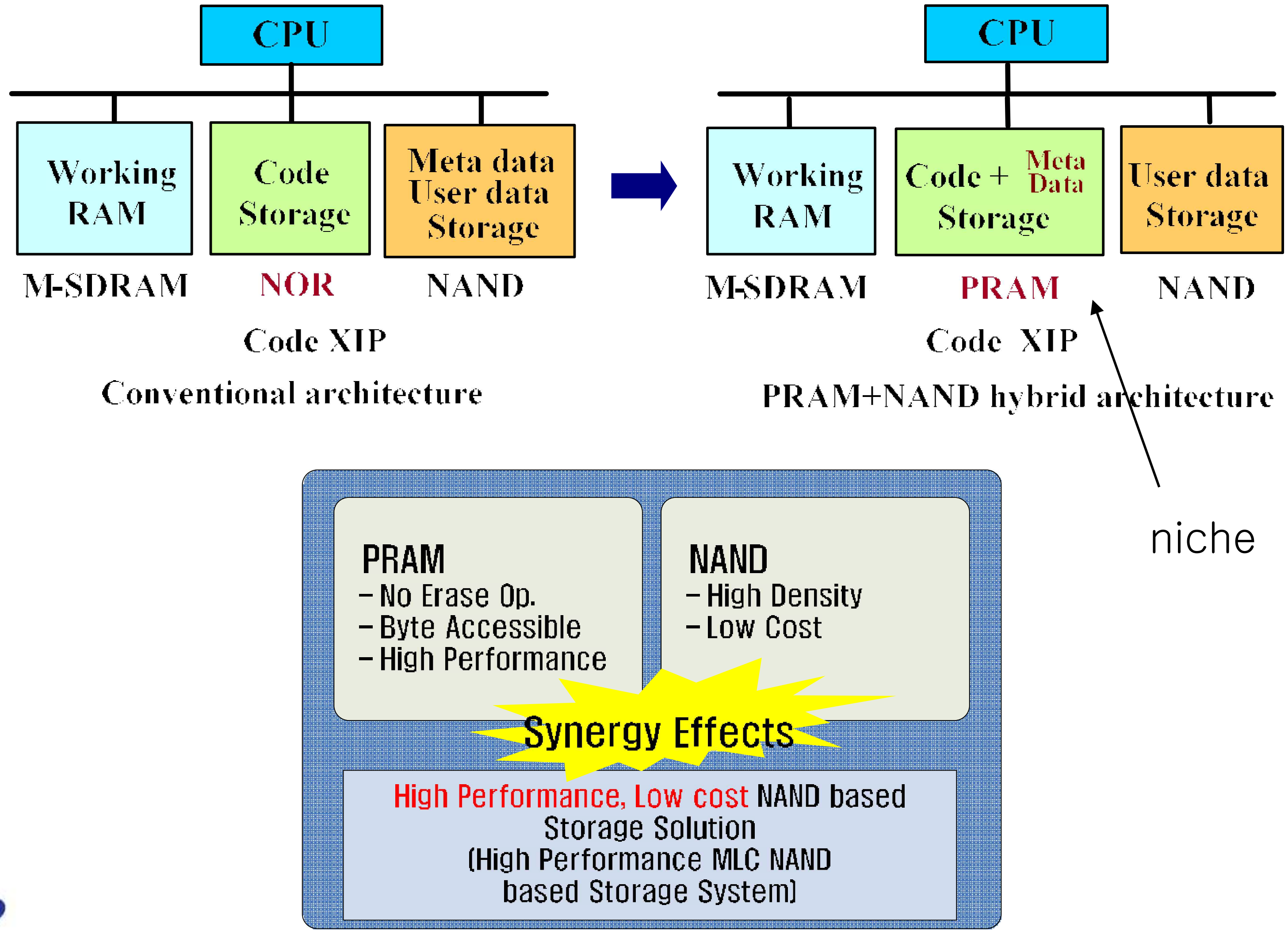
A PRAM and NAND Flash Hybrid Architecture for High-Performance Embedded Storage Subsystems

Chul Lee
Software Laboratory
Samsung Advanced Institute of Technology
Samsung Electronics

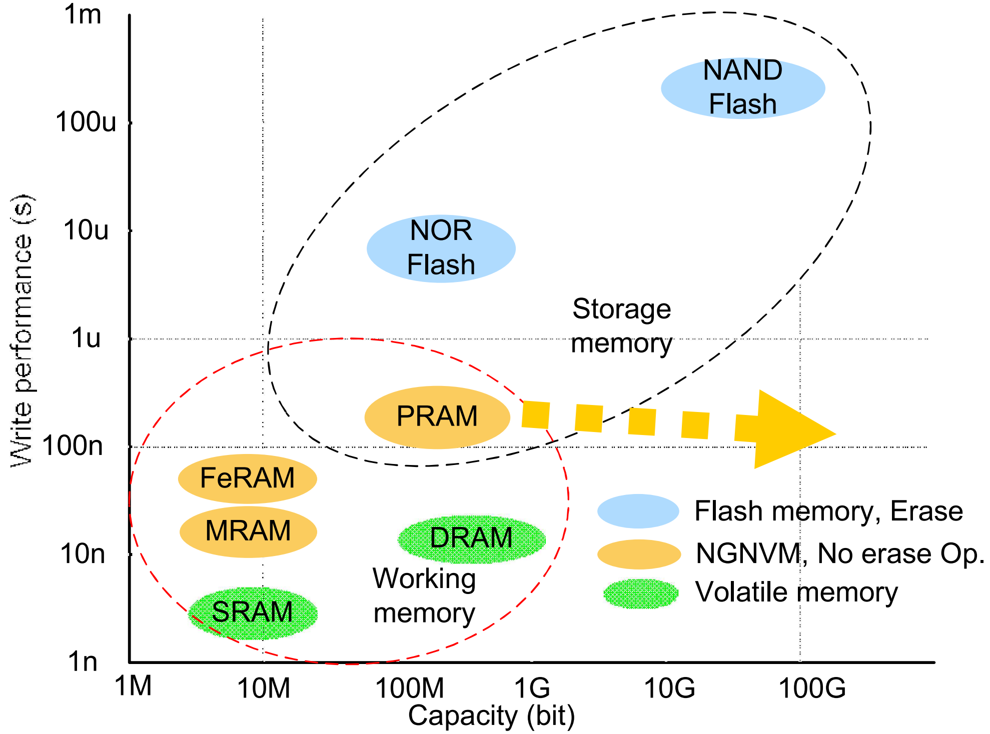
Outline

- Background
- Motivation
- Our methods
  1. File System Metadata Separation
  2. Hybrid FTL
- Evaluation
- Conclusion

Target System



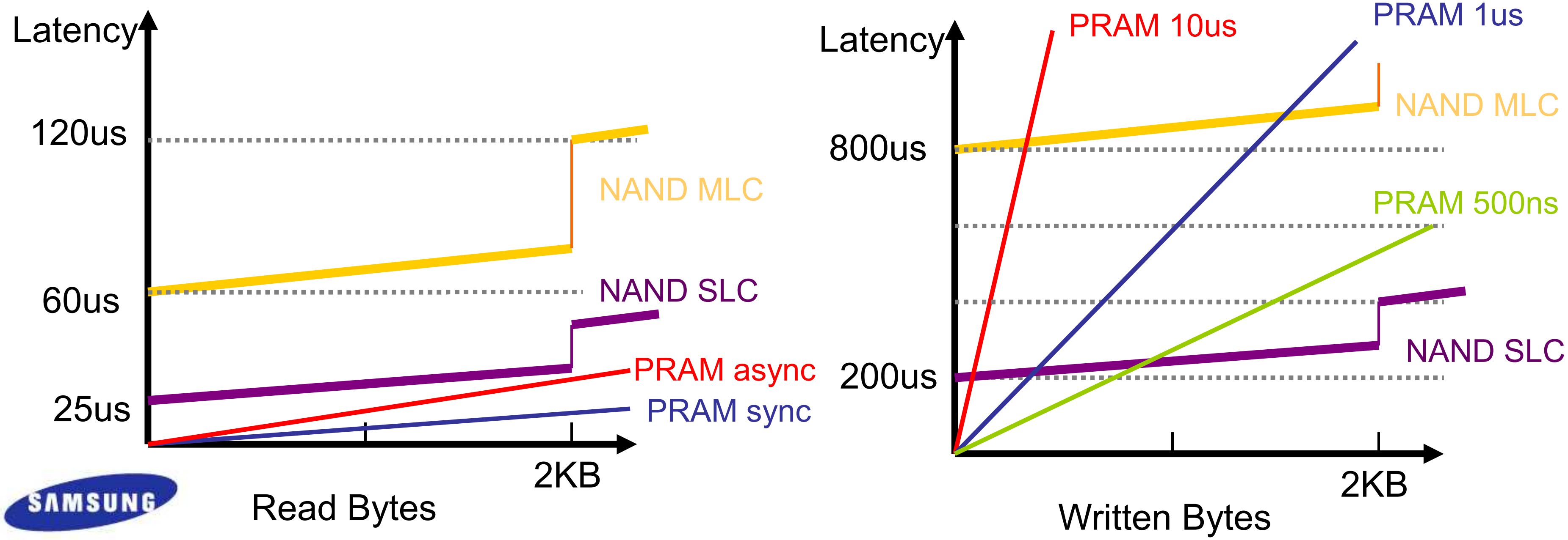
Characteristics of NVRAM (PRAM)



- **Flash memory device**
  - Erase before program
  - Different granularity on erase and program operations
  - Limited life time
  - High Density (NAND)
- **NVRAM**
  - Fast read/write
  - Byte operation (random access)
  - No erase operation
  - Capacity (cost) problem

PRAM vs. NAND

Features	PRAM	NAND(MLC)	NAND(SLC)
Interface	Byte Access No erase op	Page Access Need for Erase	Page Access Need for Erase
Endurance	1M	10K	100K
Density	512M bit	128 G bit	64G bit
Access time	Async: 80ns Sync: 10ns	60us(page) Serial access:30ns	20us(page) Serial access:25ns
Word Program Time	10us (word)	800us(page)	200us(page)
Block Erase Time	N/A	1.4ms	1.4ms



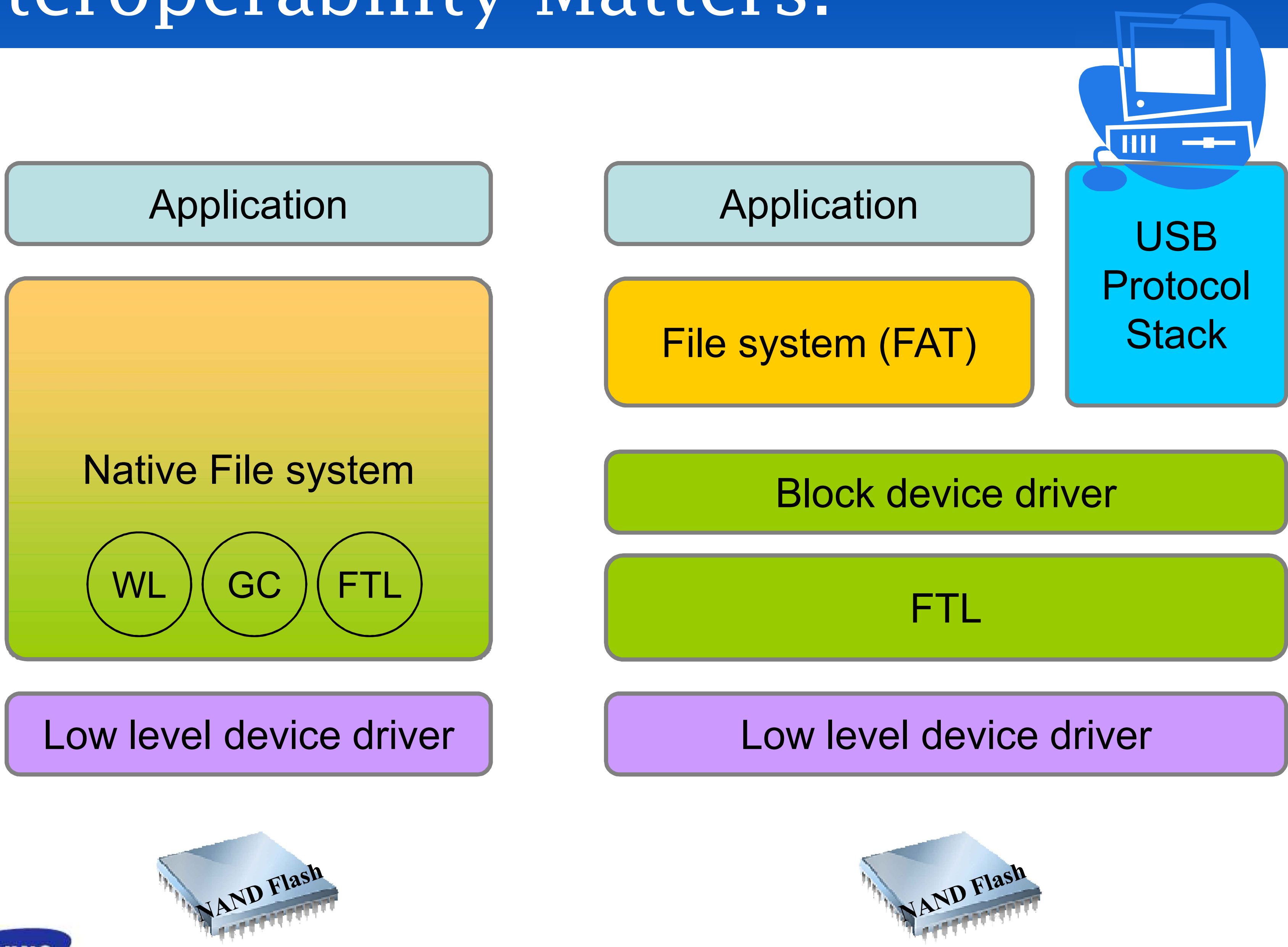
PRAM vs. NAND

	PRAM	NAND
Strength	▪ Byte access ▪ XIP ▪ No erase/in-place update ▪ Low latency ▪ Endurance	▪ Large capacity ▪ Write endurance ▪ Fast Seq. I/O
Weakness	▪ **Density** ▪ **Seq. Write speed**	▪ Erase op. ▪ FTL ▪ Wear-Leveling ▪ Garbage-collection ▪ Page/block access

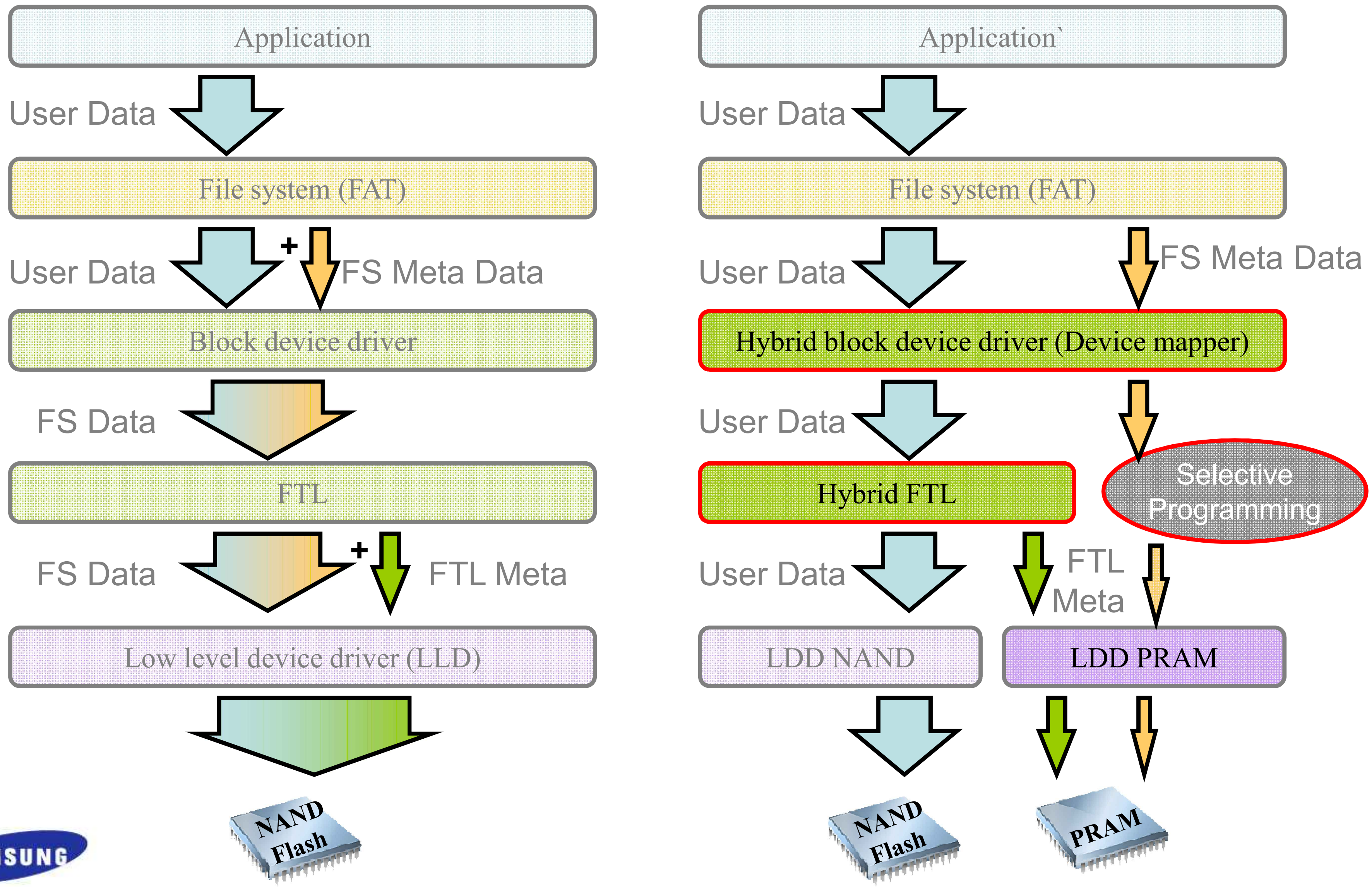
Related Work

	Arch.	Based on	NVRAM Usage	NVRAM Mgmt.	Disk(NAND) Usage	Reliability	Year
HeRMES	MRAM + Disk		Compressed **Metadata** Write buffer				2001
Conquest	BBDRAM + Disk	FFS-like	**Metadata** Small File data Shared Library	Slab/Zone/ Page allocator	Large File data	Soft update Consistency checker	2002
MRAMFS	MRAM only		Compressed **Metadata**				2004
Greenan et. al,	MRAM + Disk	LiFS	**Metadata**	MRAM allocator	File data	Online consistency checker EVENODD(ECC) MMU Page protection (H/W)	2006
PRIMS	MRAM + Disk	LiFS	**Metadata**	MRAM allocator	File data	Online consistency checker EVENODD(ECC) Erasure-encoded Log-structure (S/W)	2007
FRASH	FRAM + NAND	YAFFS	**Metadata** (replication)	RAW	Metadata File data		2007
MiNVFS	FRAM + NAND	YAFFS	**Metadata**	BGET Memory allocator	File data		2007
PFFS	PRAM + NAND	FFS-like	**Metadata**	RAW	File data		2008

Interoperability Matters.



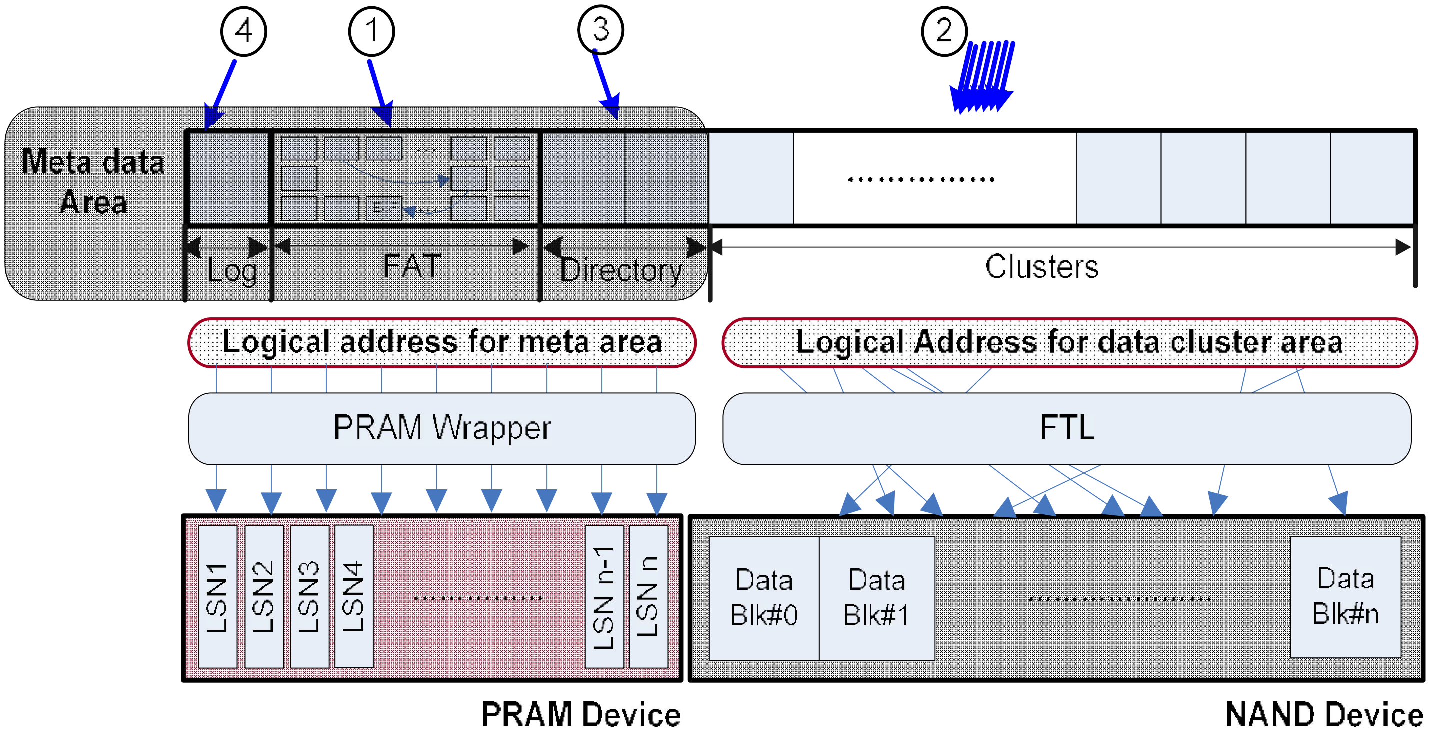
Metadata trickle down to PRAM



Approach in File System Level

FSMS
(File System Metadata Separation)

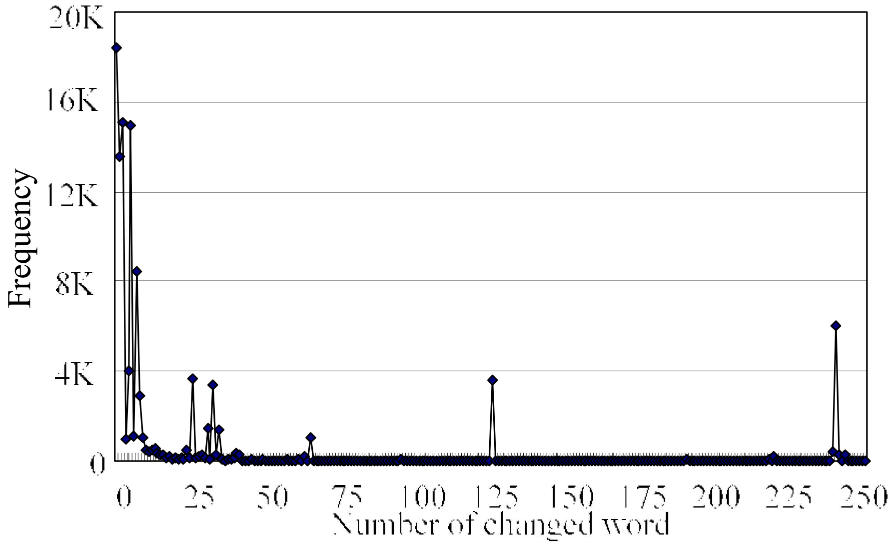
File System Metadata Separation (FSMS)



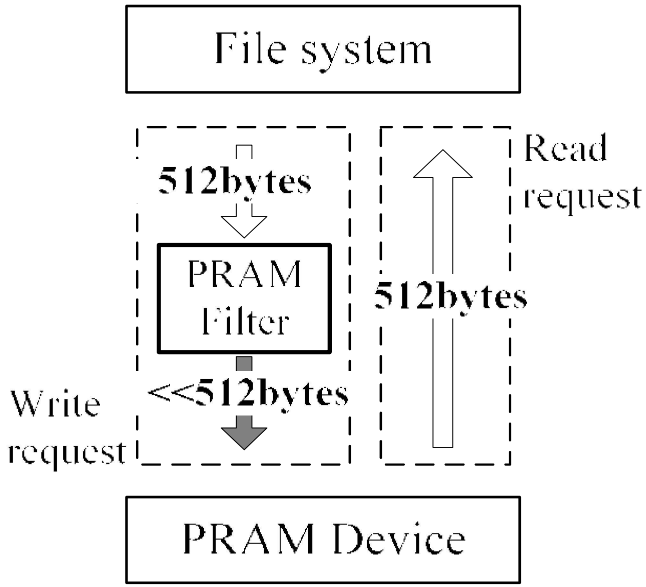
- Separate File System Meta data [FAT,DIR,Log] from NAND Flash
  - Store FS Metadata on PRAM → Decrease random access to FTL
  - Only the user data will be stored on NAND flash
  - Random Access decrement -> FTL overhead decrease -> Performance and life span of NAND flash increase

Selective Programming

- **Selectively programming programs only the modified words among metadata block.**
  - On average, only 16 % of one metadata block is changed



Distribution of changed words in meta data write requests



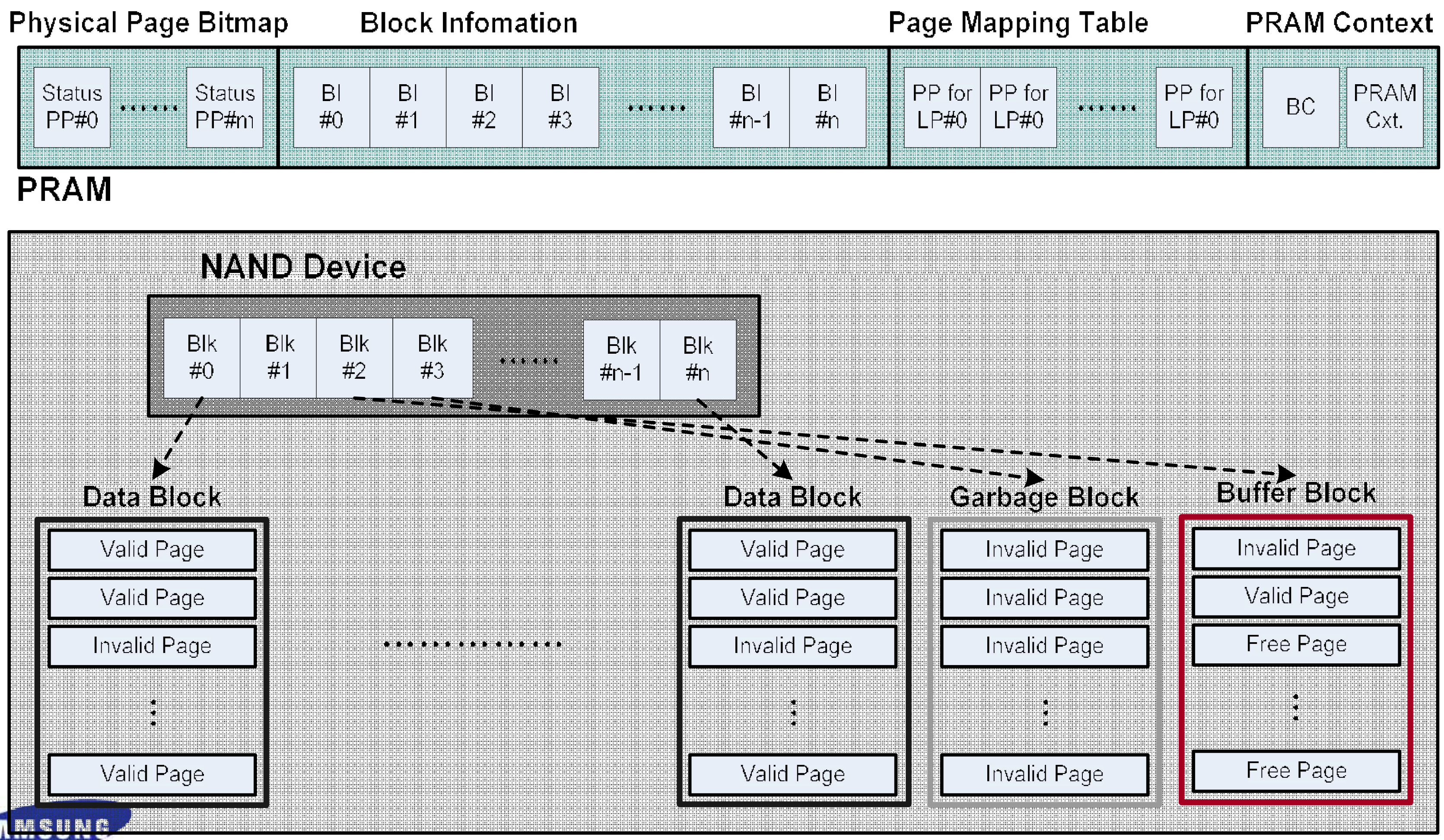
Filtering write request on PRAM

Approach in Block Device Level

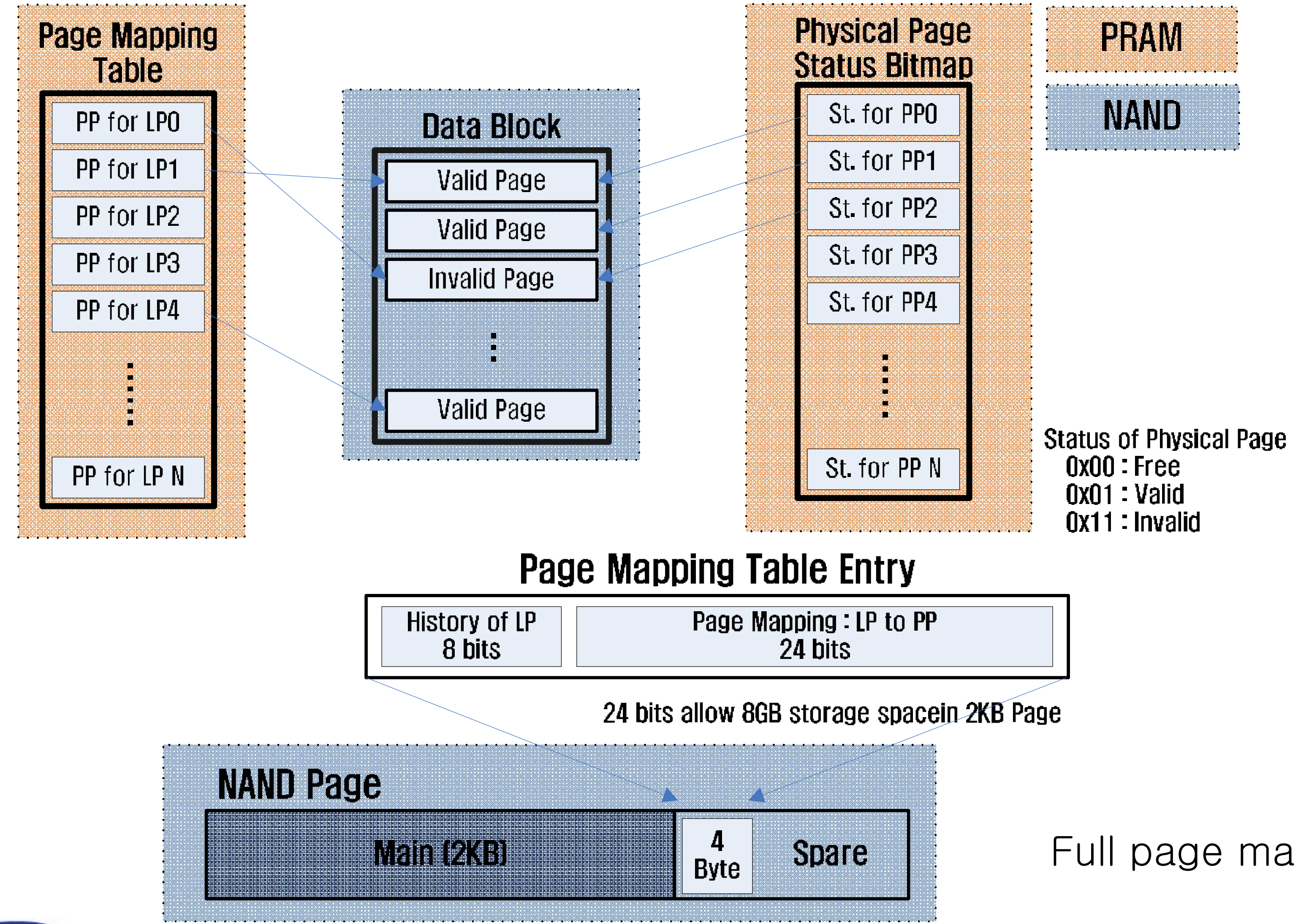
hFTL
(PRAM+NAND hybrid storage FTL)

Implementation of hybrid FTL

- **PRAM, NAND Layout**
  - No scanning is required.



Implementation of hybrid FTL

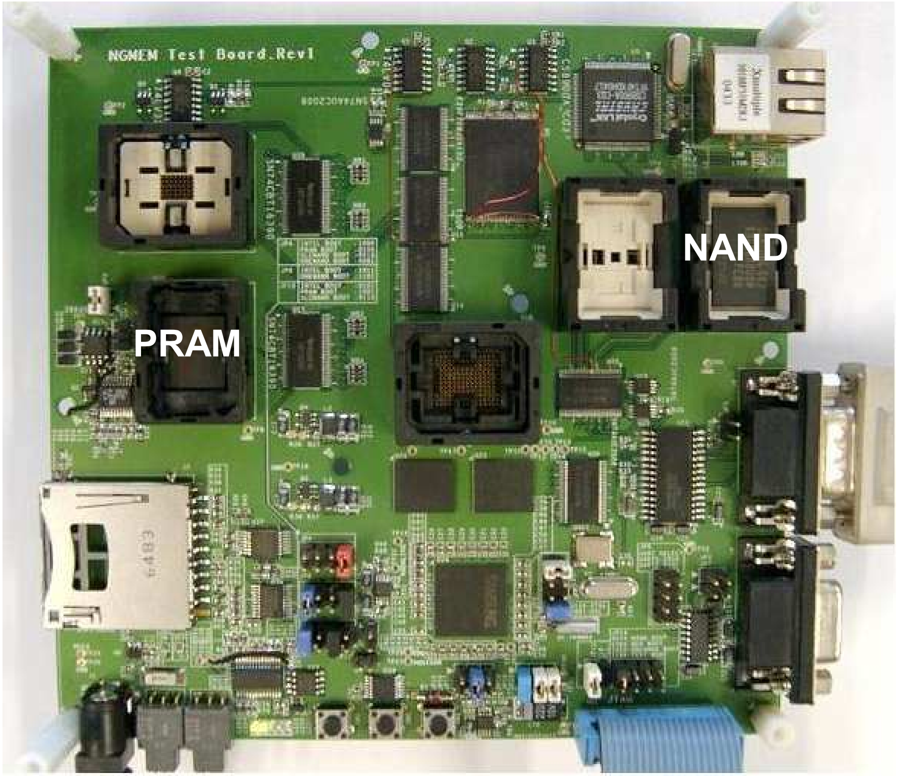


Full page mapping

Evaluation

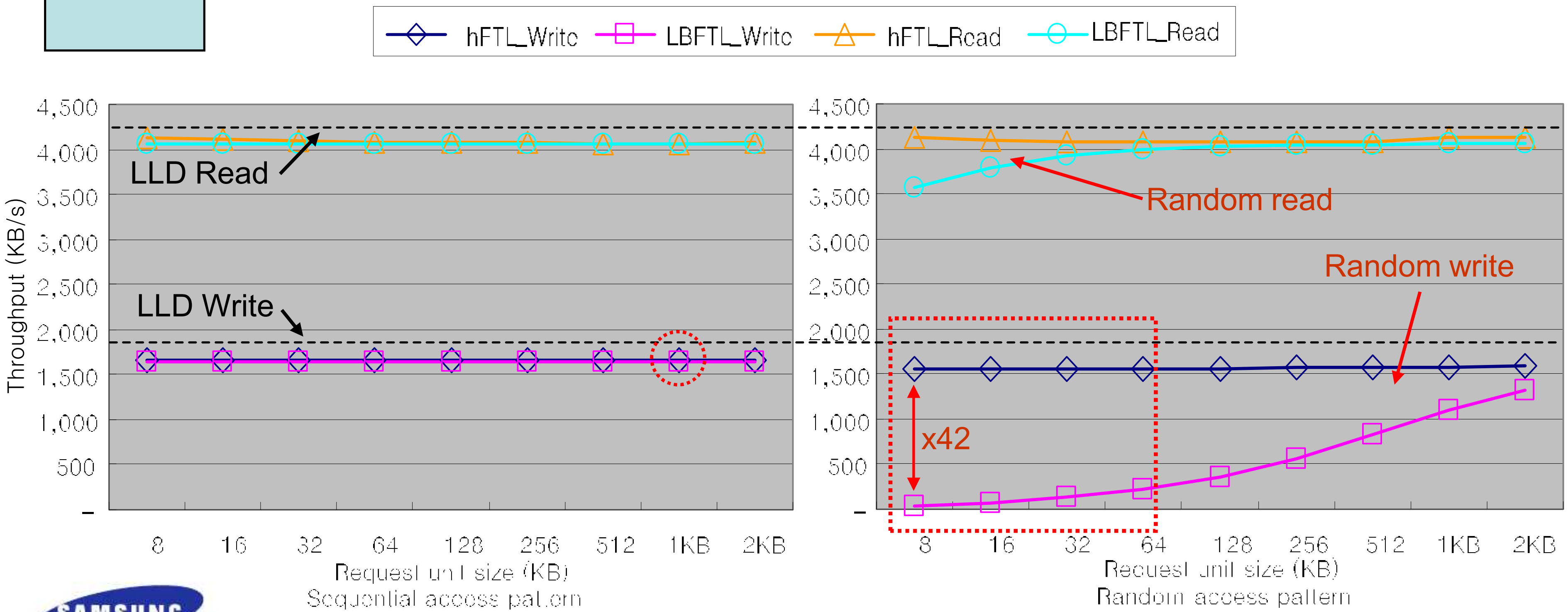
- Workload
  - Synthesized seq./ran. workloads
  - IOzone
- Metrics
  - Throughputs
  - Erase counts for life span
  - DRAM requirement for the implementation cost
- Comparison
  1. Log block FTL (LBFTL)
  2. LBFTL + FSMS,
  3. hFTL,
  4. hFTL+FSMS

Environment	
CPU	S3C2413, 200MHz
PRAM	64MB (KPS1215EZM)
NAND	1GB MLC (K9G8G08U0M)
FS/OS	TFS4/Nucleus RTOS



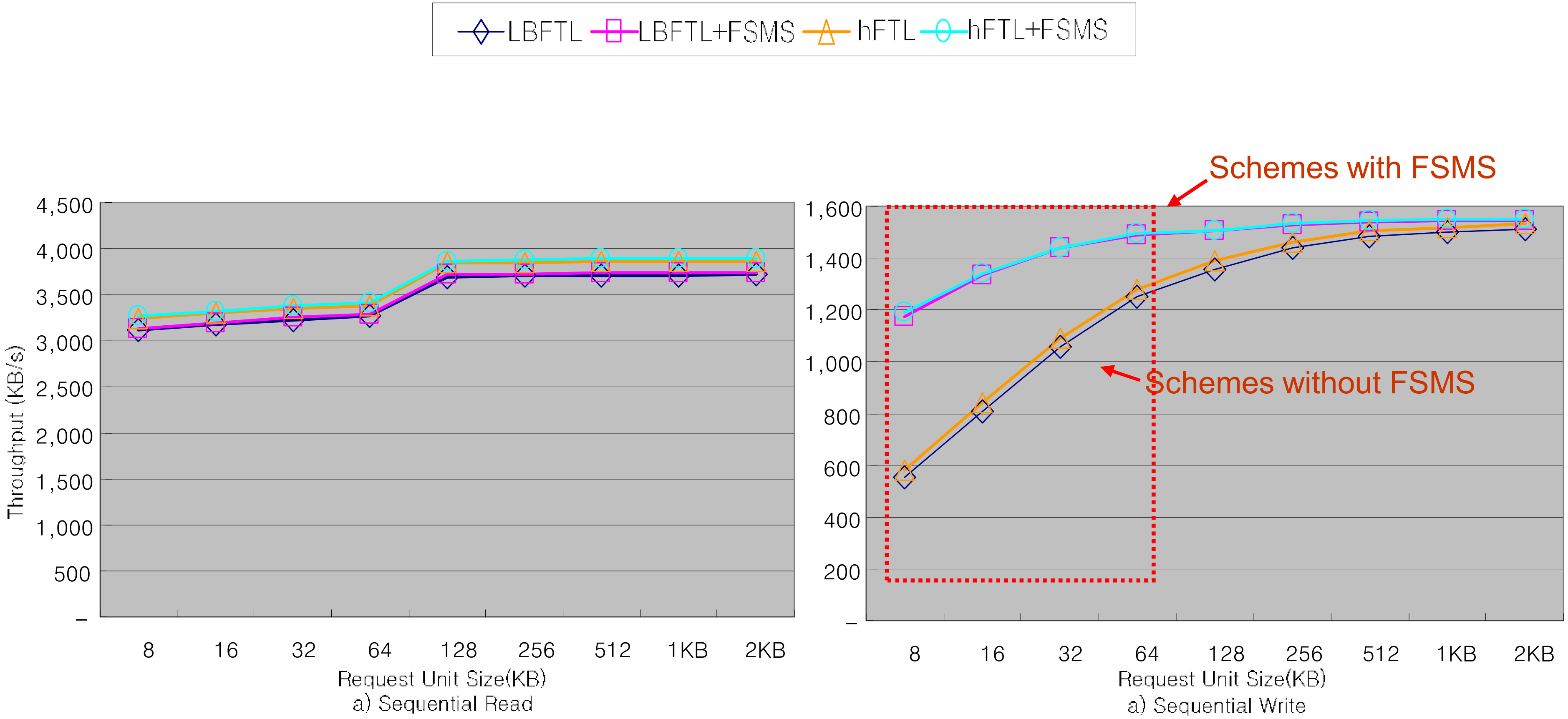
LLD, FTL-Level Evaluation

- LLD (Low Level Device driver)
  - Read: 4.25MB/s, Write 1.7MB/s
- FTL
  - Sequential Read: 4.16MB/s (LB-FTL, hFTL Similar)
  - Random Read: 4.09MB/s (LB-FTL, hFTL Similar)
  - Sequential Write: 1.62MB/s (LB-FTL, hFTL Similar)
  - Random Write :



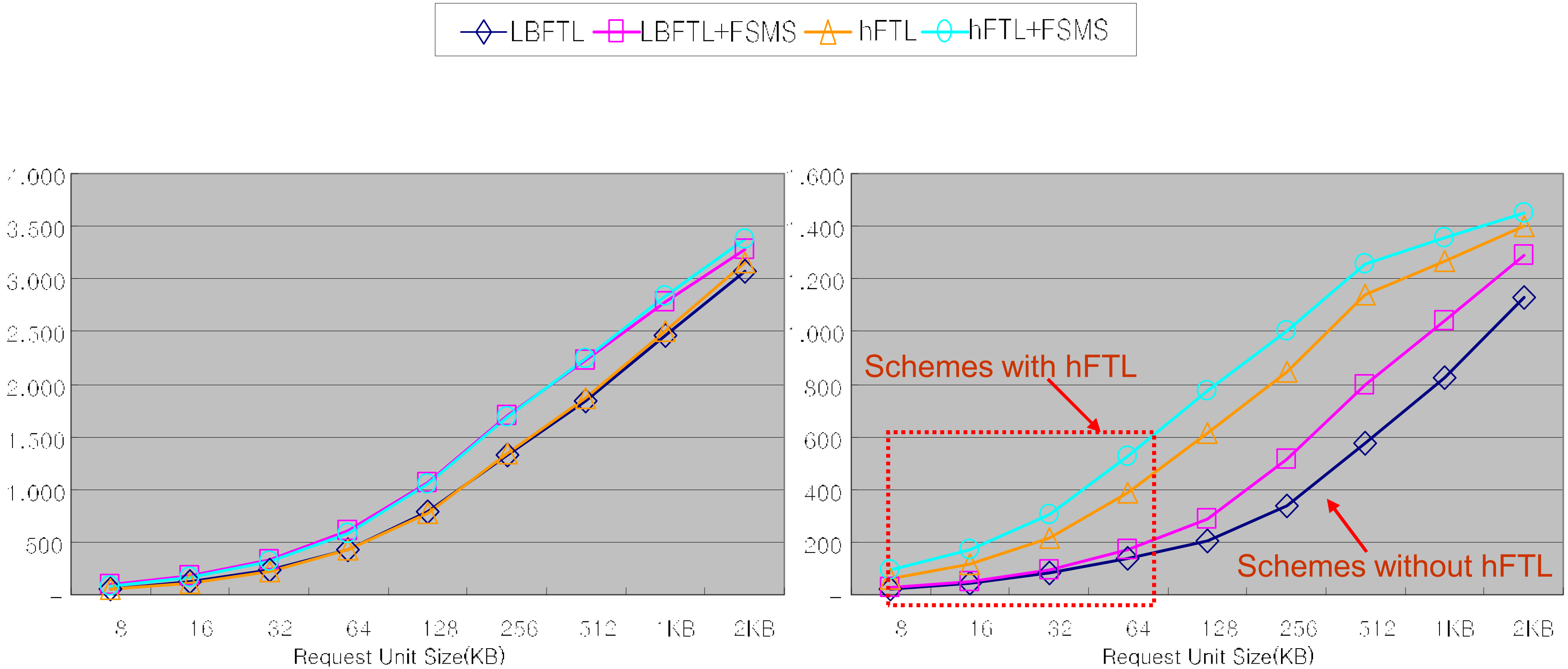
File System Level Evaluations

- IOzone benchmark for Sequential I/O



File System Level Evaluations

- **IOzone benchmark for Random I/O**

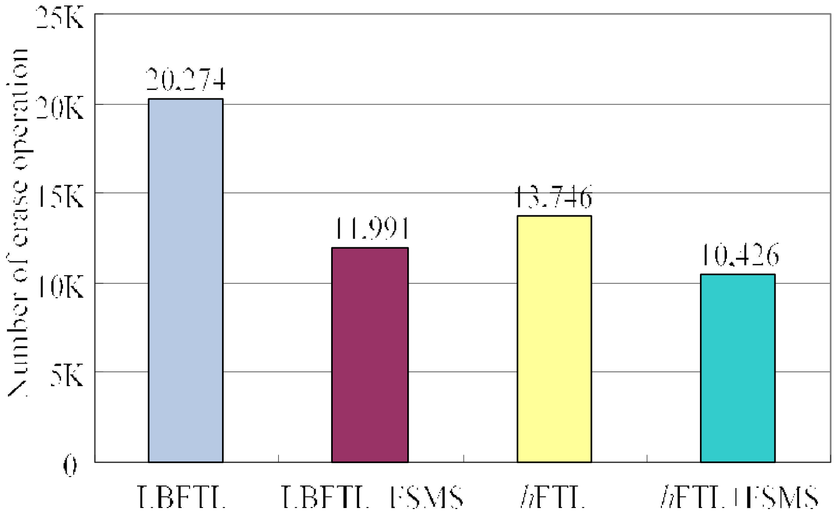


c) Random read

d) Random write

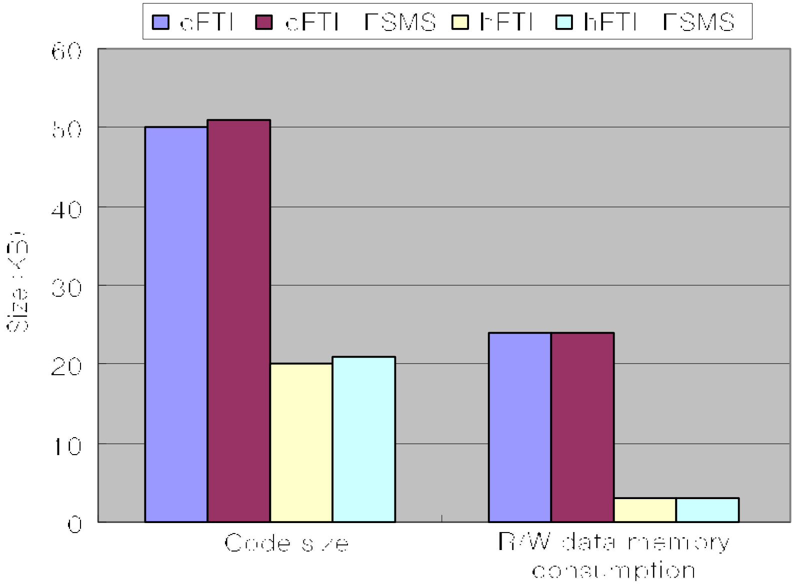
Life Span and Implementation Cost

- **Life Span**



- **Implementation Cost**

Required PRAM for 1GB NAND:
For FSMS : 2 MB
For hFTL : 2.5 MB



Conclusion

- Industrial assumption/viewpoint
  - Embedded systems are mostly handsets and digital gadgets.
  - Storage systems should be FAT-compatible.

- Contributions
  - Two heterogeneous storage memories, PRAM and NAND flash, are combined in a synergetic way.
  - Our two schemes have the metadata trickled down to PRAM instead of NAND.
    - FSMS: The hybrid block device driver locates filesystem's metadata in PRAM
    - hFTL: The hybrid FTL locates all the FTL-related metadata in PRAM. It enabled full-page mapping.

- Further work
  - Scalability for large NAND flash storage
  - PRAM Reliability

References

HeRMES	Miller, E.L. Brandt, S.A. Long, D.D.E. "HeRMES: high-performance reliable MRAM-enabled storage" Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on Publication Date: 20-22 May 2001
Conquest	Wang, A. A., Kuenning, G., Reiher, P., and Popek, G. 2006. The Conquest file system: Better performance through a disk/persistent-RAM hybrid design. Trans. Storage 2, 3 (Aug. 2006), 309-348.
MRAMFS	Nathan K.Edel, Deepa Tuteja, Ethan L.Miller, Scott A.Brandt, MRAMFS: A compressing file system for non-volatile RAM, Preceedings of the The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'04)
Greenan et. al,	Kevin Greenan, Ethan L. Miller, Reliability Mechanisms for File Systems Using Non-Volatile Memory as a Metadata Store, Proceedings of the 6th ACM & IEEE Conference on Embedded Software (EMSOFT '06), October 2006, pages 178-187
PRIMS	Greenan, K. M. and Miller, E. L. 2007. PRIMS: making NVRAM suitable for extremely reliable storage. In Proceedings of the 3rd Workshop on on Hot Topics in System Dependability (Edinburgh, UK). USENIX Association, Berkeley, CA, 10
FRASH	Eun-ki Kim, Hyungjong Shin, Byung-gil Jeon, Seokhee Han, Jaemin Jung and Youjip Won, "FRASH: Hierarchical File System for FRAM and Flash," In proceedings of International Workshop on Data Storage Device and Systems, Aug. 2007
MiNVFS	Doh, I. H., Choi, J., Lee, D., and Noh, S. H. 2007. Exploiting non-volatile RAM to enhance flash file system performance. In Proceedings of the 7th ACM &Amp; IEEE international Conference on Embedded Software (Salzburg, Austria, September 30 - October 03, 2007). EMSOFT '07
PFFS	Park, Y., Lim, S., Lee, C., and Park, K. H. 2008. PFFS: a scalable flash memory file system for the hybrid architecture of phase-change RAM and NAND flash. In Proceedings of the 2008 ACM Symposium on Applied Computing (Fortaleza, Ceara, Brazil, March 16 - 20, 2008). SAC '08

Logical Delete

- **Functionality of "Logical Delete"**