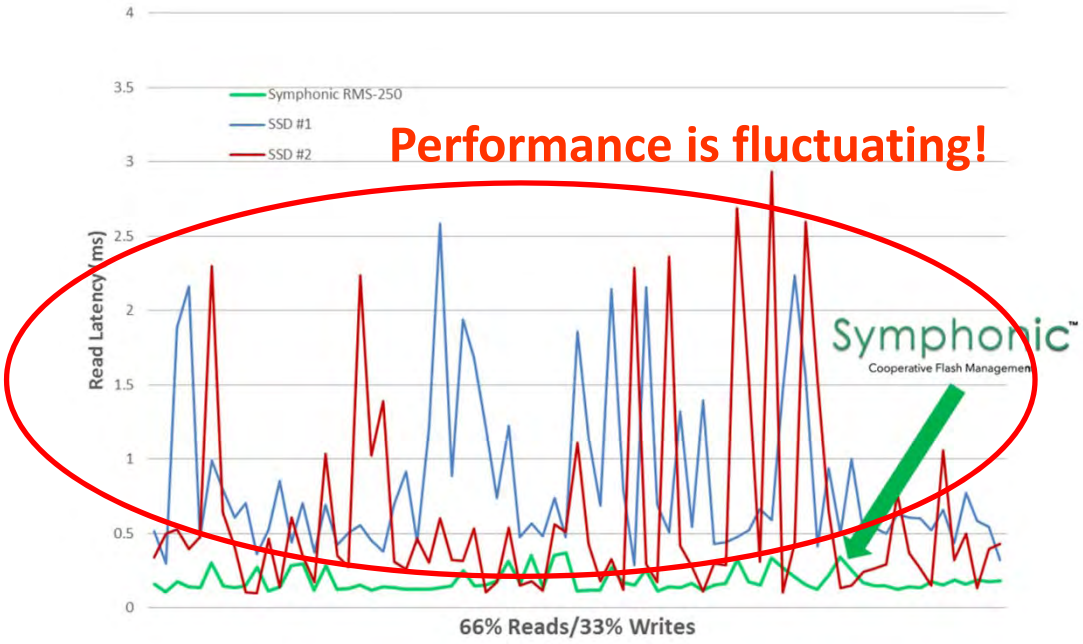
Reducing Write Amplification of Flash Storage through Cooperative Data Management with NVM

32nd International Conference on Massive Storage Systems and Technology (MSST)
May, 2016

Eunji Lee, Chungbuk Natational University
Julie Kim, Ewha University
Hyokyung Bahn, Ewha University
Sam H. Noh, UNIST

Write Amplification in SSD

- Undesirable phenomenon associated with flash memory
- Number of writes to storage is higher than the number of writes issued by a host
- Key aspect limiting **stable performance** and **endurance** of SSD

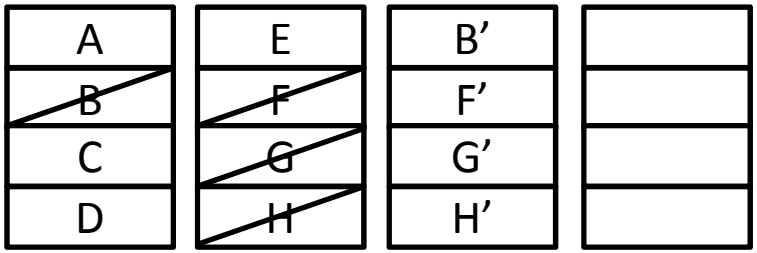


Source: Radian Memory Systems

Write Amplification in SSD

- Garbage collection is performed to recycle used blocks
- Copy out valid pages in a victim block into a free block

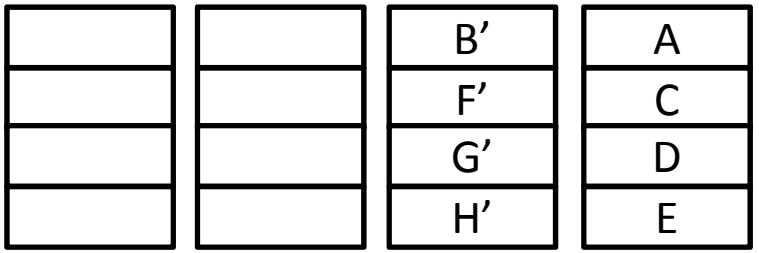
write (B’ F’ G’ H’)



Write 4 Blocks

write (B’ F’ G’ H’)

+ write (A C D E) in GC



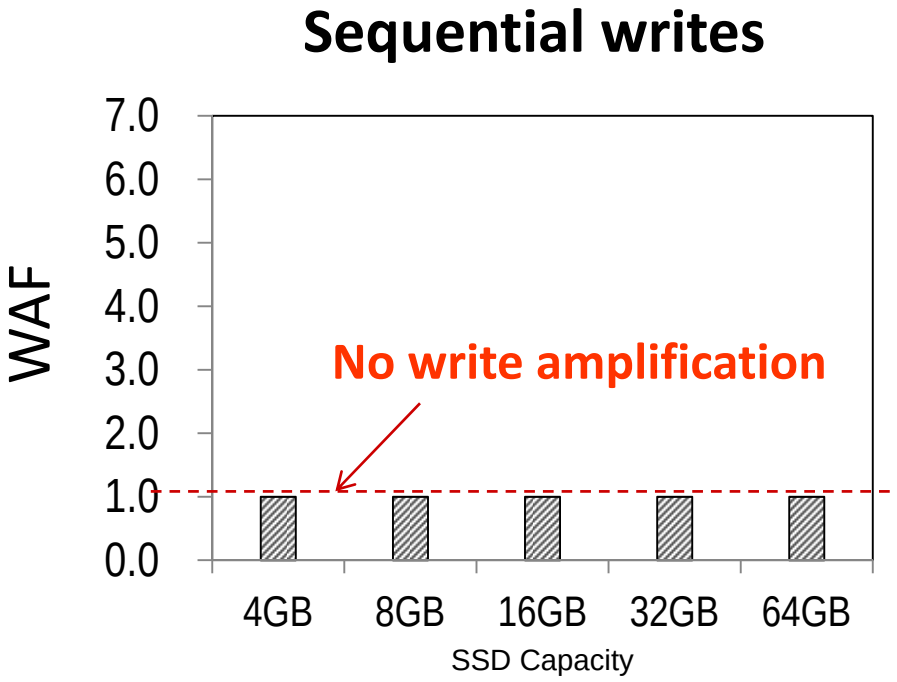
Write 8 Blocks

2x Writes!

Write Amplification Factor : 2.0

Workload and WAF Relationship

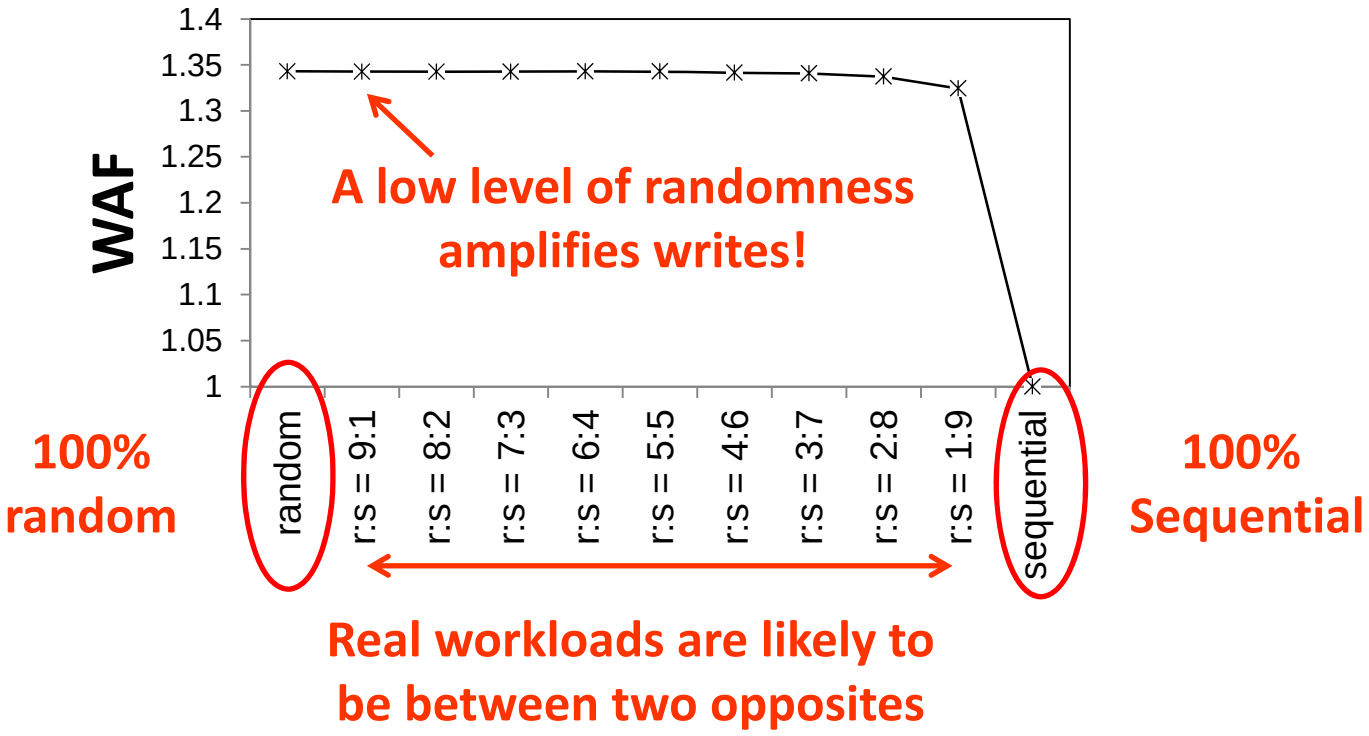
- Analyze WAF with respect to workload characteristics
- Generate two synthetic workloads using SSDsim



Random updates incur the dispersed distribution of the valid pages

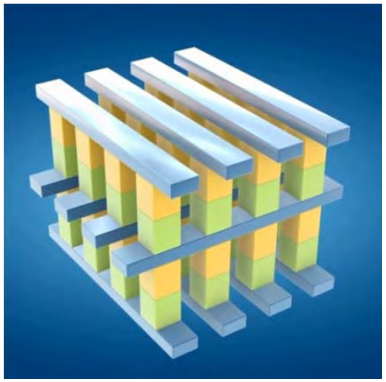
Workload and WAF Relationship

- Real workload is a mixture of random and sequential accesses
- Observe WAF varying the level of randomness



NVM Technology

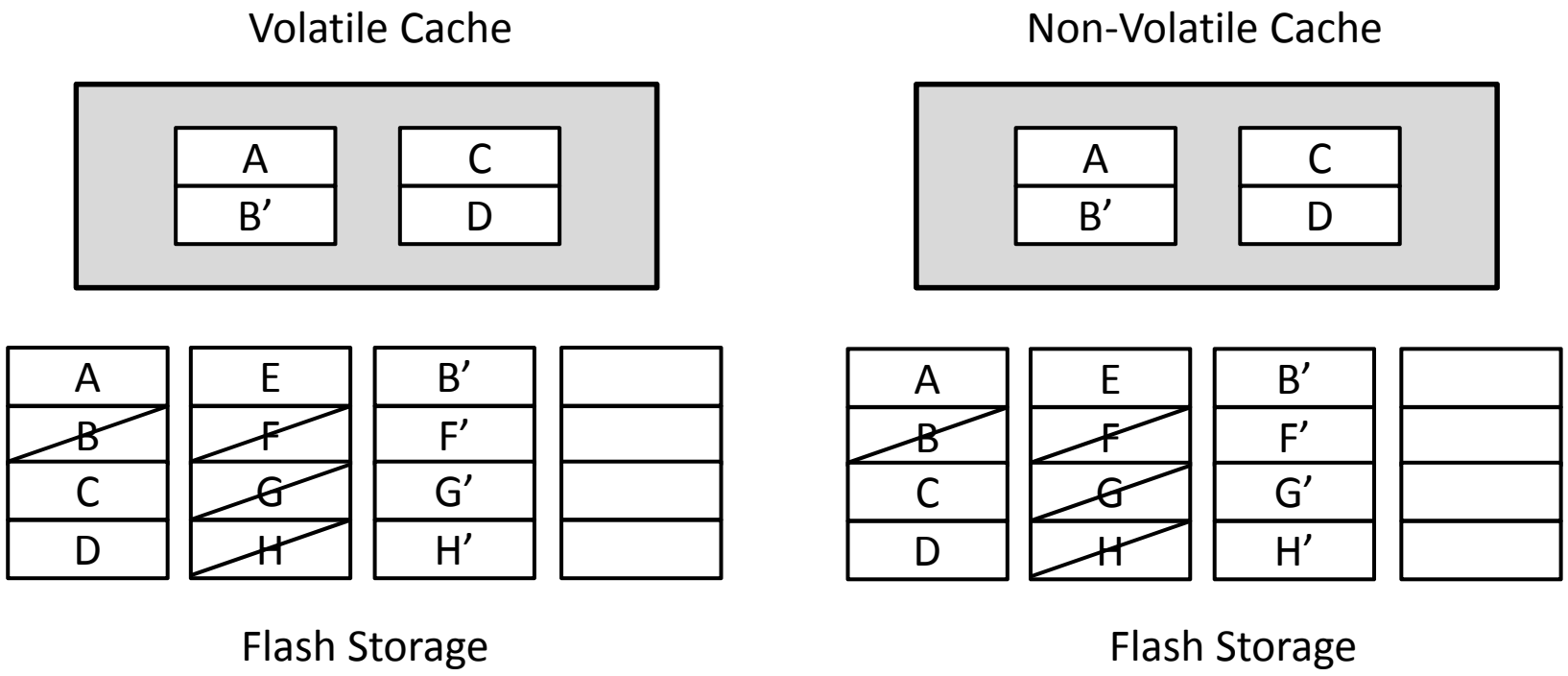
- Becoming increasingly viable as leading semi-conductor manufacturers are eagerly investing in it
  - **Diablo Technologies, *Memory 1***
    - All-flash system memory module
    - 4TB Memory
  - **Intel and Micron, *3DXpoint***
    - All-new memory technology
    - 8x-10x denser than DRAM
    - 1000 times less latency than NAND



- Fast, scalable, and persistent memory is being realized in computer systems

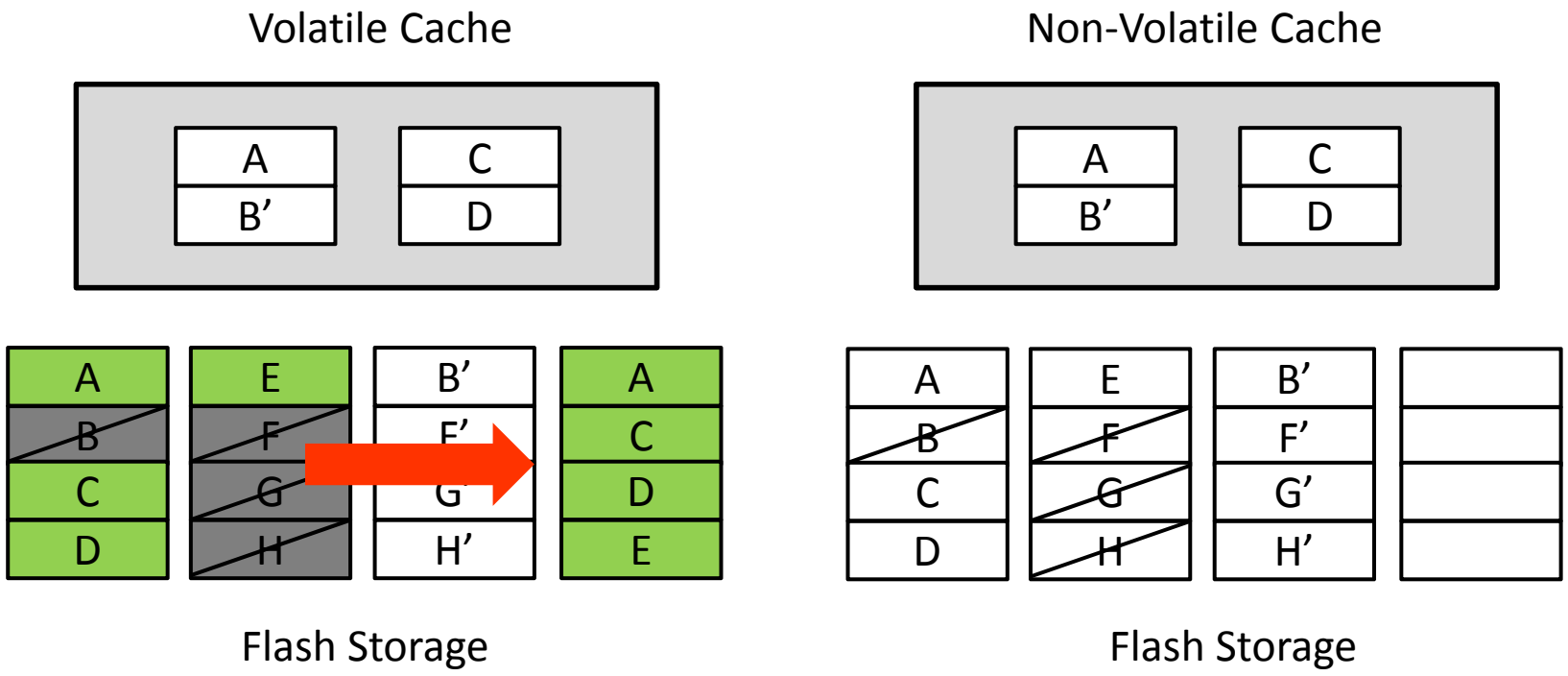
Cooperative Data Management (CDM)

- Goal: Reduce WAF by taking advantage of non-volatility of caches
- Using NVM as a storage cache – promising option



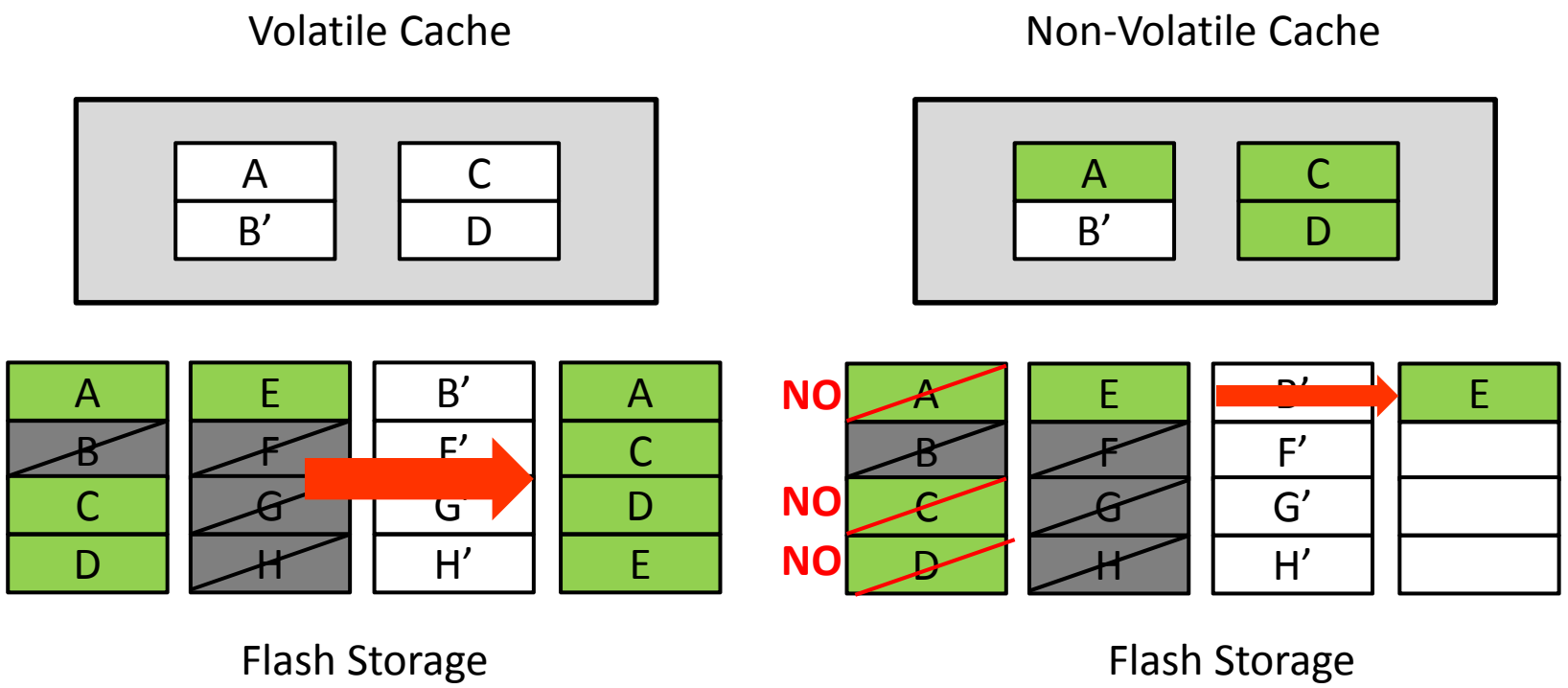
Cooperative Data Management (CDM)

- In traditional systems, all valid pages in a victim block should be copied into a free block during GC
- 4 block writes !



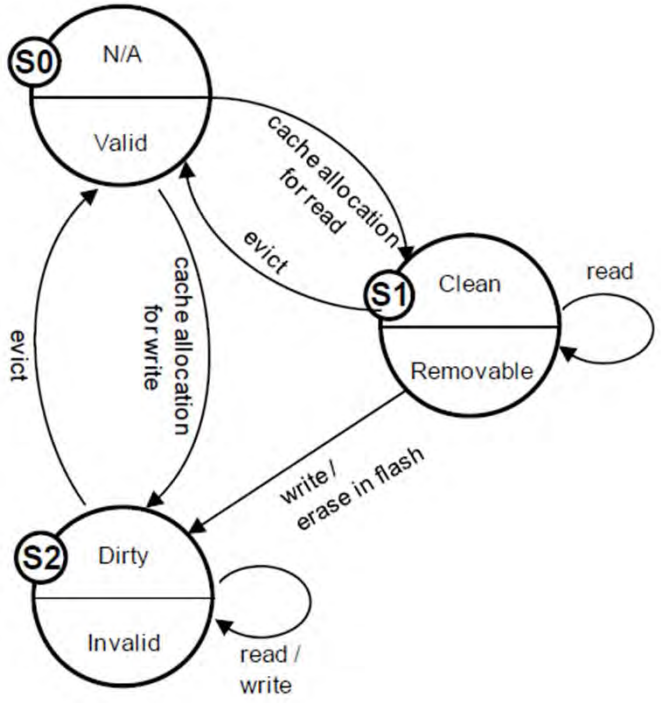
Cooperative Data Management (CDM)

- CDM **skips the copying of valid pages in GC** if the data exist in **non-volatile** cache
- Only one block write!



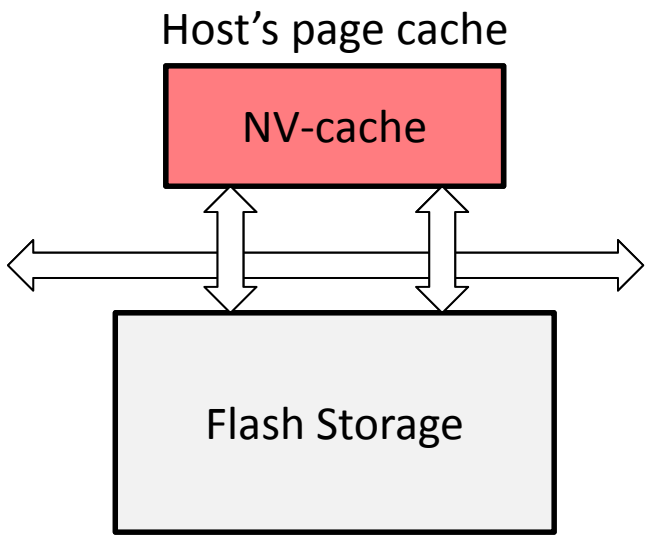
Cooperative Data Management (CDM)

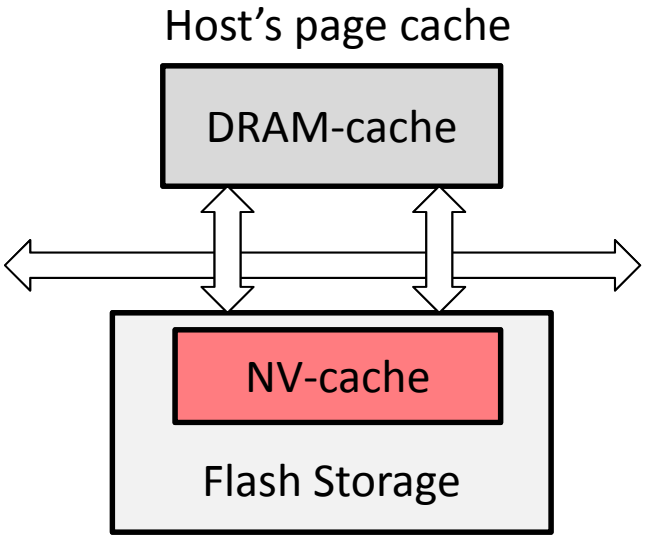
- Finite state diagram
  - "Removable" state
    - Can be erased if the data needs to be copied into another block
    - Or it is same as the valid state
  - Data is set to "Removable" when it is cached in a non-volatile cache



Getting Real: Issues with CDM

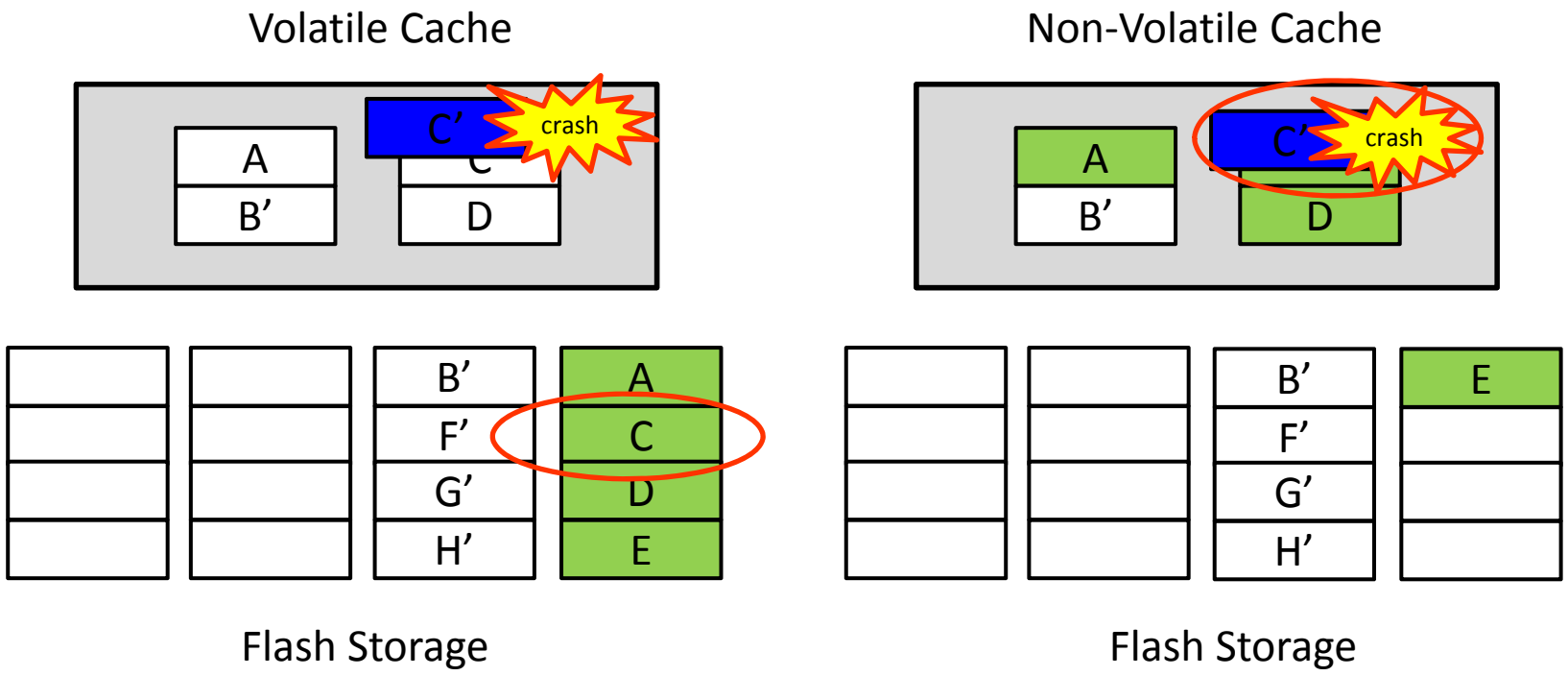
- Implementation of CDM has several issues depending on the architectures
- Feasible architectures
  - NV-cache as a page cache in host
  - NV-cache as an in-storage cache





1. NV-cache as a host cache

- **Issue 1. Consistency**
- Updating data in a cache touches a final copy
- Crash during update results in inconsistent data

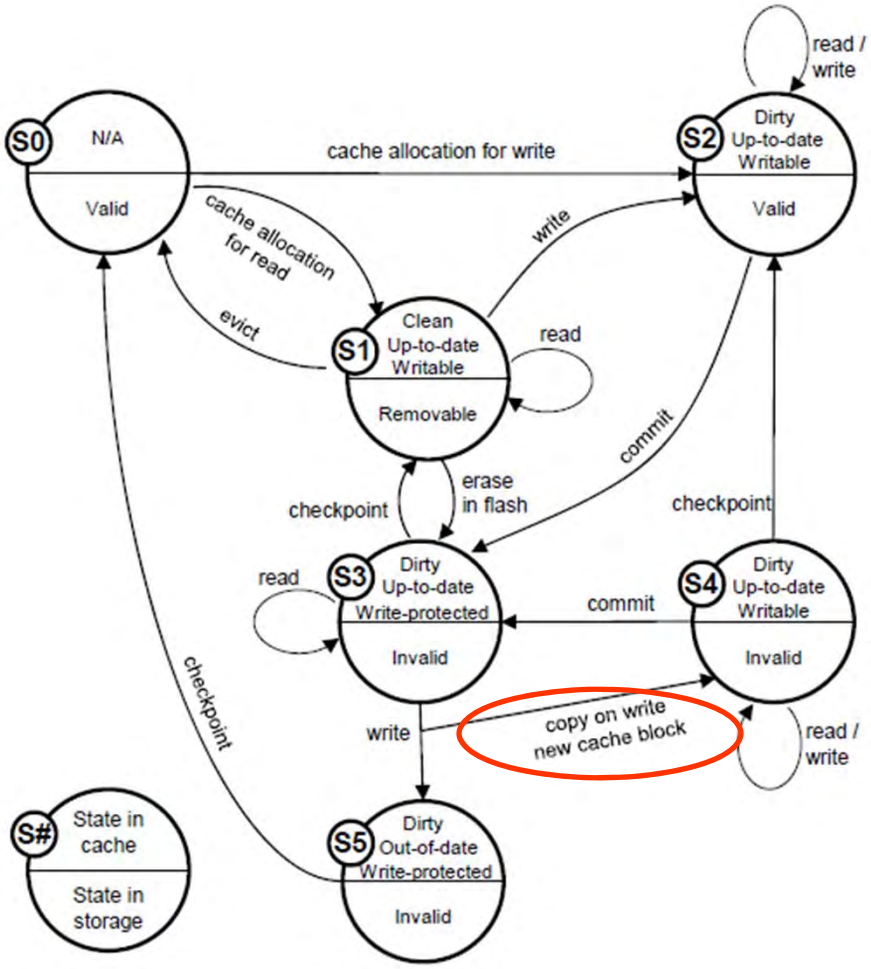


1. NV-cache as a host cache

- **Issue 1. Consistency**
- Solution is associated with specific file system implementations
- Data consistency is managed in a file system layer with different techniques
- File systems should be redesigned considering a way of handling data in CDM

Case Study: Ext4 with CDM

- Finite State Diagram
- Introduce additional states of cached data to determine whether its another copy remains in storage
- Update data with a copy-on-write technique if the cached data serves as a final copy

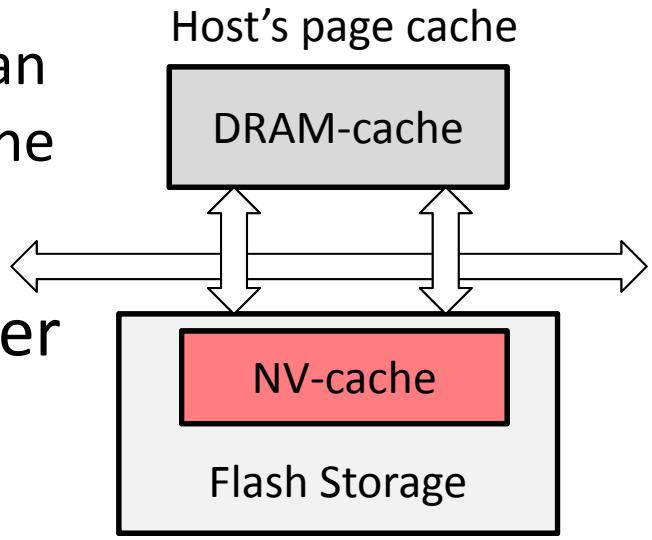


1. NV-cache as a host cache

- **Issue 2. Communication Overhead**
- Events in cache and storage should be notified to each other synchronously
  - e.g., Garbage Collection, Erase of a block, Cached data update, etc.
- Designing a new interface is no longer a big deal
  - Recent packet-based interfaces like NVM-e are easy to piggyback additional information on the original data
- However, frequent communication for additional information transfer can be a burden
  - Consider finding a way to relieve the overhead as a future work

2. NV-cache as an in-storage cache

- More deployable architecture
- No consistency issue
  - File systems assume that the data in storage can become inconsistent if a system crashes during updates
- Not serious communication overhead
  - Sharing information inside a storage device is much cheaper and easier than synchronizing storage with a host cache
- Development can be achieved by a single party - storage manufacturer
  - No need to change file systems



Performance Evaluation

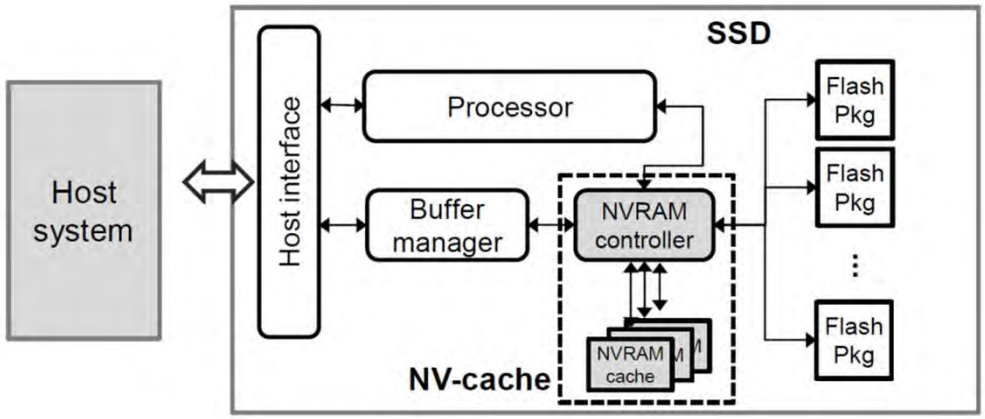
- Trace-driven simulation with SSDsim
  - Developed by MSR as an extension of Disksim
  - Emulate SLC NAND Flash

Table 2: Experimental parameters.

SSD capacity	64GB
Page size	4KB
Block size	256KB (64 pages)
Page read latency	25us
Page write latency	200us
Block erase latency	1.5ms
Data transfer latency	100us (for 4KB page)
Overprovisioning ratio	15%

Performance Evaluation

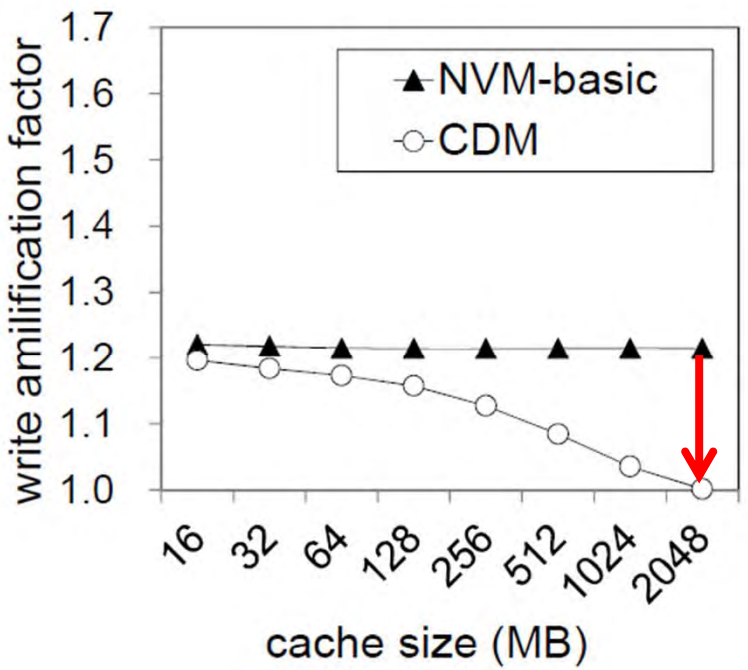
- Implement the in-storage NV-cache module and modify a storage controller to support CDM



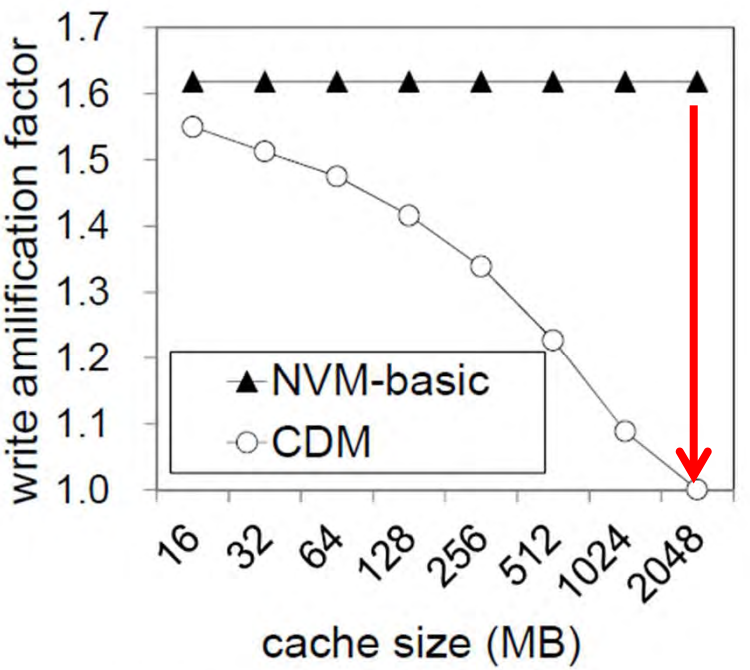
- Compare with a NVM-basic model
  - Manage NVM like a volatile cache
  - Cache data on access and evict it with LRU policy

Write Amplification Factor

- CDM reduces WAF by 2.1-17.6% and 4.3-38.2% in JEDEC and OLTP workloads



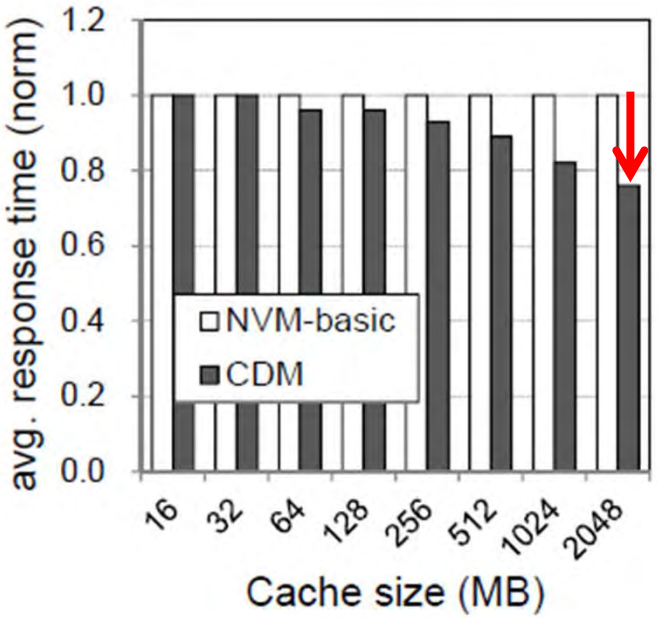
(a) JEDEC



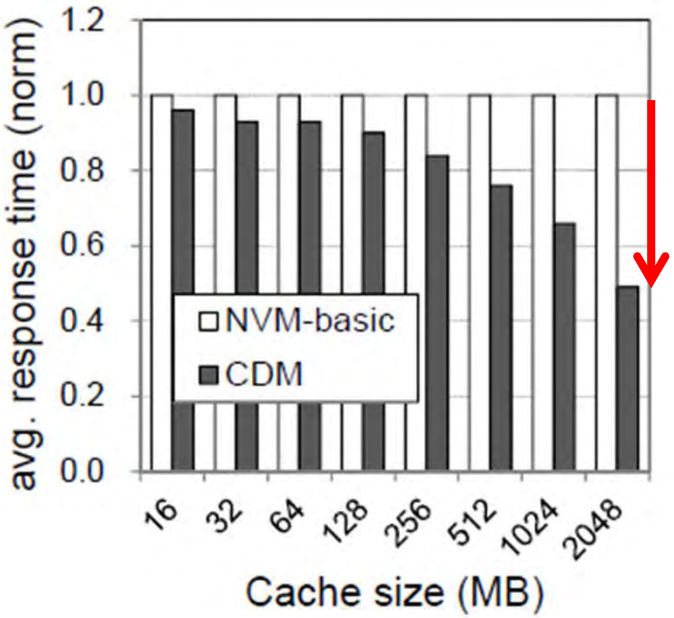
(b) OLTP

Response Time

- Average response time is improved by 9.7% and 20.3% on average in JEDEC and OLTP



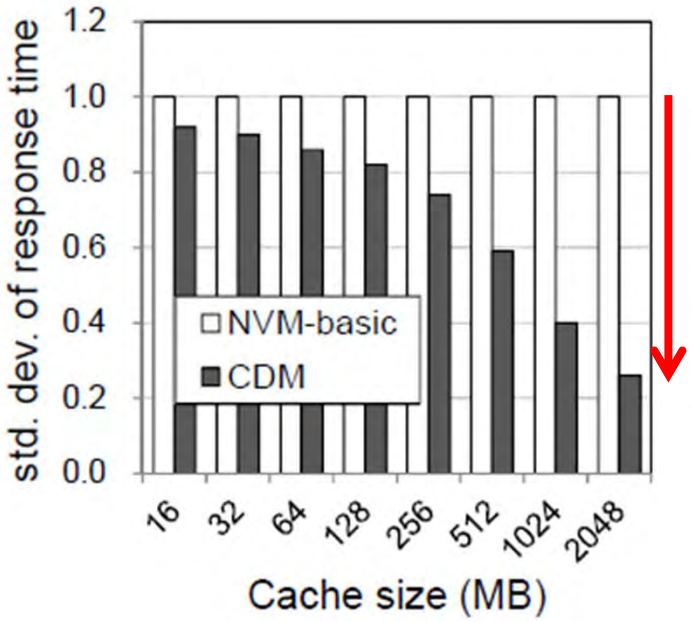
(a) JEDEC



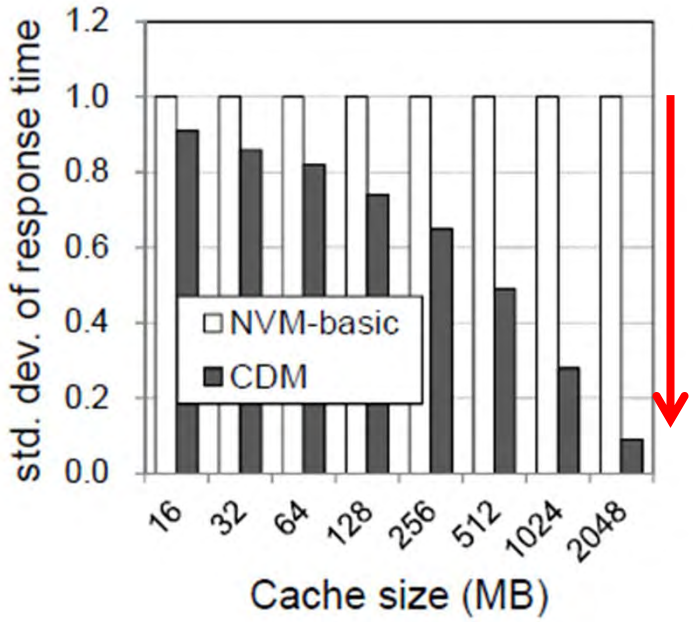
(b) OLTP

Standard Deviation of Response Time

- Reduce standard deviation of response time by 31% and 39% on average in JEDEC and OLTP
- Relieve the performance fluctuating



(a) JEDEC



(b) OLTP

Thank you!

- eunji@cbnu.ac.kr
- http://oslab.cbnu.ac.kr