Buffer Flush and Address Mapping Scheme for Flash Memory Solid State Disk

Dongkun Shin
Sungkyunkwan University, Korea
dongkun@skku.edu

Introduction

- Dramatic price reduction of flash memory
- SSD is emerging as a killer application for NAND flash (desktop PC, enterprise server, camcorder)
- Pros
  - Low power consumption, high reliability and high random access performance
- Cons
  - Expensive cost
- To reduce the cost of SSD,
  - MLC (multi-level cell) flash SSD is a popular recent solution
  - MLC has a slower performance and a shorter life span, making the performance of SSD a critical issue.

Hurdles towards High-Performance

- Slow write performance compared to read performance.
  - Use internal volatile write buffer (SDRAM)
    - long write latency is inevitable when the buffer should be flushed due to its limited capacity.
- Inferior sequential performance compared to HDD
  - Use parallel architecture (multi-channel and multi-way architecture)
    - Program multiple pages on different chips at a time
- Too large mapping information
  - Use coarse-grained mapping such as superblock
    - Large block merge overhead

MLAM

- Two critical issues on designing the NAND flash SSD
  - how to select victim pages for the write buffer flush
  - how to map logical address into physical address considering the parallel architecture of SSD
- Multi-level address mapping technique (MLAM)
  - victim page selection for the write buffer considering the block merge overhead
  - dynamically determines the mapping granularity based on the write pattern
    - Provide fast performance with small mapping table

SSD Architectures

- Park [NVSM'06] : multi-channel and multi-way controller
- Kang [JSA'07] : striping, interleaving and pipelining
- Chang [ASP-DAC'08] : hybrid SSD architecture
- Agrawal [USENIX'08] : trace-driven simulator
  - page-level mapping (async mode)
  - superpage-level mapping (sync mode)
- Shin [ICS'09] : page stripping methods

- No intensive research on the address mapping for flash memory SSD.

Multi-Level Address Mapping

- Wu [ICCAD'05] : two-level address mapping scheme that dynamically switches between page-level and block-level mappings
- Chang [TOS'05] : tree-based management scheme that adopts multiple granularities
- u-FTL [EMSOFT'08] : multi-level mapping managed by u-tree

- No consideration of the parallel handling for interleaved flash chips in SSD

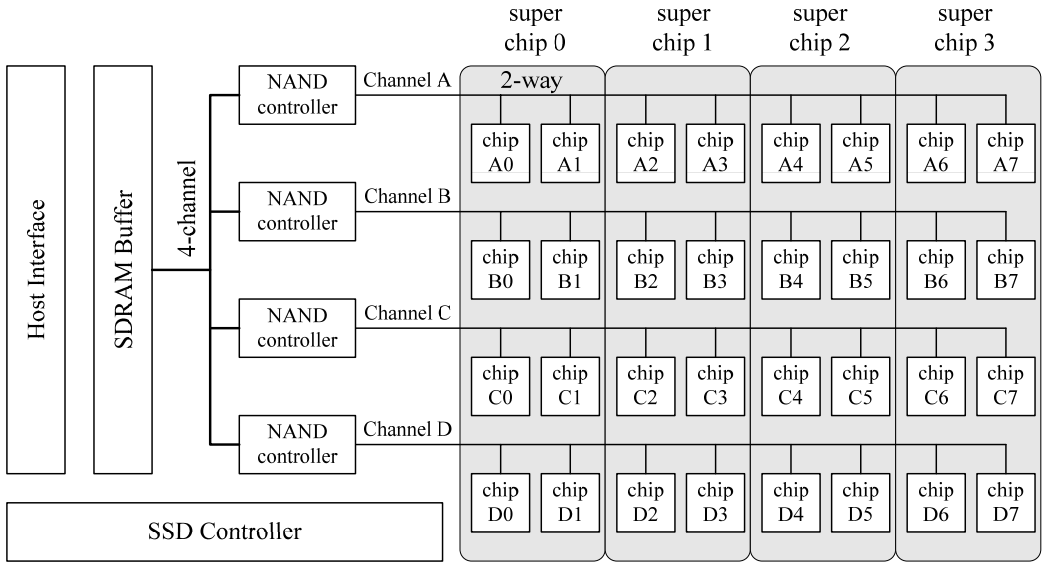
Flash-Aware Buffer Schemes

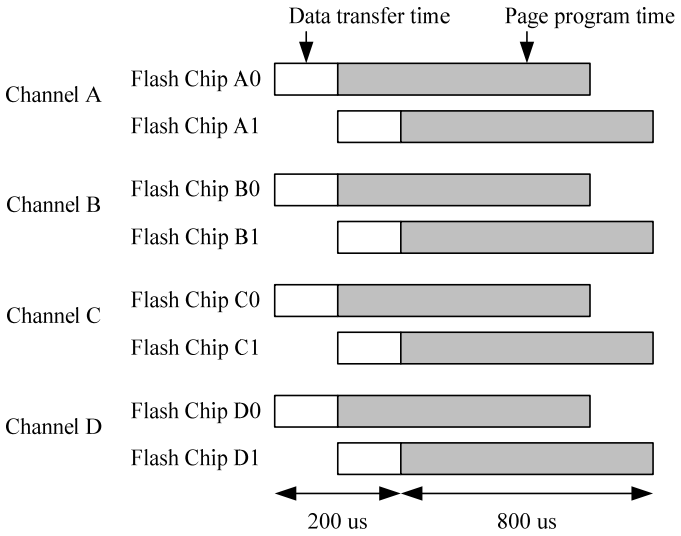
- CFLRU: delays the flush of dirty pages in buffer cache
- FAB: block-level buffer replacement
- BPLRU: block-level LRU policy and block padding
- REF: considers the recent history on log buffer

- No buffer management scheme considering the parallel architecture of SSD

SSD Internals

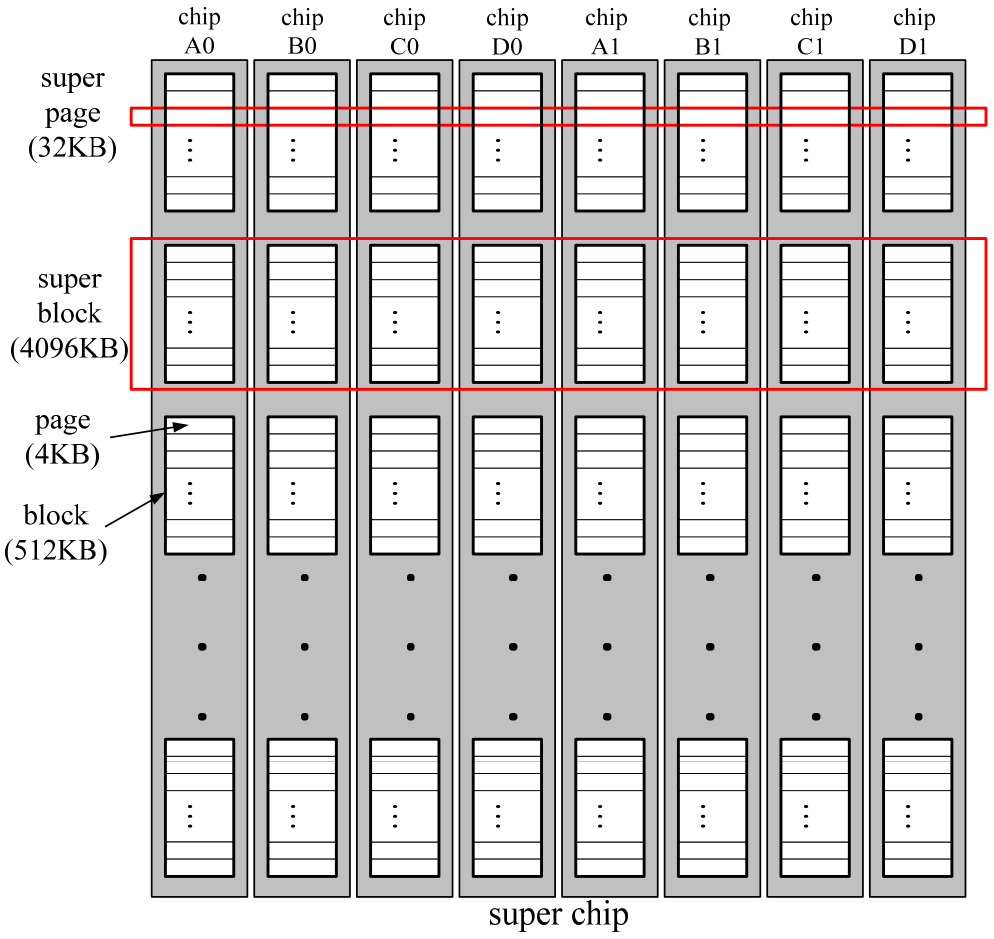
- SDRAM Buffer: temporally stores data from the host
- Multi-Channels: can be accessed simultaneously
- Multi-Ways: can be accessed in interleaved manner
- Superchip: A group of chips which can be accessed simultaneously.





Superpage and Superblock

- Superpage (page group)
  - A group of pages which can be accessed in parallel
  - All pages have the same offset within a chip
- Superblock (block group)
  - Extension of superpage to a group of blocks.



Address Mapping

- Goal: minimize block merge overhead with small mapping table
  - Page mapping: chip selection issue, async or sync, too large map table
  - Superpage mapping (hybrid mapping): fragmentation, large map table
  - Superblock mapping: fragmentation, large SB merge overhead
  - Multi-level mapping

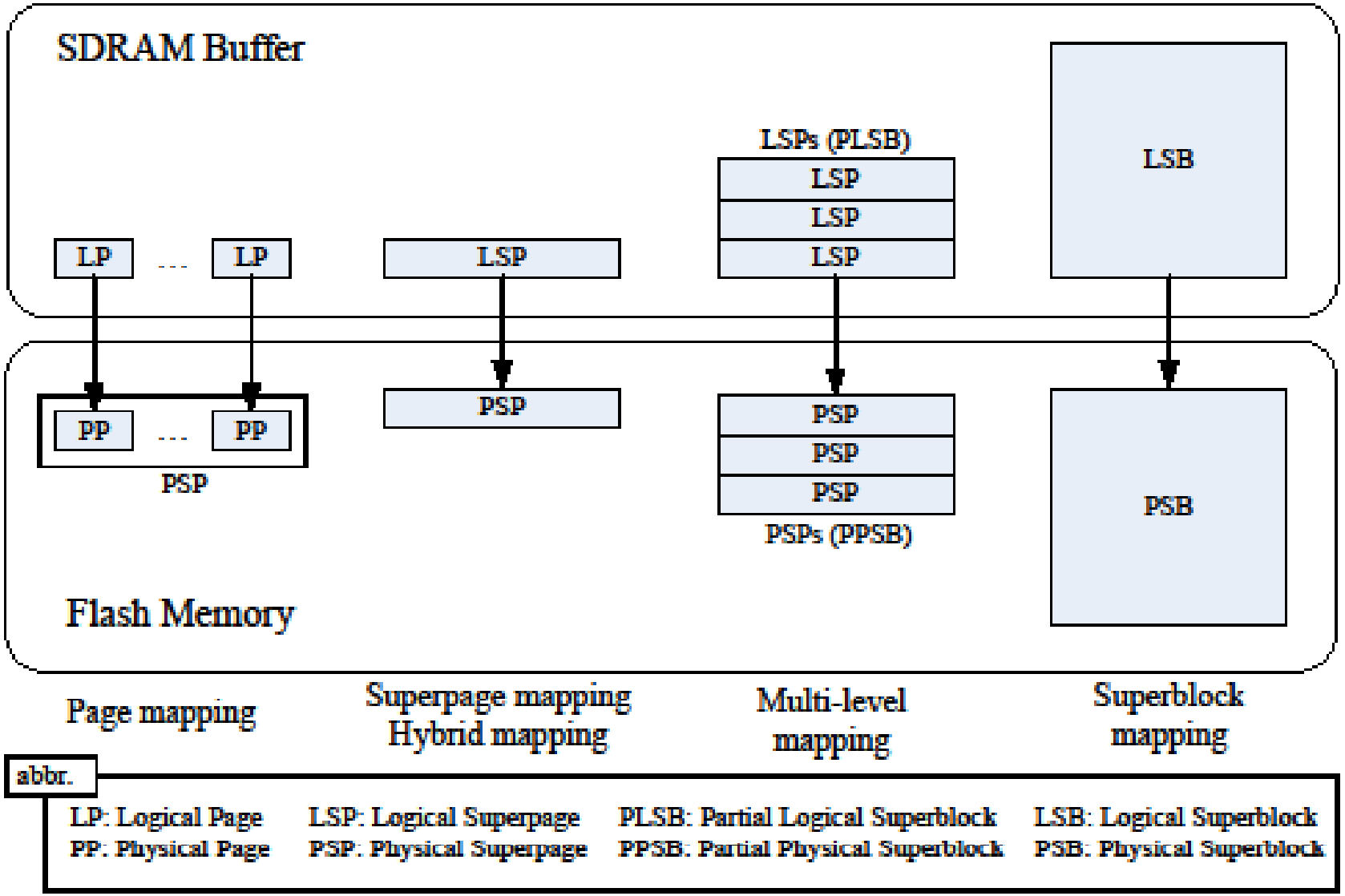
Mapping Table

128GB SSD

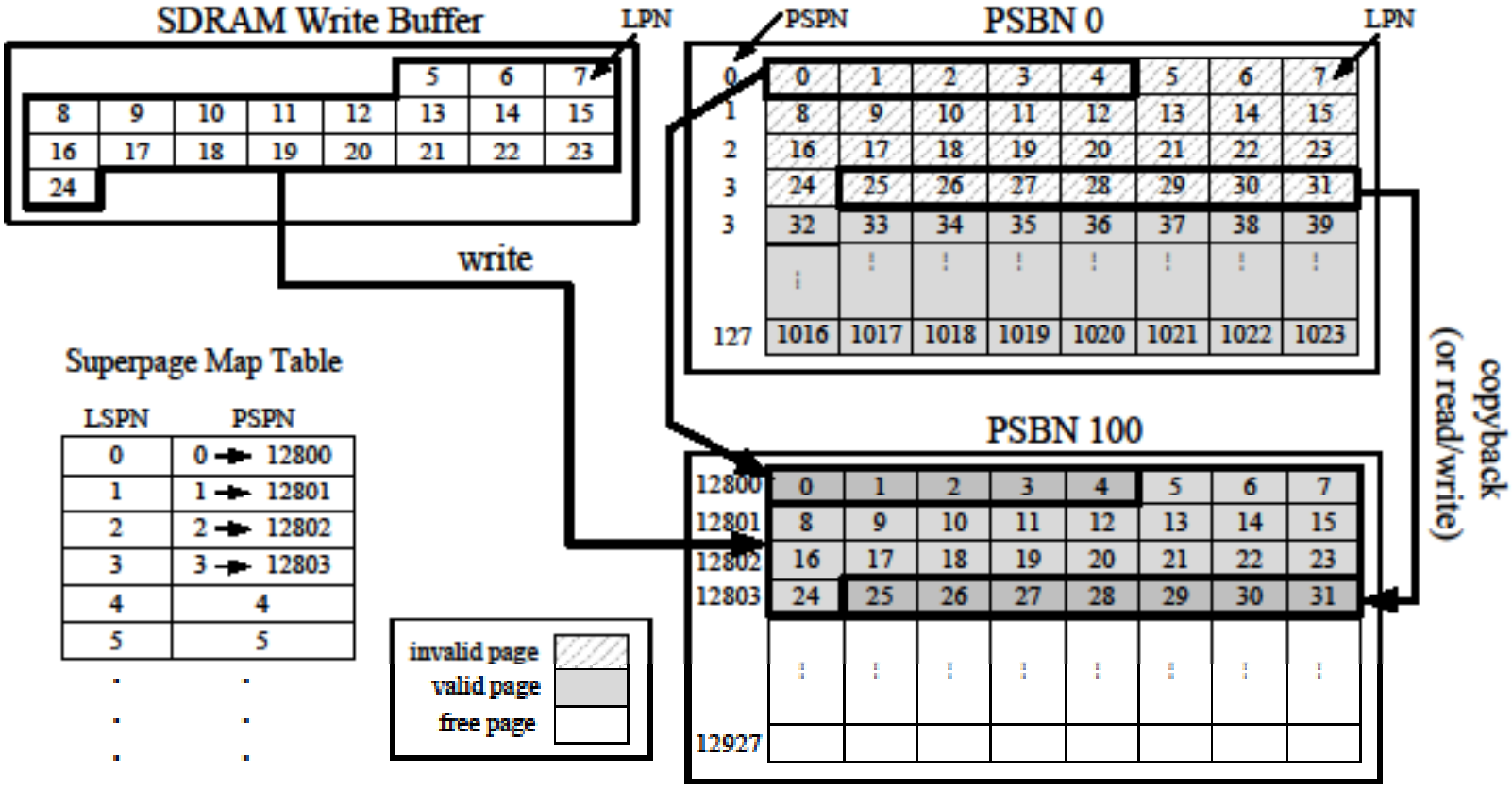
Mapping Level		Entry Size	# of Entry	Total Size
Page-level		4 bytes	128GB/4KB = 32M	128 MB
Superpage-level		3 bytes	128GB/32KB = 4096K	12 MB
Superblock-level		2 bytes	128GB/4MB = 32K	64 KB
Hybrid-level	Log	3 bytes	13GB/32KB= 400K	1.2MB
	Data	2 bytes	115GB/4MB = 29K	

Page: 4KB
Superpage: 32KB
Superblock: 4MB
Hybrid: log buffer is 10% of total storage

Mapping Levels



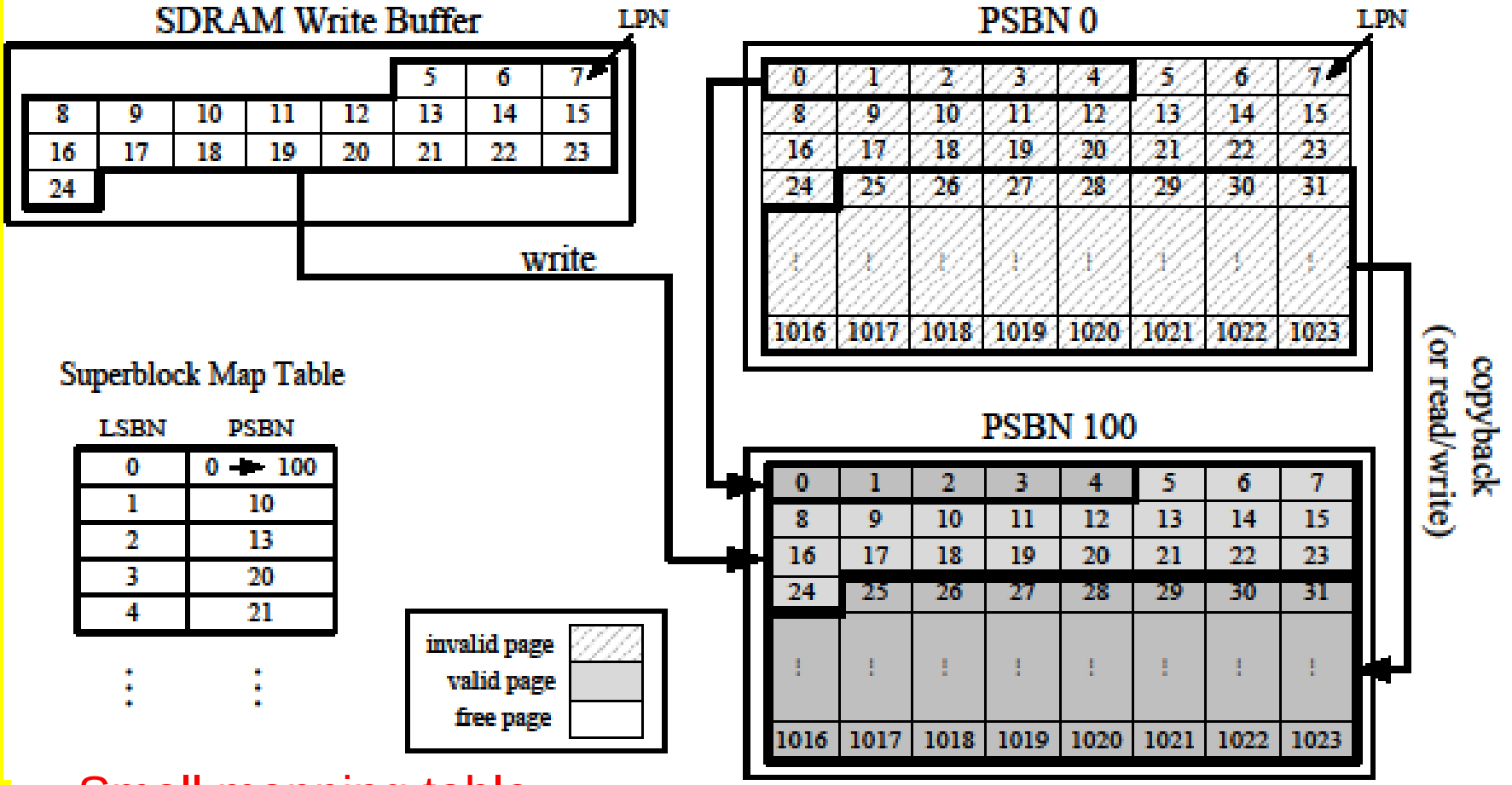
Superpage-Level Mapping



- Small mapping table compared to page-level mapping, but still too large in large-scaled SSD
- Fragmentation (there are unused pages)
- Requires copyback for unmodified pages

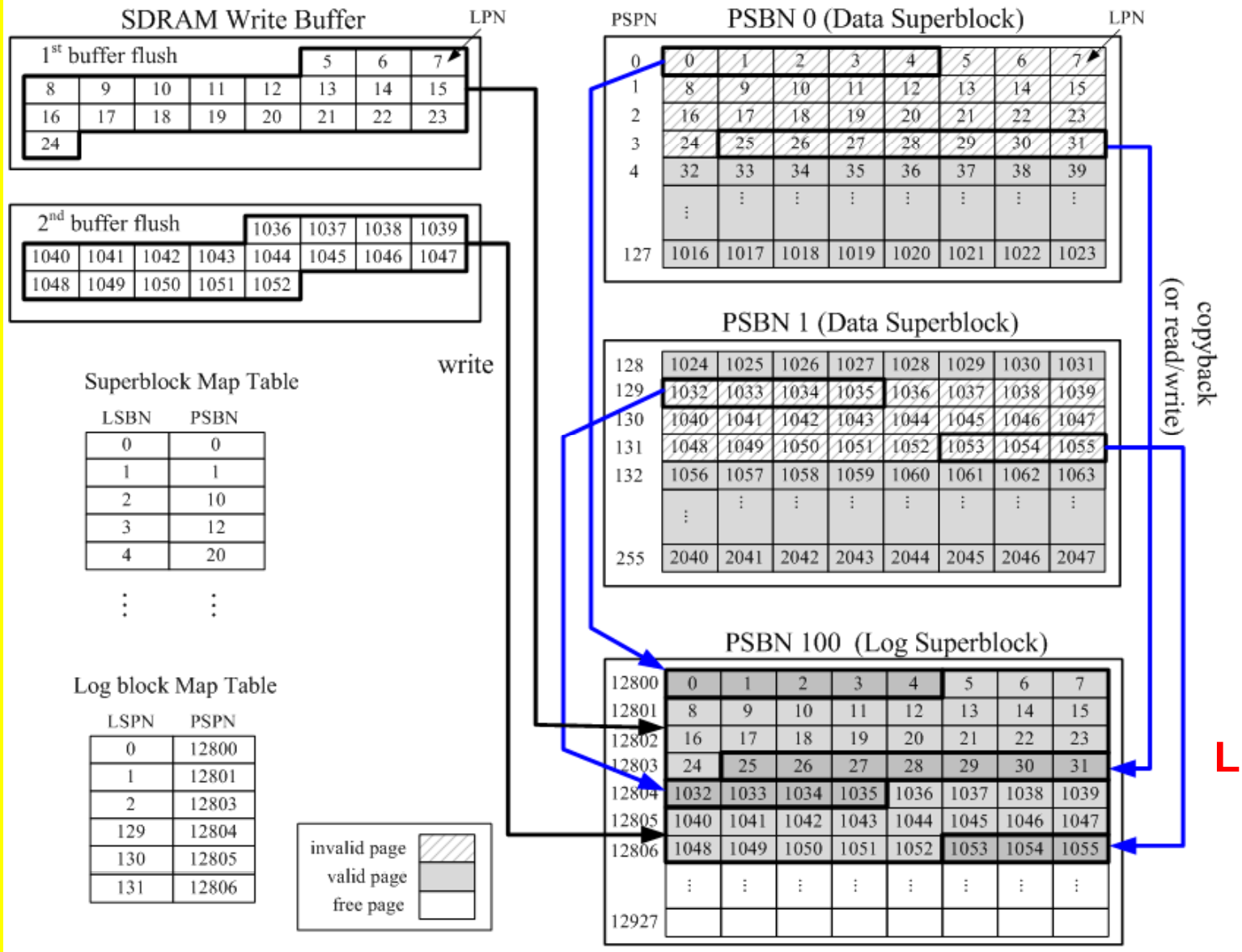
(LPN % N_{chip}) = ChipID
(in-place for all pages)

Superblock-Level Mapping



- Small mapping table
- Large fragmentation
- Superblock **merge overhead** for small-sized requests

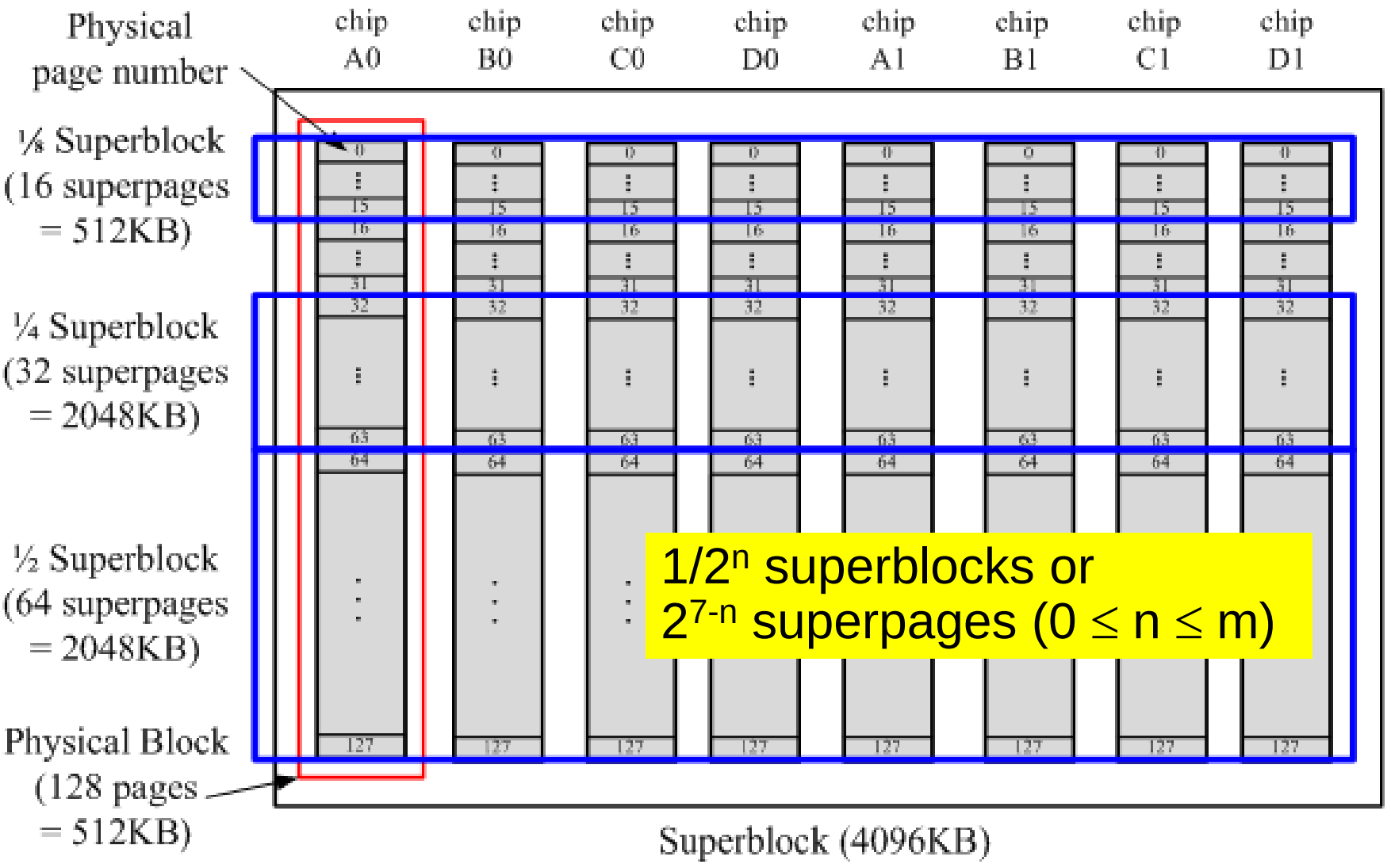
Hybrid Mapping



Log Buffer Useful in SSD ?

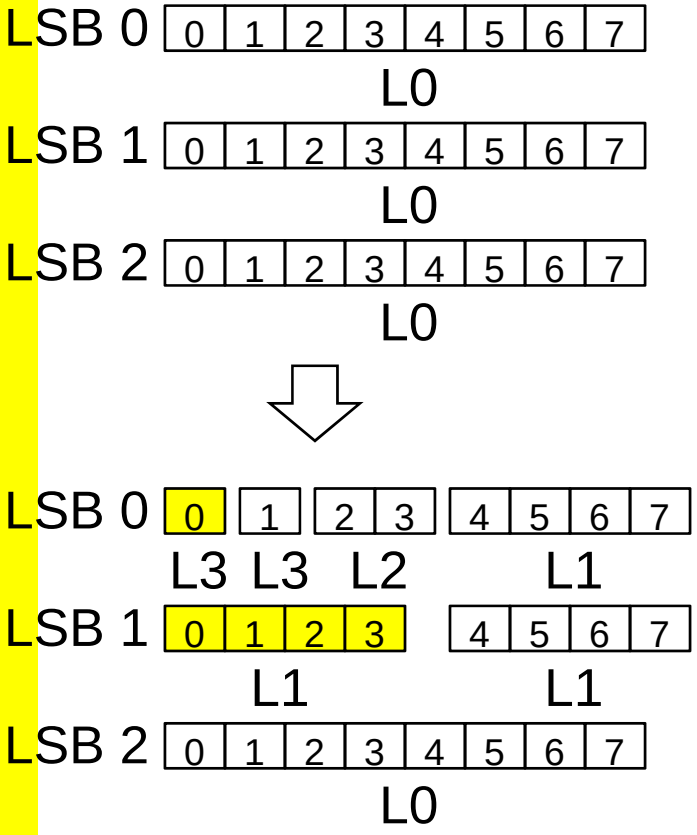
- Superpage-level or hybrid-level mapping will be more efficient than superblock-level mapping if a workload has high temporal locality and low spatial locality (random pattern).
- However, write requests on flash chips come through several buffers, which perform merging and sorting for small-sized write requests
- Therefore, they have little temporal locality but high spatial locality (due to buffer's merging operation)
- How about multiple mapping granularity?
- But arbitrary mapping granularities require high complexity (eg. u-FTL)

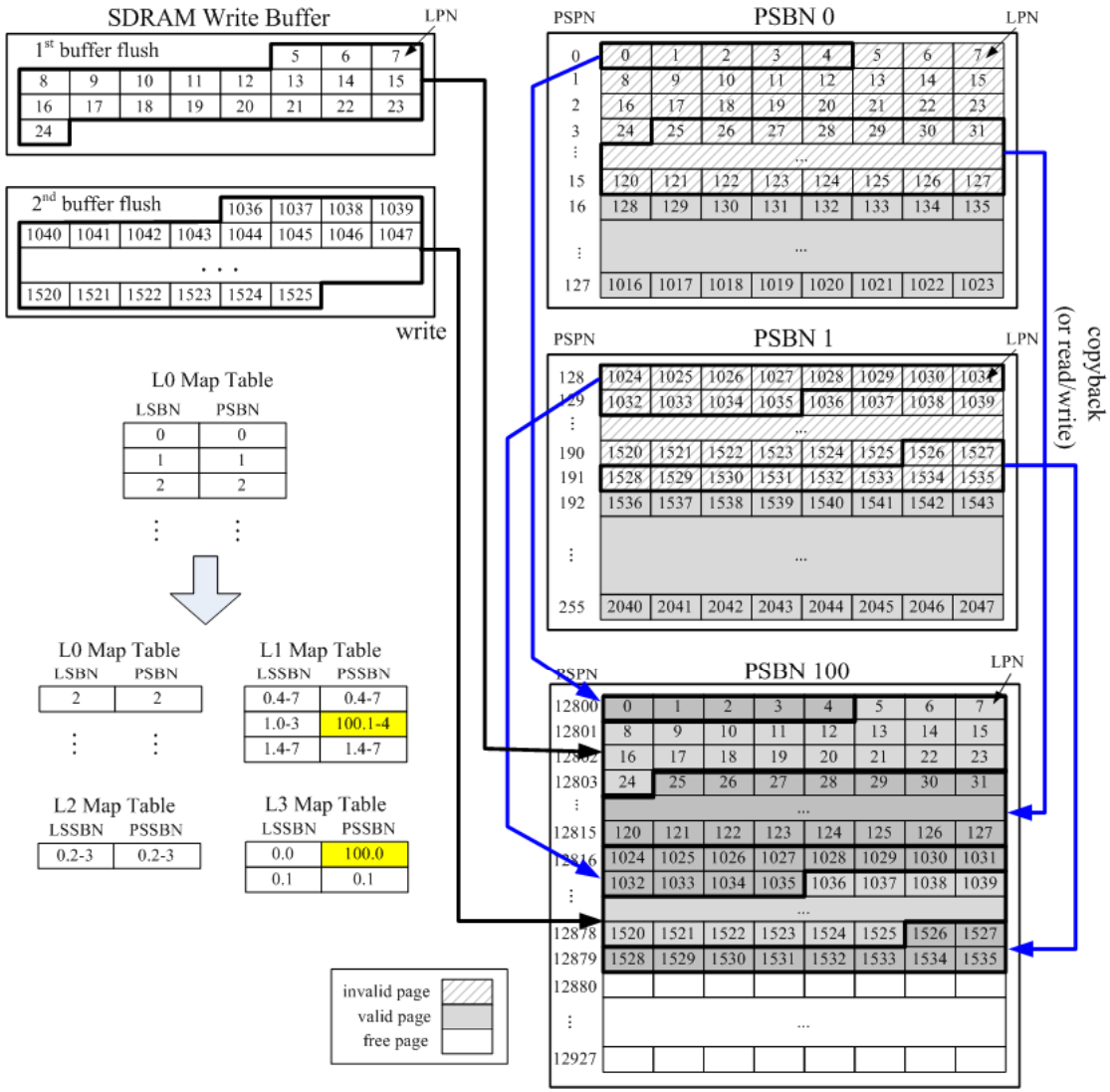
Sub-Superblock



Multi-Level Mapping

Find the largest mapping unit which invokes a merge overhead less than the predefined portion.





Virtual Superblock Composition

- Sub-superblock writing invokes the fragmentations within PSB
- Write by the unit of PSB
- Compose one virtual superblock with several sub-superblocks and write the VSB at a PSB
- We need several victim logical superblocks to compose a VSB

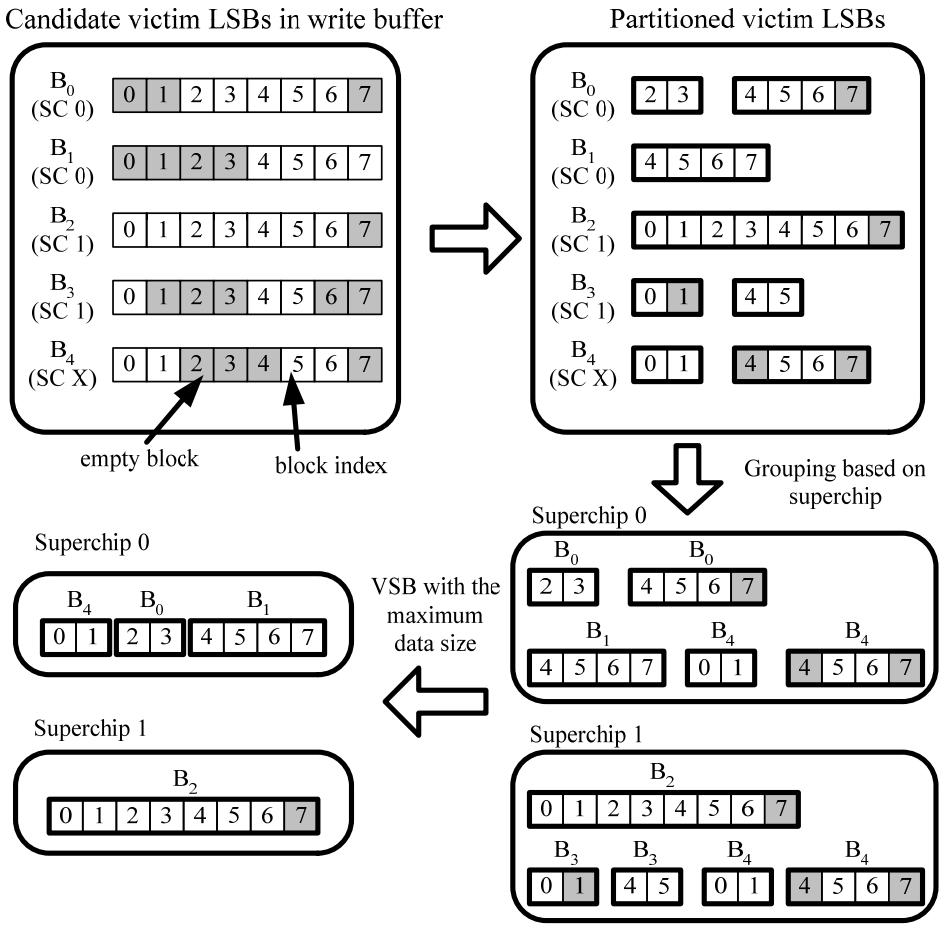
Victim LSB Selection

- SIZE policy
  - Choose the biggest LSB which means that most data are to be updated.
  - Small-sized LSB could remains without being flushed.
- LRU policy
  - Choose the LSB which has not been accessed for the longest time.
  - Old and small-sized LSB may deteriorate performance.
- LRU+Size policy
  - Consider both two factors

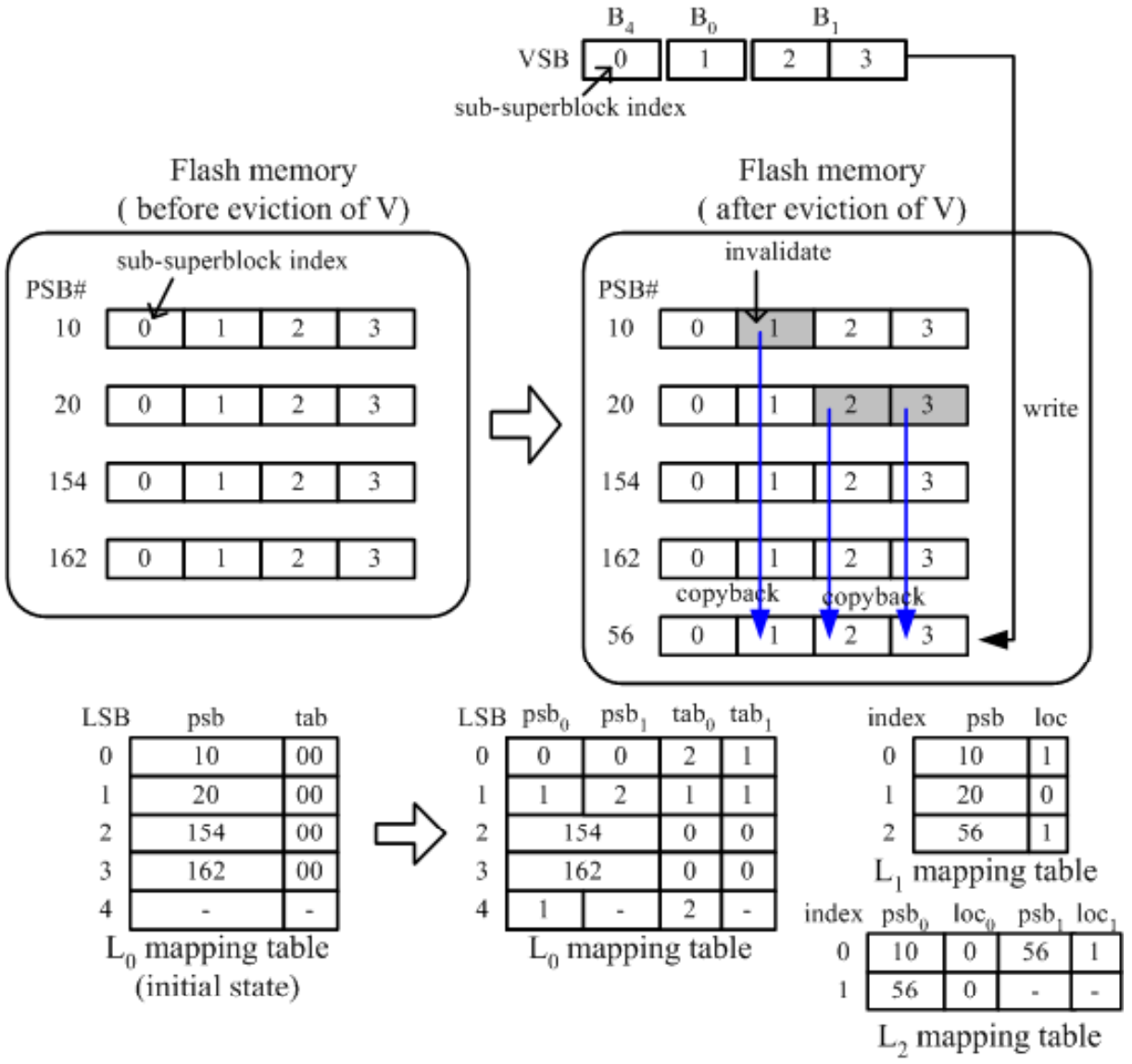
$$Pr(B_i) = \alpha \cdot \frac{t(B_i)}{T} + (1-\alpha) \cdot \frac{n_{page}(B_i)}{N}$$

Virtual Superblock Composition

- Each victim LSB is partitioned into sub-SBs if it has more than k_{empty} empty blocks
- Group the victim sub-SBs based on the superchip index
- Compose a VSB for each superchip such that it has the largest number of updated pages
- Select the largest-sized VSB among the VSBs for several superchips



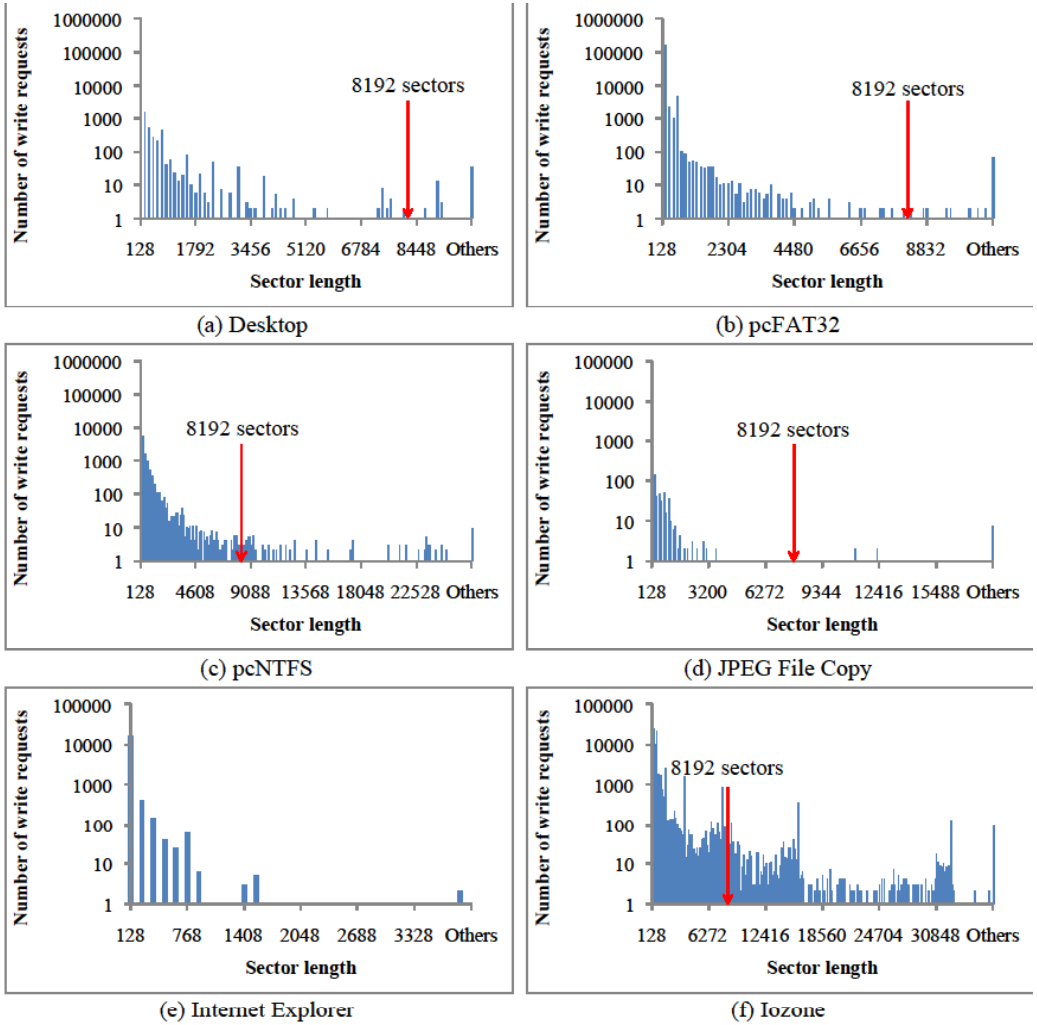
Multi-Level Address Mapping



Experiments

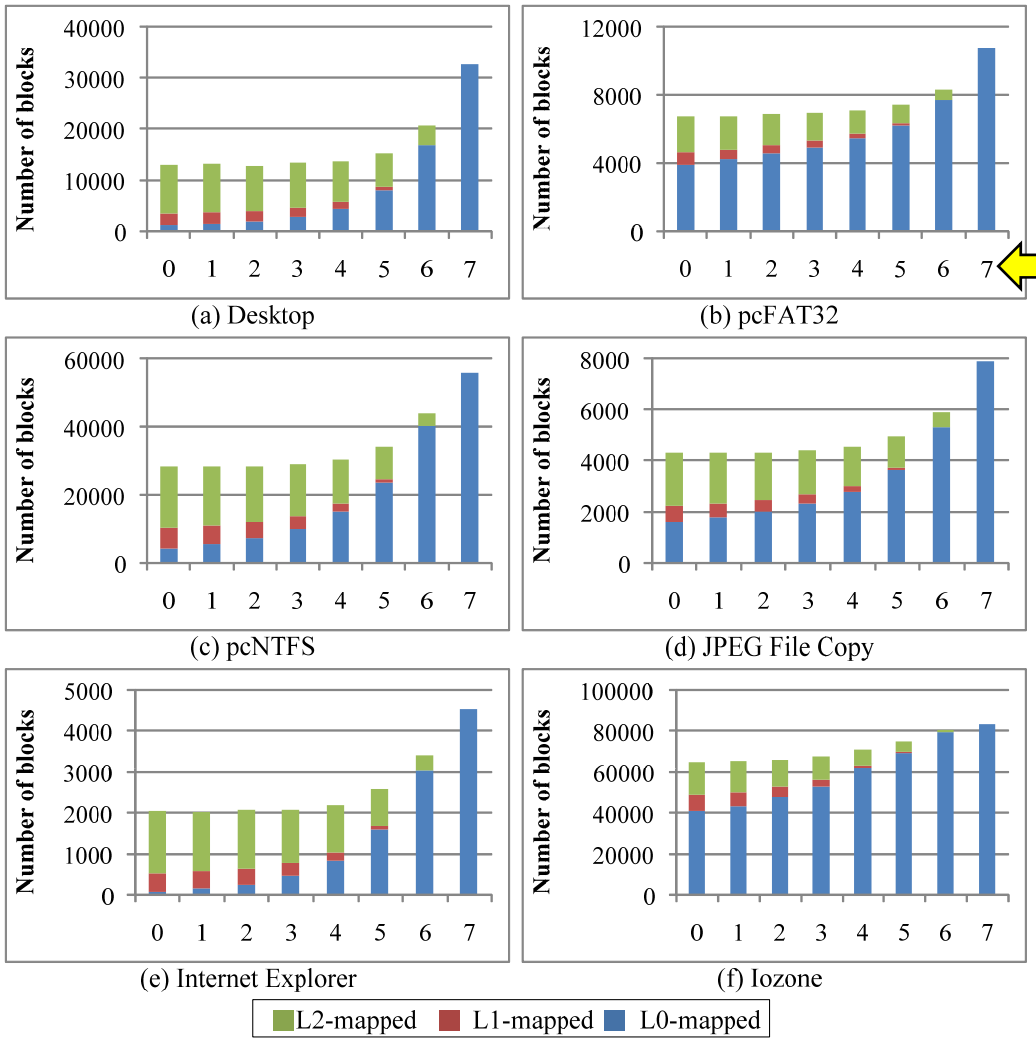
- Our SSD simulator
- 4-channel and 2-way
- 16~128 MB SDRAM
- 32 1GB MLC flash chips
- 5 real disk I/O traces and 1 benchmark trace

parameter	value	parameter	Value
Page size	4KB	Page read	60 μs
Block size	512KB (128 pages)	Page write	800 μs
Superpage size	32KB	Block erase	1.5 ms
Superblock size	4096KB	Page copyback	860 μs



Experiments

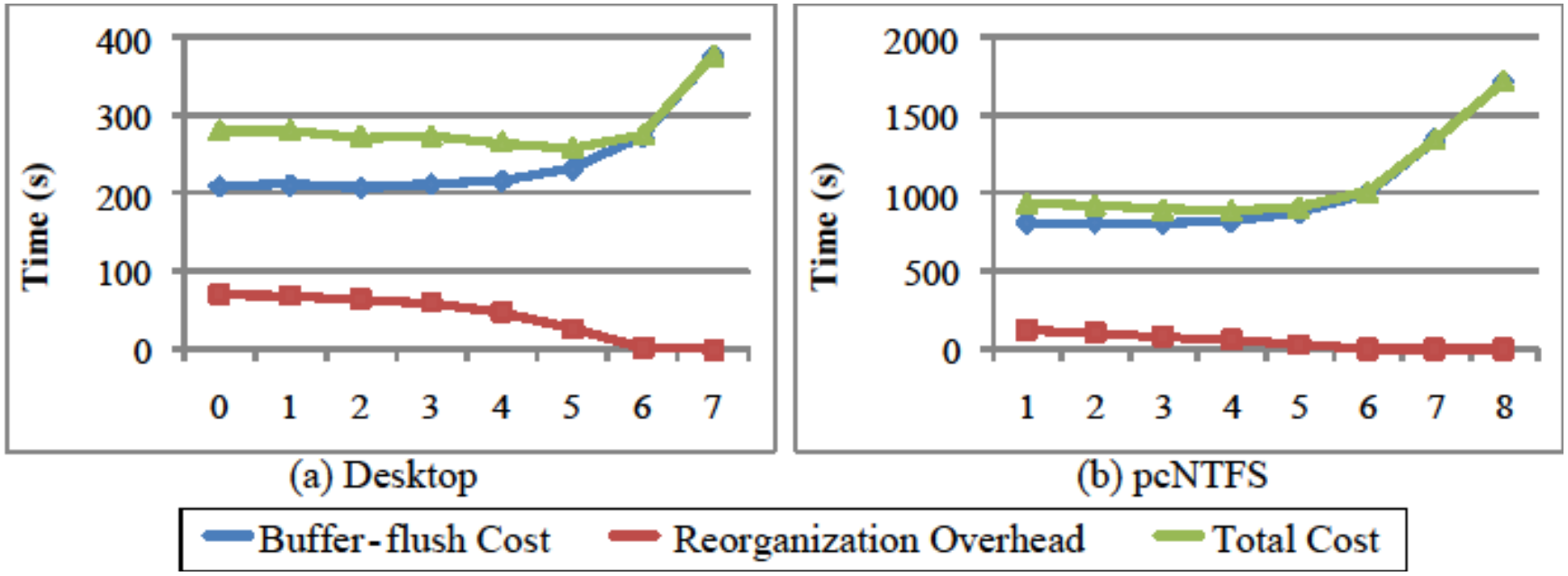
- Mapping level comparison with varying k_{empty}



same to superblock mapping

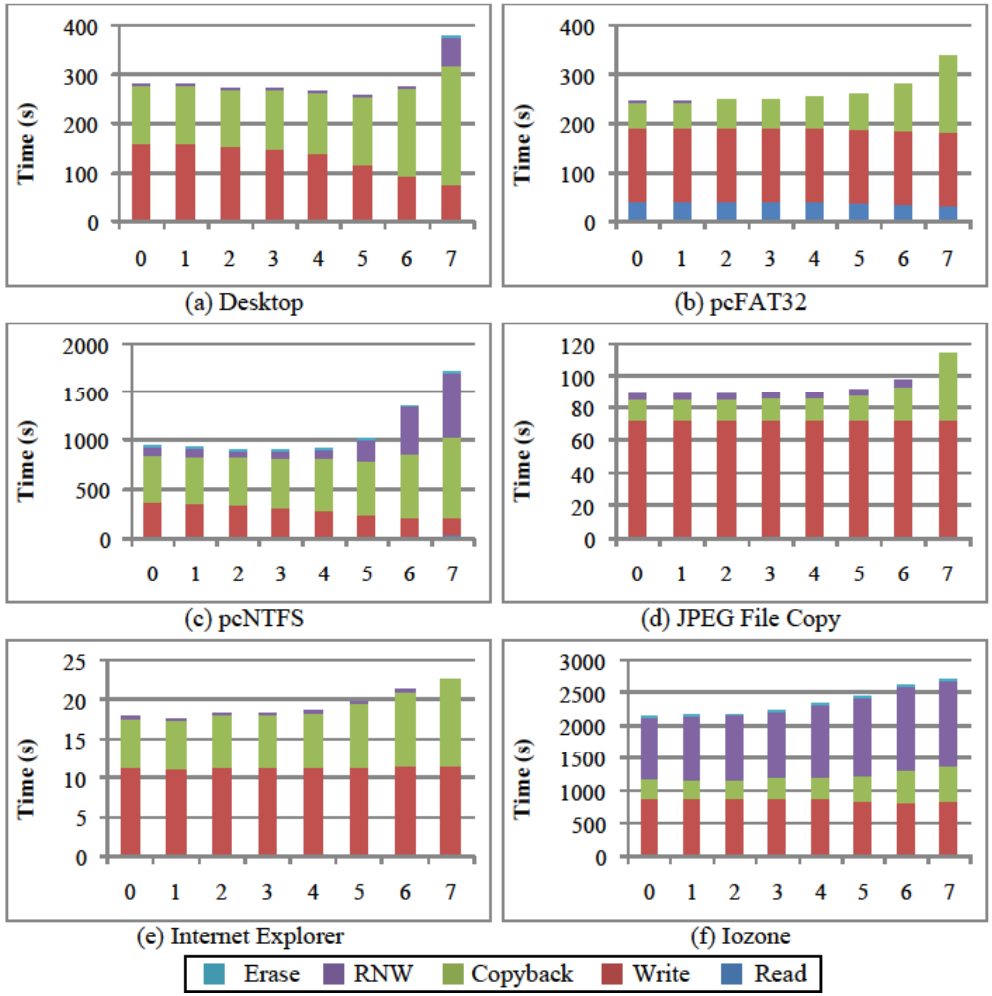
Experiments

- Execution time comparison with varying k_{empty}



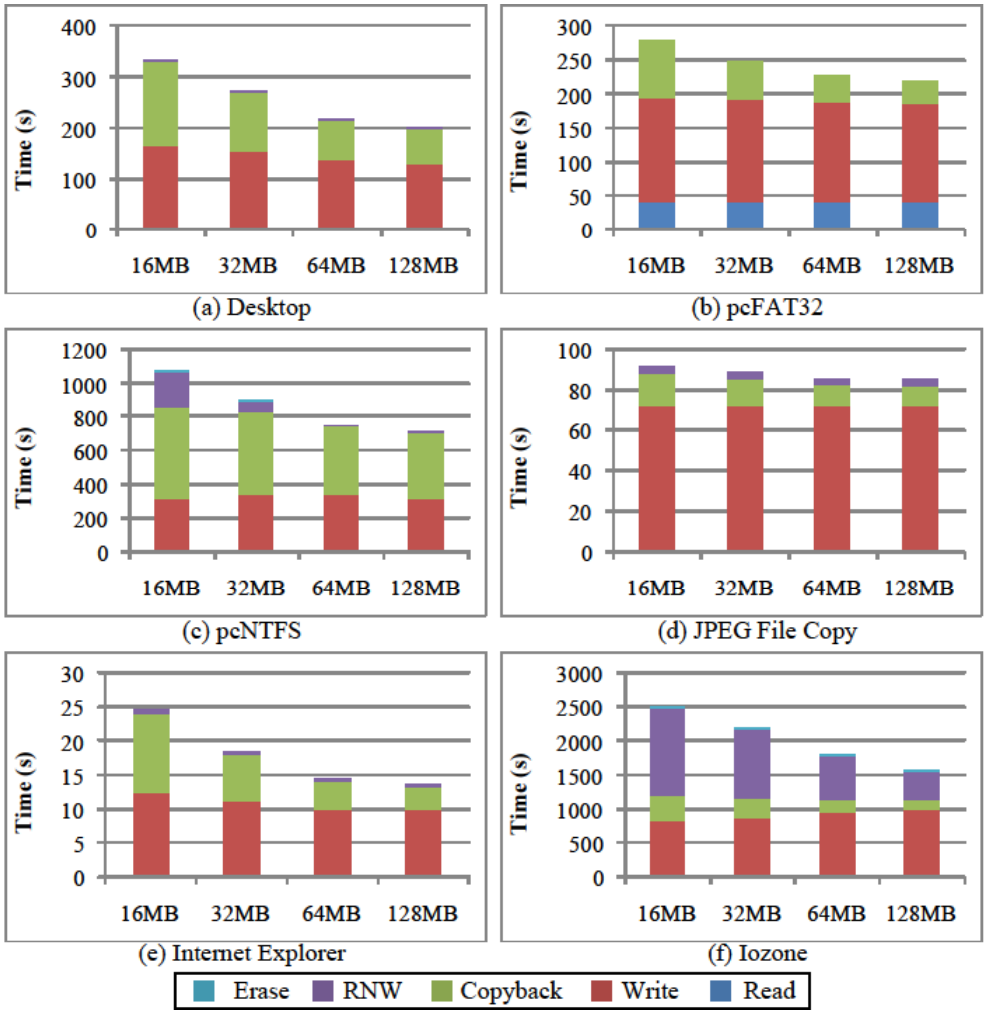
Experiments

- Execution time comparison with varying k_{empty}



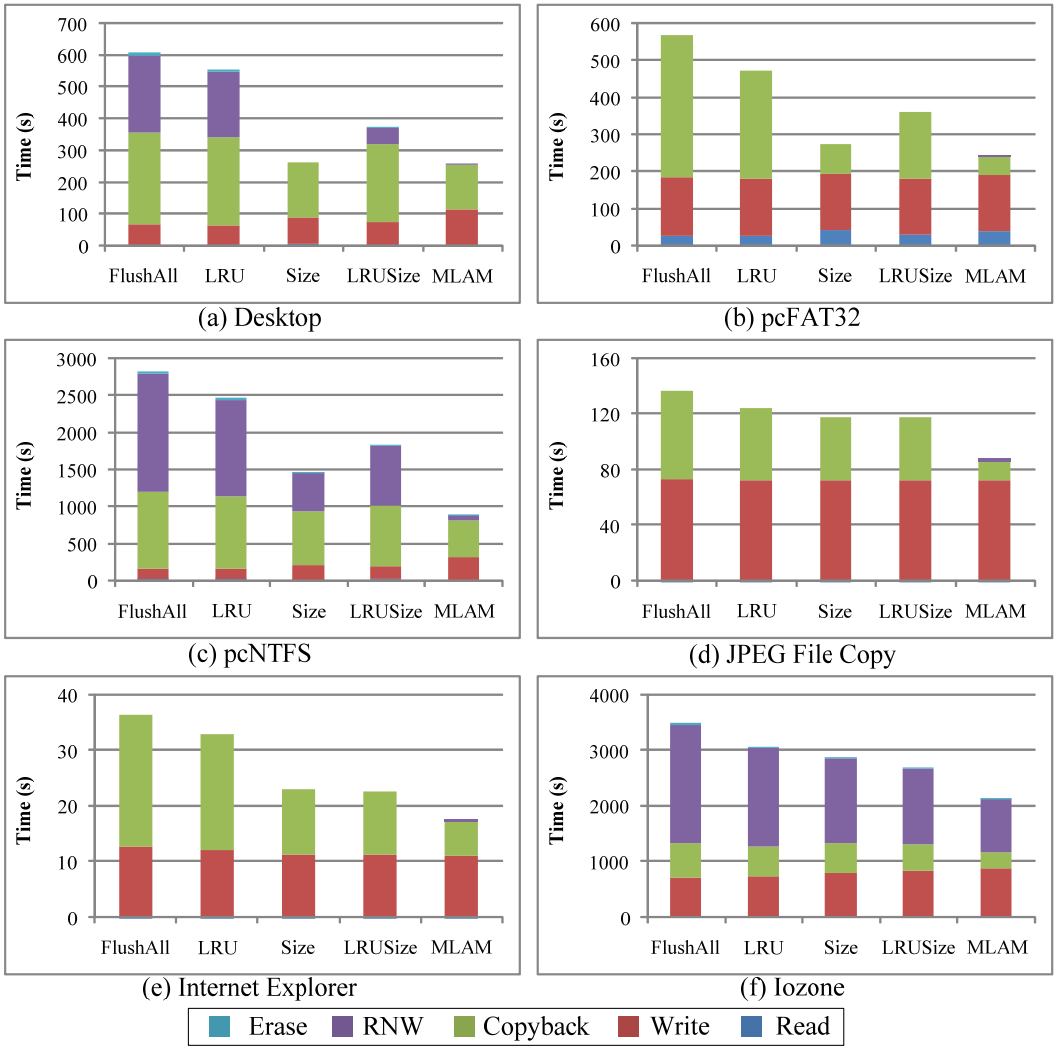
Experiments

- Execution time comparison while varying the buffer size



Experiments

- Comparison between victim selection policies



Conclusions

- The parallel architecture (multi-channel and multi-way) is essential to the high performance NAND flash SSD.
- The coarse-grained mapping can show poor performance when there are many random and scattered write requests.
- Can reduce the superblock merge overhead significantly by allowing multi-level mappings.