Performance Trade-Offs in Using NVRAM Write Buffer for Flash Memory-Based Storage Devices

Sooyong Kang, Sungmin Park,
Hoyoung Jung, Hyoki Shim, and
Jaehyuk Cha

IEEE TRANSACTIONS ON COMPUTERS, VOL. 58, NO. 6, JUNE 2009

Outline

- Introduction
- Backgrounds
- NVRAM write buffer management polices
- Performance evaluation
- Optimistic FTL
- Conclusion

Introduction(1/3)

- Flash memory has a few drawbacks, such as the asymmetric speed of read and write operations, inability to in-place updates, very slow erasure operation, among others.

- One of the approaches to achieve the objective is by exploiting the buffer cache in volatile memory to delay write operations.

Characteristics of Storage Media

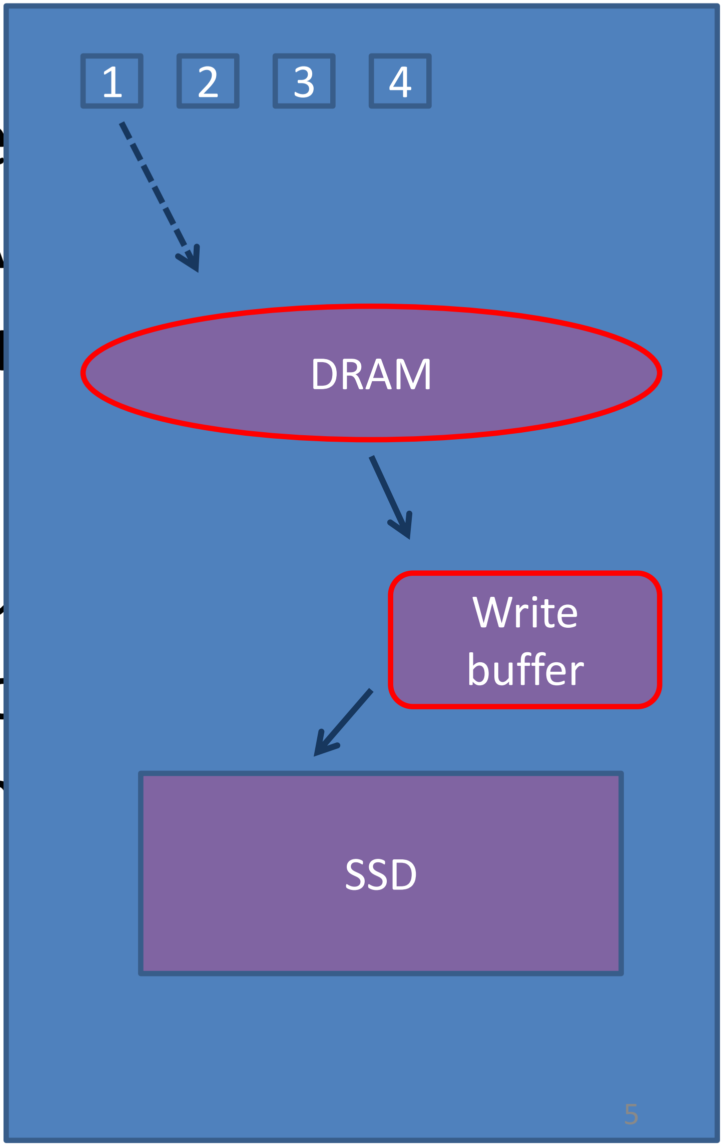
Media	Access time		
	Read	Write	Erase
MRAM	10∼50ns (2B)	10∼50ns (2B)	N/A
FeRAM	30∼100ns (2B)	30∼100ns (2B)	N/A
PRAM	20∼80ns (2B)	20∼80ns (2B)	N/A
DRAM	20ns (2B)	20ns (2B)	N/A
NAND Flash	25μs (2KB)	200μs (2KB)	1.5ms

- ory (NVRAM)
  ice has long
- For the past decade, **next-generation nonvolatile memory** has been under active development.

Introduction(3/3)

- In this paper, we suggest the
sized, next-generation NVRA
to improve the overall perfo

- We propose a novel write bu
translation layer algorithm, c
which is designed to harmon
NVRAM write buffers.



Backgrounds(1/4)

- Characteristics of the NAND Flash Memory
  - Three basic operations
  - Few drawbacks ：
    1. Asymmetric operations speed
    2. Inability to in-place update
    3. Limited lifetime
    4. Random page write prohibition within a block

Backgrounds(2/4)

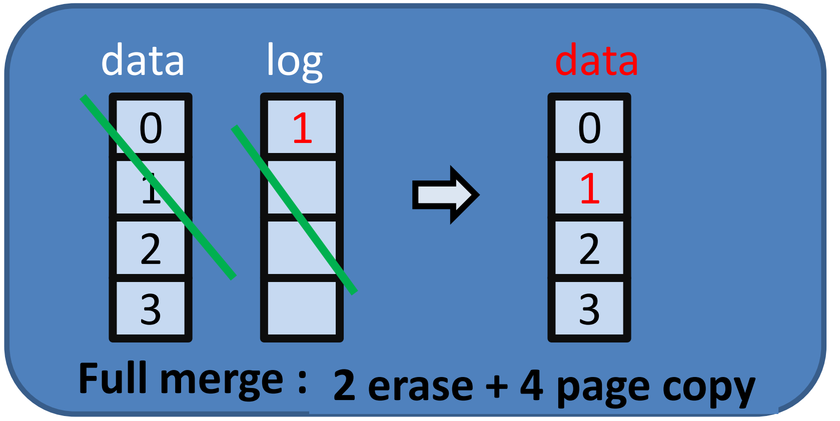
- Flash Translation Layer (FTL)
  - It provides a few core functionalities such as address mapping, bad block management, and ECC check.
  - The overall performance of the flash memory-based storage system highly depends on the mapping scheme.

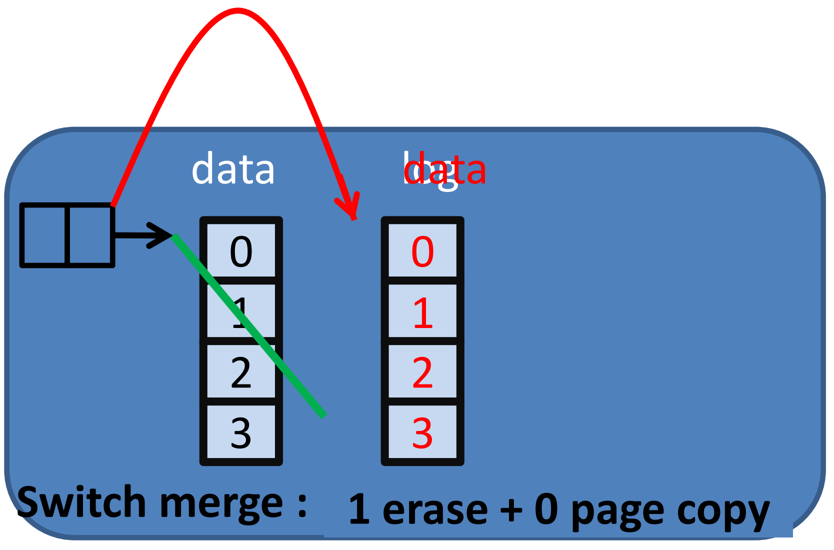
Backgrounds(3/4)

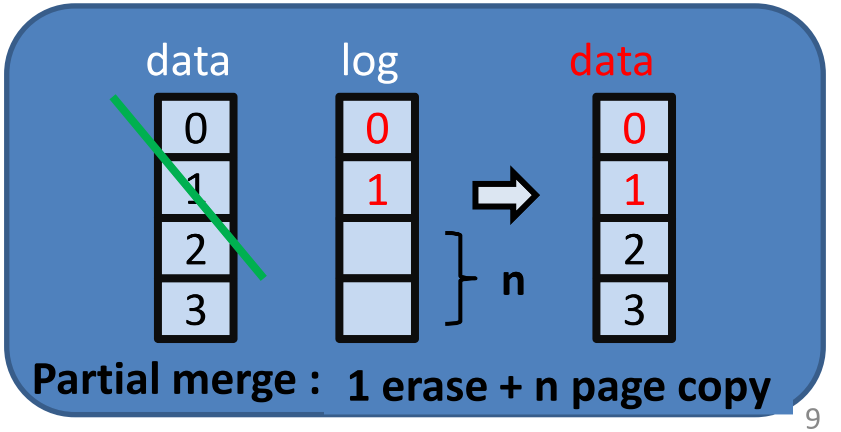
- Log Block-Based Address Mapping
  - There are two representative log block schemes: BAST and FAST.
  - When there is no available log block, they select a victim log block and merge it with its corresponding data block(s)—a **merge** operation.

Backgrounds(4/4)

- Three different forms of merge operations :
  - Switch merge
  - Partial merge
  - Full merge



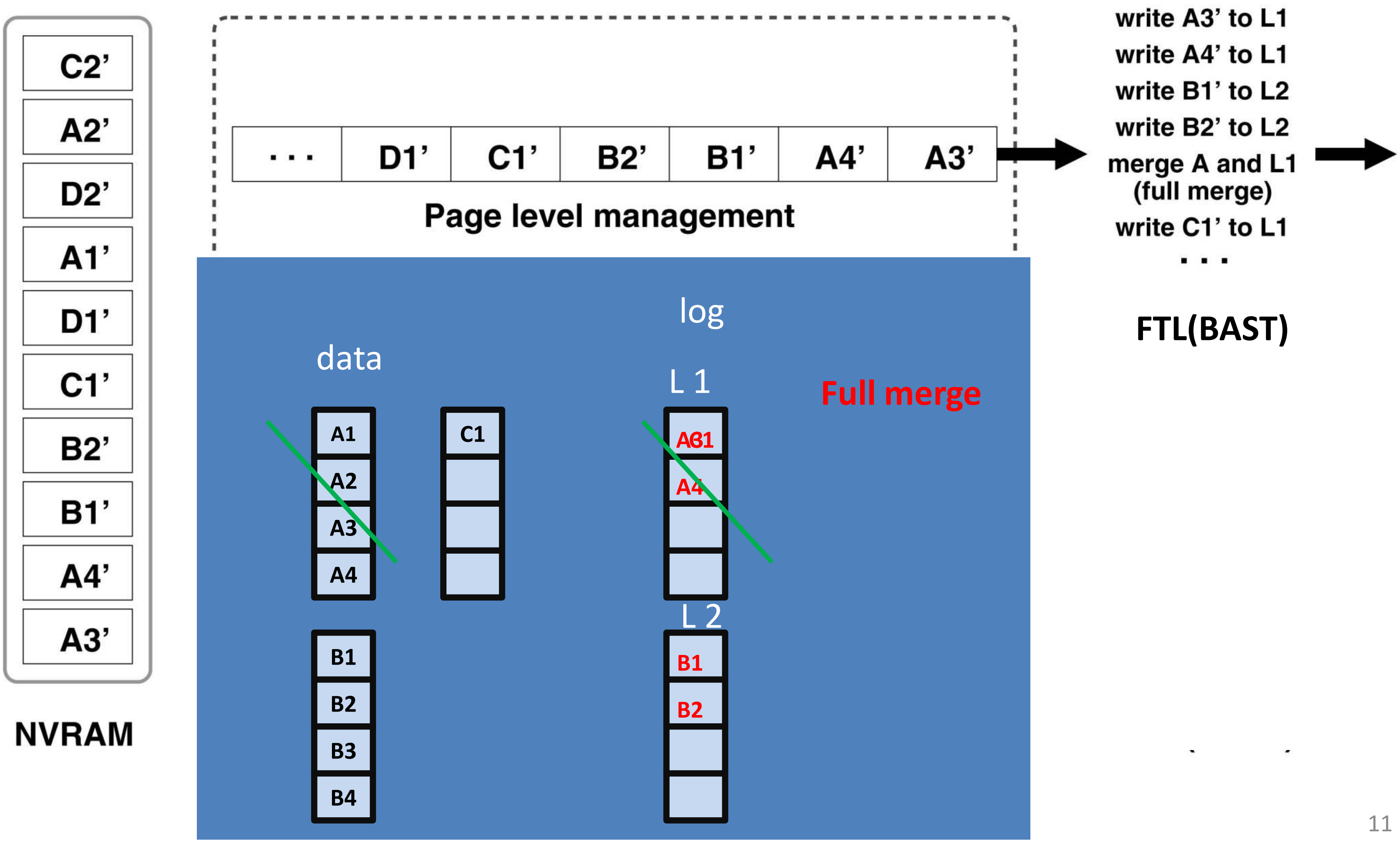


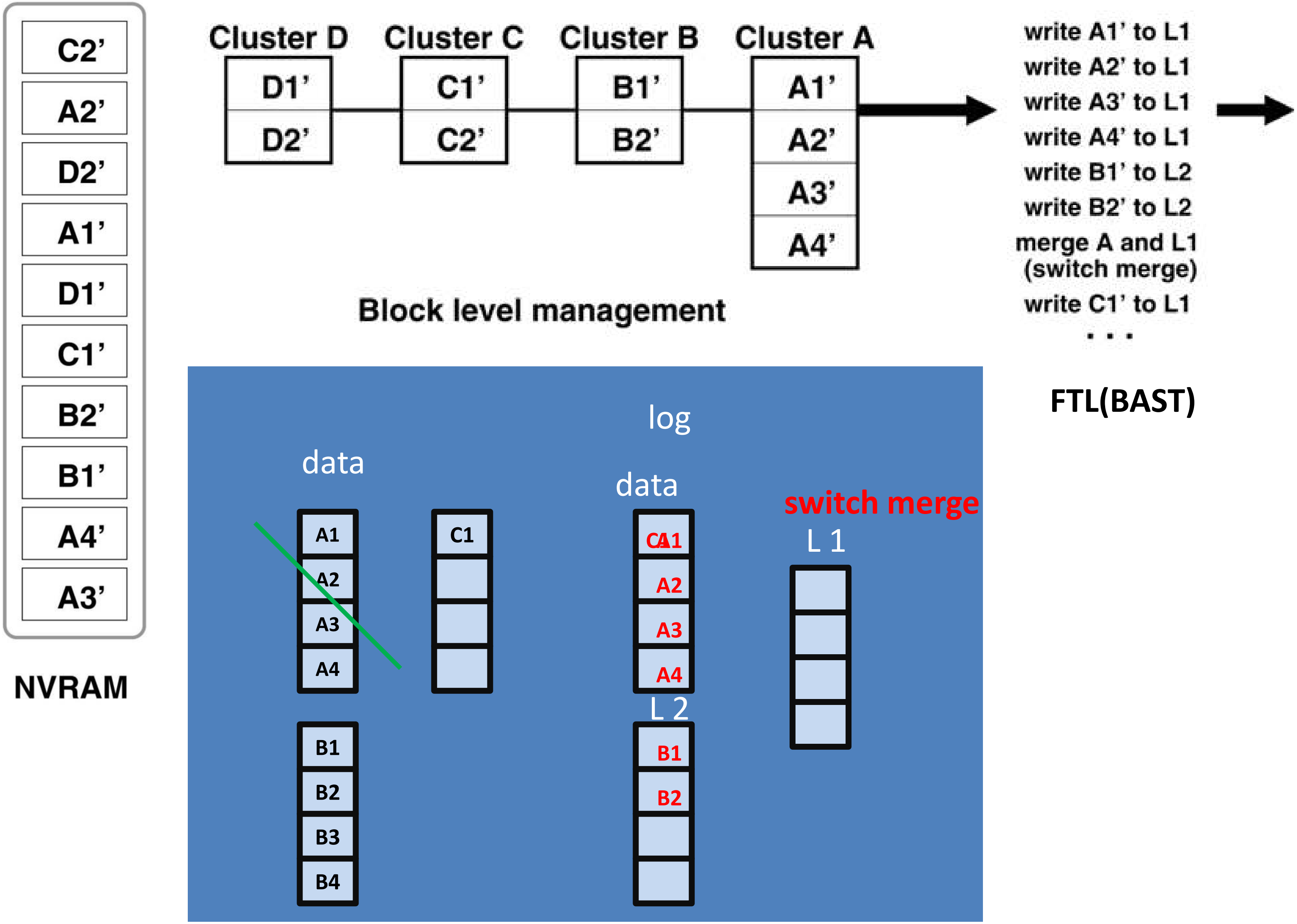


NVRAM write buffer management polices for flash memory

- Least Recently Used Page (LRU-P) Policy (page-level management)
- Least Recently Used Cluster (LRU-C) Policy
- Largest Cluster (LC) Policy
- Cold and Largest Cluster (CLC) Policy

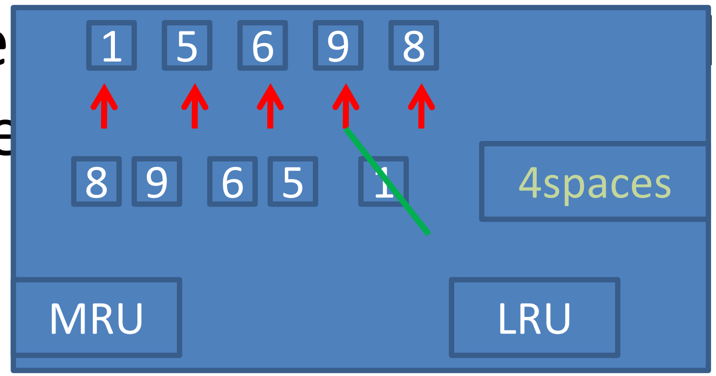
NVRAM write buffer management polices for flash memory




C2'
A2'
D2'
A1'
D1'
C1'
B2'
B1'
A4'
A3'
NVRAM
Cluster D
D1'
D2'
Cluster C
C1'
C2'
Cluster B
B1'
B2'
Cluster A
A1'
A2'
A3'
A4'
Block level management
write A1' to L1
write A2' to L1
write A3' to L1
write A4' to L1
write B1' to L2
write B2' to L2
merge A and L1
(switch merge)
write C1' to L1
. . .
FTL(BAST)
log
data
A1
A2
A3
A4
B1
B2
B3
B4
C1
data
CA1
A2
A3
A4
L 2
B1
B2
switch merge
L 1

NVRAM write buffer management polices for flash memory

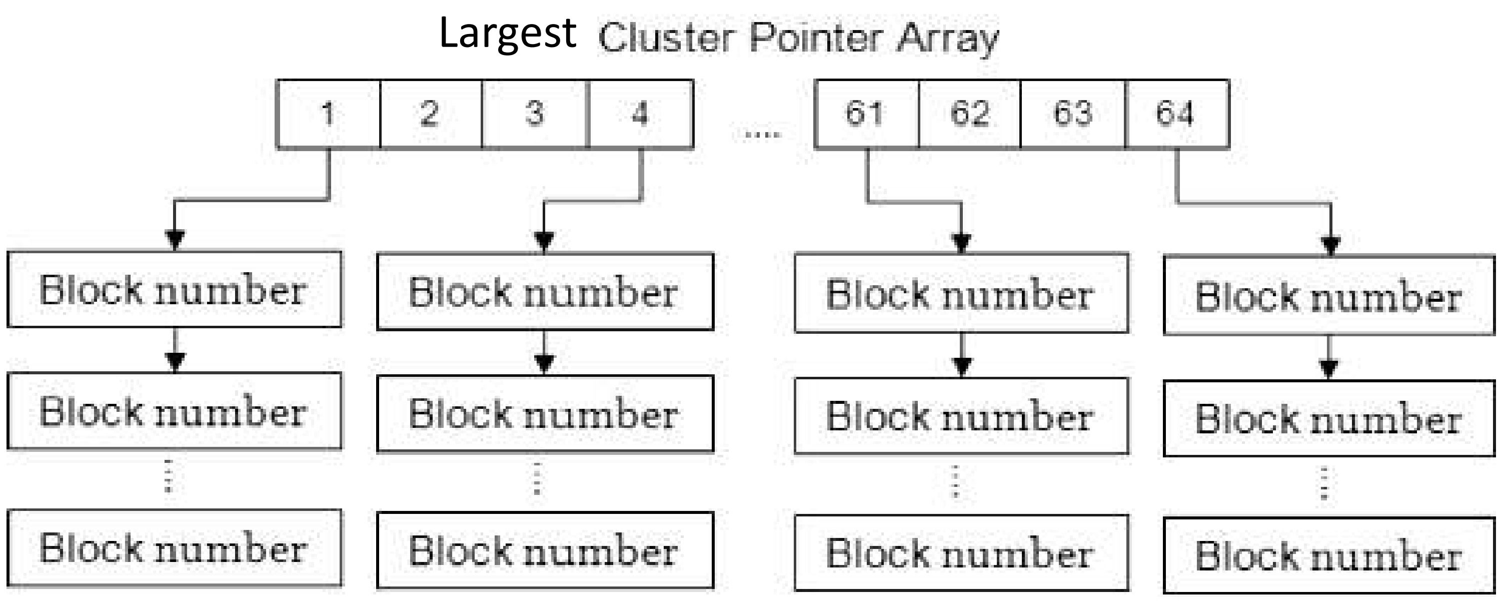
- Least Recently Used Page (LRU-P) Policy
  - The replace the least recently used (writte selected as a victim.



- Least Recently Used Cluster (LRU-C) Policy
  - The replace ster and the least recently acc d as a victim.

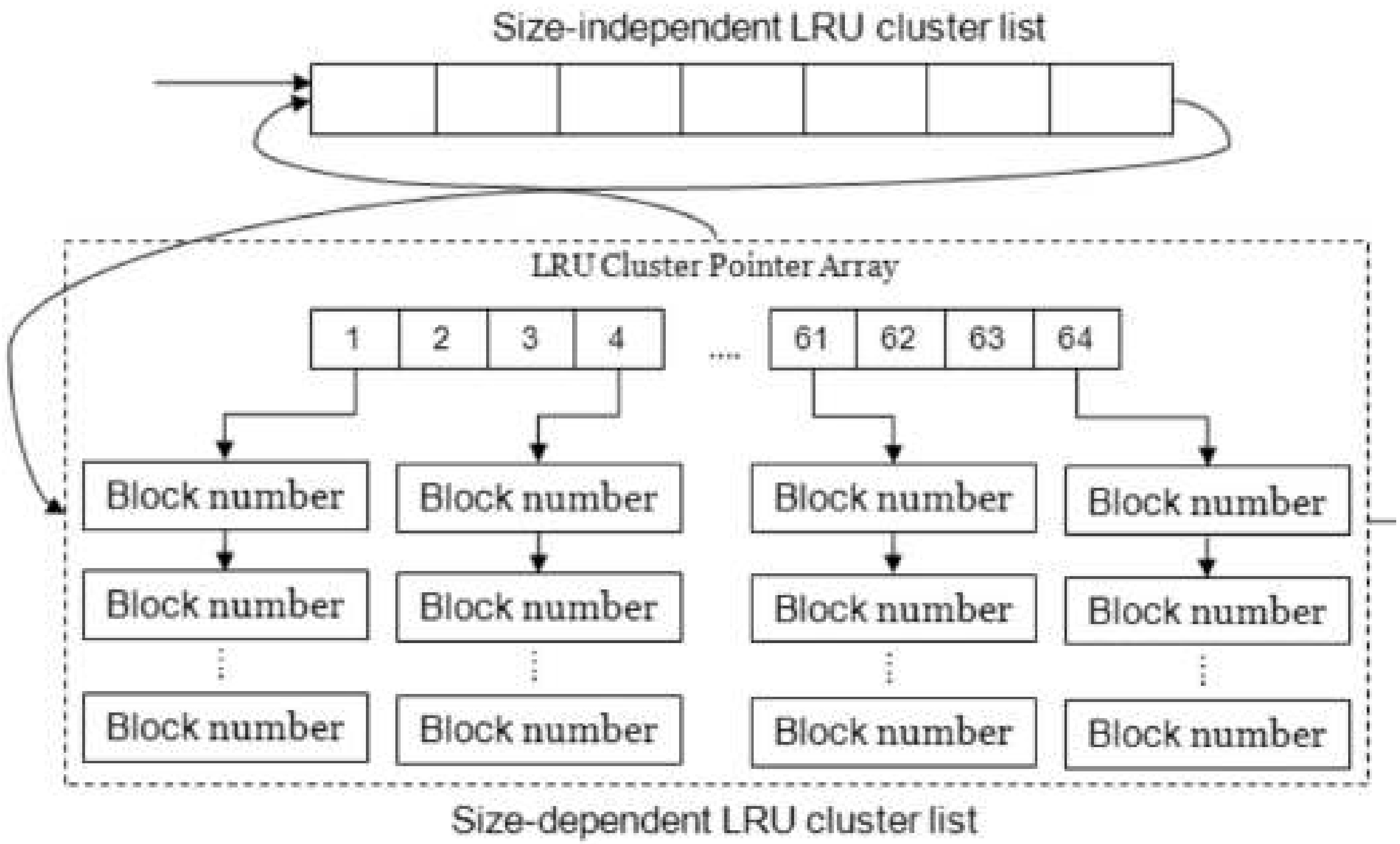
NVRAM write buffer management polices for flash memory

- Largest Cluster (LC) Policy
  - The replacement unit is a page cluster and the page cluster with the largest cluster size is selected as a victim.



NVRAM write buffer management polices for flash memory

- Cold and Largest Cluster (CLC) Policy

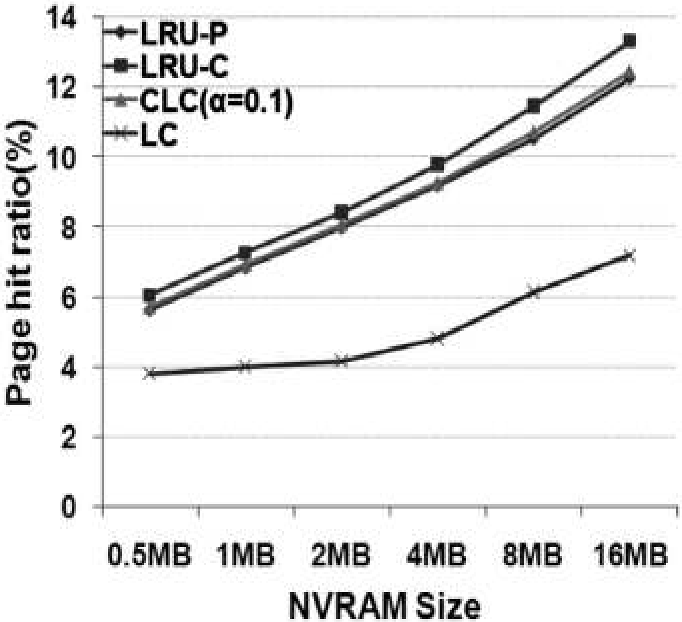


Evaluation

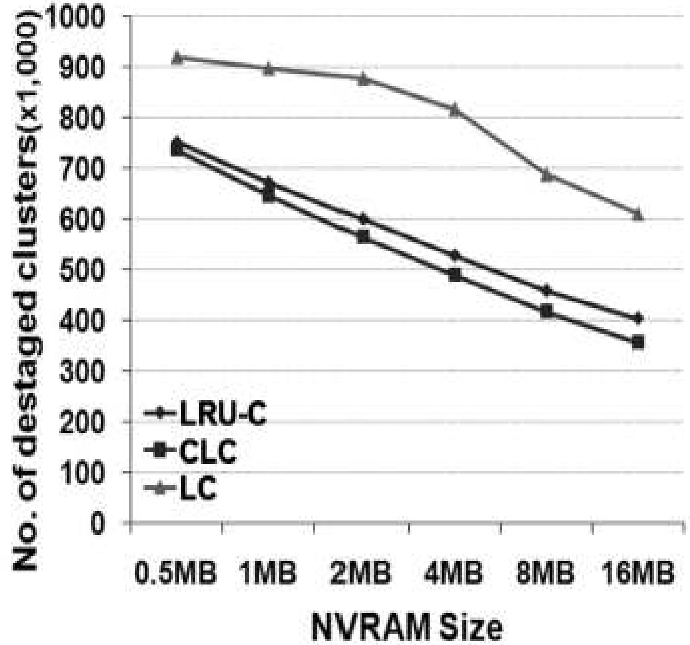
Characteristics of Disk I/O Trace

File system	FAT32, NTFS
Running application	Web surfing, sending/receiving e-mail, movie playback and downloading, document typesetting, and gaming
Sector size	512 bytes
Duration	One month
Final capacity utilization	FAT: 71% (initially empty), NTFS: N/A
Total data written	FAT: 18,284 MB, NTFS: 118,550 MB
Mean write size	FAT: 5.7 sectors, NTFS: N/A
Write locality	FAT: 65-32, NTFS: 70-17 (A-B: A% of total requests access B% of total LBA's)

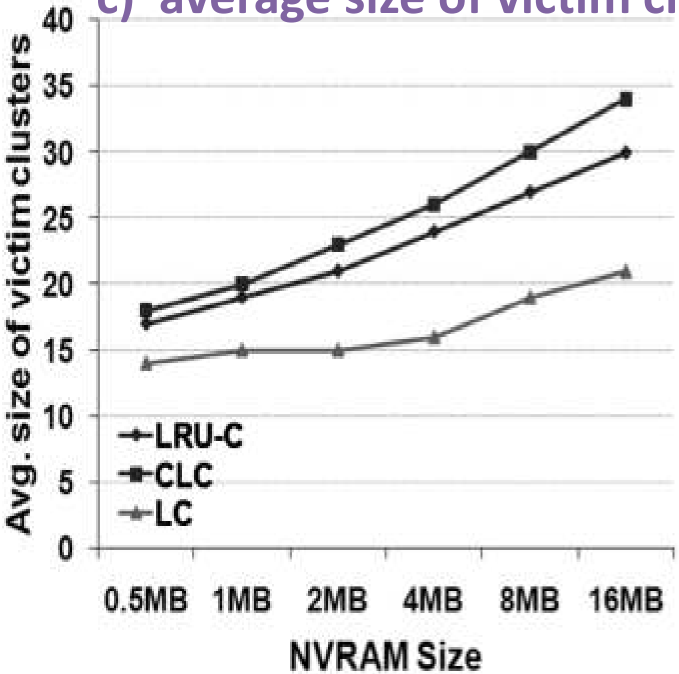
a) Page hit ratio



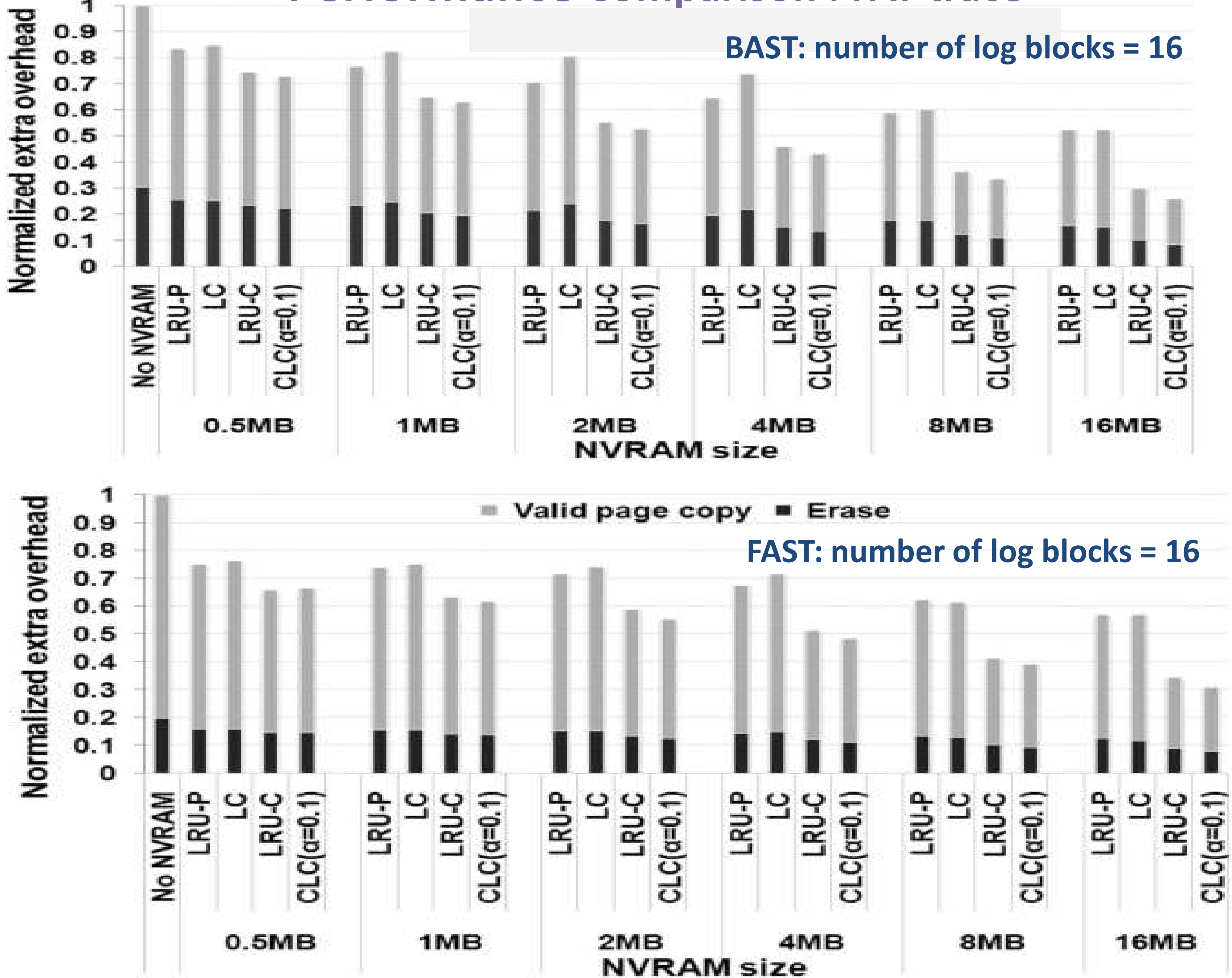
b) number of destaged clusters



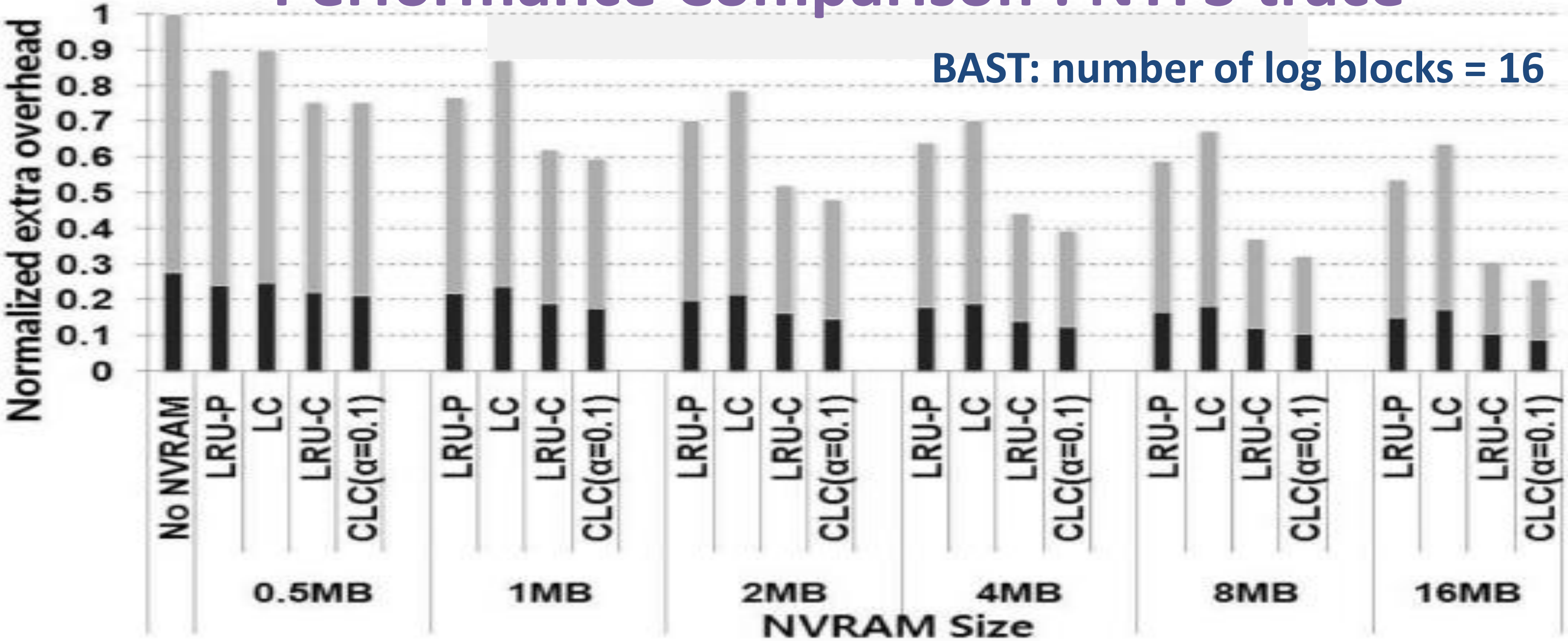
c) average size of victim clusters

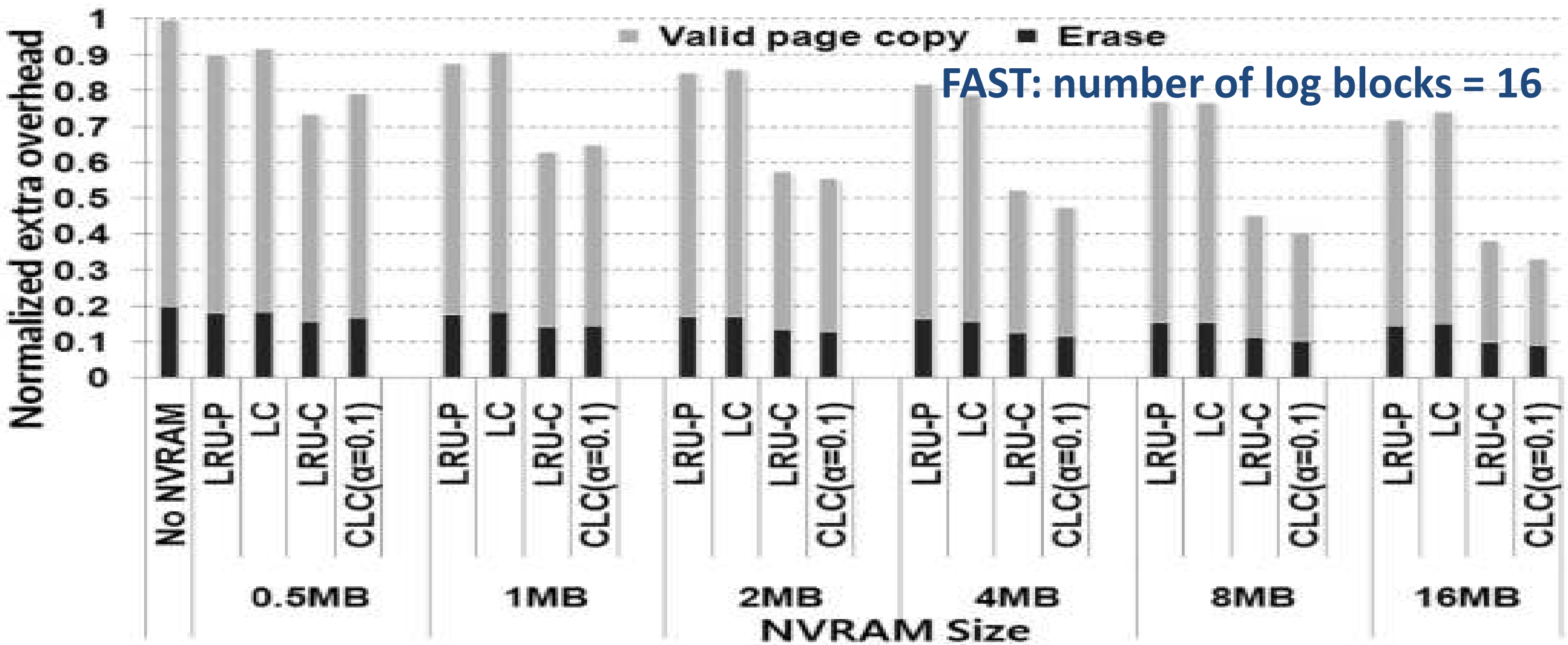


Performance Comparison : FAT trace



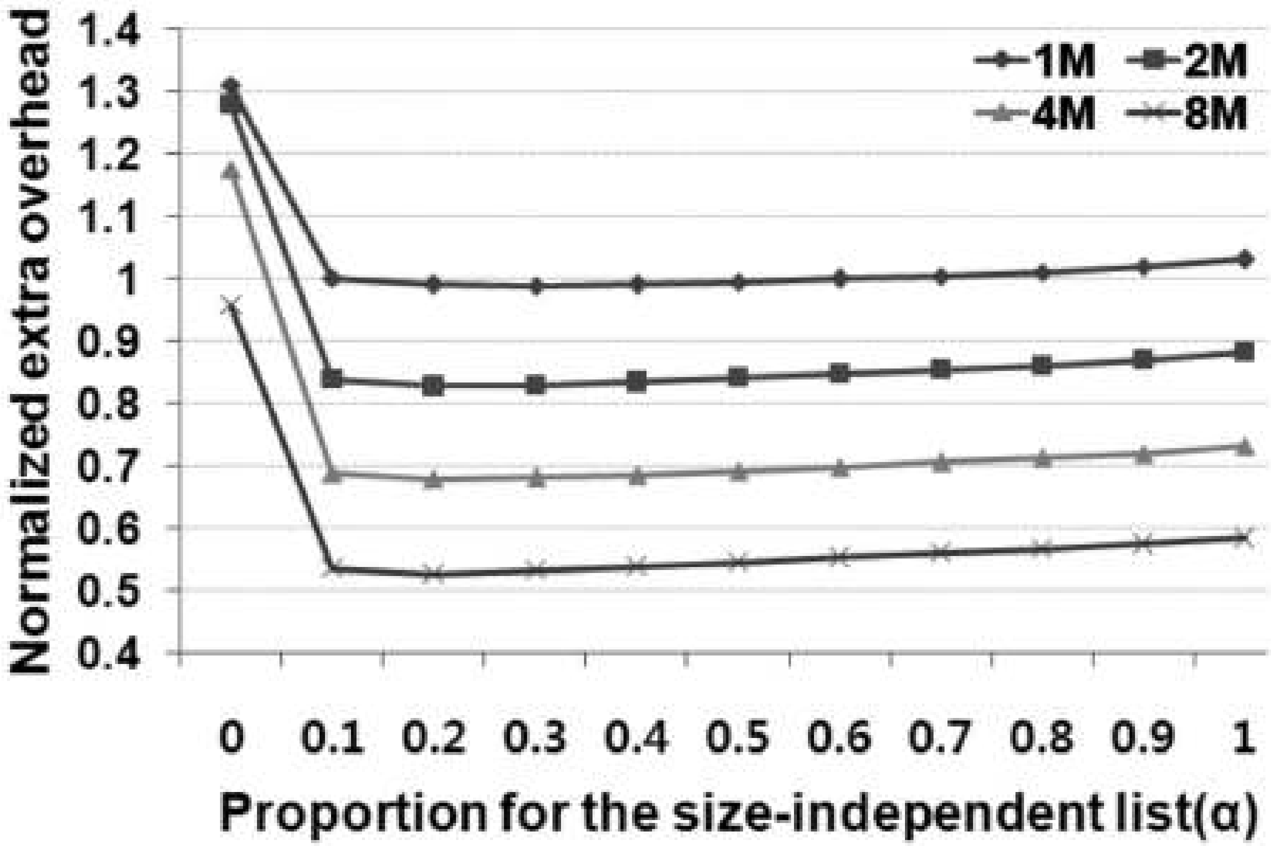
Performance Comparison : NTFS trace





Performance of CLC policy

CLC = α *LRUC +(1-α)LC



α =0.1

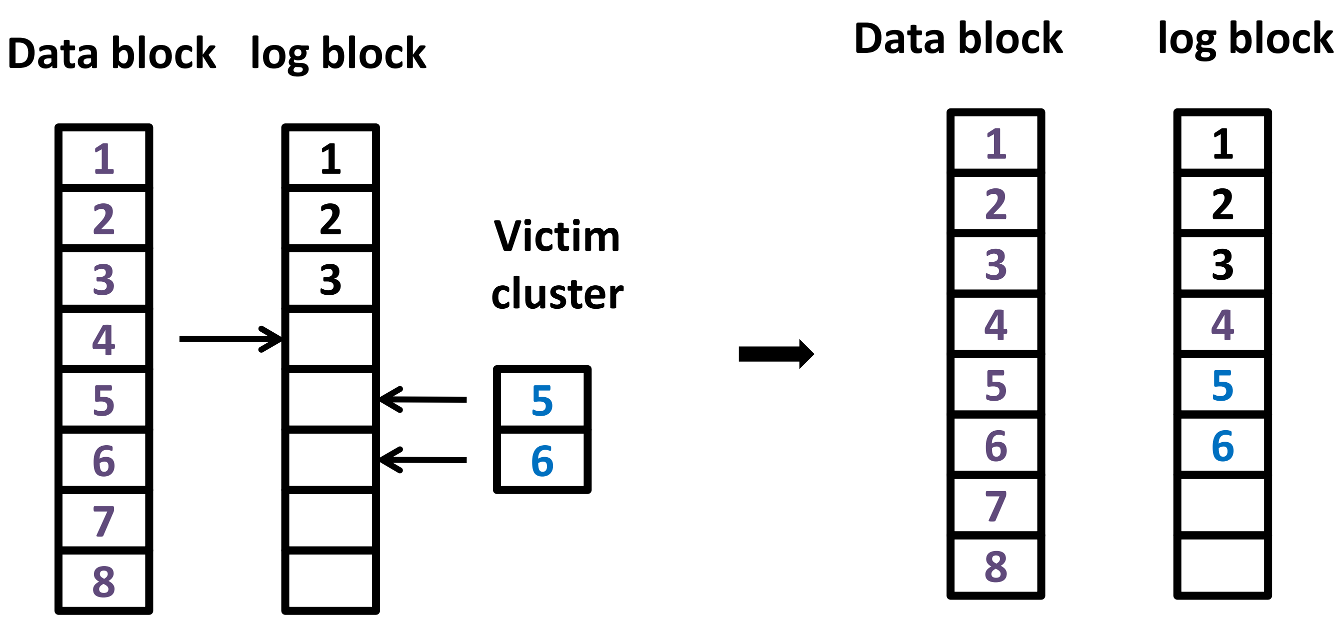
Optimistic FTL

- Log block management :
  a) Append operation
  b) Data block switch operation
  c) Log block switch operation
- **N_B**: the block size in number of pages
- **LPI** stores the index of the last page stored in the log block.
- **I_{min}** and **I_{max}** be the smallest and largest page index in the victim cluster, respectively.

Append operation

- When LPI < l_{min} :

No erase operation is needed.

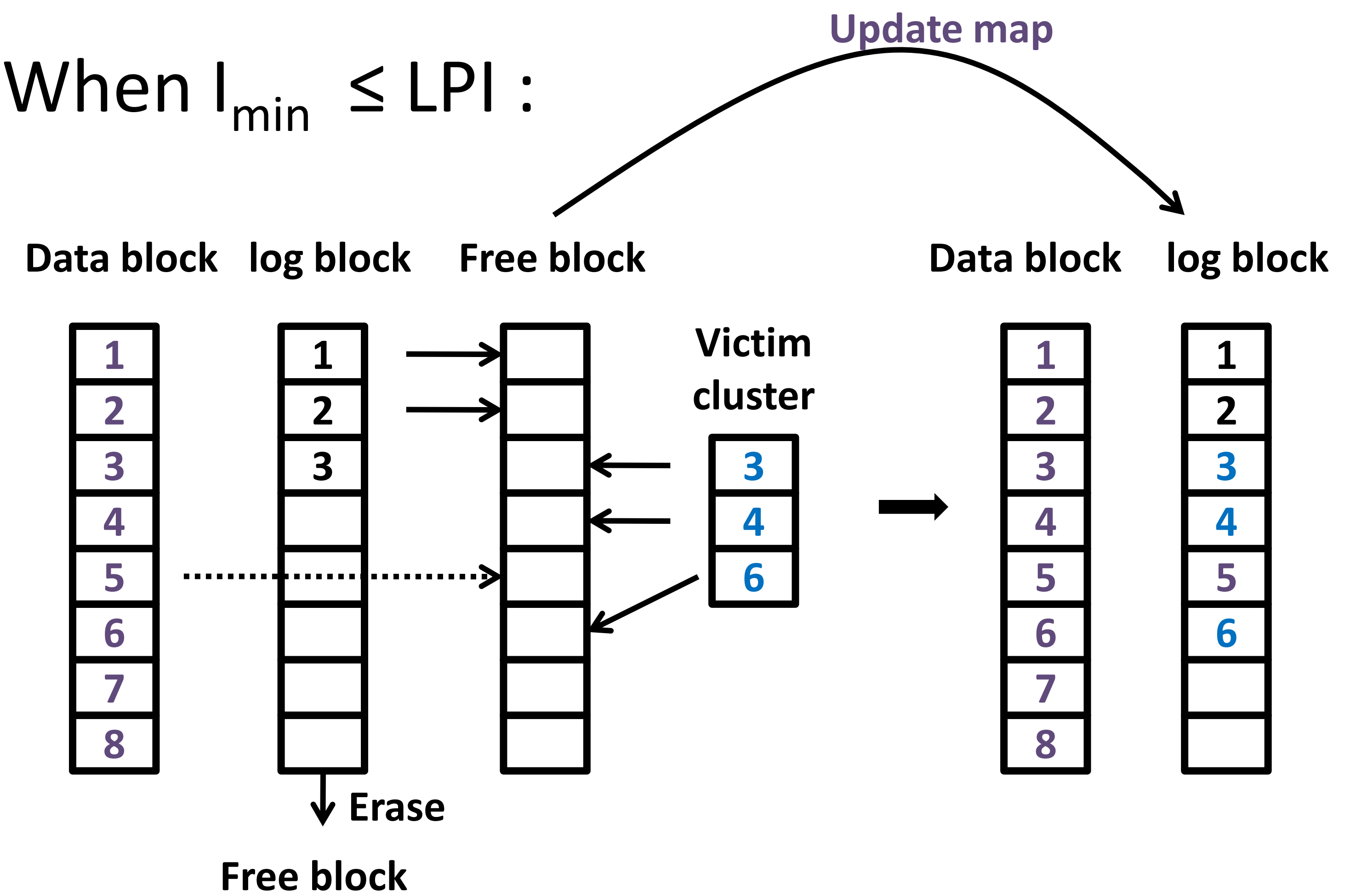


LPI = 3 ; l_{min} =5 ; l_{max} = 6 ;

LPI = 6 ;

Log block switch operation(1/2)

- When $I_{min} \le LPI$:

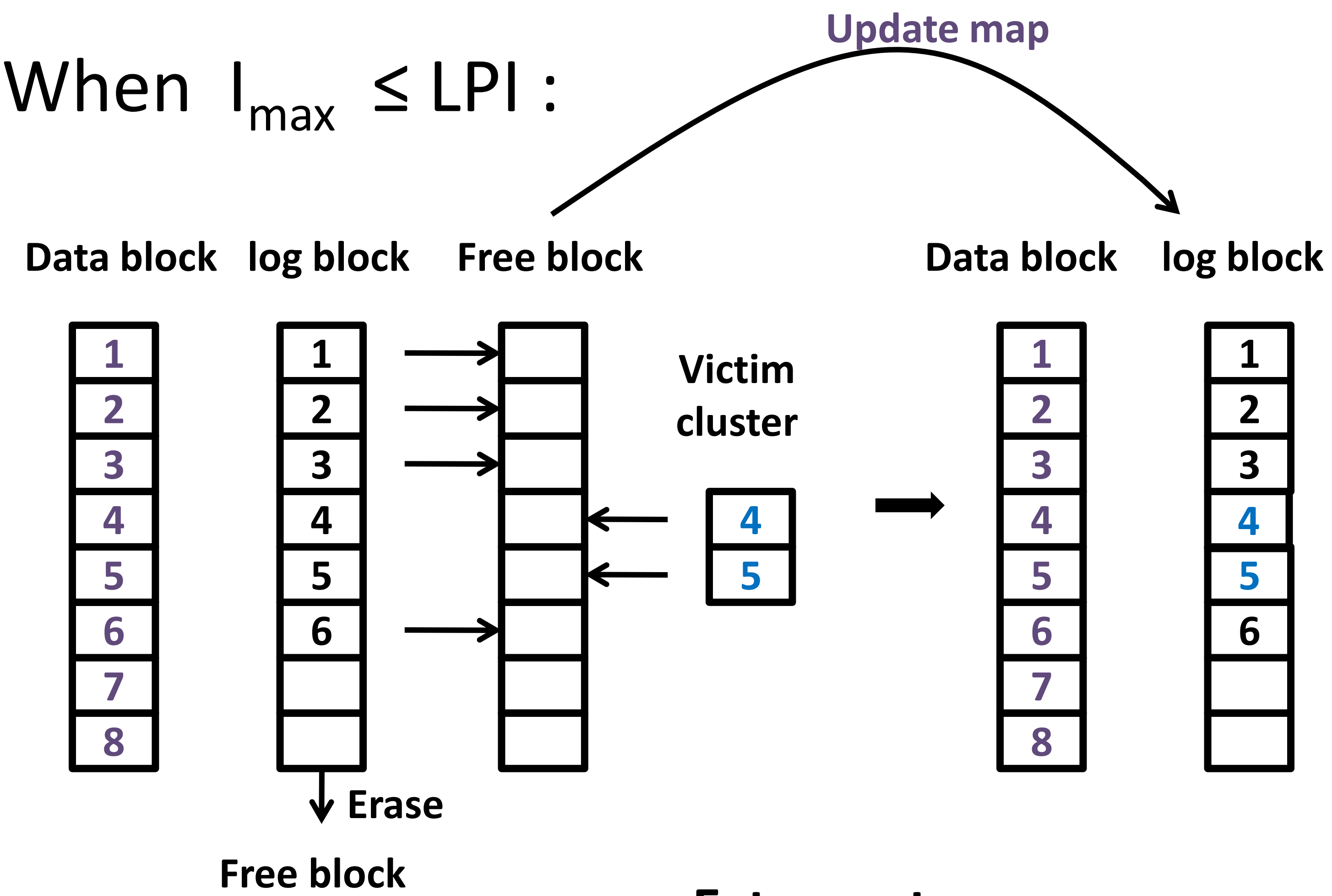


$LPI = 3$; $I_{min} = 3$; $I_{max} = 6$;

$LPI = 6$;

Log block switch operation(2/2)

- When $l_{max} \le LPI$:



LPI = 6 ;

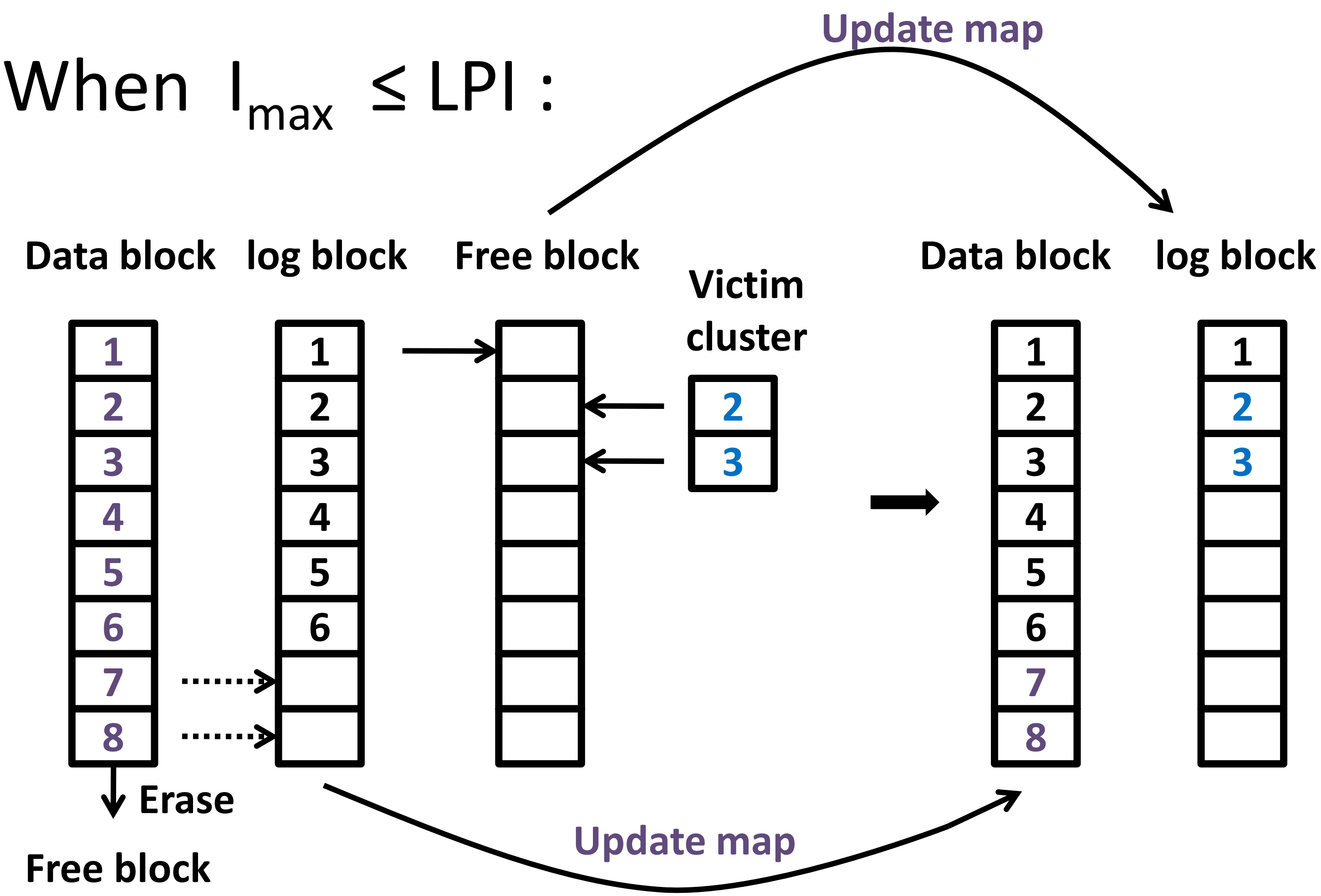
Extra cost :

$LPI - (l_{max} - l_{min} + 1) = LPI - l_{max} + l_{min} - 1$

$LPI = 6$; $l_{min} = 4$; $l_{max} = 5$;

Data block switch operation

- When $l_{max} \leq LPI$:



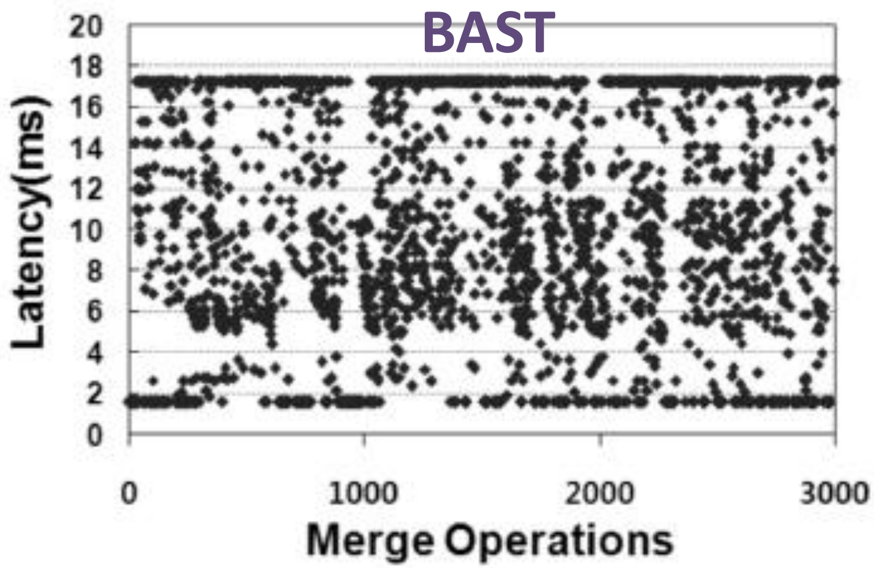
LPI = 3 ;

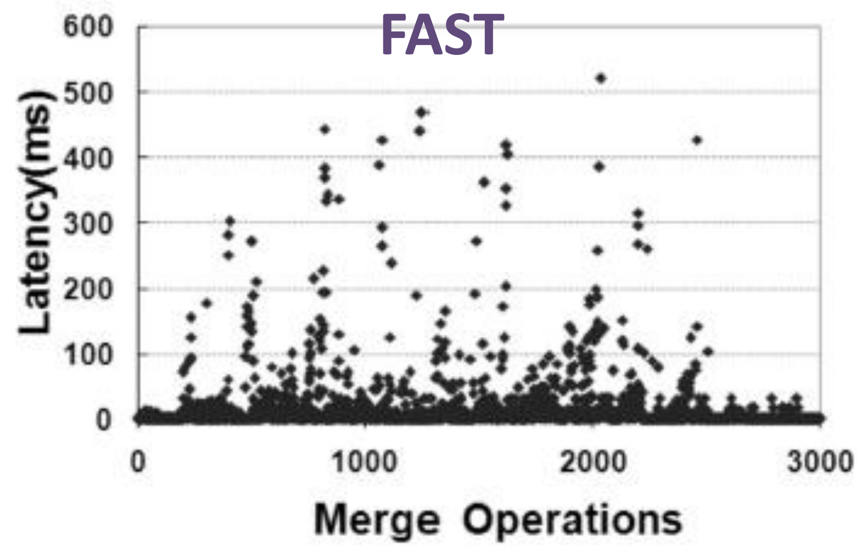
$LPI = 6$; $l_{min} = 2$; $l_{max} = 3$;

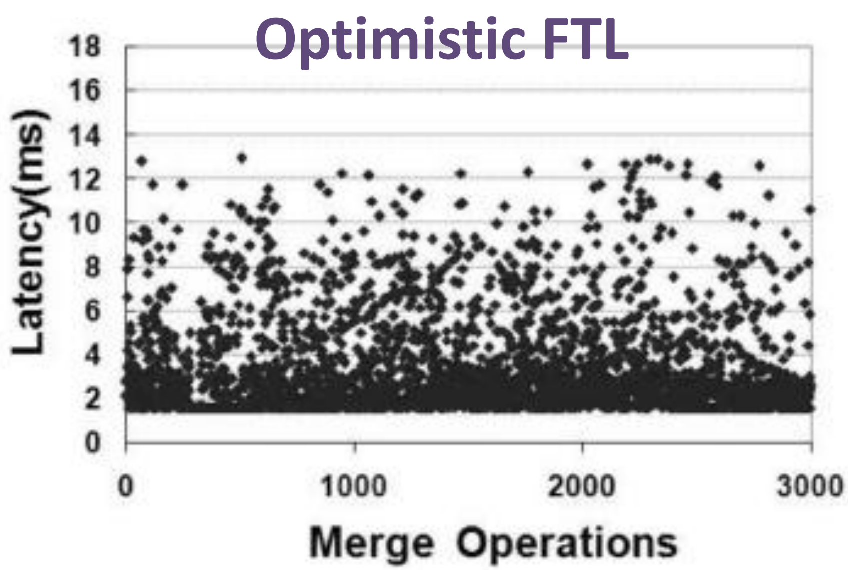
Extra cost :

$N_B - LPI + l_{min} - 1$

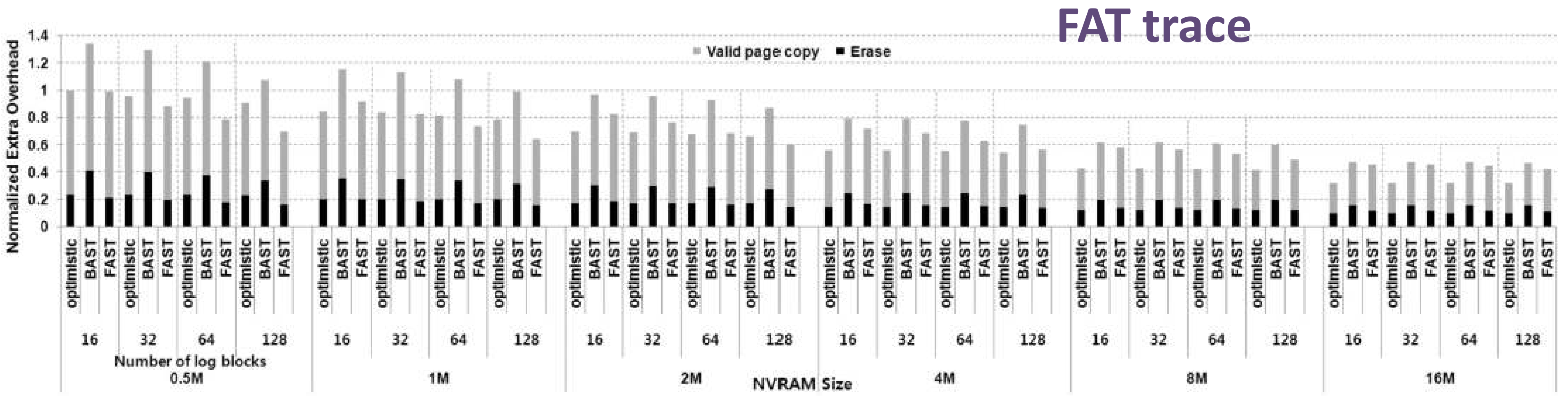
Merge latencies in each FTL algorithm

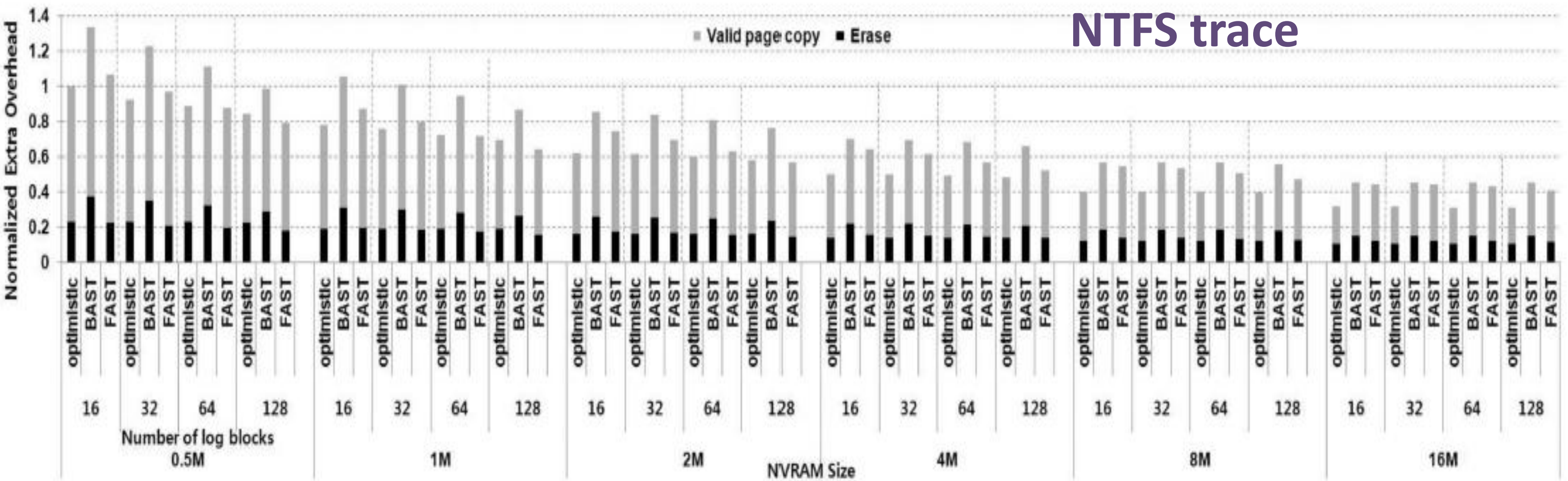






Extra overhead in each FTL algorithm.





Conclusion

- The CLC policy not only exploits the temporal locality but also maximizes the number of simultaneously destaged pages.

- Simulation results have shown that the CLC policy outperforms traditional pagelevel LRU policy (LRU-P) by a maximum of 51 percent.