Extending the Lifetime of Flash-based Storage through Reducing Write Amplification from File Systems

Youyou Lu, Jiwu Shu, Weimin Zheng

Tsinghua University

Outline

- Background and Motivation
- Object-based Flash Translation Layer
- System Co-design with Flash Memory
- Evaluation
- Conclusion

Flash Memory

- Gained Popularity
  - High performance, low energy, reduced cost
  - Wide deployment: embedded devices -> enterprise servers
- Endurance
  - SLC (100,000)
  - MLC (10,000)
  - TLC (1,000)

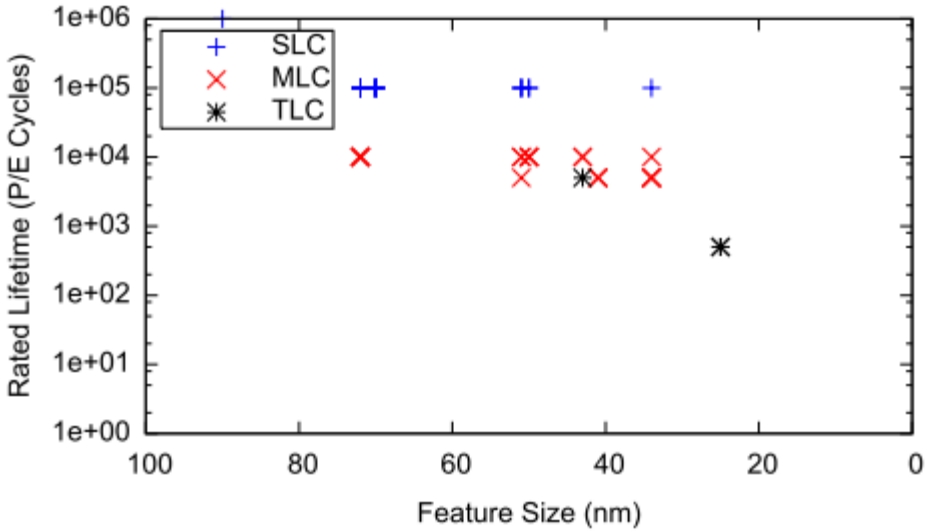


Figure from "The Bleak Future of NAND Flash Memory" in FAST'12

Existing Approach to Flash Endurance

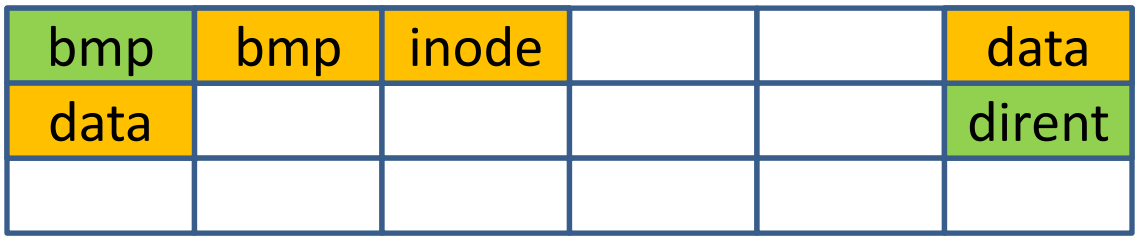
- Wear Leveling
  - To make all the blocks evenly worn out
  - Fundamental part of the FTL
- Data Reduction
  - To reduce the amount of data to be written
  - Data De-duplication and Compression
  - Used either in FTL or in FS

Write Amplification from File Systems

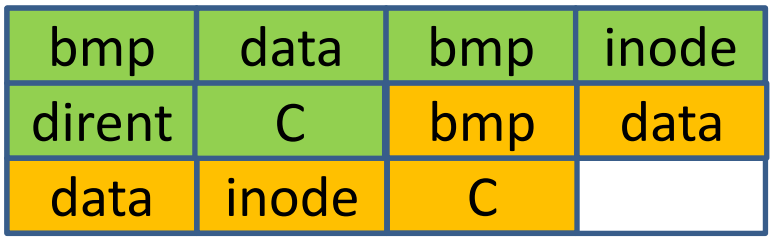
- Write Amplification from File Systems
  - Pre-FS writes vs. Post-FS writes
- Journaling
  - Keep the journals in the logs first,
  - And then, checkpoint them in-place
- Metadata synchronization
  - Frequent persistence in case of data lost or inconsistency
- Page-aligned update
  - Wasted space within one page

A simple example in ext3

- Echo “title” > foo.txt
  - Effective Data: 6 bytes
  - Flash Writes: 11 pages * 4KB/page = 44KB
- Echo “texttexttext…”(4KB) >> foo.txt
  - Effective Data: 4KB
  - Flash Writes: 9 pages * 4KB/page = 36KB



Data Area



Journal Area

Flash Opportunities

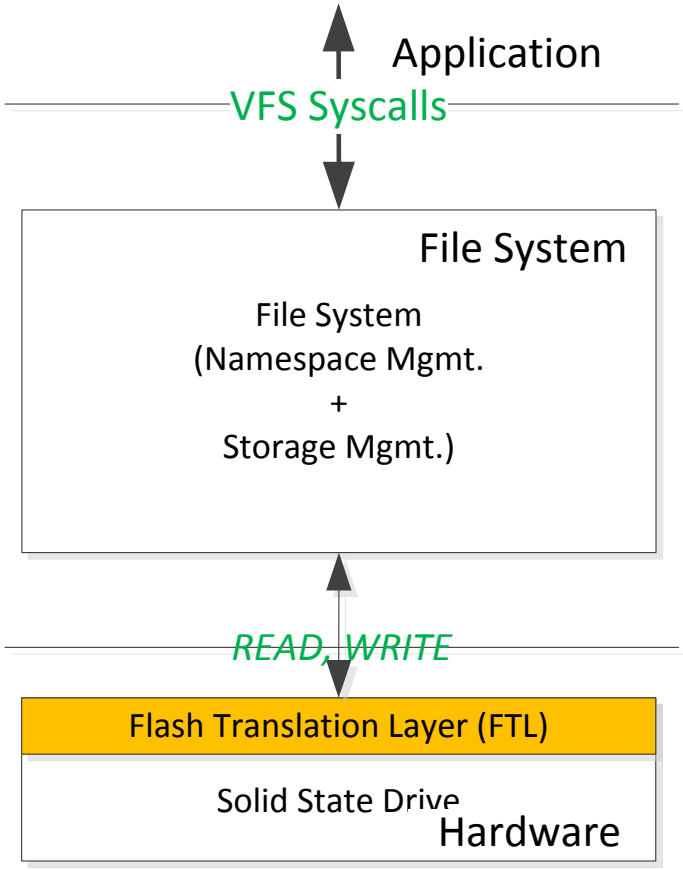
- No-overwrite
  - Can the journaling use it without writing twice in the file system?
- Page metadata
  - Can we store the backpointer to lazily write back the index while keeping consistency?
- Erase-before-Update
  - Can we track the free space in a coarse-grained way?

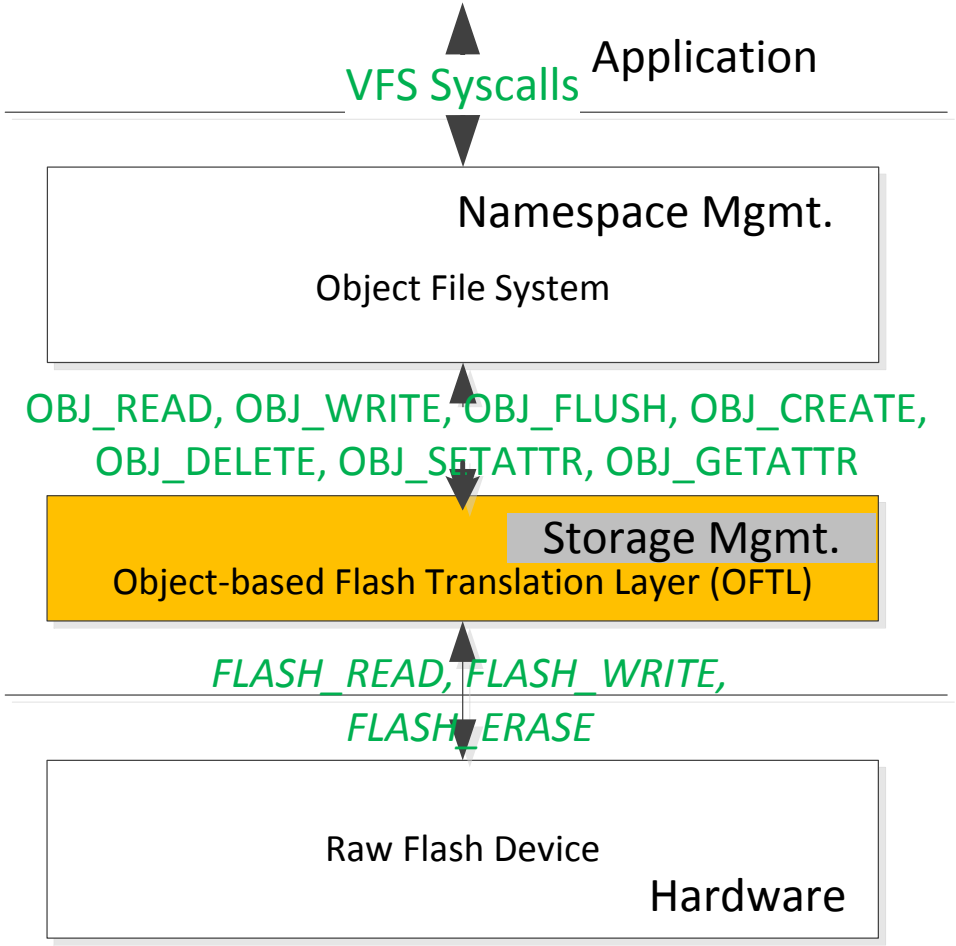
Outline

- Background and Motivation
- Object-based Flash Translation Layer
- System Co-design with Flash Memory
- Evaluation
- Conclusion

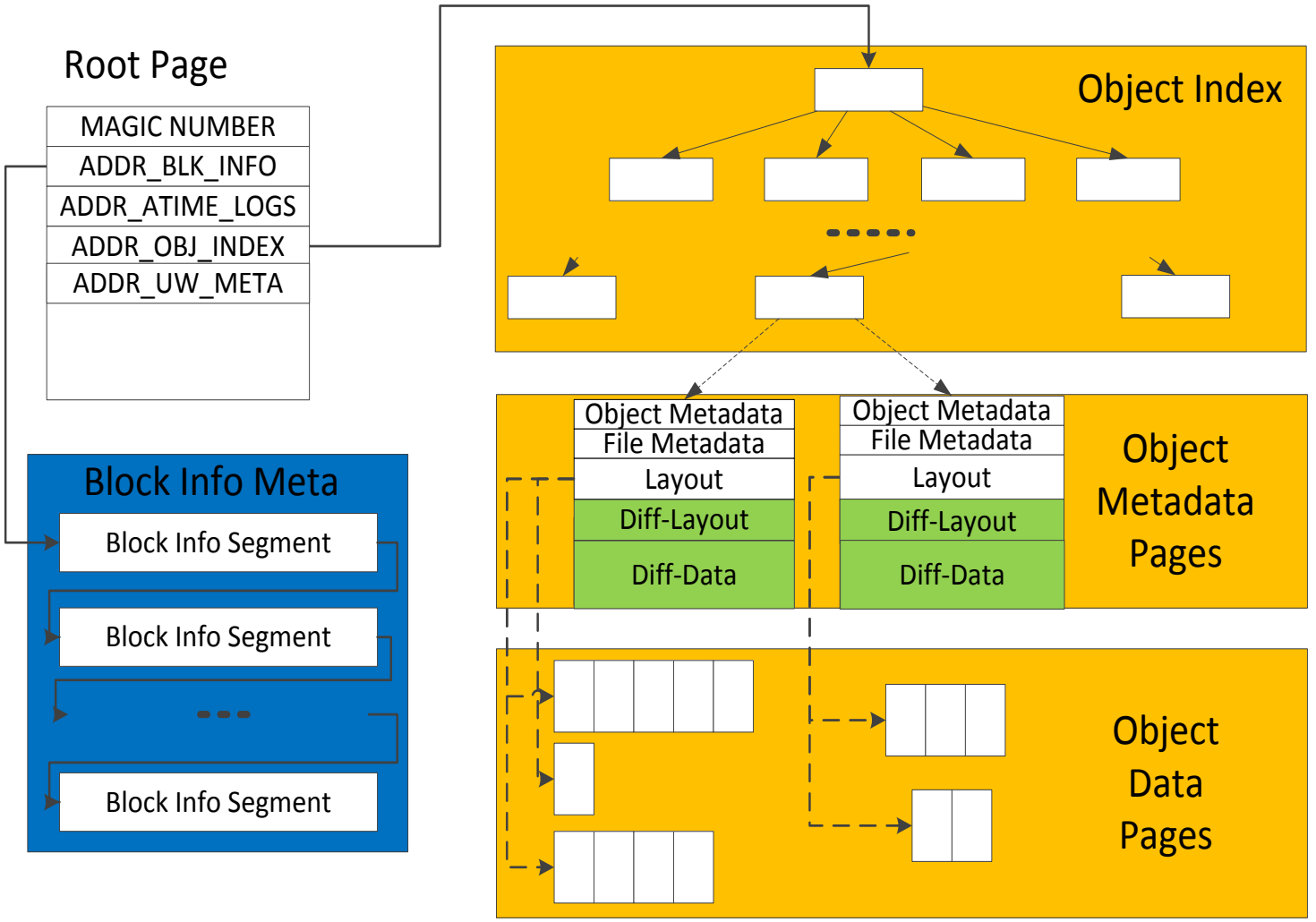
Architecture & Data Organization

- Intelligent Storage Mgmt.
- Optimized FS Mechanism





OFTL Data Layout



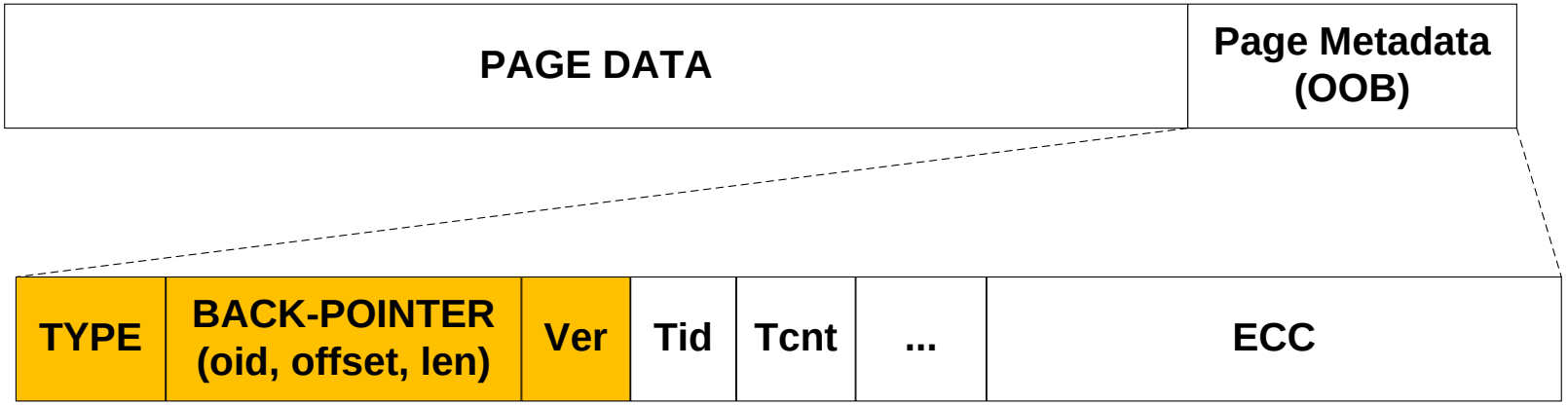
Coarse-grained Block State Maintenance

Lazy Indexing

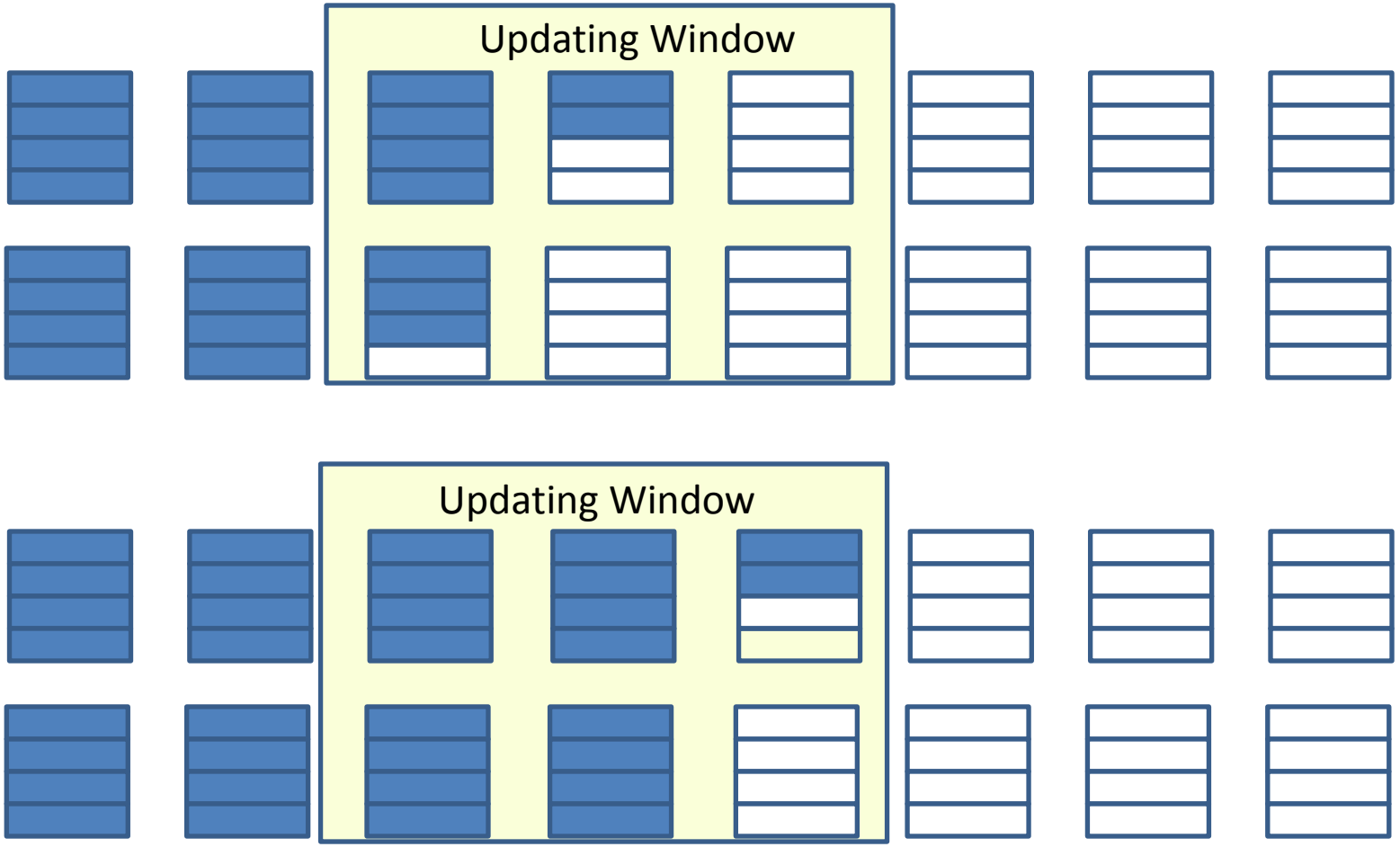
Compacted Update

T1. Lazy Indexing

- Index Metadata
  - The metadata that stores the pointers (the physical addresses of other pages).
- Object Index -> Object Metadata Page -> Object Data Page
  - Type-specific backpointers

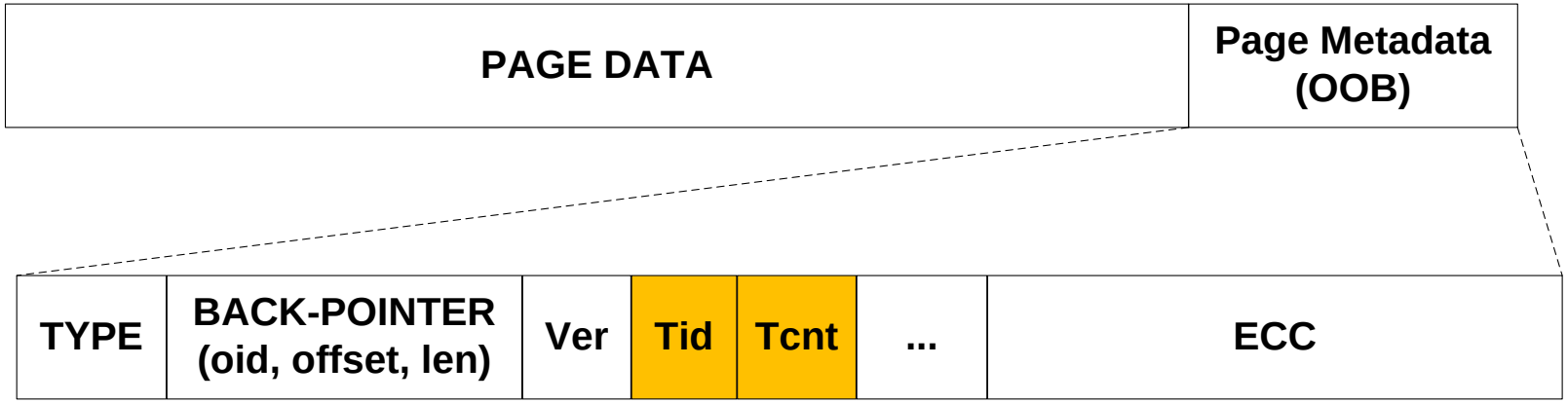


Updating Window & Checkpointing



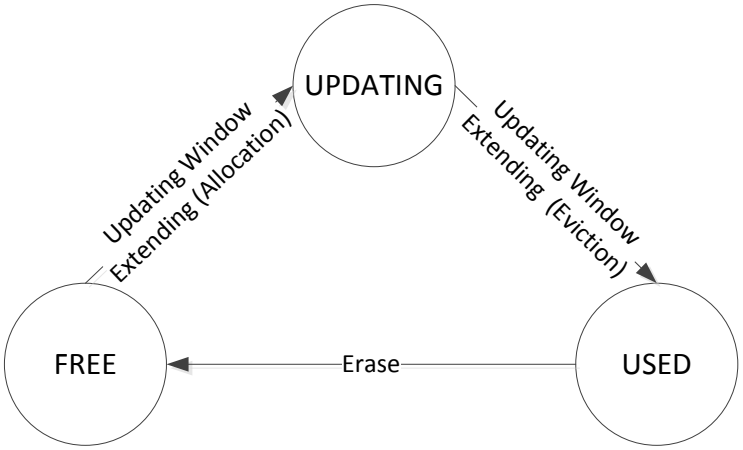
1. Make sure the mappings are persistent
2. Write back the updating window metadata

- Transactional write
  - <tid, tcnt>
  - Count the total number of the pages with the same tid, and compare with the stored tcnt



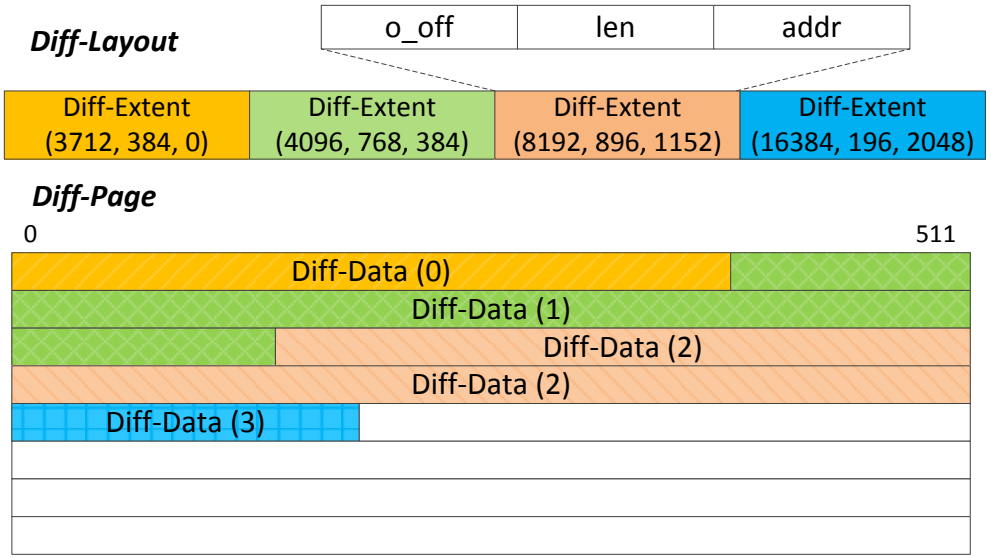
T2. Coarse-grained Block State Maintenance

- Free space in flash block units
  - Page states can be identified using the block state
    - Pages in FREE blocks are all free
    - Pages in USED blocks are all used
    - Pages in UPDATING blocks need to be further identified
- Relaxed Metadata Persistence

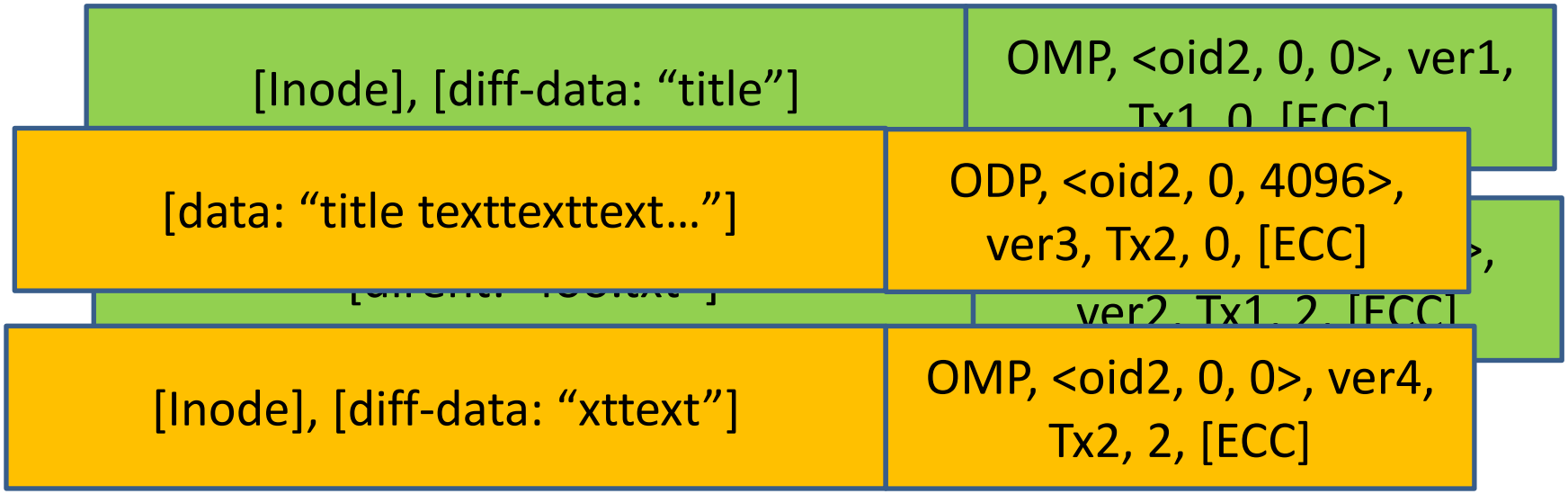


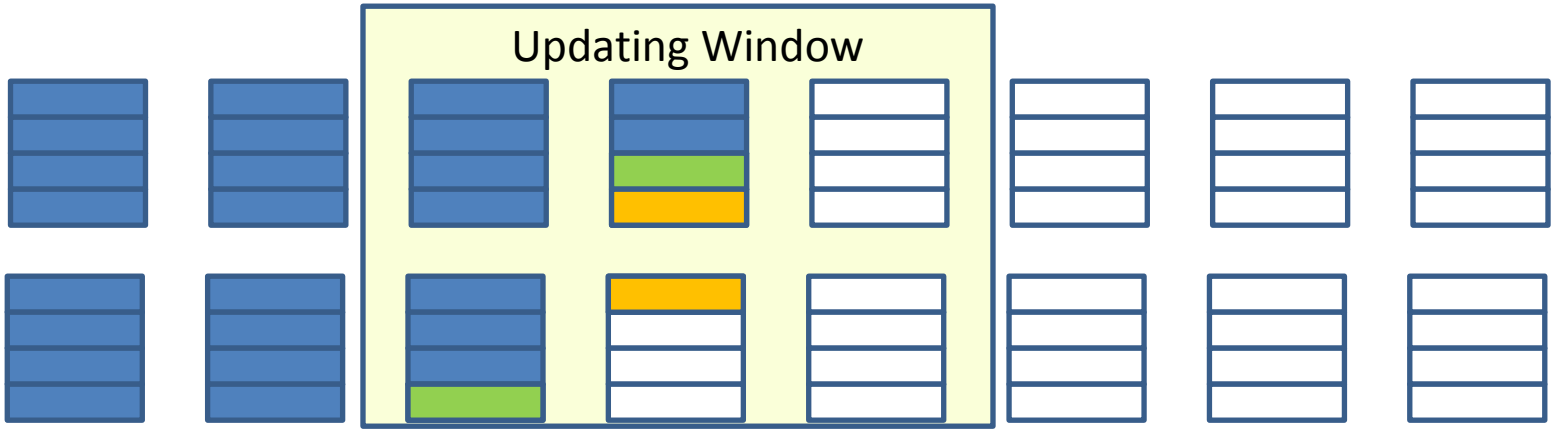
T3. Compacted Update

- Compact multiple partial page updates into one flash page
- Co-locate the diff-page with the metadata page



An Example in OFSS





- Echo “title” > foo.txt
- Echo “texttexttext…”(4KB) >> foo.txt

Comparison of Ext3 and OFSS

80 KB (ext3) -> 16 KB (OFSS)

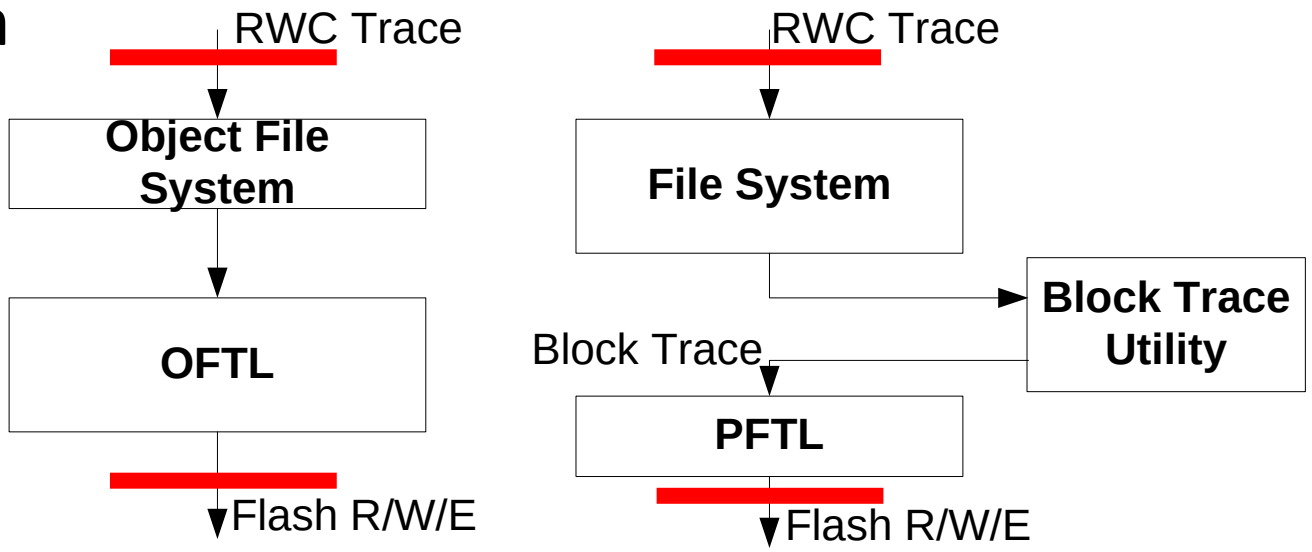
- Journals => Transactional Metadata in Page Metadata
- Inode => Reverse Index in Page Metadata
- Block/Inode Bitmap => Free Space Mgmt. in Flash Block Units
- Page Un-aligned Update => Compaction and Co-location

Outline

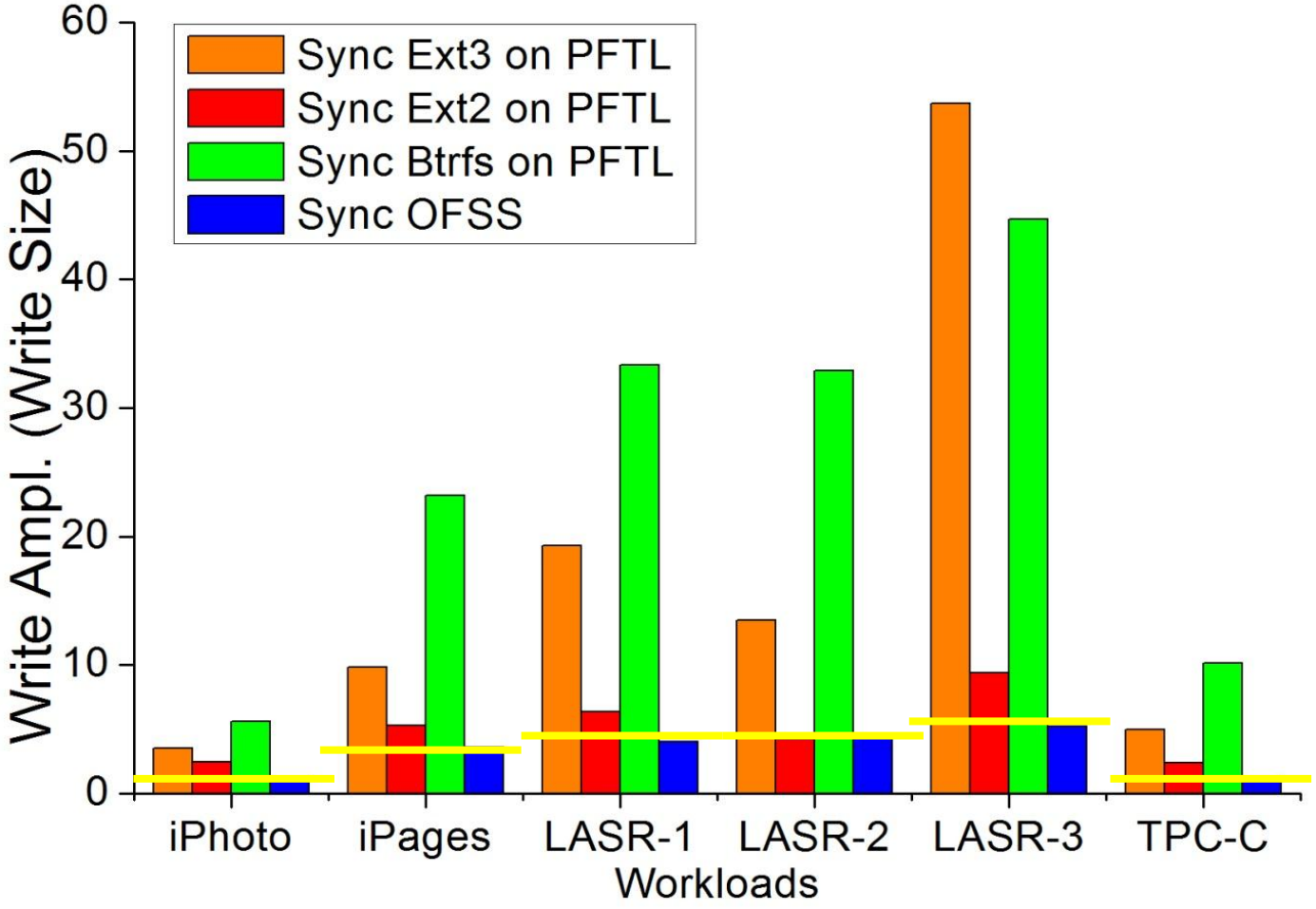
- Background and Motivation
- Object-based Flash Translation Layer
- System Co-design with Flash Memory
- Evaluation
- Conclusion

Evaluation Method

- Evaluation Metric
  - write amplification = flash_writes / app_writes
- System Evaluation Framework
  - OFSS
  - File system



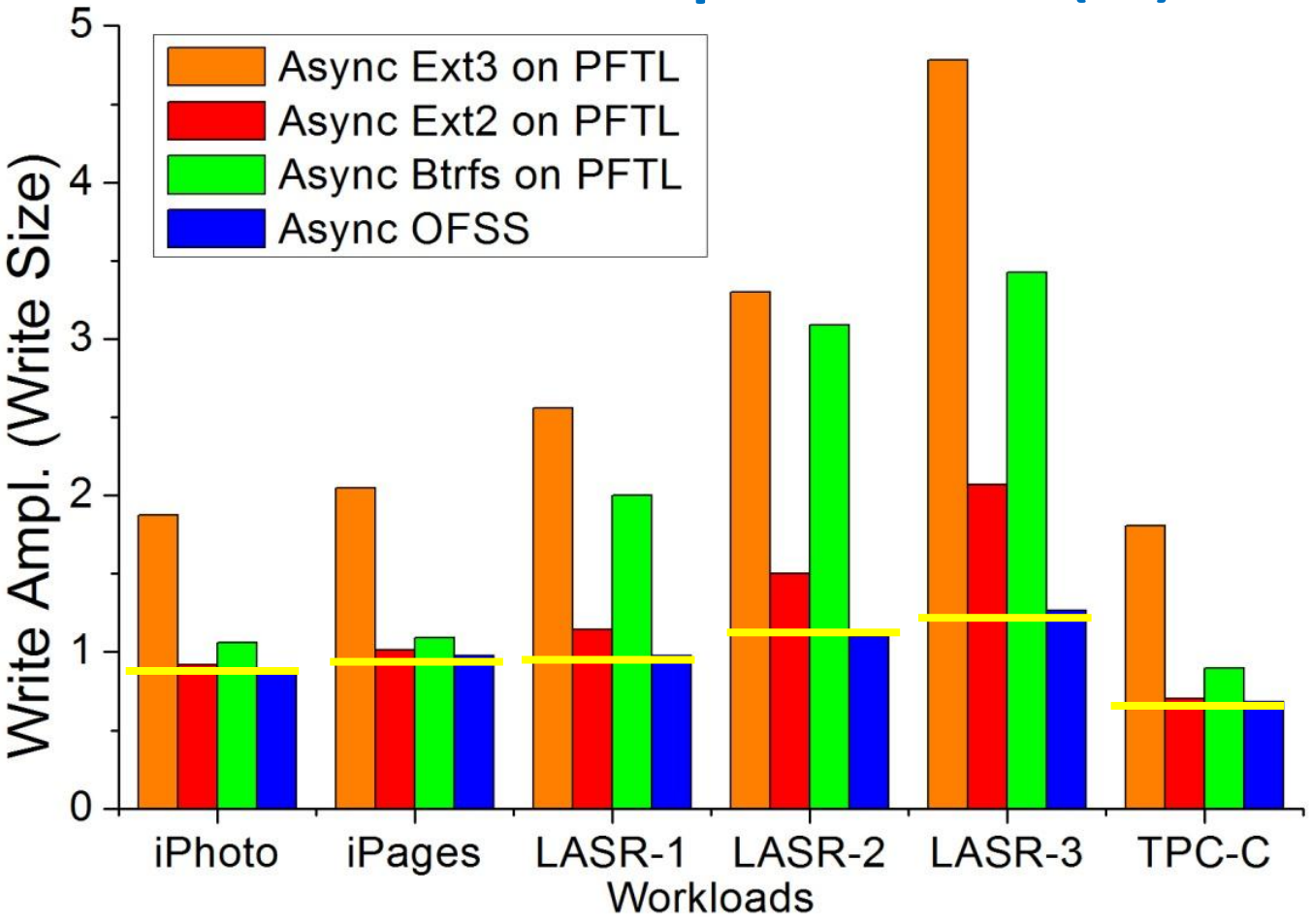
Overall Comparison (1)



Write Amplification:

OFSS = 15.1% * ext3 = 52.6% * ext2 = 10.6% * btrfs

Overall Comparison (2)



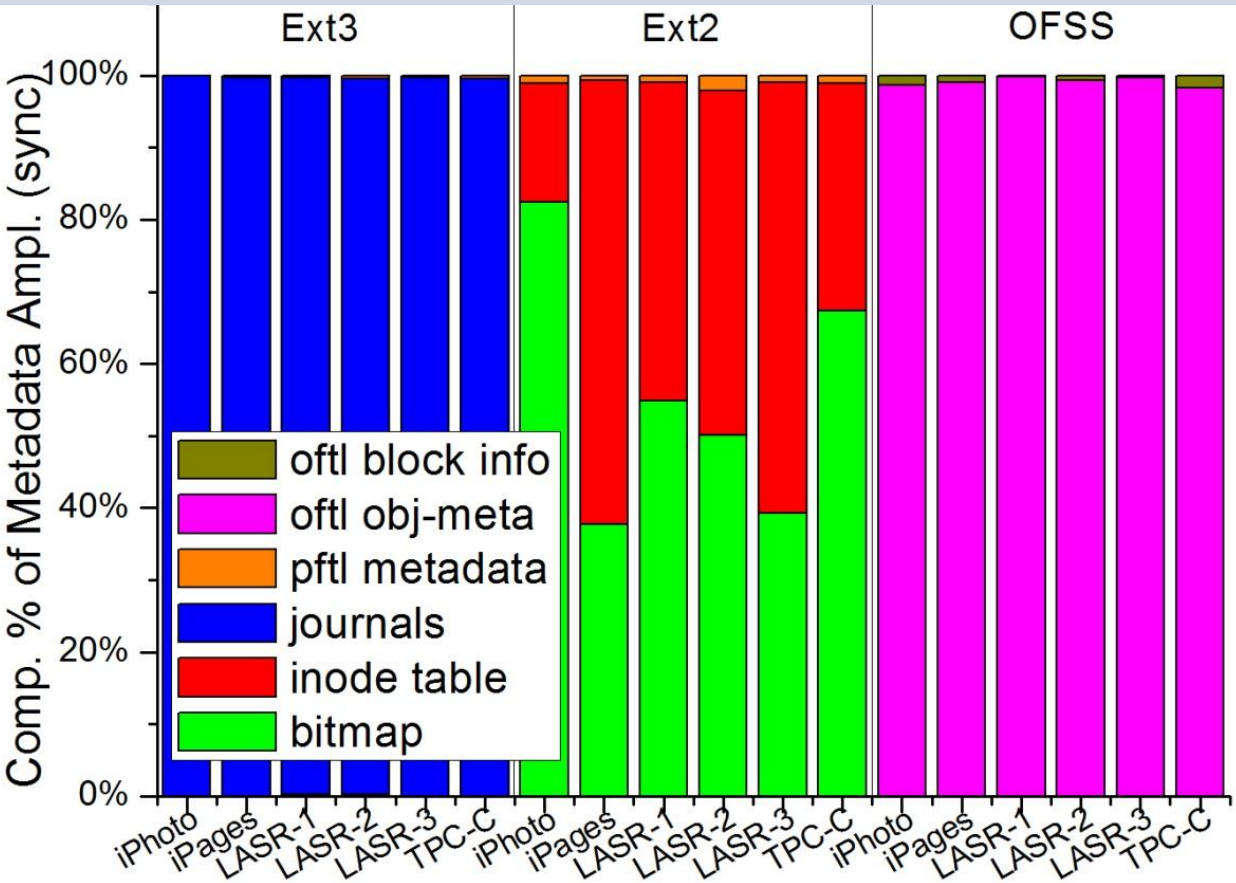
Write Amplification:

OFSS = 36.0% * ext3 = 80.2% * ext2 = 51.0% * btrfs

Metadata Amplification

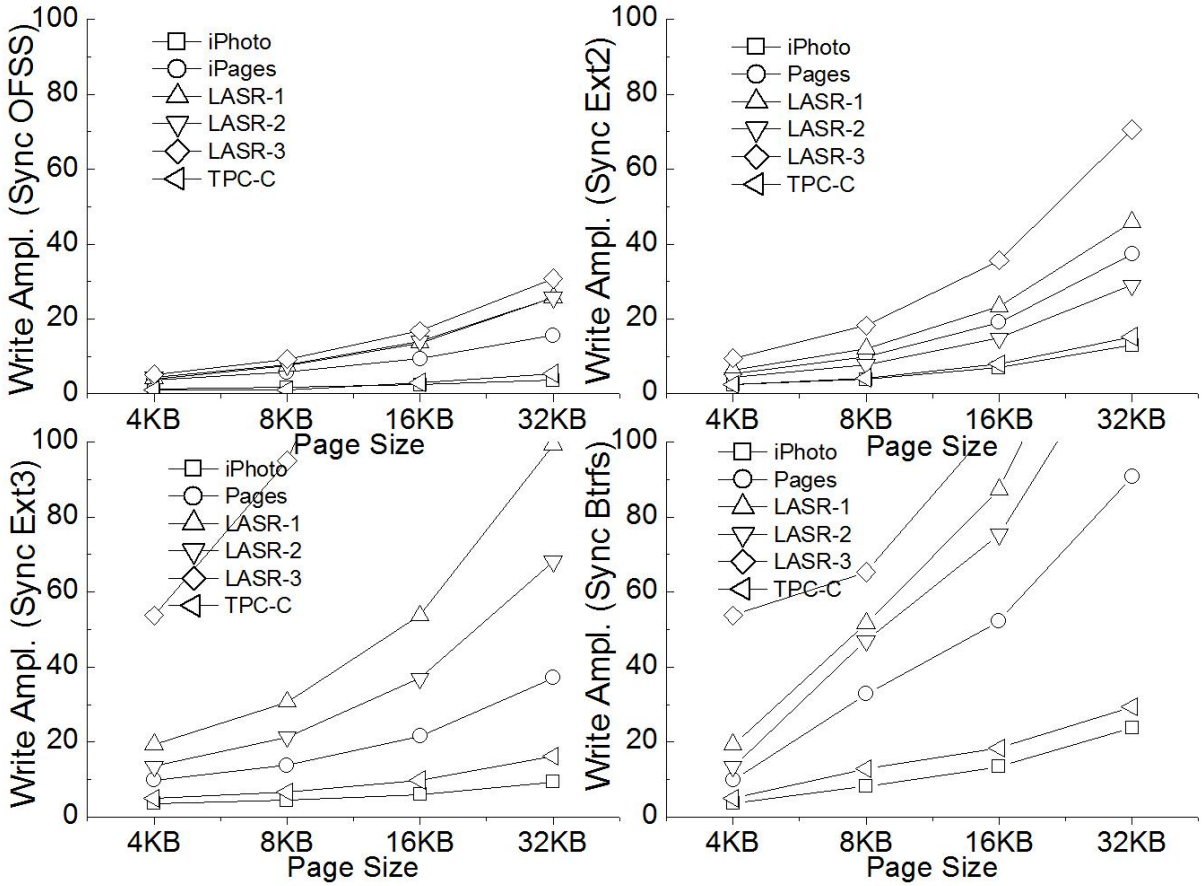
- ✓In OFSS, meta ampl. is dramatically reduced
- ✓Ext3: Journaling
- ✓Ext2: Bitmap and Inode Table

	iPhoto	iPages	LASR1	LASR2	LASR3	TPCC
Ext3	2.57	8.59	17.91	11.84	51.04	3.73
Ext2	1.06	2.68	2.00	0.91	4.11	1.04
OFSS	0.05	0.30	1.13	0.45	1.05	0.03



Refer to the paper for more details of the async mode

Flash Page Size Impact



- Write amplification gets worse and worse as the flash page size increases
- The sync mode is much more worse than the async

Refer to the paper for more details of the async mode

Outline

- Background and Motivation
- Object-based Flash Translation Layer
- System Co-design with Flash Memory
- Evaluation
- Conclusion

Conclusion

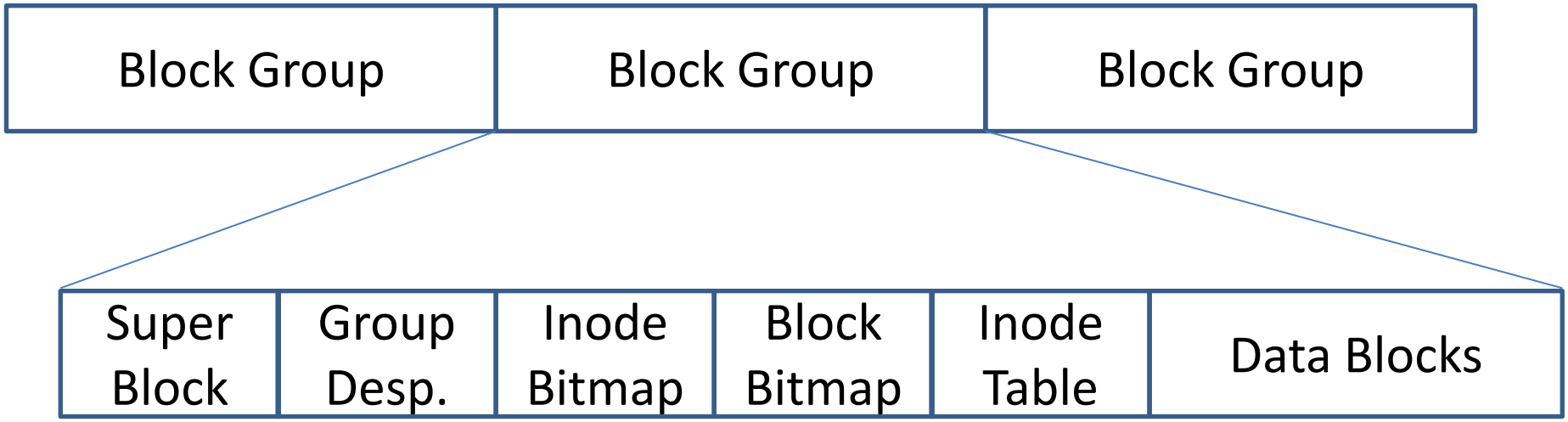
- Metadata in file systems are frequently written back for consistency and durability, amplifying the writes to the flash memory
- Flash memory offers opportunities for endurance-aware file system mechanisms
  - Journaling: transactional write
  - Metadata Synchronization: lazy indexing, coarse-grained block state maintenance
  - Page-aligned Update: compacted update

Thanks!

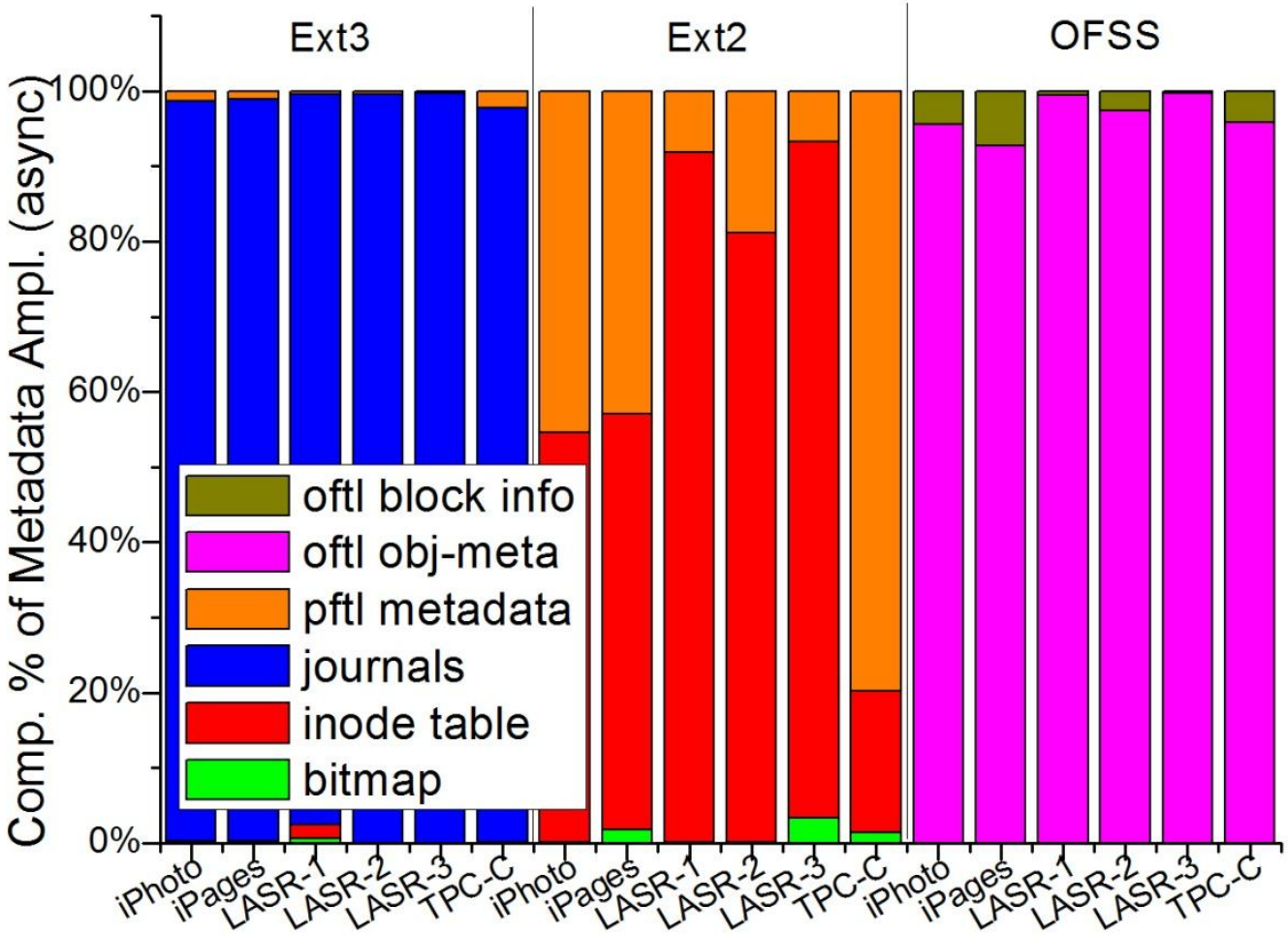
Questions?

Youyou Lu, Jiwu Shu, Weimin Zheng
(luyy09@mails.tsinghua.edu.cn)
http://storage.cs.tsinghua.edu.cn/~lu

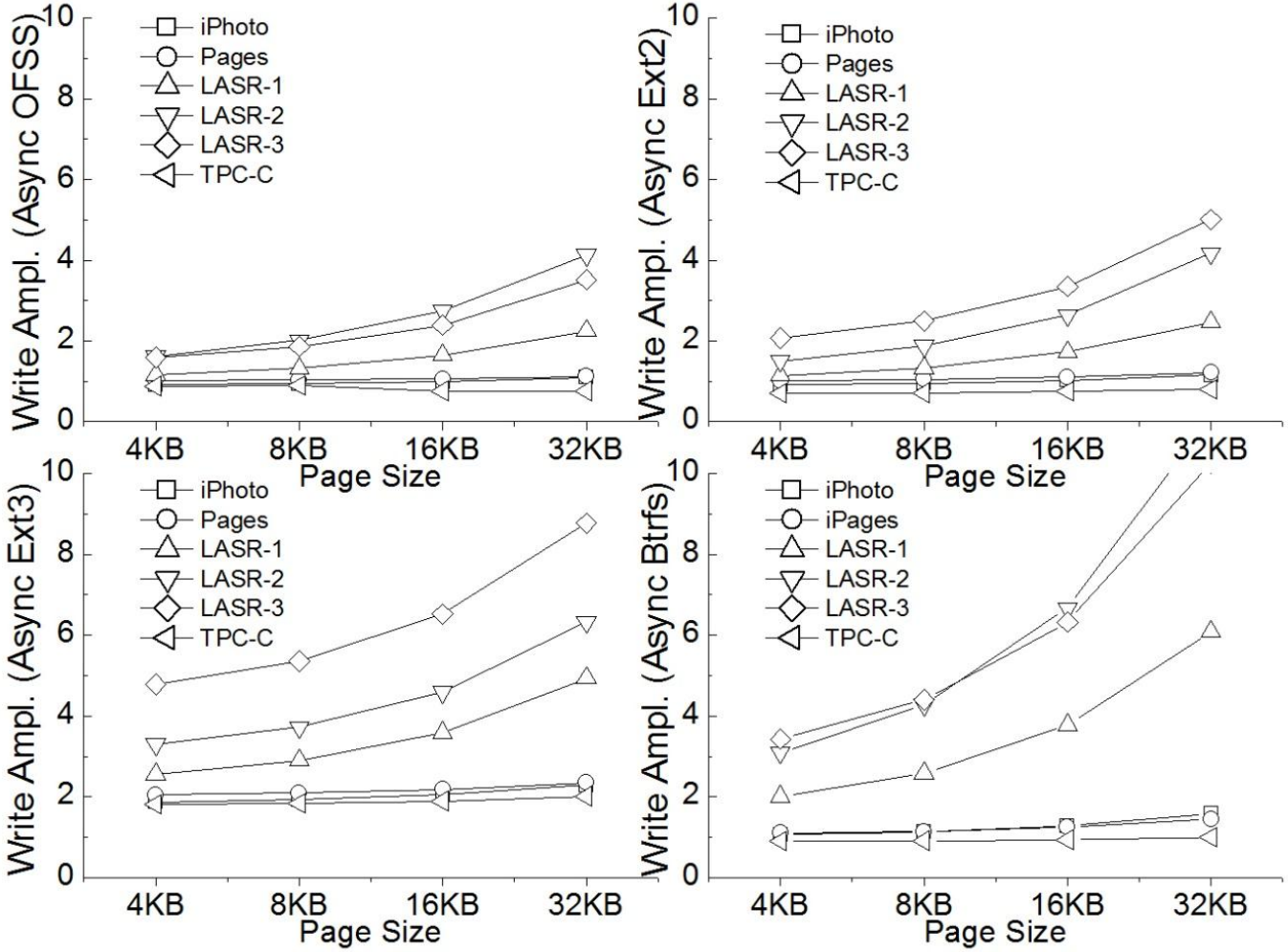
Backup – ext3 layout



Backup – Metadata Amplification (async)



Backup – Impact of Flash Page Size (async)



Backup – Overhead of Window Extending