



DESIGN, AUTOMATION & TEST IN EUROPE

17 – 19 April 2023 · Antwerp, Belgium

The European Event for Electronic
System Design & Test



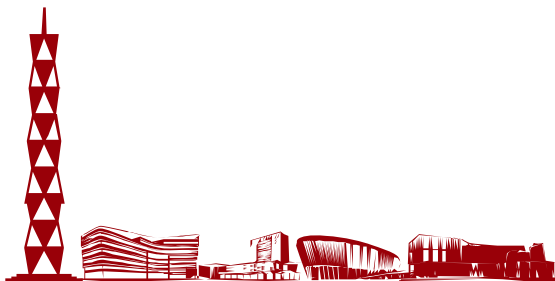
上海科技大学
ShanghaiTech University

Atomic but Lazy Updating with Memory-mapped Files for Persistent Memory

Qisheng Jiang, Lei Jia, and Chundong Wang

Toast Lab

ShanghaiTech University



立志成才 报国强民

Persistent Memory (PMEM)

- PMEM

- NVDIMM

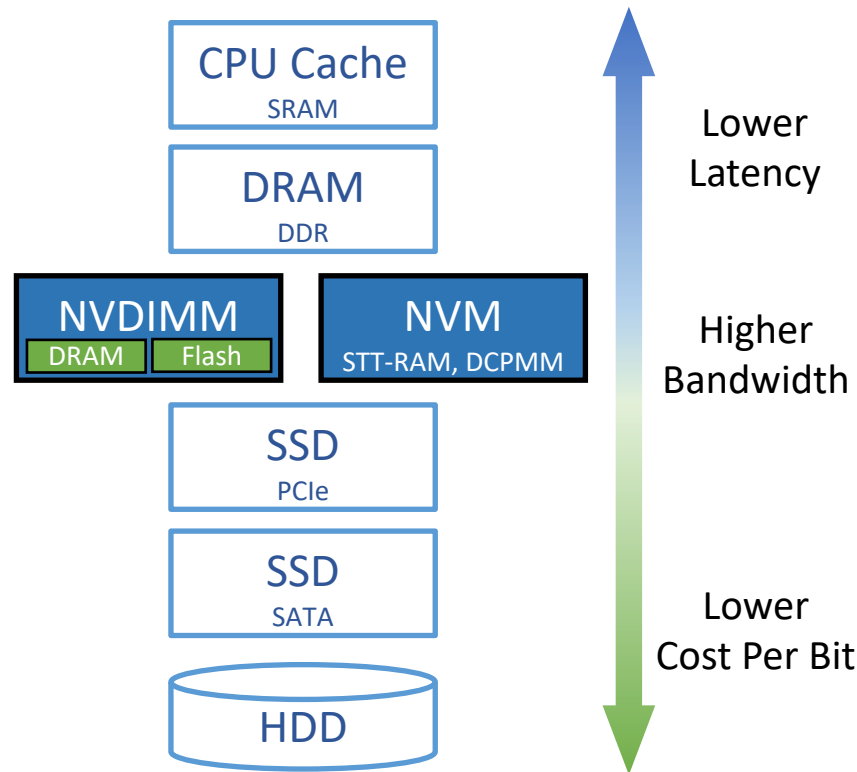
- DRAM backed by Flash Memory

- NVM Technologies

- E.g., STT-RAM and DCPMM

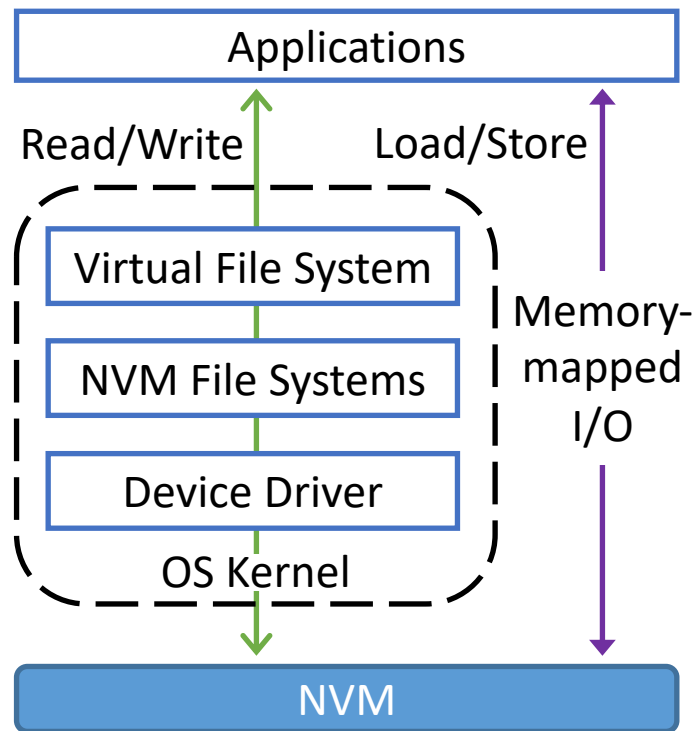
- Features

- Byte-addressability
 - Persistency
 - Fast Access Speed



NVM File System

- Direct Access (DAX)
 - Bypass DRAM page cache
 - E.g., NOVA and Ext4-DAX
- Software Overhead for Read/Write
 - Non-trivial
- Memory-mapped I/O
 - Access files directly using *memcpy*
 - Reduce user/kernel mode switch
 - Avoid multiple software layers



Failure Atomicity

- CoW

- E.g., NOVA (Replica Page)

- Logging

- E.g., Libnvmio

- Performance Overhead

- Frequent data movements
- Lack of adaptability and flexibility
- Complex data management policy

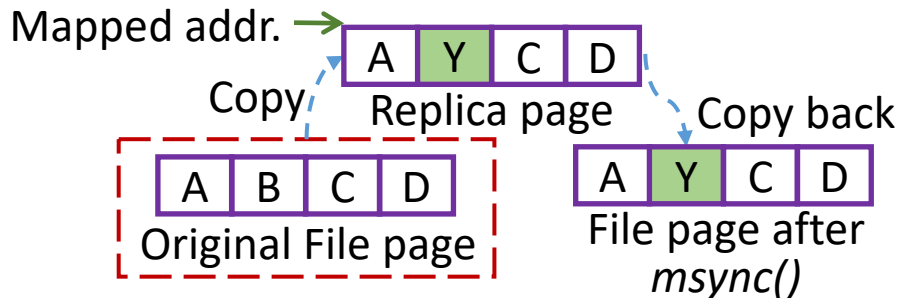


Fig. (a): NOVA' s CoW

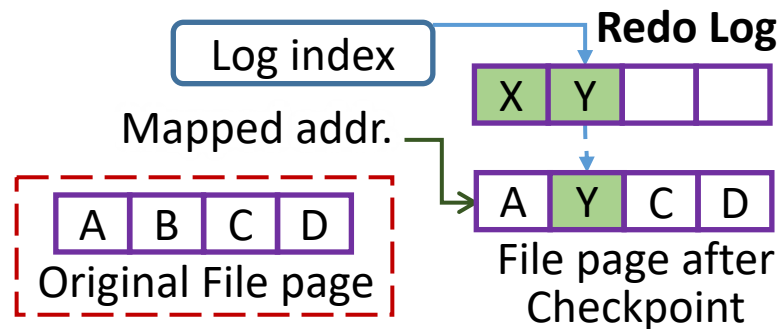


Fig. (b): Libnvmio' s Redo Logging

Failure Atomicity

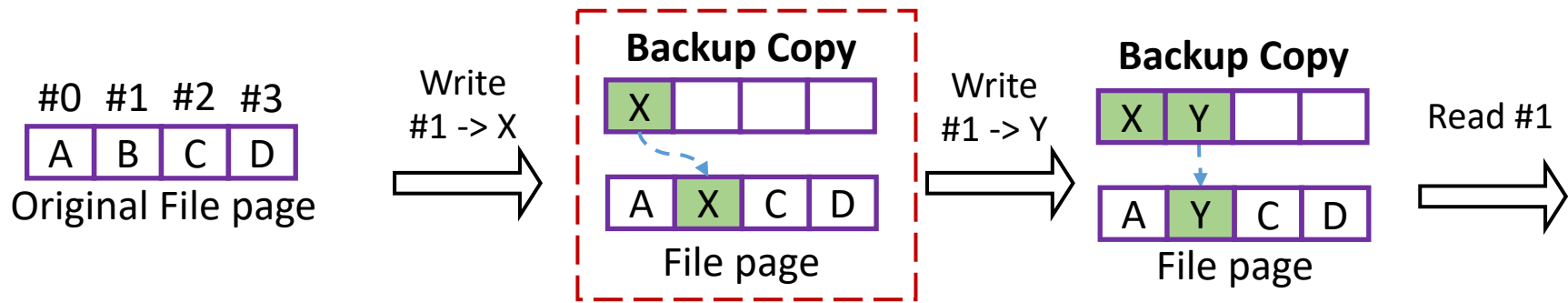


Fig. (a): Keeping data up-to-date eagerly

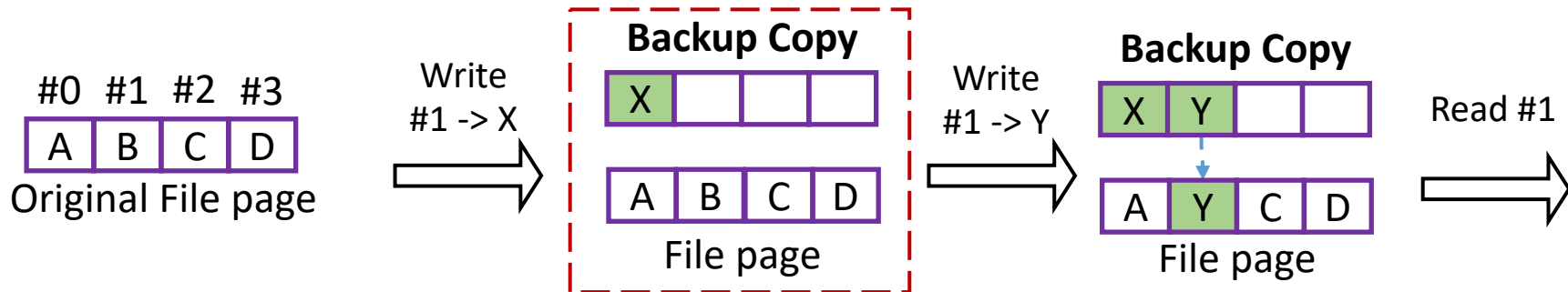


Fig. (b): Updating data in place lazily

Failure Atomicity

- Observation

- Data needs not be ready in place immediately until the application is to read it.

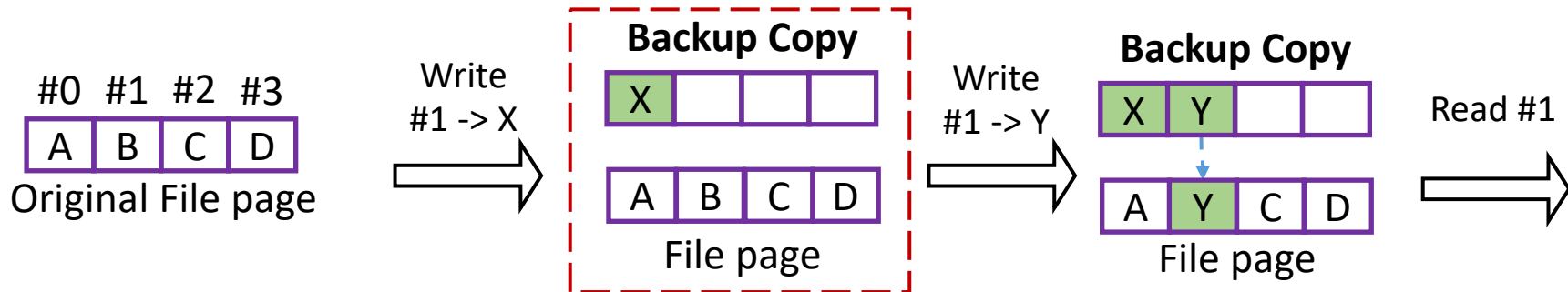
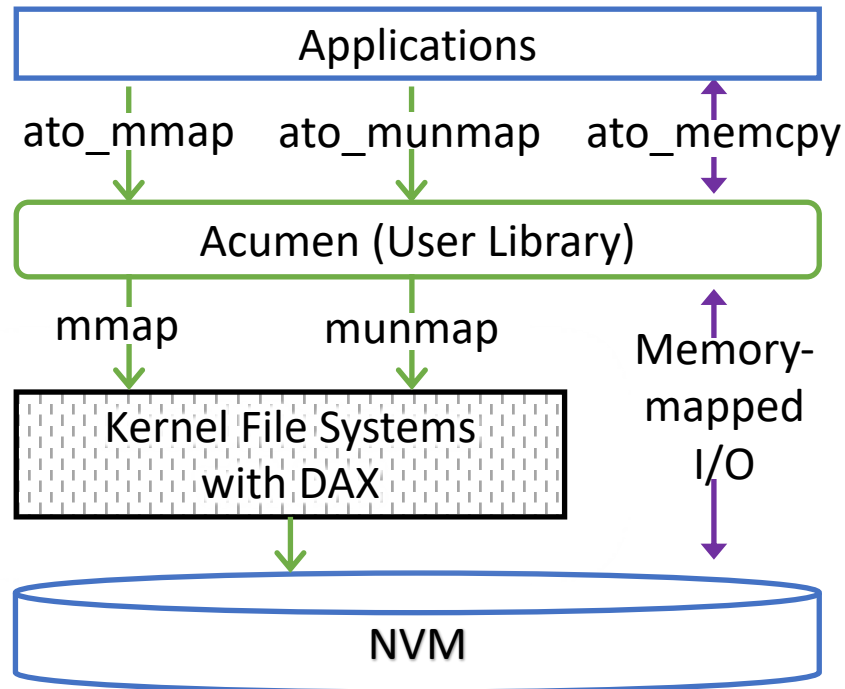


Fig. (b): Updating data in place lazily

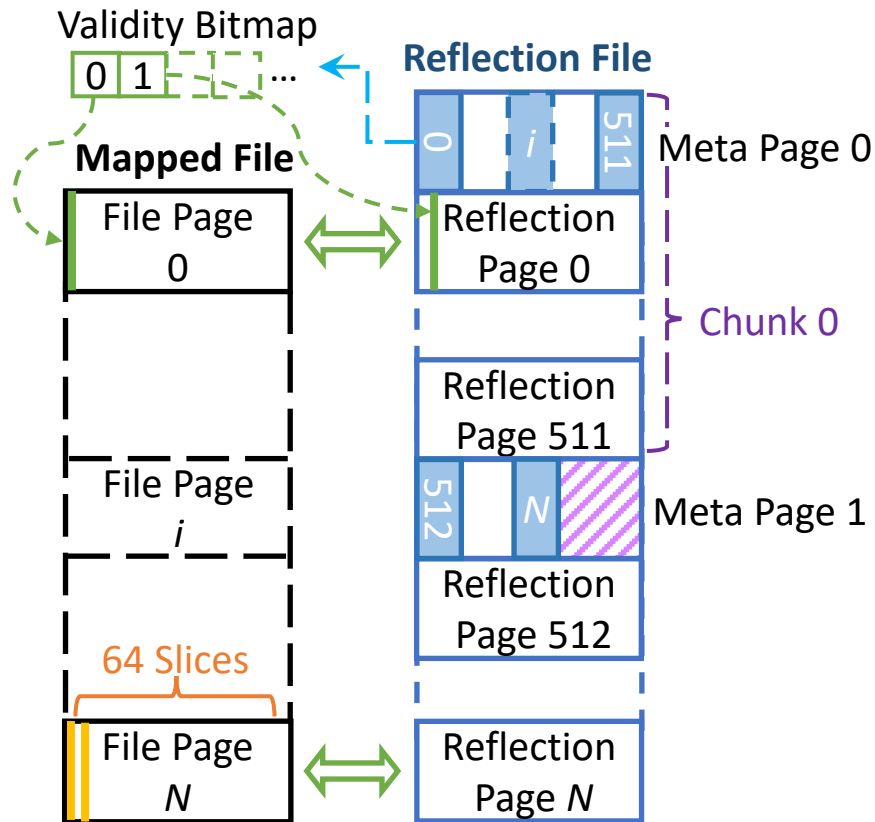
Acumen

- A Library in User Space
 - User-friendly interfaces
 - Unbound to any particular file system
- Write
 - Write data only once
 - Avoid eager in-place updating
 - Failure atomicity
- Read
 - Trigger a deployment lazily
 - Amortize cost with subsequent read requests



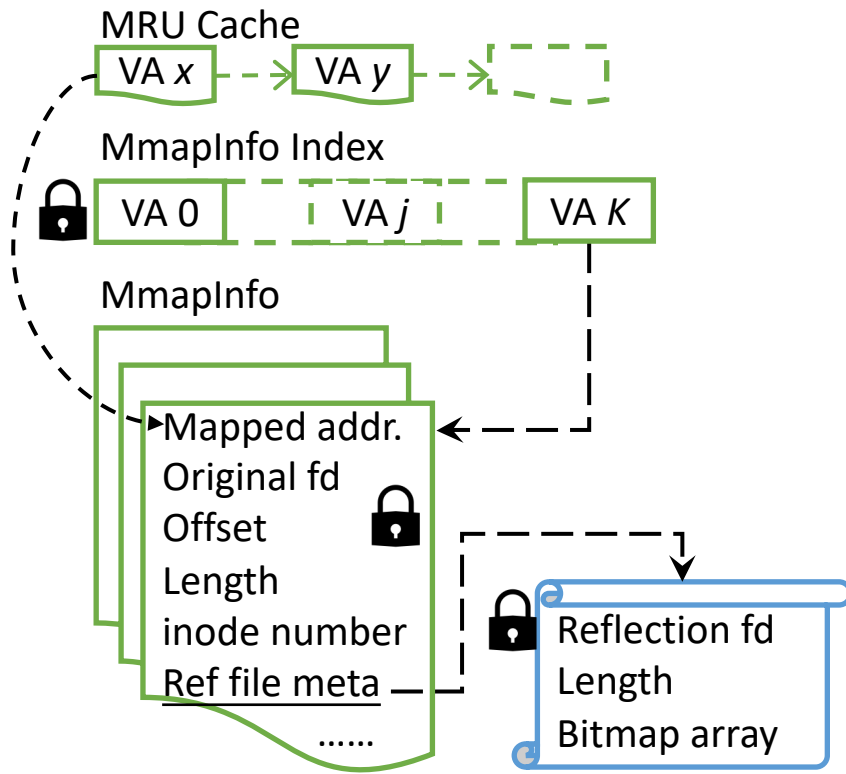
In-PMEM

- Guarantee failure atomicity
- Original File
 - Each slices 64B (cache line size)
 - 64 slices per file page (4KB)
- Reflection File
 - Meta page (4KB)
 - Reflection page (4KB)
- Validity Bitmap
 - 64 bits (8B)
 - Location of the slice's valid data
 - 0: original page / 1: reflection page



In-DRAM

- Accelerate indexing and searching
- Metadata
 - MmapInfo
 - Ref File Meta
 - Index structure
 - MRU cache
- Read-write Locks



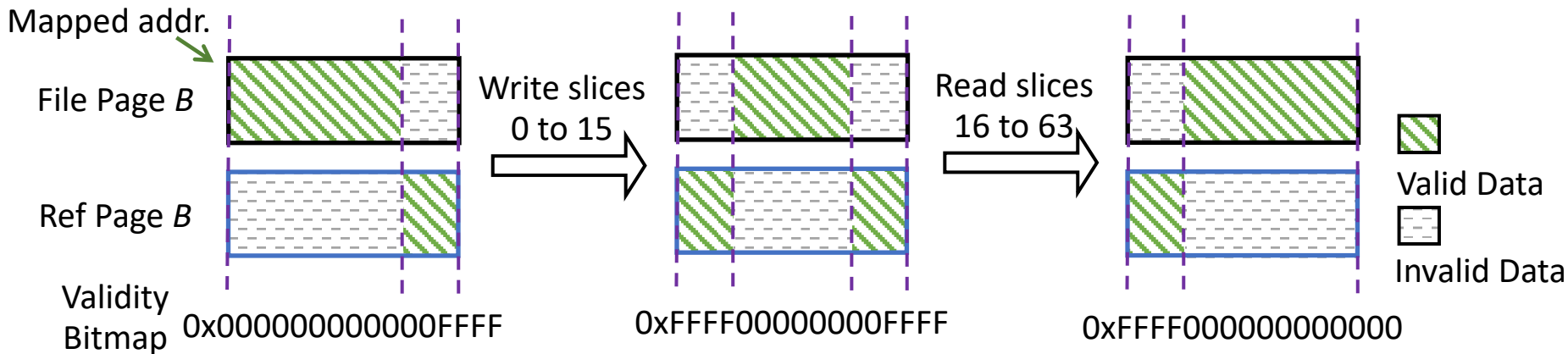
Procedures

• Write

- Write data alternately
- Update bitmap in an 8B atomic write

• Read

- Deploys valid data from the ref page to file page
- Update bitmap in an 8B atomic write
- Read from mapped address



Evaluation

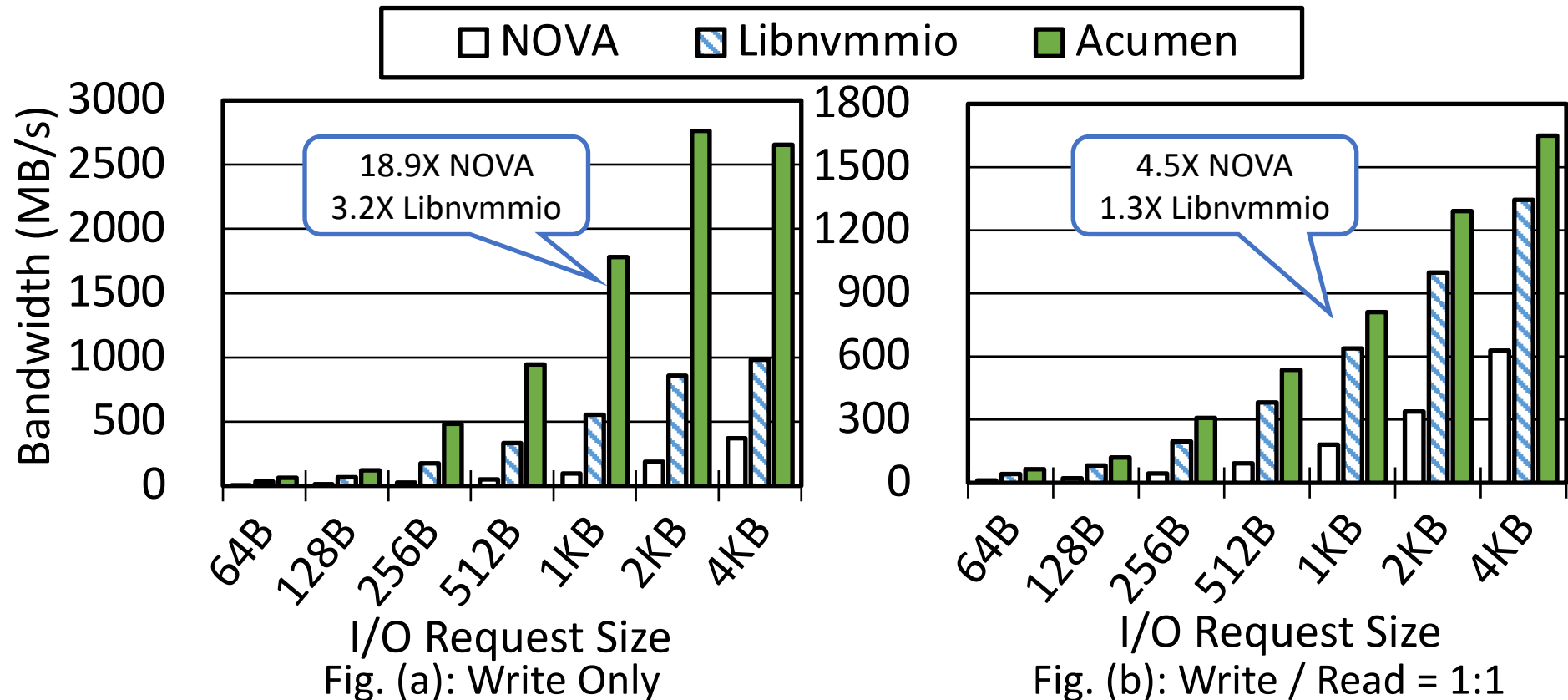
- Platform

- CPU: Intel Xeon Gold 6342
- OS: Ubuntu 21.04 with kernel 5.4.206
- Intel Optane PMEM in 1024GB with eADR
- File System: NOVA with page size of 4KB

- Benchmark

- Micro
 - FIO with mmap I/O engine
- Macro
 - SQLite 3.39.0 with the memory-mapped I/O path
 - YCSB, TPC-C

FIO with Random Read/Write



TPC-C with Multi-threads

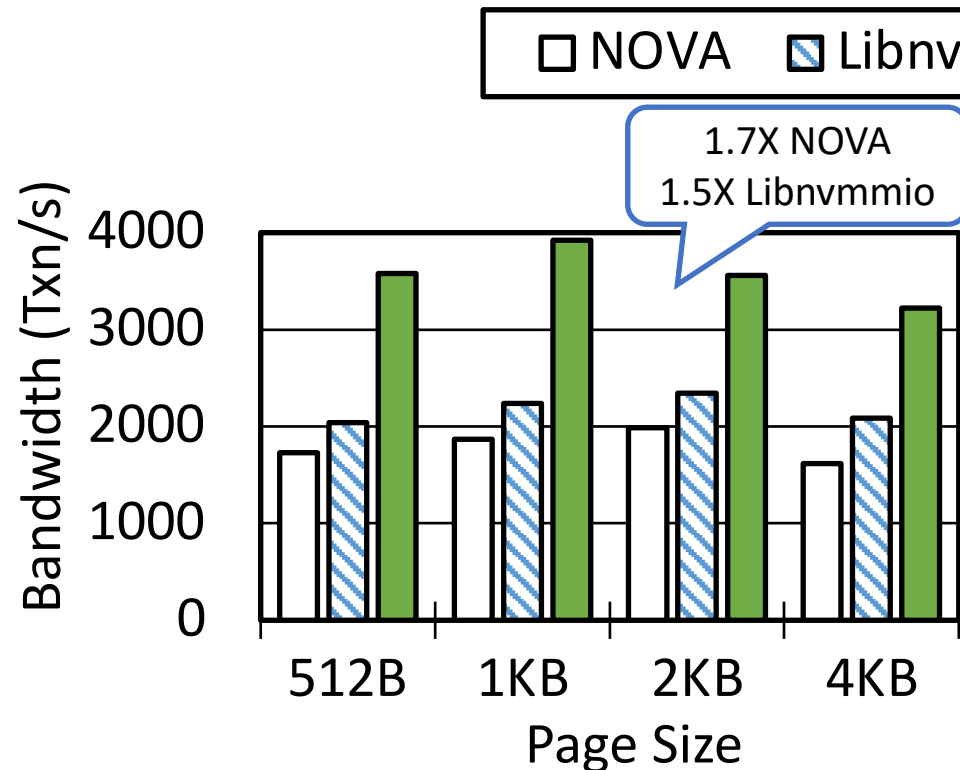


Fig. (a): 1 Thread

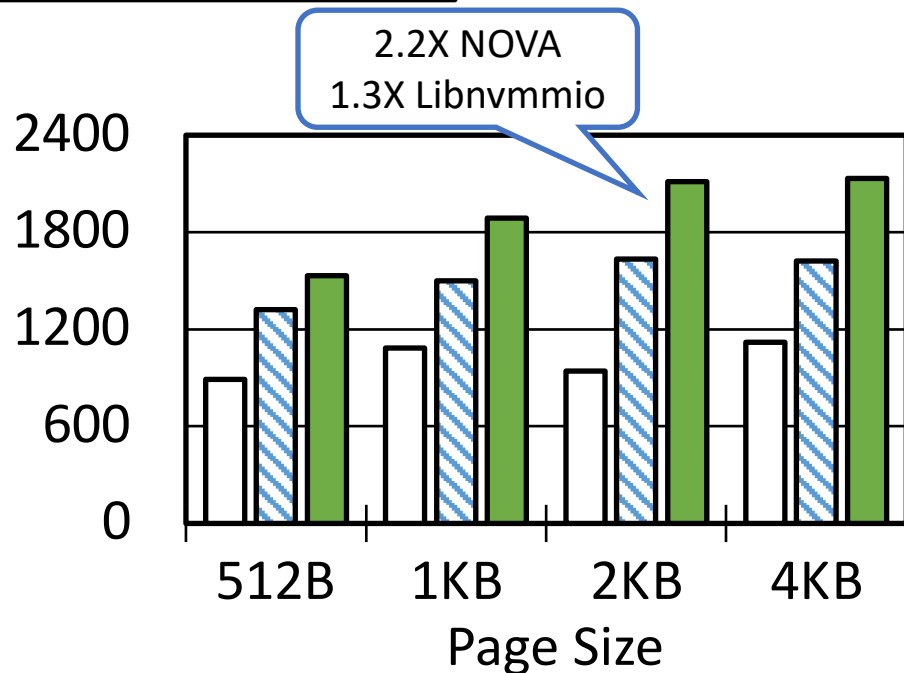


Fig. (b): 4 Threads

Conclusion

- Acumen is a simple and practical solution
 - Reflection pages
 - Simplistic data structures
 - Lightweight management policy
- Reduce software overhead
- Guarantee failure atomicity
- High throughput
- Scalable IO

Thank you!

- Lab website
 - <https://toast-lab.tech>
- Contact
 - jiangqsh@shanghaitech.edu.cn