

I/O Stack Optimization for Smartphones



**Sooman Jeong¹, Kisung Lee², Seongjin Lee¹,
Seoungbum Son², and Youjip Won¹**

¹ Dept. of Electronics and Computer Engineering, Hanyang University

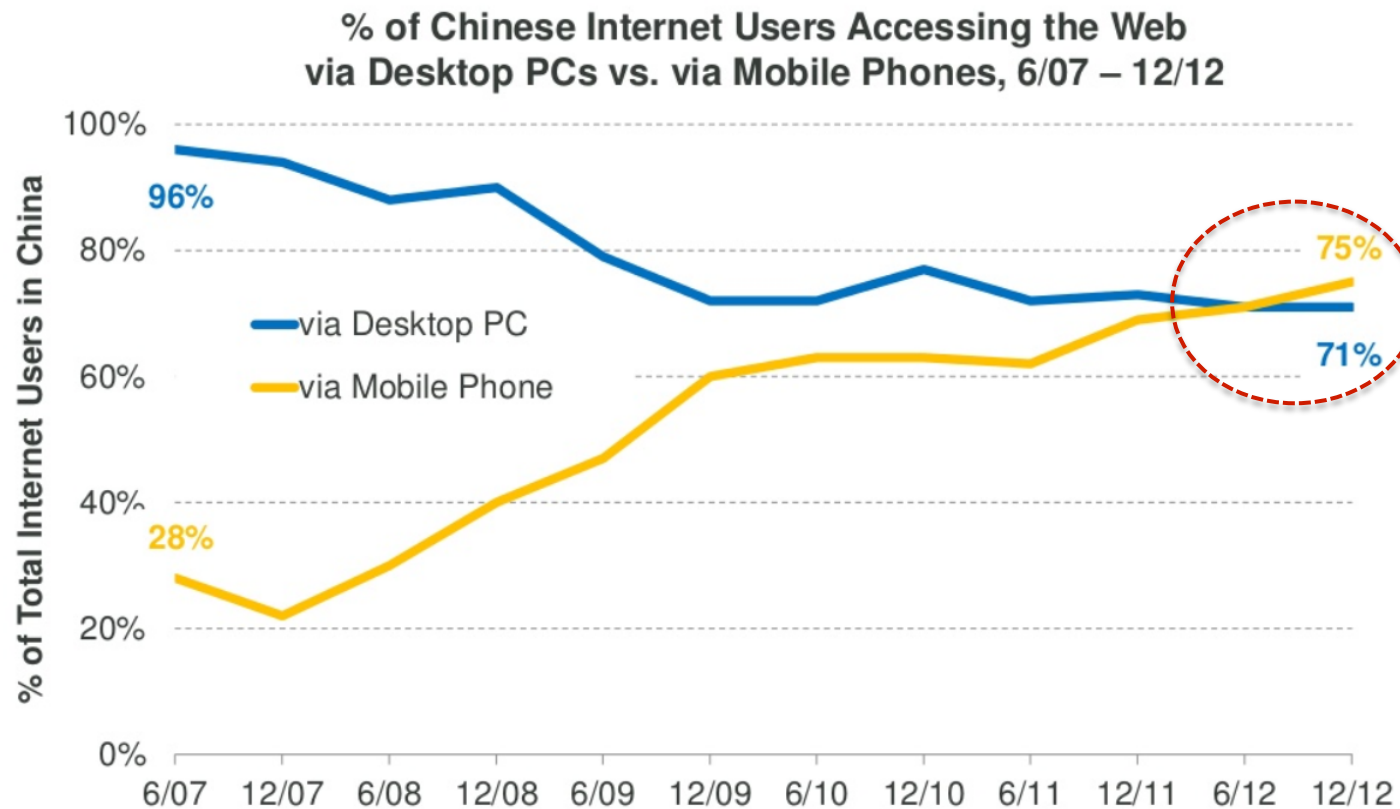
² Samsung Electronics

Outline

- Motivation
- Background
- Analysis of the Android I/O Stack
- Optimizations of the Android I/O Stack
 - ◆ Using the optimal journaling mode in SQLite
 - ◆ Alternative Filesystems
 - ◆ Eliminating unnecessary metadata flushes
 - ◆ External journaling
 - ◆ Using polling based I/O
- Evaluations

Motivation

Smartphone is everywhere!♪



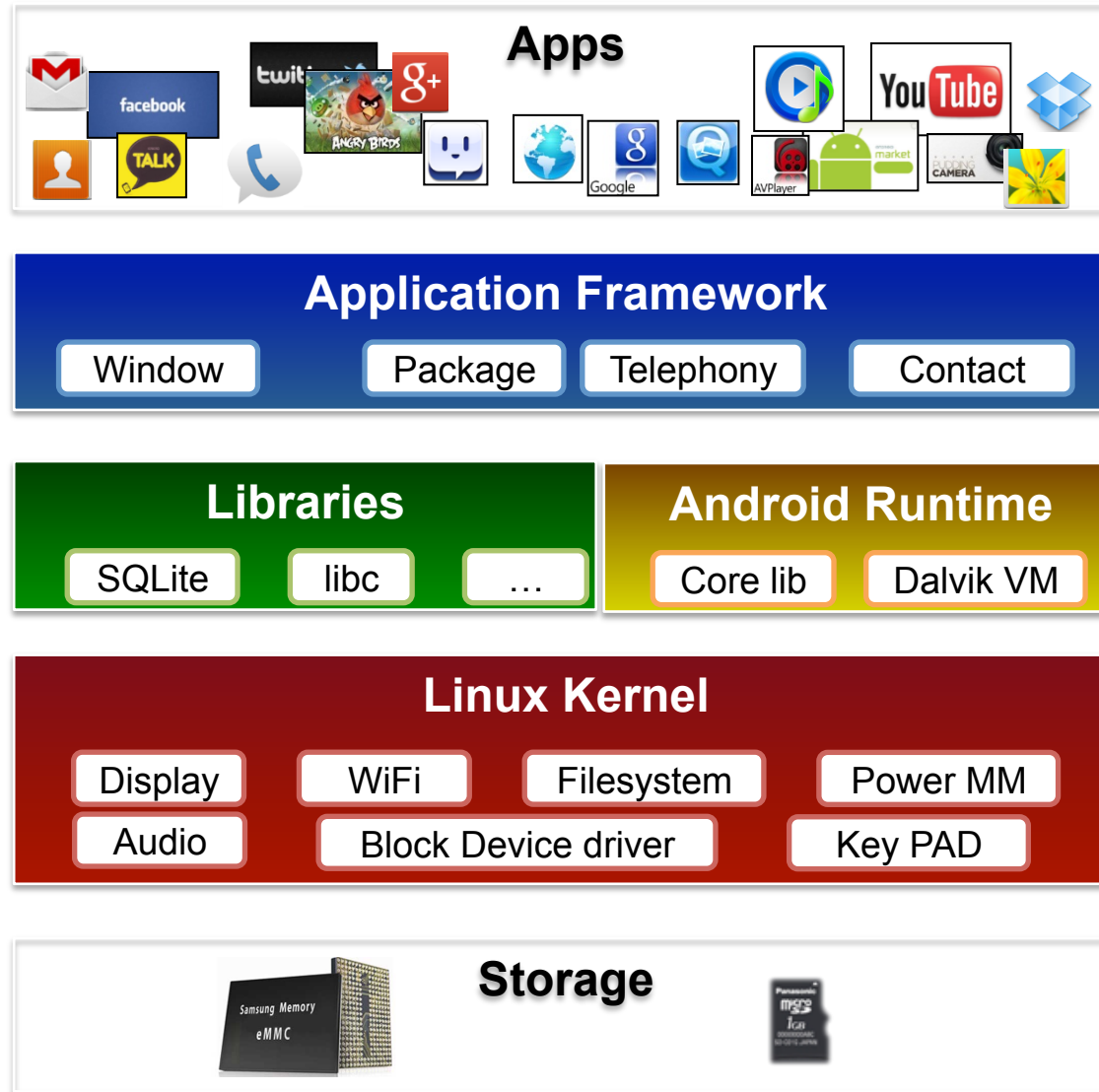
[KPCB Internet Trends 2013]♪

Motivation

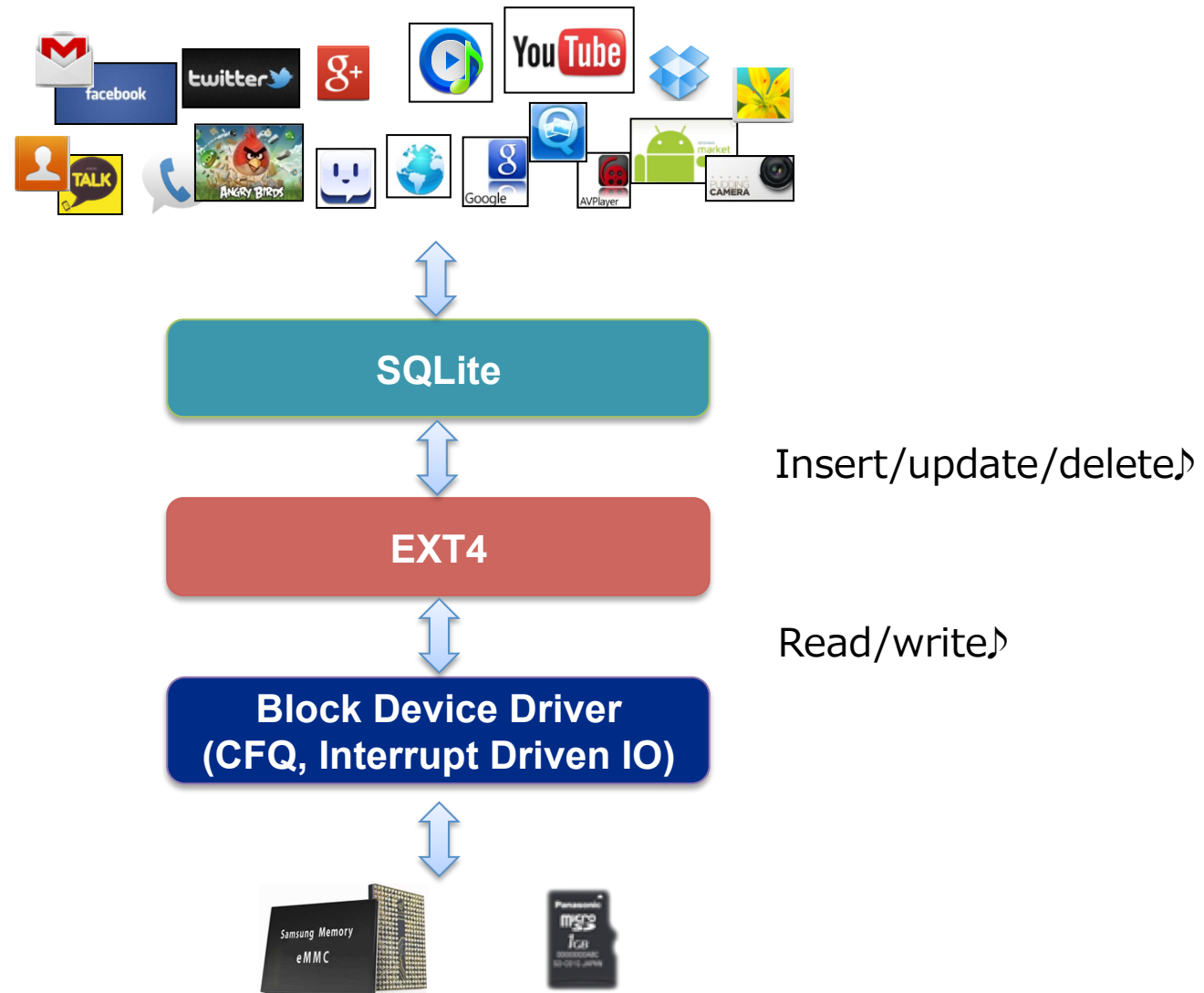
Storage I/O is the performance bottleneck in Android



Android Platform

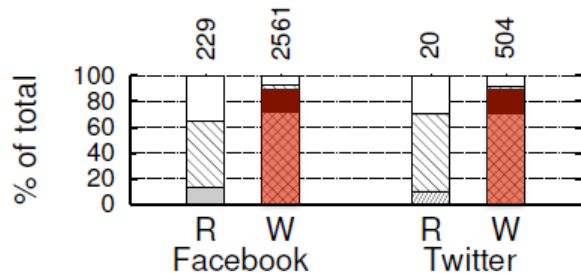


I/O stack of Android Platform



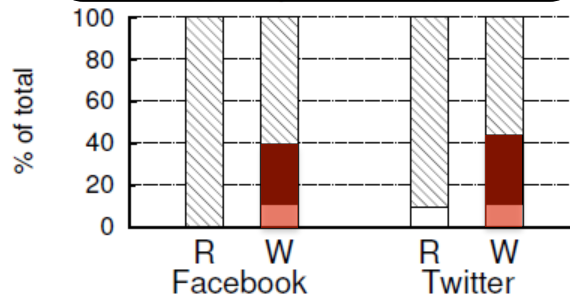
I/O characteristics of Android Apps (GS3, ICS)

SQLite > 90%



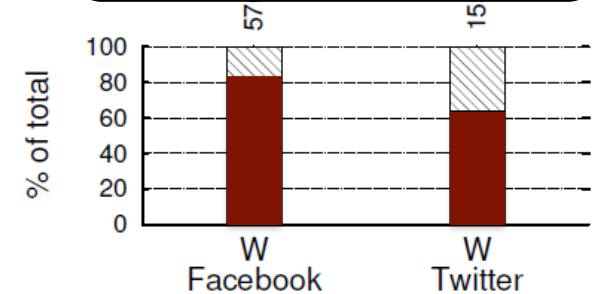
File Types

Metadata & Journal > 40%



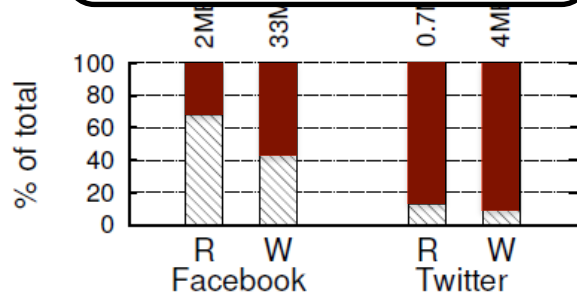
Block Types

Synchronous > 70%



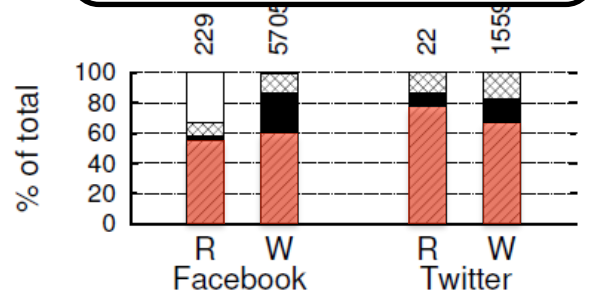
I/O Modes

Random > 80%



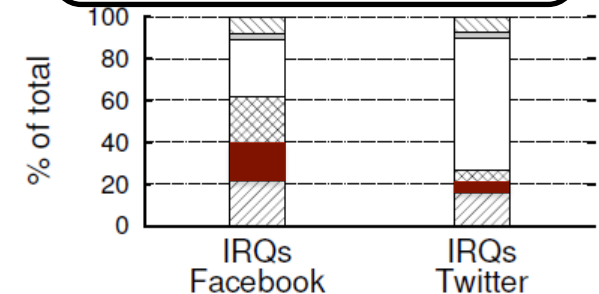
Locality

4KB I/O > 64%



I/O Size

IRQ for eMMC > 18%



IRQs

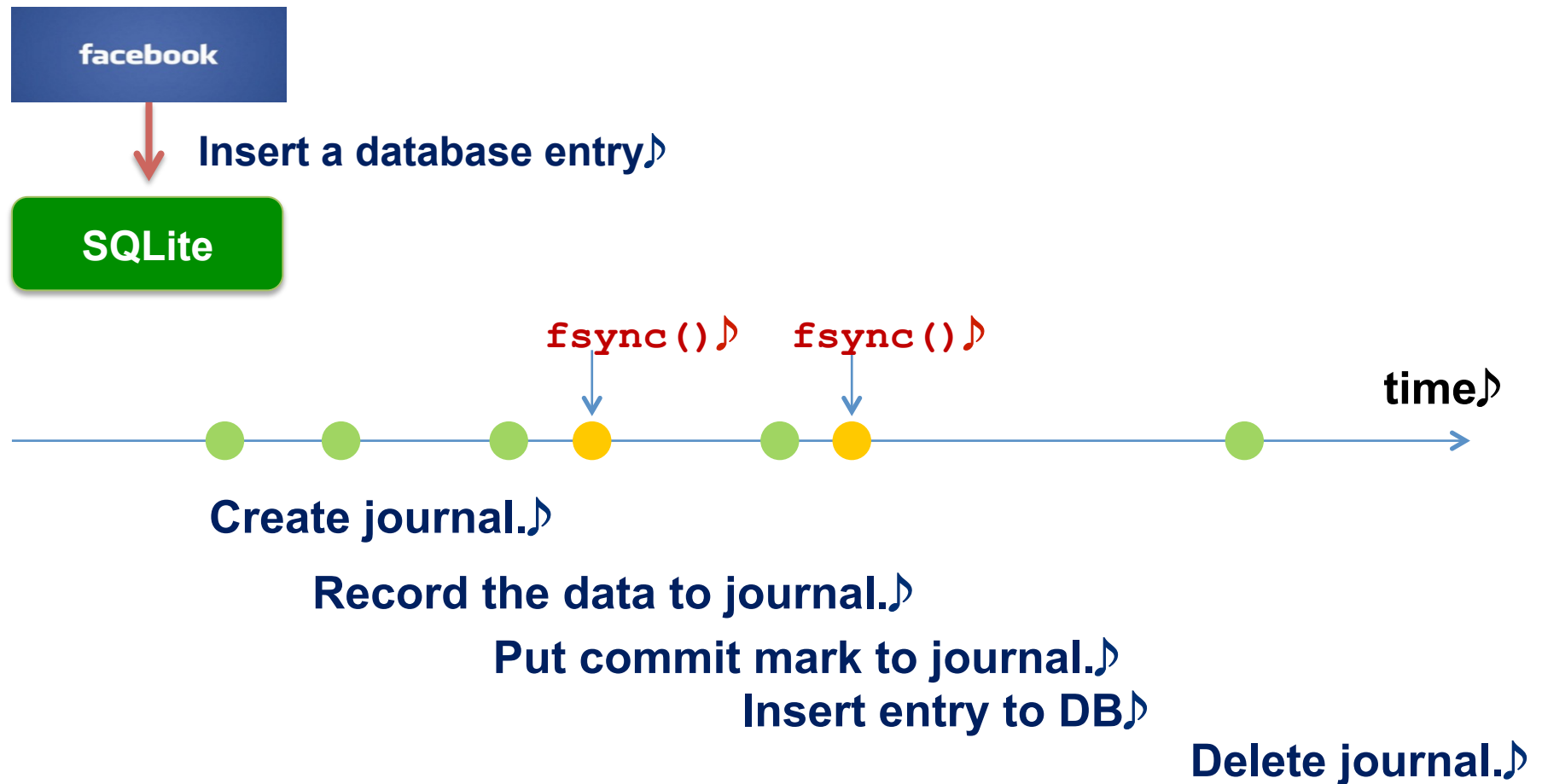
SQLite > 90% !!!

Metadata & Journal > 40% !



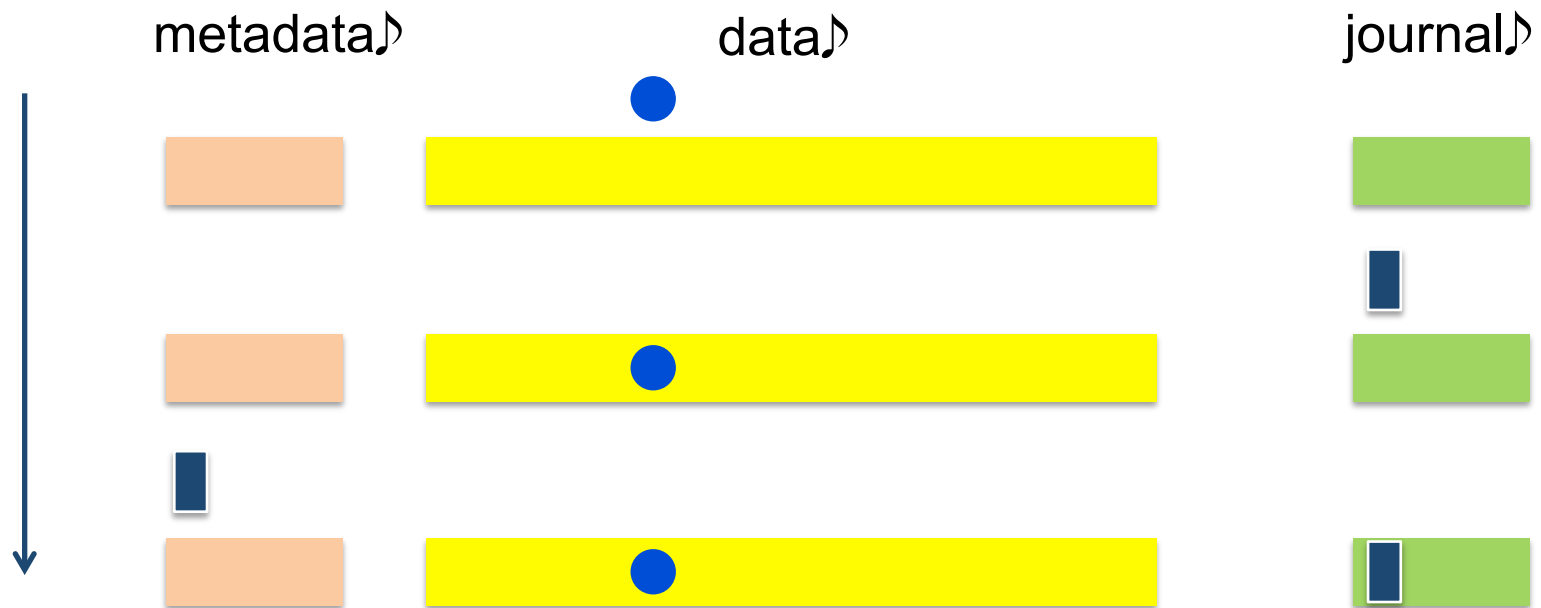
**Synchronous Write > 70% !!
!**

Journaling in SQLite (Delete Mode)

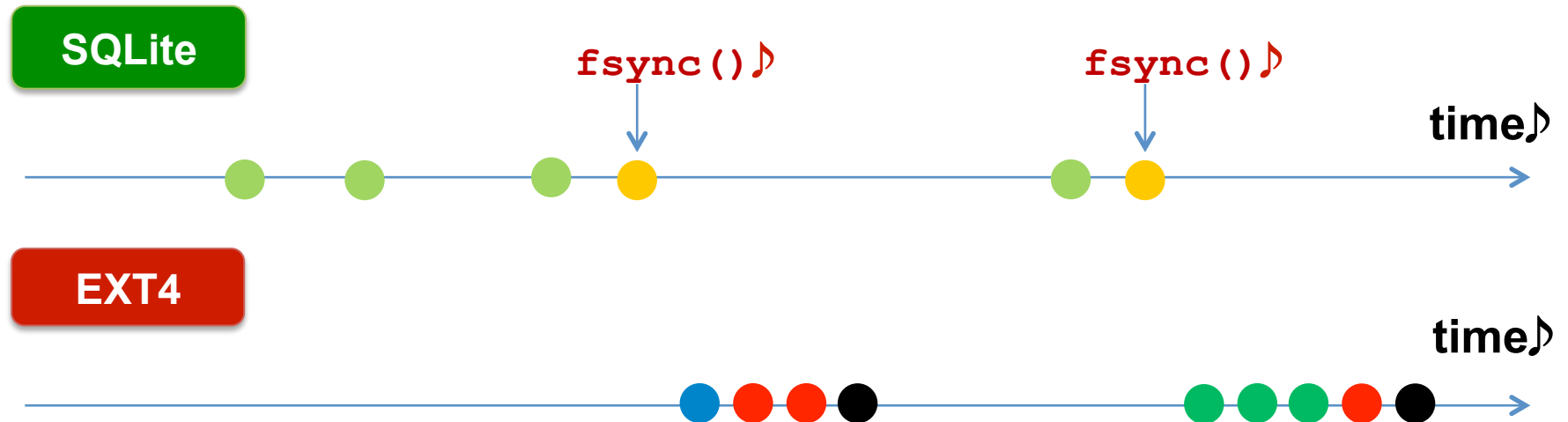


Journaling in EXT4 (ordered mode)

`write(fd, ● )`



SQLite and EXT4



- write SQLite journal to storage.♪
- write SQLite DB to storage.♪
- write EXT4 journal (descriptor, metadata) to storage.♪
- write EXT4 journal (commit) to storage.♪

Summary



Journaling of Journal

SQLi

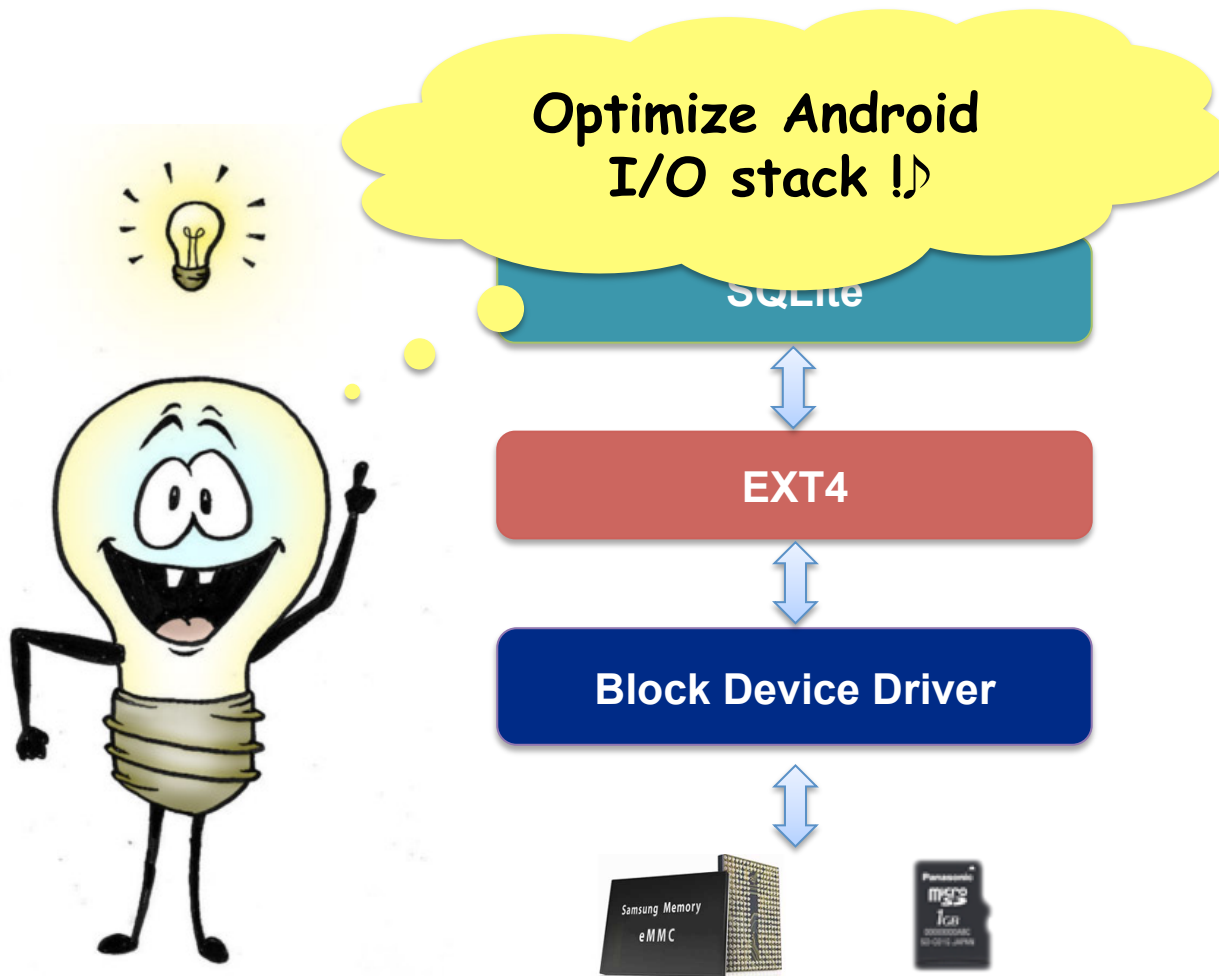


EXT4 jo
70% of the wr

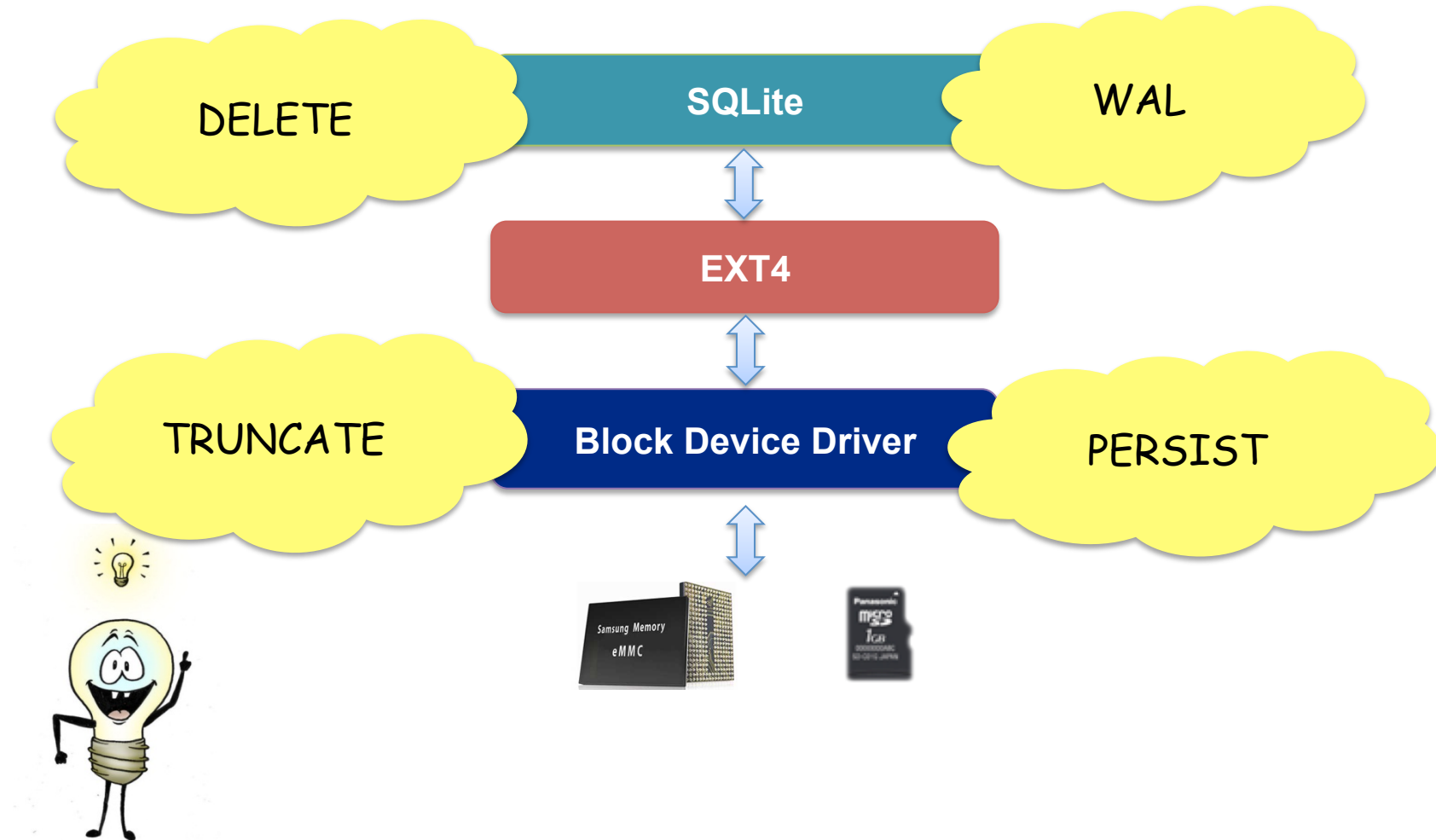
ng activity.
agerial purpose!♪

EXT4 jo

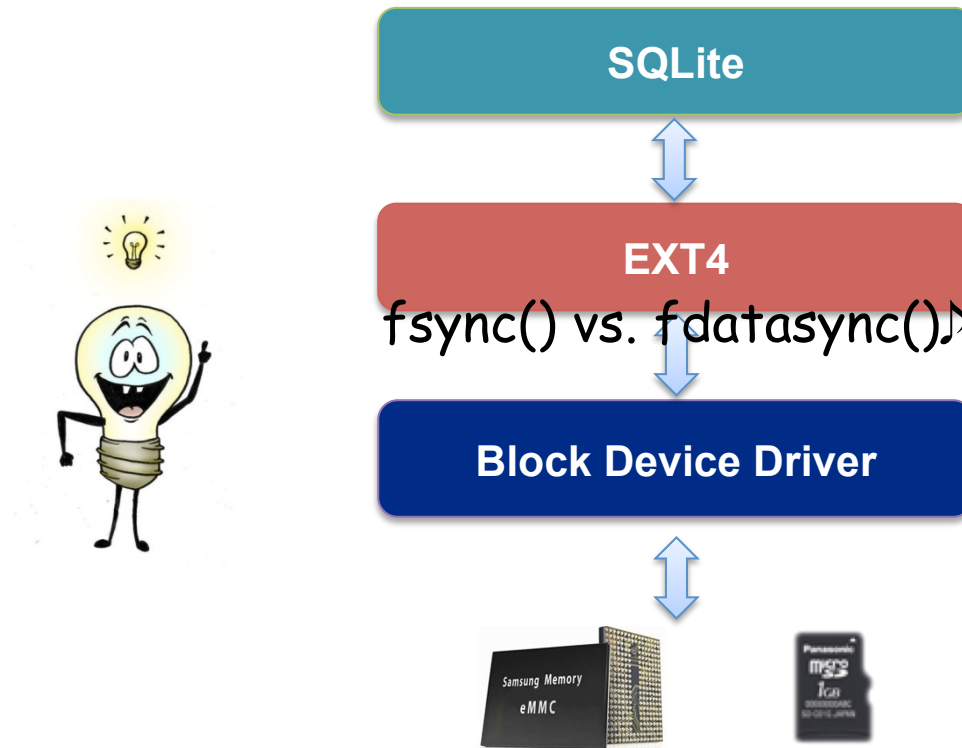




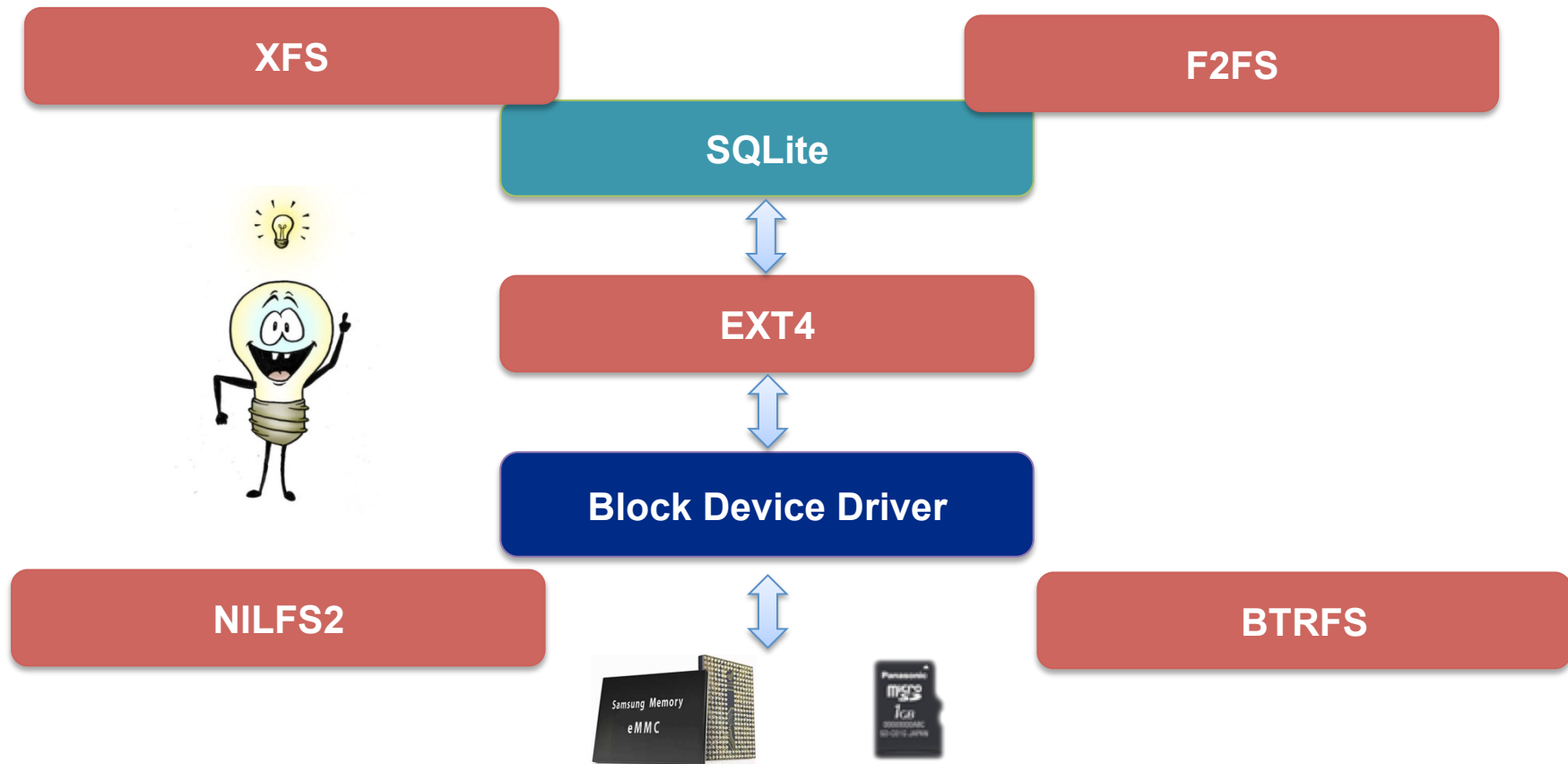
SQLite Journaling mode



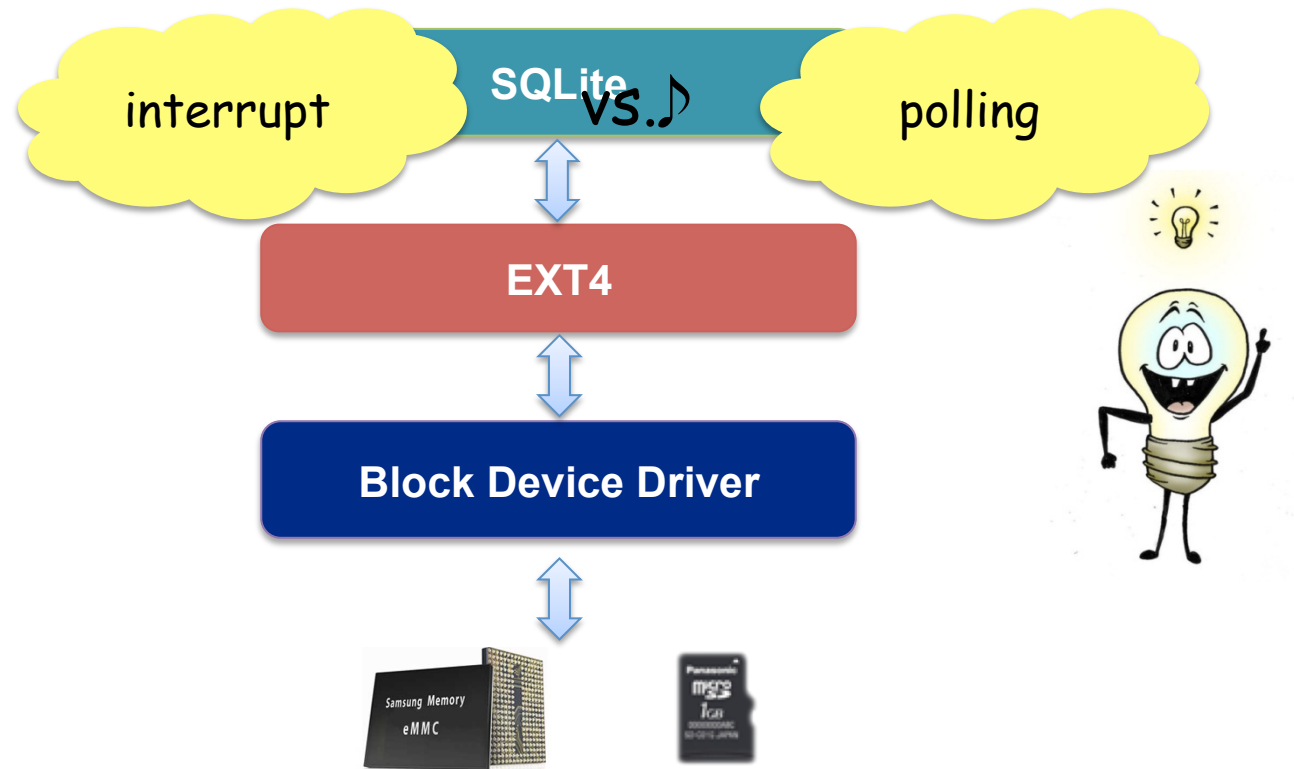
Eliminating unnecessary metadata flushes



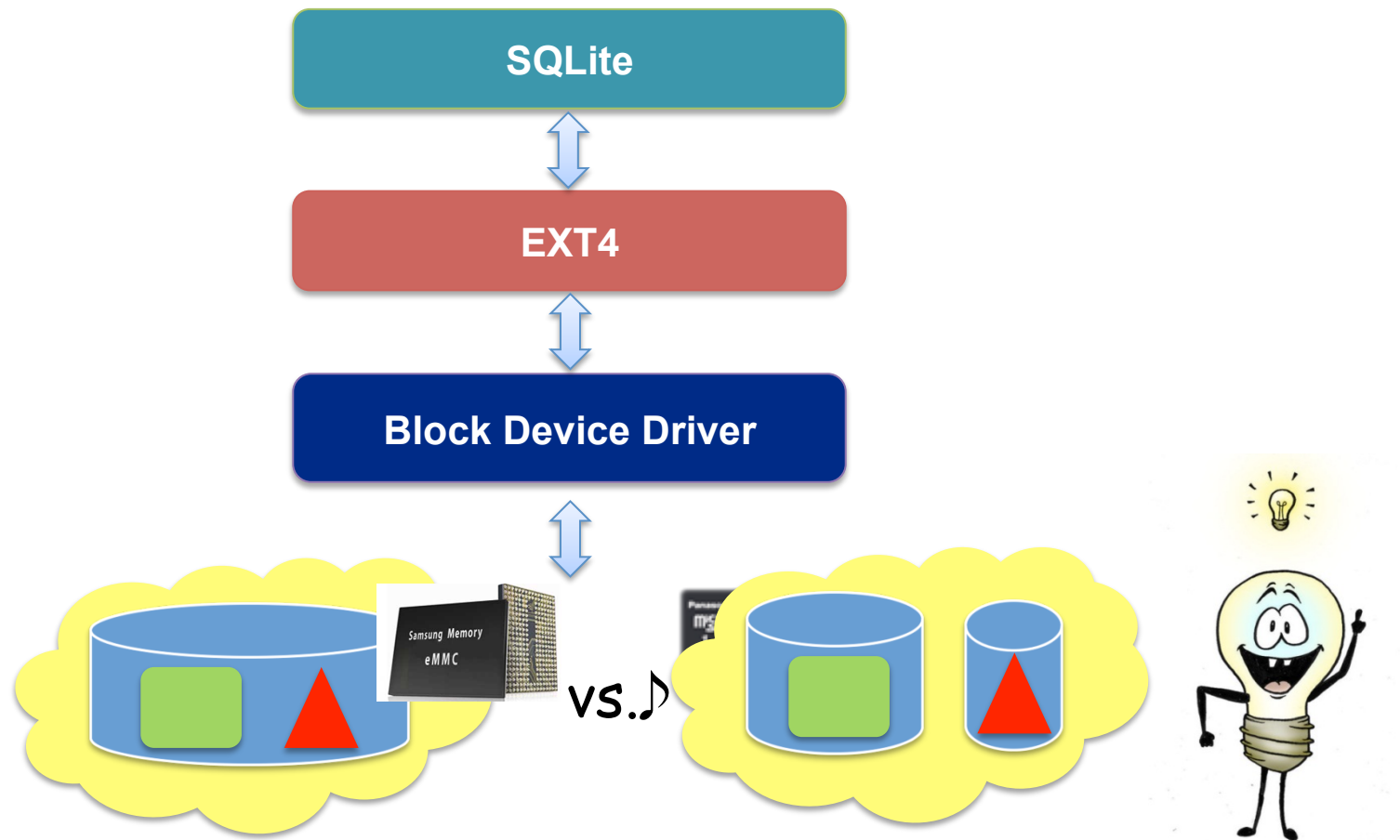
Alternative Filesystems



Interrupt vs. Polling



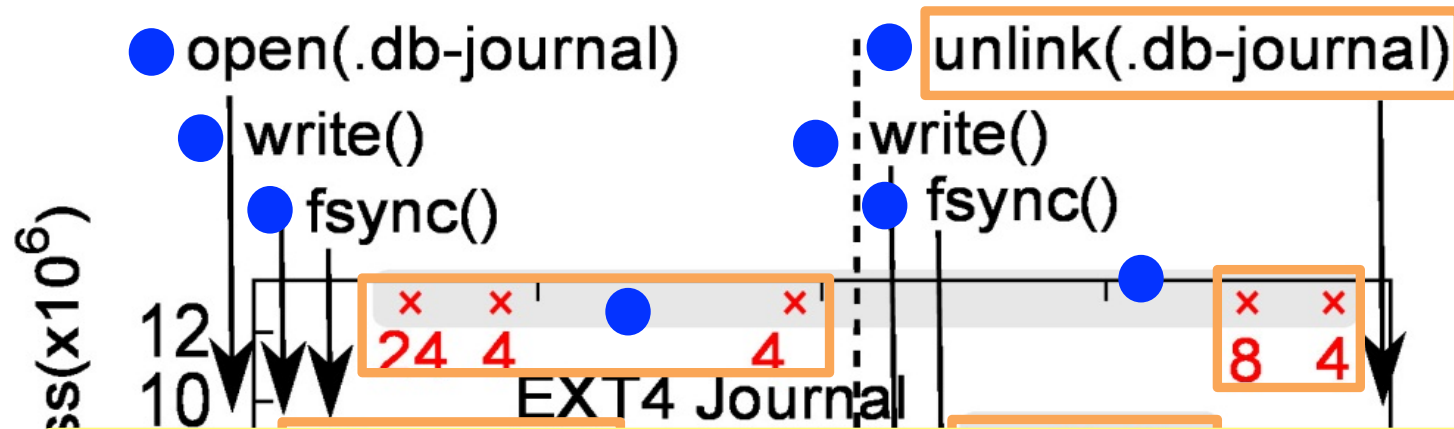
External Journaling



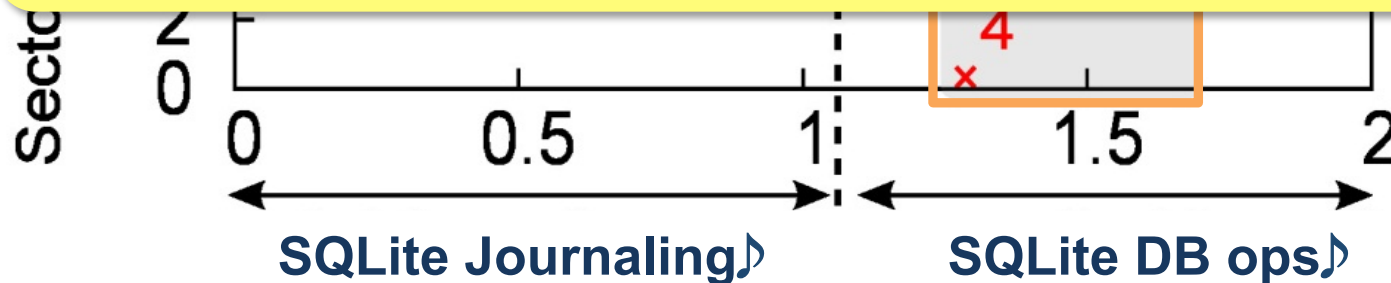


SQLite Journaling Modes

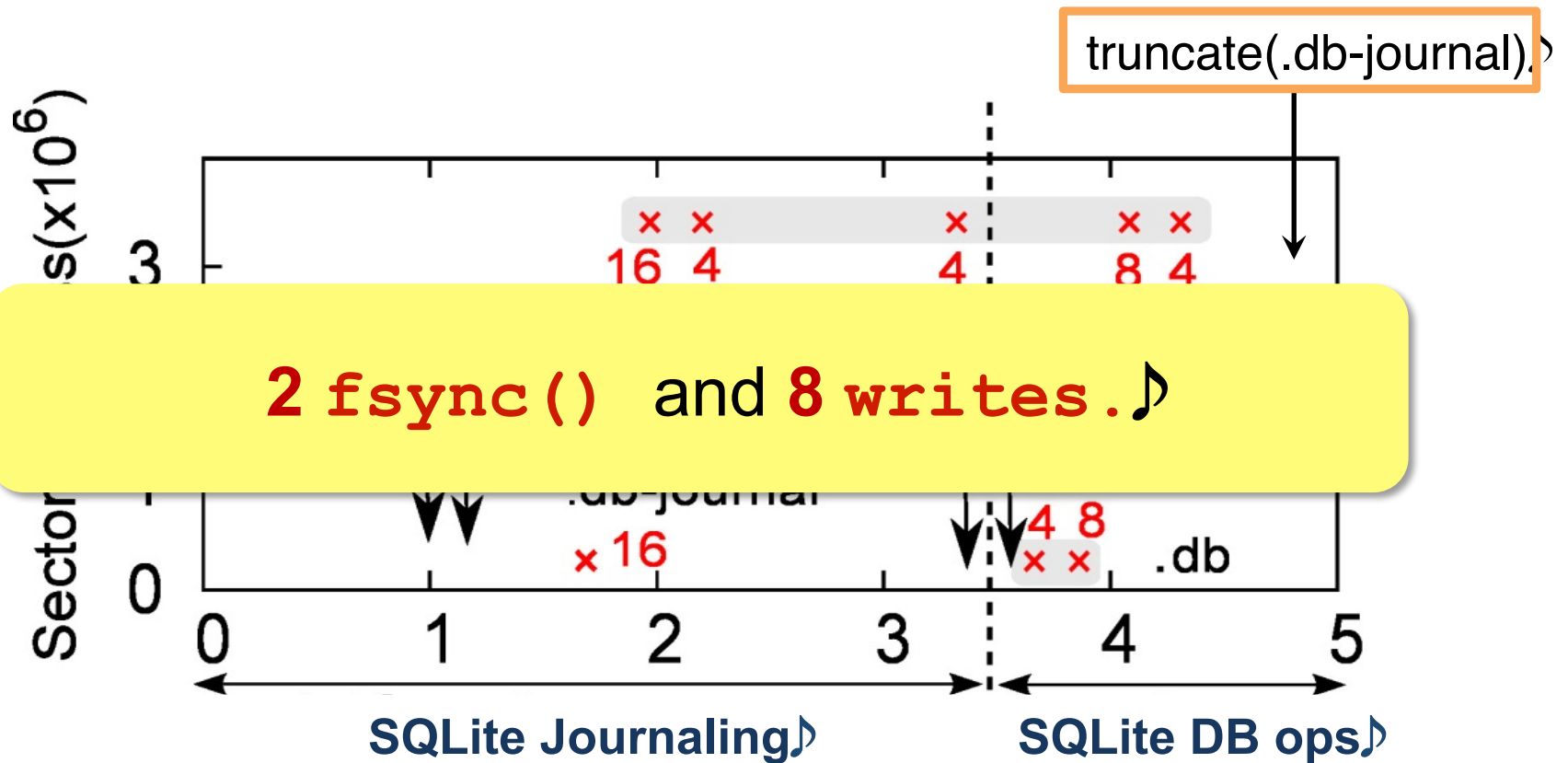
Delete (GS3, ICS)



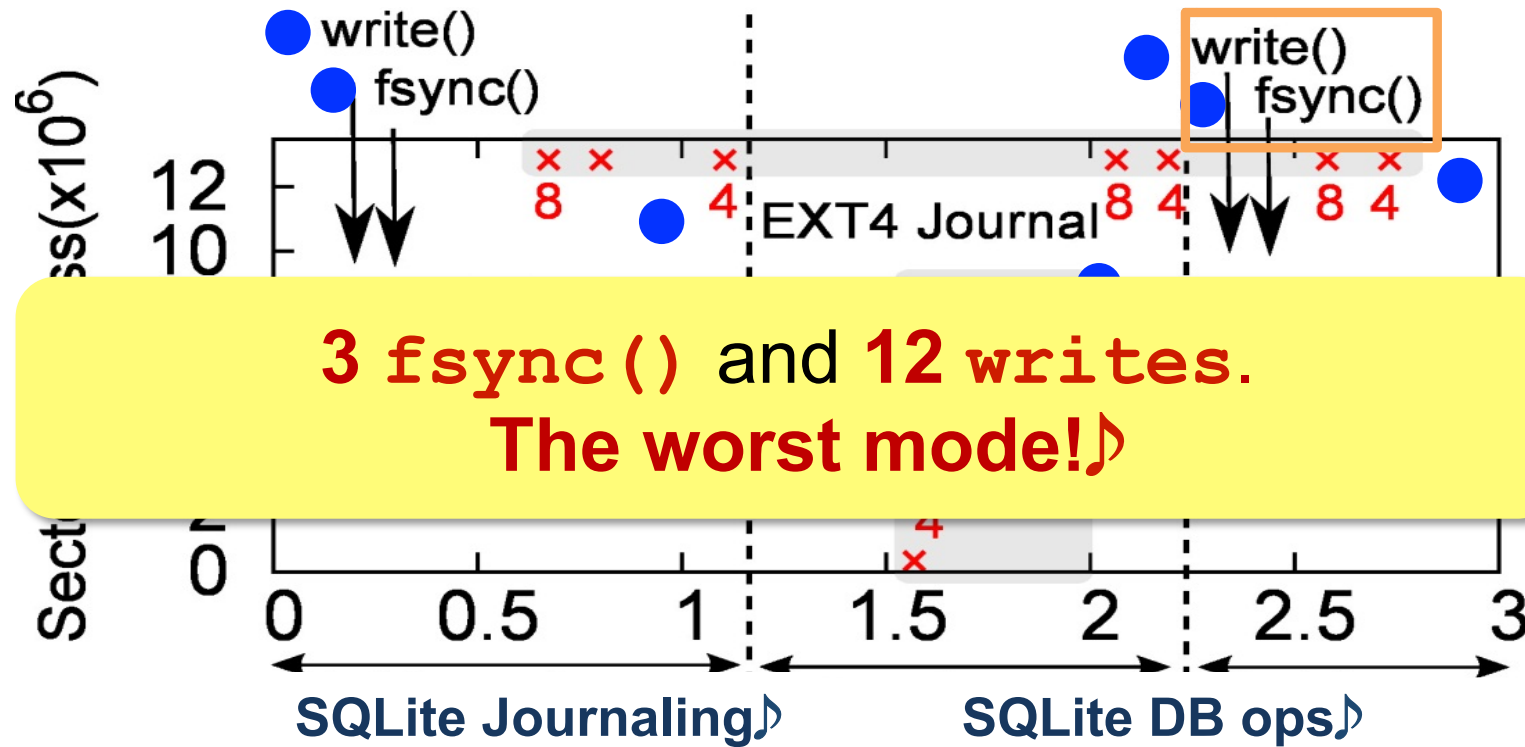
2 fsync() and **9 writes** for one insert() ! 🎵



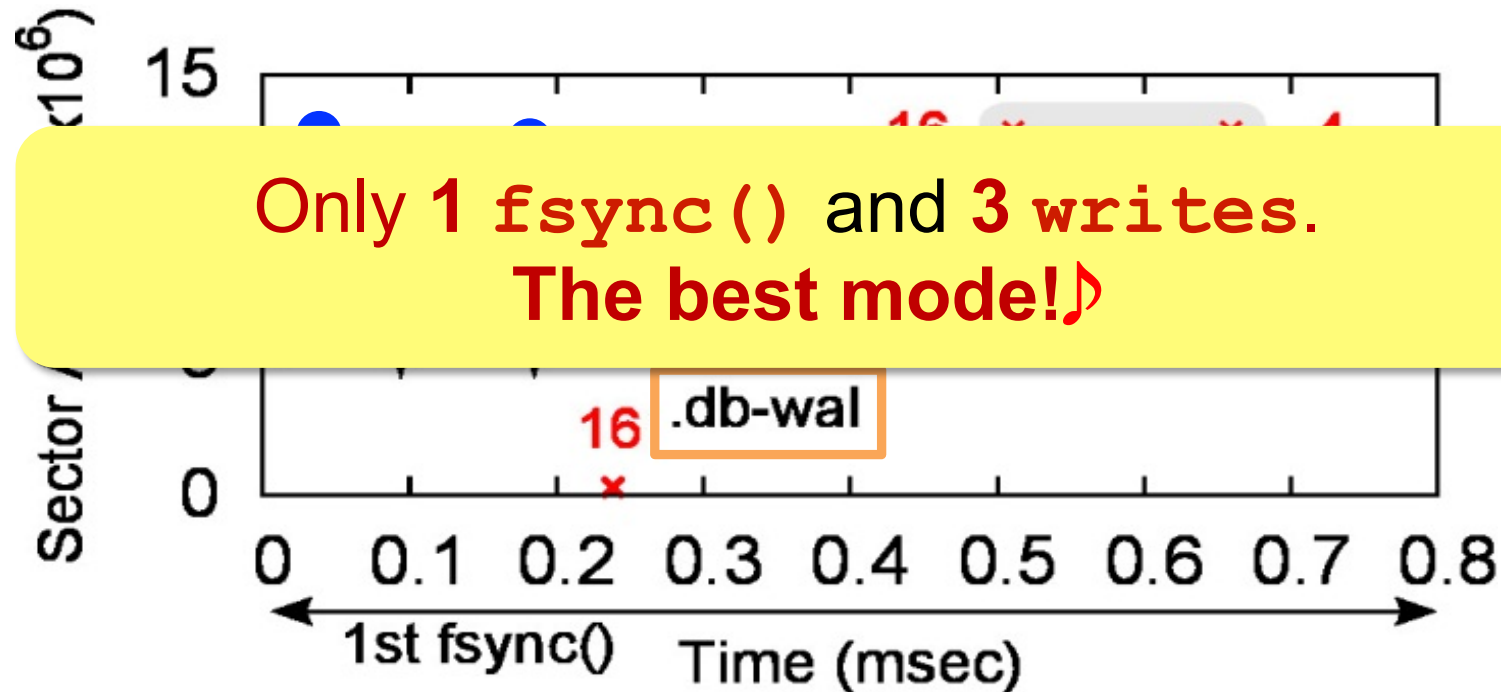
Truncate (GS3, ICS)



Persist (GS3, ICS)



WAL Mode (GS3, ICS)



SQLite Journaling Mode

Summary

SQLite Journaling Mode	DELETE	TRUNCATE	PERSIST	WAL
Number of fsync() calls	2	2	3	1
Number of IOs	9	8	12	3
EXT4 Journal size (metadata)	24 KB	16 KB	8 KB	16 KB
Total IO Volume	72 KB	64 KB	72 KB	36 KB





Filesystems

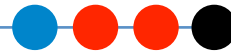
facebook

↓ insert()

EXT4

**“write() followed by fsync()”
is the essence of the Android I/O.**

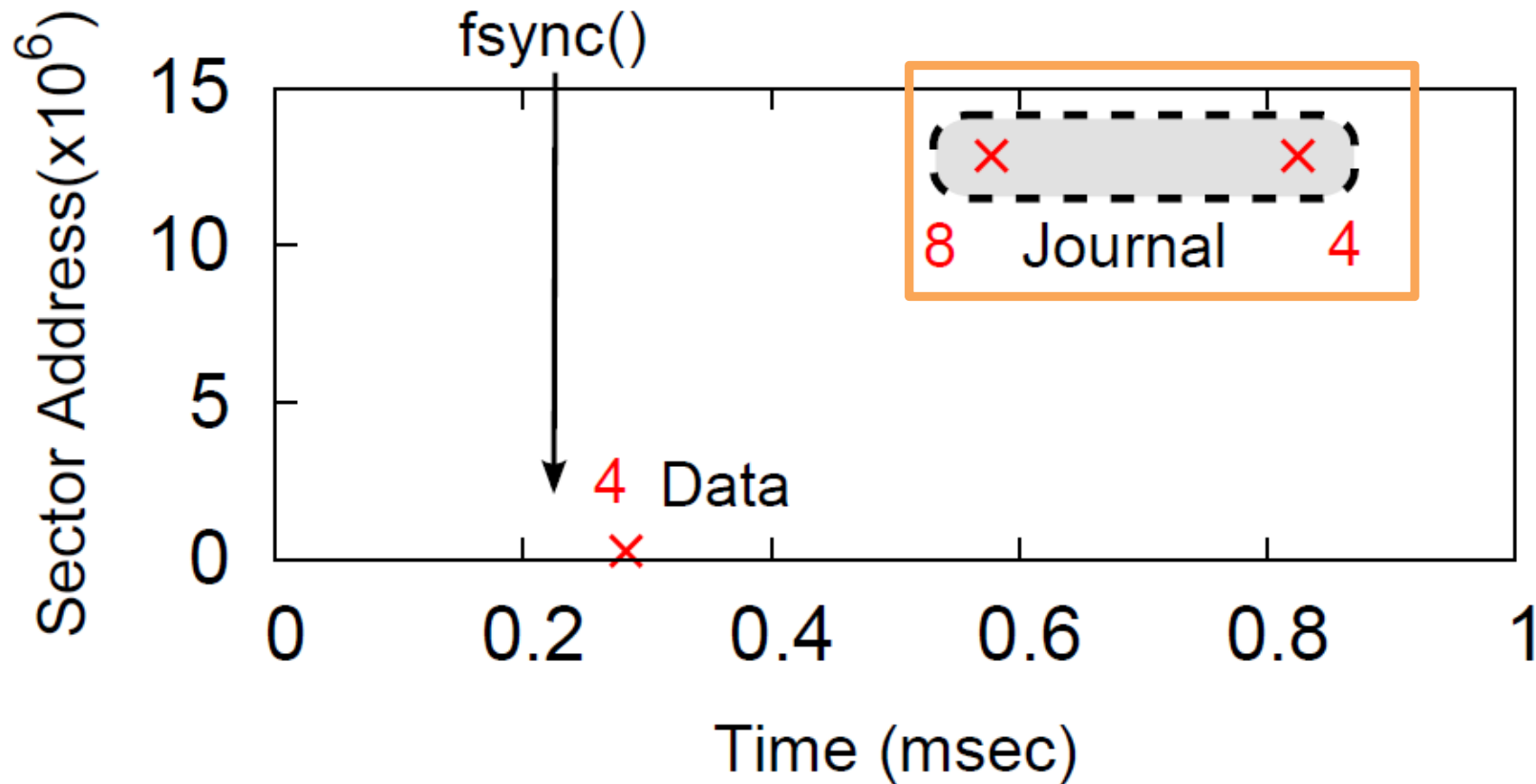
EXT4



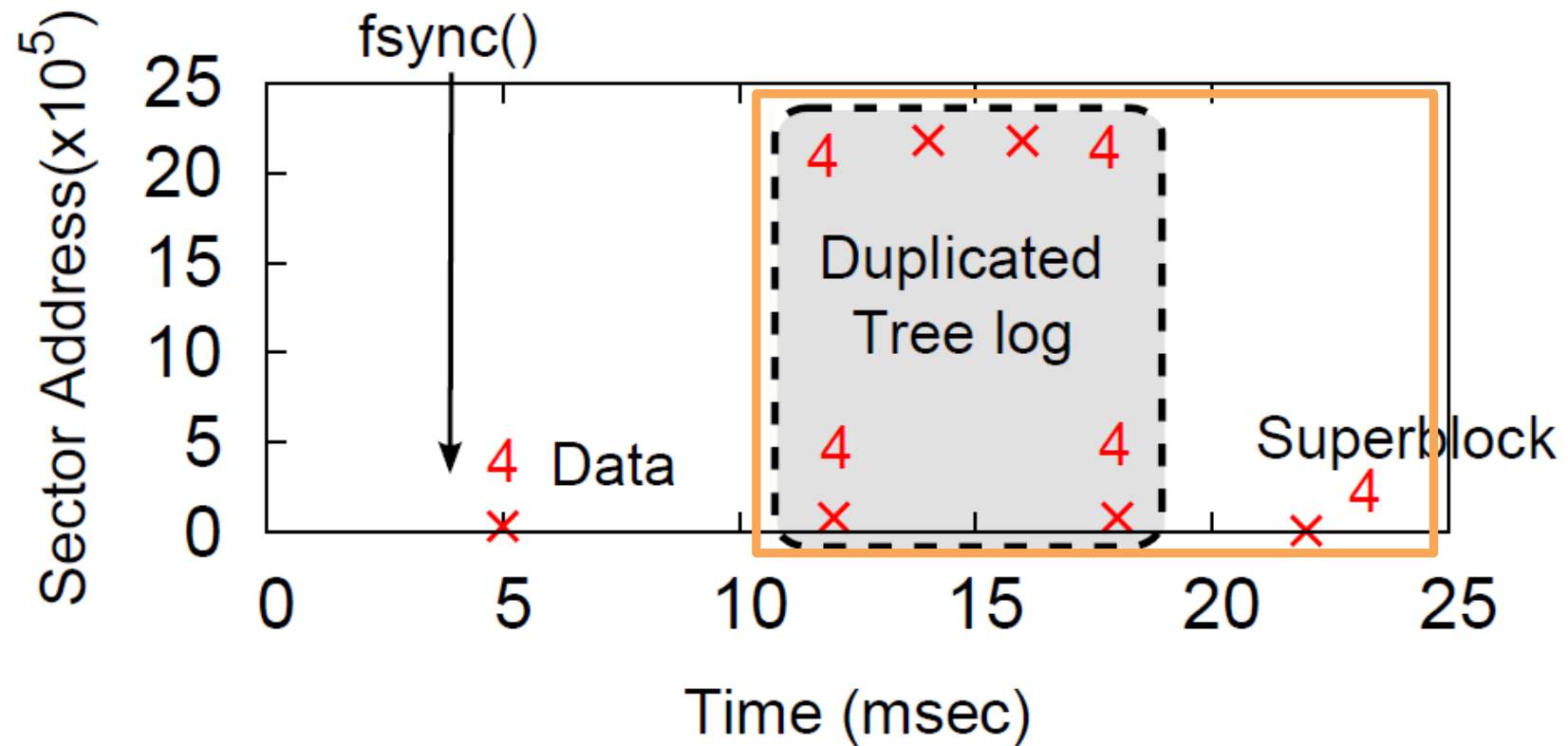
time

EXT4

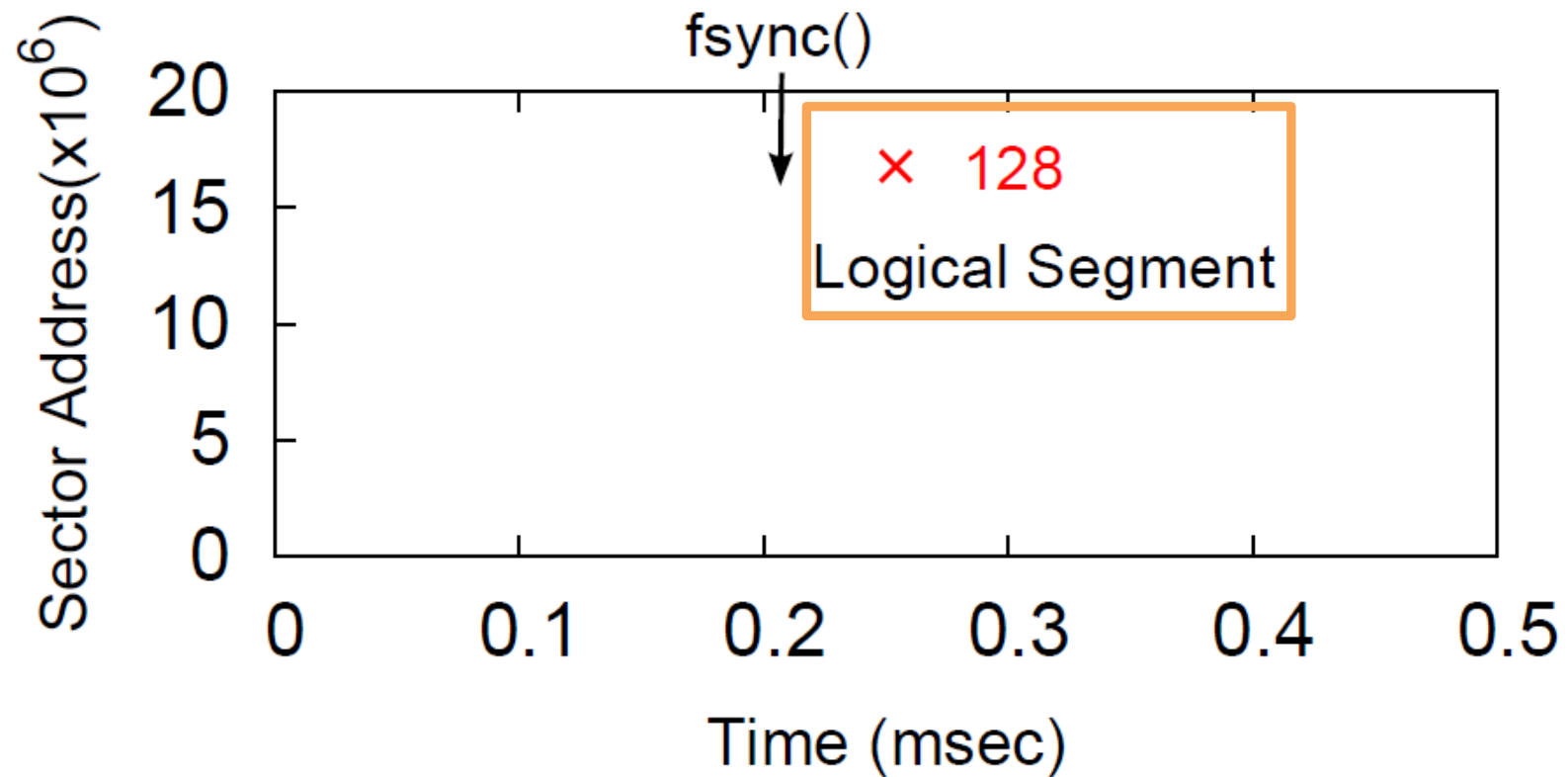
4 KB `write()` followed by `fsync()`



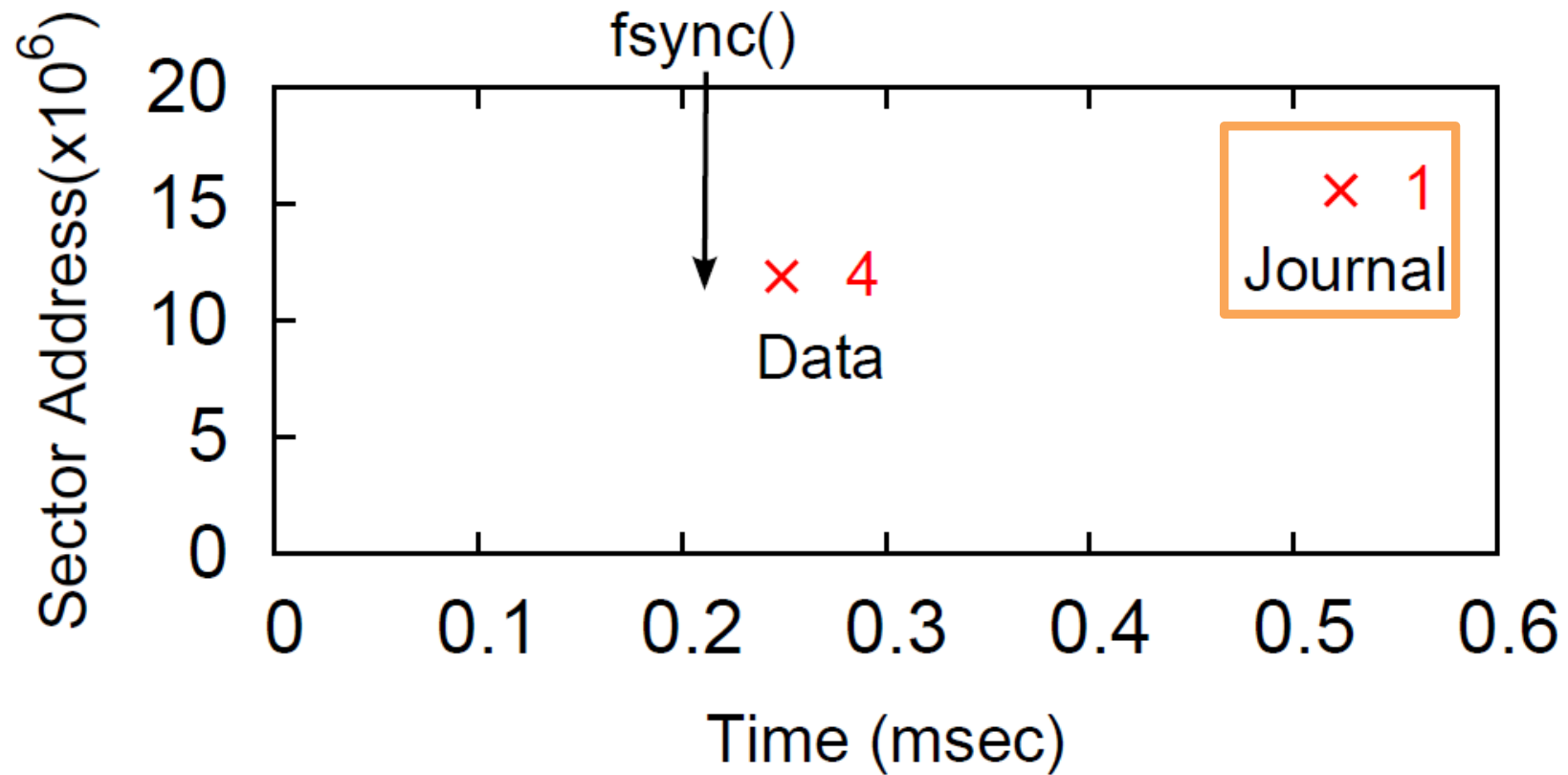
BTRFS



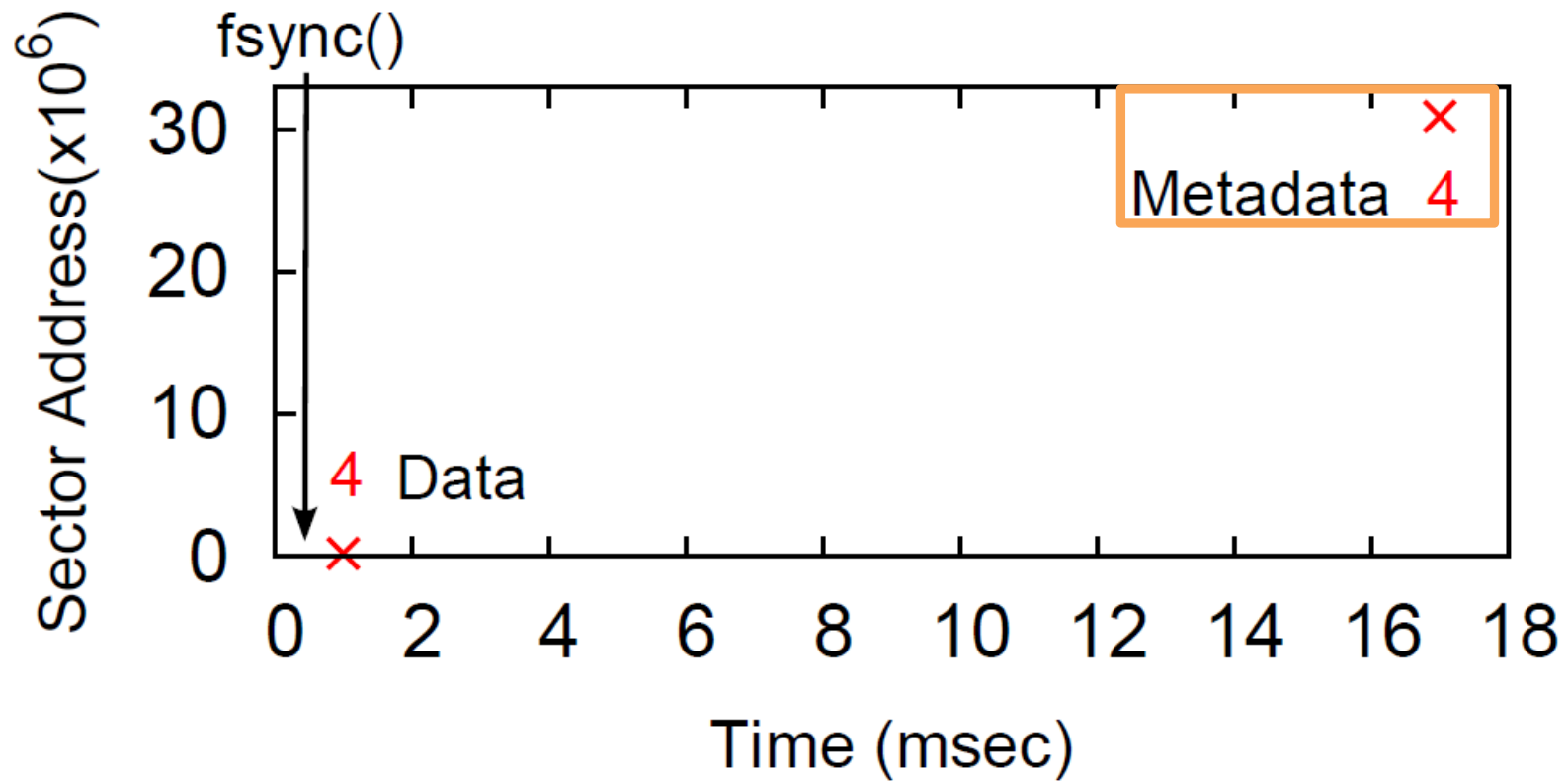
NILFS2



XFS

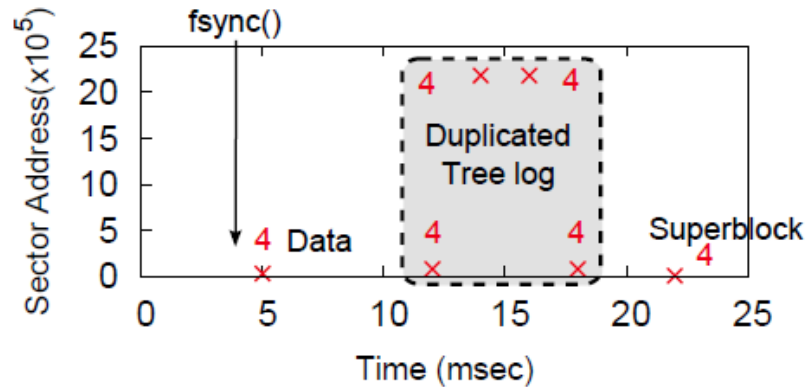


F2FS (Flash Friendly Filesystem)

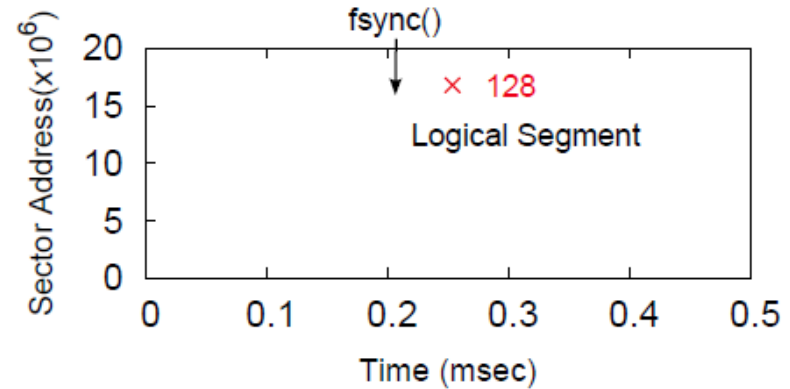


write () followed by fsync ()

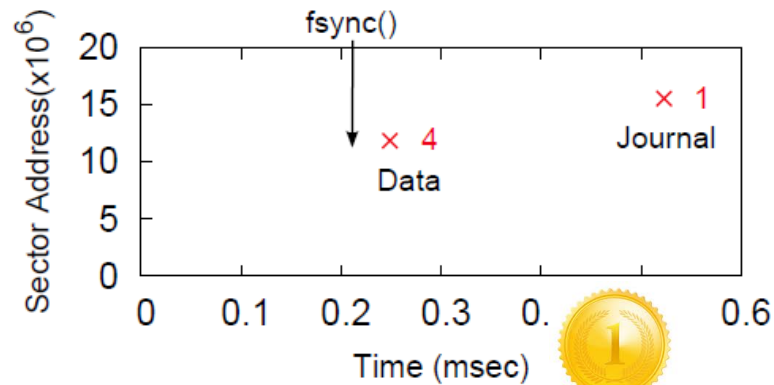
Summary



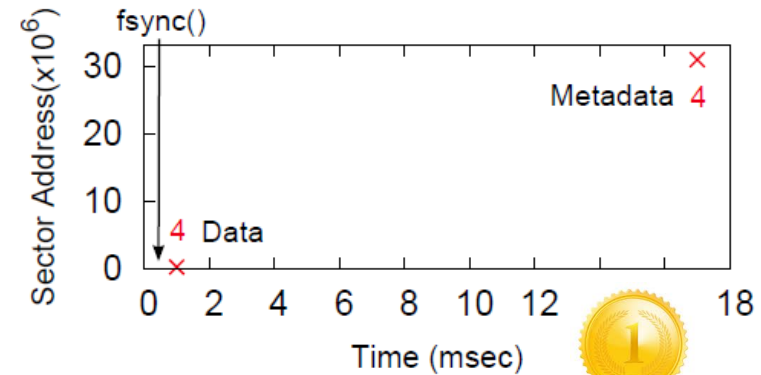
BTRFS



NILFS2



XFS



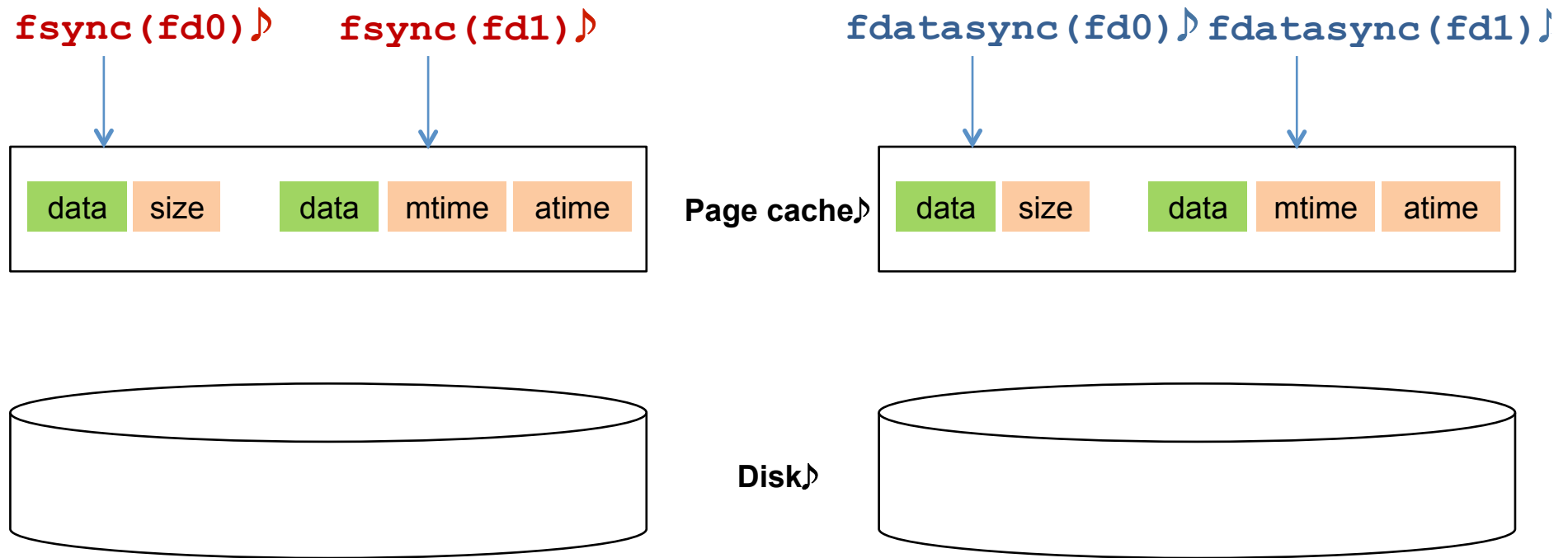
F2FS





`fsync()` vs. `fdatasync()`

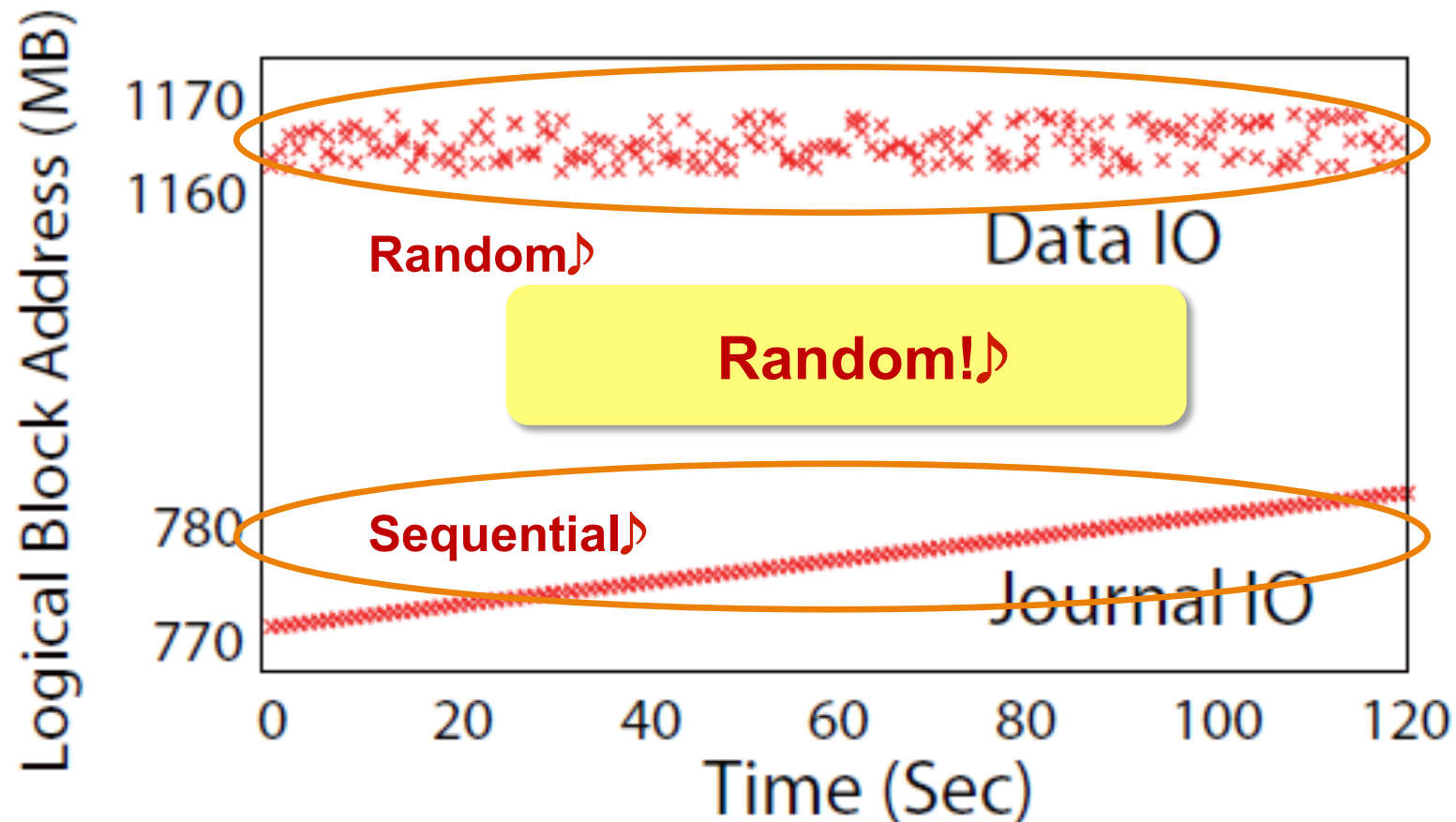
Eliminating Unnecessary Metadata Flushes



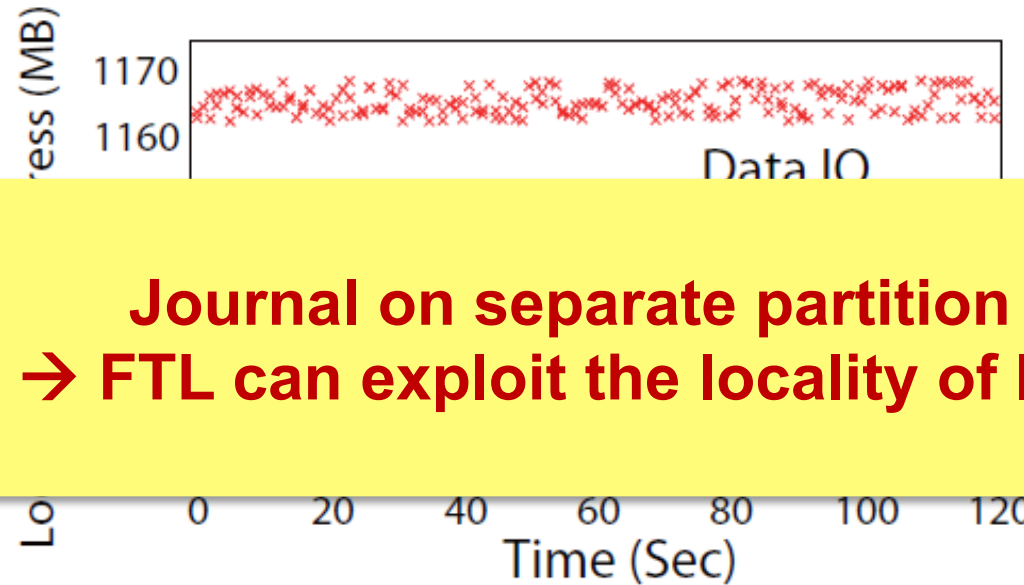


External Journaling

4K random write() followed by fsync()

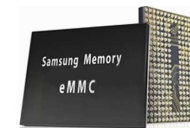
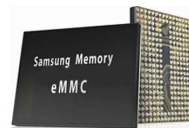


External journaling



Journal on separate partition
→ FTL can exploit the locality of I/O!♪

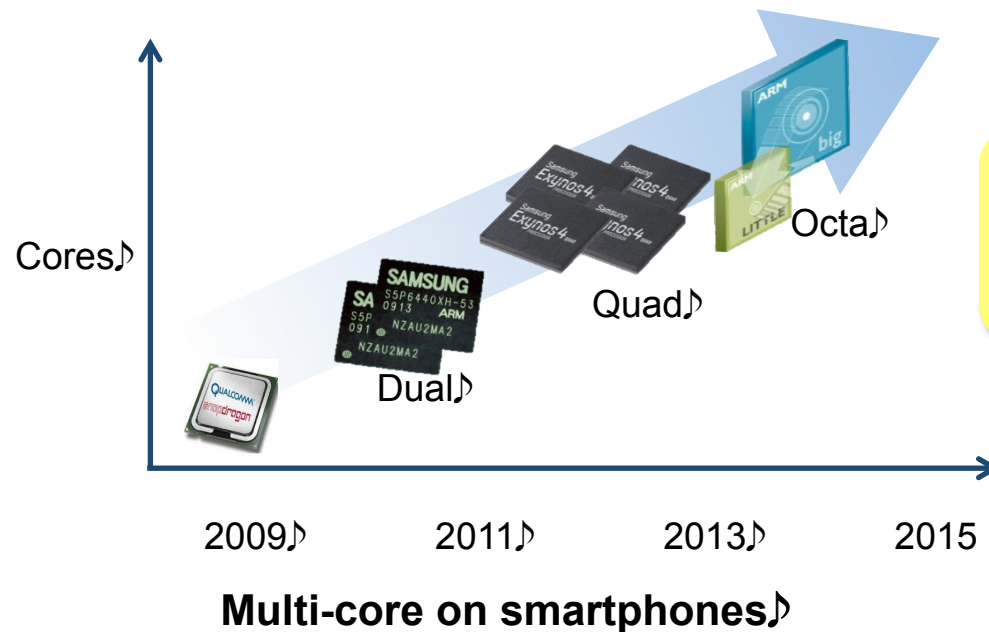
sequential♪





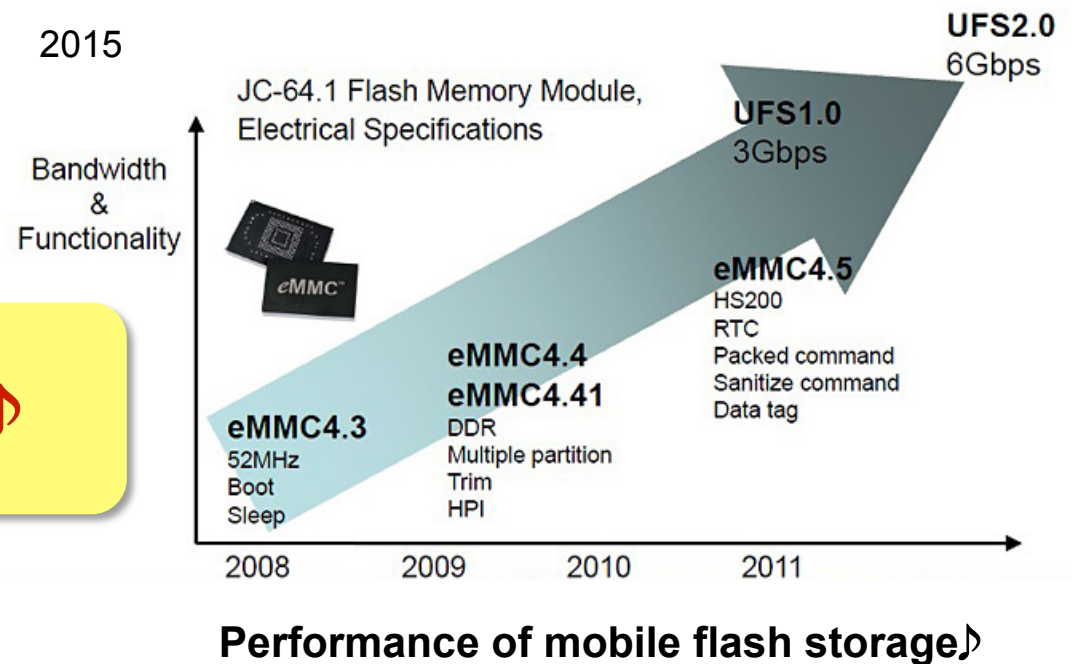
Interrupt driven I/O vs. Polling based I/O

Hardware trend

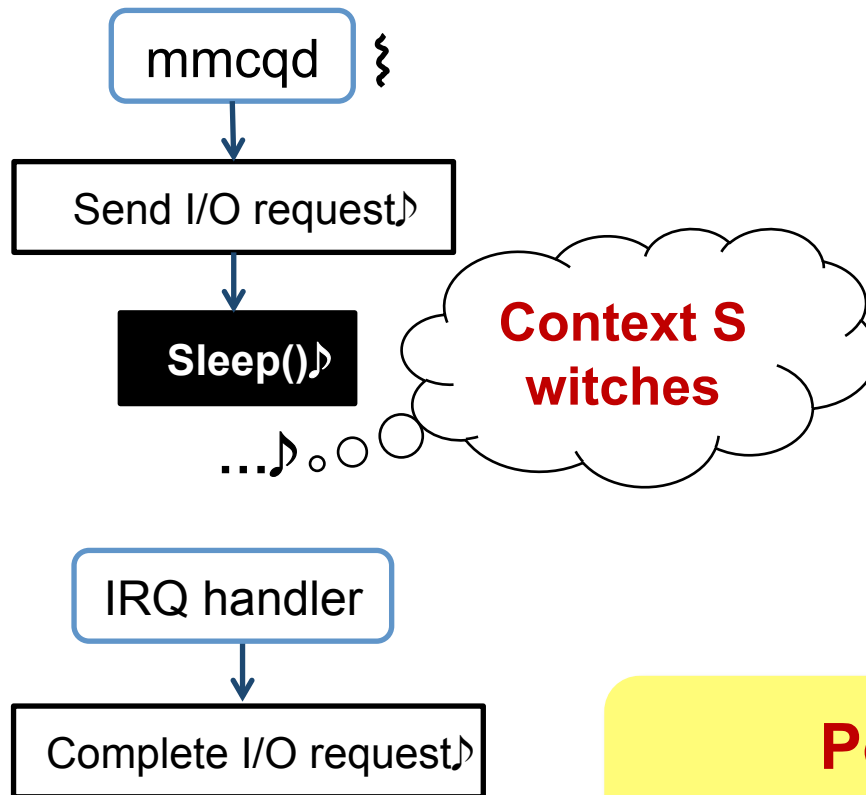


The number of CPU cores ↑

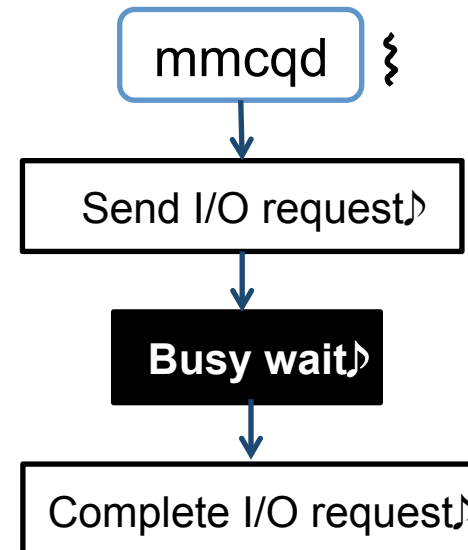
I/O latency of eMMC ↓



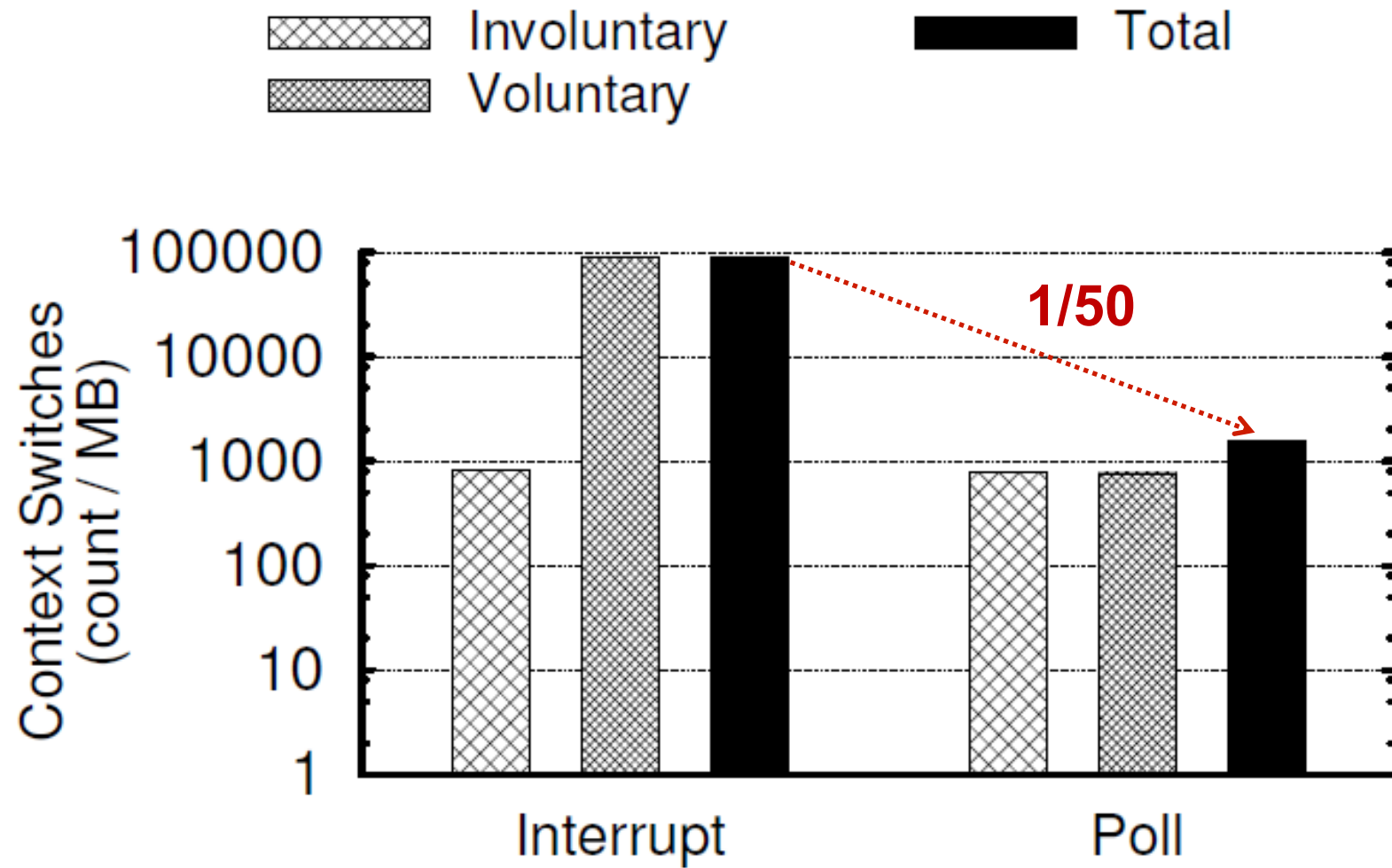
Interrupt driven I/O



Polling based I/O



Polling can reduce context switching overhead!





Experiment

Implementation

Galaxy S3(ICS 4.0.4, Linux 3.0.15)



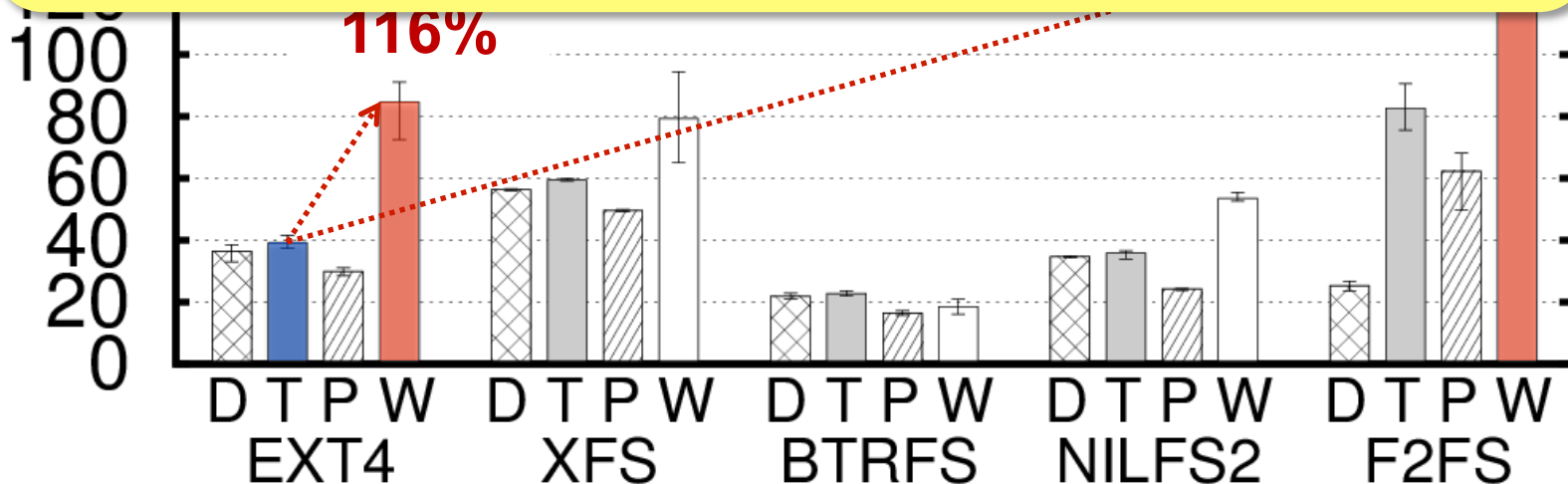
Component♪	Specification♪
CPU♪	Exynos 4412 1.4 GHz Quad-core♪
RAM♪	2 GB♪
Internal Storage♪	32 GB eMMC♪
External Storage♪	16 GB Transcend u-SD Card♪

SQLite performance: journaling modes

SQLite Insert

- TRUNCATE(default) → WAL : 116% up
- TRUNCATE, EXT4(default) → WAL,F2FS: 281% up♪

Insert / sec

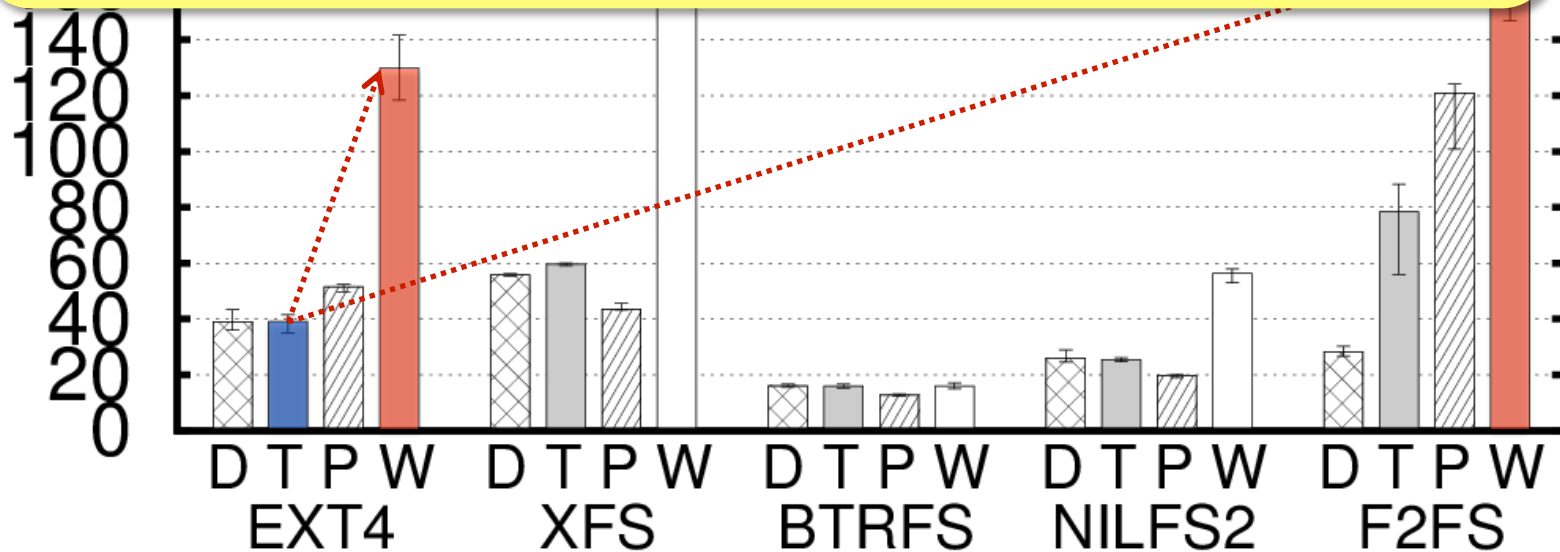


SQLite performance: journaling modes

SQLite Update

- TRUNCATE(default) → WAL : 232% up
- TRUNCATE, EXT4(default) → WAL,F2FS: 348% up♪

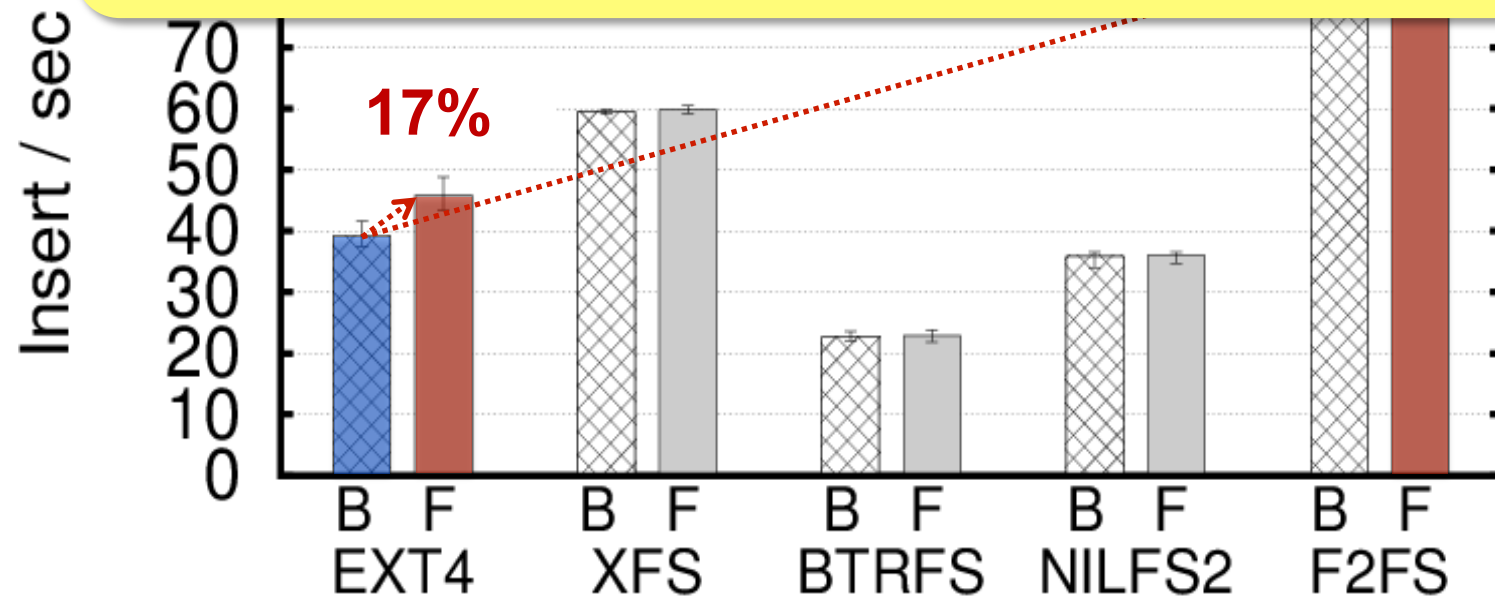
Update / sec



fsync() vs. fdatasync()

SQLite Insert

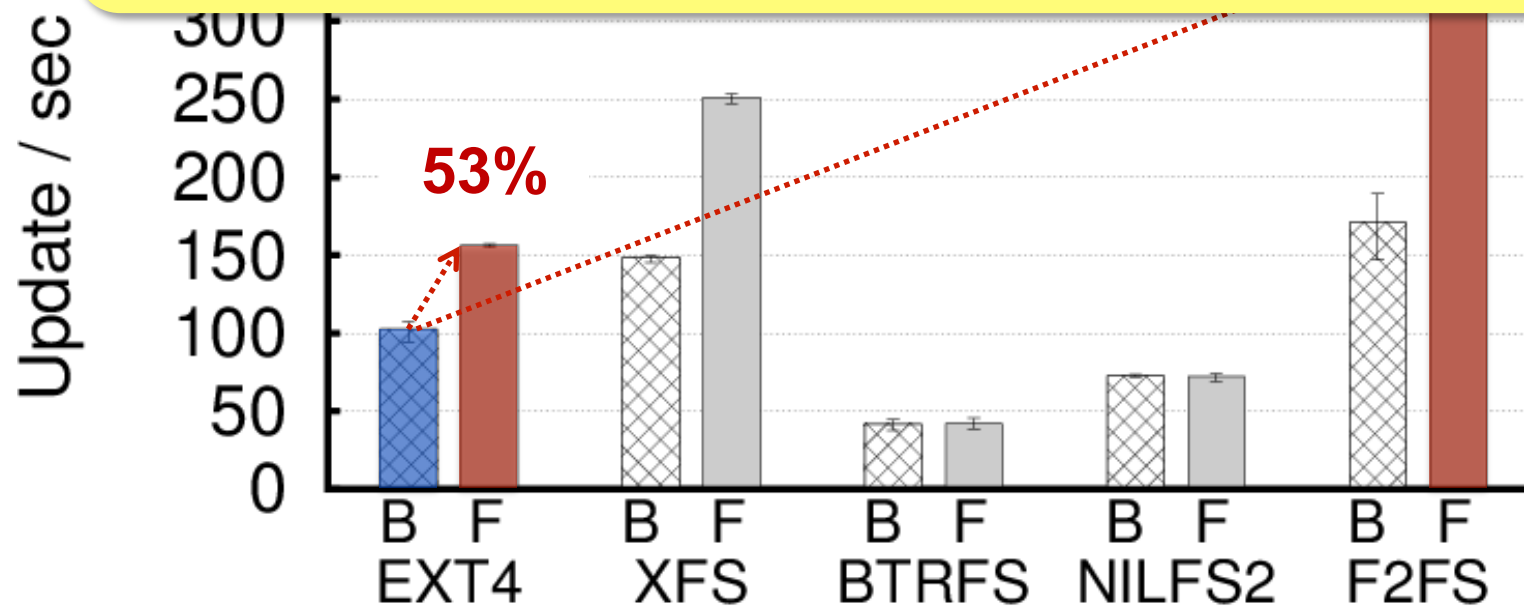
- fsync() → fdatasync() : 17% up
- fsync() → fdatasync() and F2FS : 126% up♪



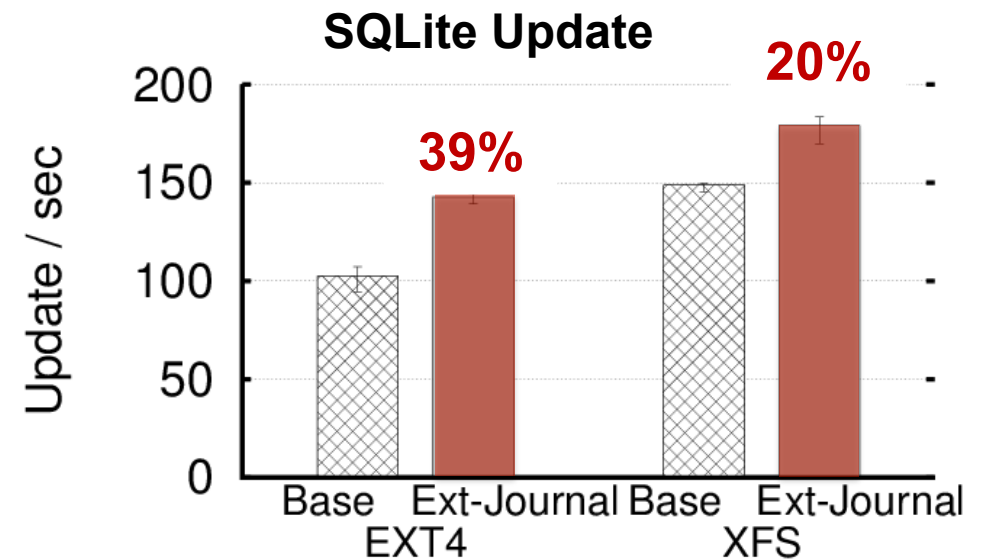
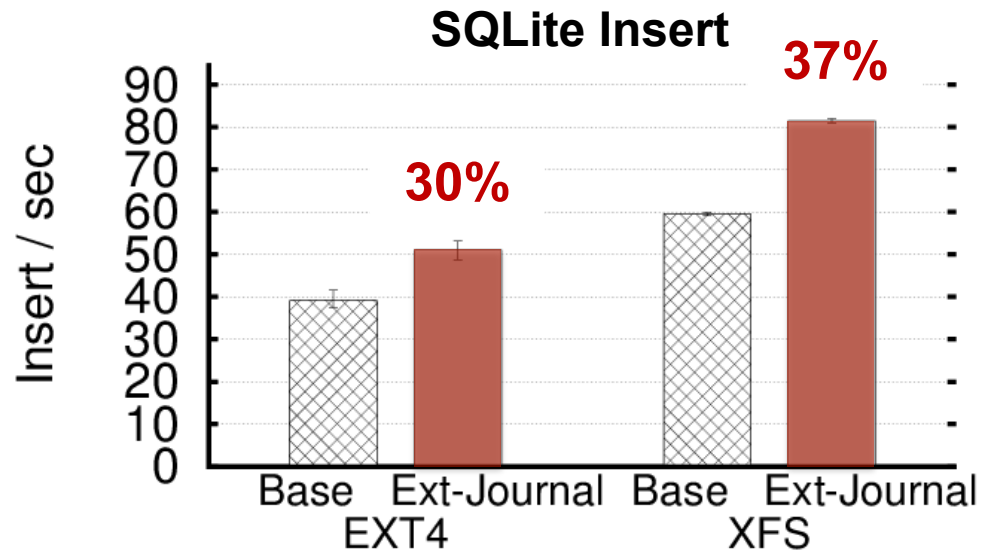
fsync() vs. fdatasync()

SQLite Update

- fsync() → fdatasync() : 53% up
- fsync() → fdatasync() and F2FS : 250% up



External journaling



Polling

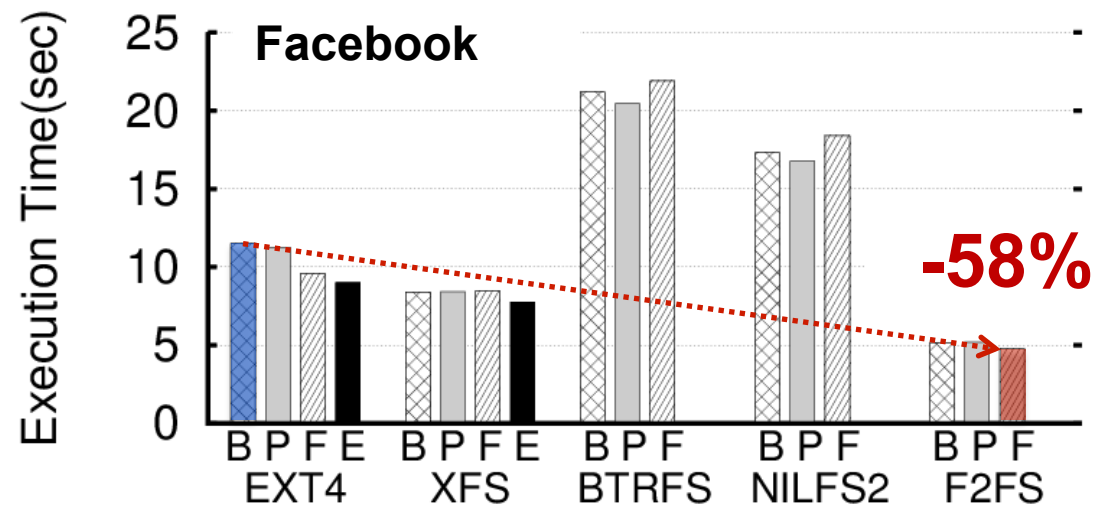
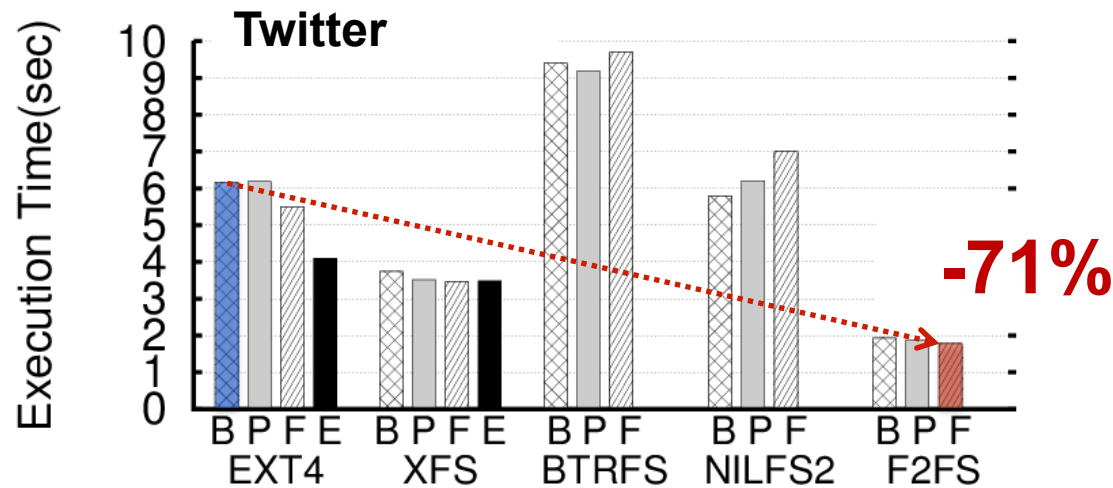
4 KB random write+fsync()

# of thread♪	Scenario♪	Idle♪		HD Record♪	
		base♪	poll♪	base♪	poll♪
1♪	KIOPS♪	1002♪	981♪	667♪	756♪
	CPU (%)♪	7.5♪	10.9♪	26.4♪	30.2♪
10♪	KIOPS♪	2609♪	2705♪	2136♪	2351♪
	CPU (%)♪	11.1♪	12.9♪	30.1♪	33.1♪

- Marginal gain (1~2%) when CPU is IDLE.
- 13% gain when we record HD video in background.

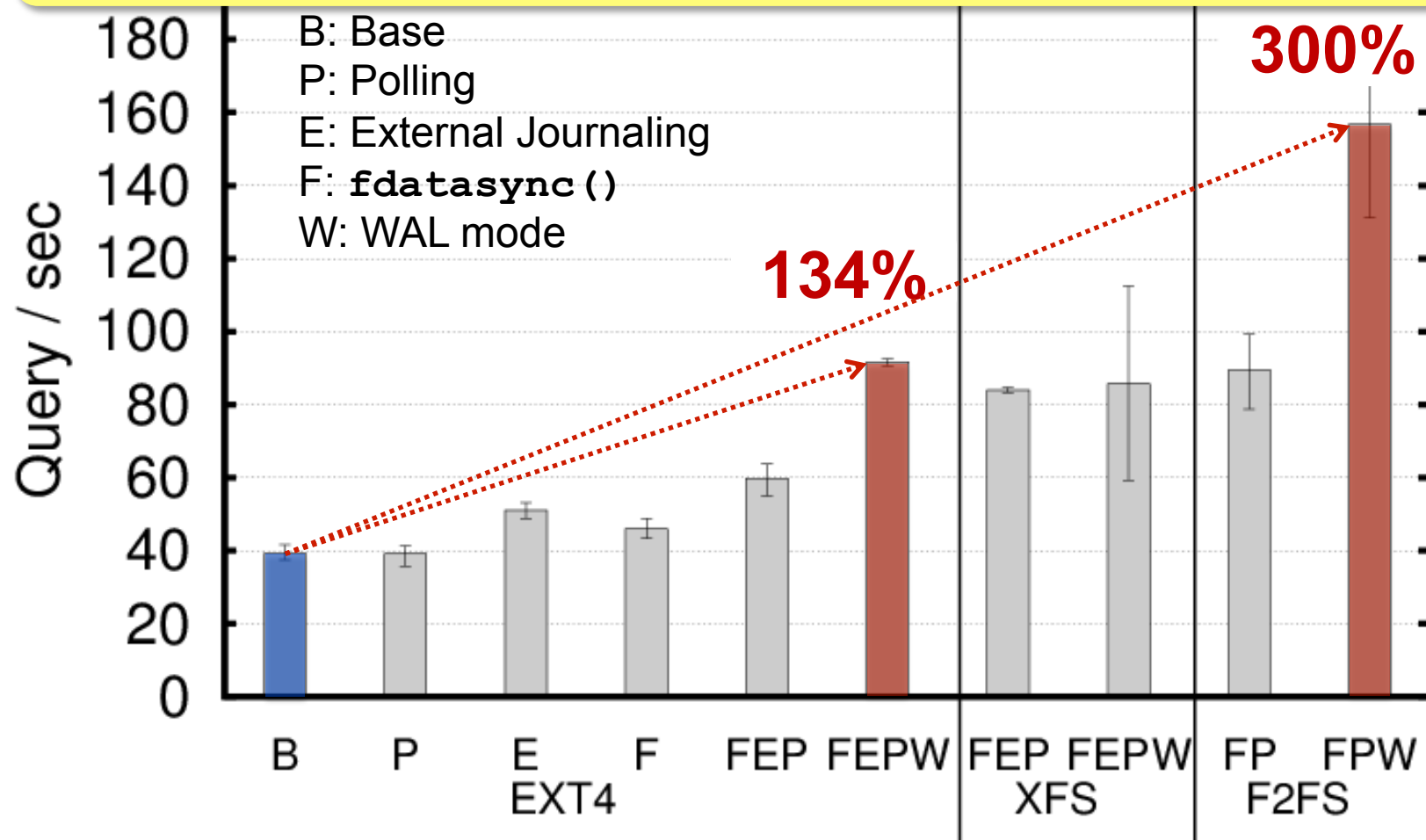
Real Workload

Replay Twitter and Facebook by Mobigen



Combining All the Improvements

fdatasync(), Ext. J, Polling, WAL: 134% up♪



Finally,

Polling

WAL[♪]

F2FS[♪]

Ext. J[♪]

atasync[♪]



Conclusion

- Android IO stack is collection of unorchestrated layers.
- Journaling of Journal(JOJ) lies at the core of the problem.
- We optimize Android I/O stack with WAL mode in SQLite, F2FS, `datasync()`, External journaling, polling based I/O.

What we achieved is...

- With legacy EXT4, SQLite performance improves by 134%.
- With F2FS, SQLite performance improves by 300%

solely via software modification on existing smartphone!

Best Paper Award



2013 USENIX ATC Web Site



2013 USENIX Annual Technical Conference

JUNE 26—28, 2013 • SAN JOSE, CA



- Overview
- Conference Organizers
- Registration Information
- Registration Discounts
- At a Glance
- Calendar
- Activities
- Technical Sessions
- Birds-of-a-Feather Sessions
- Sponsorship
- Hotel and Travel Information
- Services
- Students
- Questions
- Help Promote!
- For Participants
- Call for Papers
- Past Conferences

Home » I/O Stack Optimization for Smartphones

 0  0

 0

I/O Stack Optimization for Smartphones

CONNECT     

Conference: USENIX ATC '13
Sooman Jeong, *Hanyang University*; Kisung Lee, *Samsung Electronics*; Seongjin Lee, *Hanyang University*; Seoungbum Son, *Samsung Electronics*; Youjip Won, *Hanyang University*
Awarded Best Paper!

PRESENTATION VIDEO

I/O Stack Optimization for Smartphones

Sooman Jeong
Hanyang University

Awarded Best Paper!

0:00:06 0:28:59

SPONSORS

Silver Sponsor



Thank you...

