Address Mapping Scheme for Very Large Scale SSD

Dongkun Shin
Sungkyunkwan University
dongkun@skku.edu

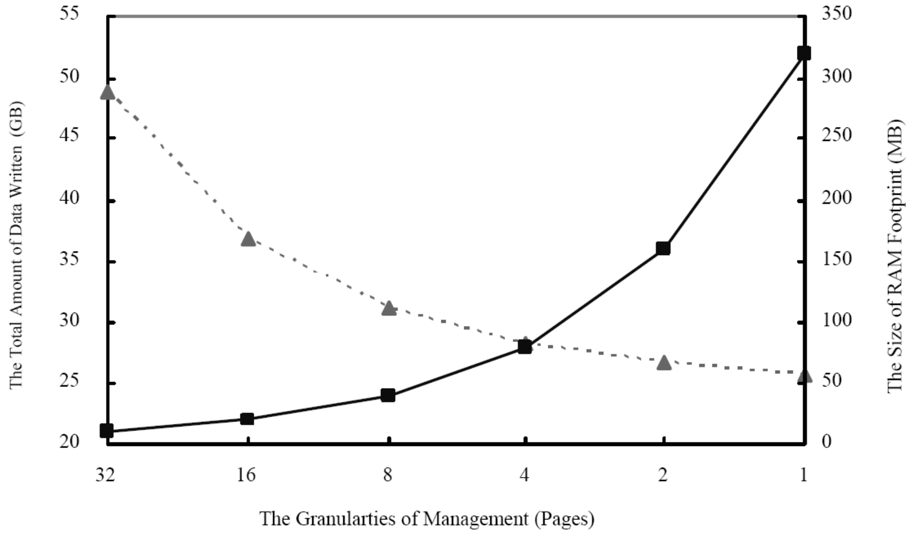
Mapping Table

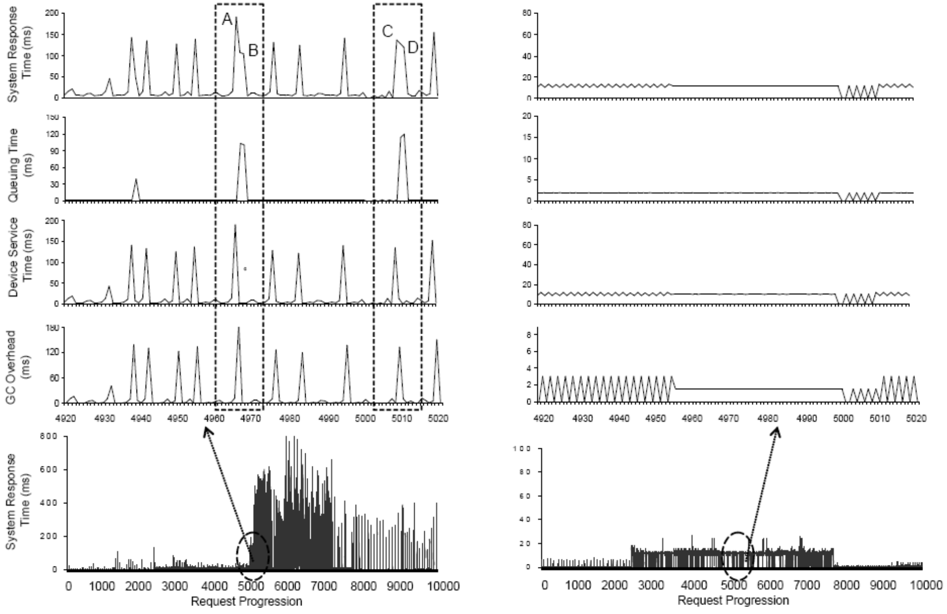
1TB SSD

Mapping Level		Entry Size	# of Entry	Total Size
Page-level		8 bytes	1TB/4KB = 256M	2 GB
Block-level		8 bytes	1TB/256KB = 4M	32 MB
Superpage-level		8 bytes	1TB/16KB = 64M	512 MB
Superblock-level		8 bytes	1TB/1MB = 1M	8 MB
Hybrid-level	Log	8 bytes	100GB/16KB= 6M	55.2MB
	Data	8 bytes	900GB/1MB = 900K	

page: 4KB, block: 256KB
Superpage: 16KB, Superblock: 1MB
Hybrid: log buffer is 10% of total storage
Page-level and superpage-level mappings require meta-data for block management additionally.

Trade-off





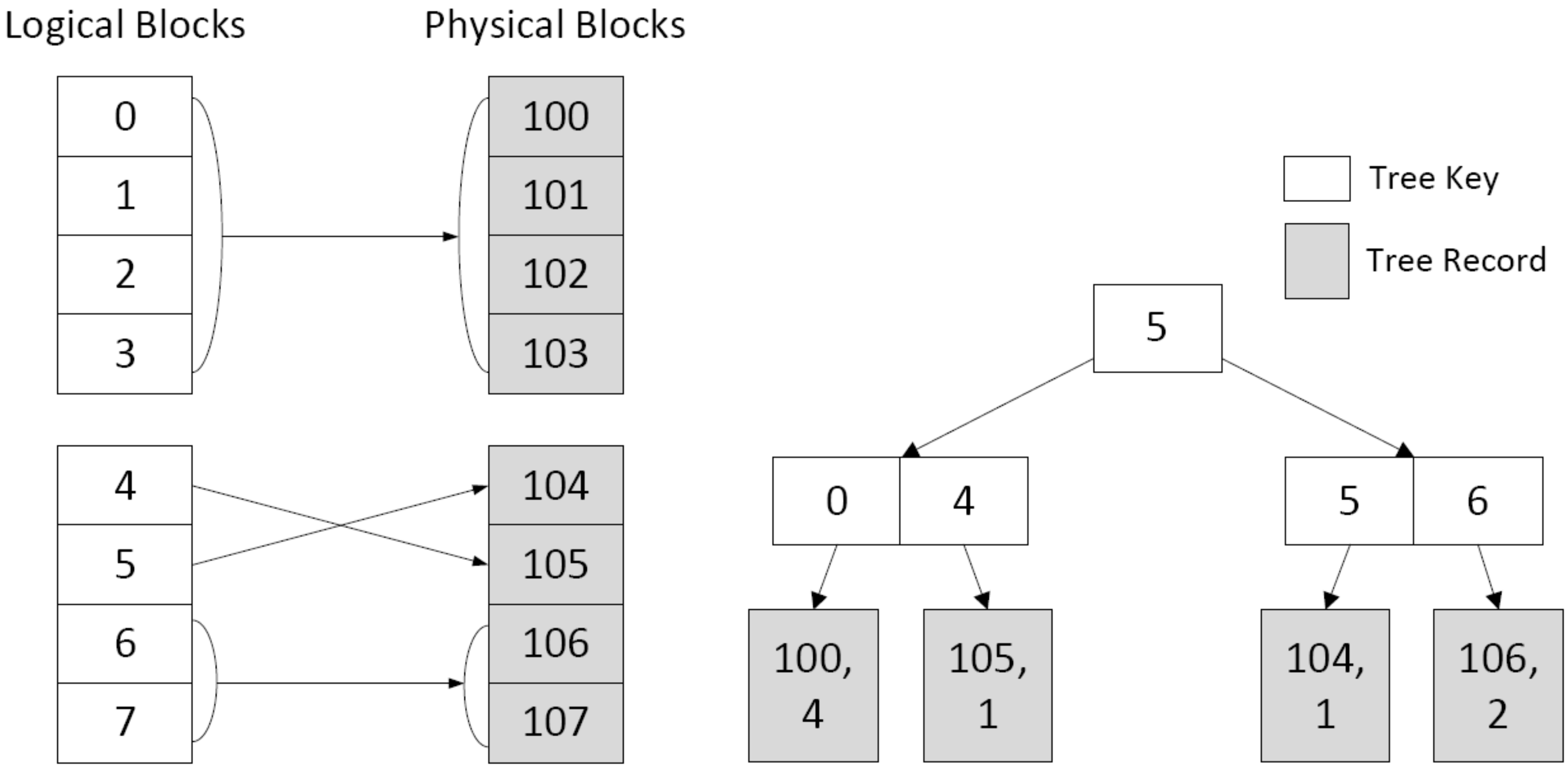
(a) FAST

(b) DFTL

How to reduce meta-data?

- Log-based FTL
- Region-based FTL
  - u-FTL: A memory-efficient flash translation layer supporting multiple mapping granularities. EMSOFT'08.
- Multi-level Mapping
  - An adaptive two-level management for the flash translation layer in embedded systems. ICCAD’06.
  - Efficient management for large-scale flash-memory storage systems with resource conservation. SAC’04.
  - Buffer Flush and Address Mapping Scheme for Flash Memory Solid State Disk, JSA’10
- Meta-data demand loading
  - DFTL: A Flash Translation Layer Employing Demand-based Selective Caching of Page-level Address Mapping, ASPLOS’09

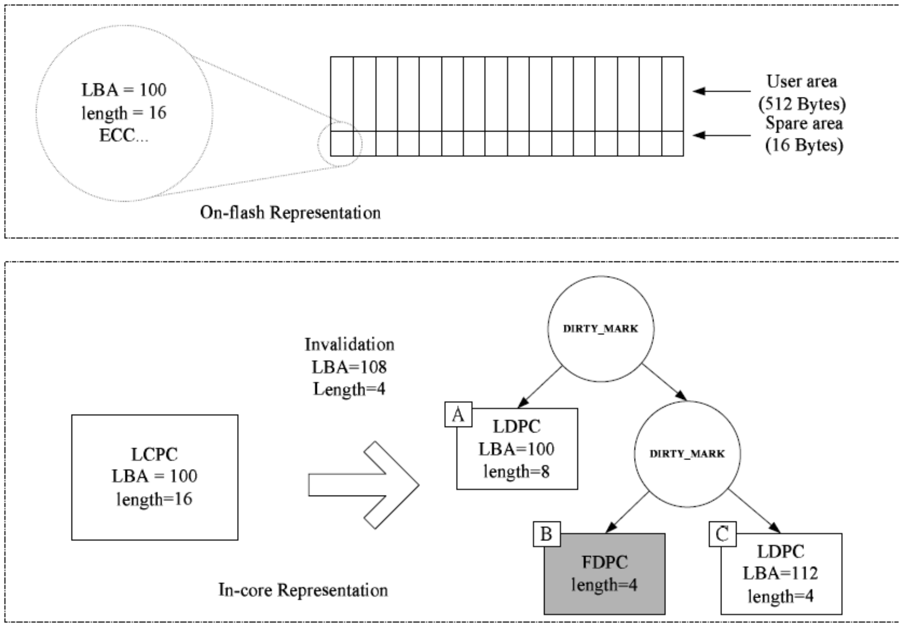
Region-based FTL

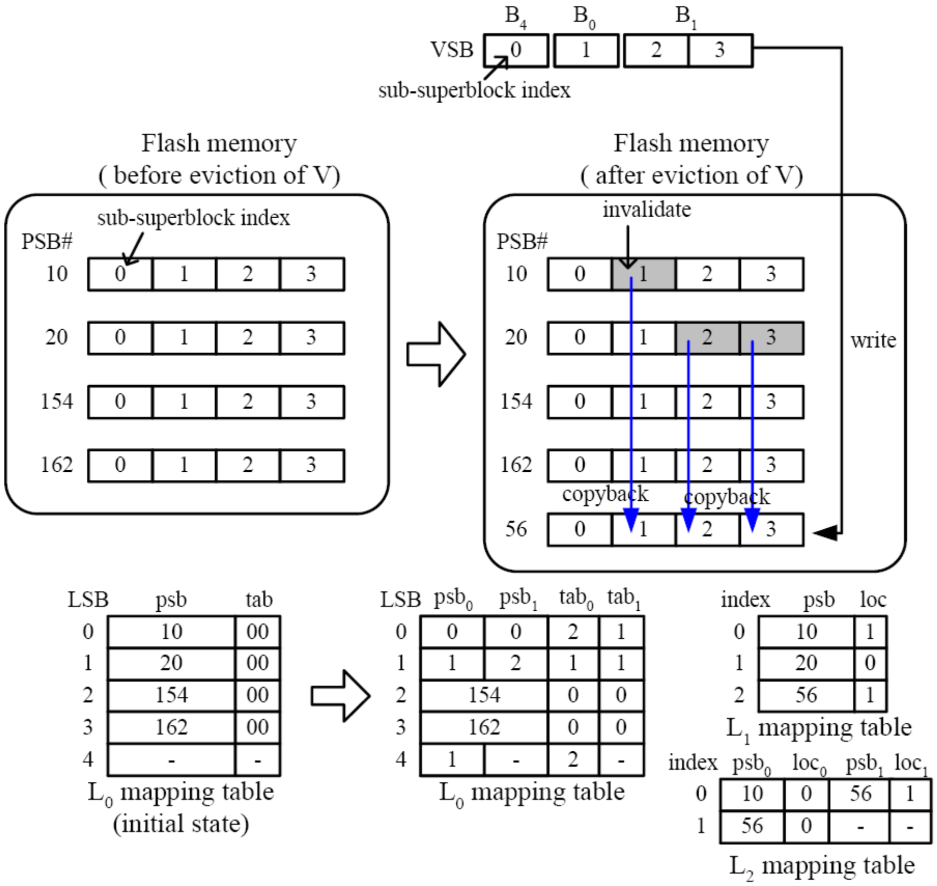


(a) Mapping example

(b) μ-Tree example

Multi-Level FTL





LSB	psb	tab
0	10	00
1	20	00
2	154	00
3	162	00
4	-	-

L_0 mapping table (initial state)

LSB	psb_0	psb_1	tab_0	tab_1
0	0	0	2	1
1	1	2	1	1
2	154		0	0
3	162		0	0
4	1	-	2	-

L_0 mapping table

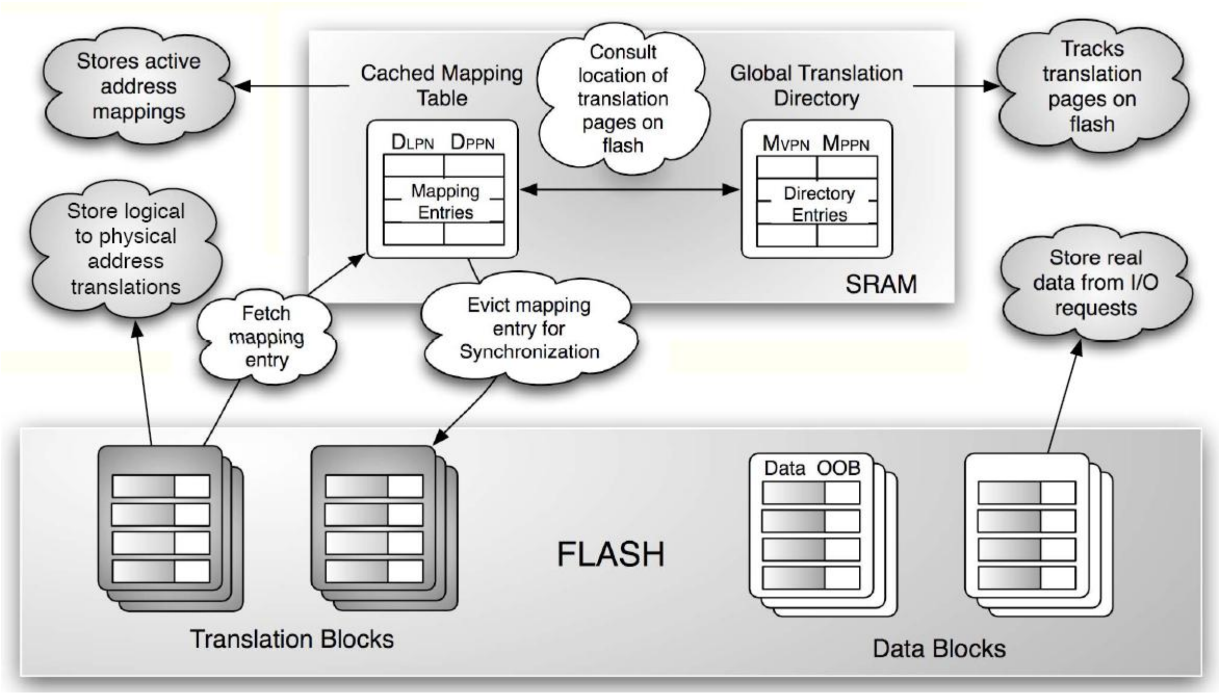
index	psb	loc
0	10	1
1	20	0
2	56	1

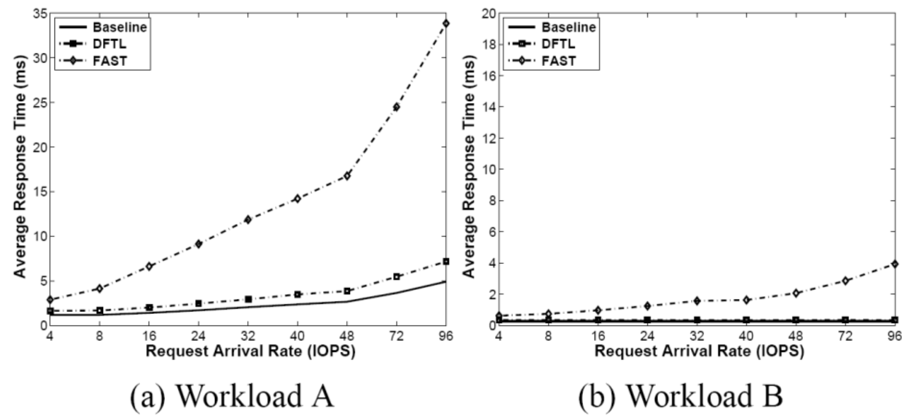
L_1 mapping table

index	psb_0	loc_0	psb_1	loc_1
0	10	0	56	1
1	56	0	-	-

L_2 mapping table

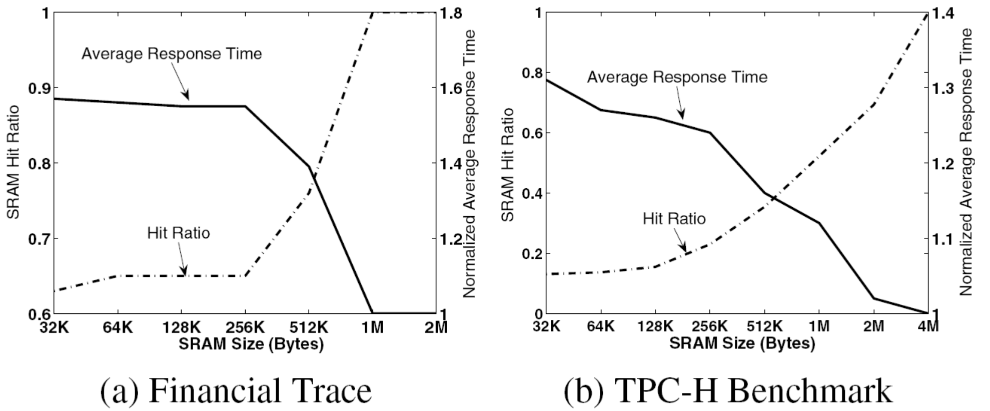
Meta-data demand loading


Stores active address mappings
Cached Mapping Table
Consult location of translation pages on flash
Global Translation Directory
Tracks translation pages on flash
DLPN DPPN
Mapping Entries
MVPN MPPN
Directory Entries
Store logical to physical address translations
SRAM
Store real data from I/O requests
Fetch mapping entry
Evict mapping entry for Synchronization
FLASH
Data OOB
Translation Blocks
Data Blocks



(a) Workload A

(b) Workload B



(a) Financial Trace

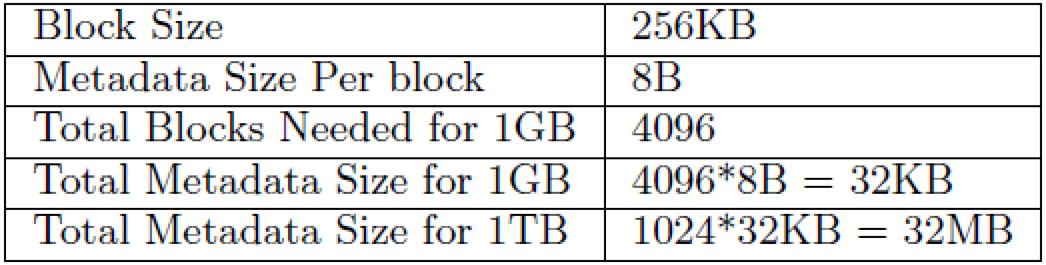
(b) TPC-H Benchmark

Power-Off Recovery

- Whenever the mapping information in SRAM is changed, it should be written at flash memory against sudden power-off.
- Otherwise, full scanning for flash device at booting time.
- How we can reduce the scan time?
  - Check point

Sub-sampling

- All the current garbage collection algorithms use score-based heuristics to select a victim block for reclaiming free space and wear-leveling.(utilization, age, erase count)



Block Size	256KB
Metadata Size Per block	8B
Total Blocks Needed for 1GB	4096
Total Metadata Size for 1GB	4096*8B = 32KB
Total Metadata Size for 1TB	1024*32KB = 32MB

Table 1: Metadata Size Estimation

- Store metadata only for N number of randomly selected blocks, where N << K (K is total number of blocks)
- Group-based wear leveling algorithm