qNVRAM: quasi Non-Volatile RAM for Low Overhead Persistency Enforcement in Smartphones

Hao Luo, Lei Tian, Hong Jiang
University of Nebraska, Lincoln

Living in a mobile era

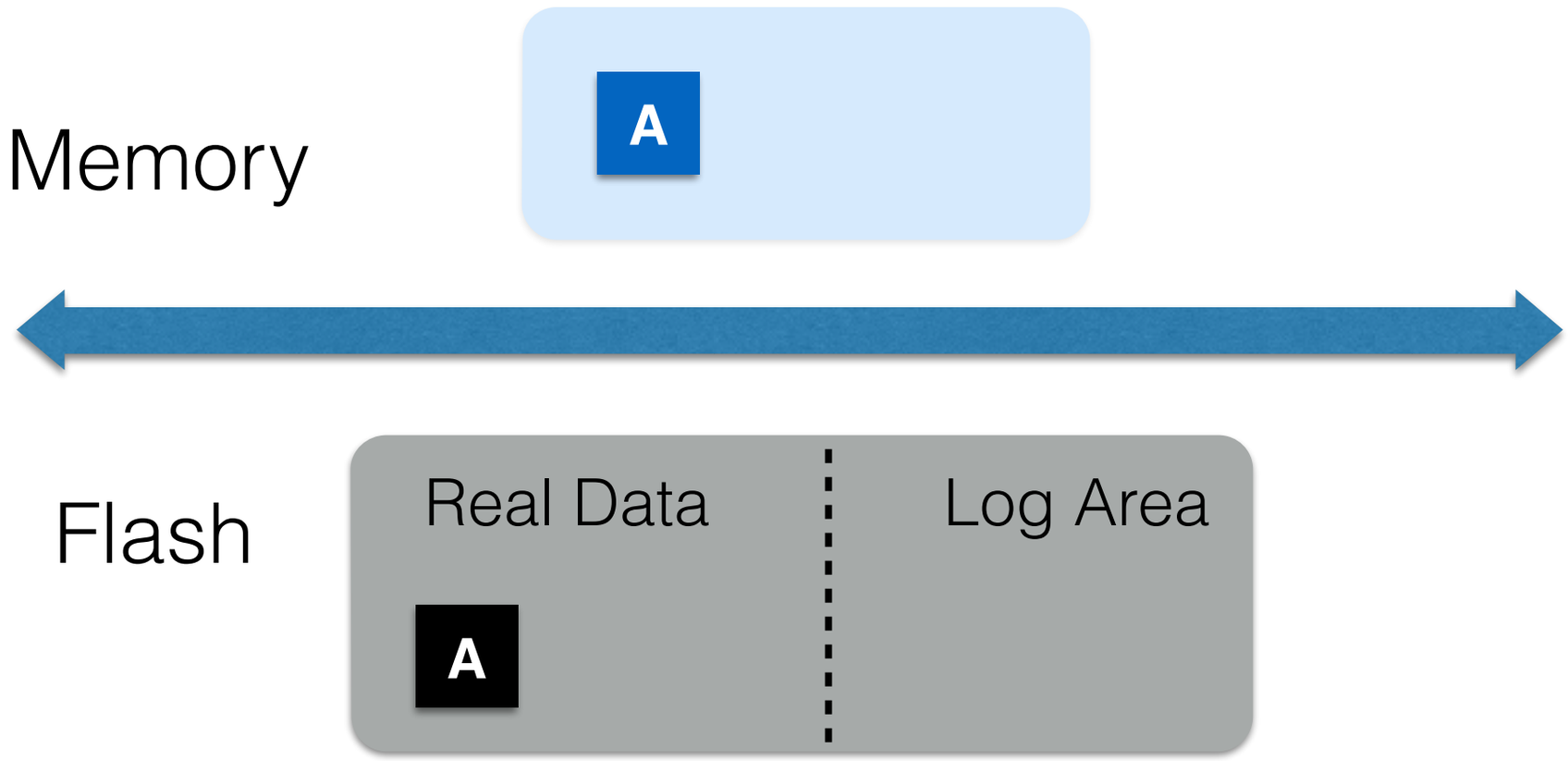
- We are in the mobile era
- The I/O performance dominates the overall application performance.
  - persistency = atomicity + consistency + durability
  - To ensure data persistency logging is employed by different components.

Outline

- Introduction
- **Overhead of Persistency**
- Rethinking Memory Volatility in Smartphones
- Design of qNVRAM
- Case Study

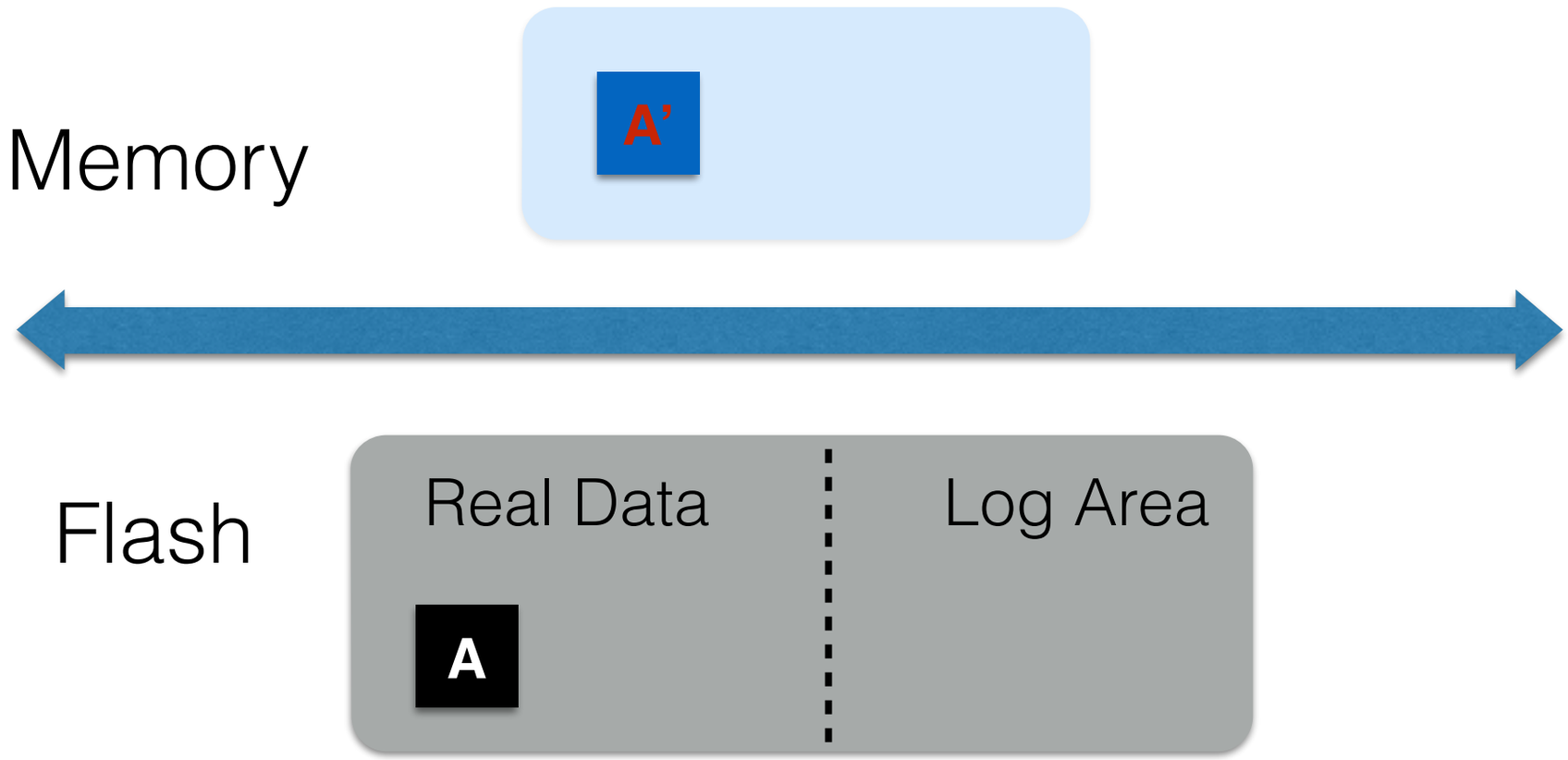
Overhead of Persistency

- Logging



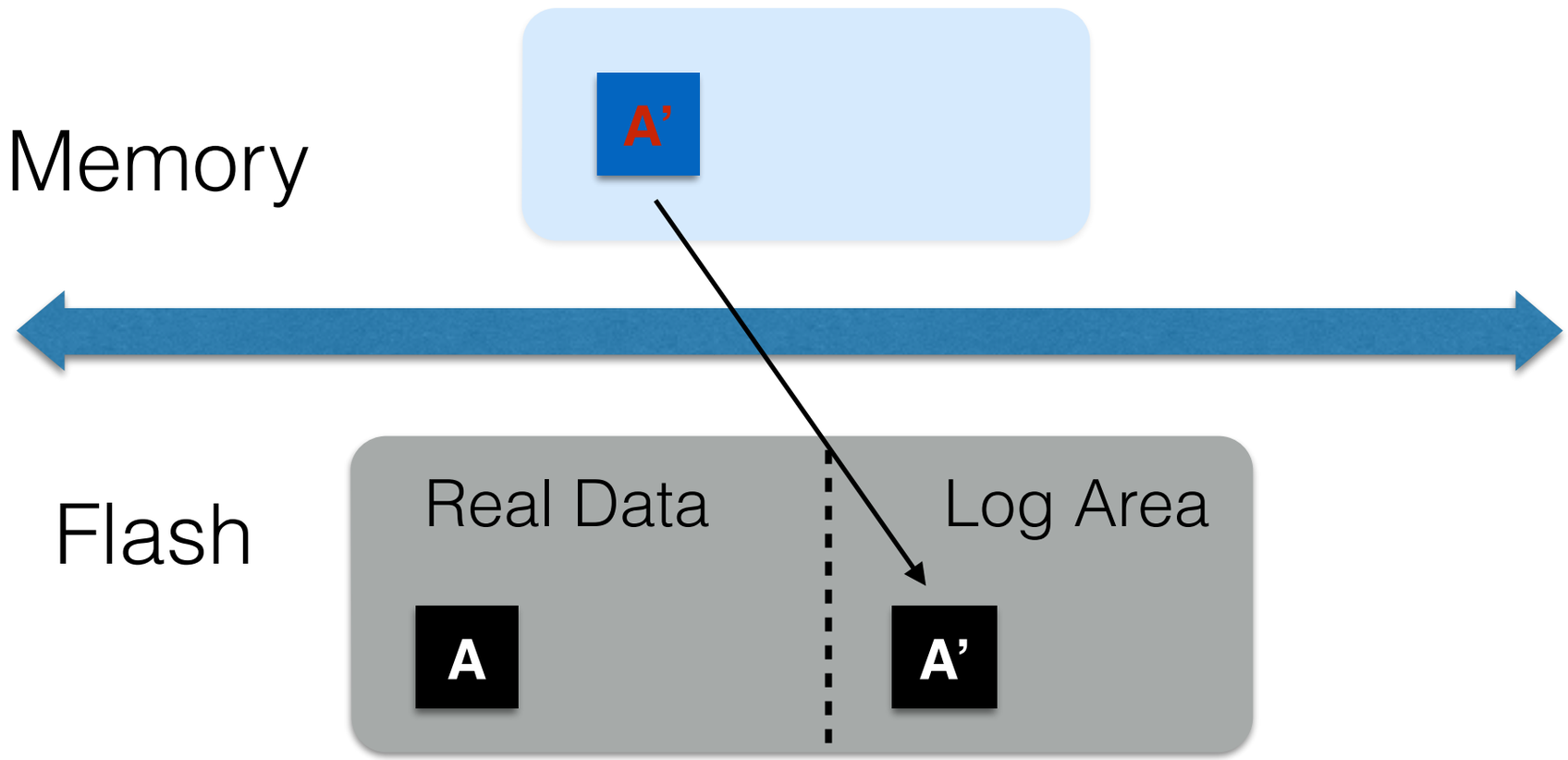
Overhead of Persistency

- Logging



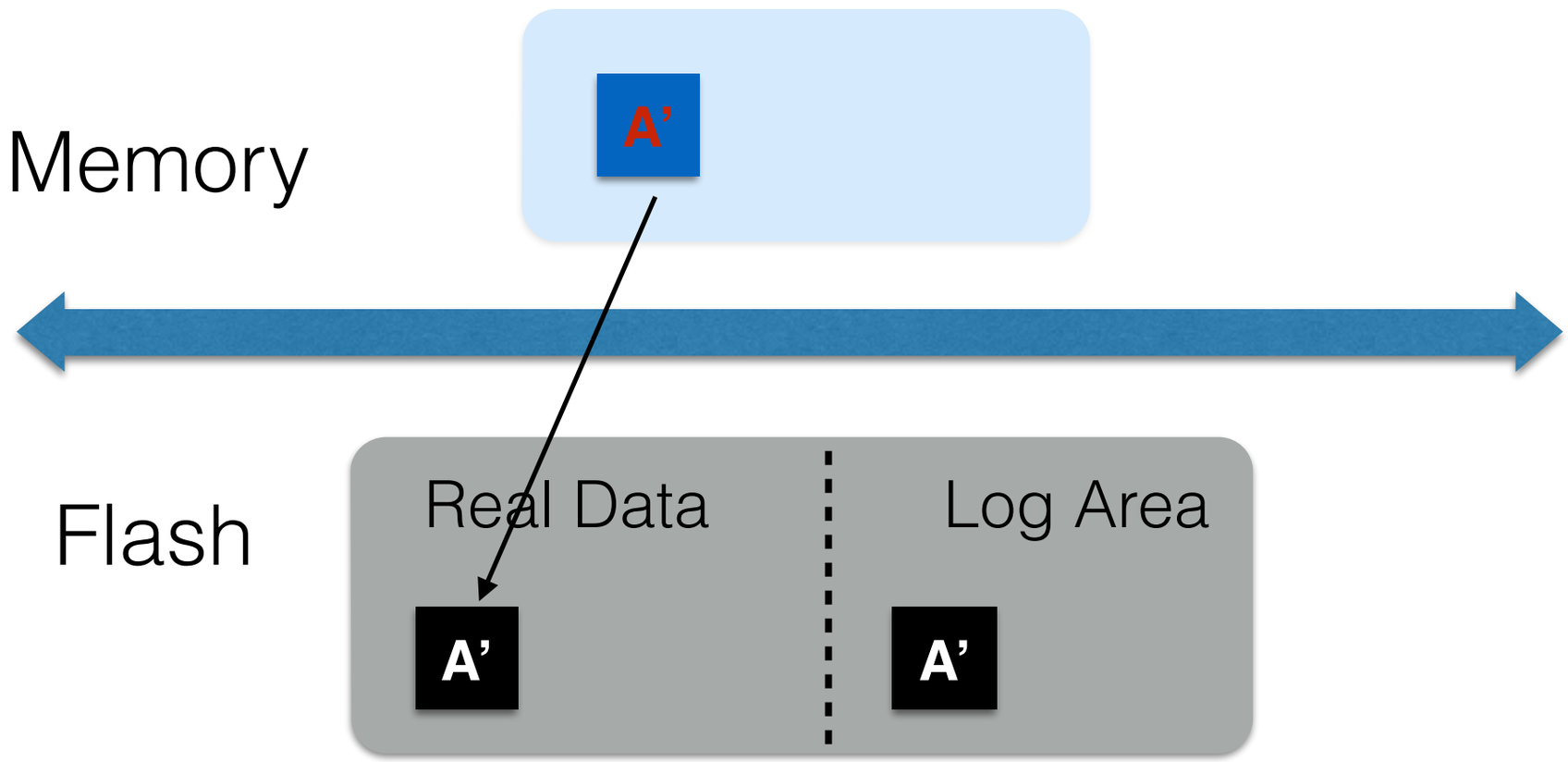
Overhead of Persistency

- Logging



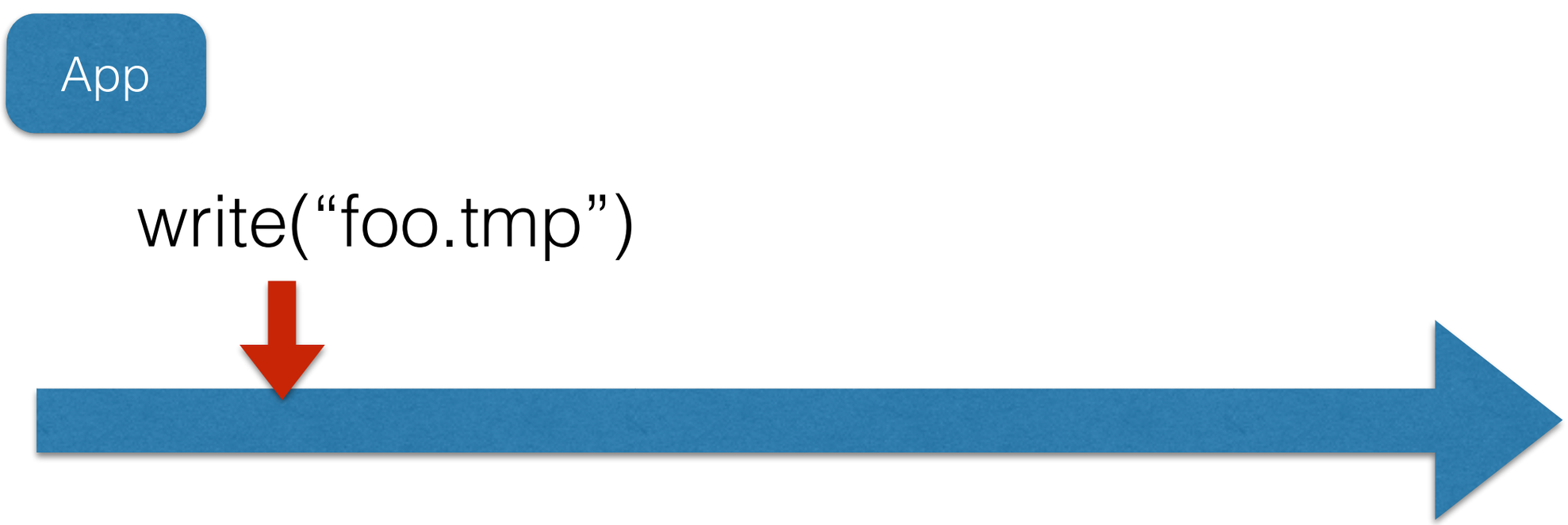
Overhead of Persistency

- Logging



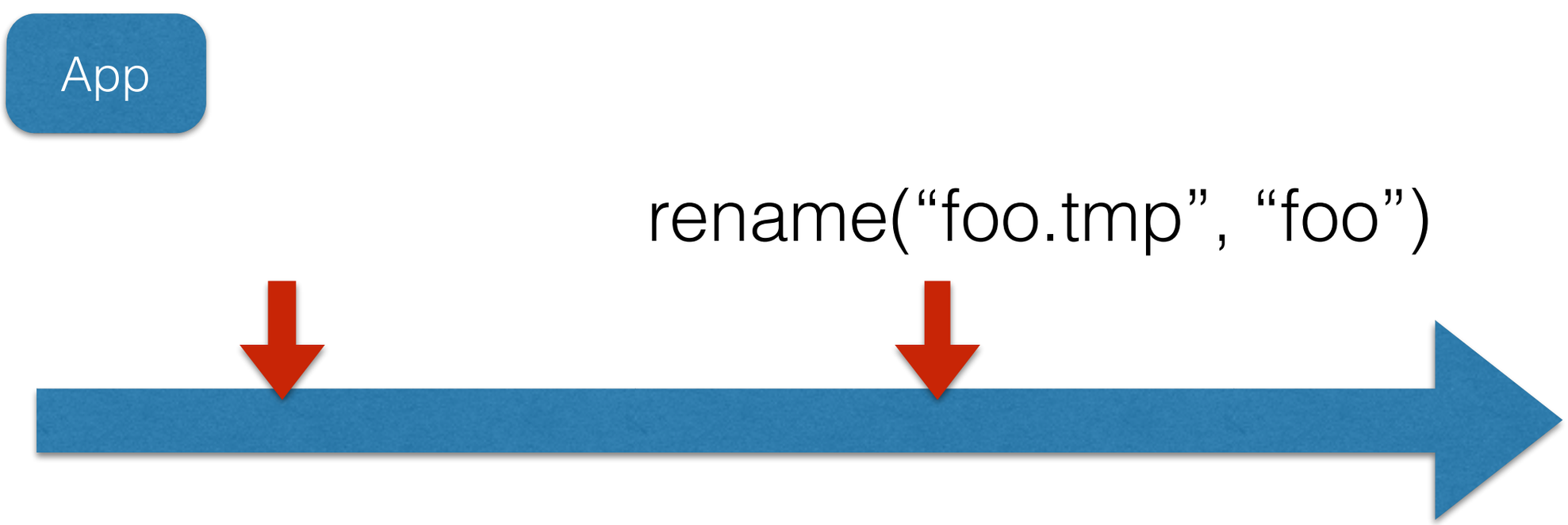
Overhead of Persistency

- Double Write
  - Some applications store data in raw files
  - Applications ***overwrite the whole file*** when update it.

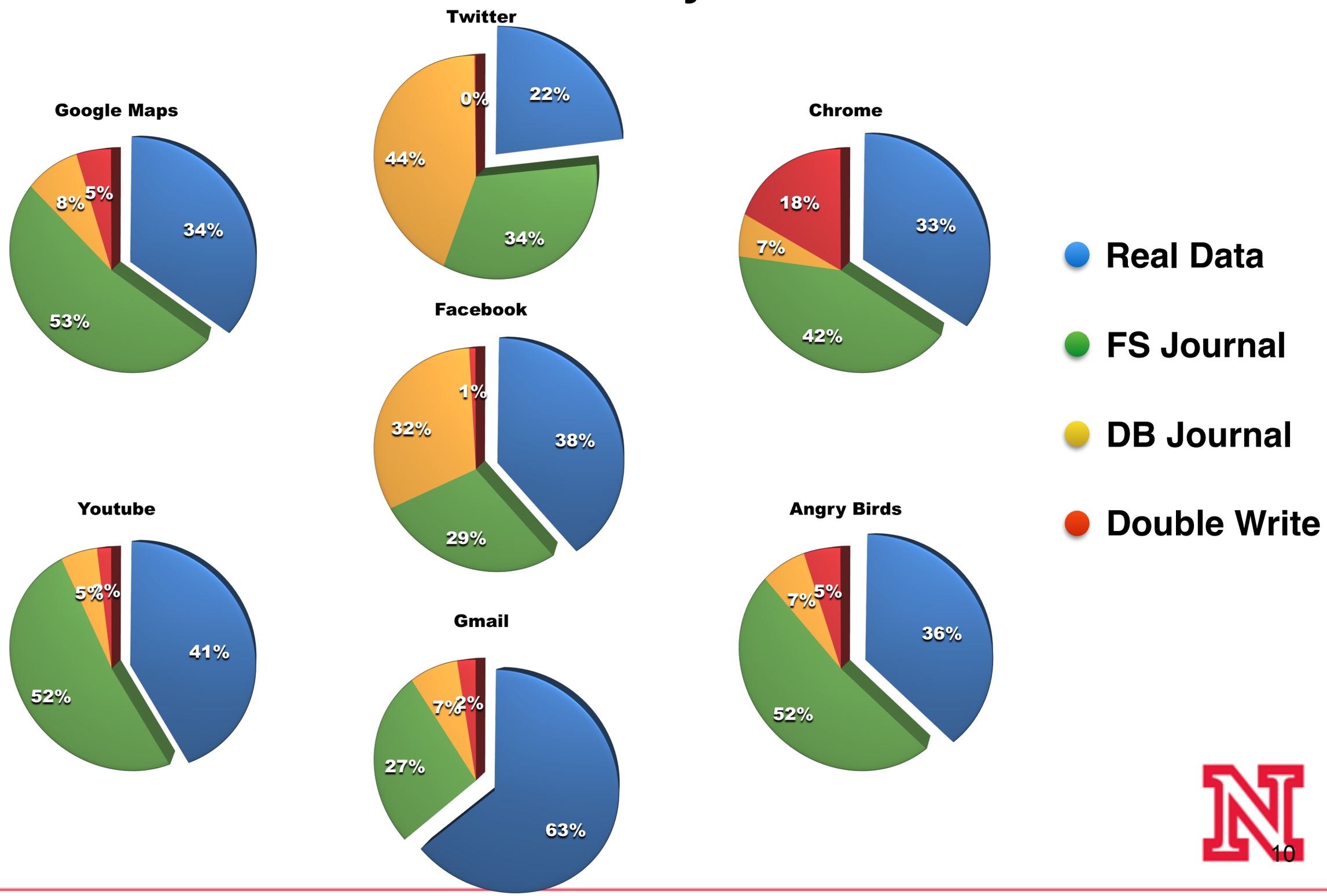


Overhead of Persistency

- Double Write
  - Some applications store data in raw files
  - Applications ***overwrite the whole file*** when update it.



Overhead of Persistency

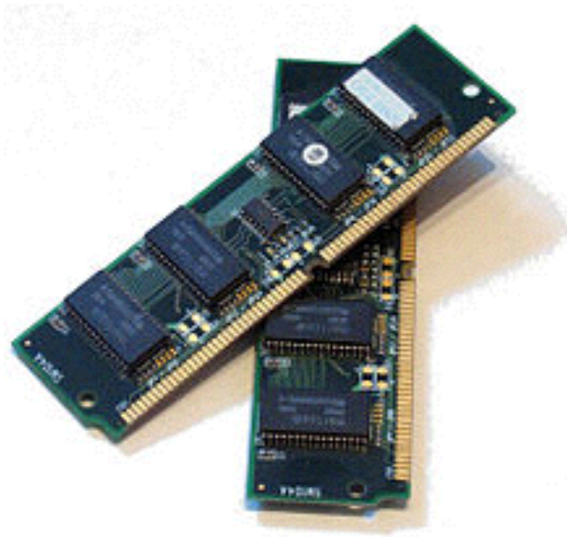


- Why do we need logging?
  - Memory is volatile.

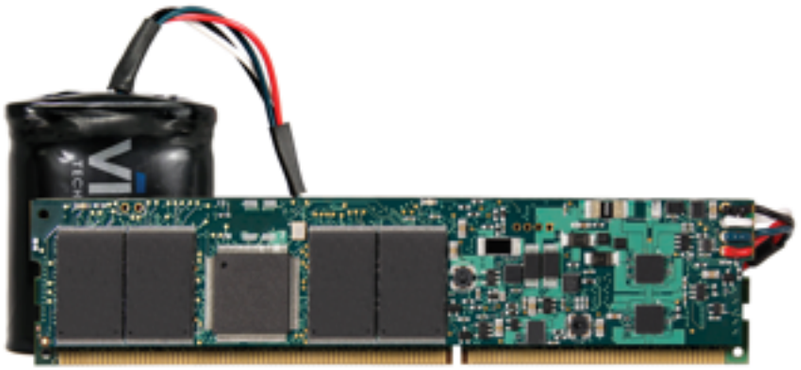
- Why do we need logging?
  - Memory is volatile.
- Every smartphone has battery

- Why do we need logging?
  - Memory is volatile.
- Every smartphone has battery
- DRAM + Battery = ?

+



$\stackrel{?}{=}$



Outline

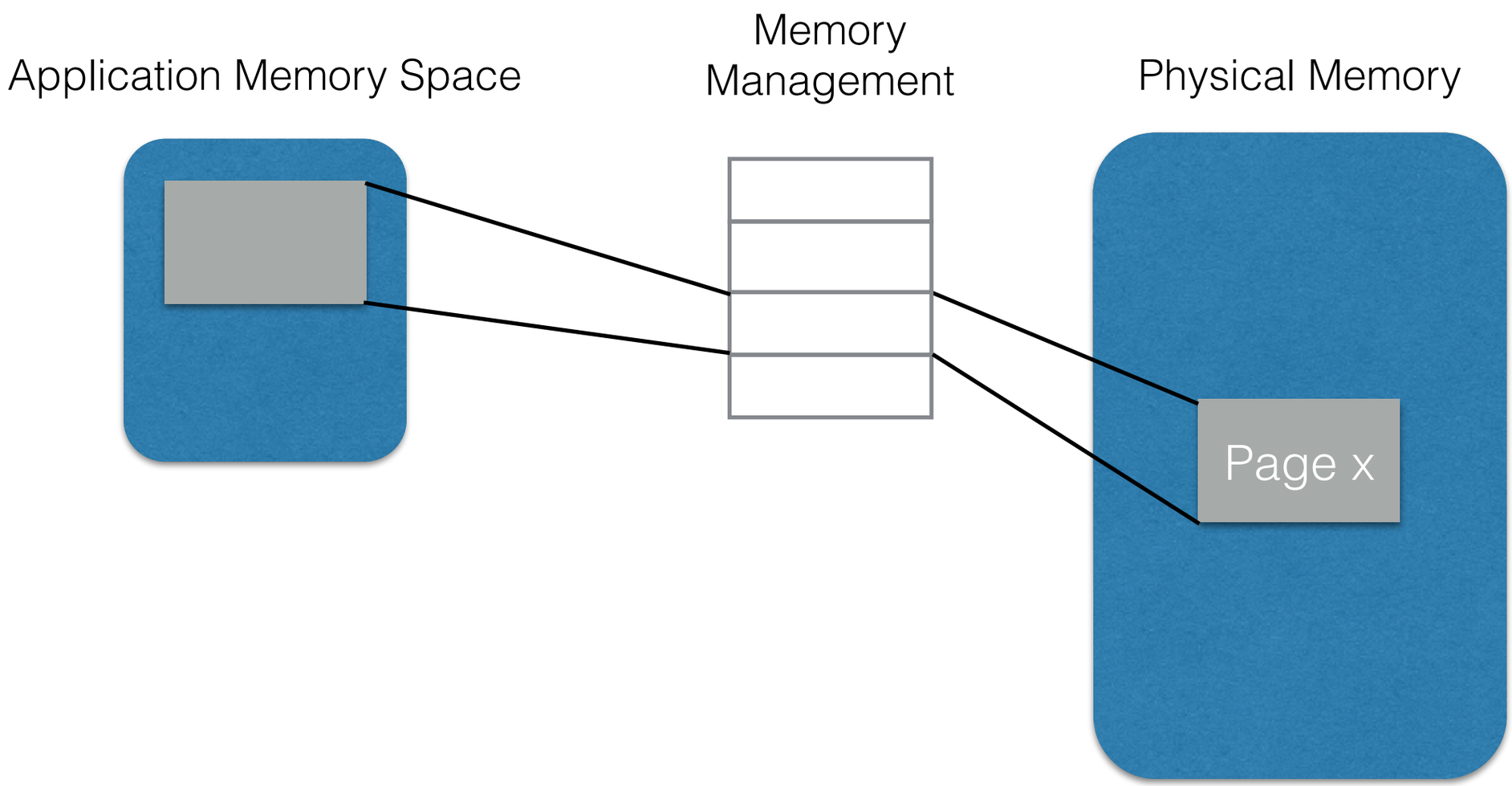
- Introduction
- Overhead of Persistency
- **Rethinking Memory Volatility in Smartphones**
- Design of qNVRAM
- Case Study

Rethinking Memory Volatility in smartphones

- Four types of failures in smartphones
  - Application crash
  - Application hang
  - Self-reboot
  - System freeze

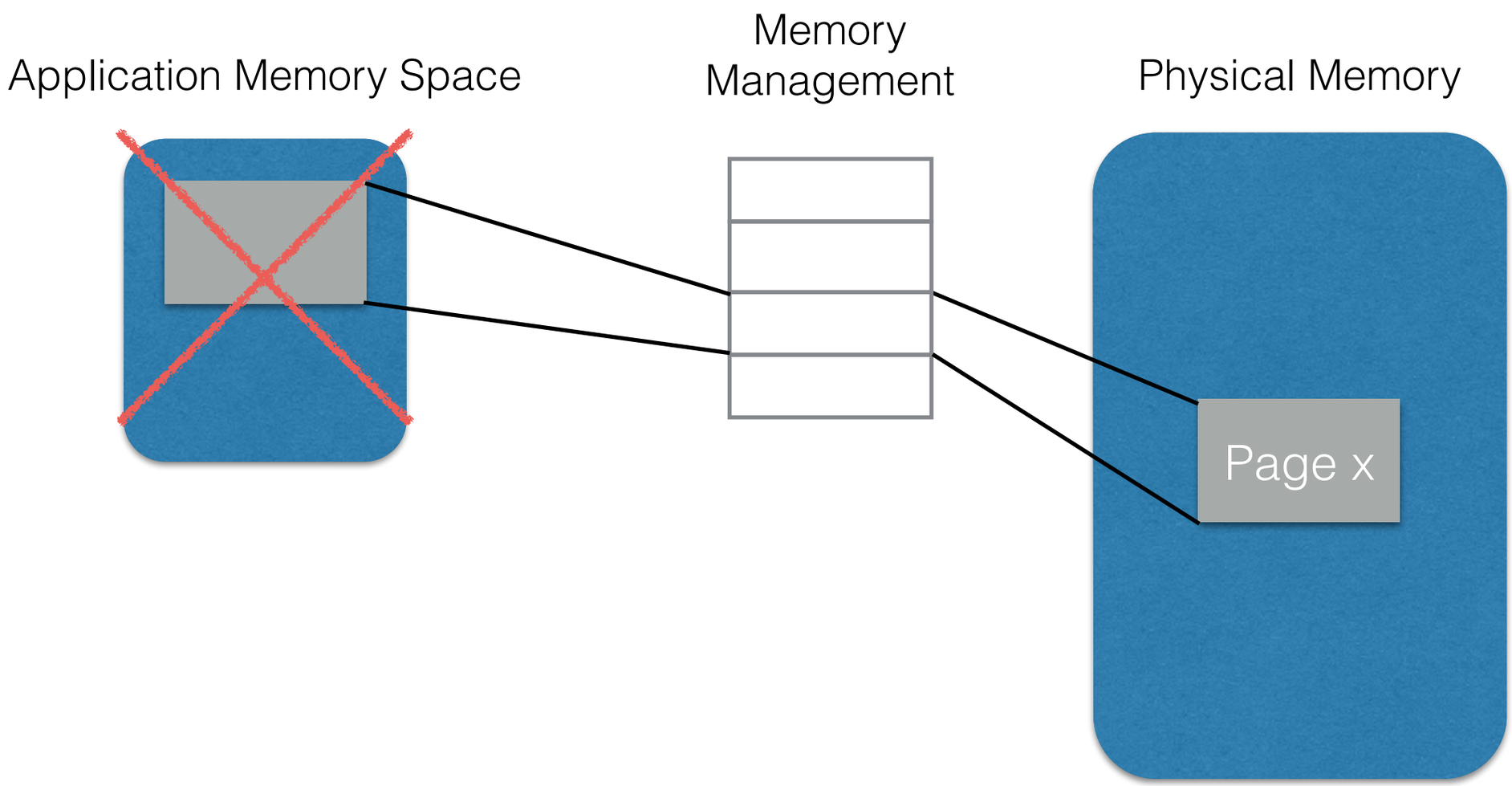
Rethinking Memory Volatility in smartphones

- Application crash / Application hang



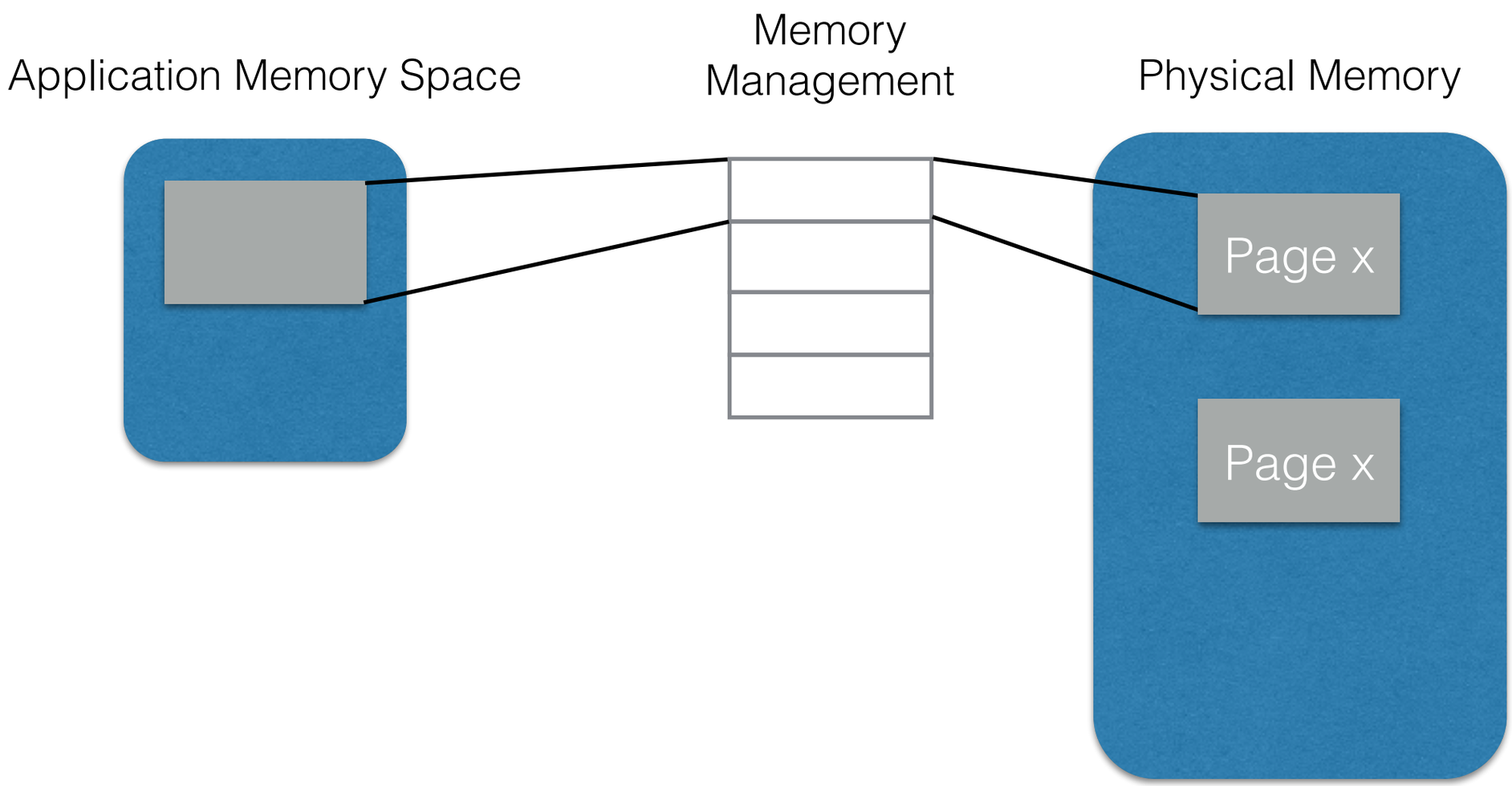
Rethinking Memory Volatility in smartphones

- Application crash / Application hang



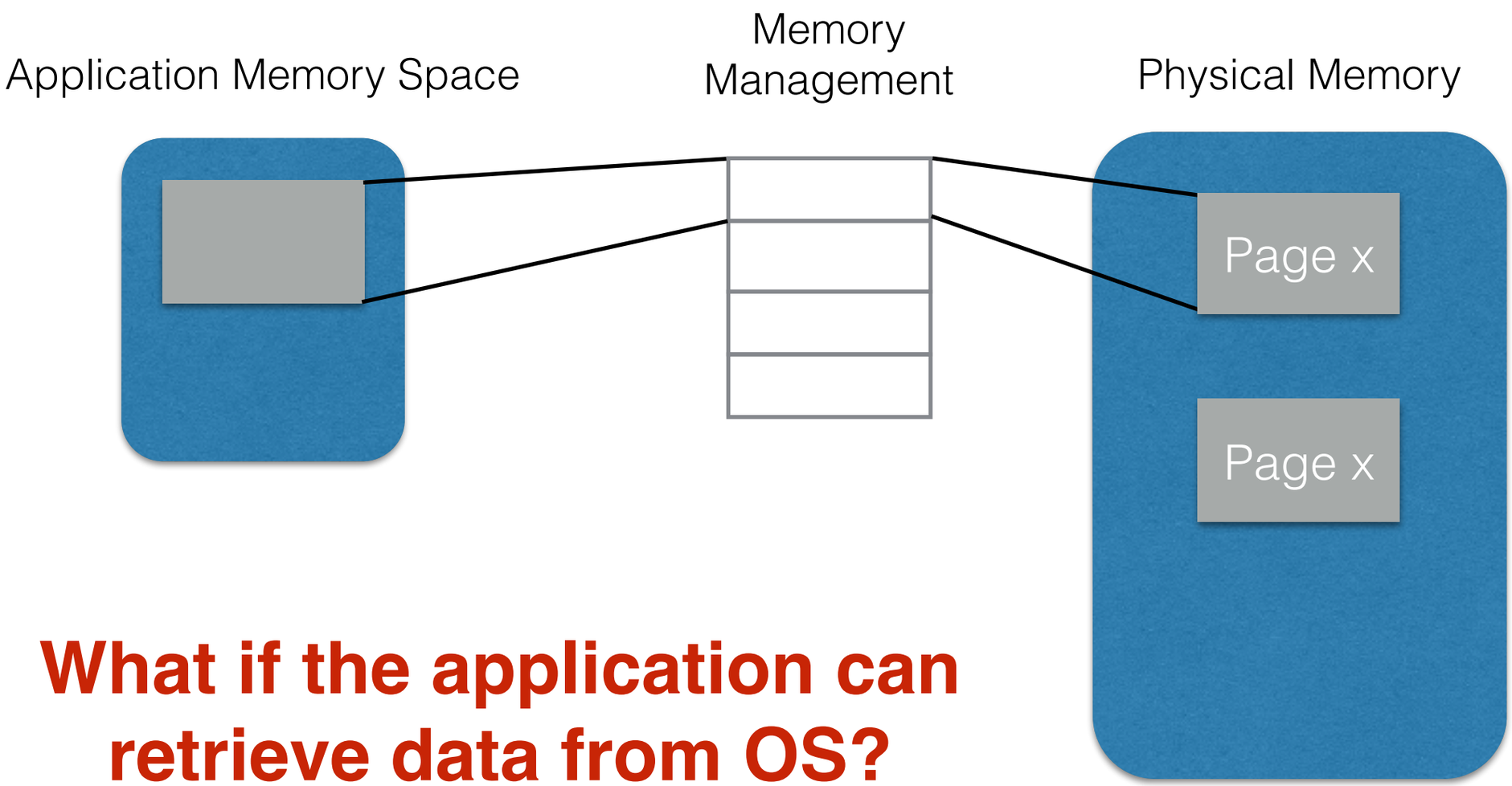
Rethinking Memory Volatility in smartphones

- Application crash / Application hang



Rethinking Memory Volatility in smartphones

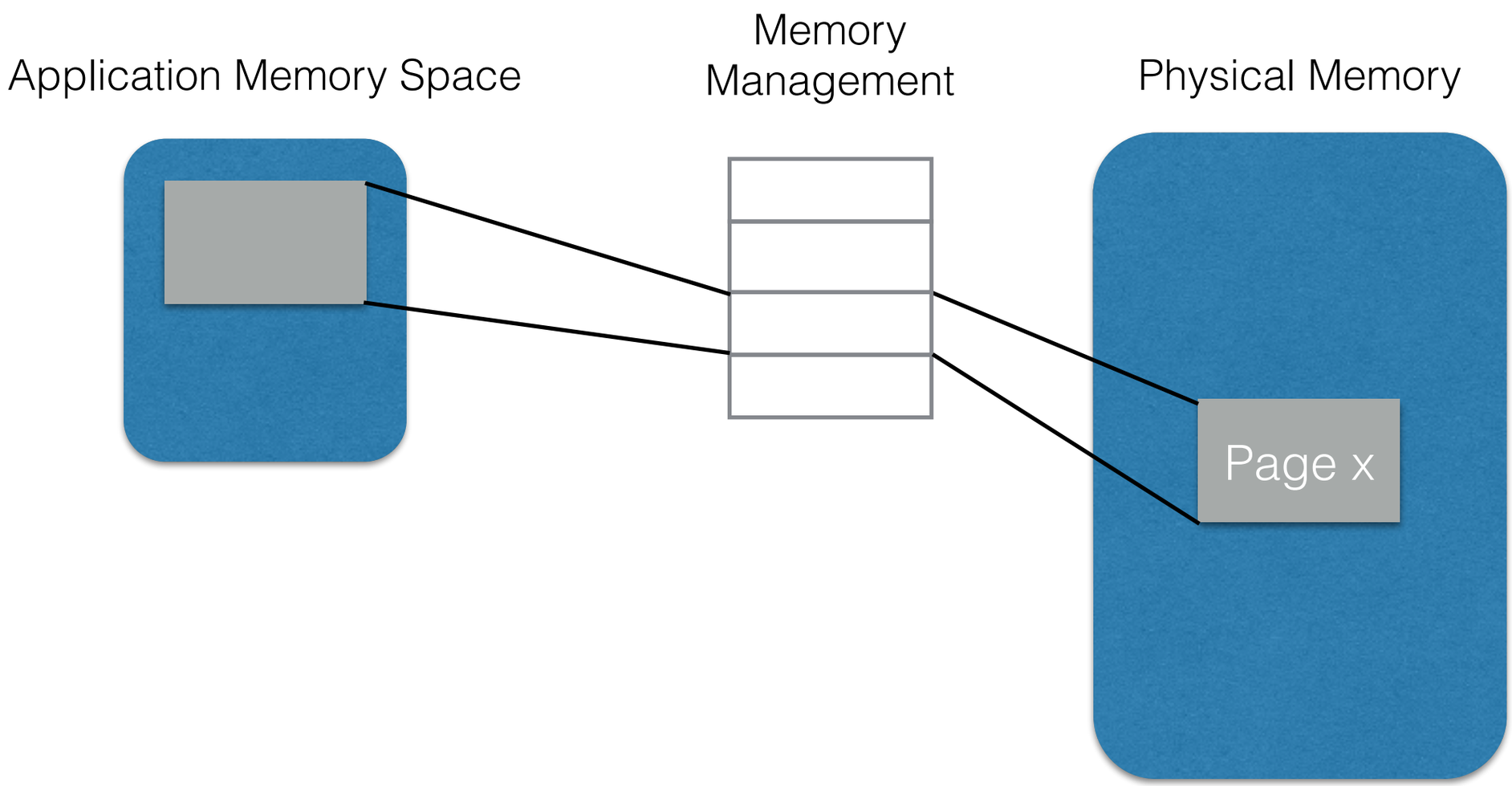
- Application crash / Application hang



What if the application can retrieve data from OS?

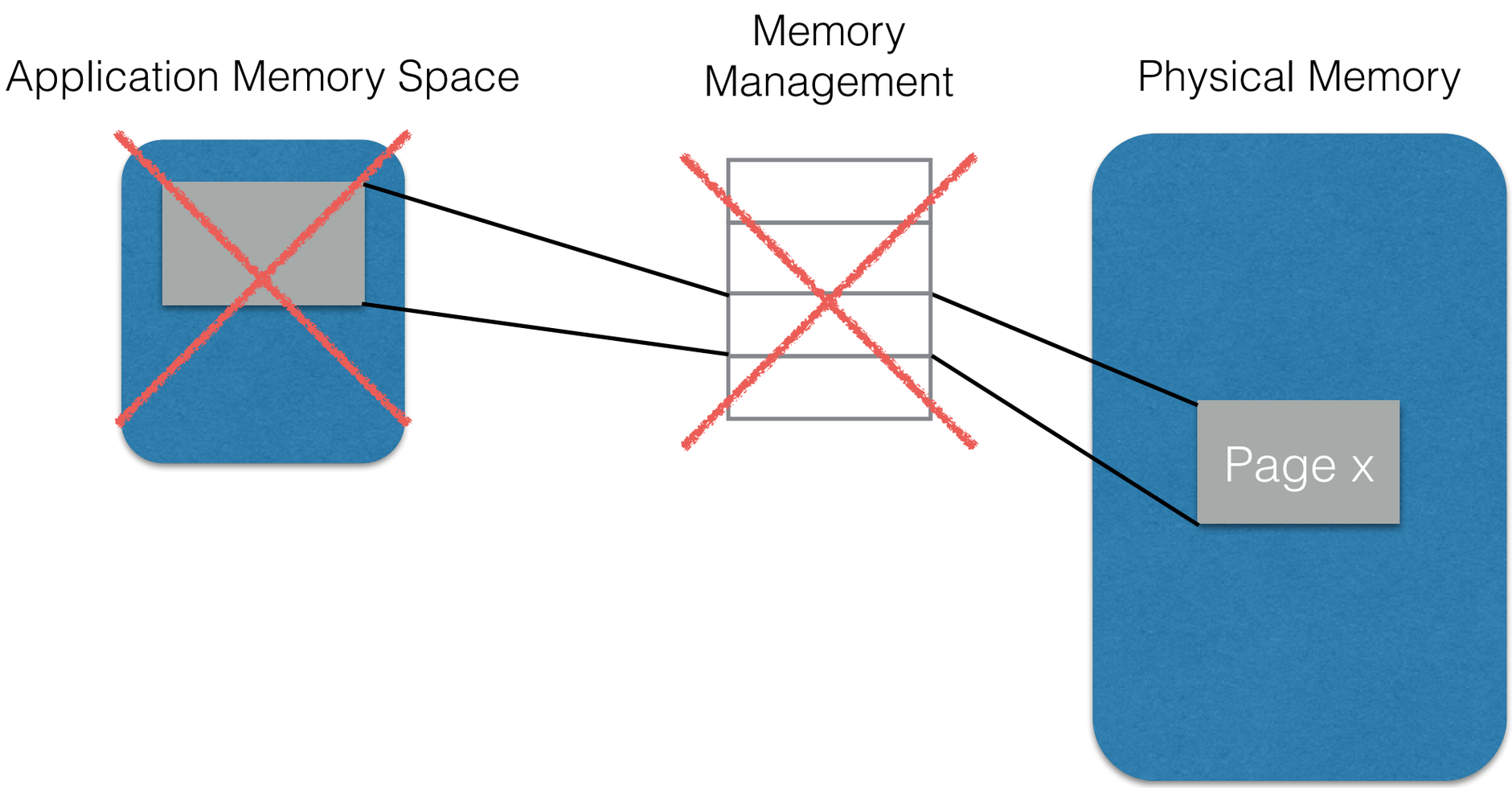
Rethinking Memory Volatility in smartphones

- Self-reboot



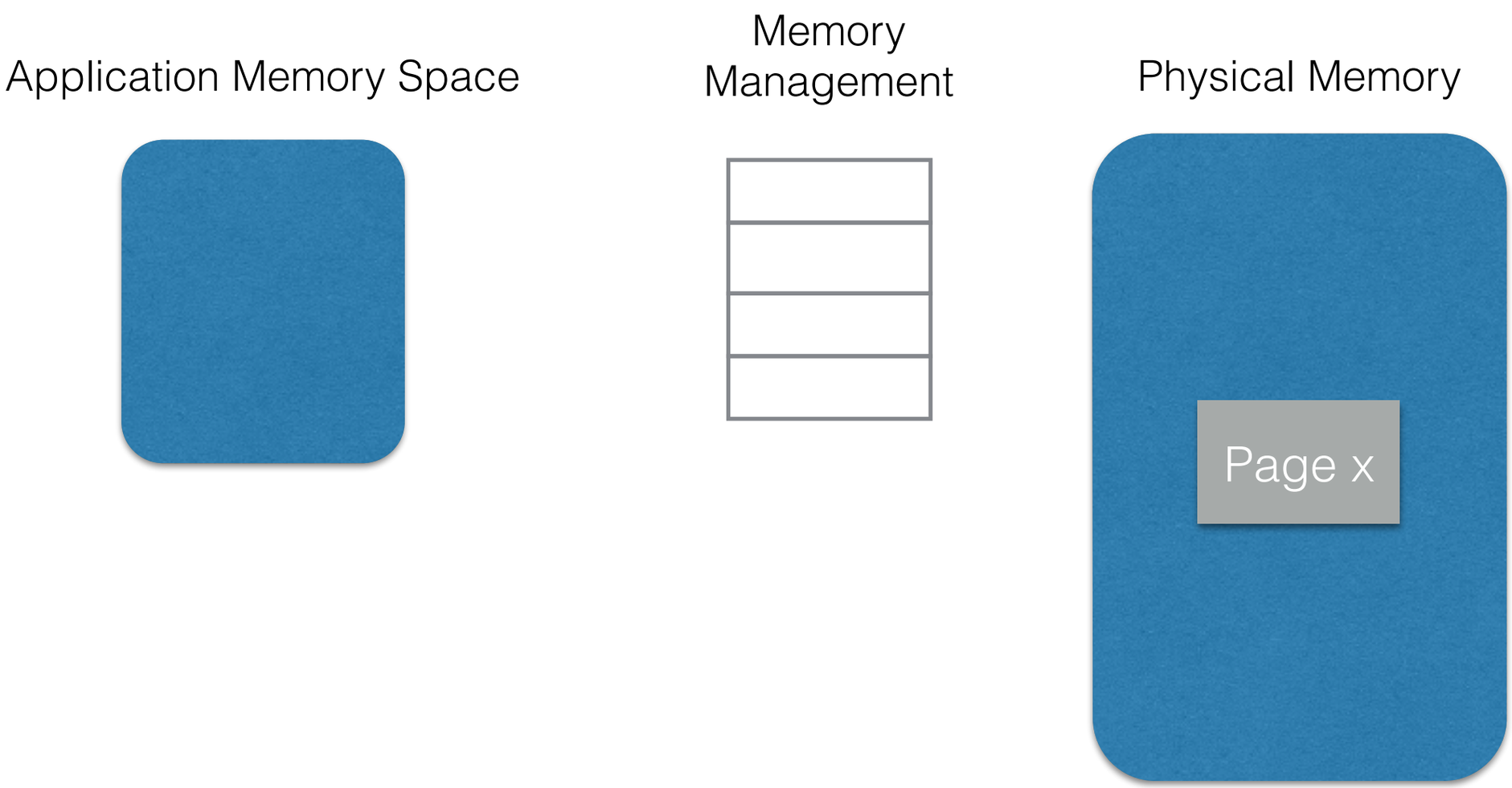
Rethinking Memory Volatility in smartphones

- Self-reboot



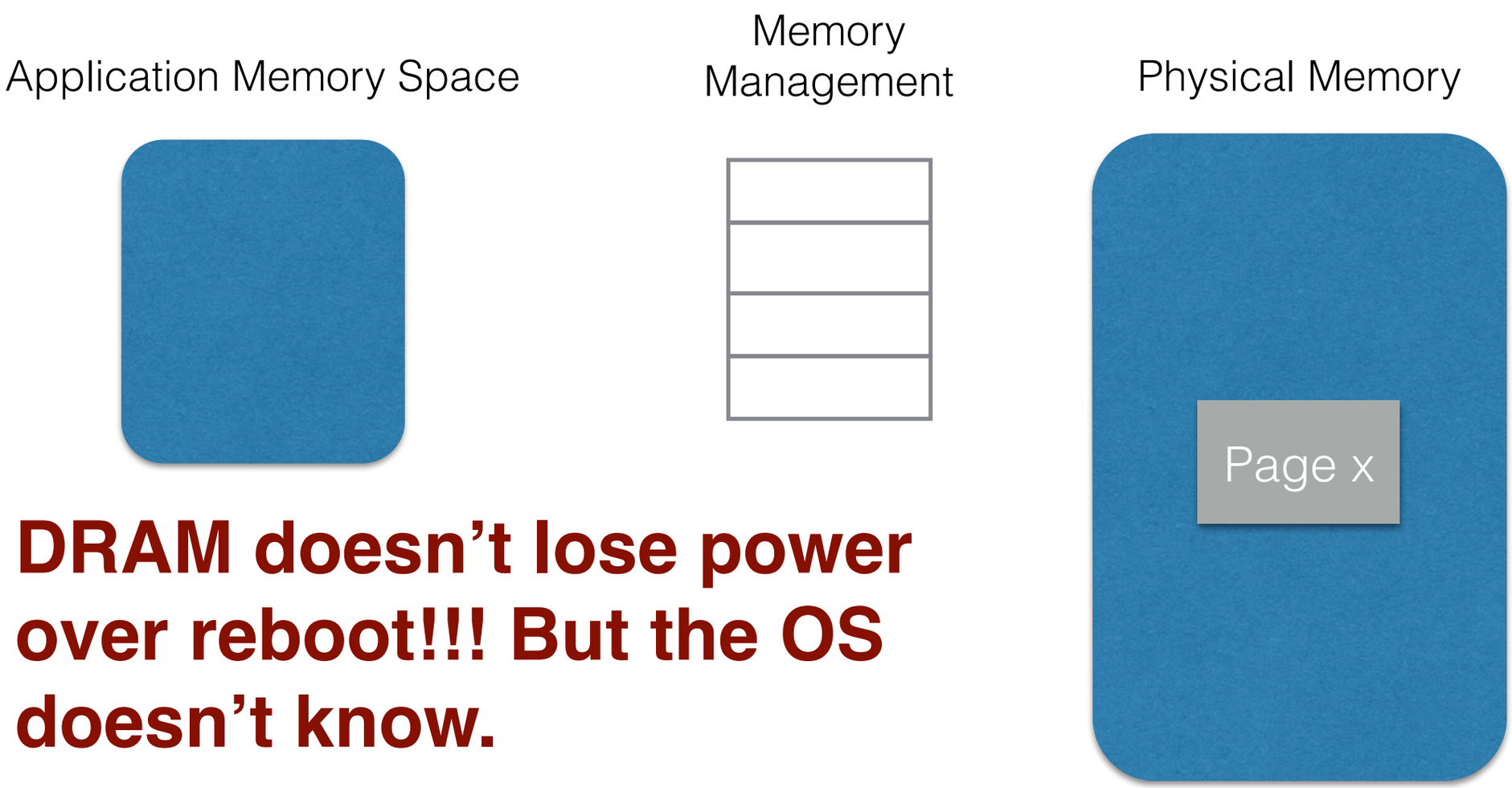
Rethinking Memory Volatility in smartphones

- Self-reboot



Rethinking Memory Volatility in smartphones

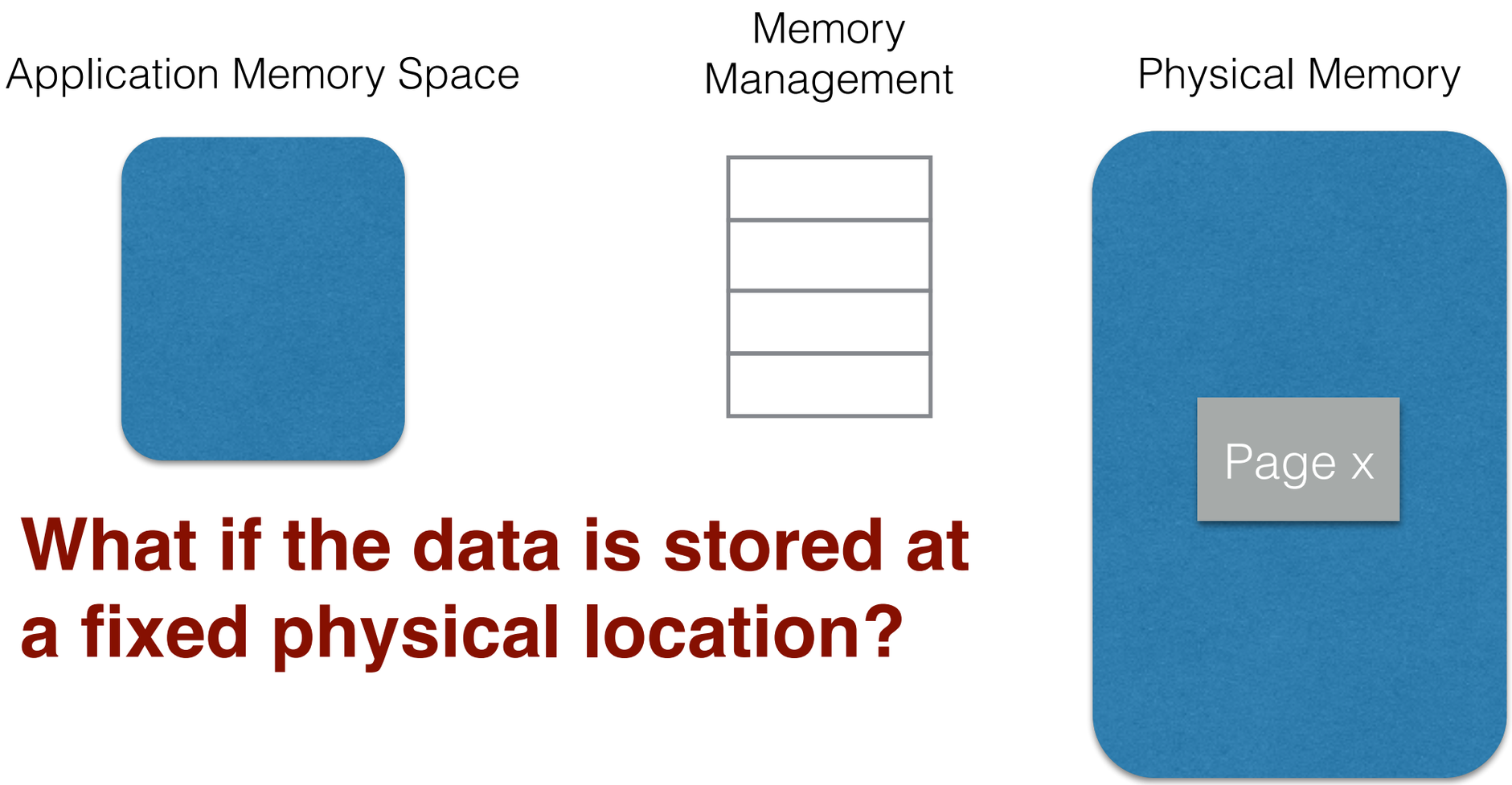
- Self-reboot



DRAM doesn't lose power over reboot!!! But the OS doesn't know.

Rethinking Memory Volatility in smartphones

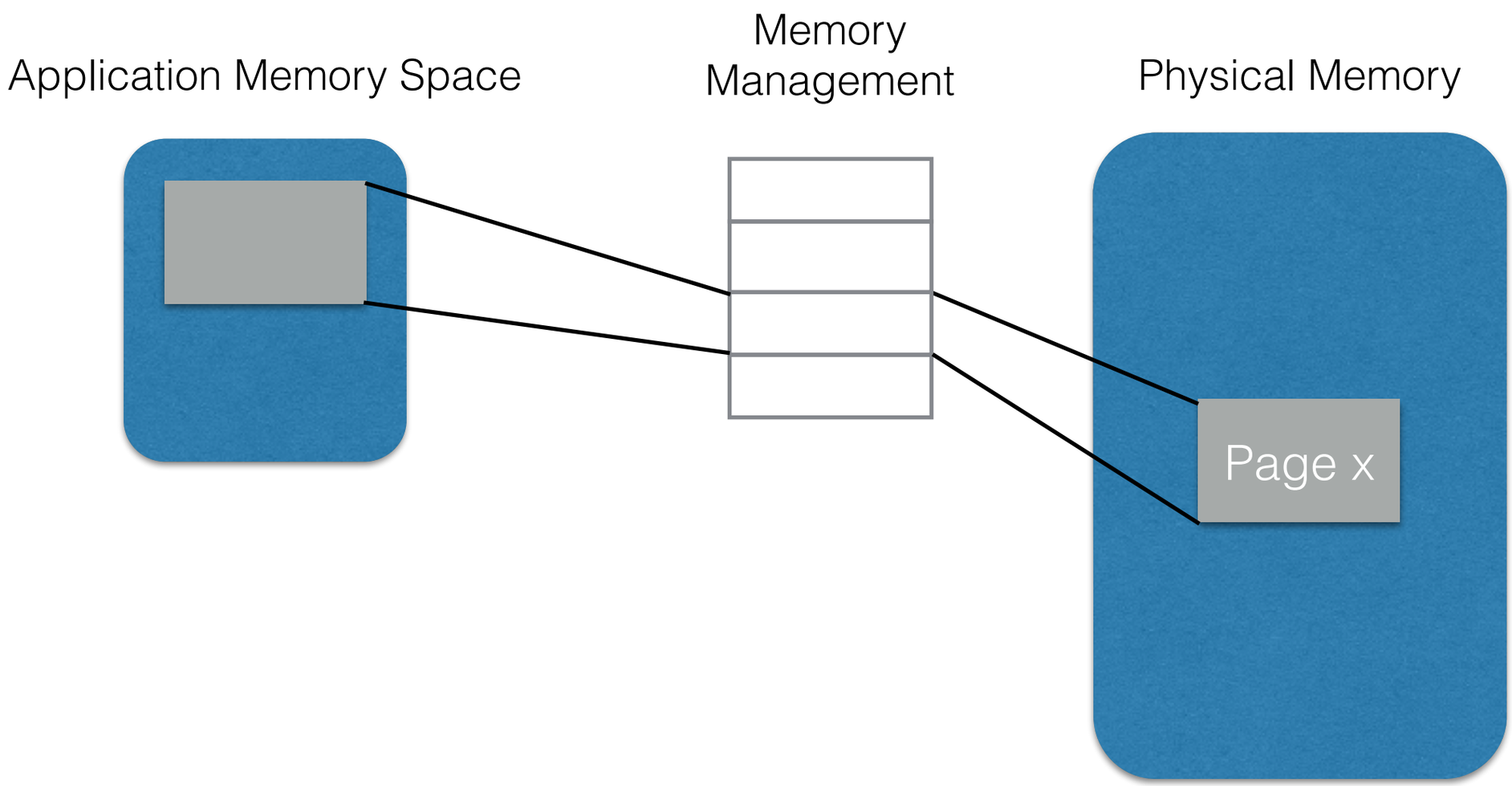
- Self-reboot



What if the data is stored at a fixed physical location?

Rethinking Memory Volatility in smartphones

- System freeze

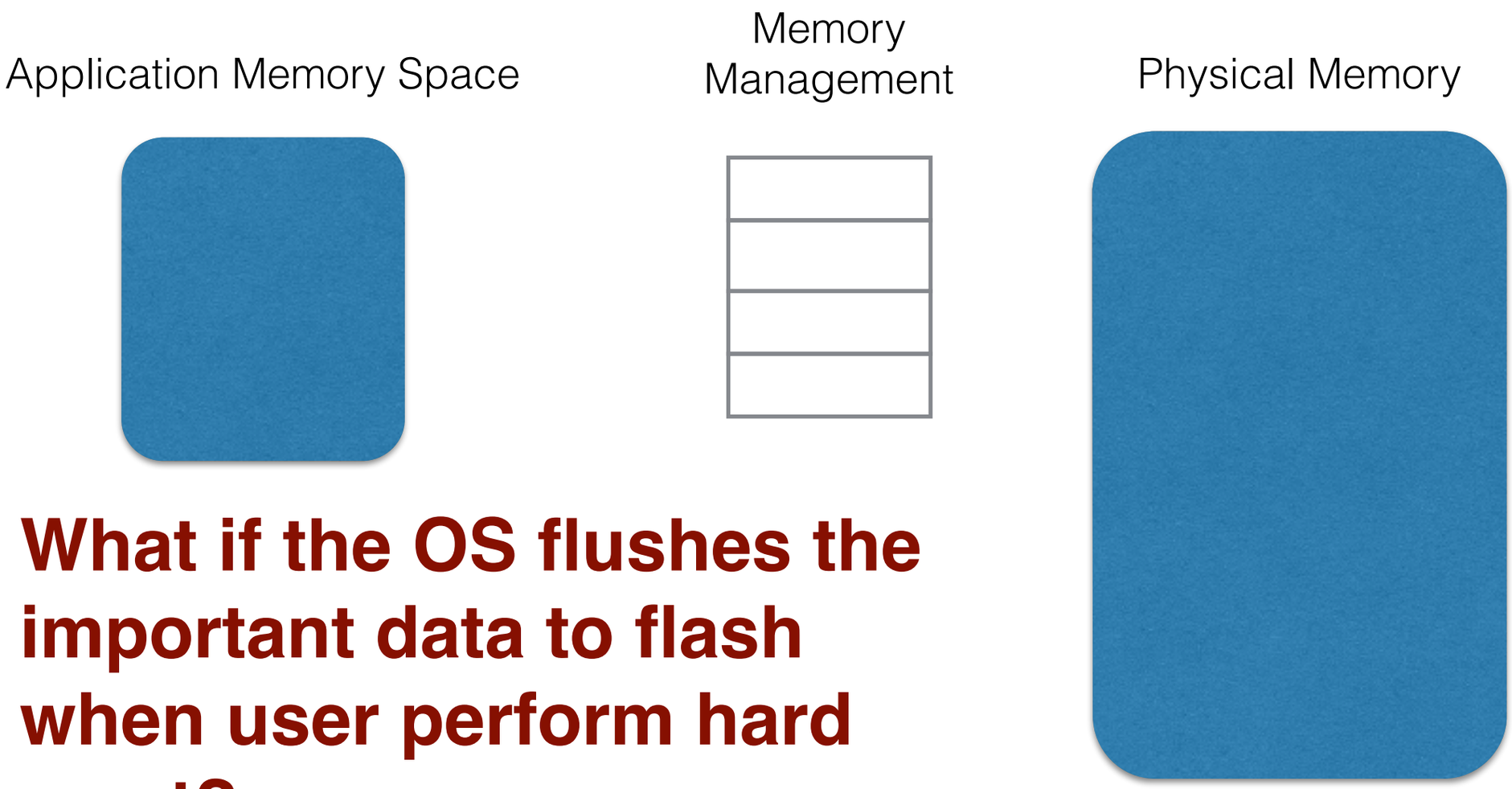


Rethinking Memory Volatility in smartphones

- System freeze
  - user perform hard reset

Rethinking Memory Volatility in smartphones

- System freeze



What if the OS flushes the important data to flash when user perform hard reset?

Rethinking Memory Volatility in smartphones

- Failure mode and DRAM data

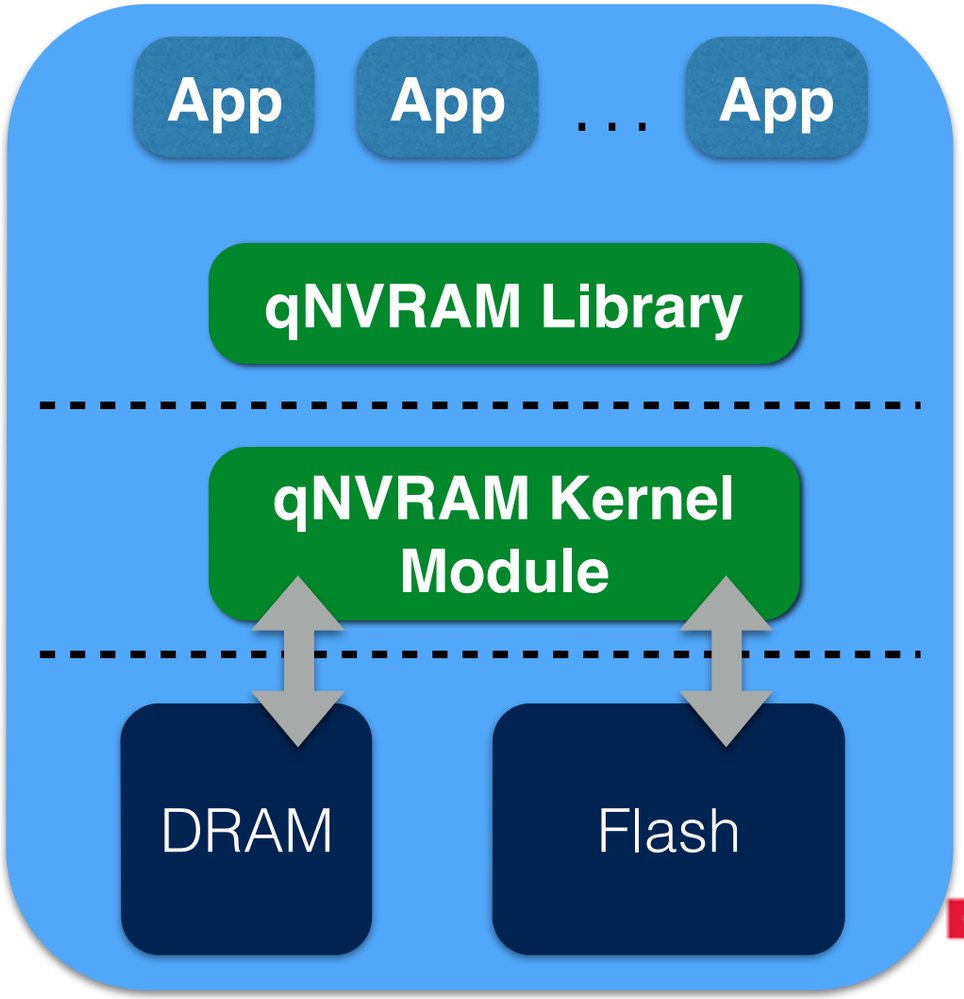
	Application space	OS	Physical memory
Application Crash	✗	✓	✓
Application Hang	✗	✓	✓
Self-reboot	✗	✗	✓
System Hang	✗	✗	✗

Outline

- Introduction
- Overhead of Persistency
- Rethinking Memory Volatility in Smartphones
- **Design of qNVRAM**
- Case Study

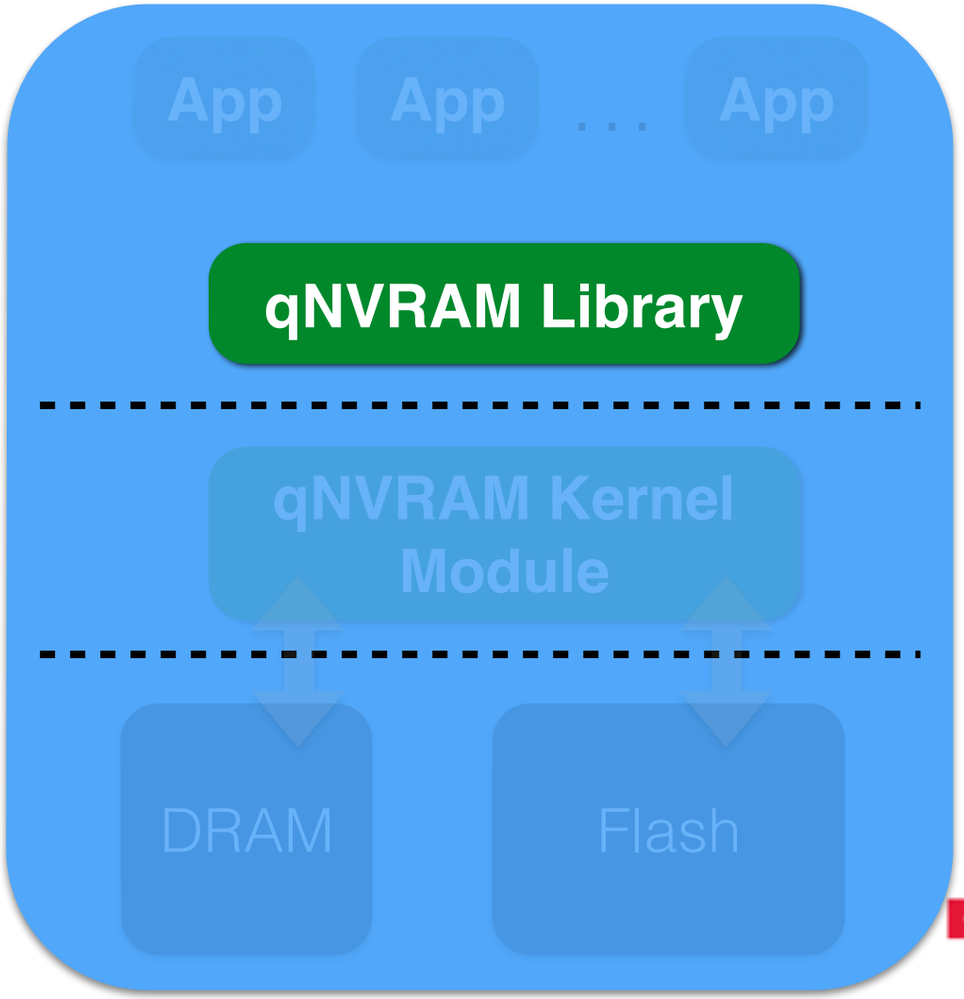
Design of qNVRAM

- qNVRAM: quasi Non-Volatile RAM
  - user library
  - kernel module



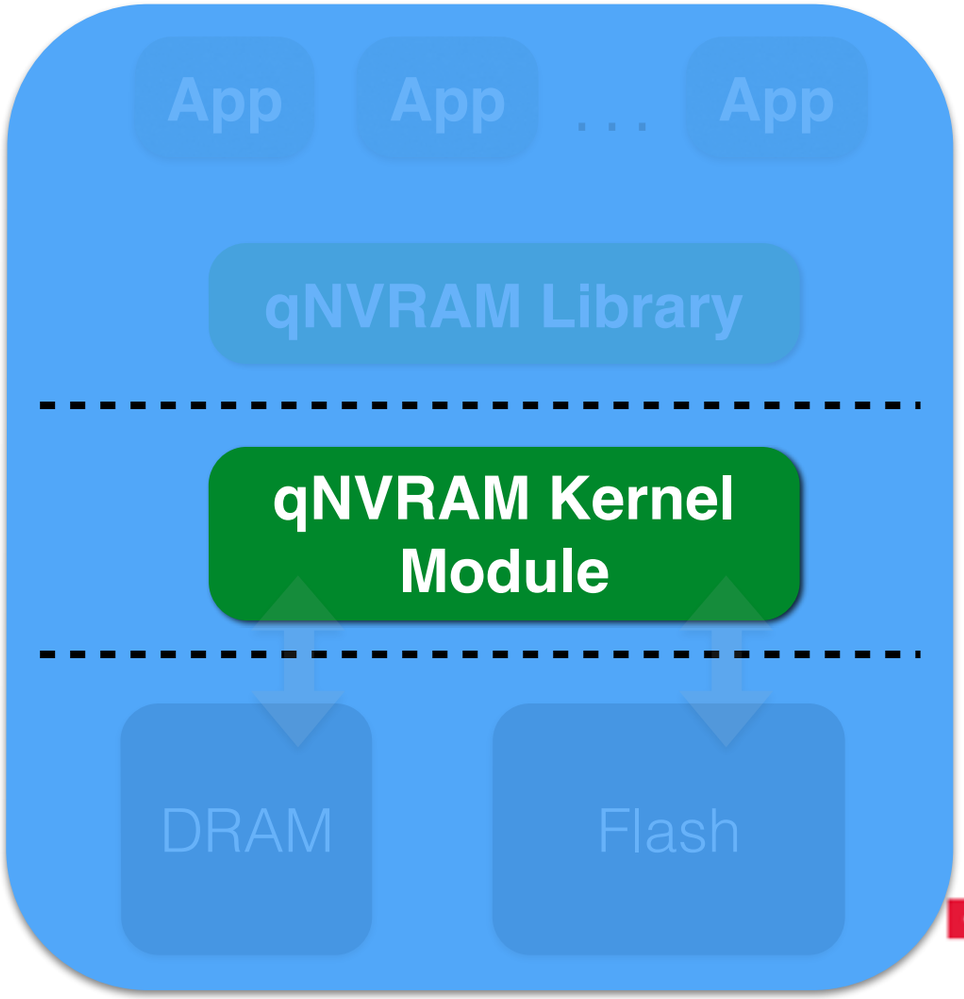
Design of qNVRAM

- qNVRAM library provides easy-to-use APIs
  - alloc()
  - free()
  - retrieve()



Design of qNVRAM

- Kernel module manages a small chunk of physical memory.
  - reserved during boot up
  - fixed physical address
  - emergency flush

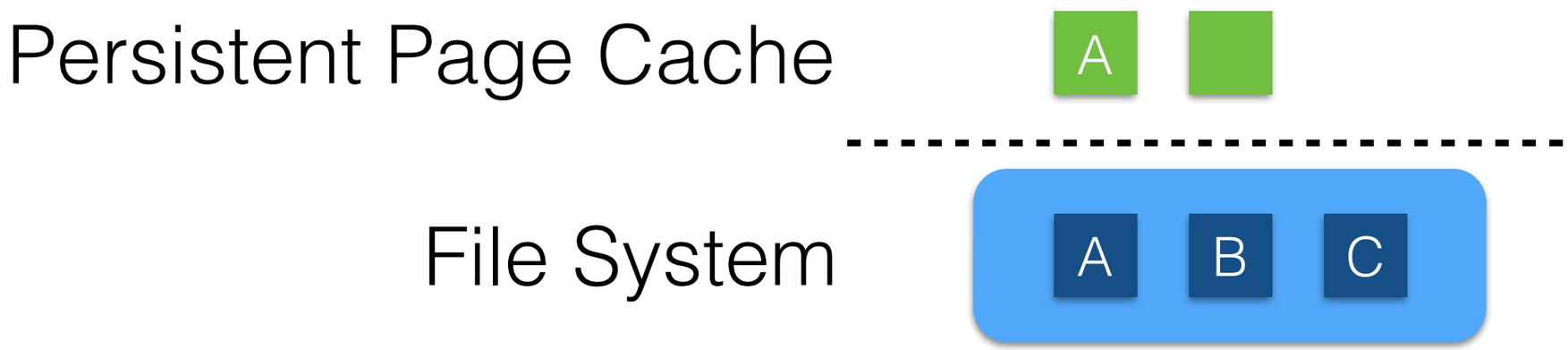


Outline

- Introduction
- Overhead of Persistency
- Rethinking Memory Volatility in smartphones
- Design of qNVRAM
- **Case Study**

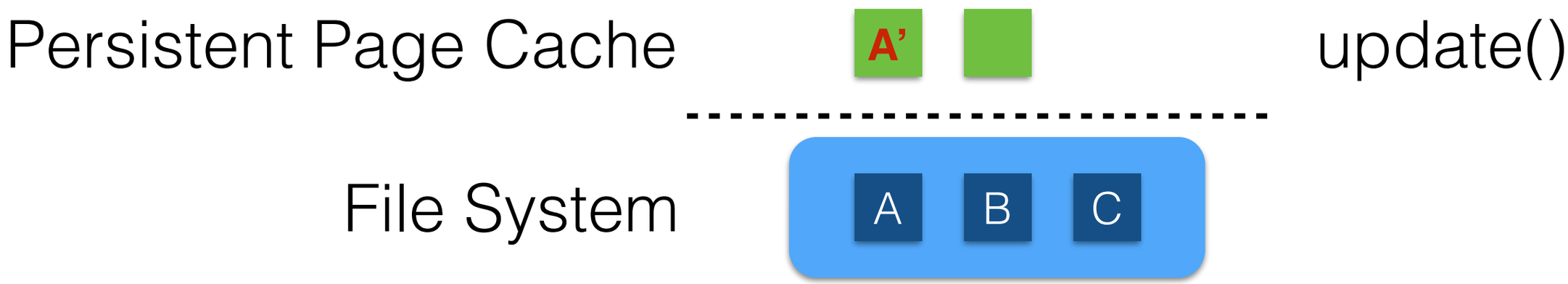
Case Study

- Persistent Page Cache in SQLite
  - In-place update



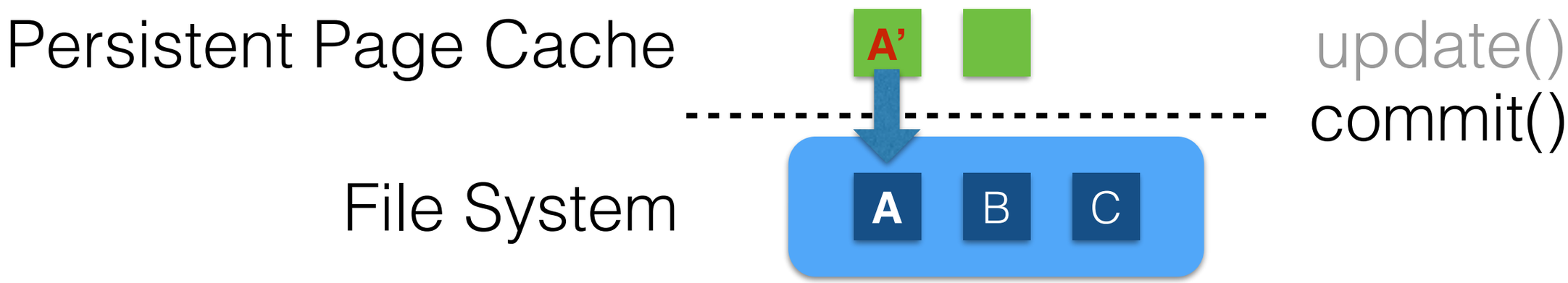
Case Study

- Persistent Page Cache in SQLite
  - In place update



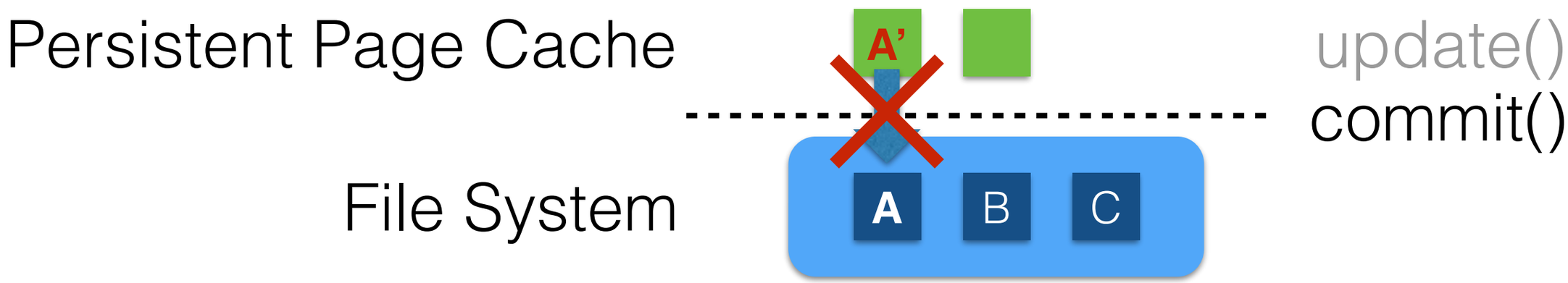
Case Study

- Persistent Page Cache in SQLite
  - In place update



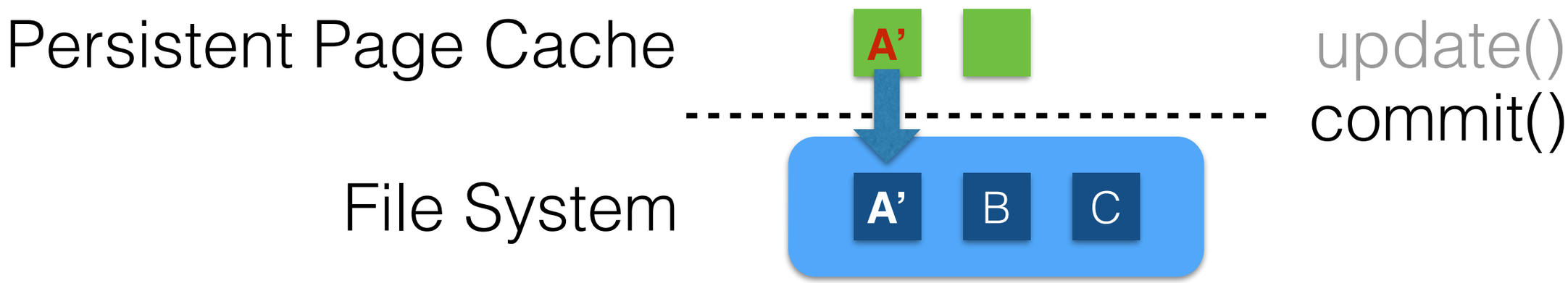
Case Study

- Persistent Page Cache in SQLite
  - In place update
  - Recover from persistent page cache upon failure.



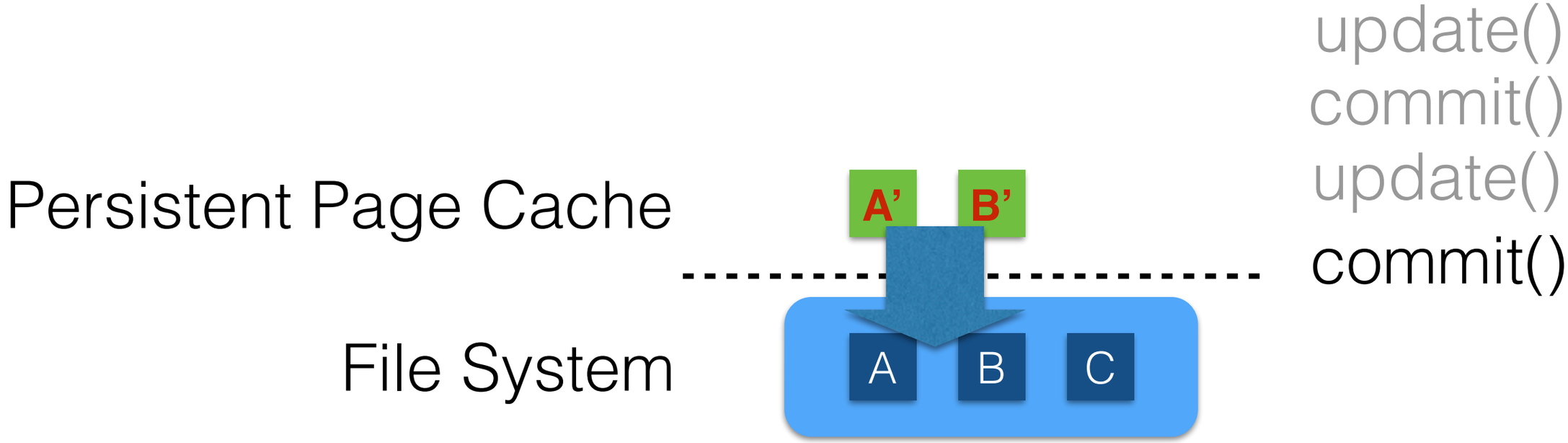
Case Study

- Persistent Page Cache in SQLite
  - In place update
  - Recover from persistent page cache upon failure.



Case Study

- Lazy Flush
  - Dirty pages are flushed asynchronously.

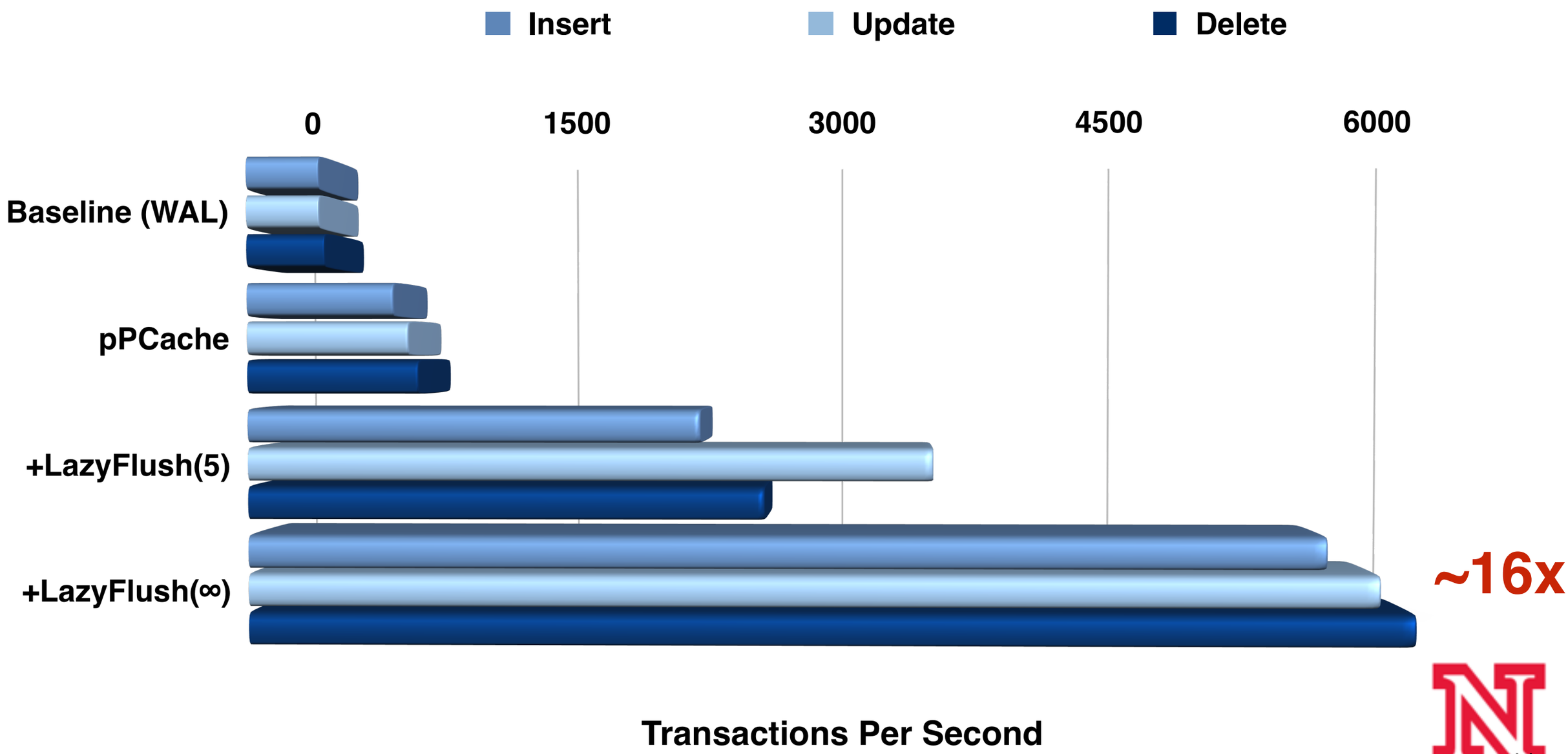


Case Study

- Performance evaluation
  - Samsung Galaxy S4 Google Edition
    - 4 core CPU
    - 2GB RAM
    - 16GB flash
  - SQLite 3.7.12
    - Persistent page cache
    - Lazy Flush

Case Study

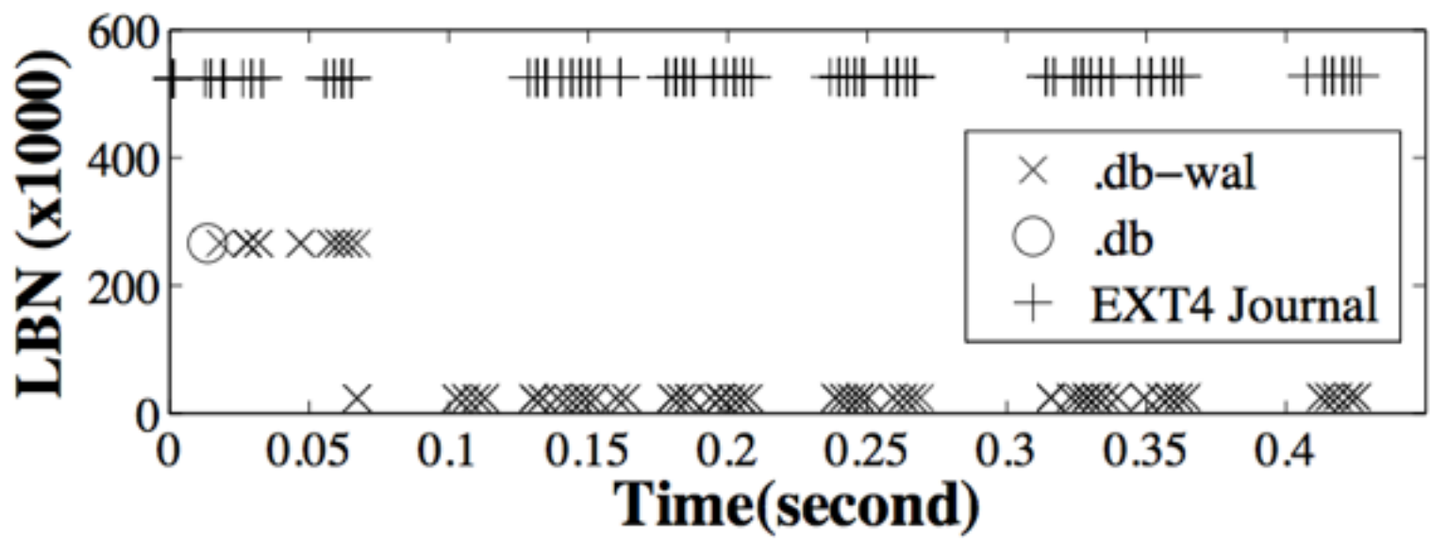
- Performance evaluation



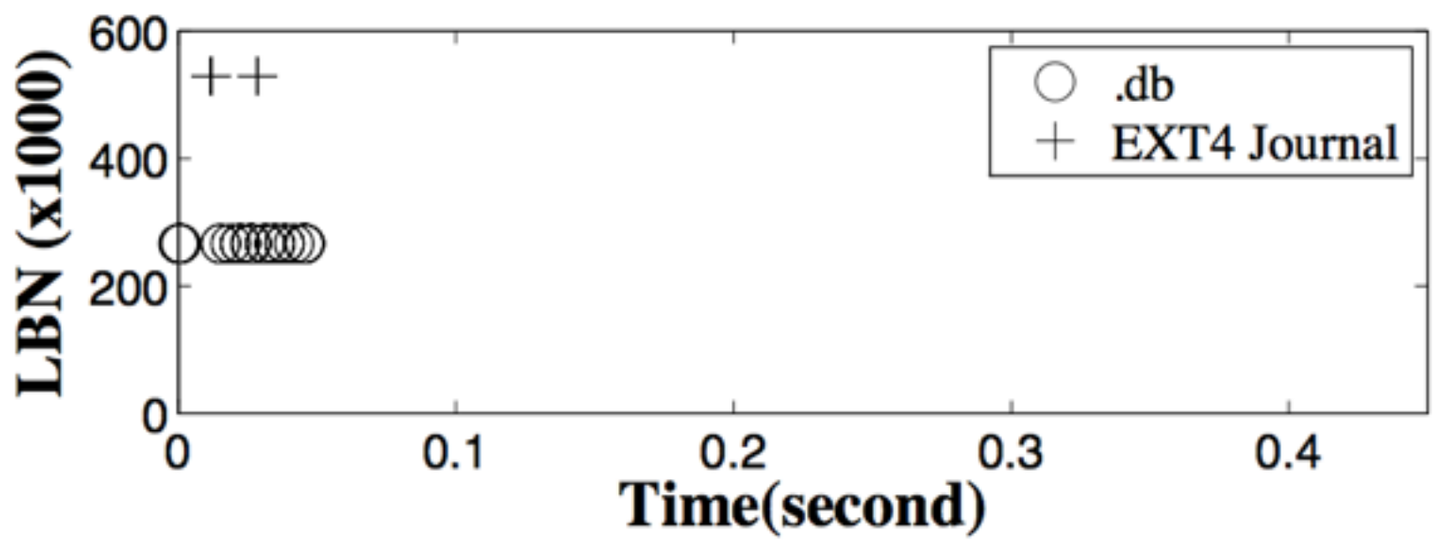
Case Study

- Performance evaluation

WAL



LazyFlush (5)



Conclusion

- qNVRAM takes advantage of the battery-backed characteristic of smartphones.
- qNVRAM can speed up SQLite by up to 16x.


Q.
A.