

Why Do We Always Blame Storage Stack?

Hao Luo Nimble Storage

Hong Jiang University of Texas Arlington

Myra B. Cohen University of Nebraska Lincoln

Storage for Smartphones...

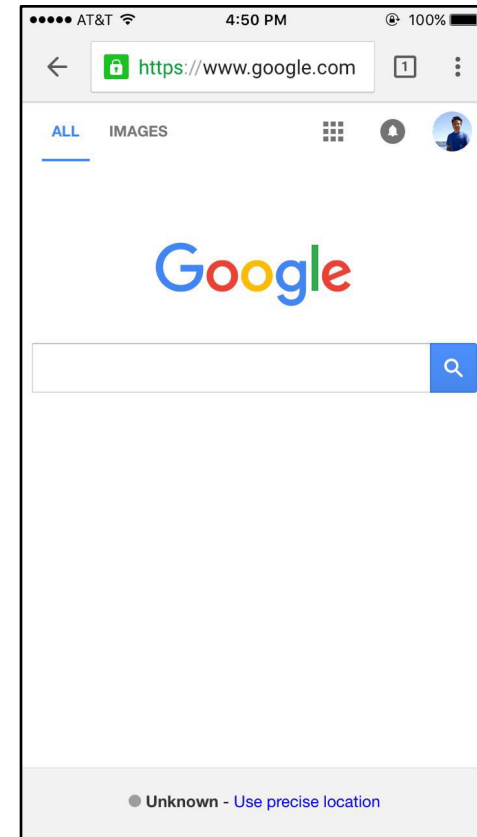
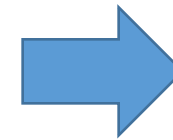
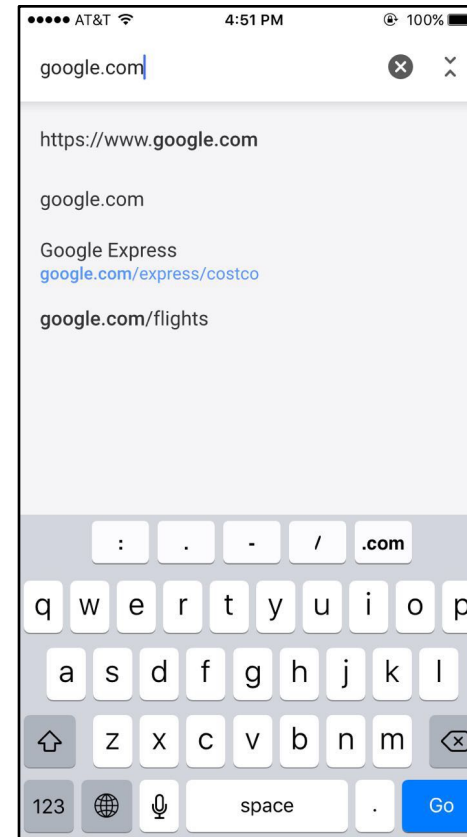
- Revisiting storage for smartphones [FAST'12]
- I/O Stack Optimization for Smartphones [ATC'13]
- Database optimizations
 - MVBT with lazy split [FAST'14]
 - WALDIO [ATC'15]
- File system optimizations
 - Single I/O commit path [FAST'14]
 - MobiFS [ATC'15]
- Other optimizations
 - qNVRAM [HotStorage'14]
 - Quasi-Asynchronous I/O [FAST'15]
 - WearDrive [ATC'15]

How much does the application performance benefit from storage stack optimization?



Application Performance in Smartphone

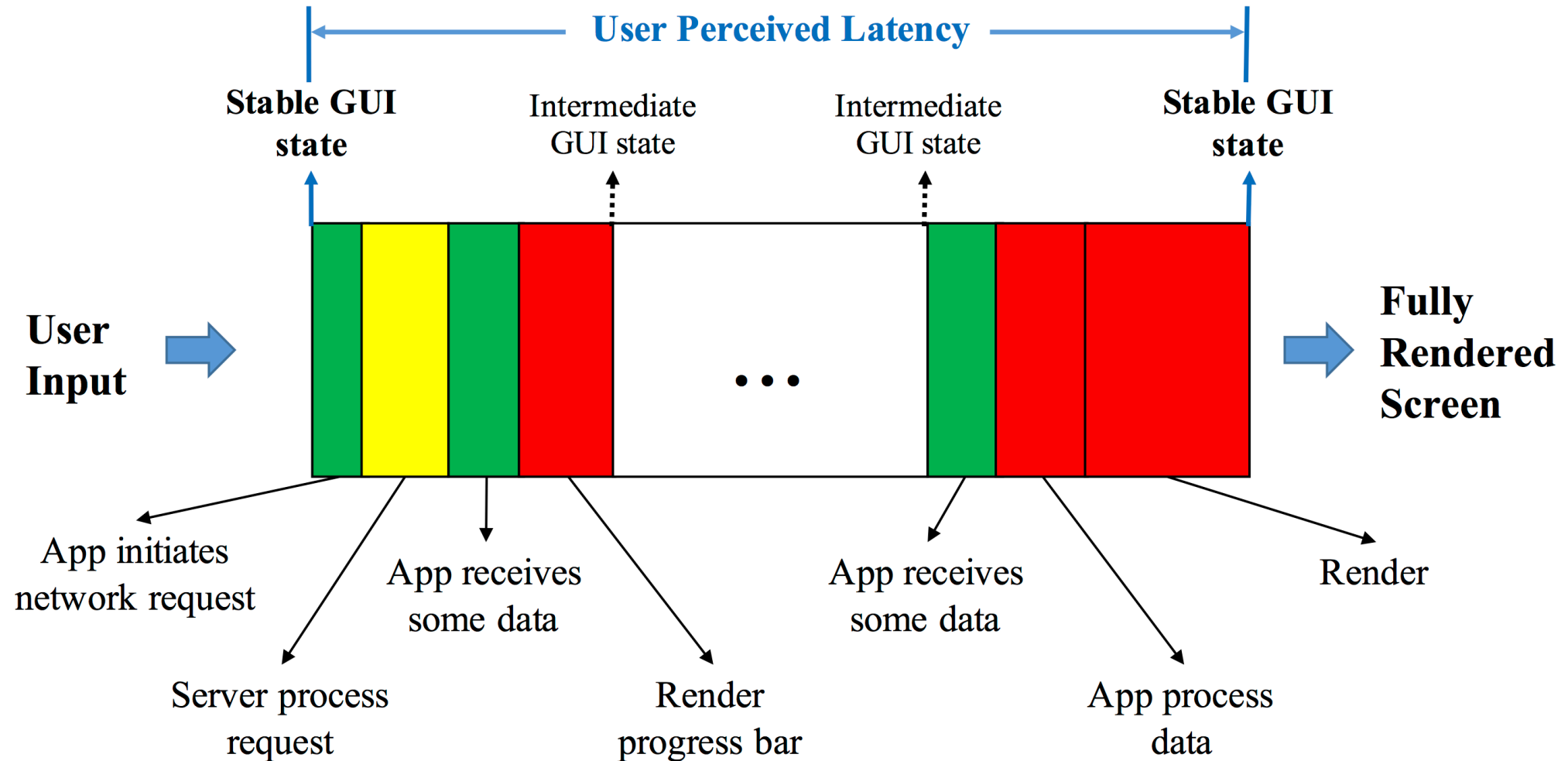
- Smartphone apps are GUI based interactive application.
- What would a smartphone user do?
 - Find a view
 - Interact with it
 - Check some state



User-Perceived Latency

- GUI state at time t
 - A set of widgets and their prosperities.
 - $S_{GUI} = \{(w, p, v) \mid w \in W, p \in P_w, v \in V_p\}$.
- Stable GUI state
 - the GUI state (S_{GUI}) remains unchanged without further user input.
 - Background jobs directly related to the operation are completed.
- Latency
 - The transition time between two consecutive stable GUI states

User-Perceived Latency



Response Time Limit

- Basic advice regarding response time [Usability Engineering, 1993]
 - 0.1 second: **react instantaneously**
 - 1.0 second: **keep user's flow of thought**
 - 10 second: **keep user's attention**
- Fast enough is good enough

Measuring User Perceived Latency

- Synchronization between benchmark tests and application under test.
 - Android test frameworks provide APIs for test writer to sync with the application.
- What most test frameworks do...
 - `Thread.sleep(10000);`
 - loops and retries and maybe with an exponential backoff
 - Slow and inaccurate....
- What Espresso do...
 - Automatically sync with UI events, asynchronous tasks, etc..
 - No wait-untils, return when app becomes idle.

MobiReplayer

- Android benchmark tool based off Espresso.
- Replay GUI traces and measure response time for every interaction.

Outline

- Introduction
- **Test Setup**
- **Evaluation**
- **Conclusion**

Application Benchmark

App	Workload
Web	Loading top 50 websites in U.S. one by one
Facebook	Swipe up the screen 50 times to load news feed
Messenger	Send 50 messages
Twitter	Post 50 tweets

Test Setup

- Two latest smartphones
 - Samsung Galaxy S4 GPE (2013) [KitKat]
 - Nexus 5X (2015) [Marshmallow]
- Two storage stack optimizations
 - SQLITE_NO_SYNC: disable fsync() in SQLite
 - EXT4_NO_JOURNAL: turn off Ext4 file system journaling

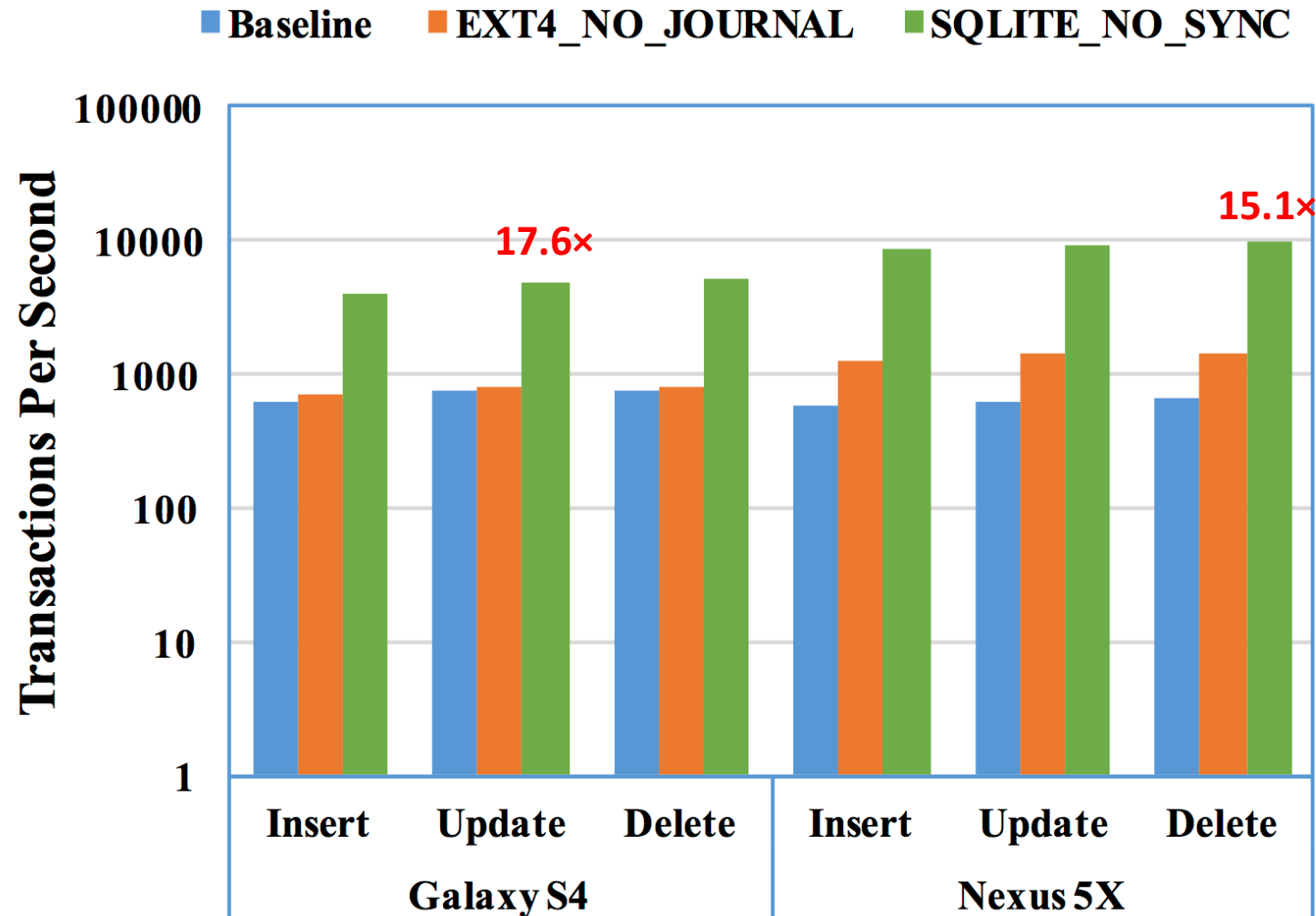
Outline

- Introduction
- Test Setup
- **Evaluation**
- **Conclusion**

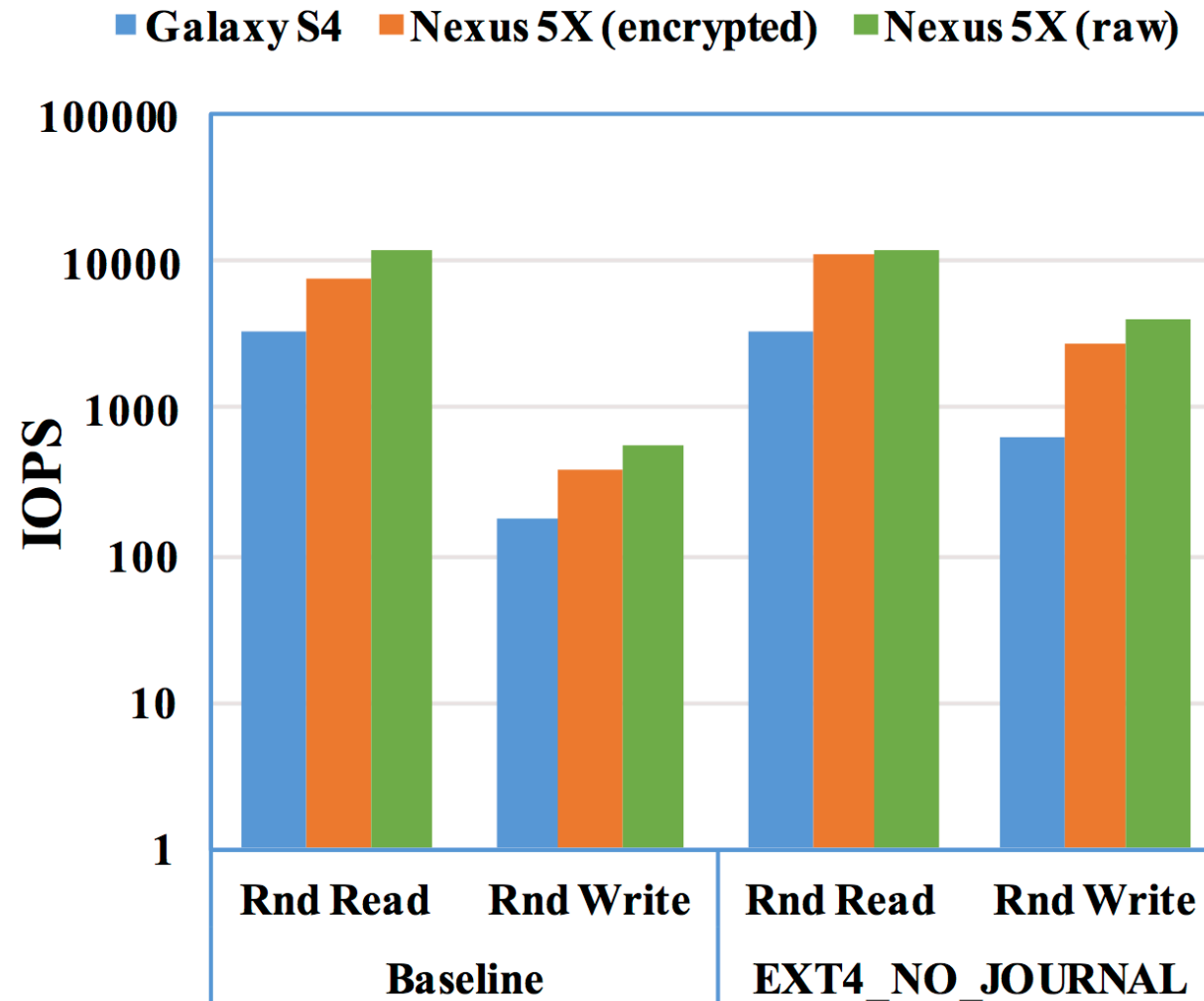
Evaluation

- **Q1:** How much do the database and file system benefit from storage stack optimizations?
- **Q2:** How much does the application performance benefit from storage stack optimization?
- **Q3:** If the application doesn't benefit from better storage performance, why?

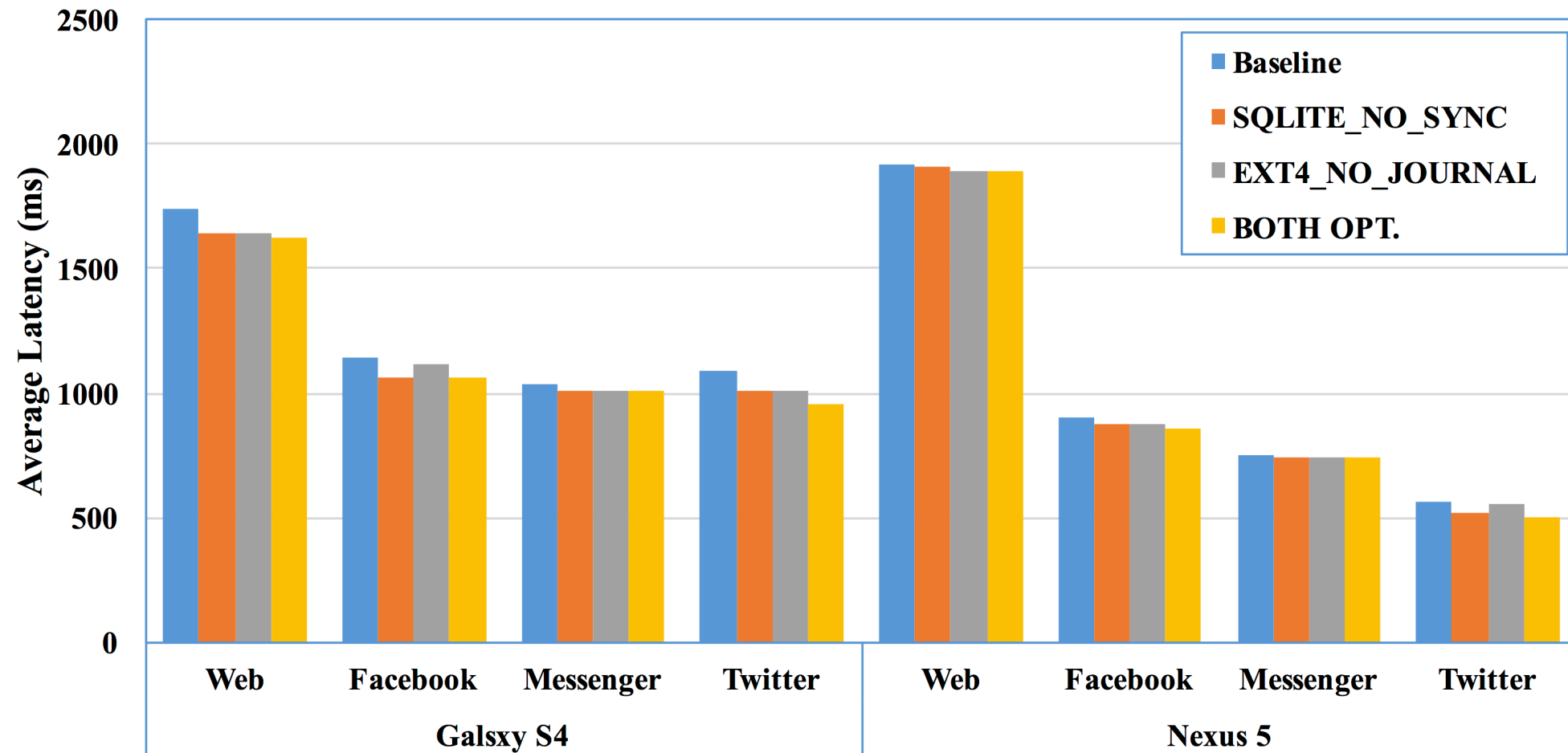
Q1: How much does SQLite benefit from storage stack optimizations?



Q1: How much does Ext4 file system benefit from storage stack optimizations?



Q2: How much does the application performance benefit from storage stack optimization?

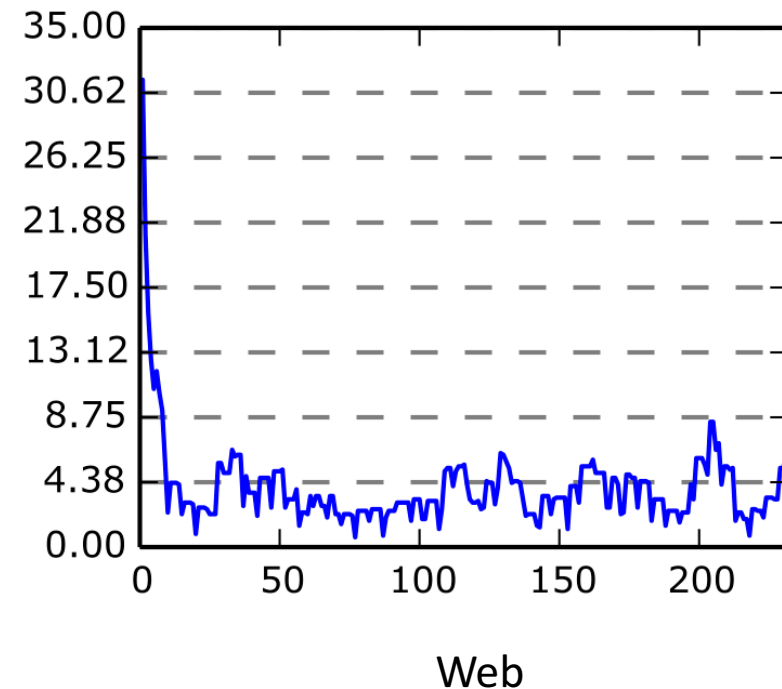
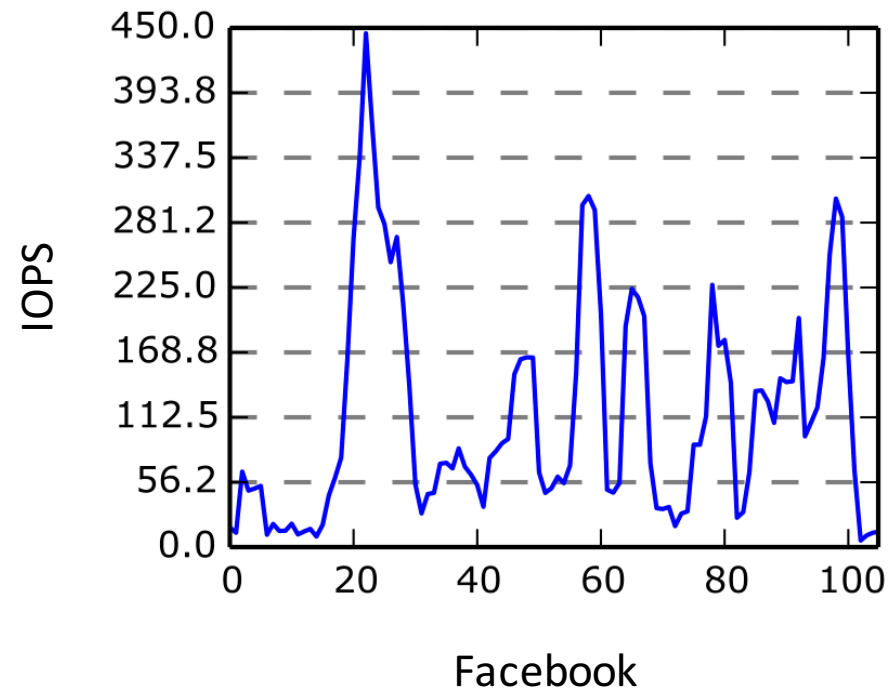


Q3: Why doesn't the application benefit from better storage performance?

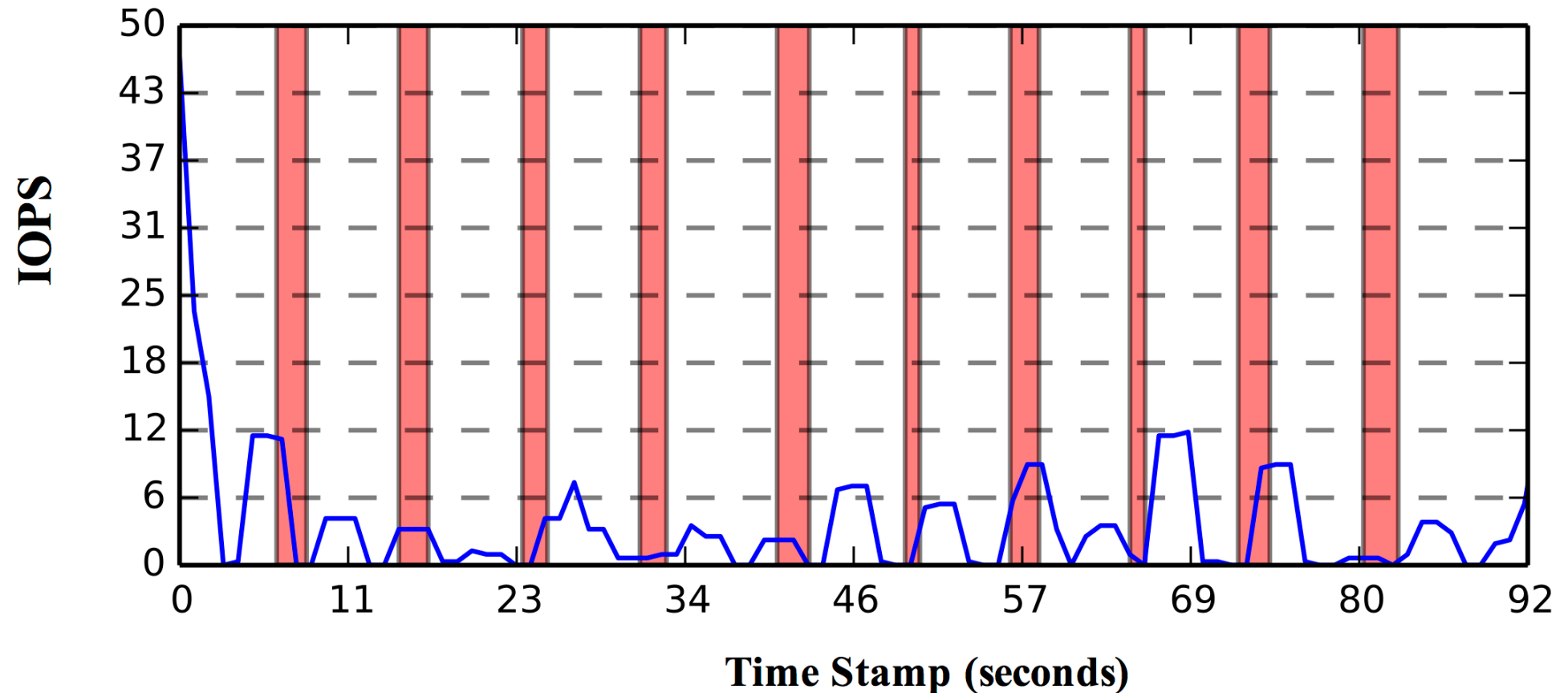
- Application now use the storage more wisely.
 - The disk I/O are not that intensive.
 - Move synchronous I/O out of critical path.

I/O activity of four benchmarks

- I/O activity in four benchmark runs



Correlation between user-perceived latency and I/Os



Outline

- Introduction
- Test Setup
- Evaluation
- **Conclusion**

Conclusion

- We develop a methodology for quantifying user-perceived latency and use it to evaluate four common application benchmarks with I/O stack optimization on two of the latest smartphones.
- The applications we tested respond reasonably fast
- The user-perceived latency does not drastically (at most 11.8%) benefit from I/O stack optimizations.

Thank you!