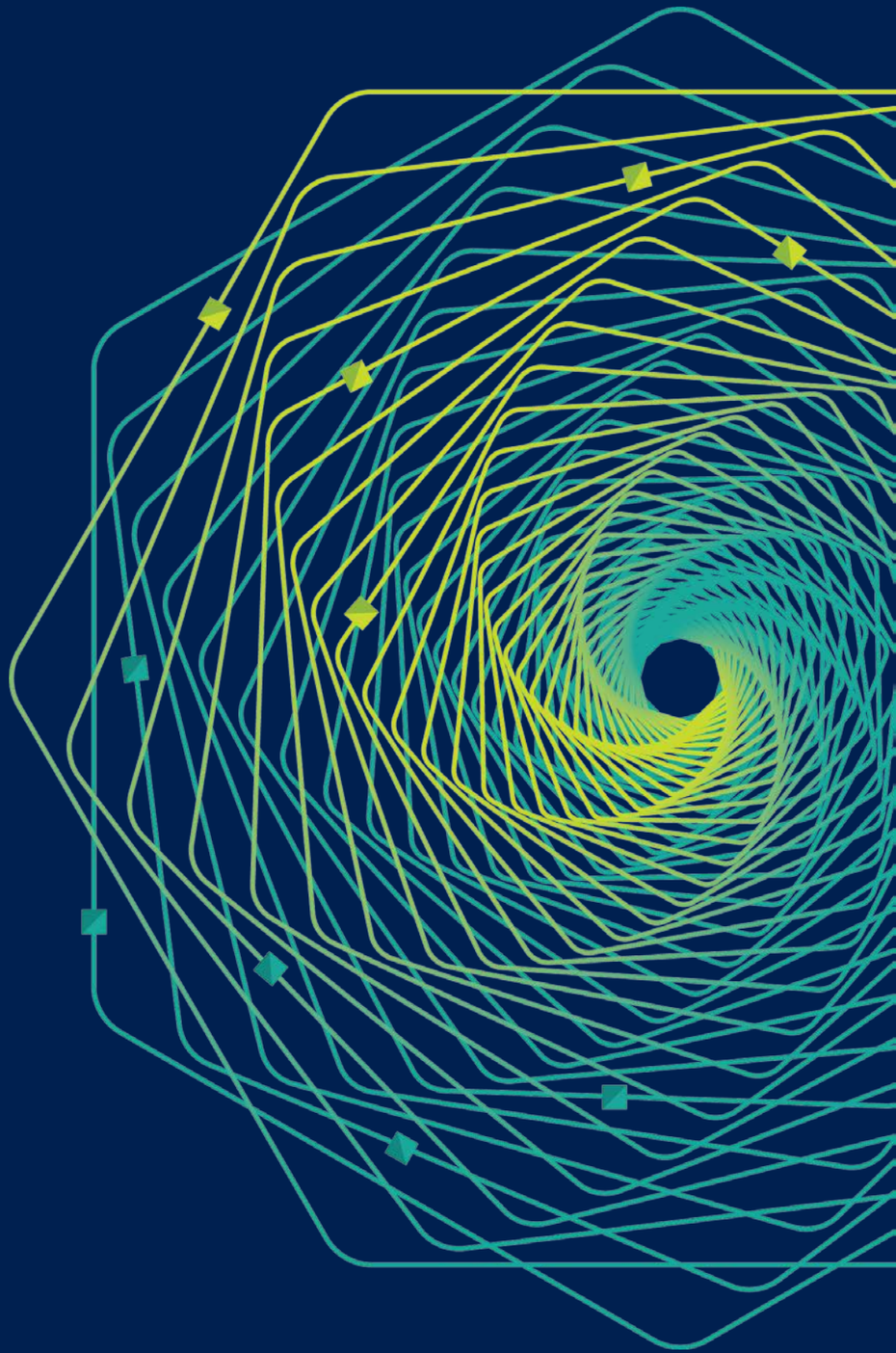
Microsoft

Research Faculty Summit 2018

Systems | Fueling future disruptions



What Should You Do With Persistent Memory?

Steven Swanson

Director, Non-Volatile Systems Lab

Computer Science and Engineering

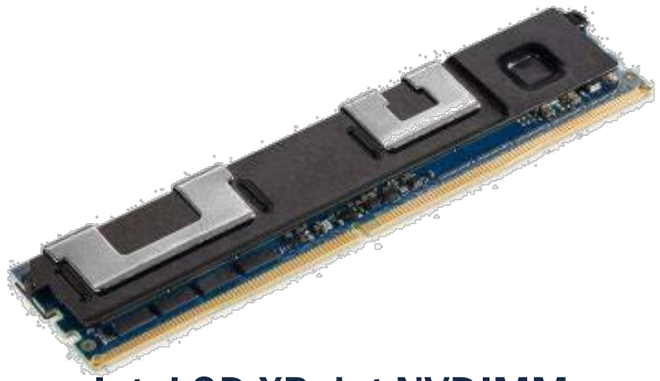
UC San Diego

NVSL

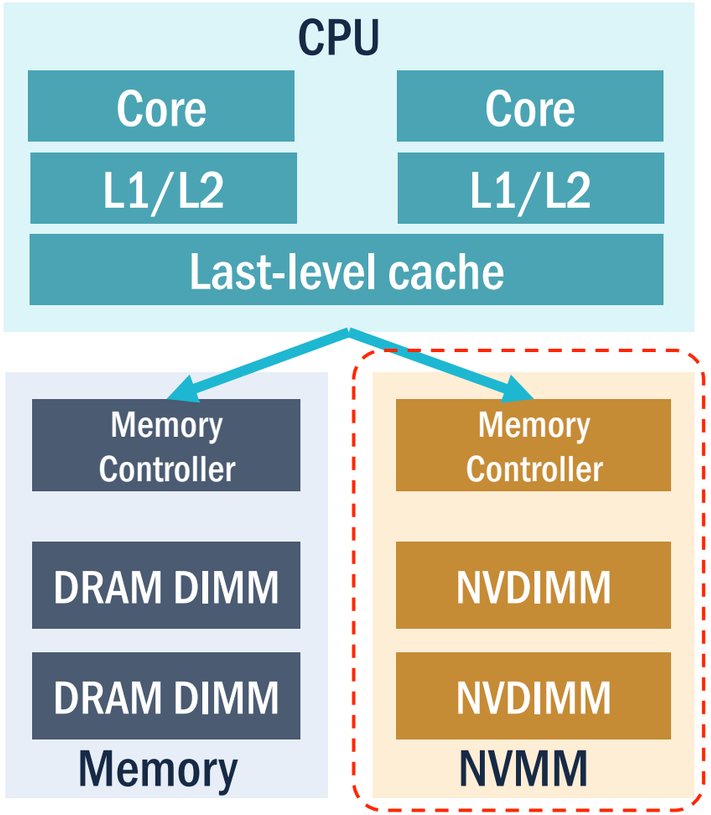
UCSDCSE Computer Science and Engineering

Non-volatile main memory (NVMM)

- Byte-addressable
- Denser than DRAM
- DRAM-comparable latency
- Higher bandwidth than SSD
- Ready for DMA / RDMA



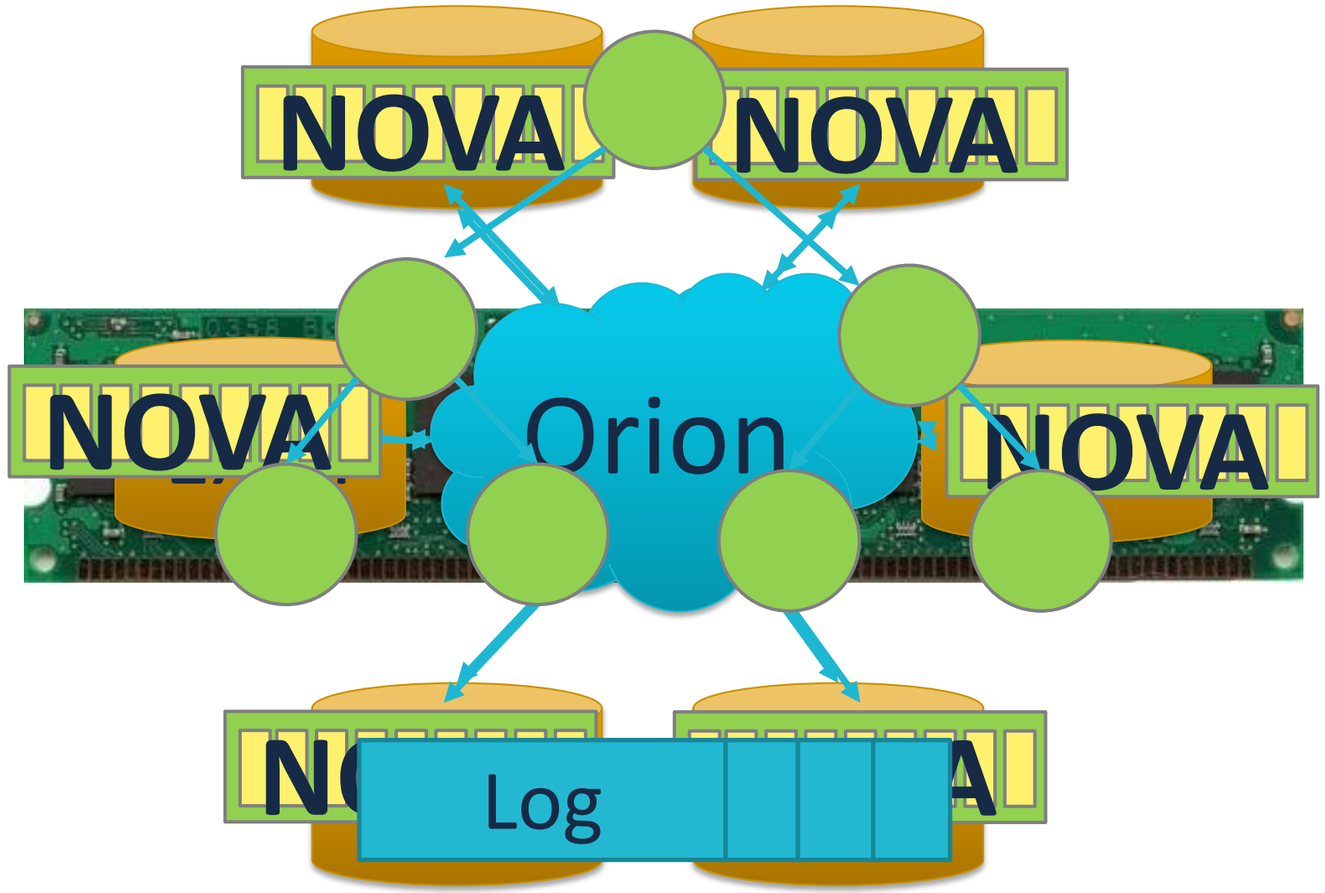
Intel 3D XPoint NVDIMM



NVSL

What Should You Do With NVMM?

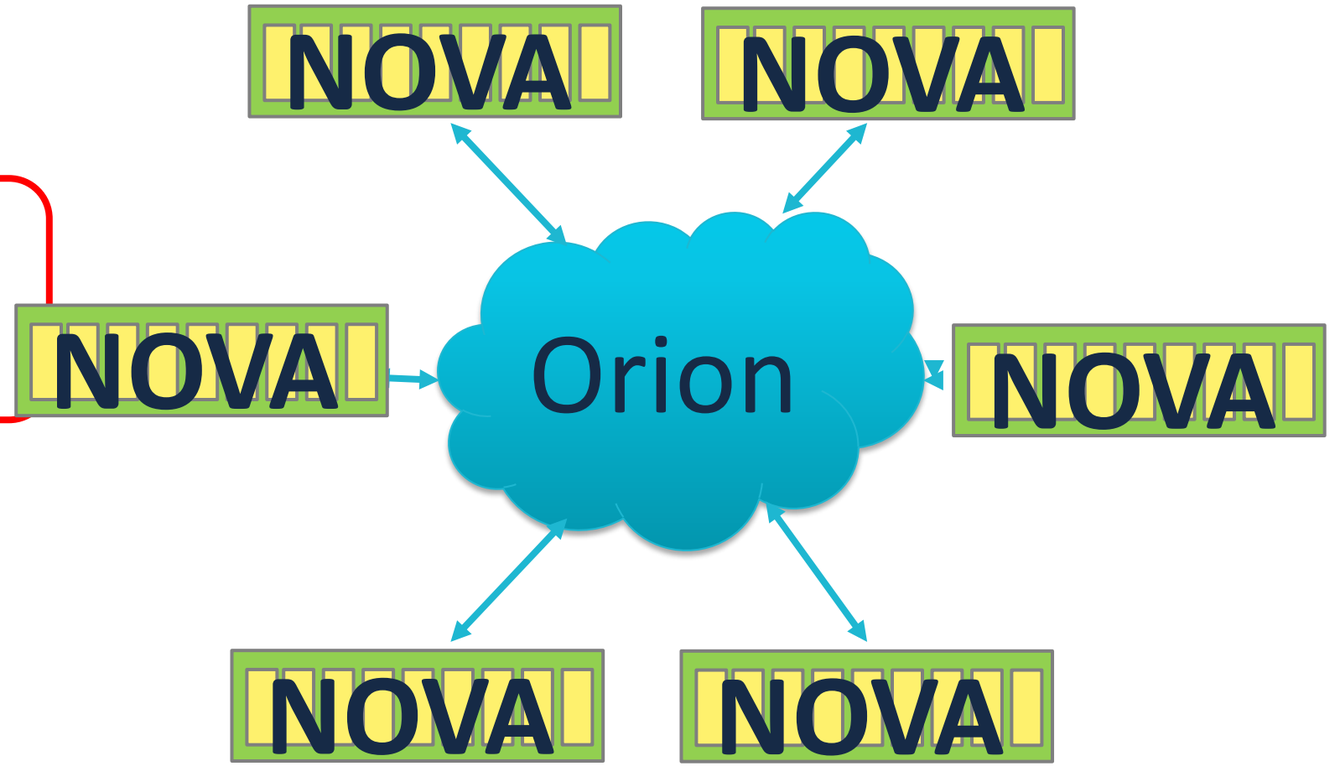
1. Use files and a conventional (distributed) file system
2. Use files and better file (distributed) system
3. Build persistent data structures
4. Use it as slow DRAM



NVSL

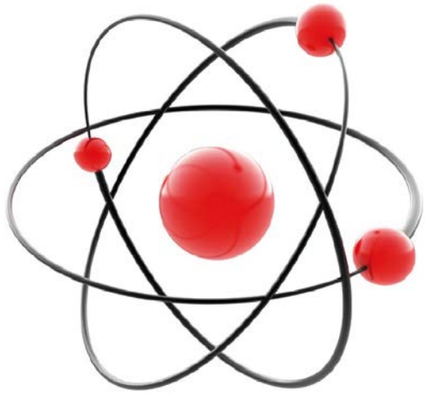
What Should You Do With NVMM?

1. Use files and a conventional (distributed) file system
2. Use files and better file (distributed) system
3. Build persistent data structures
4. Use it as slow DRAM



NVSL

File IO



Atomicity

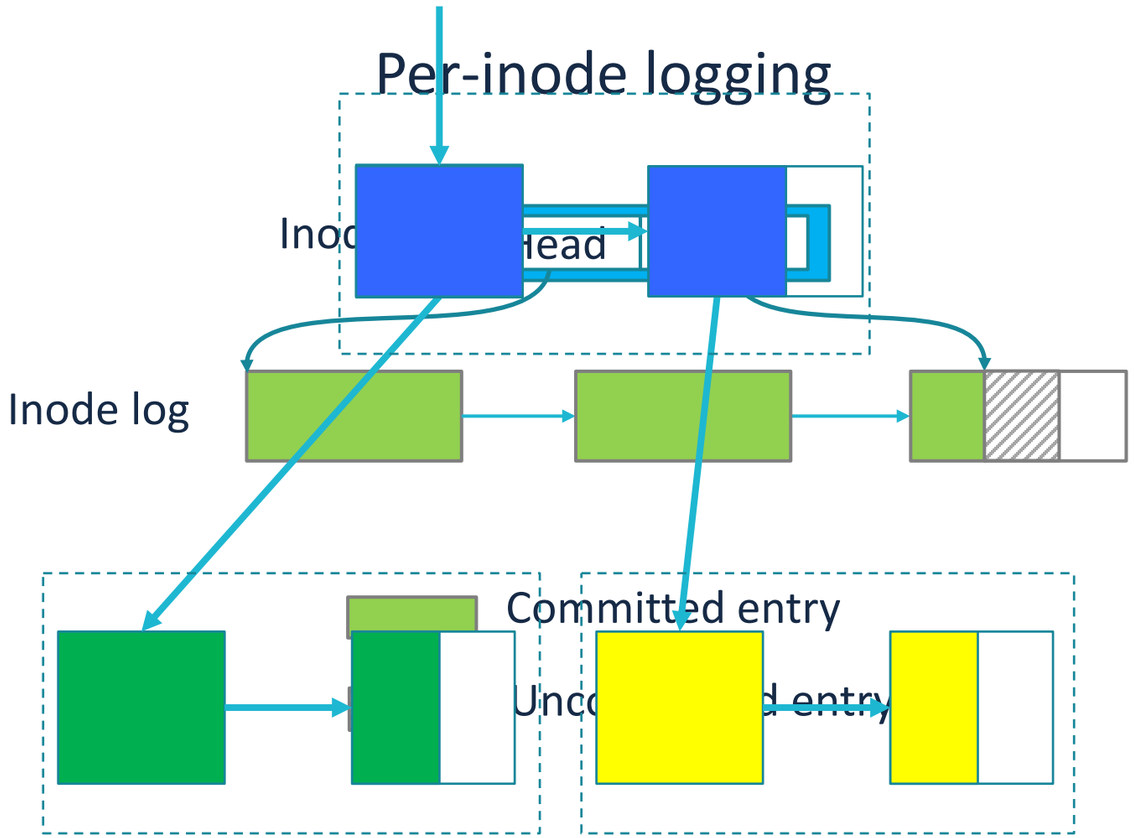
Fault Tolerance

Direct Access

Speed

NOVA: A File System for NVMM

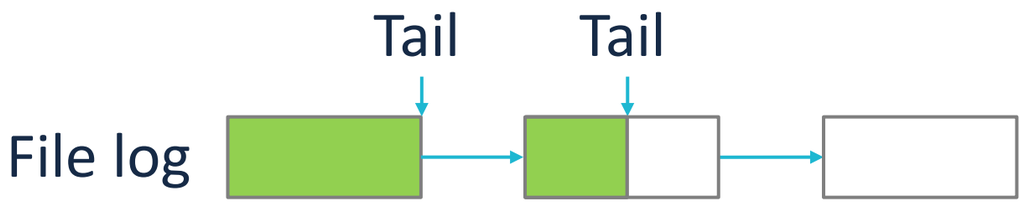
- A NOVA FS is a tree of logs
- One log per inode
  - Inode points to head and tail
  - Logs are not contiguous
- Many Logs -> high concurrency
- Strong consistency guarantees
- Log-structured + journals + copy-on-write



NVSL

Atomicity: Logging for Simple Metadata Operations

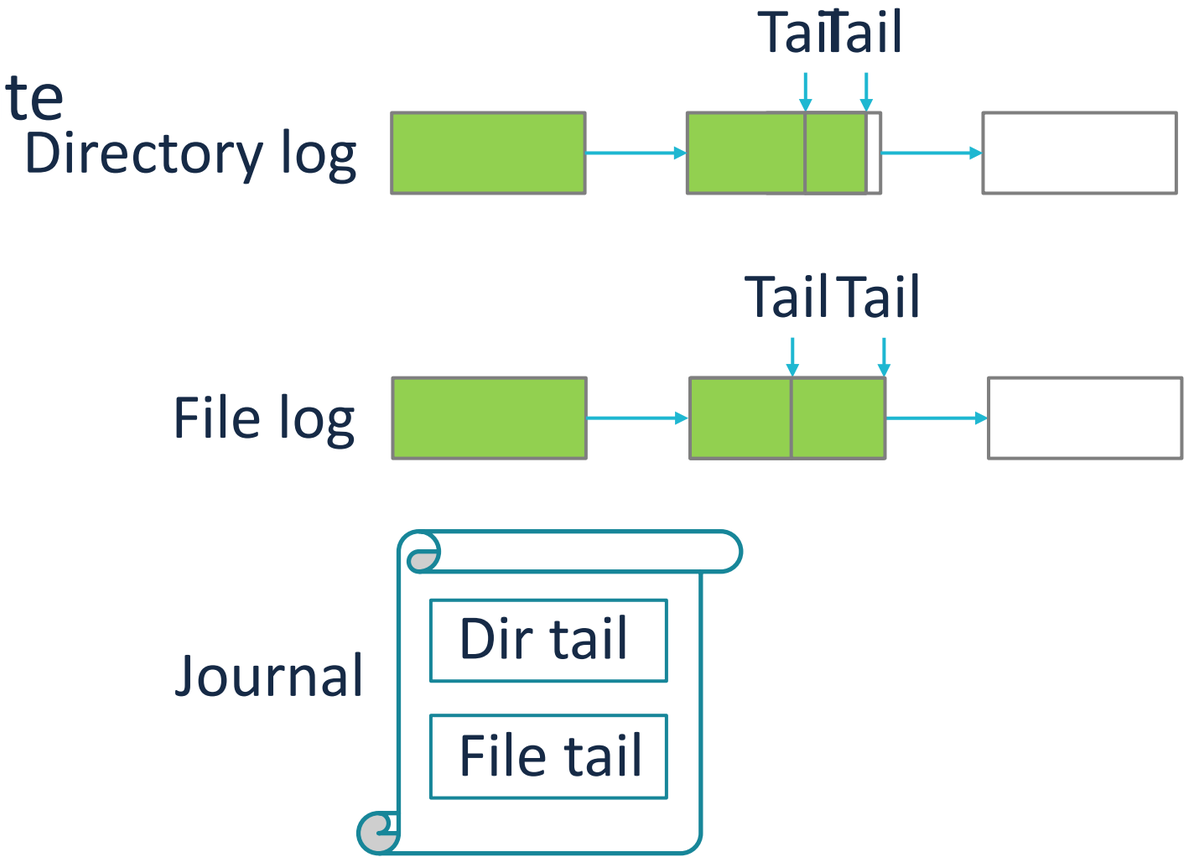
- Combines log-structuring, journaling and copy-on-write
- Log-structuring for single log update
  - Write, msync, chmod, etc
  - Lower overhead than journaling and shadow paging



NVSL

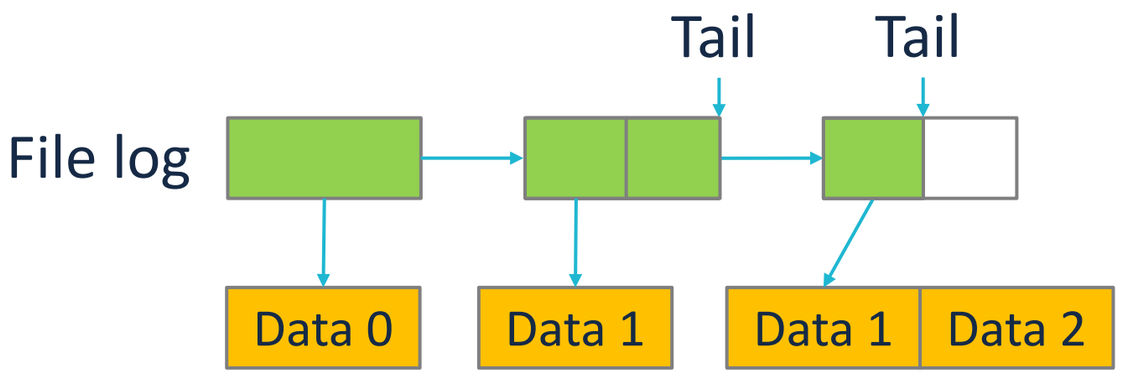
Atomicity: Lightweight Journaling for Complex Metadata Operations

- Lightweight journaling for update across logs
  - Unlink, rename, etc
  - Journal log tails instead of metadata or data



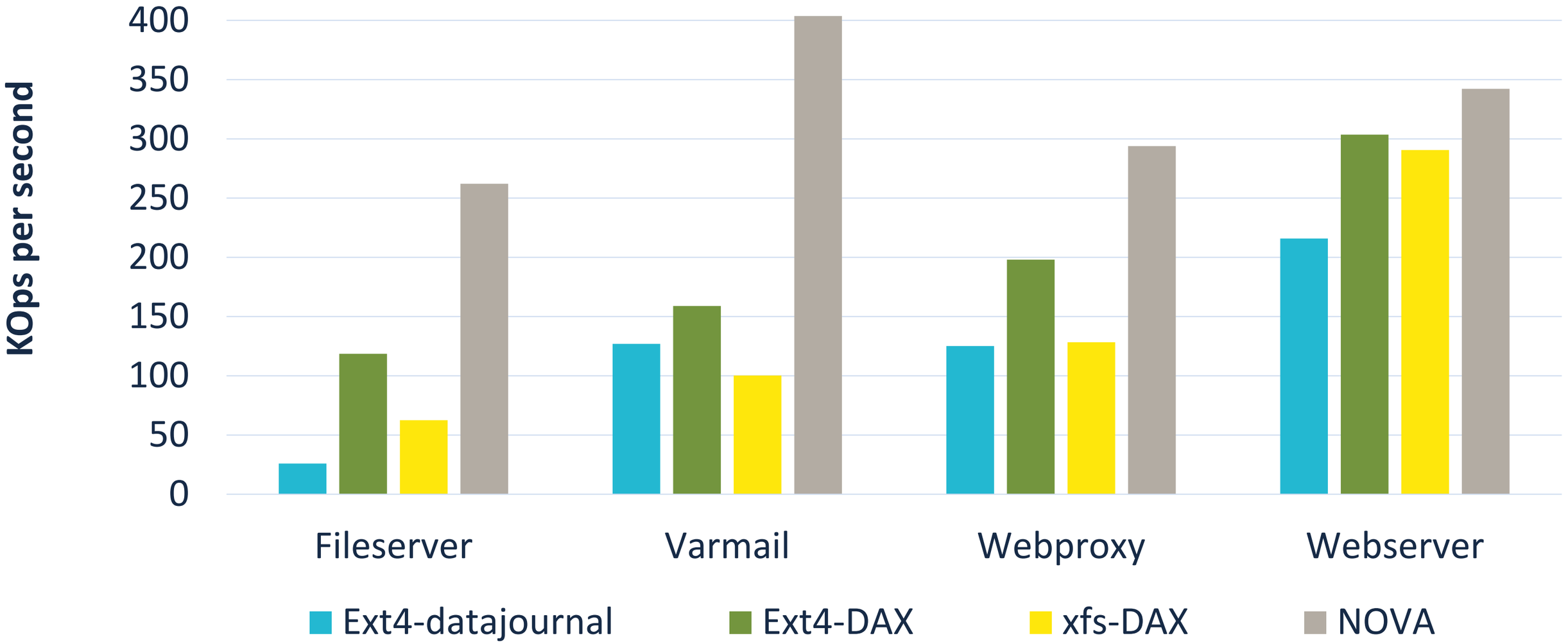
Atomicity: Copy-on-write for file data

- Copy-on-write for file data
  - Log only contains metadata
  - Log is short
  - Instant data GC



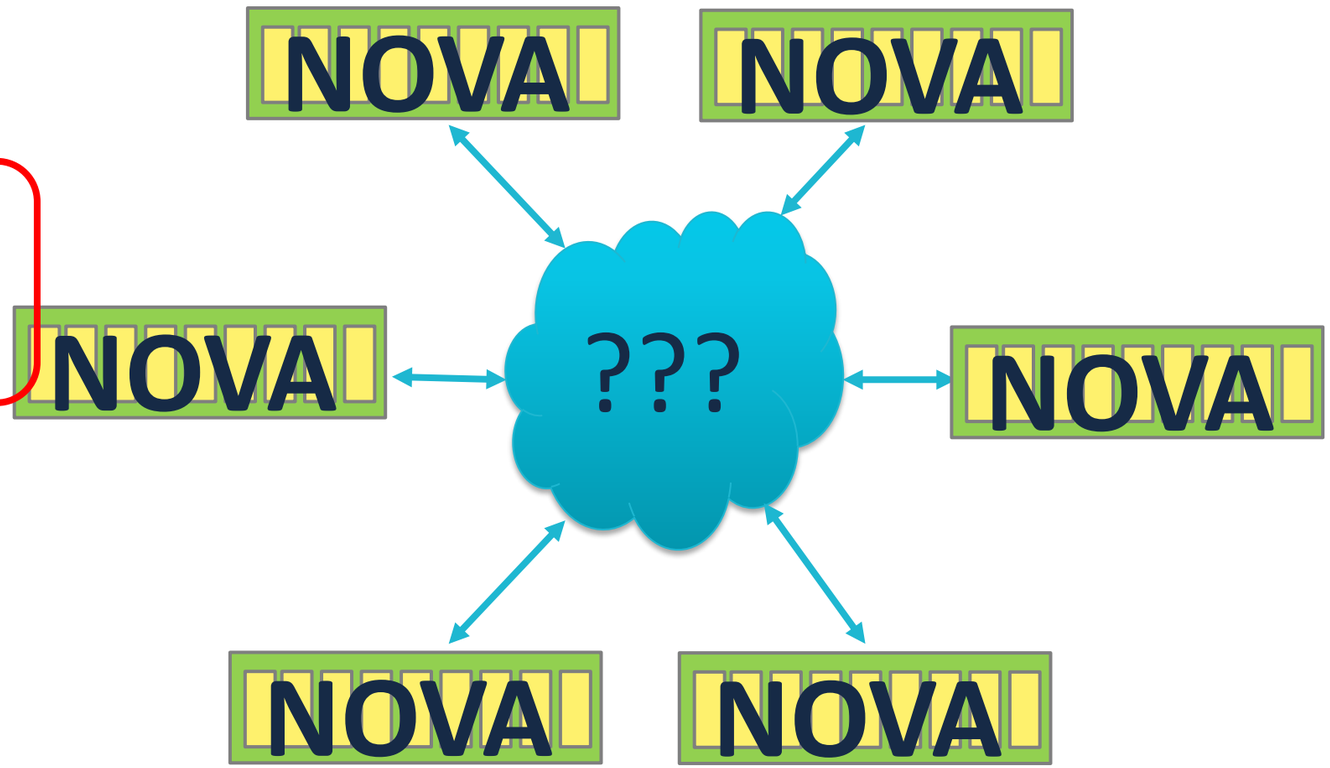
NVSL

Filebench throughput

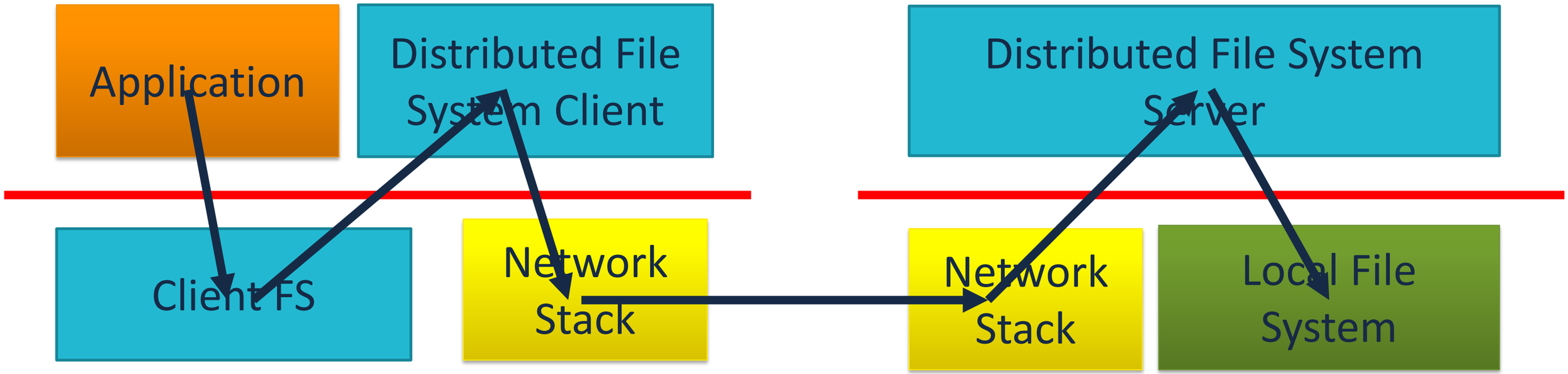


What Should You Do With NVMM?

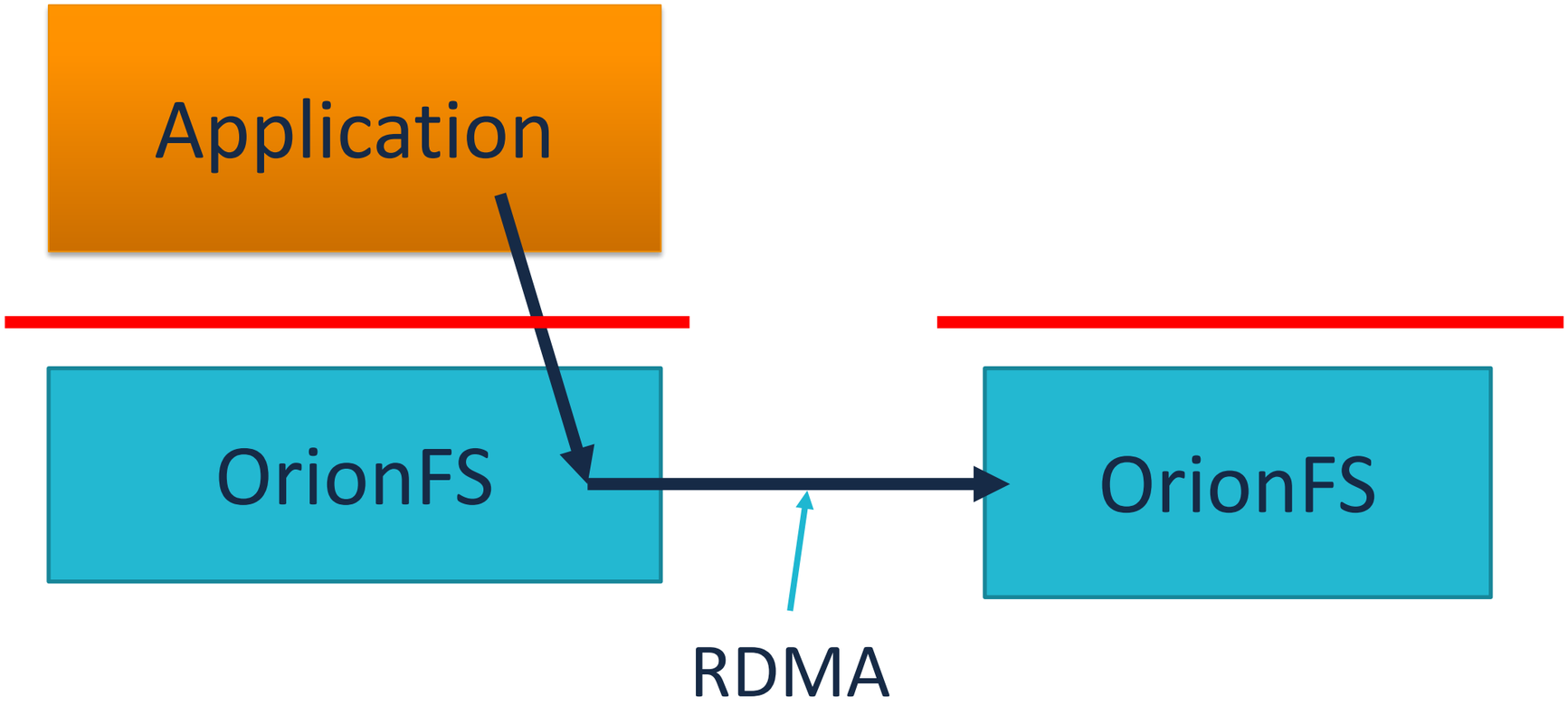
1. Use files and a conventional (distributed) file system
2. Use files and better file (distributed) system
3. Build persistent data structures
4. Use it as slow DRAM



Existing Distributed File Systems are Slow



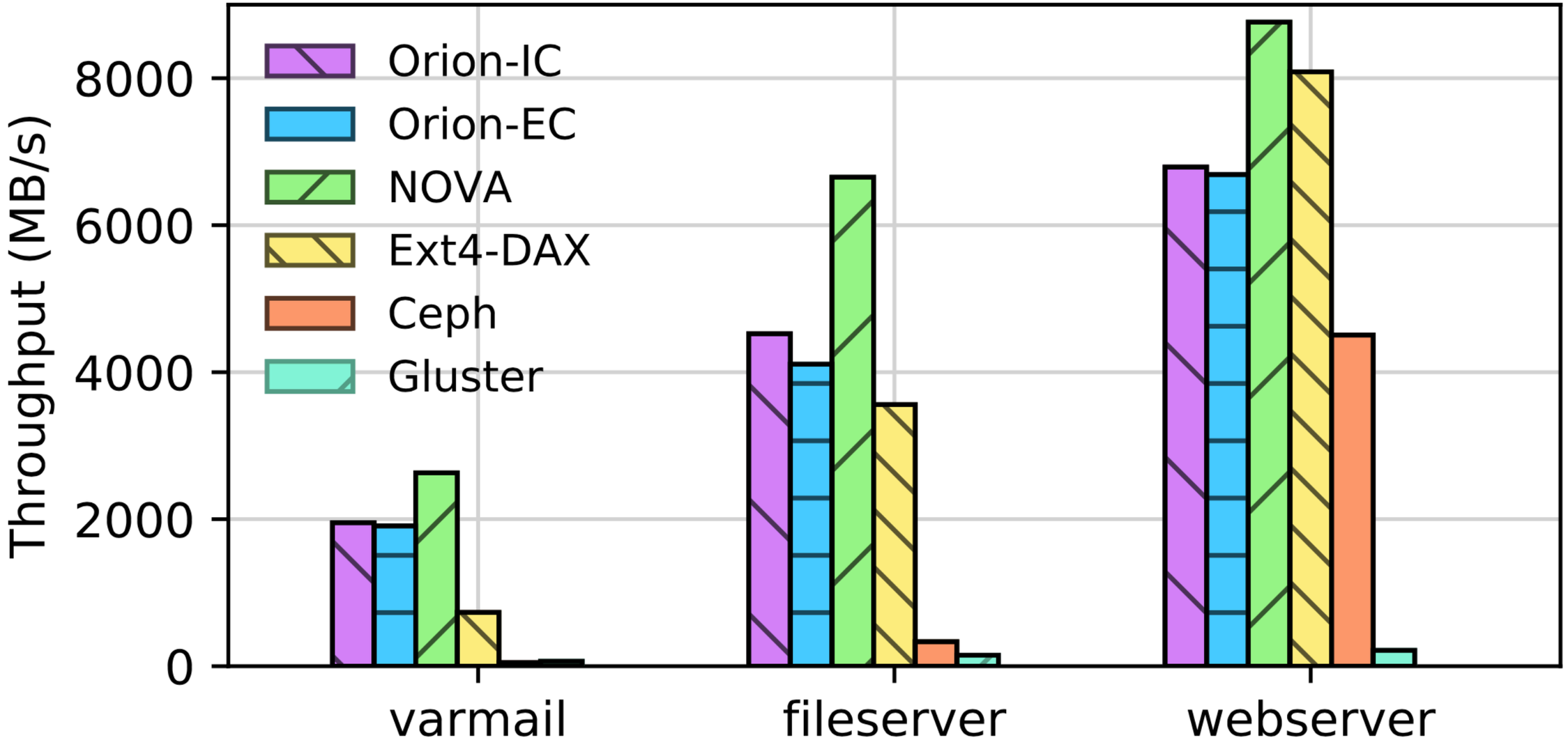
Orion: A Distributed Persistent Memory File System



Orion: Key Features

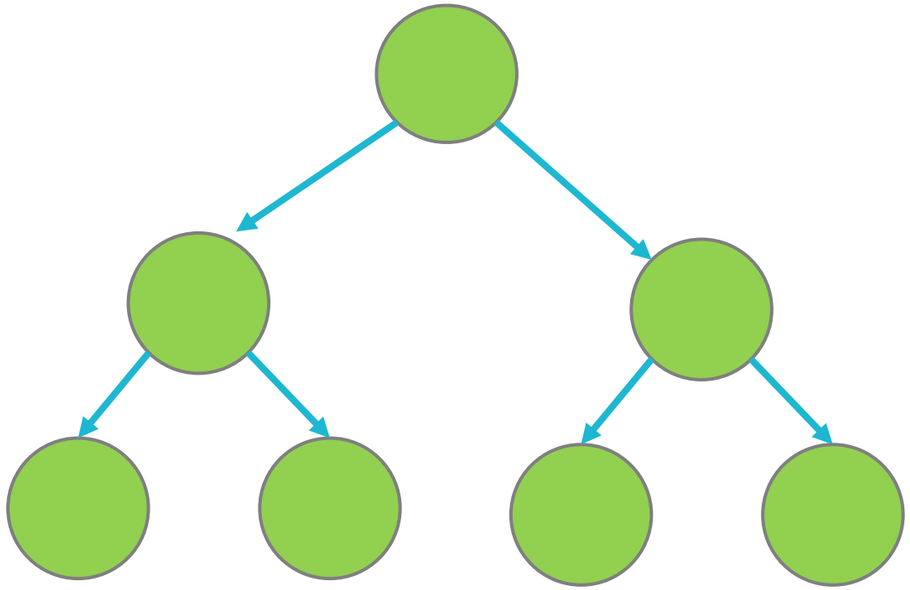
- Based on NOVA
- Mirrored metadata on client
  - Client keep local, NVMM copy of inode's log
  - Leases + simple arbitration for concurrent updates
- Mostly-local operation
  - Local read cache
  - CoW creates new, local copy
- Pervasive RDMA
  - All addresses/pointers are RDMA-friendly
  - Zero-copy IO for most transfers (NOVA data structures are RDMA targets)
  - Single-ended remote data access

Application performance on Orion



What Should You Do With NVMM?

1. Use files and a conventional (distributed) file system
2. Use files and better file (distributed) system
3. Build persistent data structures
4. Use it as slow DRAM

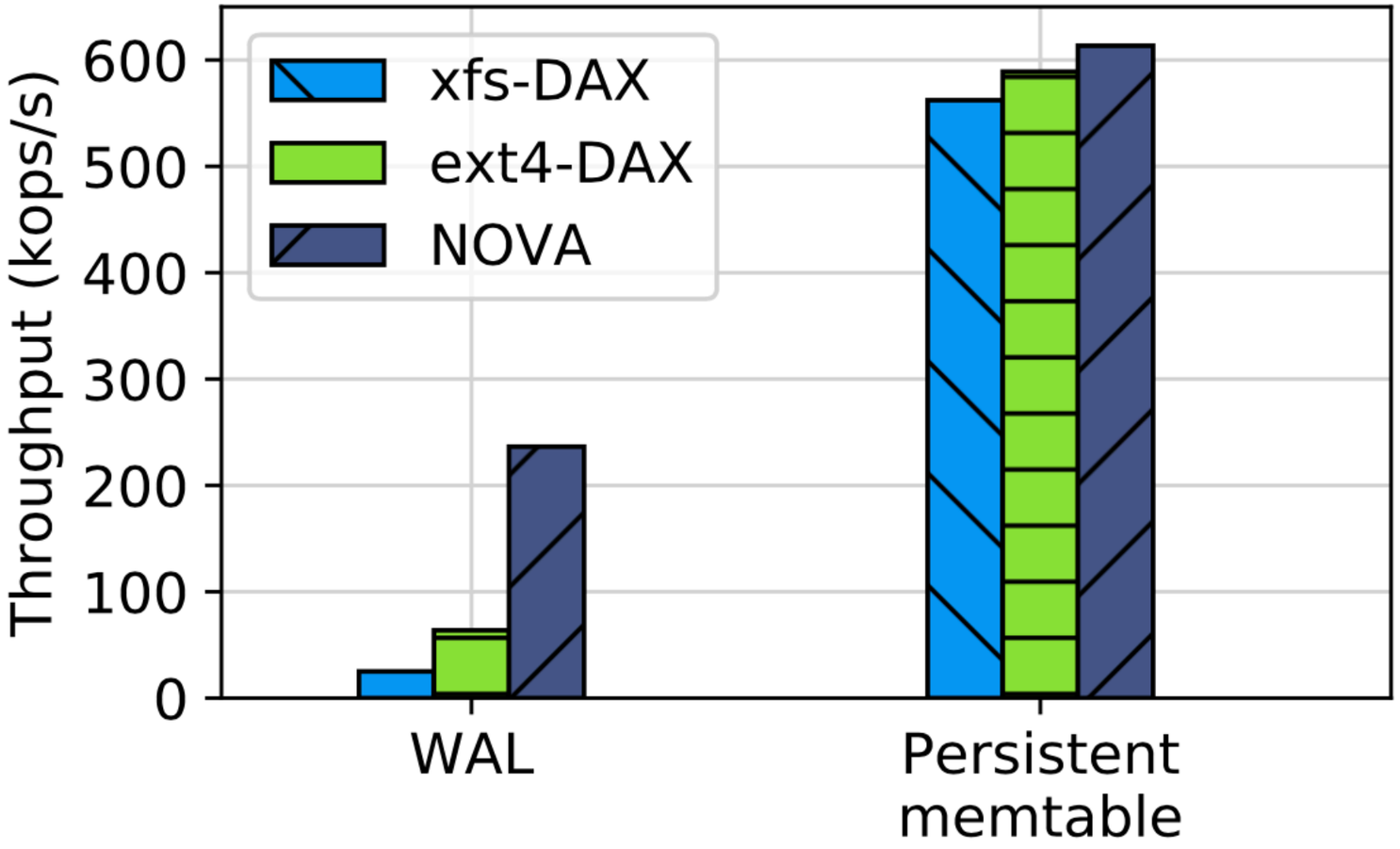


NVSL

Build NVMM Data Structures Is Hard

- All existing programming errors are still possible
  - Memory leaks
  - Multiple frees
  - Locking errors
- There are new kinds of errors
  - Pointers between NV memory pools
  - Pointers from NVMM to DRAM
- Programmers get this stuff wrong
- Rebooting/restarting won't help!
- Language + Compiler support will come, but slowly

Optimizing RocksDB



What Should You Do With NVMM?

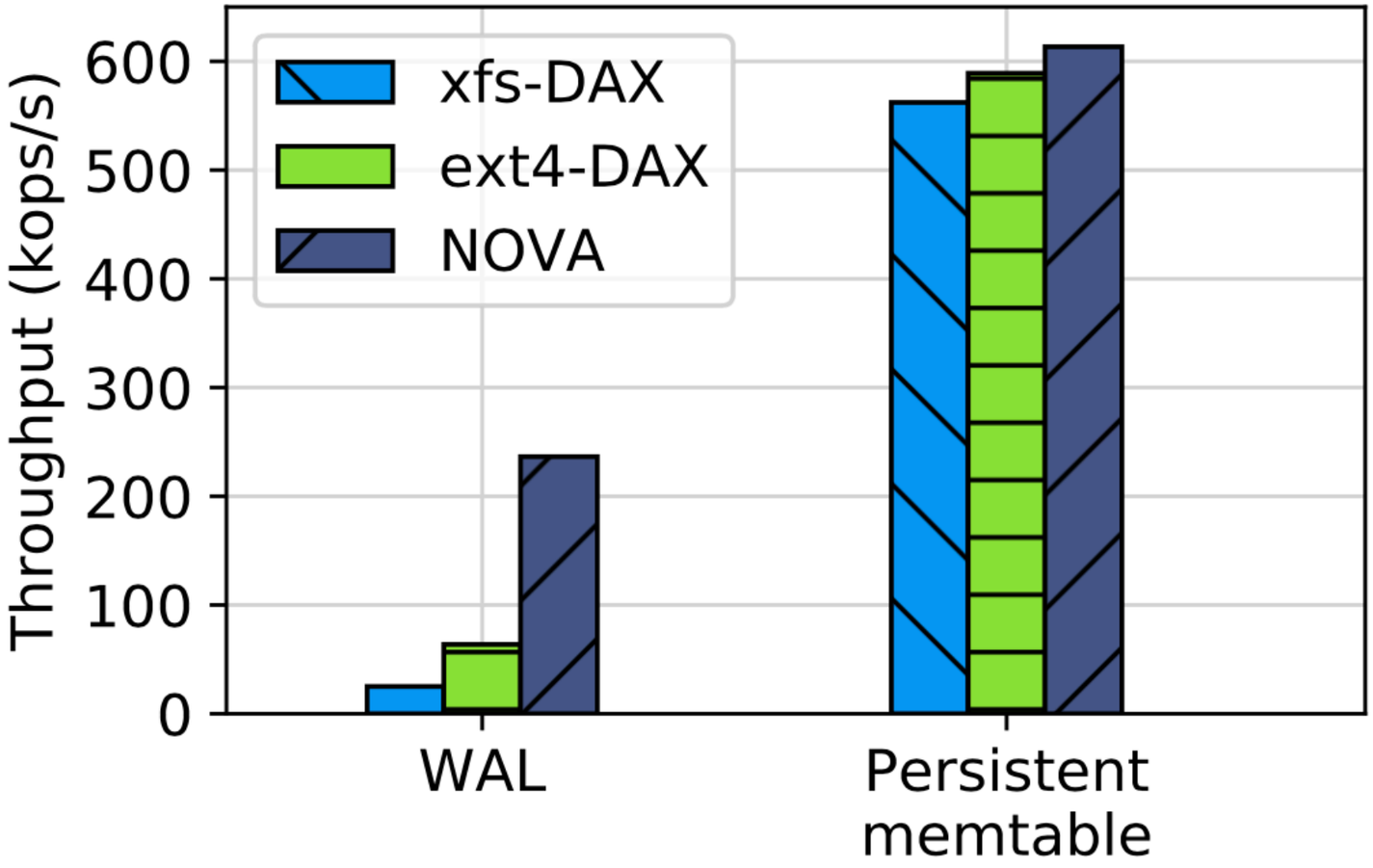
1. Use files and a conventional (distributed) file system → Easy; ~5x gains
2. Use files and better file (distributed) system → Pretty easy; ~10x gains

The NVMM Programmability Gap

3. Build persistent data structures → Really hard; ~30x gains
4. Use it as slow DRAM

NVSL

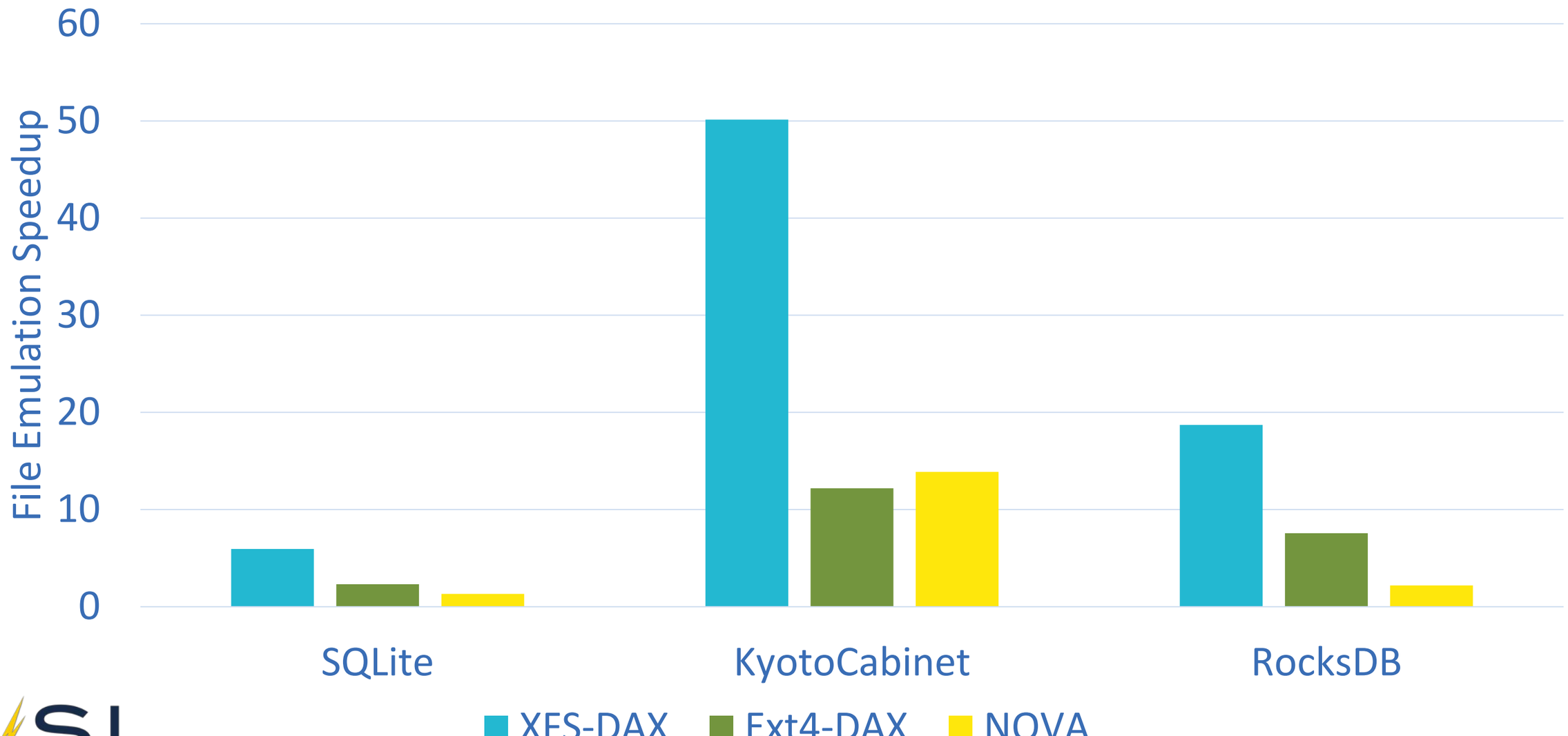
Optimizing RocksDB



File Emulation

- Normal write-ahead logging
  - open();
  - write(); sync();
- Emulate read/write in user space
  - open(); mmap();
  - memcpy() + clwb + fence
- Almost POSIX semantics
  - Minimal changes to app logic
  - No complex logging, allocation, or locking
- Almost persistent data structure performance
  - Just 10% slower.

File Emulation Speedups



NVSL

What Should You Do With NVMM?

- You should study it!
  - Many interesting, open problems remain
  - Lots of PhDs to come
- You should use it!
  - Use a file system!
  - Want more performance? Use file emulation!
  - Want more performance? Build persistent data structures.

NOVA is open source.
We are preparing it for “upstreaming” in to Linux.

To help or try it out:
https://github.com/NVSL/linux-nova

Thanks!

NVSL

UCSDCSE Computer Science and Engineering

Thank you!

Microsoft