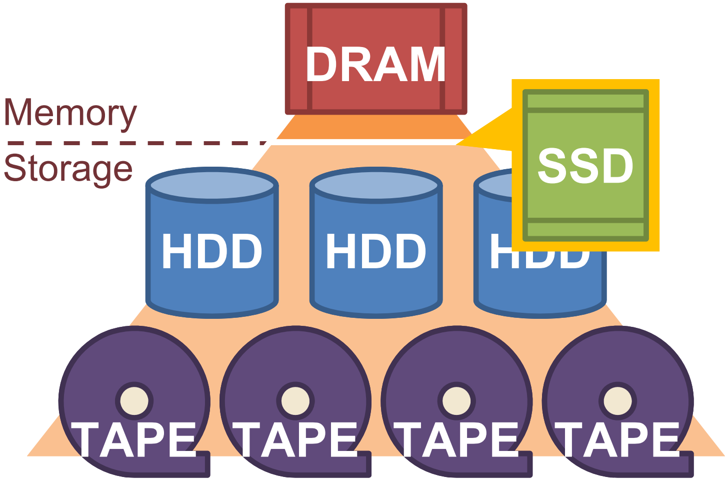
Device

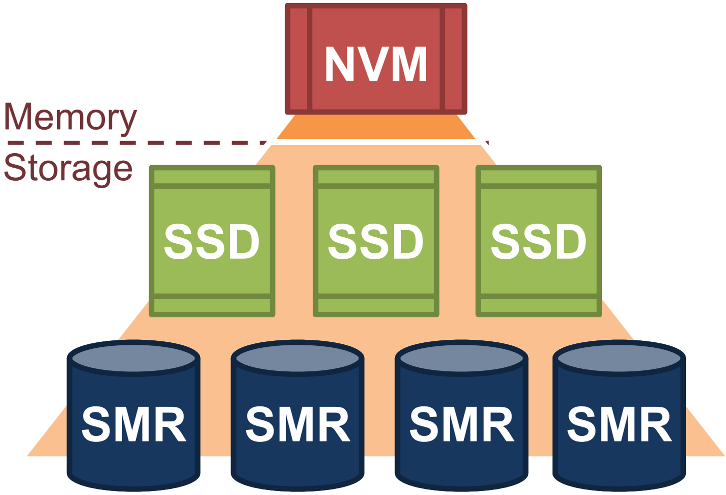
One-Memory System Architecture



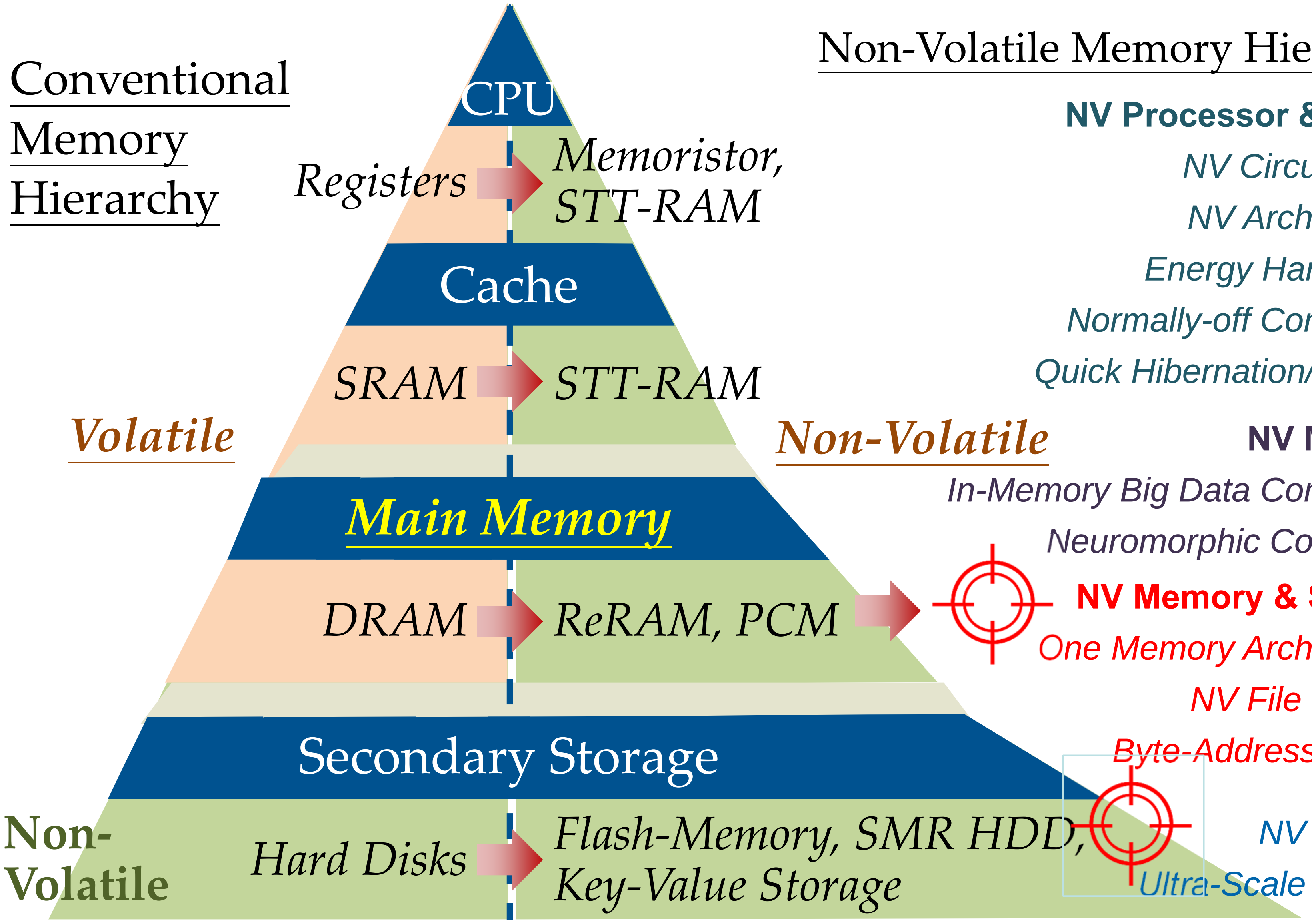
Next-generation Storage Systems

- ***Present***:
  - **DRAM:** Main memory
  - **Solid-state Drive (SSD):** Cache/buffer for high performance environments
  - **Hard Disk (HDD):** Main storage media.
  - **Magnetic tapes:** Deep data archiving
- ***Foreseeable Future***:
  - **Non-volatile memory (NVM):** Enlarge the scalability of in-memory computing
  - **SSDs:** Take over the main storage media
    - Reasons: Density ↑ and cost ↓
  - **HDDs** & **Tapes:** Replaced by new magnetic recording technologies
    - Objectives: Density ↑ and cost ↓
    - Promising Candidate: **SMR**





Non-volatile Computing



Conventional Memory Hierarchy

CPU

Registers → Memoristor, STT-RAM

Cache

SRAM → STT-RAM

Volatile

Non-Volatile

Main Memory

DRAM → ReRAM, PCM

Secondary Storage

Non-Volatile

Hard Disks → Flash-Memory, SMR HDD, Key-Value Storage

Non-Volatile Memory Hierarchy

NV Processor & Cache

NV Circuit/Chip,

NV Architecture,

Energy Harvesting,

Normally-off Computing,

Quick Hibernation/Restore

NV Memory

In-Memory Big Data Computing,

Neuromorphic Computing

NV Memory & Storage

One Memory Architecture,

NV File System,

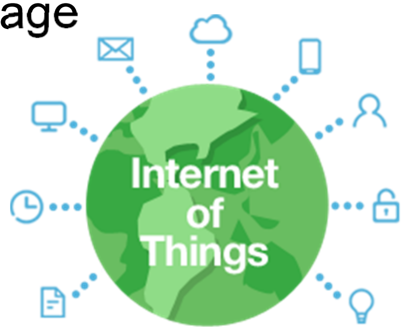
Byte-Addressable FS

NV Storage

Ultra-Scale Storage

Future Research Directions

- Software: **Non-Volatile Operating System**
  - Most researches are based on the existing operating system designs.
    - Memory Space: Managed by the memory management.
    - Storage Space: Managed by the file systems.
  - A *new non-volatile operating system design* is needed by the non-volatile computing environment.
    - The objective is to get rid of unnecessary software stacks (e.g., page cache, swap).
- System: **Non-Volatile SiC** (System in a Chip) with *on-chip storage*
  - SiC: Control units + memory + storage are all in a chip.
  - Process-in-memory (PIM): Computing + Memory + Storage
  - **Self-sustainable Sensor Nodes** (Green IoT)
    - Instant backup & restore capabilities of NV processors.
    - Low standby power of NV memory (& storage).
    - Energy harvesting systems.



- Storage:
  - **Approximate storage**
  - **In-memory approximate computing storage**

Research Directions

- Software: **Non-Volatile System Software**
  - Most researches are based on the existing operating system designs.
    - Memory Space: Managed by the memory management.
    - Storage Space: Managed by the file systems.
  - *New non-volatile operating system and system software are* needed by the non-volatile computing environment.
    - The objective is to get rid of unnecessary software stacks (e.g., page cache, swap).
- System: **Non-Volatile Systems**
  - System in a Chip (SiC): Control units + memory + storage are all in a chip.
  - **Process-in-memory (PIM)**: Computing + Memory + Storage
  - **Self-sustainable Sensor Nodes** (Green IoT)
    - Instant backup & restore capabilities of NV processors.
    - Low standby power of NV memory (& storage).
    - Energy harvesting systems.

- Storage:
  - Approximate storage
  - Smart storage