Flash Translation Layers III

Jin-Soo Kim (*jinsookim@skku.edu*)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu

Contents

- Superblock FTL
- DFTL
- μ-FTL

Superblock FTL

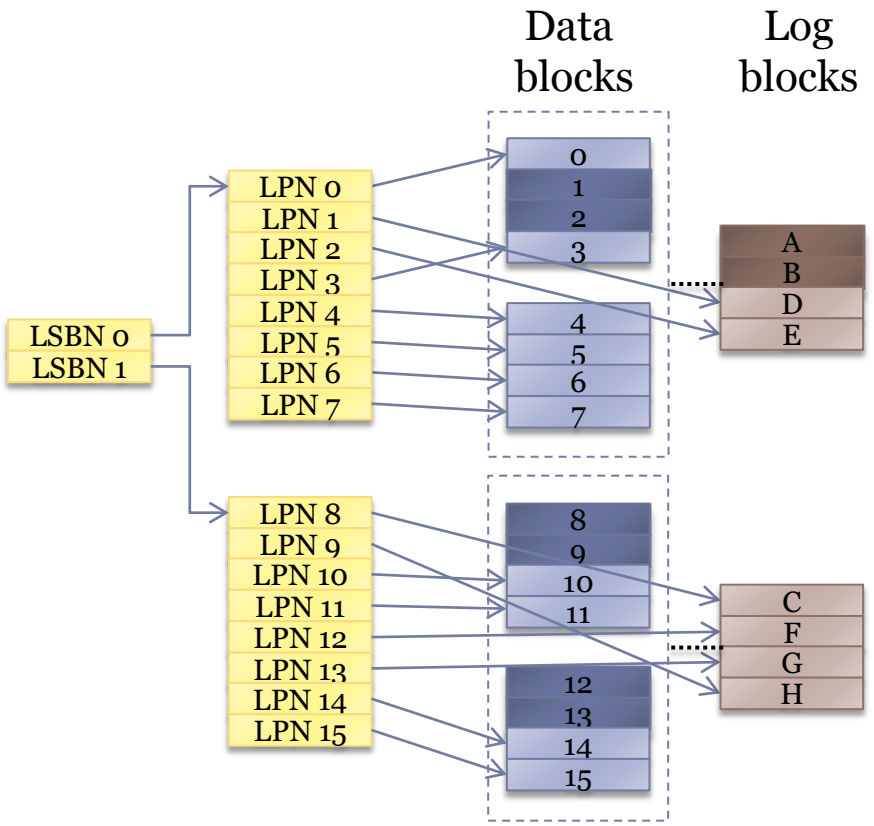
Reference: Superblock FTL: A Superblock-based Flash Translation Layer with a Hybrid Address Translation Scheme

Superblock FTL

- A superblock shares log blocks
- Up to M log blocks per superblock
- Block mapping at the superblock level, Page mapping within a superblock
- Hot/cold pages separation
- Map cache
- The amount of mapping information increased

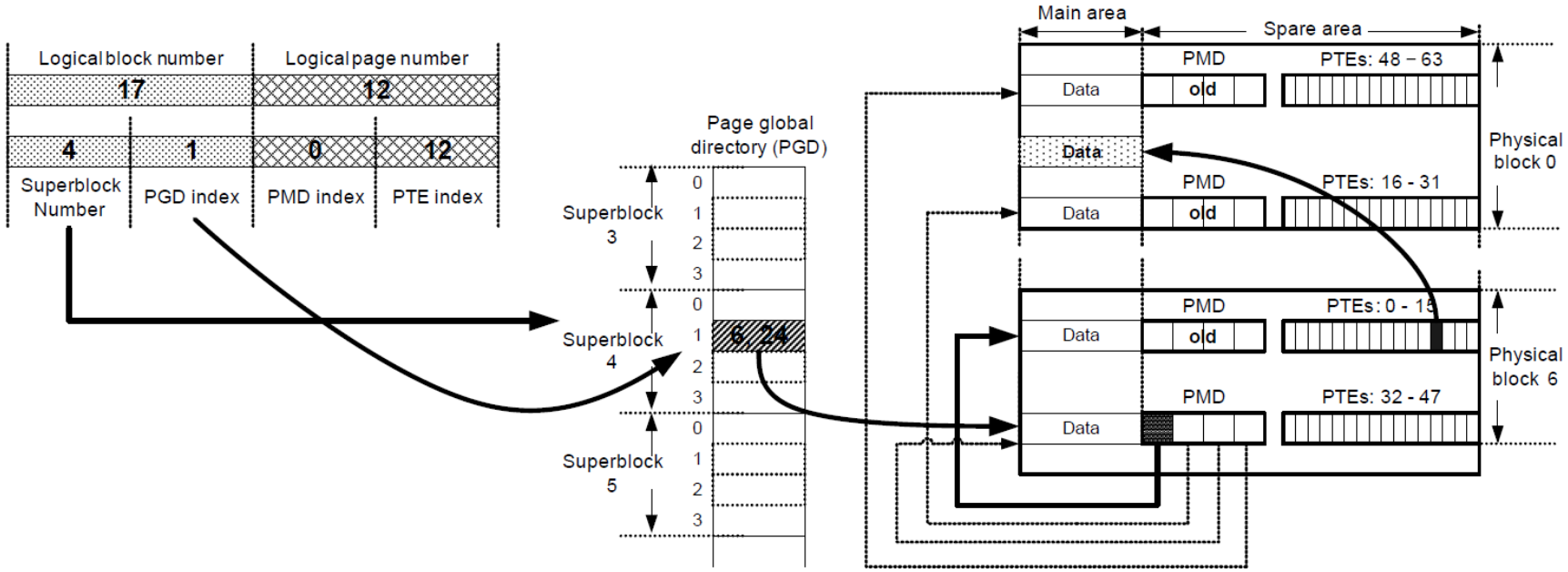
Example

- W = <{1,2}, {8}, {1,2}, {12}, {13}, {9}>
  - Write ({1,2}, AB)
  - Write ({8}, C)
  - Write ({1,2}, DE)
  - Write ({12}, F)
  - Write ({13}, G)
  - Write ({9}, H)



Mapping Information

- PGD (Page Global Directory)
- PMD (Page Middle Directory)
- PTE (Page Table Entry)



Comparison

		ANAND	Log block scheme	FAST	Superblock FTL	Page mapping
Data blocks	Terminology	Data blocks	Data blocks	Data blocks	D-blocks	Data blocks
	Management scheme	In-order	In-order	In-order	Out-of-order	Out-of-order
	The degree of sharing	1	1	1	Min(P, S)	P
Update blocks	Terminology	Replacement blocks	Log blocks	Sequential / random log blocks	U-blocks	Update blocks
	Management scheme	In-order	Out-of-order	Out-of-order	Out-of-order	Out-of-order
	The degree of sharing	1	1	1 (sequential) or P (random)	Min(P, S)	P

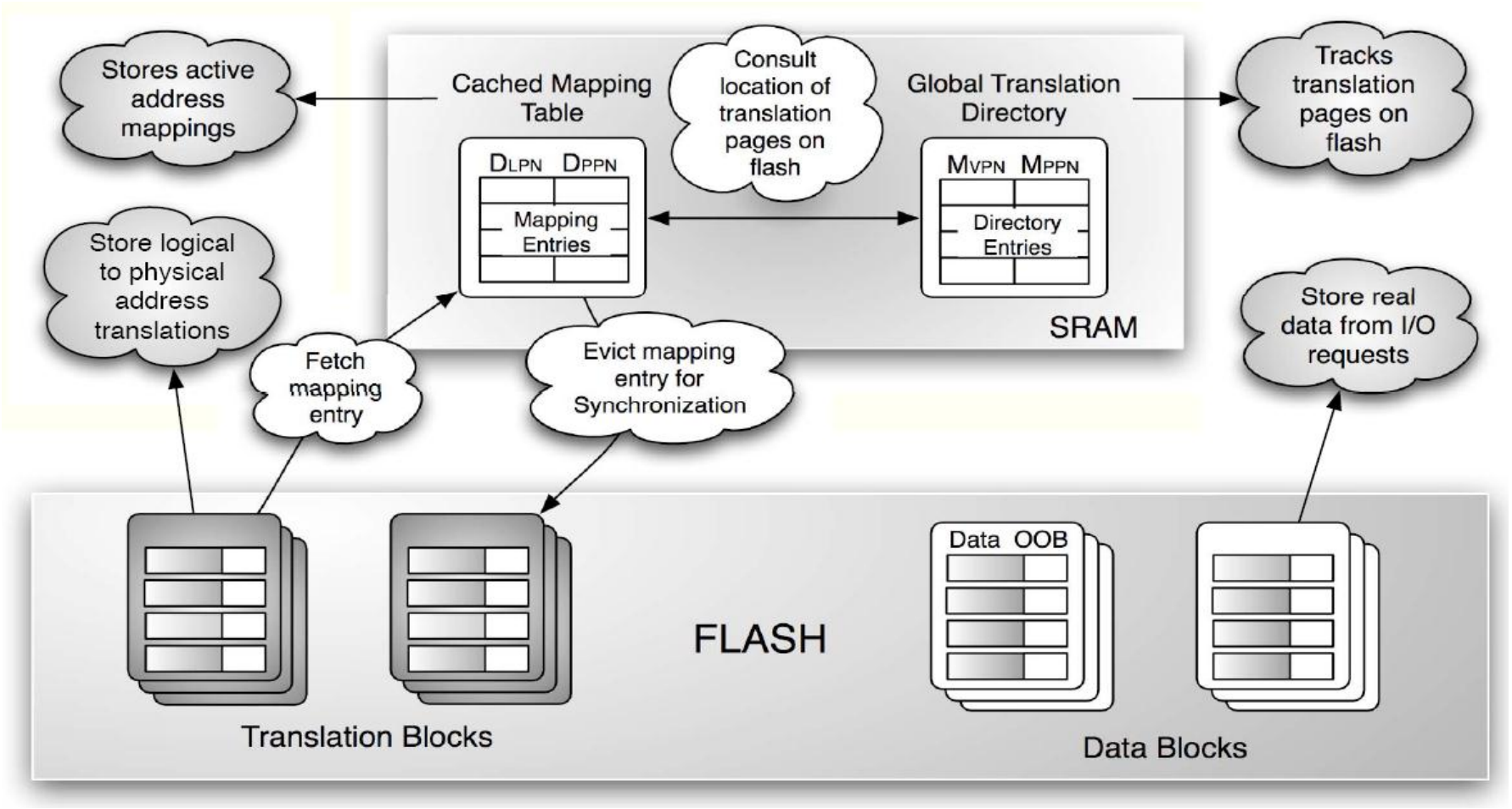
DFTL

Reference: DFTL: A Flash Translation Layer Employing Demand-based Selective Caching of Page-level Address Mappings

DFTL

- Page mapping
- Keeps full mapping on flash to reduce SRAM use
- Demand-based selective caching of page-level address mappings in SRAM
- Exploits temporal locality in workloads
- Data pages vs. Translation pages

DFTL Architecture



Garbage Collection

- Current data block: updated data pages
- Current translation block: updated translation pages

- GC invoked if the number of free blocks < GC_threshold
- Selecting a victim block: simple cost-benefit policy

μ-FTL

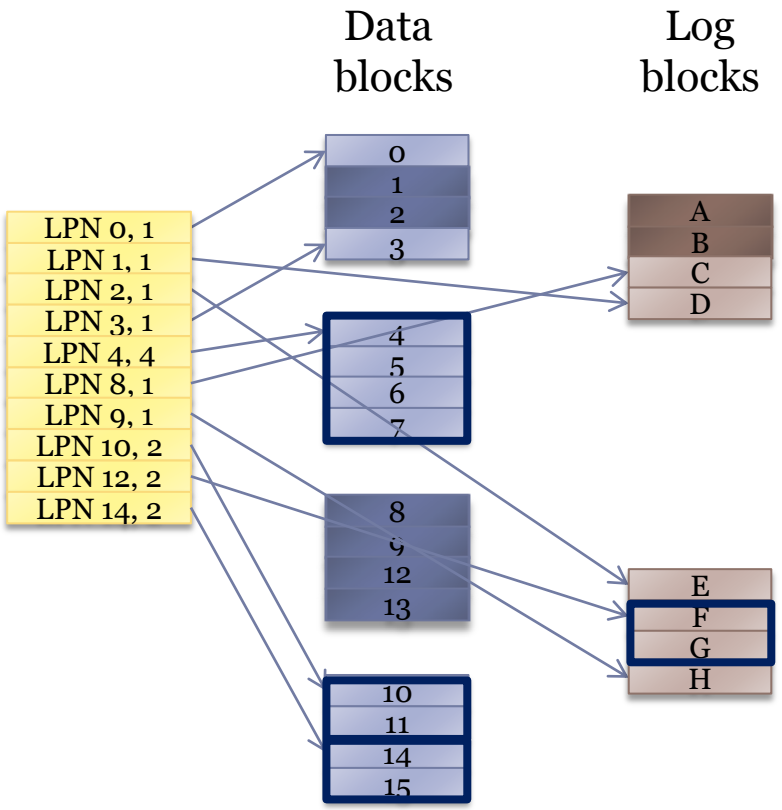
Reference: μ-FTL: A Memory-efficient Flash Translation Layer Supporting Multiple Mapping Granularities

μ-FTL

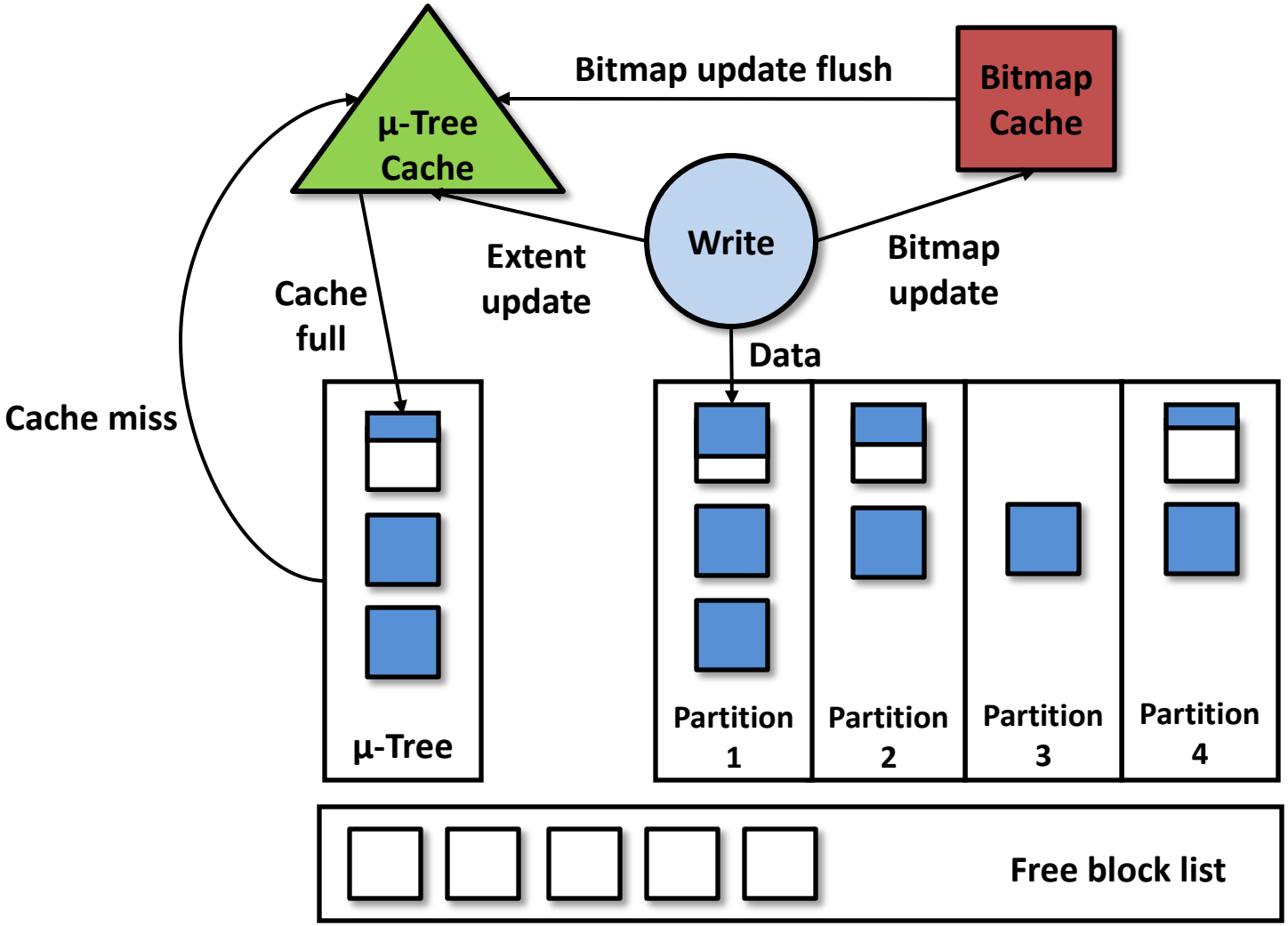
- Minimally-updated FTL
- Supports multiple mapping granularities
  - Based on extents
  - Reduce the amount of mapping information
- Requires more sophisticated index structure
  - μ-Tree is used to store the mapping info.
- Tunable memory footprint
  - Frequently accessed mapping info. cached

Example

- W = <{1}, {2}, {8}, {1}, {2}, {12,13}, {9}>
  - write ({1}, A)
  - write ({2}, B)
  - write ({8}, C)
  - write ({1}, D)
  - write ({2}, E)
  - write ({12,13}, FG)
  - write ({9}, H)



µ-FTL Architecture



Conclusion

- Garbage collection issues
- Mapping information management issues
- Platform-dependent issues