7th IEEE International Workshop on Storage Network Architecture and Parallel I/O
SNAPI 2011 Denver, Colorado – May 25, 2011

Addressing Scalability and Consistency Issues in Hybrid File System for BPRAM and NAND Flash

Embedded Software Systems Lab.

Dept. of Electronics and Computer Engineering
Hanyang University

Quan Taizhong Jinsoo Yoo Jaemin Jung Youjip Won

Outline

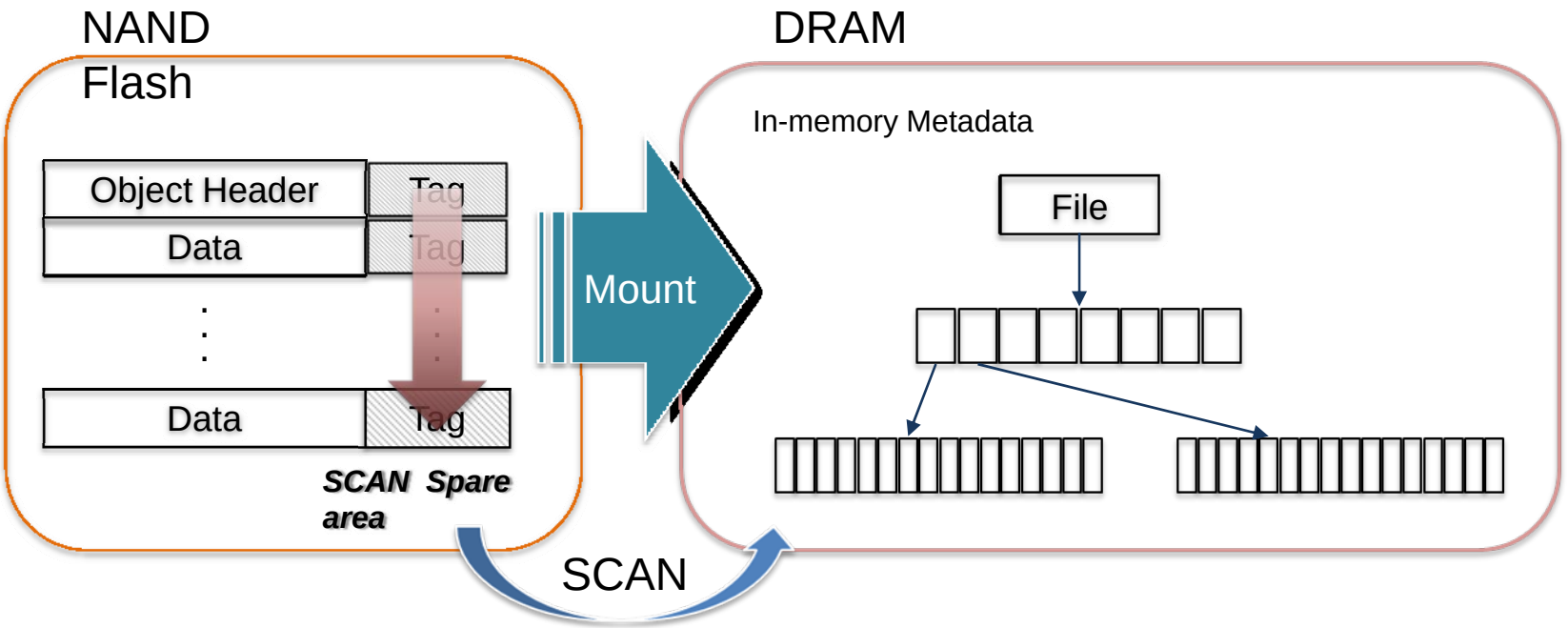
- Background
- Structure of Compressed Metadata File System
- New Functionalities
- Experiment
- Conclusion

Advancement of Byte Addressable RAM(BPRAM)

- BPRAM
  - Byte-addressable
  - Non-volatile
  - In-place update
  - Faster than Flash memory

ITEM	DRAM	FRAM	PRAM	MRAM	NOR	NAND
Centering Byte Addressable	YES	YES	YES	YES	Read only	NO
Non-volatile	NO	YES	YES	YES	YES	YES
Read	10ns	70ns	68ns	35ns	85ns	15us
Write	10ns	70ns	180ns	35ns	6.5us	200us
Erase	none	none	none	none	700ms	2ms
Power consumption	High	Low	High	Low	High	High
Capacity	High	Low	High	Low	High	Very High
Endurance	10^{15}	10^{15}	$> 10^{7}$	10^{15}	100K	100K
Prototype Size		64Mbit	512Mbit	16Mbit		

Issue for File System for NAND Flash



Scan metadata **from NAND Flash** at mount time

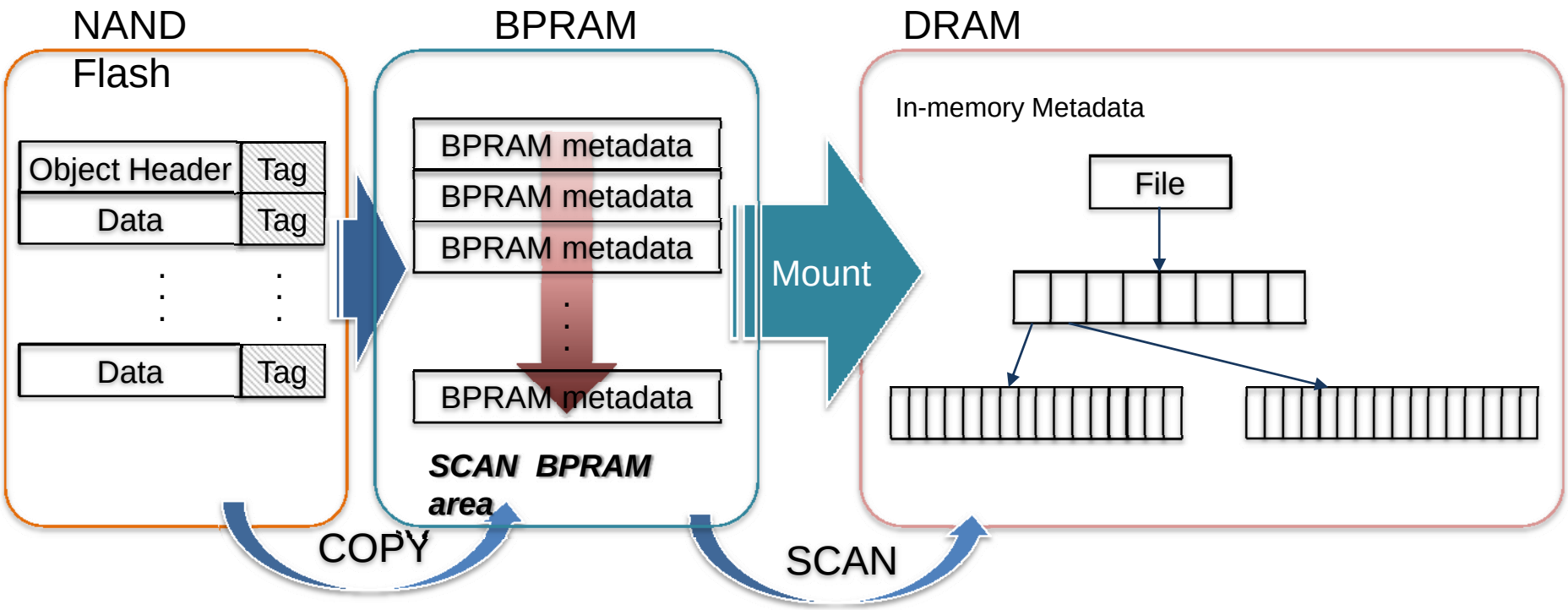
The mount time

Hierarchical File System for BPRAM and NAND Flash

- N.Edel, D.Tuteja, E.miller, and S.Brandt ”MRAMFS“ (2004, MASCOTS)
- S.Park, T.Lee, and K, Chung “RFFS” (2006, ECS)
- J.Jung, Y.Won, E.Kim, H.Shin and B.Jeon ”FRASH” (2010, TOS)
- I.Doh, J.Choi,D.Lee and S.Noh “MiNVFS” (2007,ACM)
- Y.Park, S.Lim,C.Lee, and K.Park “PFFS” (2008,ACM)
- A.Wang, P.Reiher, G.Popek, and G.Kuenning “The Conquest File System” (2002, USENIX)

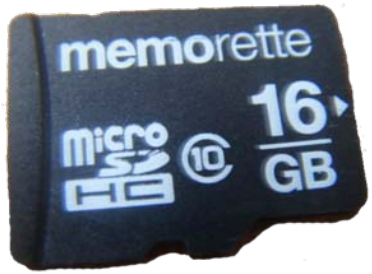
MRAMFS	Hybrid system that stores small files and metadata in MRAM
RFFS	Stores the page data and page metadata separately
FRASH	Use BPRAM as storage of metadata in flash file system. It uses “Copy-On-Mount” technique
MiNVFS	Maintains all the metadata in NVRAM
PFFS	Construct accessed small files in persistent RAMs double indirect index structure for data page management
The Conquest File System	Storing metadata and frequently accessed small files in persistent RAMs double indirect index structure for data page management

Hierarchical File System for BPRAM and NAND Flash

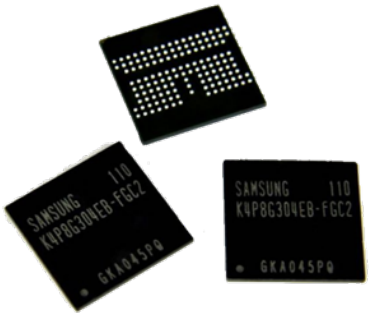


Scalability Issue in Hierarchical File System

Flash memory

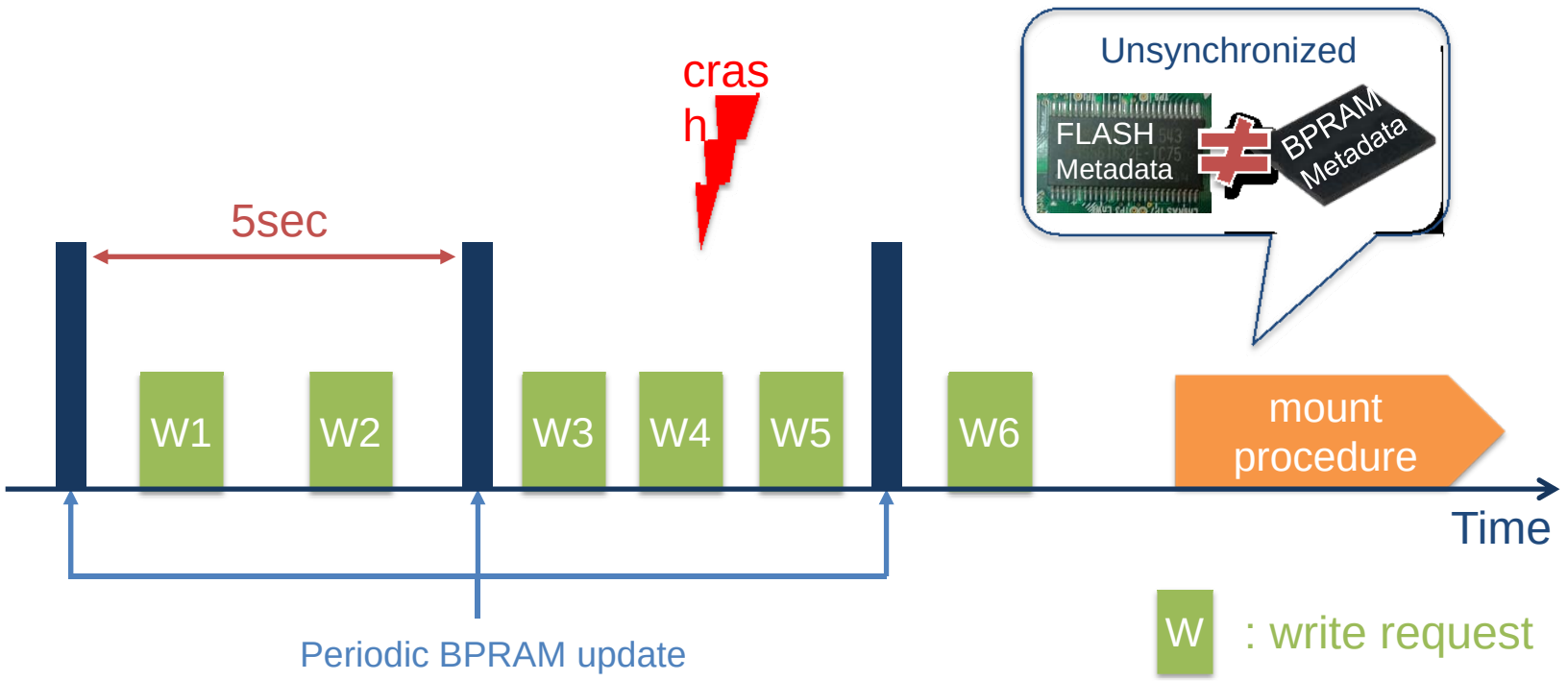


BPRAM

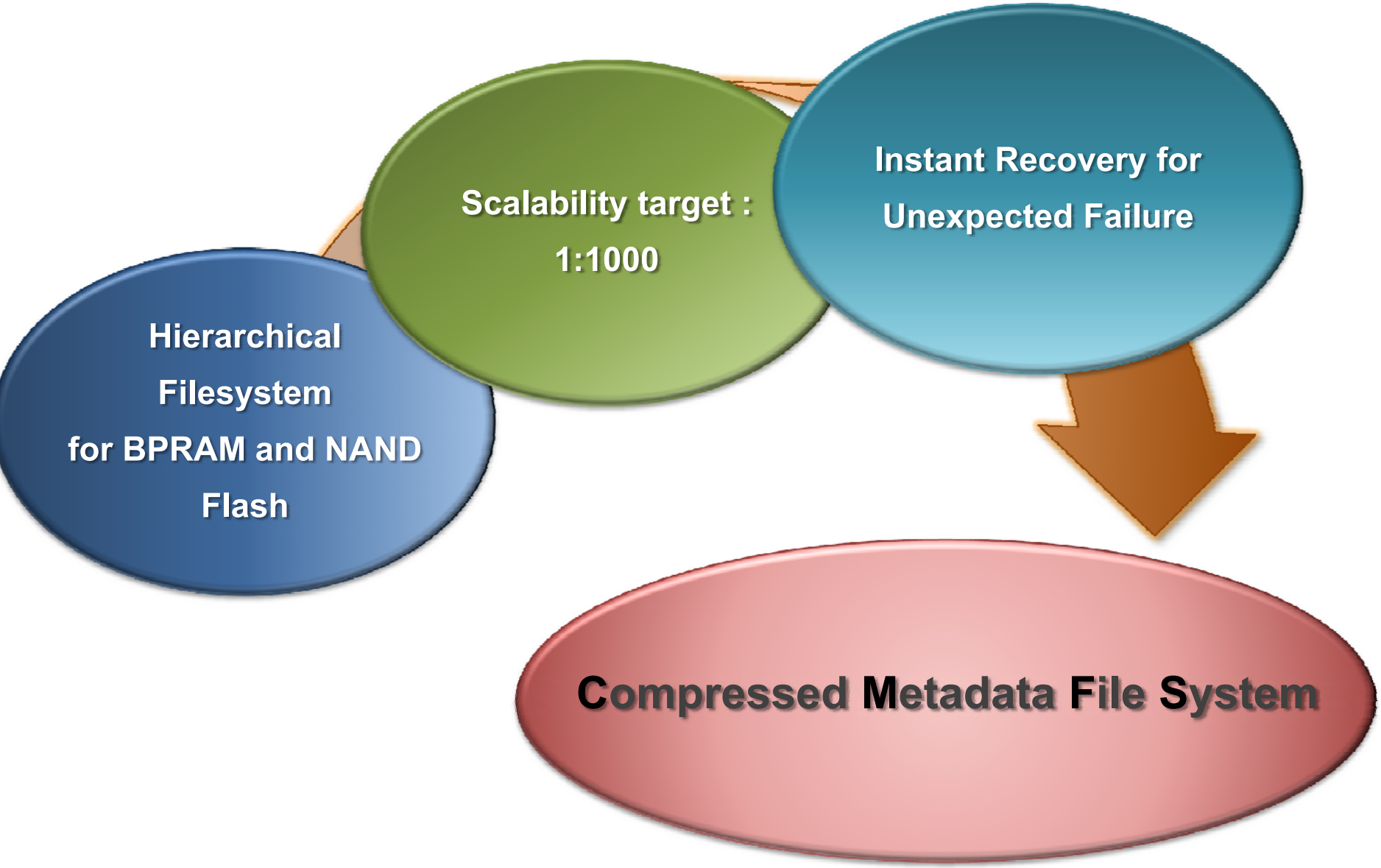


CPM (Cost per MB)	Low	High
Size	~ 64GB or more	~ 64MB
Current Ratio	100 : 3 (NAND : Data 512B , Metadata : 16B)	
Needed Ratio	1000 : 1	

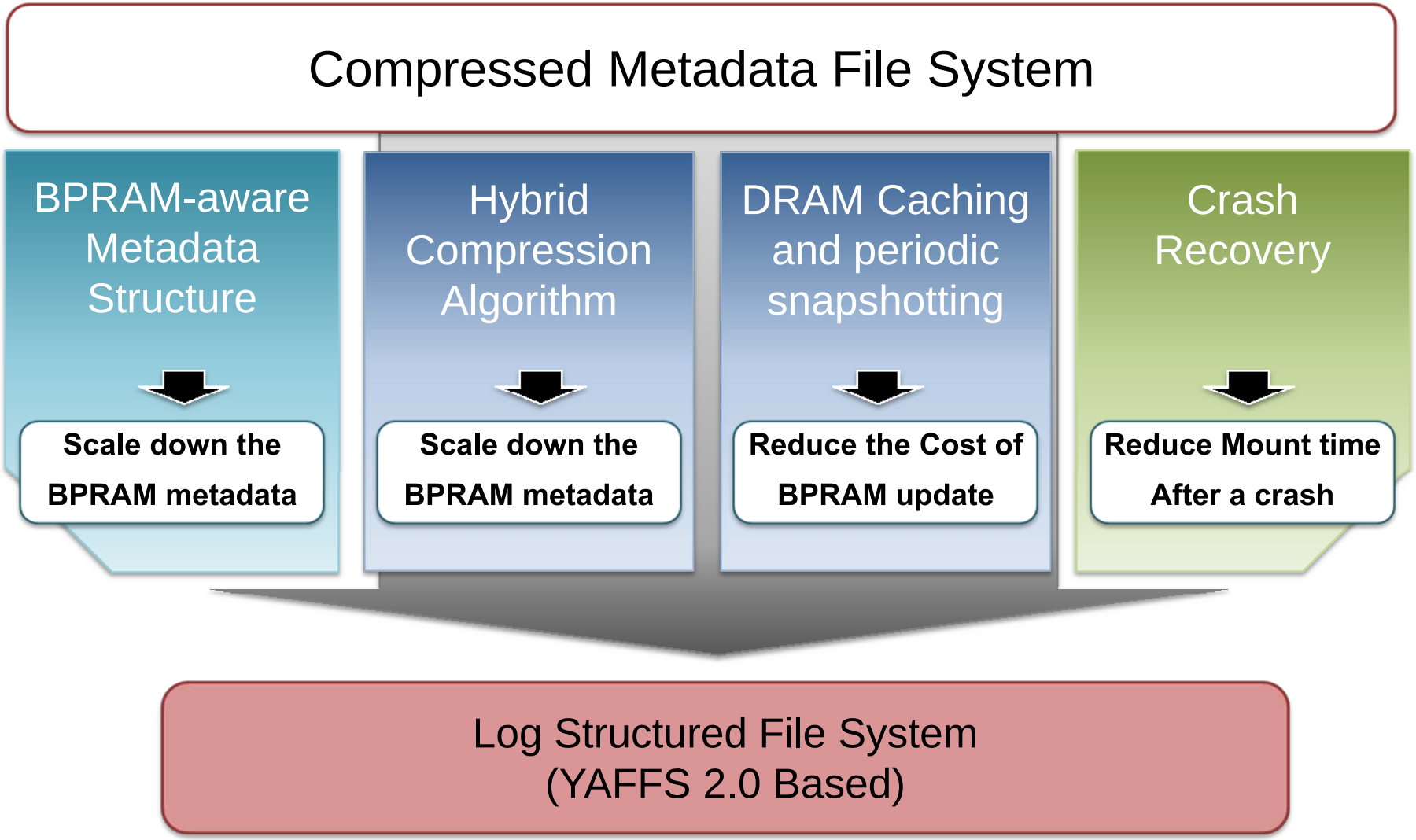
Consistency Issue in Hierarchical File System



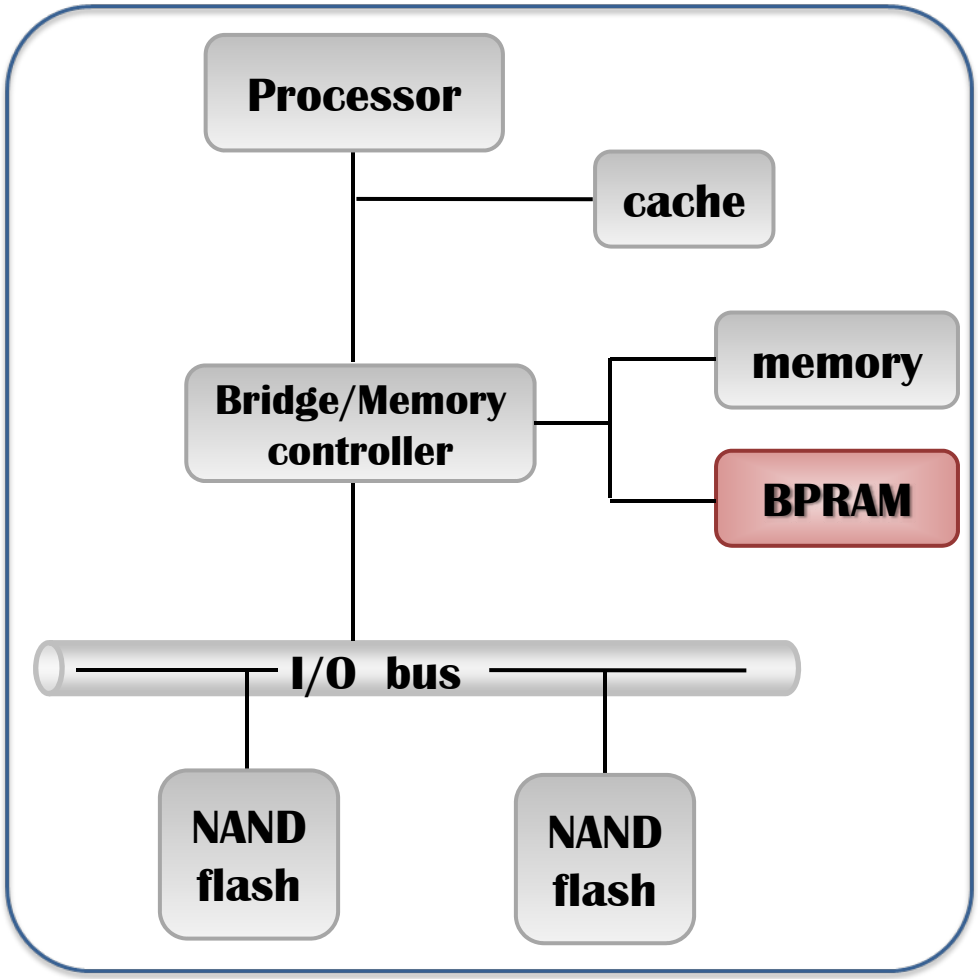
Objective


Scalability target :
1:1000
Instant Recovery for
Unexpected Failure
Hierarchical
Filesystem
for BPRAM and NAND
Flash
Compressed Metadata File System

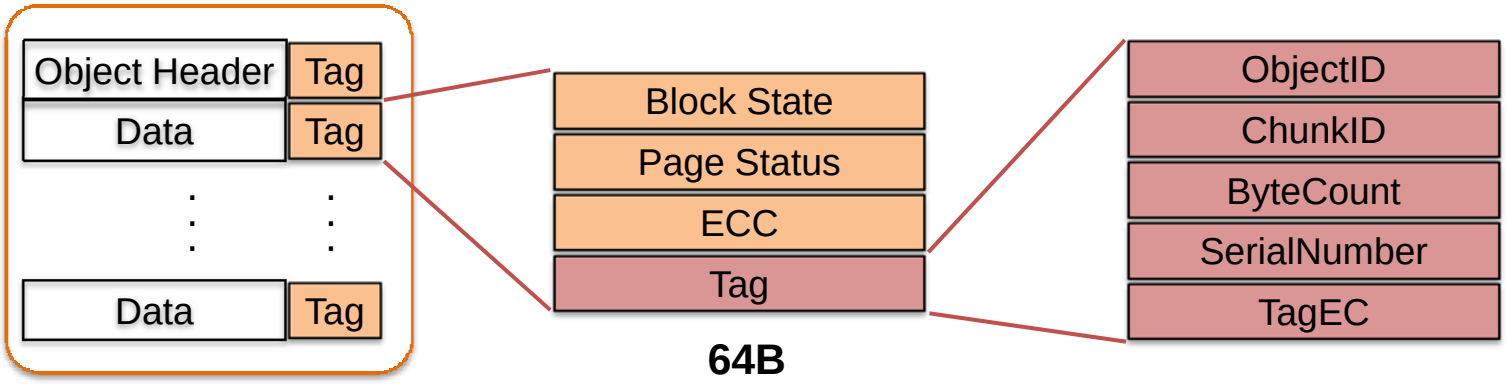
Key Technical Ingredients



System Architecture



BPRAM-aware Meta Structure



NAND Flash

BPRAM

CMFS_validitymarker

- 8byte

SCM_vaild

data_valid

CMFS_device

- 40B + 8B * nblocks

nErasedBlock

nFreeChunck

sequenceNumber

Block info

:

CMFS_object

- 24B + 36B * nObject

structType

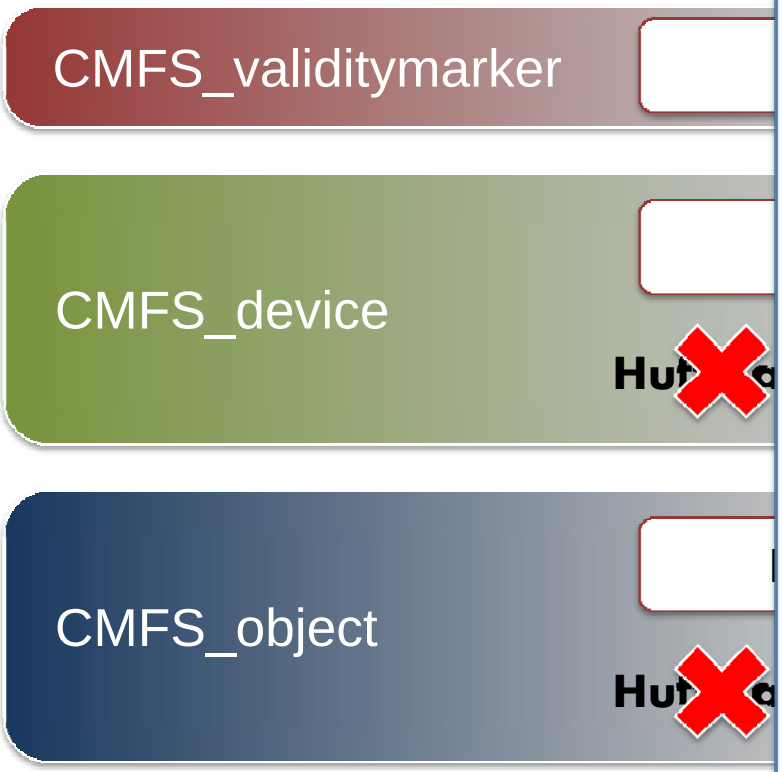
objectID

parentID

PAT

:

Hybrid Compression Algorithm in CMFS



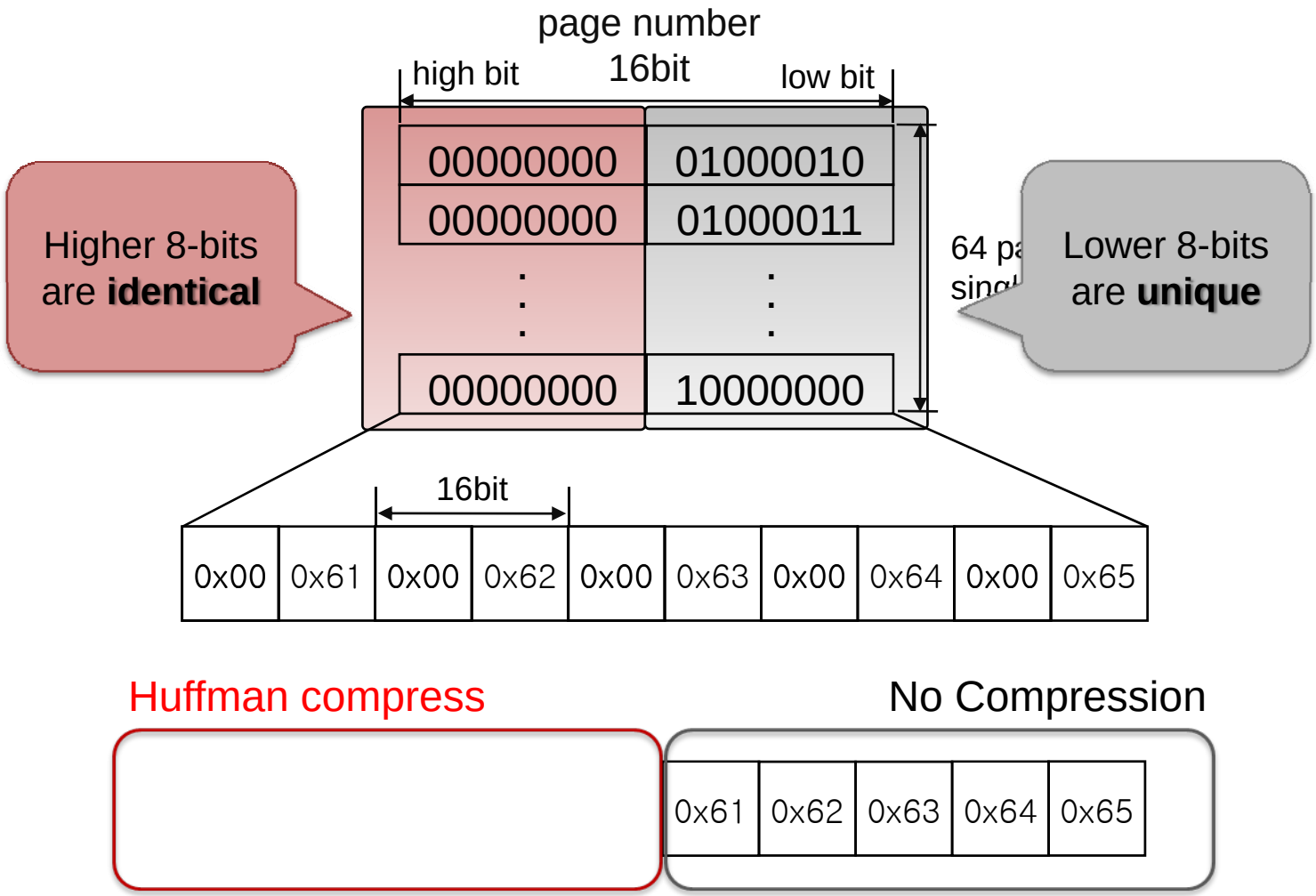
CMFS_object 24bytes + 36bytes * n

field type	field name	size
int	structType	4bytes
unsigned int	objectId	4bytes
unsigned int	parentId	4bytes
int	hdrChunk	4bytes
enum	variantType:3	2bytes
char	deleted:1	1byte
char	softDeleted:1	
char	unlinked:1	
char	fake:1	
char	renameAllowed:1	
char	unlinkAllowed:1	
char	serial	4bytes
int	nDataChunks	4bytes
array	PAT	(36bytes*n)

Target for Compression: PAT object

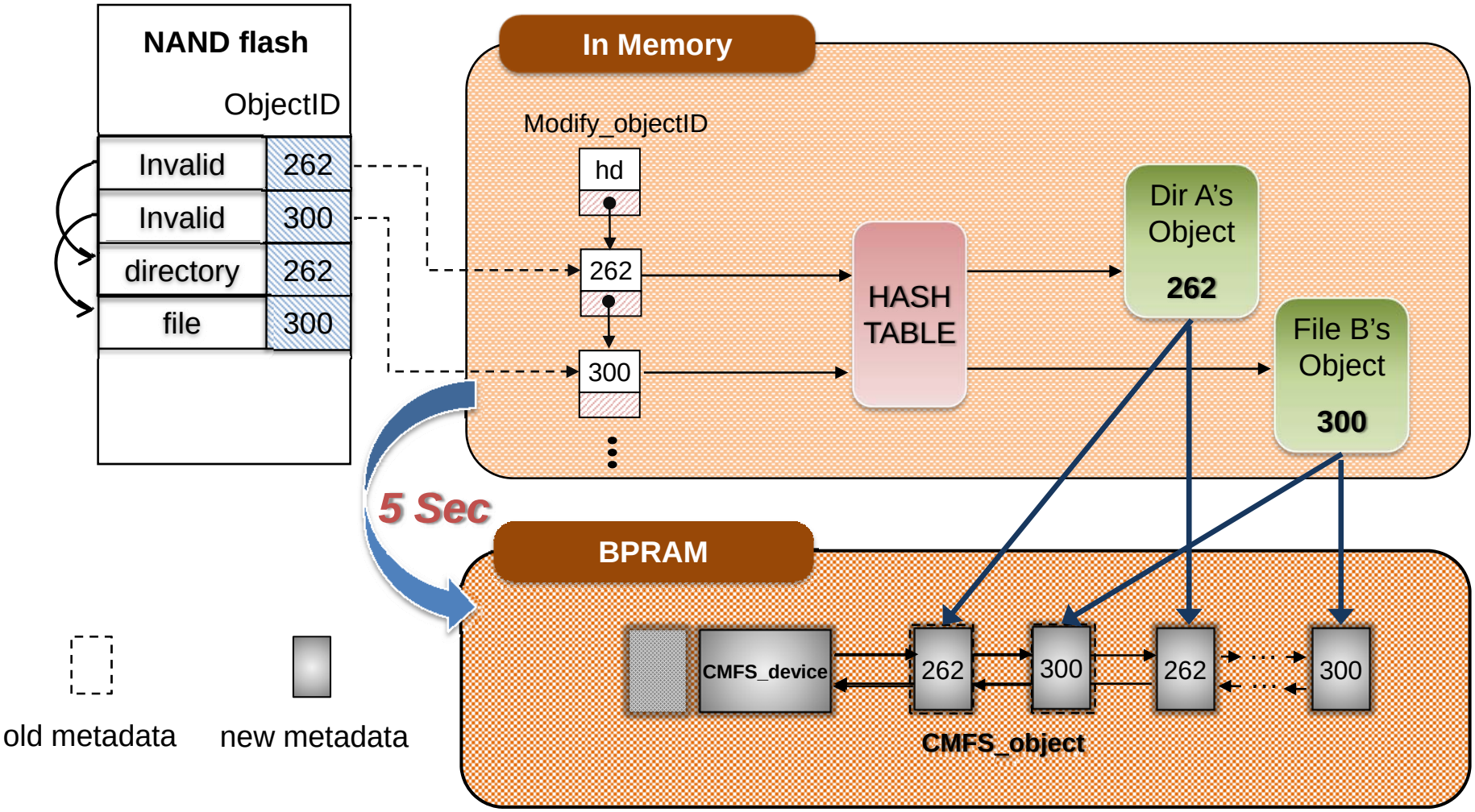
- PAT (Physical Address Translation) Tree

Hybrid Compression



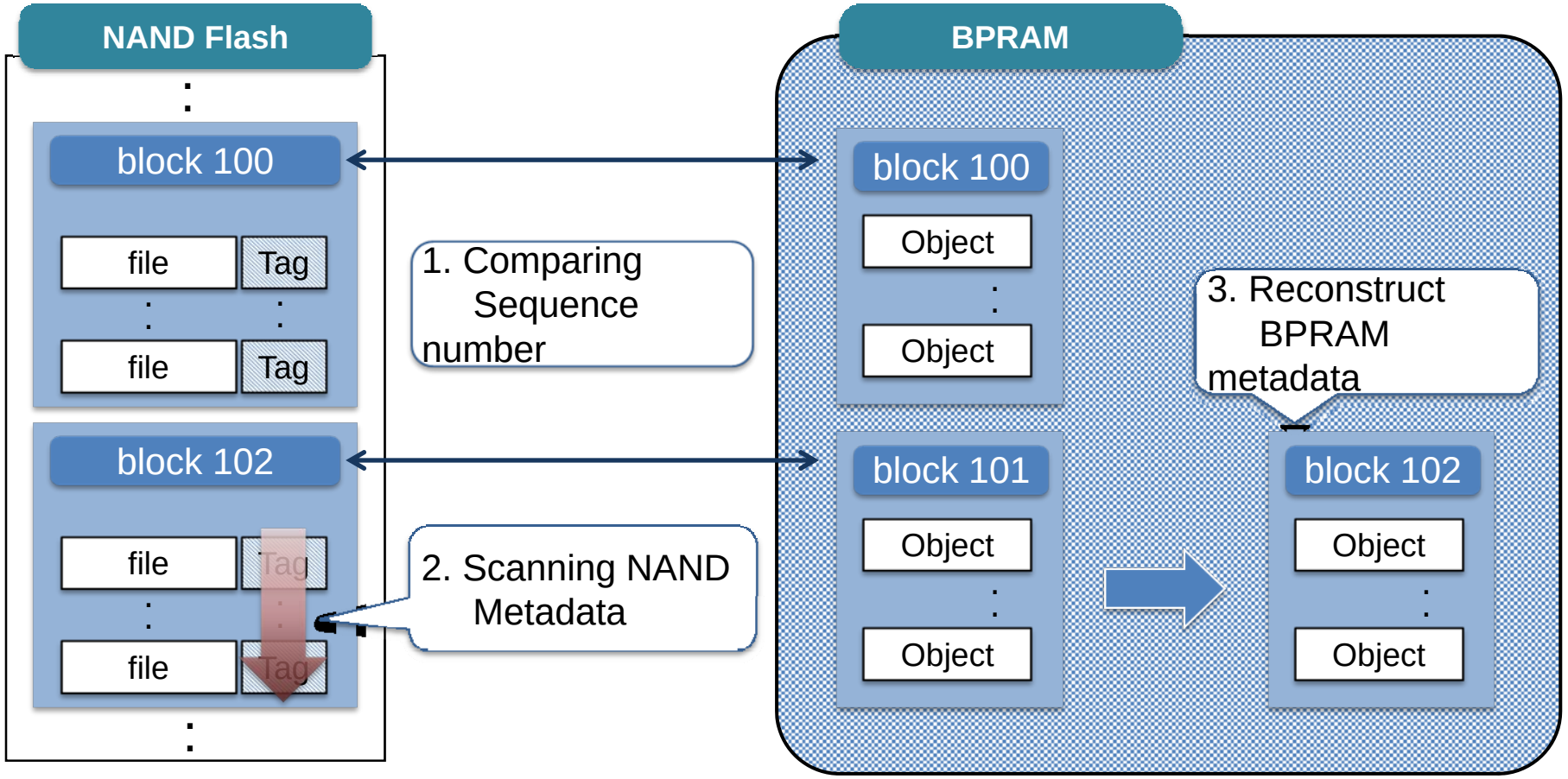
Coping with Size Variability in Compressed Metadata

- Link Based Object organization



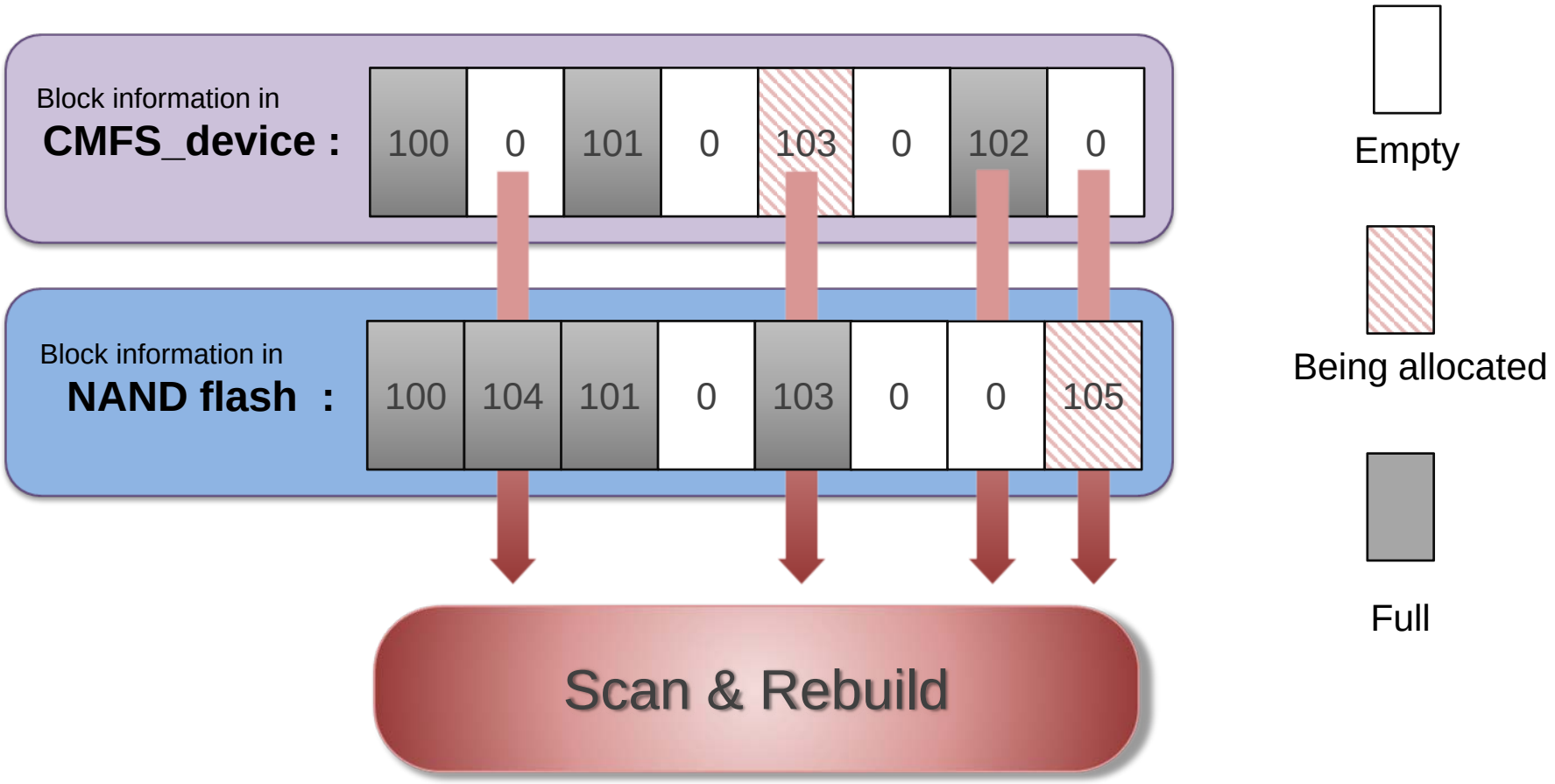
Crash Recovery

- Minimizing Recovery Overhead:
  - Using ***sequence number*** in the BPRAM



Detecting Inconsistency

- Comparing sequence number



Compar sequence number

Experiment

Experimental Environment

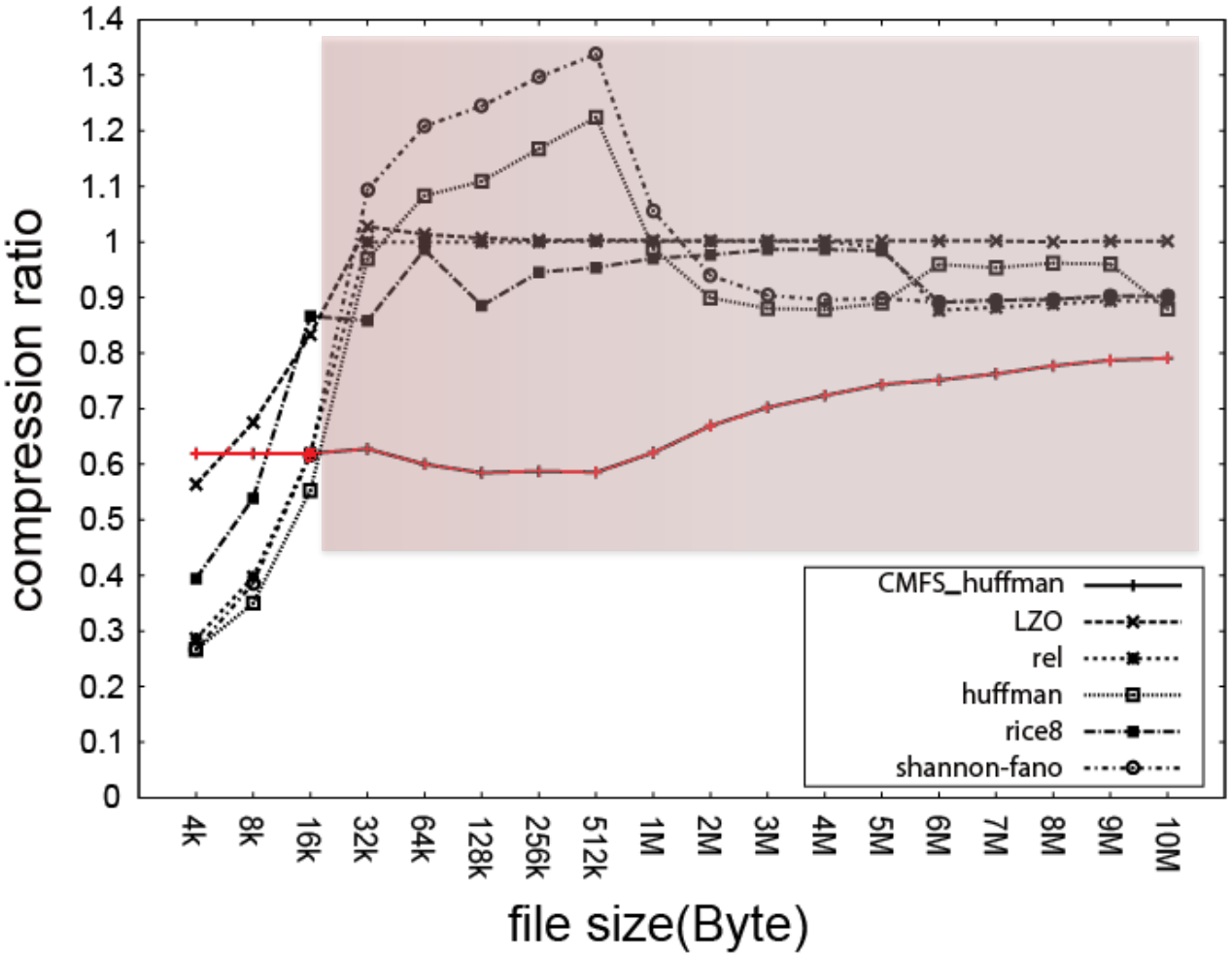
Marvell PXA320 Board

- 806 MHz
- 128 MB SDRAM
- 128 MB large block NAND flash memory
- Linux kernel 2.6.24

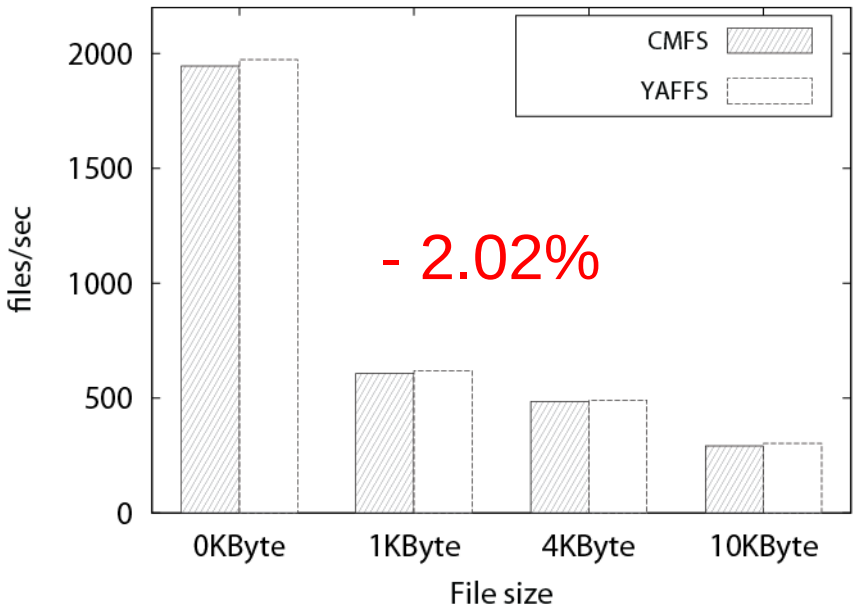
A portion of DRAM is used to emulate BPRAM.

Experiment: Hybrid Compression

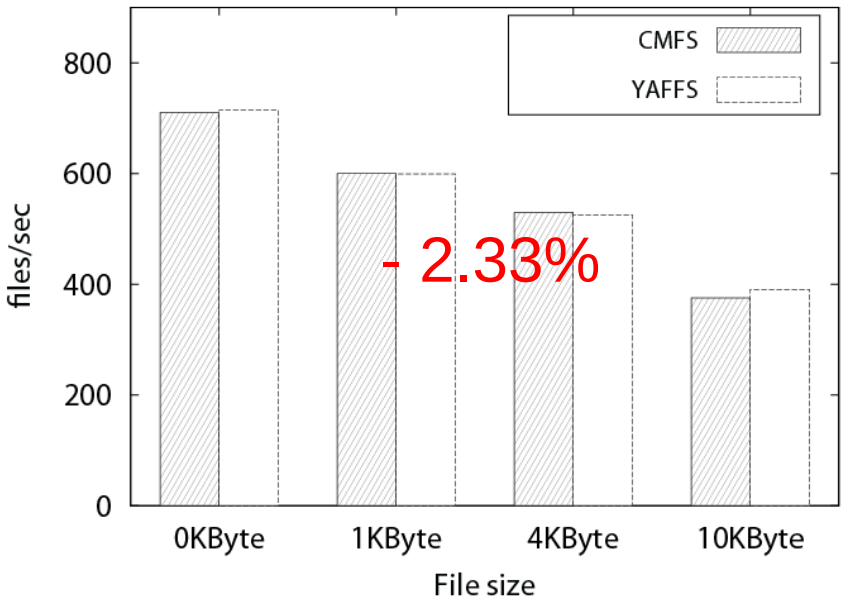
When the file size > 16KB, Hybrid compression 31% better than LZO.



Experiment: Metadata Update(LMBEMCH)



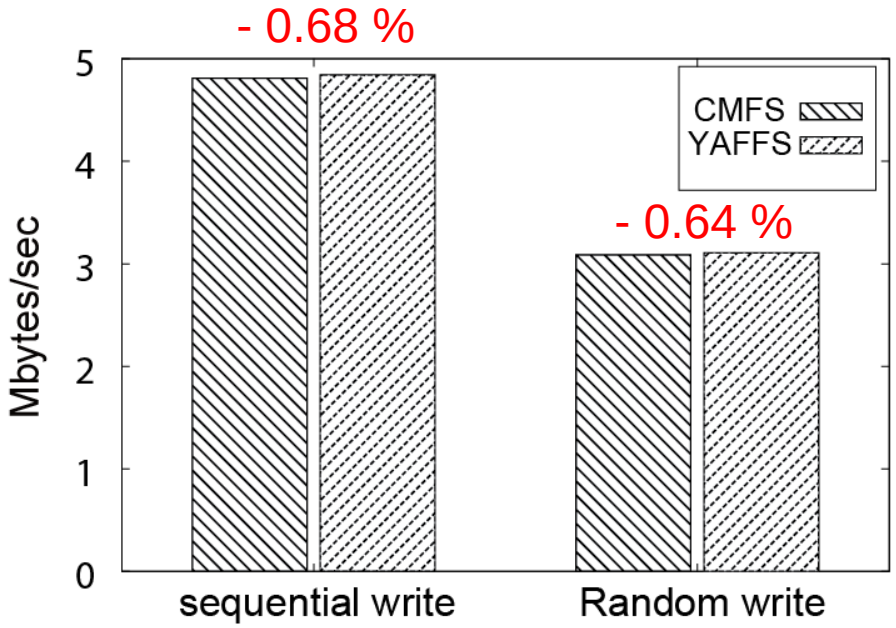
File Creation



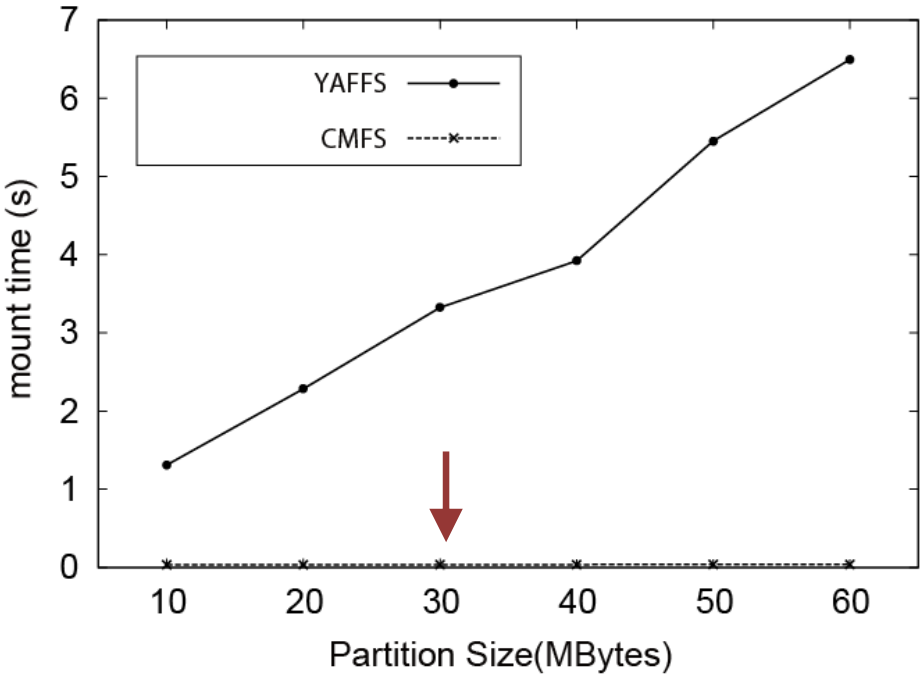
File Deletion

Experiment: Random/Sequential Write(IOZONE)

No performance hit even with compression and metadata synch operation.



Experiment: Mount Overhead



- YAFFS : the mount latency linearly increases
- CMFS : the mount latency does not vary subject to the change in file system partition size.

Conclusion

***CMFS** is a novel hierarchical file system which provides*

efficient management method** of **BPRAM

- BPRAM-aware metadata design for scalability
- Hybrid compression for scalability
- BPRAM update method to reduce the Cost of BPRAM update
- Efficient Detection of Inconsistent Blocks and Crash recovery

The end

Thank You