Write Dependency Disentanglement with HORAE

Xiaojian Liao, Youyou Lu, Erci Xu, Jiwu Shu

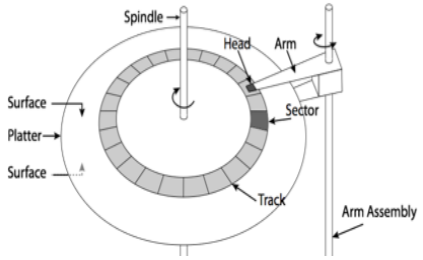
Tsinghua University

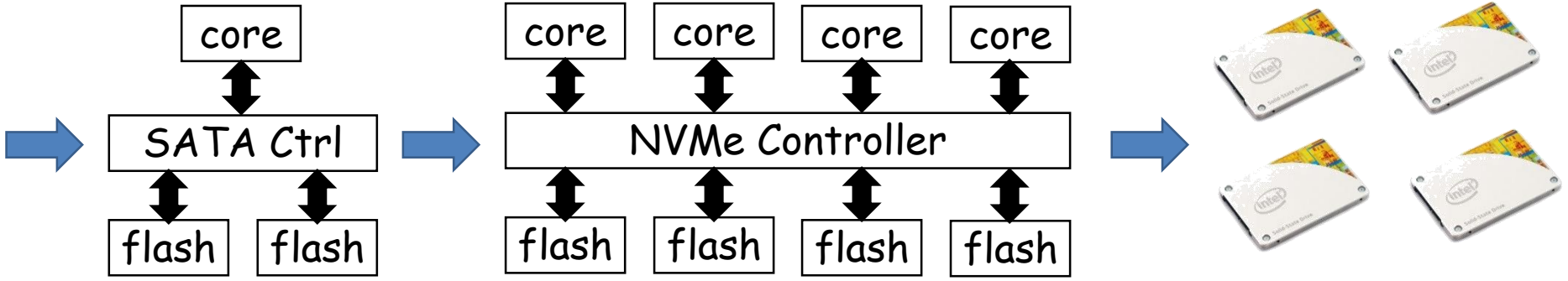
1. Background and Motivation

Background: Storage Trend

Modern storage delivers higher bandwidth and concurrency.

	HDD	SATA SSD	NVME SSD	All SSD Array
Bandwidth	~80MB/s	~500MB/s	~6GB/s	~48GB/s





Single-Head Concurrency (1)

Multi-Channel Concurrency (1~8)

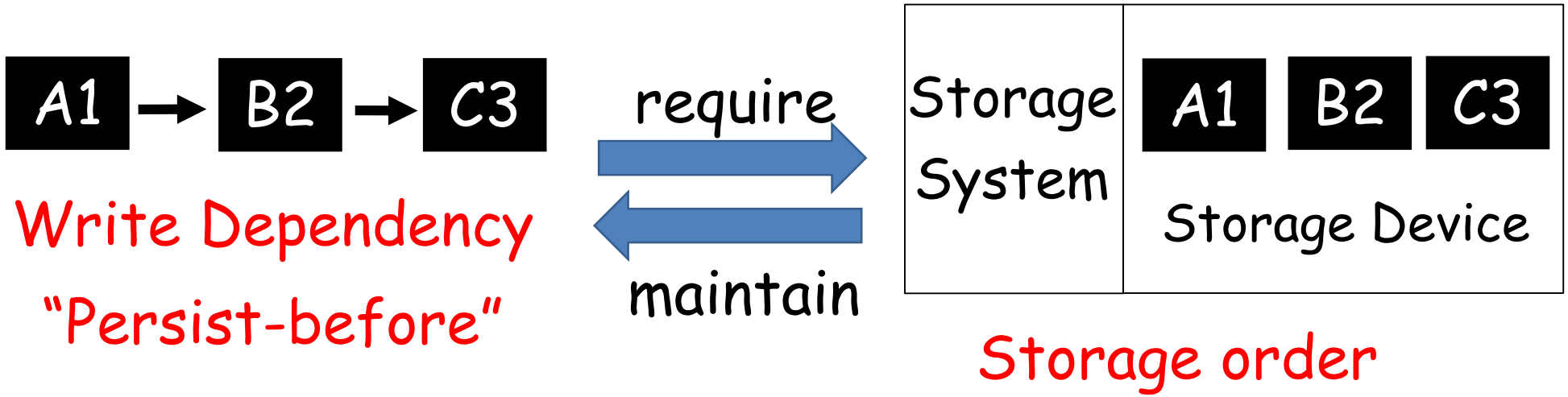
Multi-Channel Concurrency (>=32)

Multi-Device Concurrency (> =256)

Background: Write Dependency

File System Journaling | COW-based File System | Soft Updates | ……

Consistency



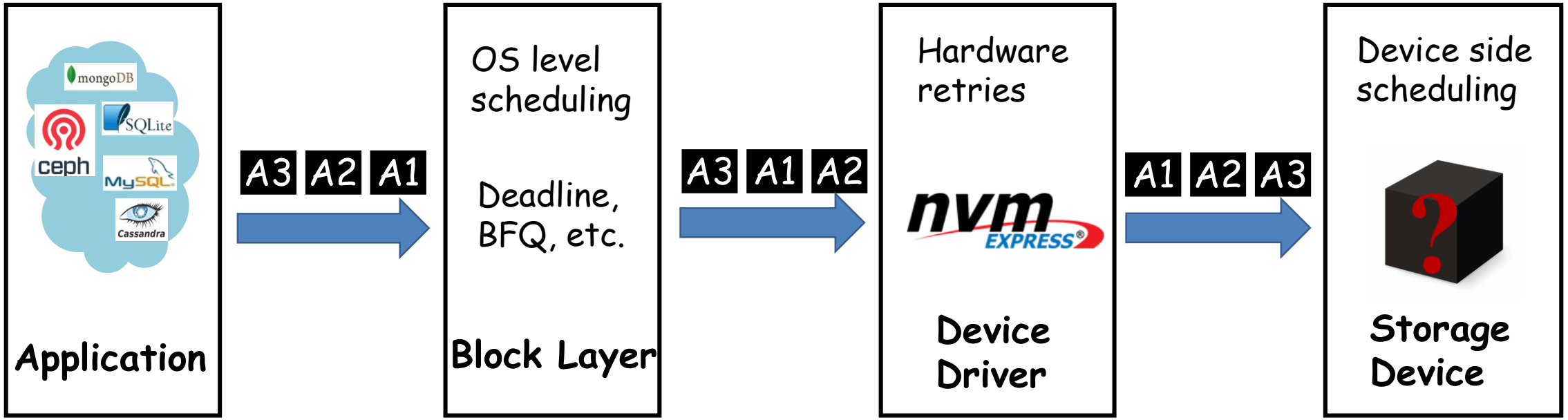
Write Dependency

"Persist-before"

Storage order

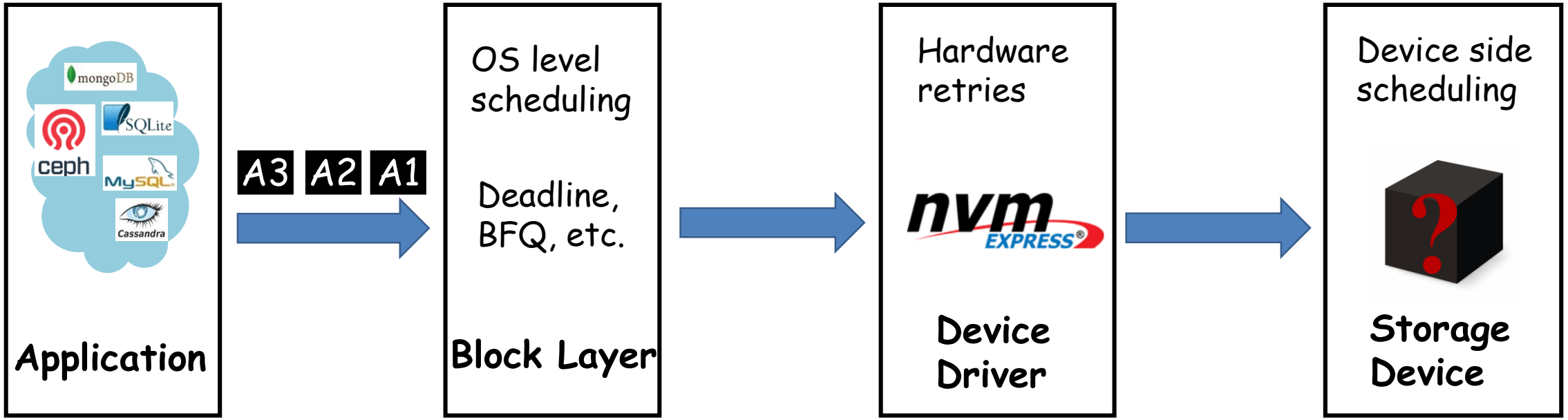
Background: Storage Order

Modern IO stack is Orderless.



Background: Storage Order

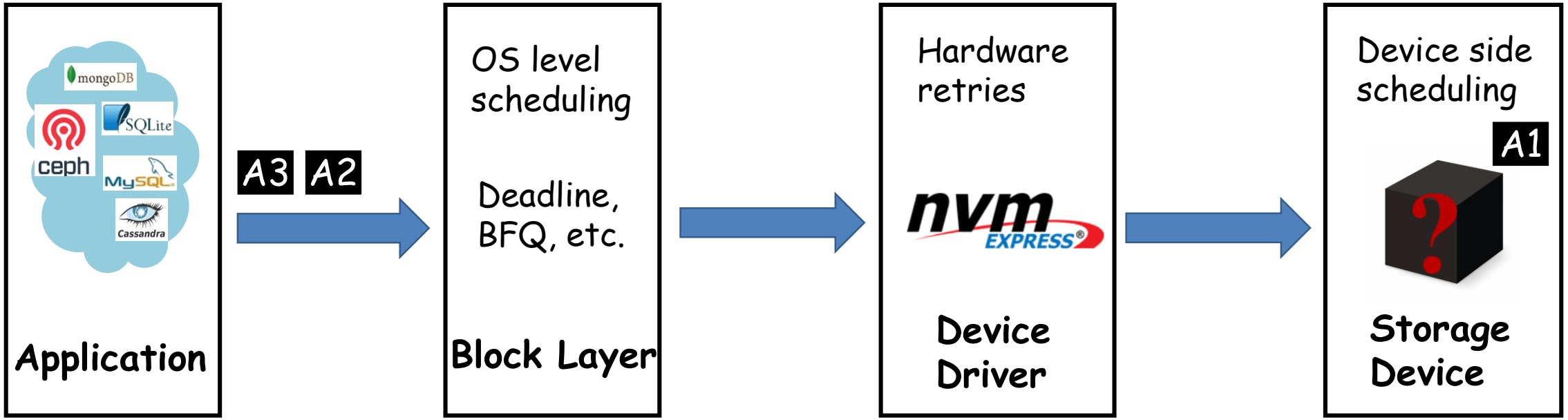
To enforce storage order, current IO stack should process only one set of orderless IOs at a time.



Serialization ✖ **Concurrency**

Background: Storage Order

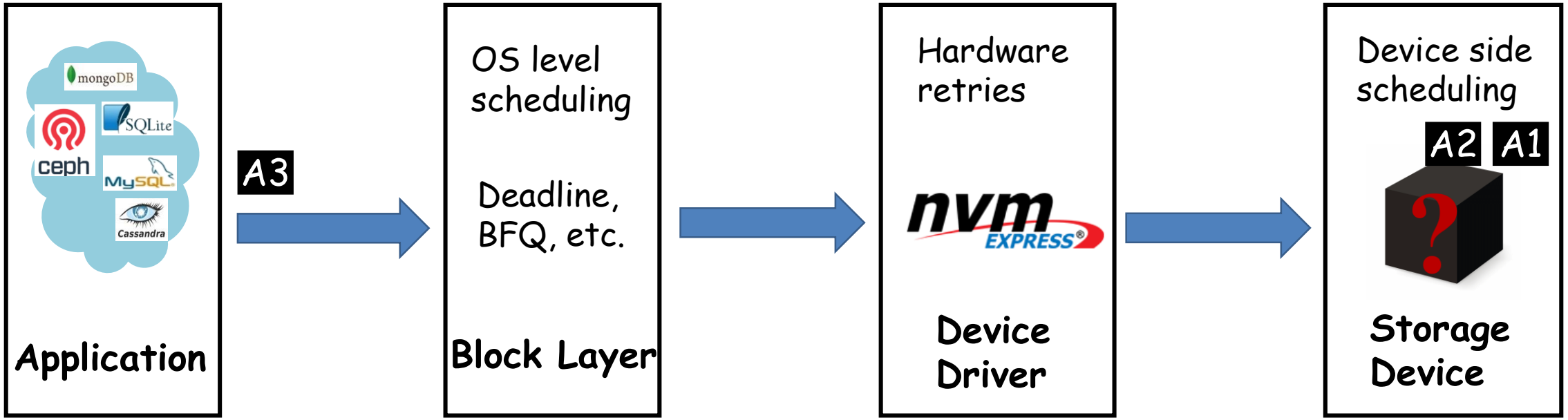
To enforce storage order, current IO stack should process only one set of orderless IOs at a time.



Serialization ✖ Concurrency

Background: Storage Order

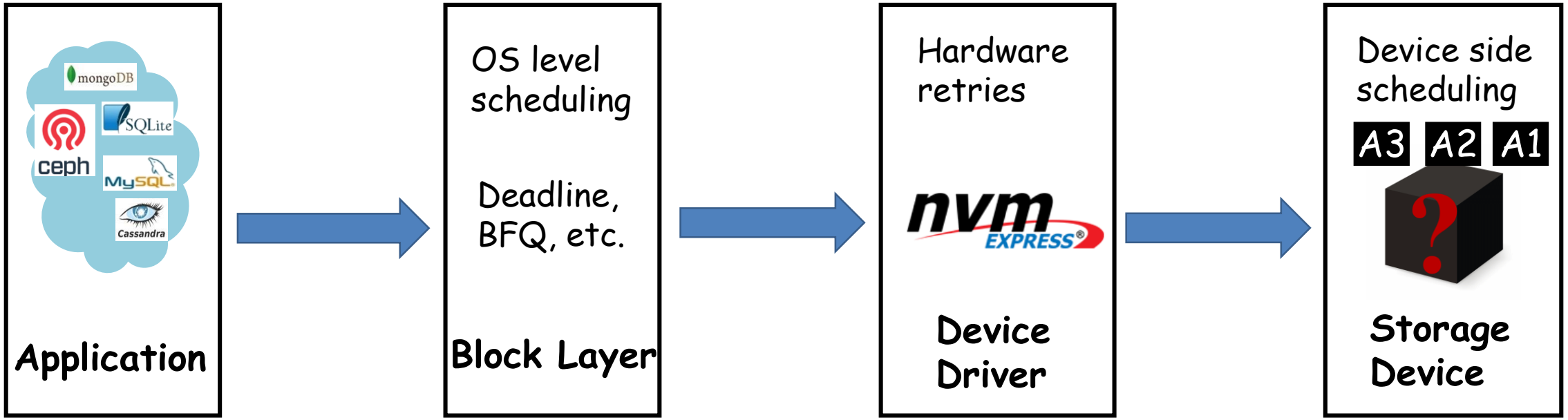
To enforce storage order, current IO stack should process only one set of orderless IOs at a time.



Serialization ✖ Concurrency

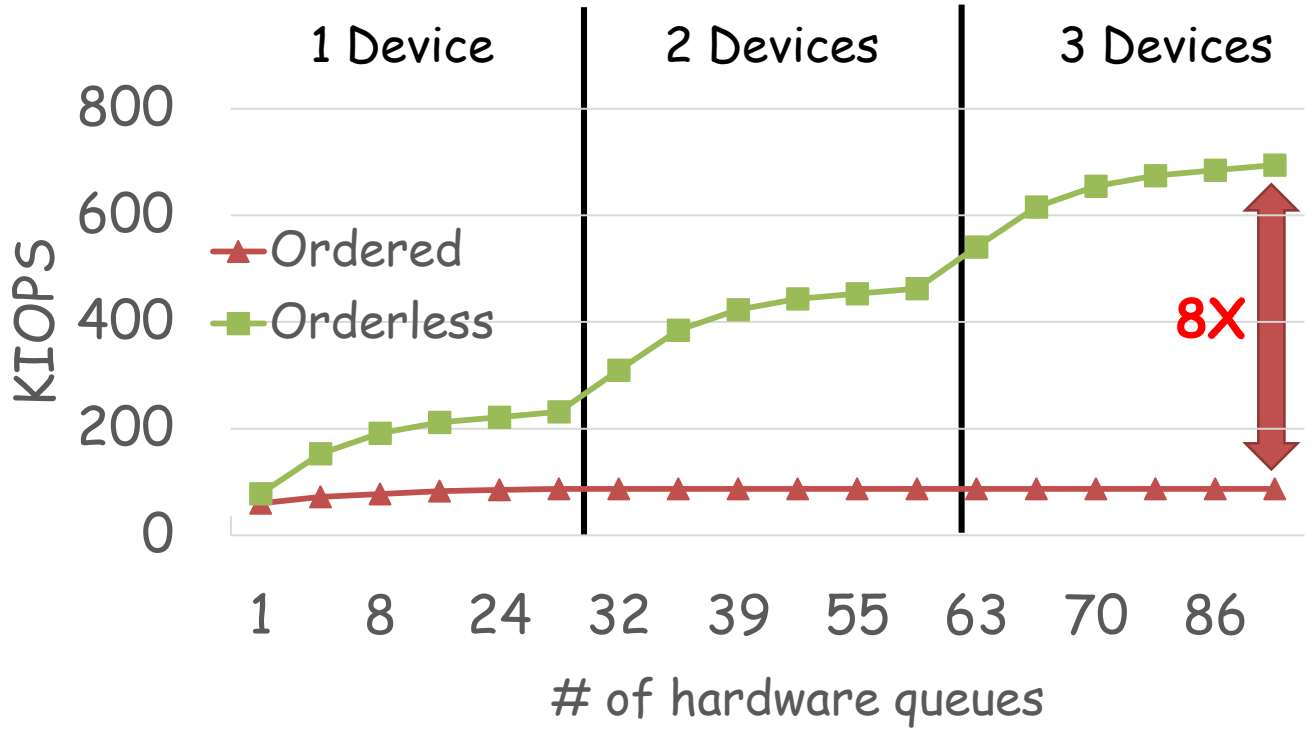
Background: Storage Order

To enforce storage order, current IO stack should process only one set of orderless IOs at a time.



Serialization ✖ **Concurrency**

Motivation



Test tool: fio

IO size: 4 KB

Threads: up to 4

Test SSD: Intel 750

Ordered writes -> Hard to grow!

Orderless writes-> Full bandwidth!

2. Design

Can we and how to achieve storage order with full bandwidth?

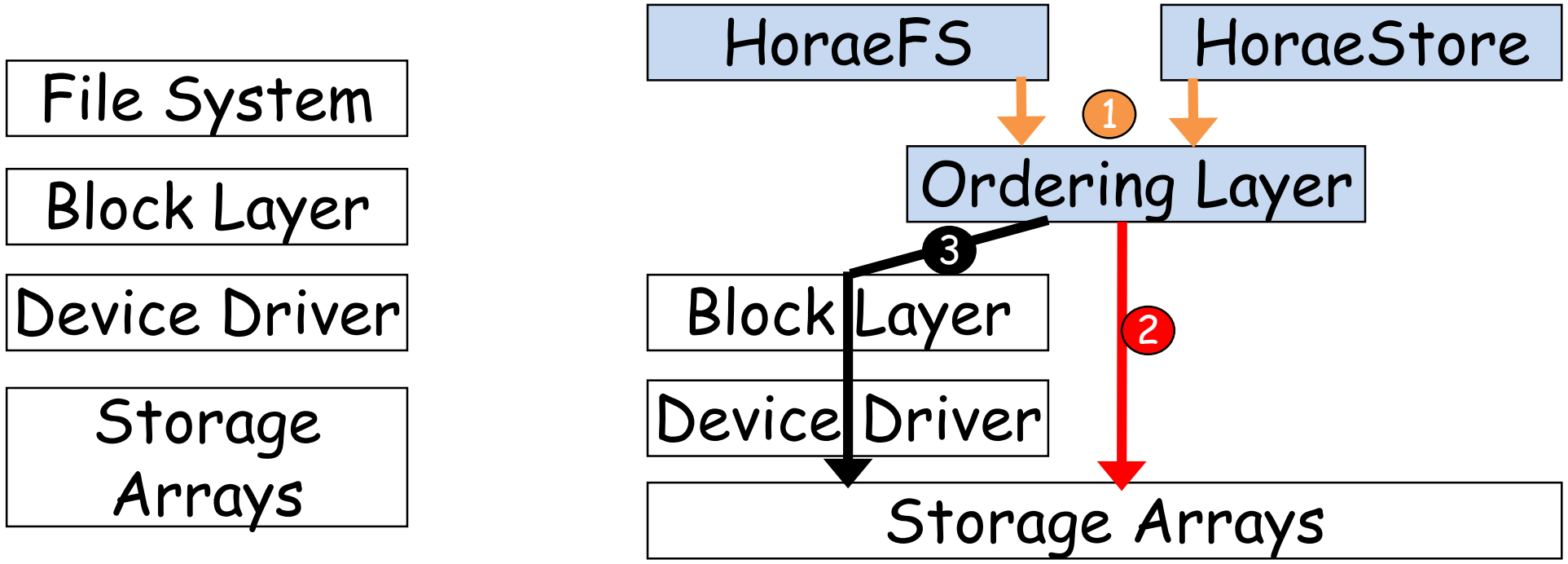
Yes, with Horae.

Horae IO stack Overview

Key idea: Split the IO stack into an ordered control path ❷ and an orderless data path ❸

Full bandwidth

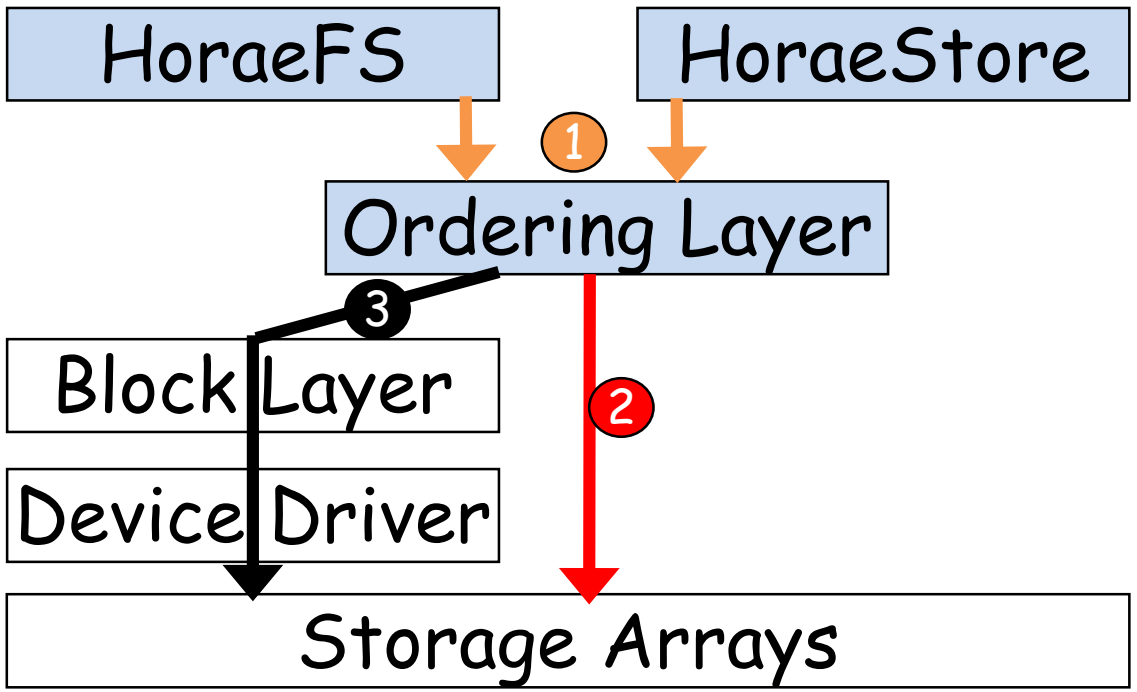
Storage order



Linux IO Stack

Horae IO Stack

Horae's Key Design



Horae IO Stack

Key idea: 2

Dedicated Control Path

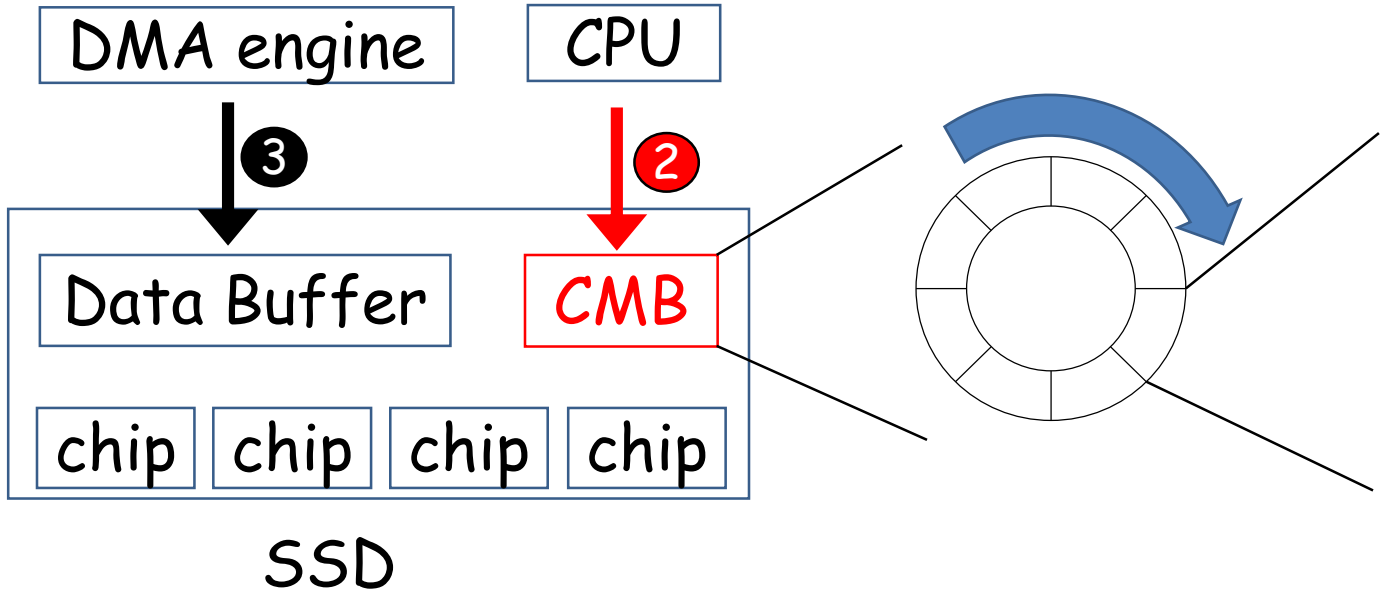
Opportunities:

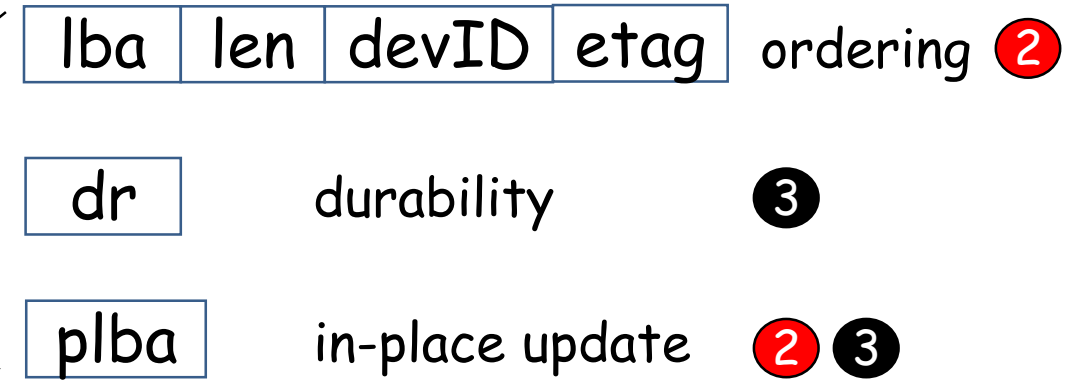
- Parallelizing durability commands 3
- Parallelizing in-place updates 2 3

Use cases: 1

- File System: HoraeFS
- Object Store: HoraeStore

Dedicated Control Path





CMB (persistent Controller Memory Buffer) In NVMe spec.

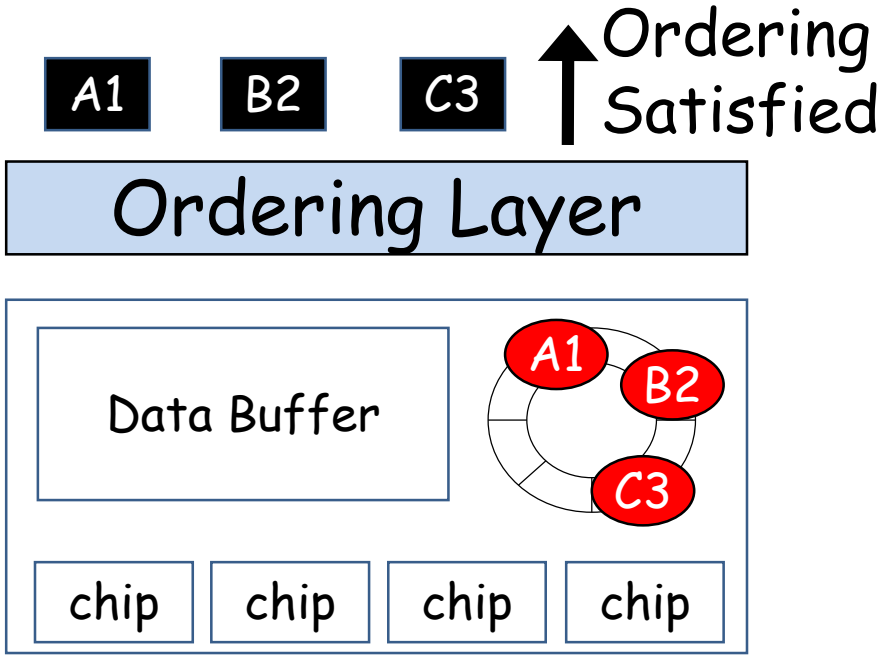
Circular Ordering Queue

16-Byte Ordering Metadata, persist latency ~0.5us

A1

Dedicated Control Path

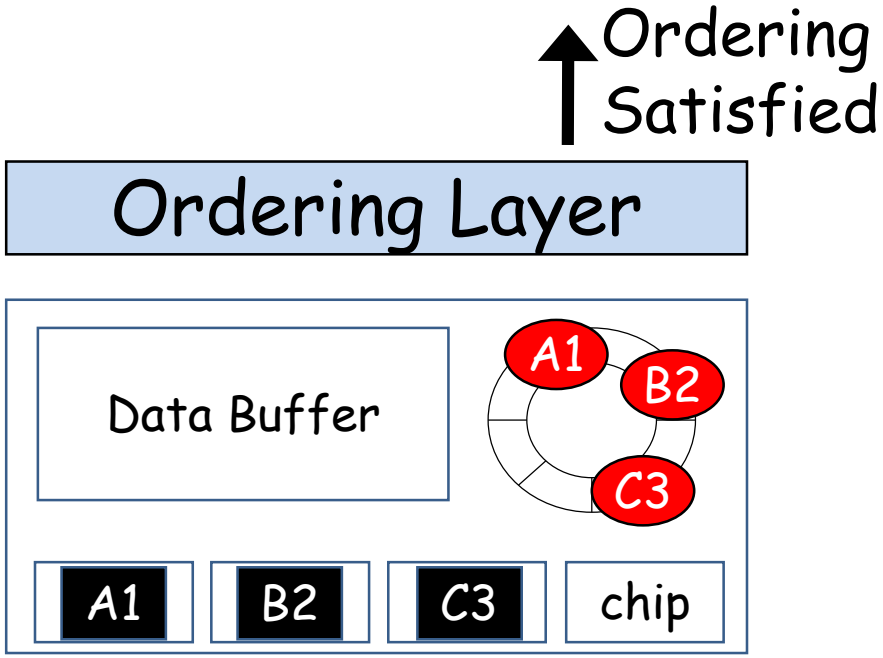
Use the ordering metadata to provide ordering guarantee during normal execution and crash recovery.



(a) Normal execution

Dedicated Control Path

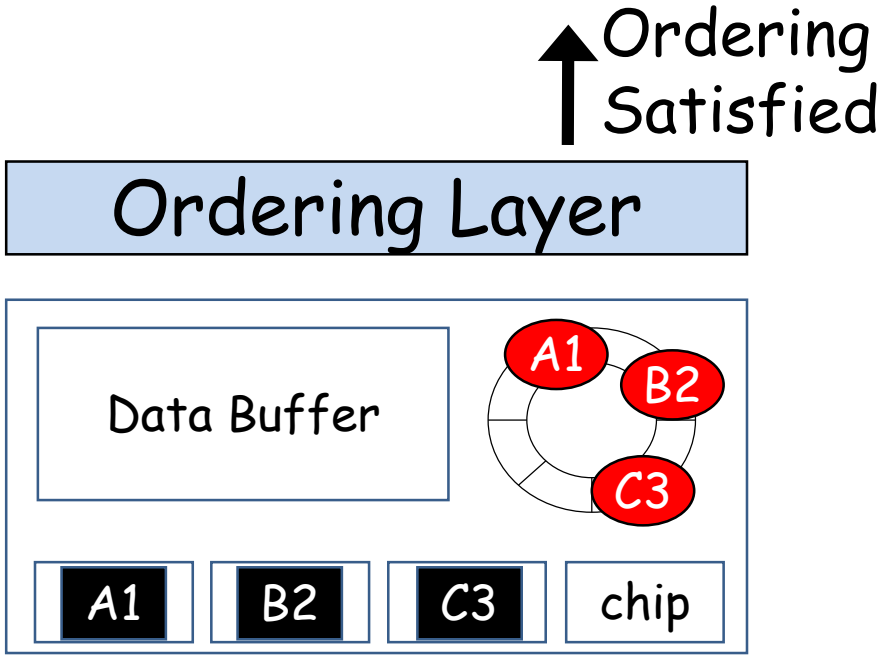
Use the ordering metadata to provide ordering guarantee during normal execution and crash recovery.



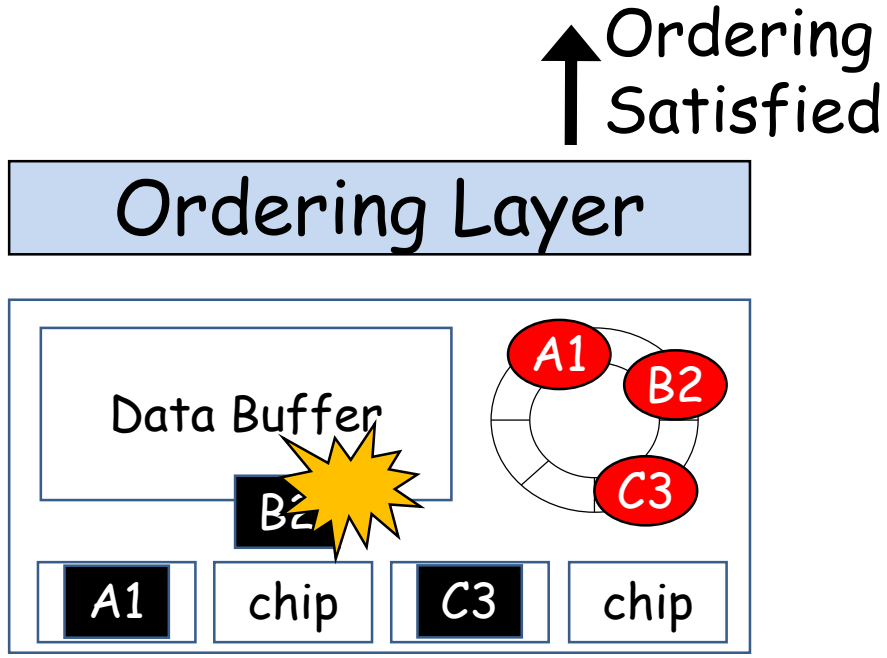
(a) Normal execution

Dedicated Control Path

Use the ordering metadata to provide ordering guarantee during normal execution and crash recovery.



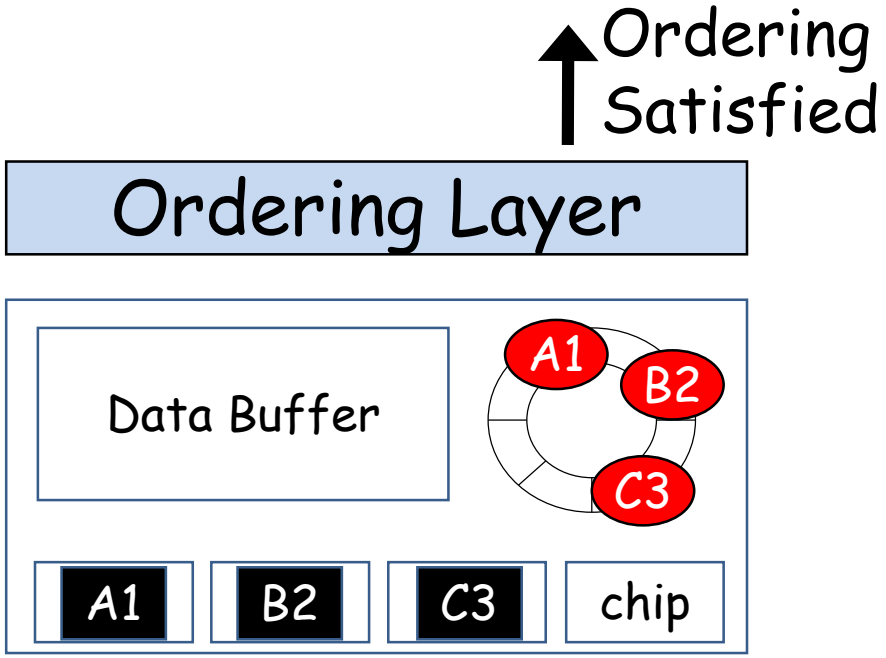
(a) Normal execution



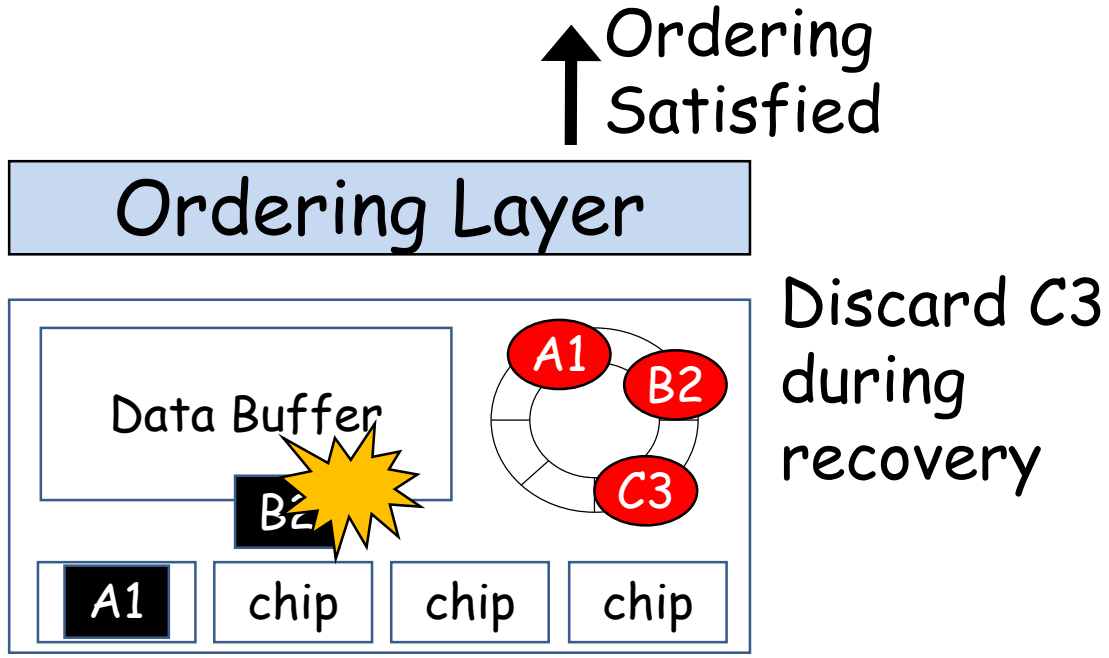
(b) Crash recovery

Dedicated Control Path

Use the ordering metadata to provide ordering guarantee during normal execution and crash recovery.



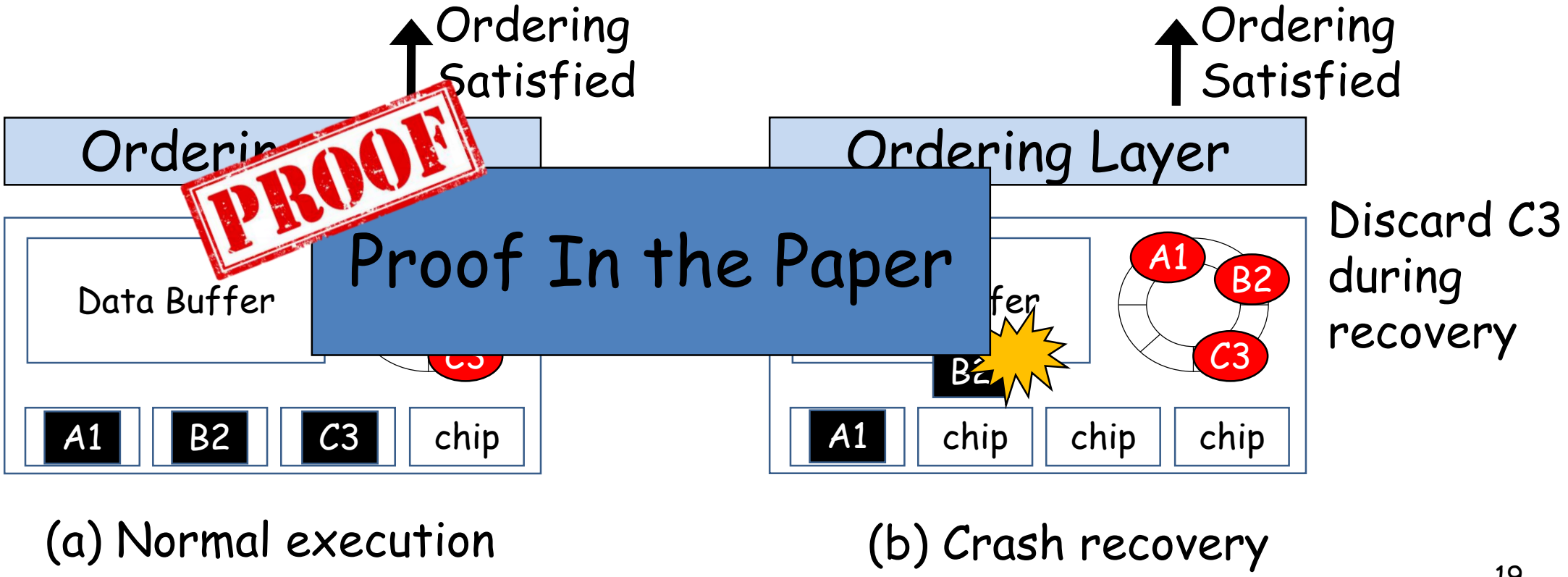
(a) Normal execution



(b) Crash recovery

Dedicated Control Path

Use the ordering metadata to provide ordering guarantee during normal execution and crash recovery.

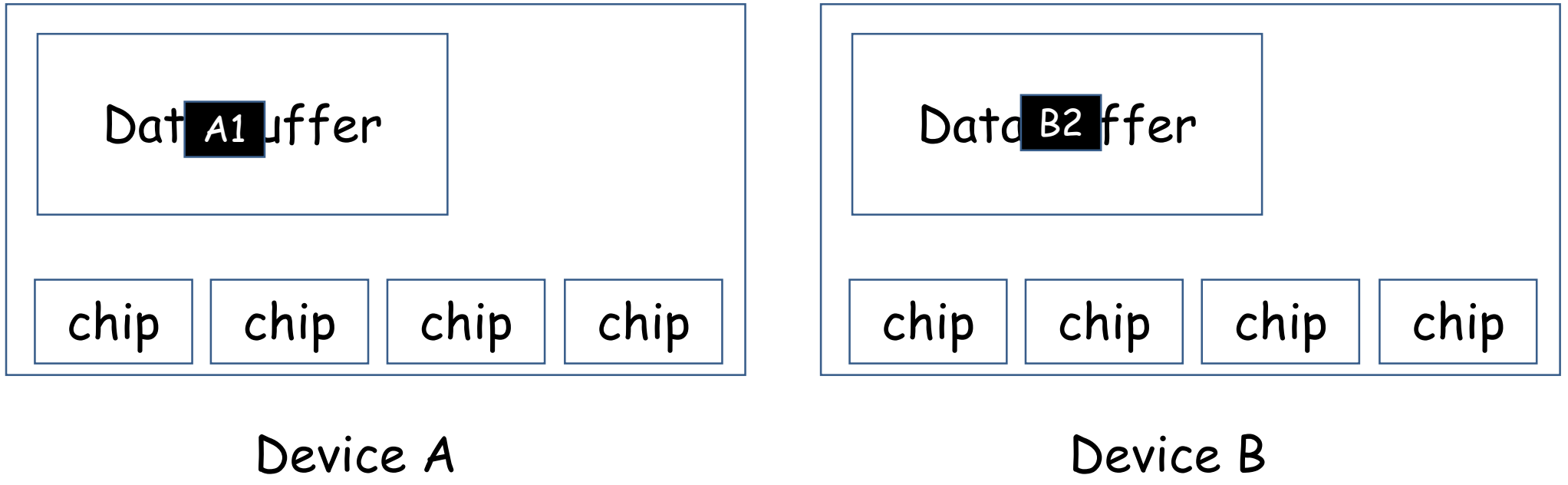


(a) Normal execution

(b) Crash recovery

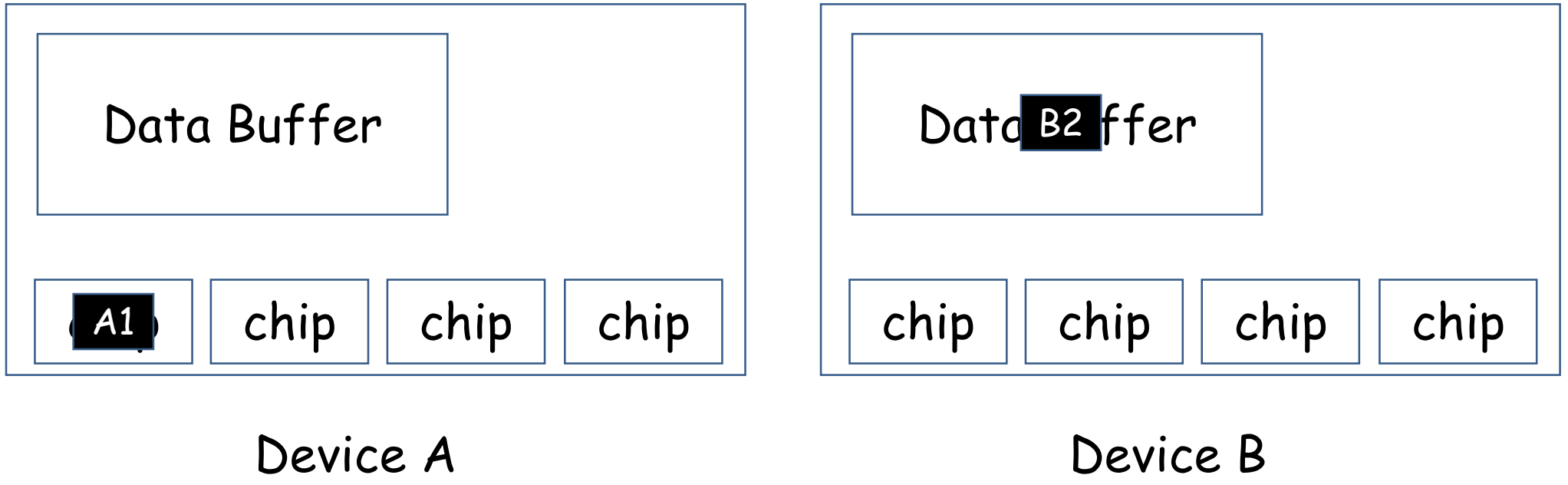
Classic Durability Guarantee

- Linux uses FLUSH command to synchronously and serially drain out the possibly volatile data buffer in each device.



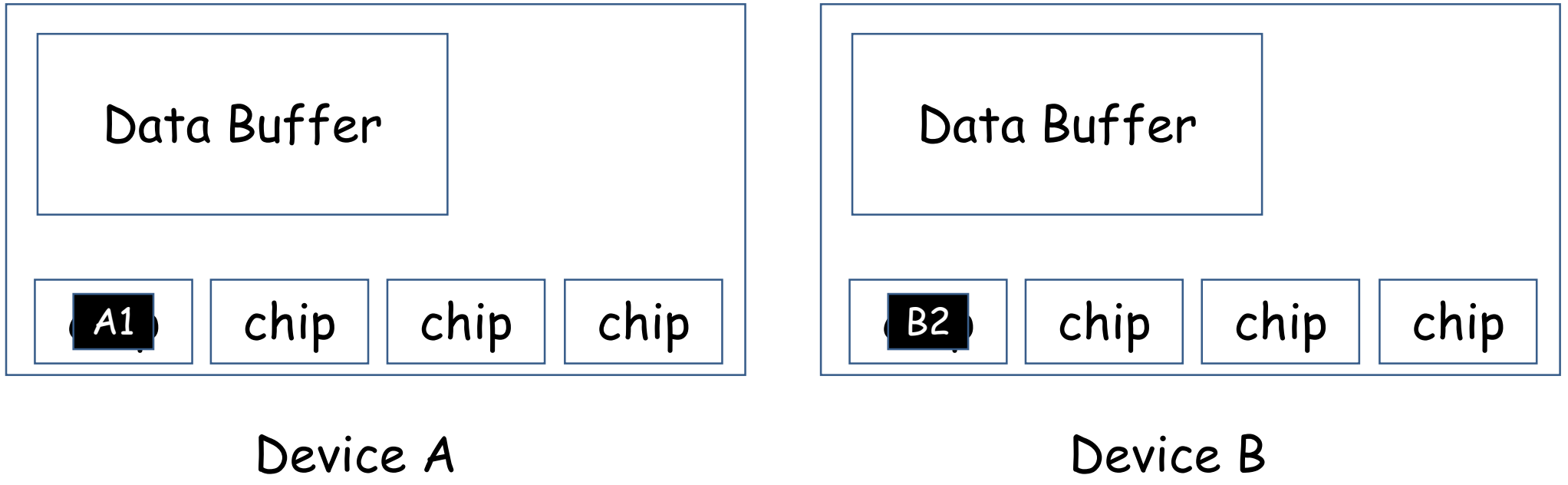
Classic Durability Guarantee

- Linux uses FLUSH command to synchronously and serially drain out the possibly volatile data buffer in each device.



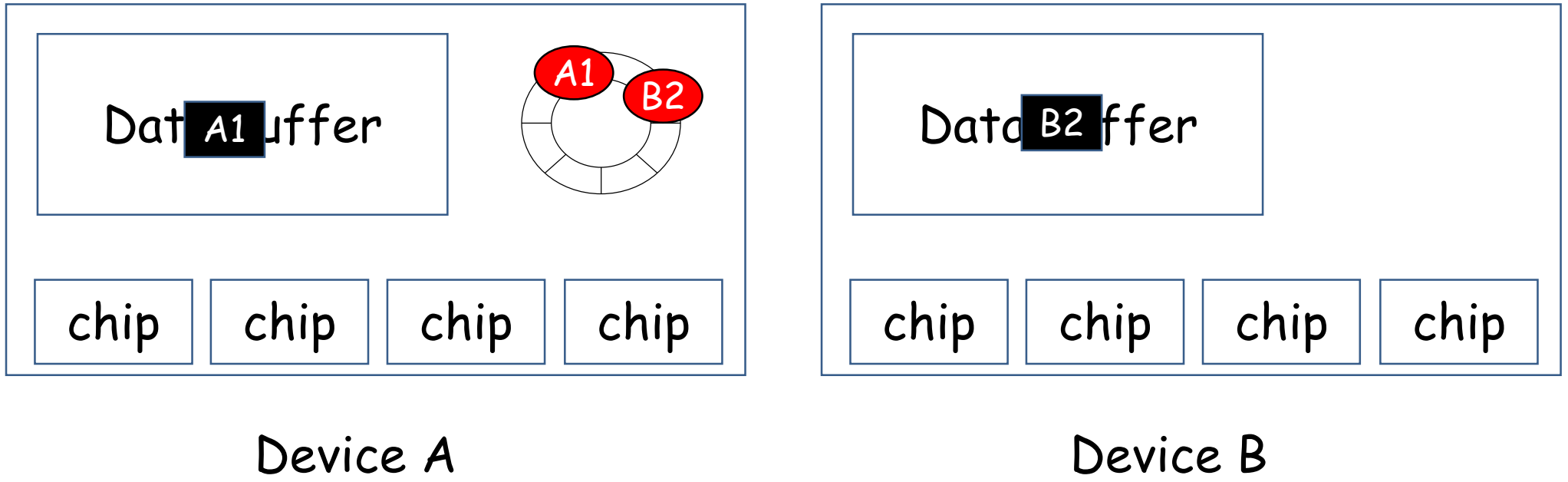
Classic Durability Guarantee

- Linux uses FLUSH command to synchronously and serially drain out the possibly volatile data buffer in each device.



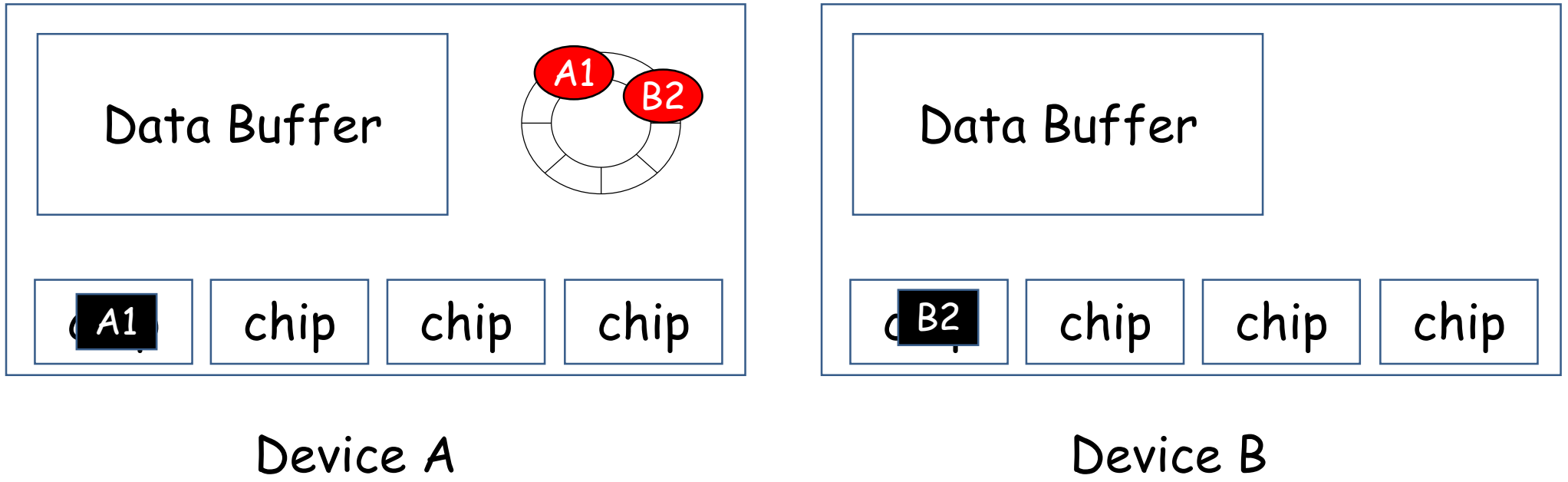
Horae's Durability Guarantee

- Horae introduces joint FLUSH to perform parallel FLUSHes in separate devices.



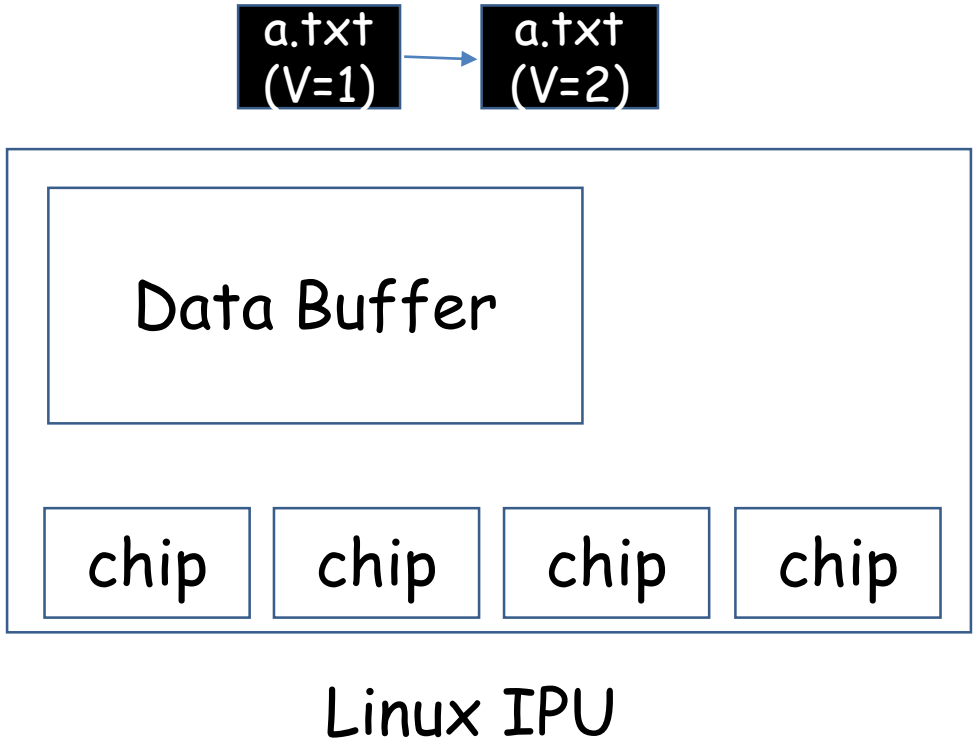
Horae's Durability Guarantee

- Horae introduces joint FLUSH to perform parallel FLUSHes in separate devices.



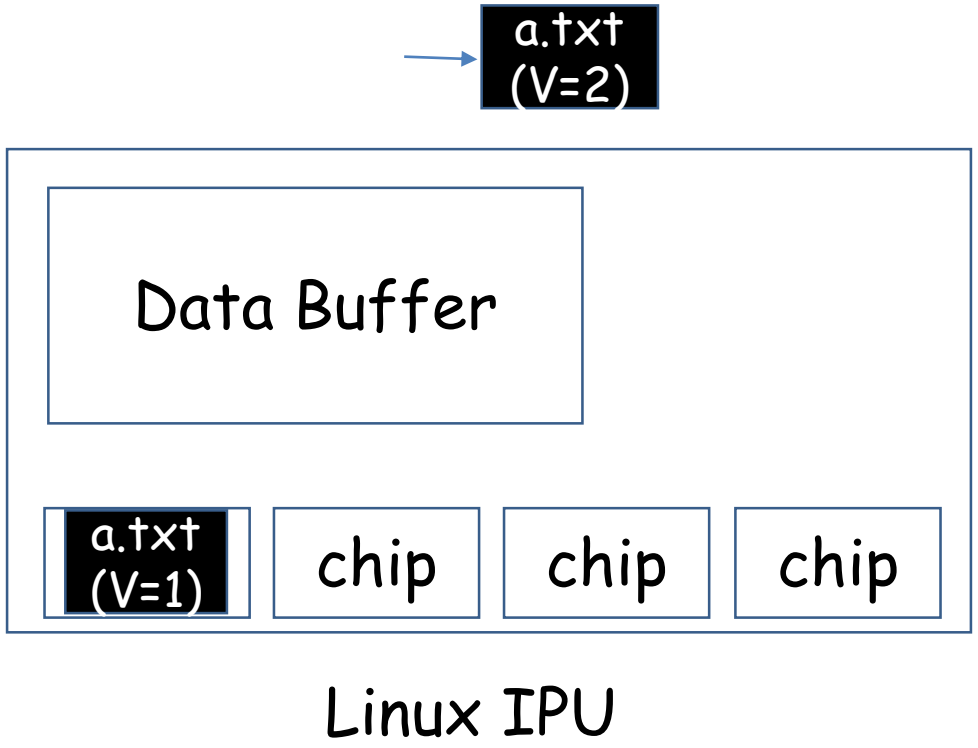
Parallelizing In-Place Updates

- Classic IO stacks serialize in-place updates.



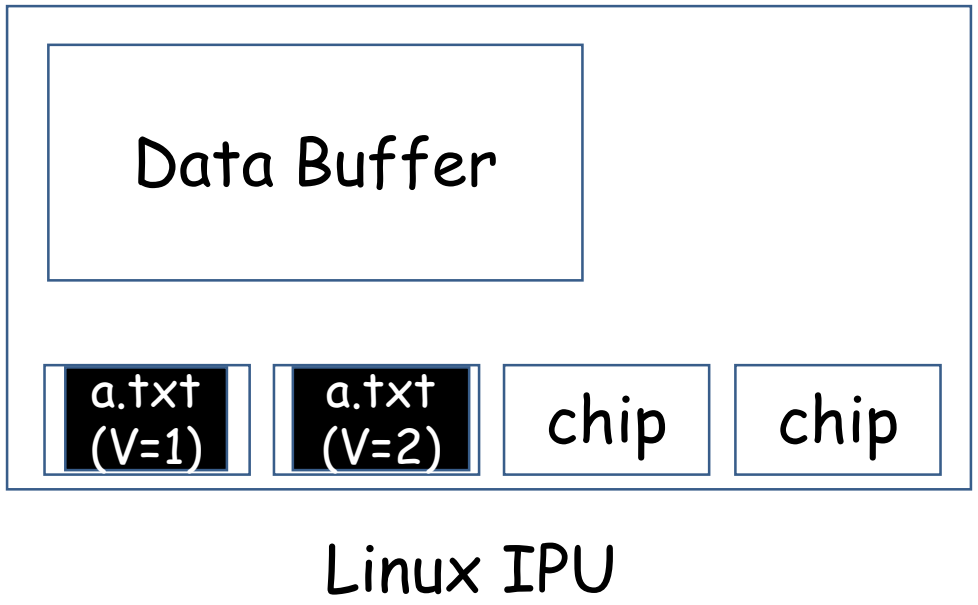
Parallelizing In-Place Updates

- Classic IO stacks serialize in-place updates.



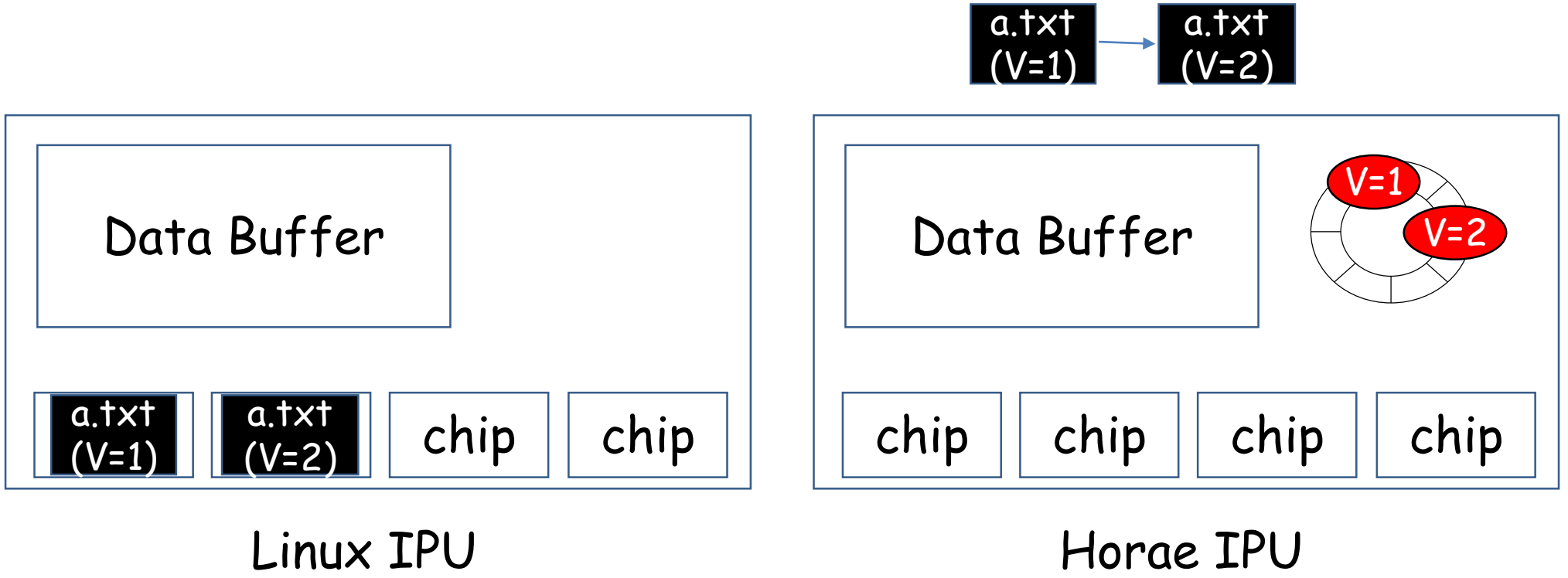
Parallelizing In-Place Updates

- Classic IO stacks serialize in-place updates.



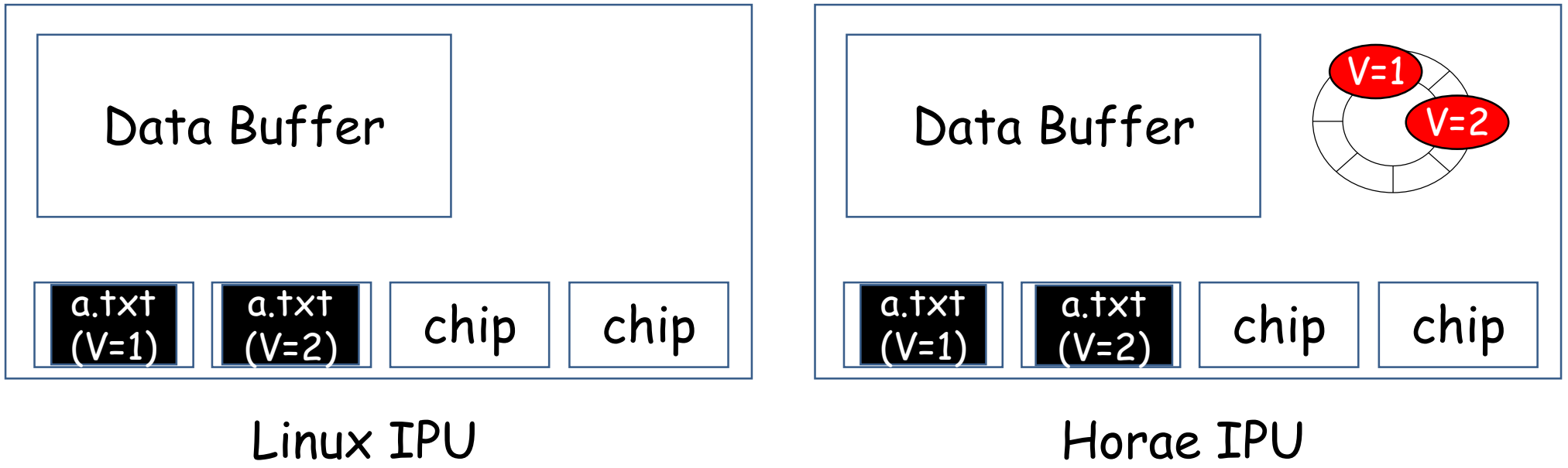
Parallelizing In-Place Updates

- Classic IO stacks serialize in-place updates.
- Horae parallelizes in-place updates through write redirection using the plba field of ordering metadata.



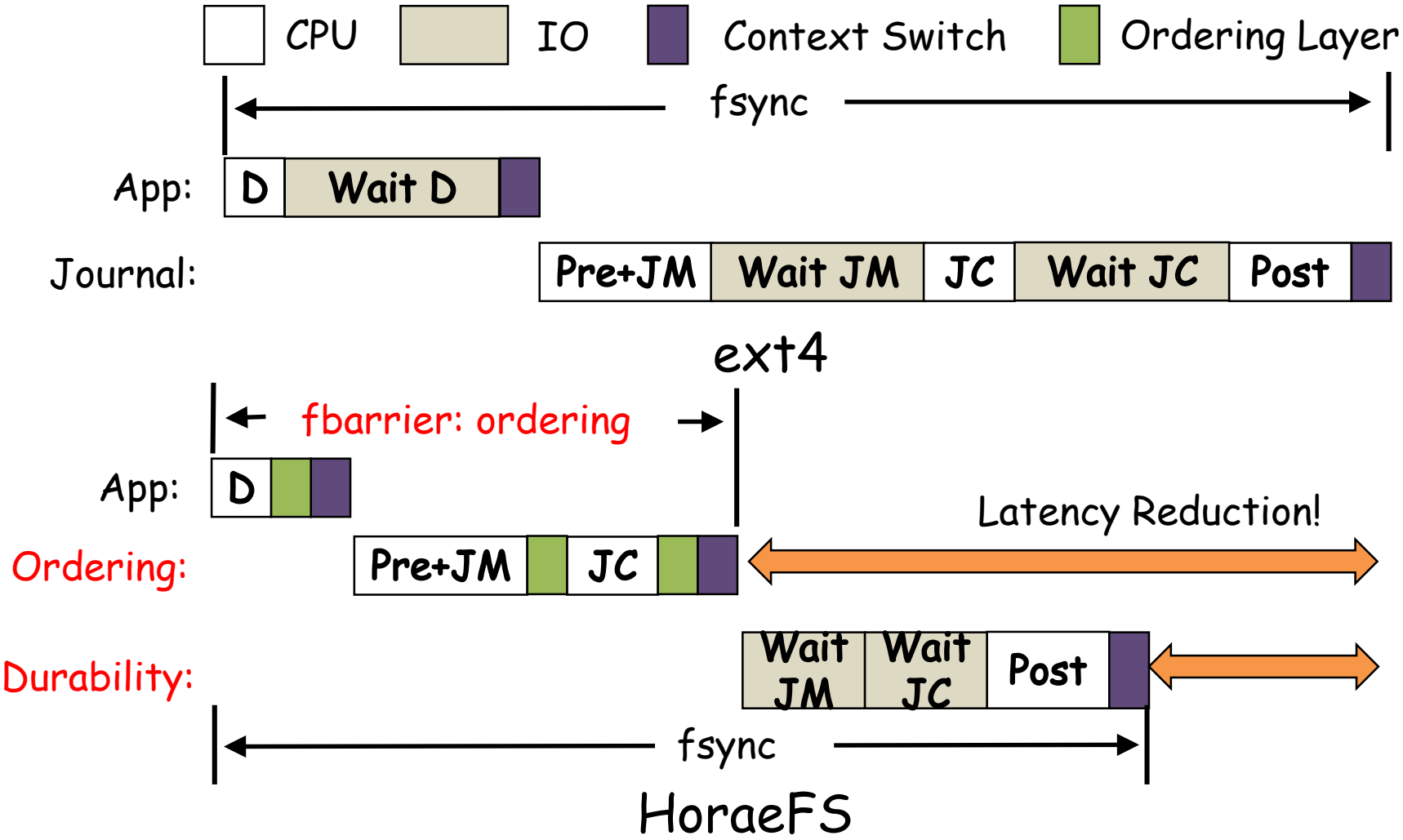
Parallelizing In-Place Updates

- Classic IO stacks serialize in-place updates.
- Horae parallelizes in-place updates through write redirection using the plba field of ordering metadata.



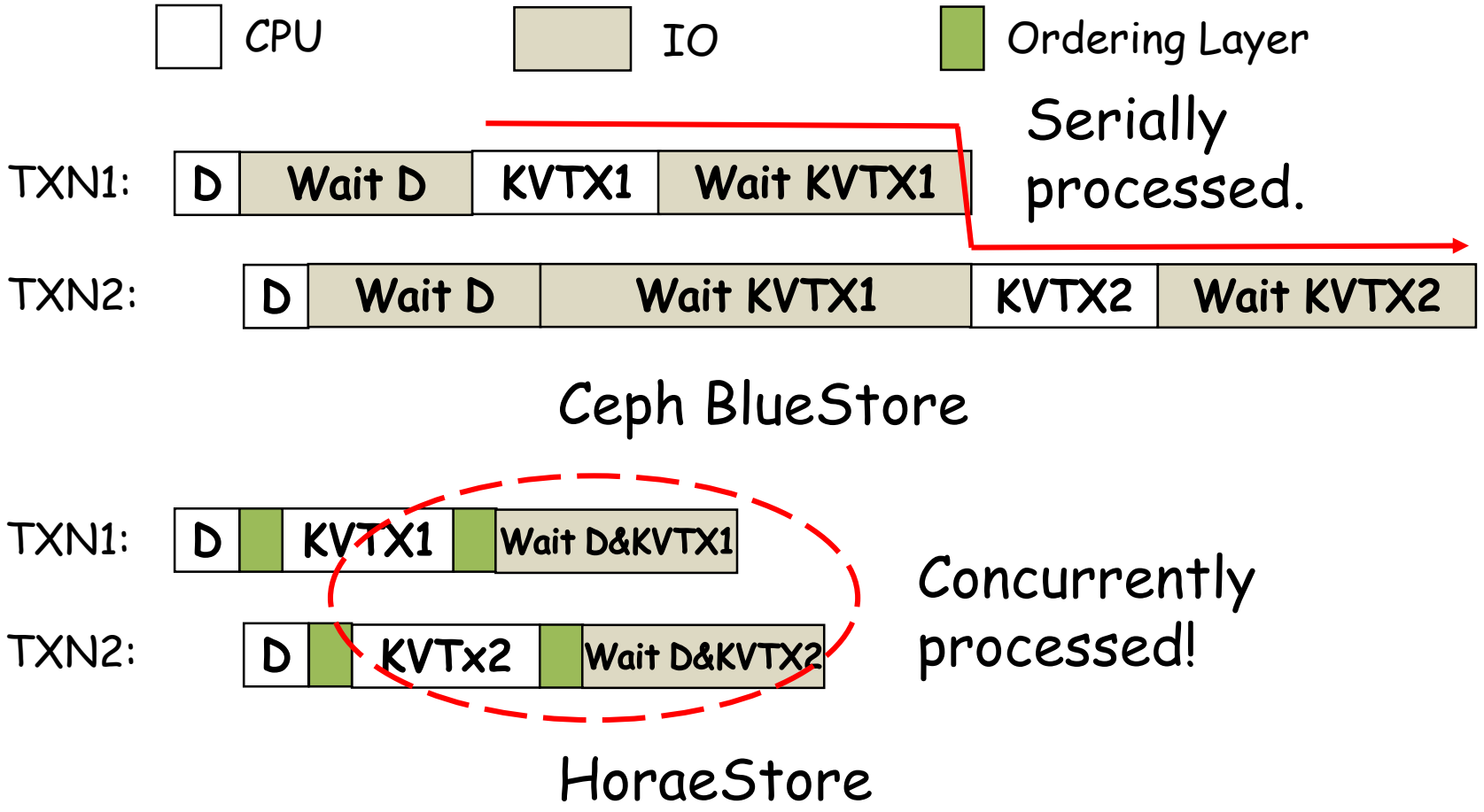
Use Case 1: HoraeFS

separate ordering logic from durability logic



Use Case 2: HoraeStore

parallelize dependent transactions



3. Evaluation

Test layer

Application
File System
Block Device
Storage

Benchmarks

- OLTP-insert, dbbench, objectbench
- FIO allocating write
- FIO random overwrite

Hardware

- Lower-end SATA SSD A: Samsung 860 Pro
- Medium-end NVMe SSD B: Intel 750 (consumer-grade)
- High-end NVMe SSD C: Intel DC P3700 (datacenter-grade)

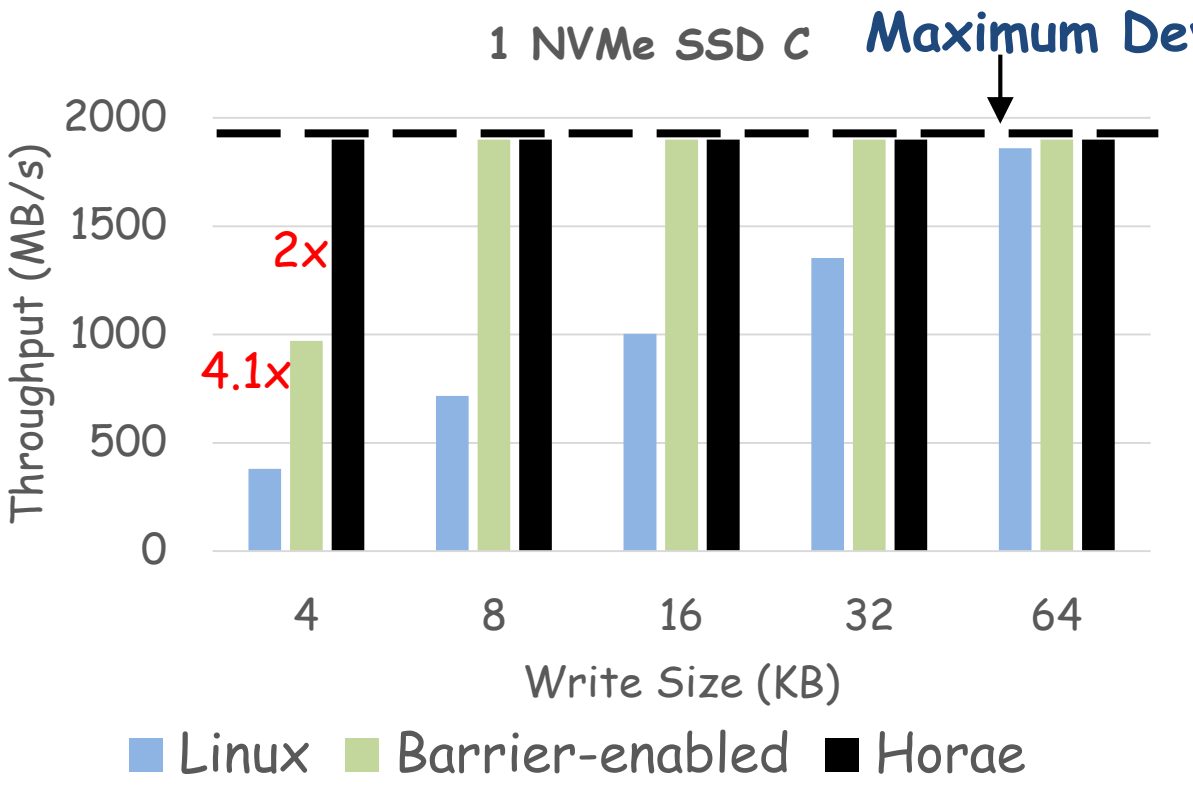
Compared systems

- Linux VS. Barrier[1] VS. Horae

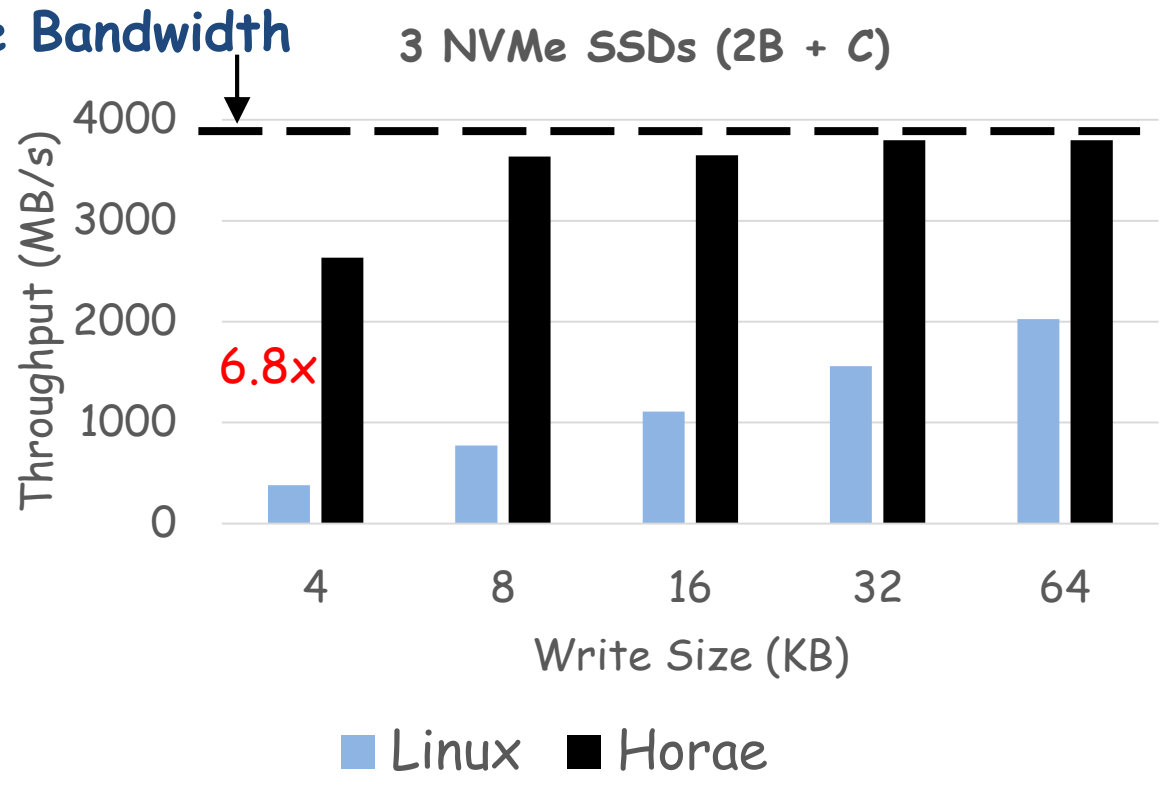
[1] Barrier-Enabled IO stack for Flash Storage, FAST'18

Block Device Performance

- **Single Device**



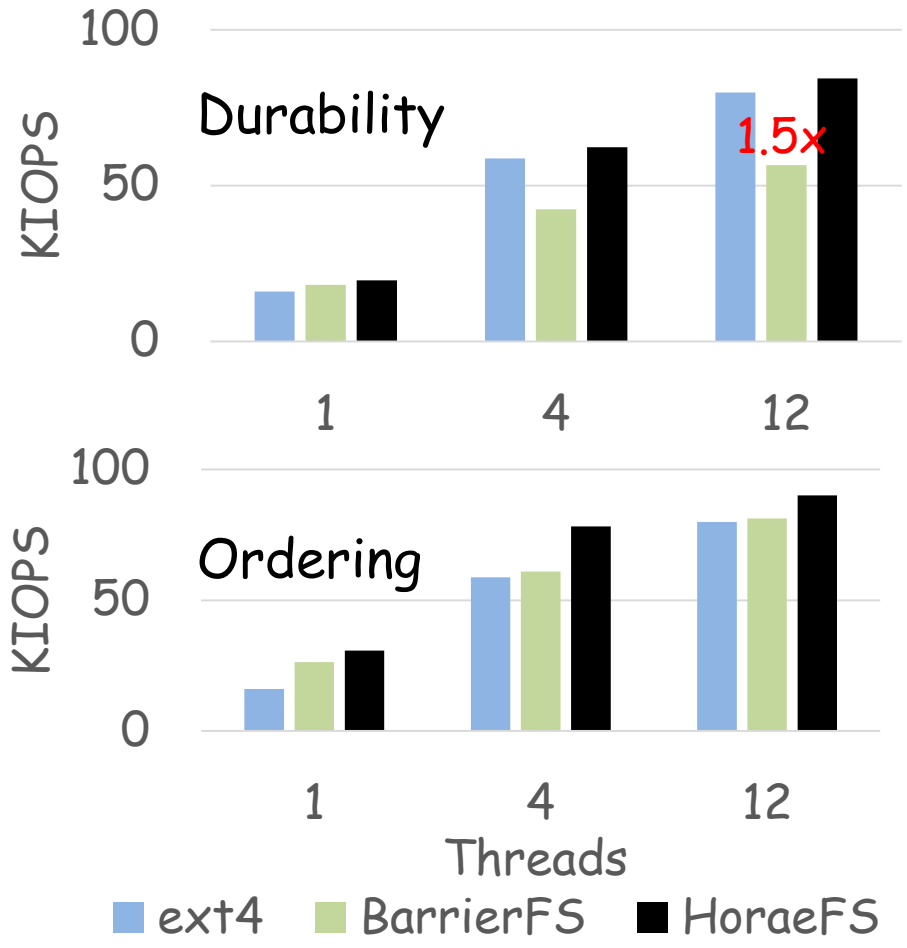
- **Multiple Devices**



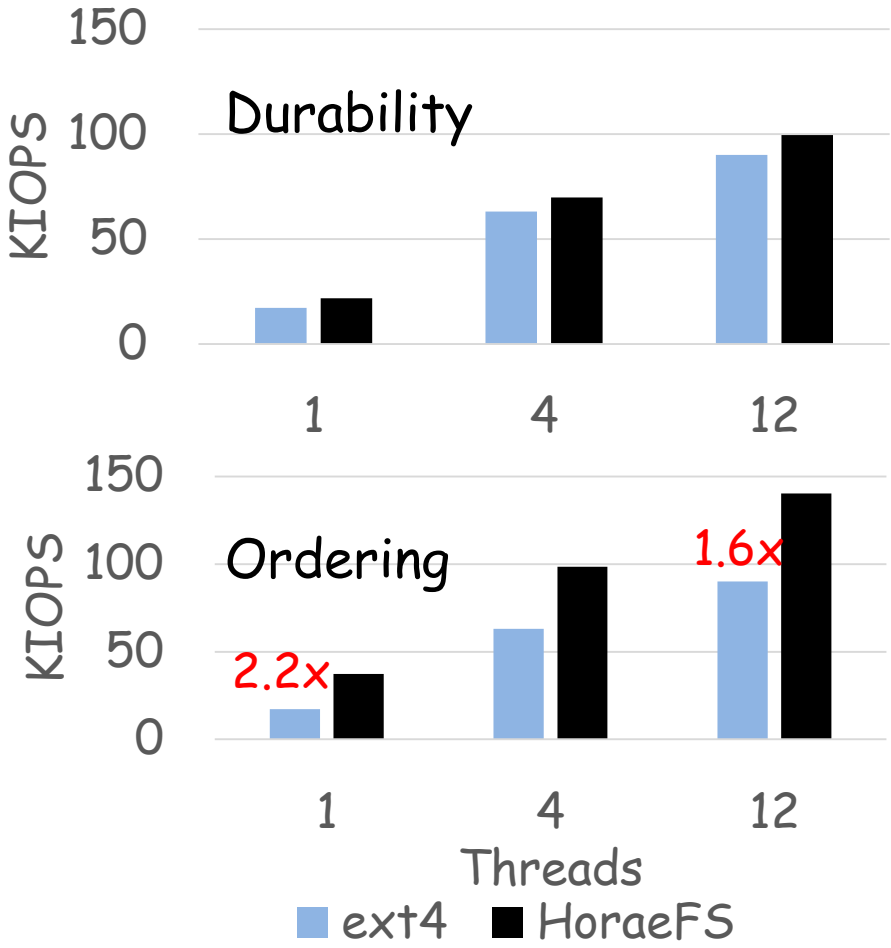
Separating the ordering control path is efficient.

File System Performance

- 1 NVMe SSD (B)



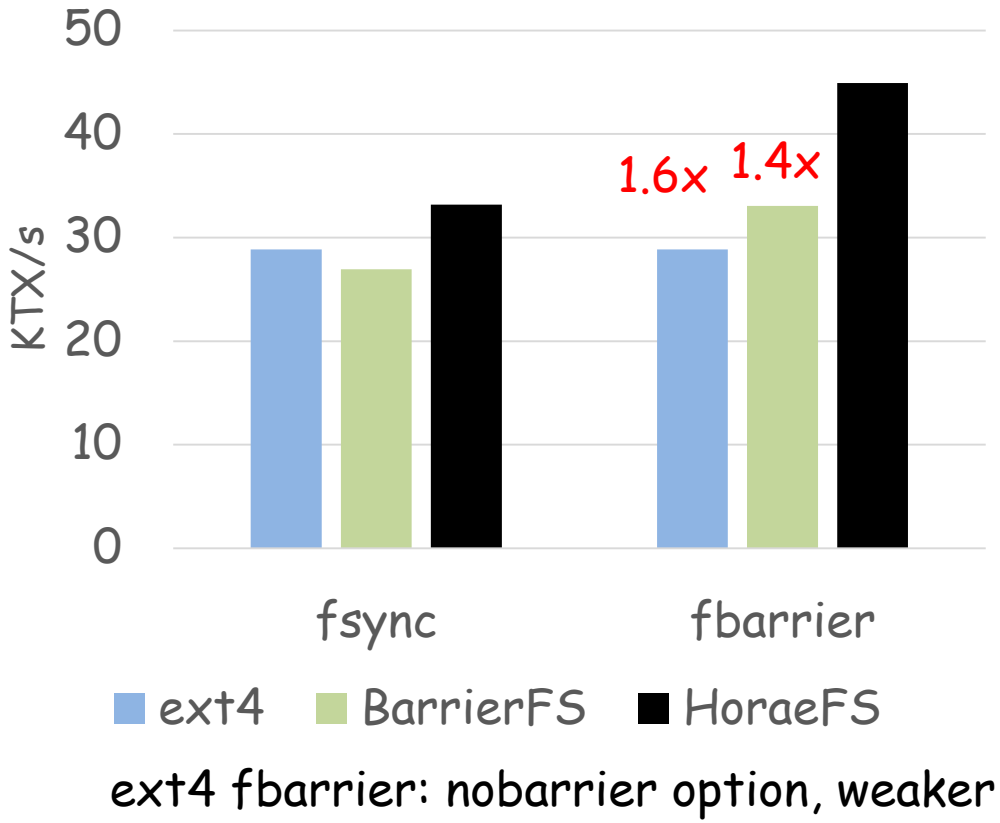
- 2 NVMe SSDs (B+C)



Parallelizing the data and journal processing is efficient.

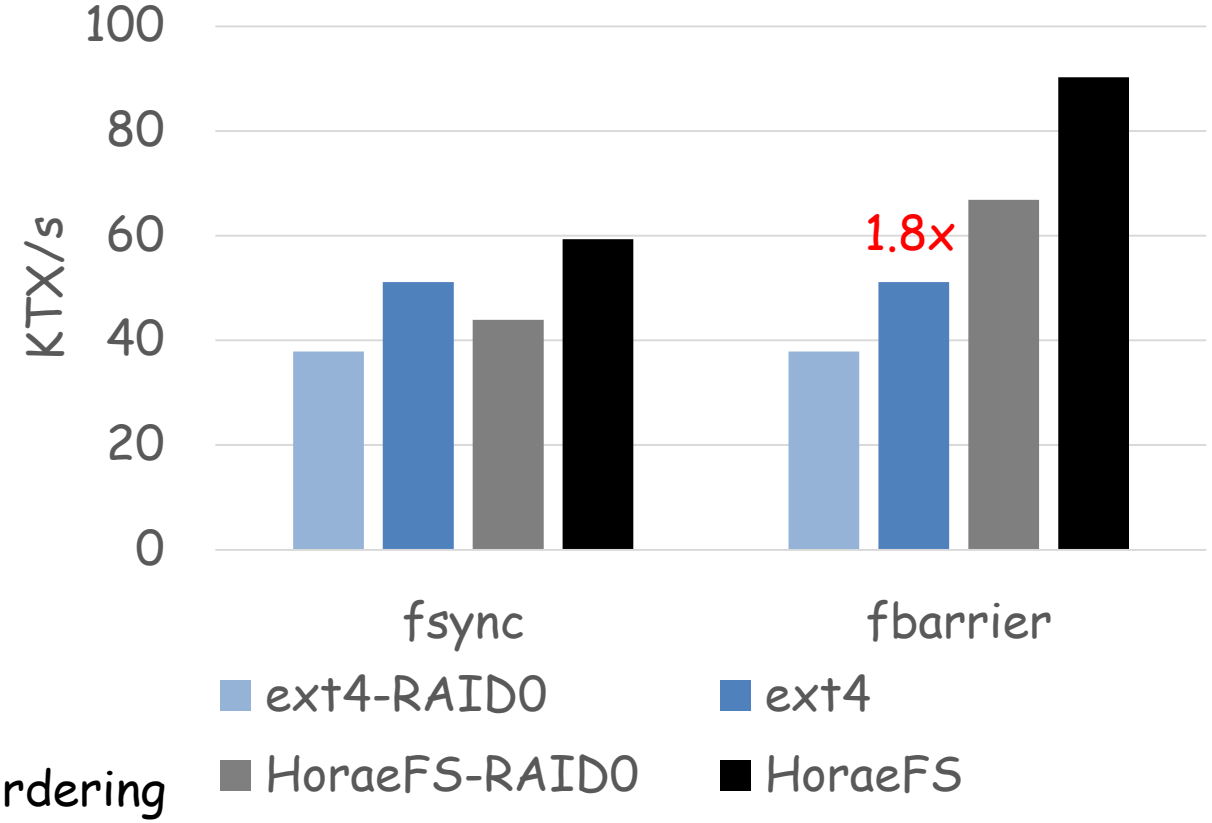
Application Performance-MySQL

- **Sole**



ext4 fbarrier: nobarrier option, weaker ordering

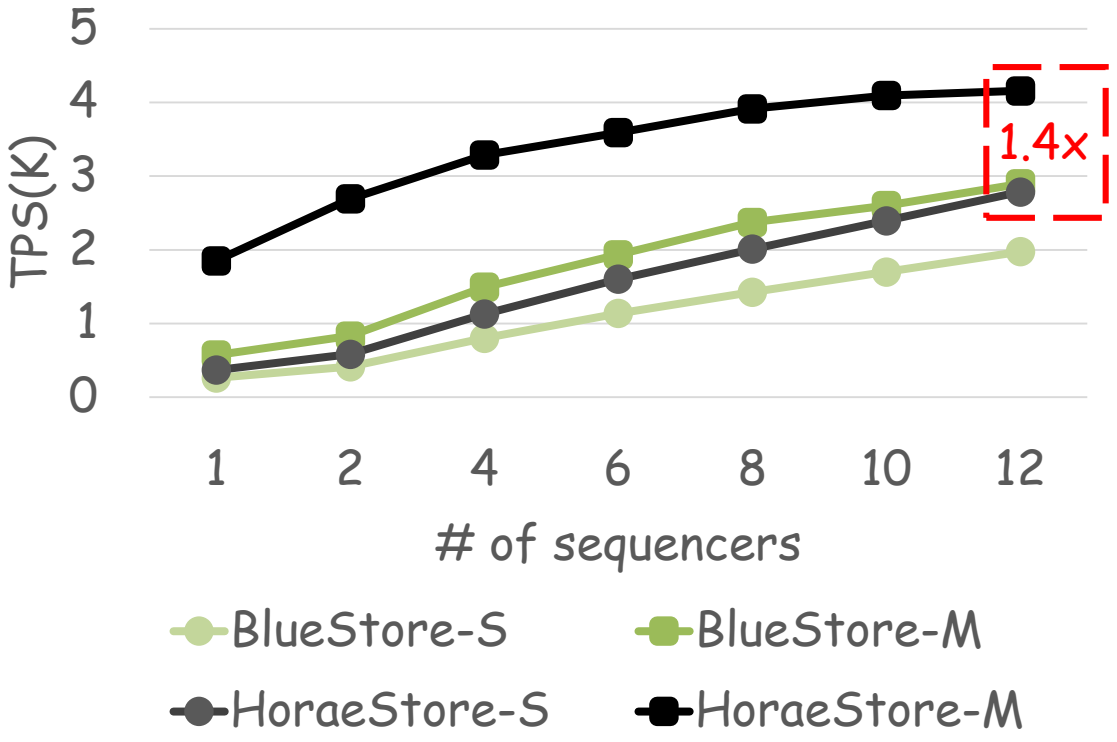
- **Separate redo log**



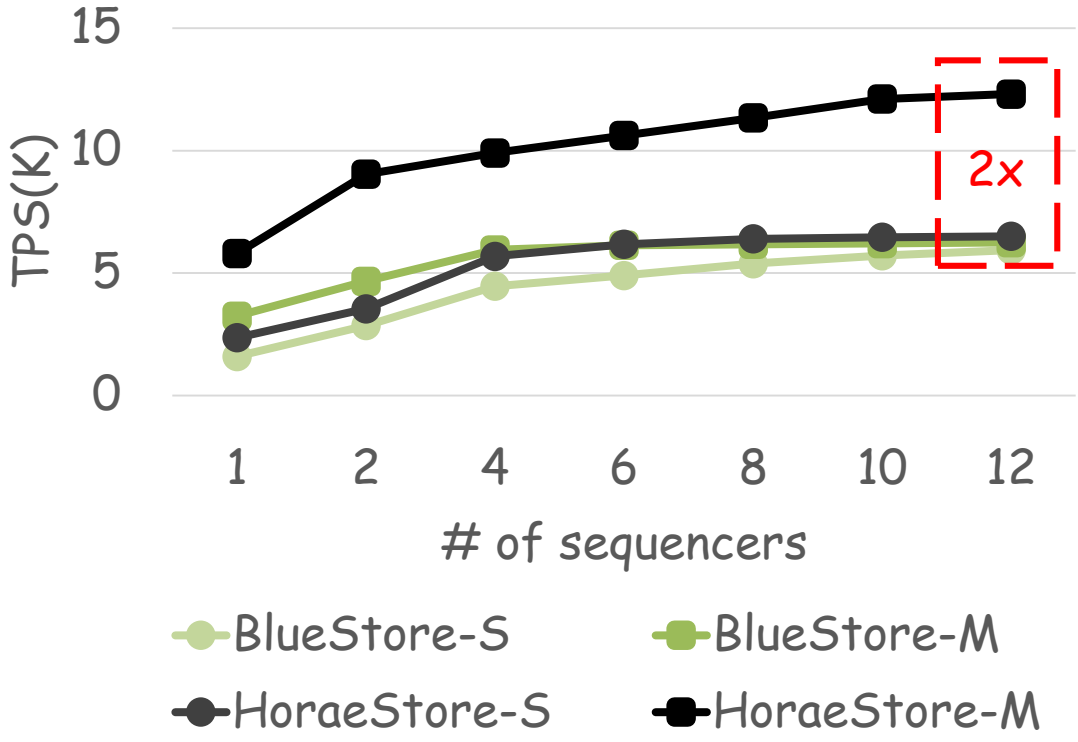
- Horae boosts MySQL performance by up to 1.8x
- HoraeFS-RAID0 < HoraeFS

Application Performance-HoraeStore

- **apply_transaction**



- **queue_transaction**



Sole: SSD A; Multiple: SSD A+B+C

HoraeStore outperforms BlueStore by up to 2x.

Conclusion

- The write dependency overhead becomes more severe with the scaling of storage concurrency.
- We re-architect IO stack with HORAE, providing a dedicated control path to reduce the write dependency overhead.
- Horae boosts file system level and application level performance by up to 2.2x and 2x.

Thank You!

Write Dependency Disentanglement with HORAE

Xiaojian Liao, Youyou Lu, Erci Xu, Jiwu Shu

liao-xj17@mails.tsinghua.edu.cn