USENIX ATC '24

Ethane: An Asymmetric File System for Disaggregated Persistent Memory

Miao Cai†, Junru Shen‡, Baoliu Ye*

†*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics*

‡*College of Computer Science and Software Engineering, Hohai University*

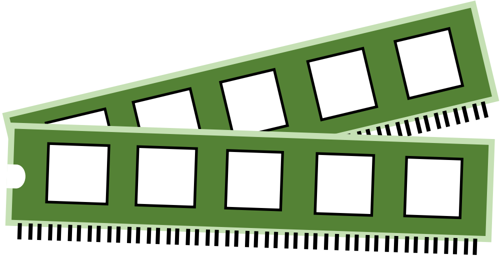
**State Key Laboratory of Novel Software Technology, Nanjing University*

usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

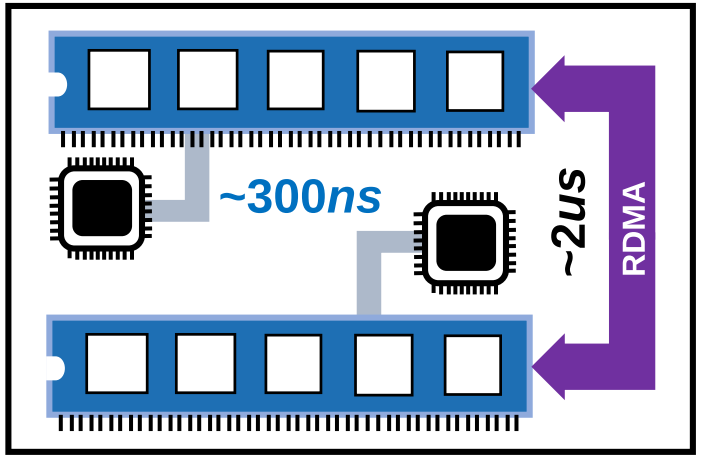
Outline

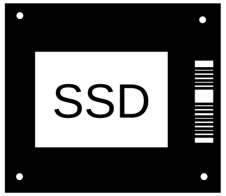
1. **Background and Motivation**
2. **Design and Implementation**
3. **Evaluation Results**
4. **Conclusion**

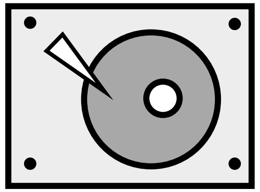
“Killer Microsecond” Problem



<100*ns*







70*us* - 10*ms*

COMMUNICATIONS OF THE ACM
CACM.ACM.ORG 04/2017 VOL.60 NO.04

Microsecond-scale I/O means tension between performance and productivity that will need new latency-mitigating ideas, including in hardware.

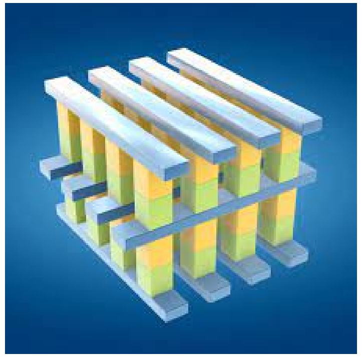
BY LUIZ BARROSO, MIKE MARTY, DAVID PATTERSON, AND PARTHASARATHY RANGANATHAN

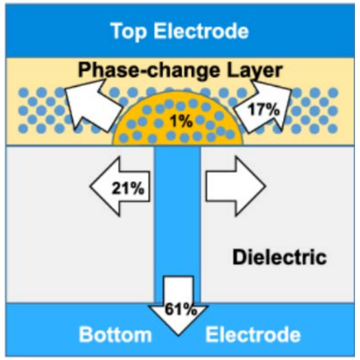
Attack of the Killer Microseconds

Google

Taming the Killer Microsecond

- **Ultra-fast Persistent Memory**



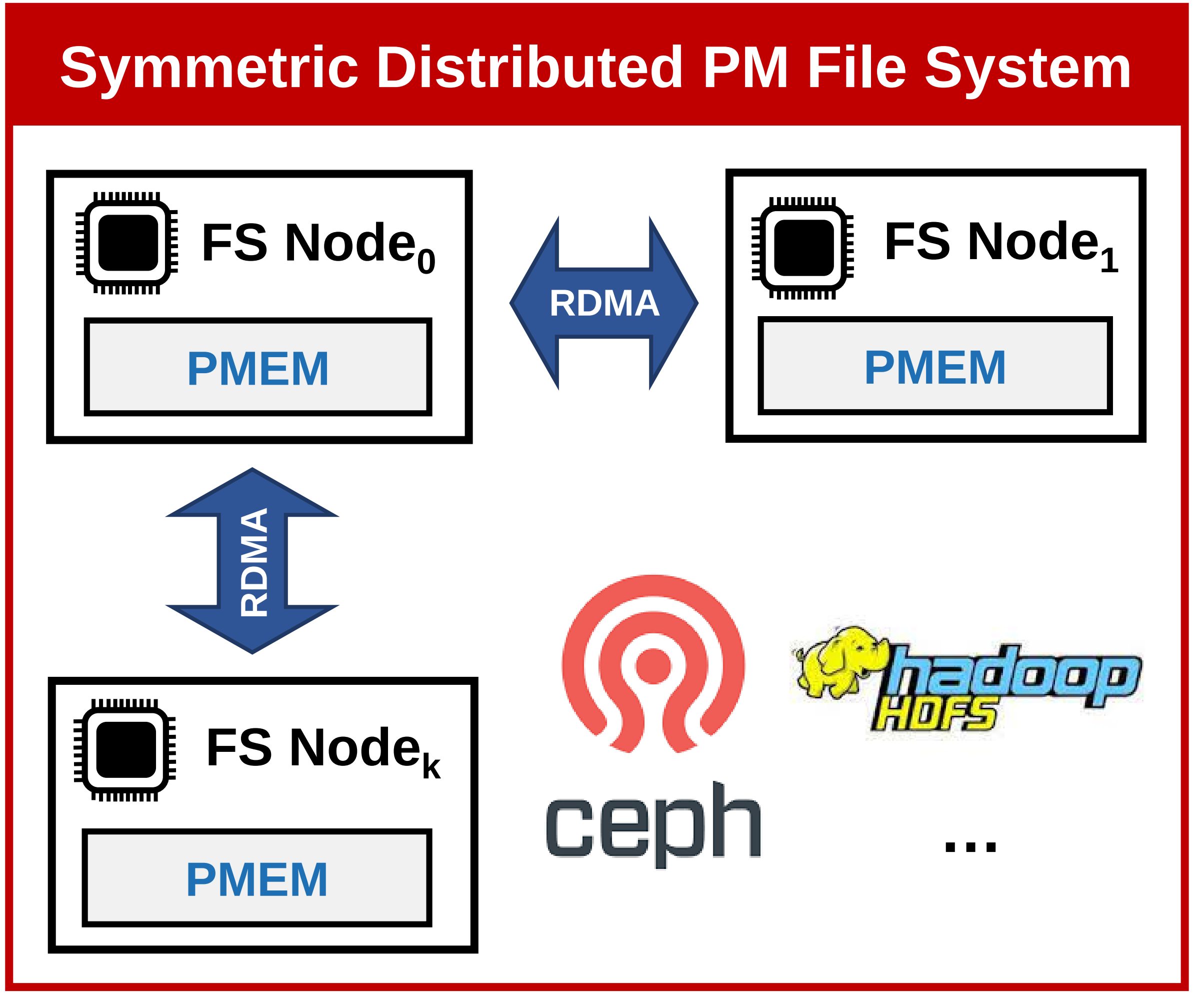


- **Low-latency Data Connectivity**

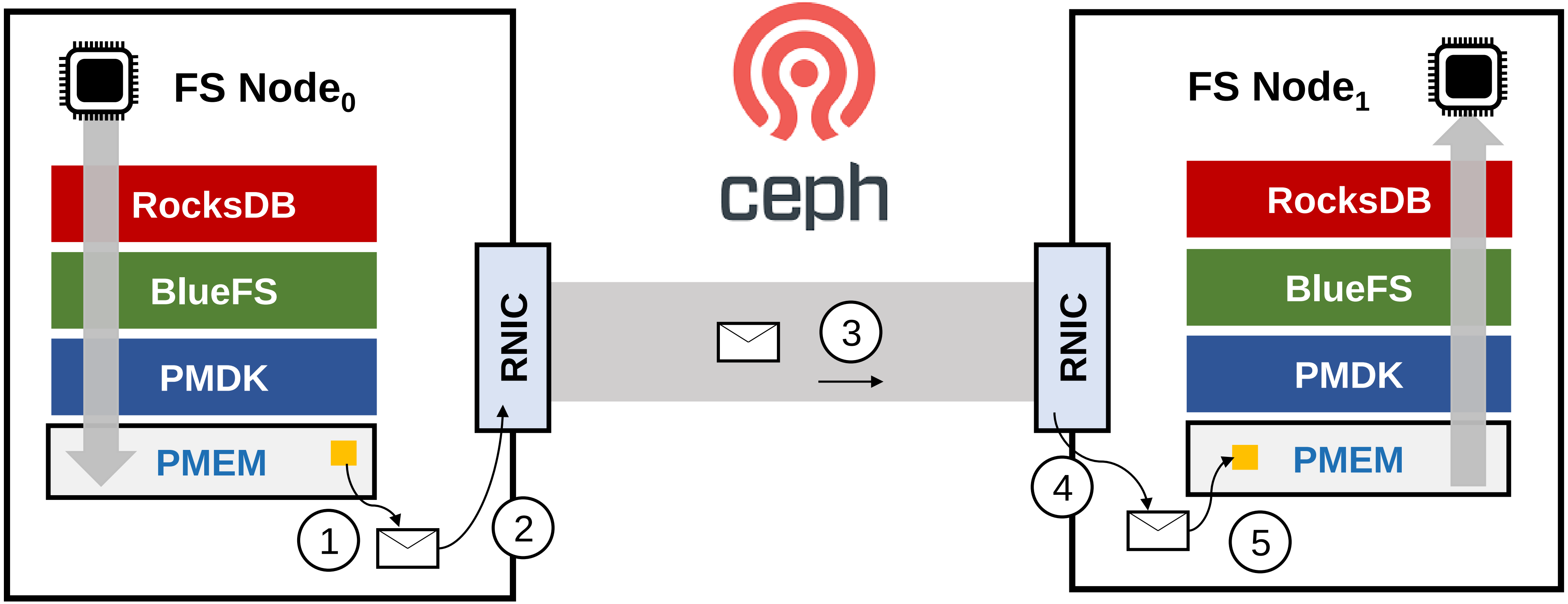
CXL Compute Express Link

NVIDIA NVLINK

RDMA Remote Direct Memory Access



#1 Expensive Cross-Node Interaction



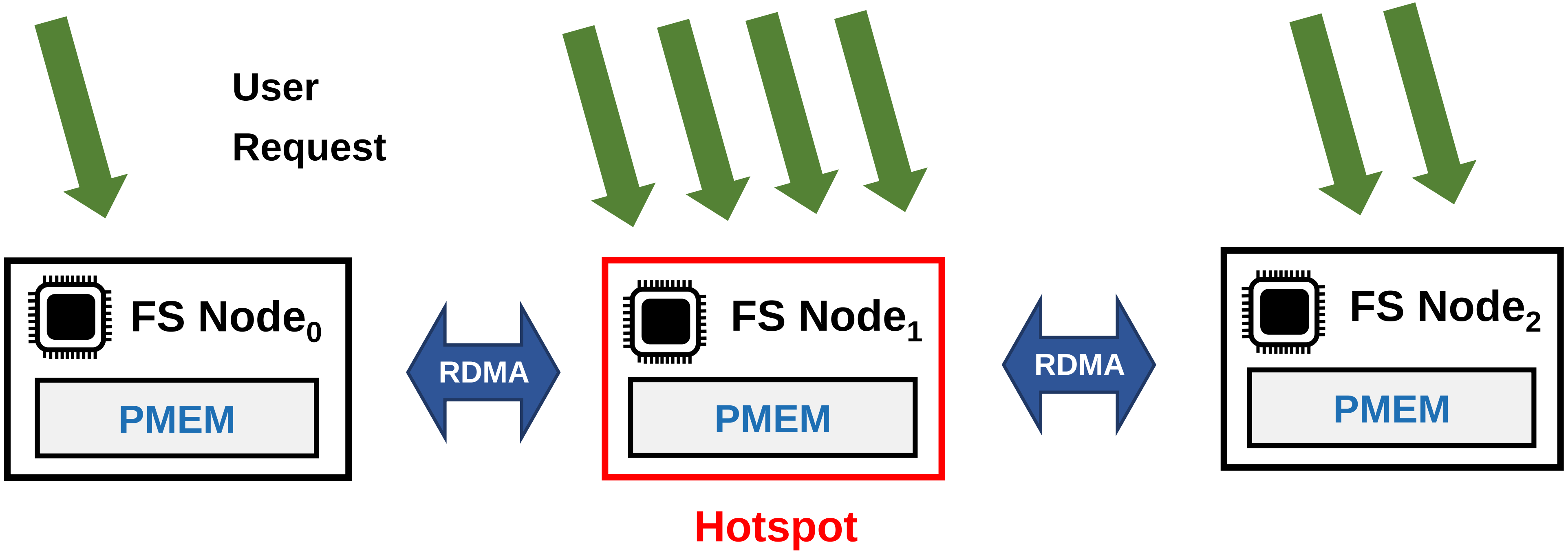
Scattered Data Storage

Interaction takes ~162us

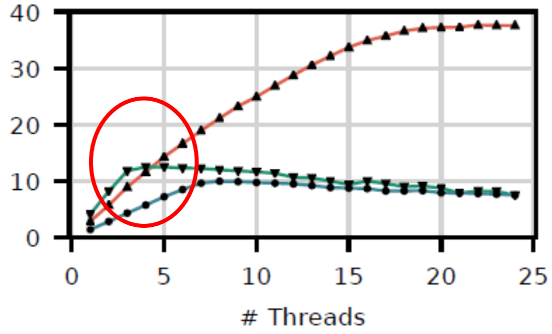
60.24% of total time

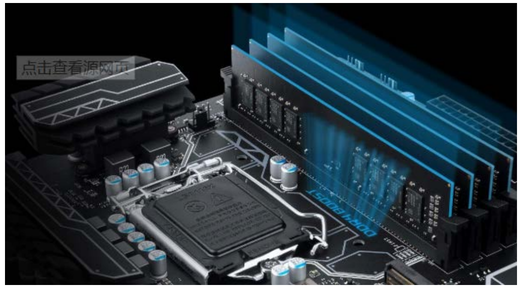
Long Data Path

#2 Weak Single-node Capability



Small Bandwidth

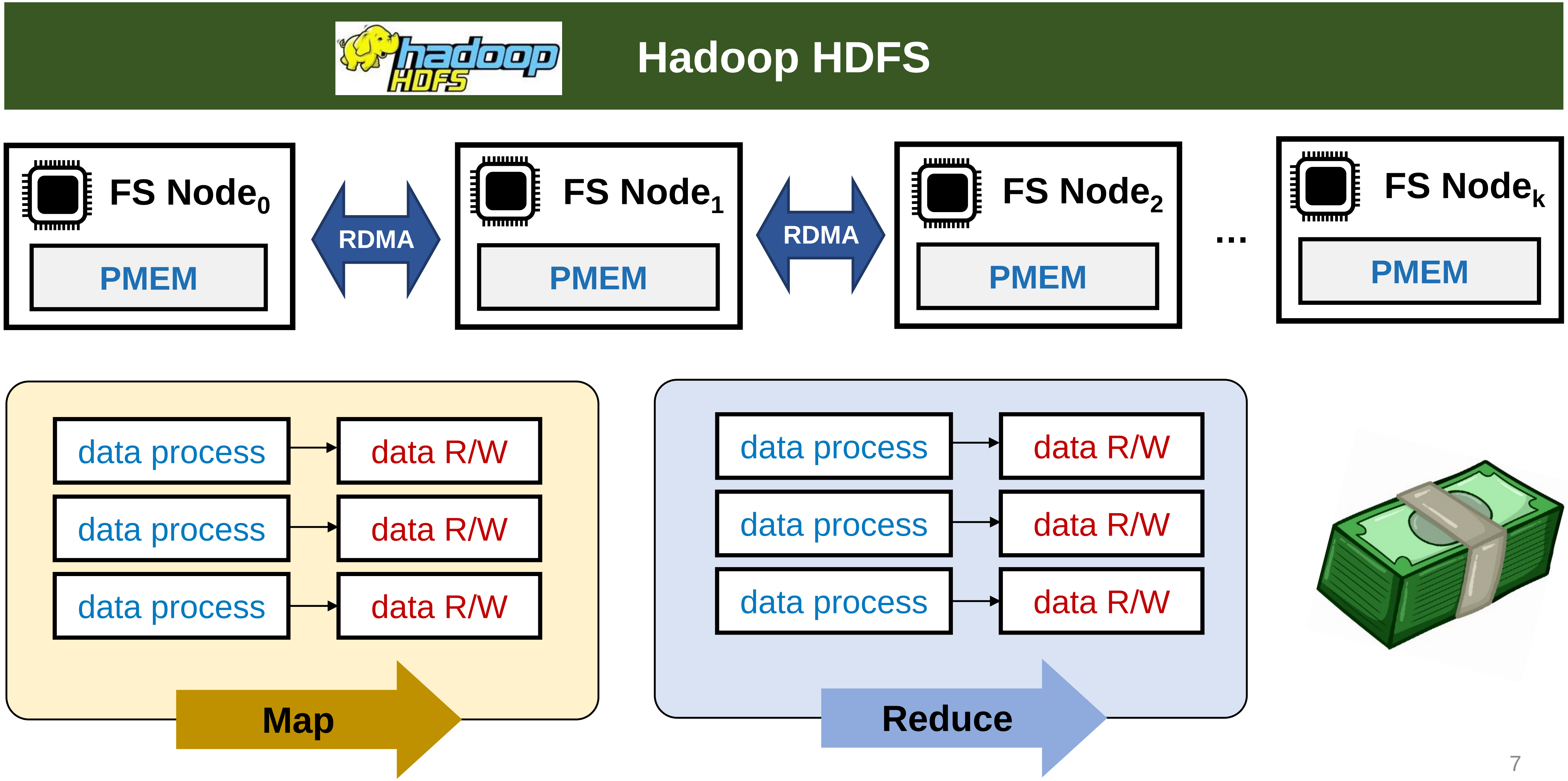




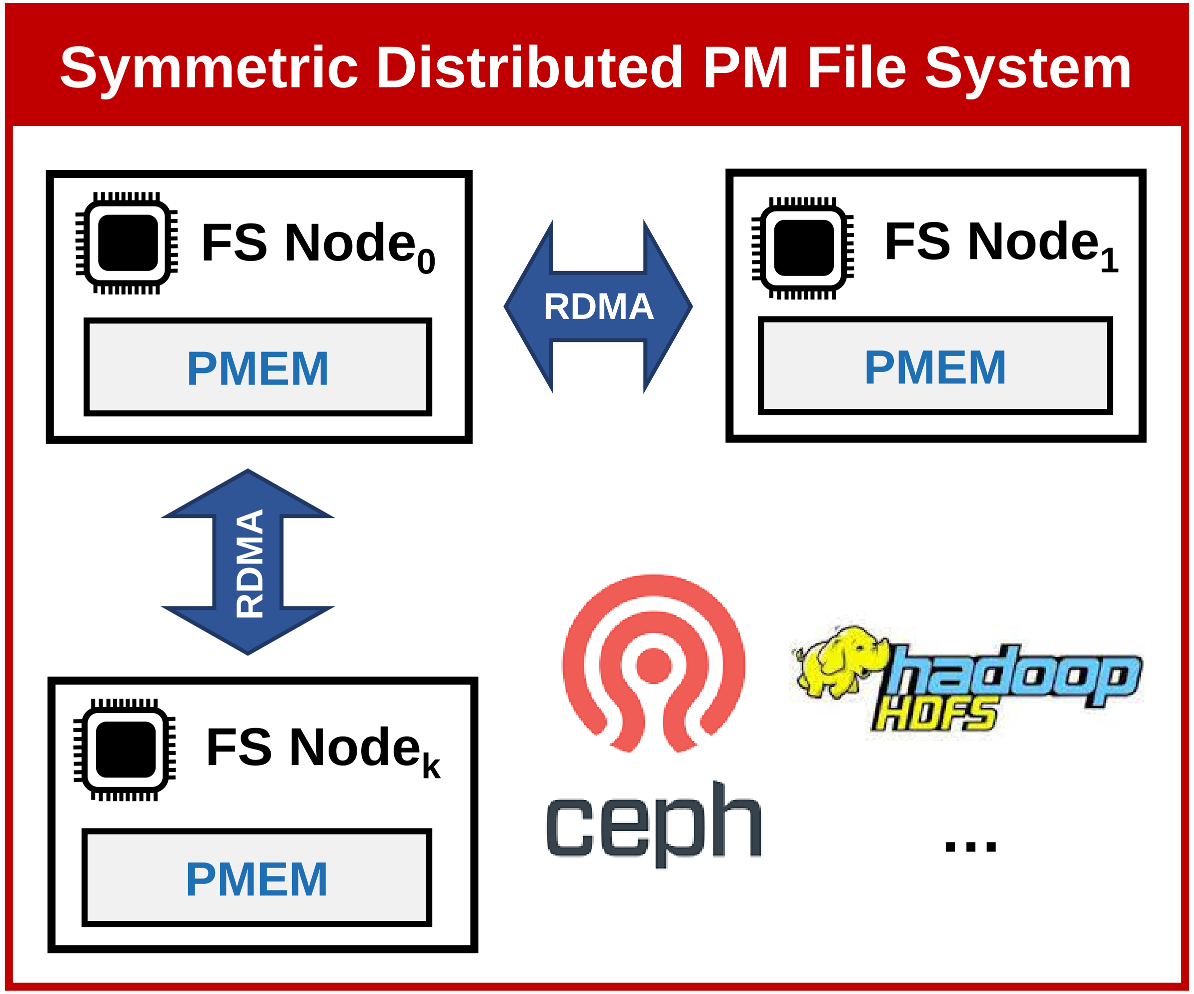
Limited Capacity

[1] Yang et al. *An Empirical Guide to the Behavior and Use of Scalable Persistent Memory*. FAST 2020

#3 Costly Scale-out Performance



Summary



Expensive Cross-node Interaction

Unpredictable Latency

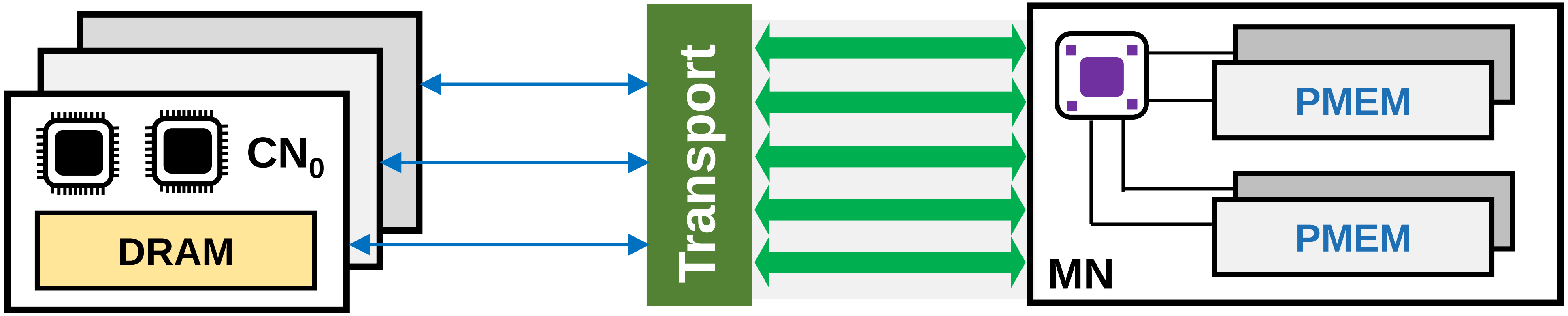
Weak Single-node Capacity

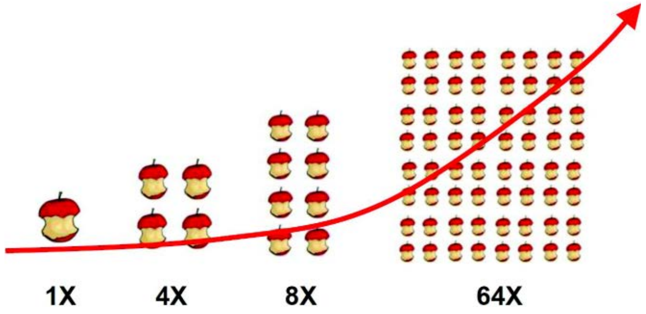
Load Imbalance

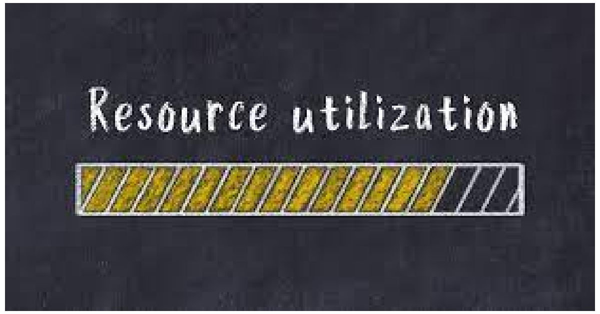
Costly Scale-out Performance

Monetary Cost

Disaggregated Persistent Memory





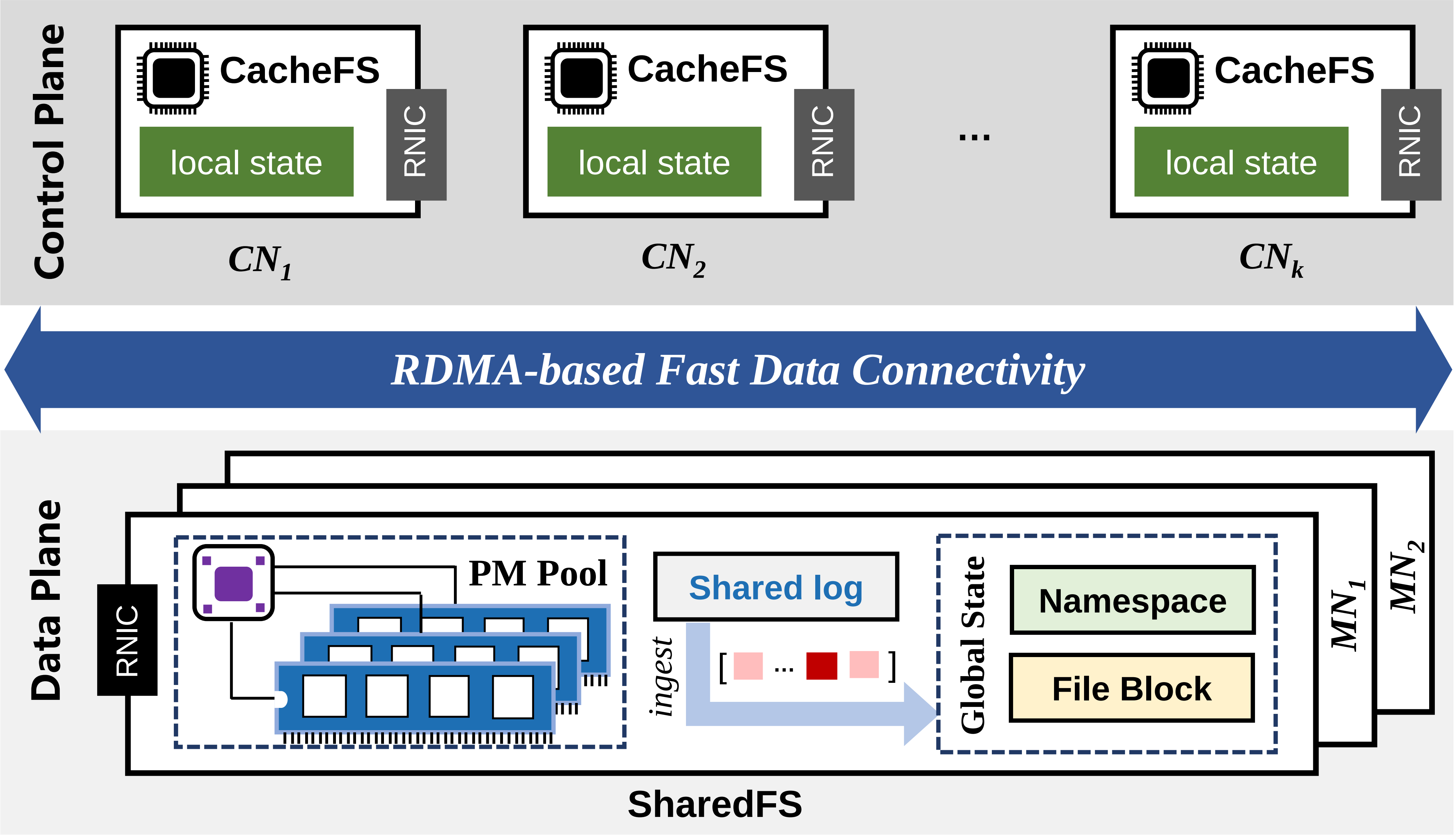


High Resource Elasticity

Strong Fault Isolation

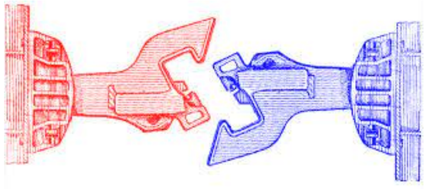
High Resource Utilization

Asymmetric File System Architecture



Asymmetric File System Architecture

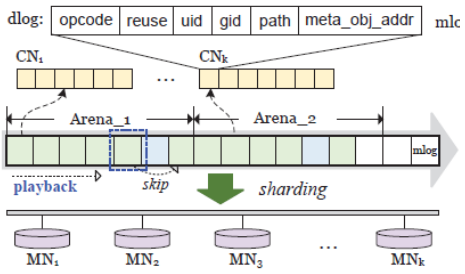
- **Separation of control and data plane**



Principle: separating FS object manipulation (e.g., *namespace query*) from storage (e.g., *meta- and data access*).

- **Best use of hardware resource**

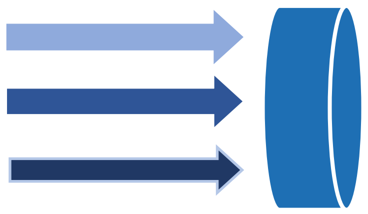
CacheFS: deploy on CNs to handle complex control logics
SharedFS: provide a global FS view and exploit hardware parallelism



- **Shared-log-based control-plane FS**

Why: 1) MN inherently supports efficient data sharing;
2) Handle data persistence, concurrency control, and state coherency at a time

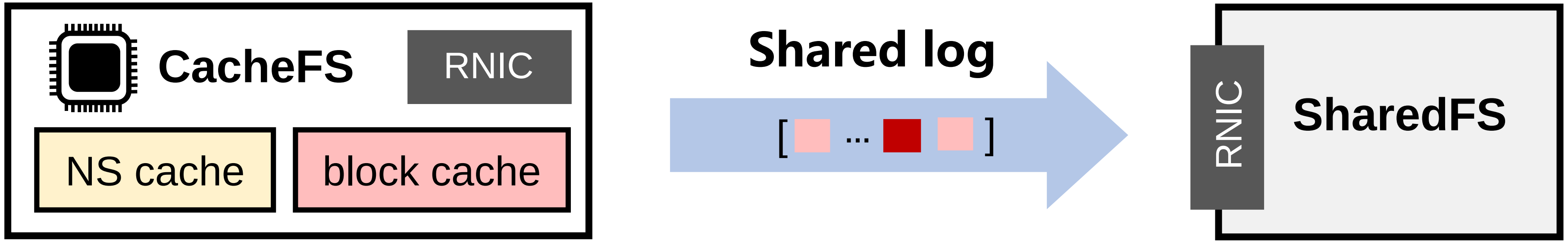
- **Access-disentangled data-plane FS**



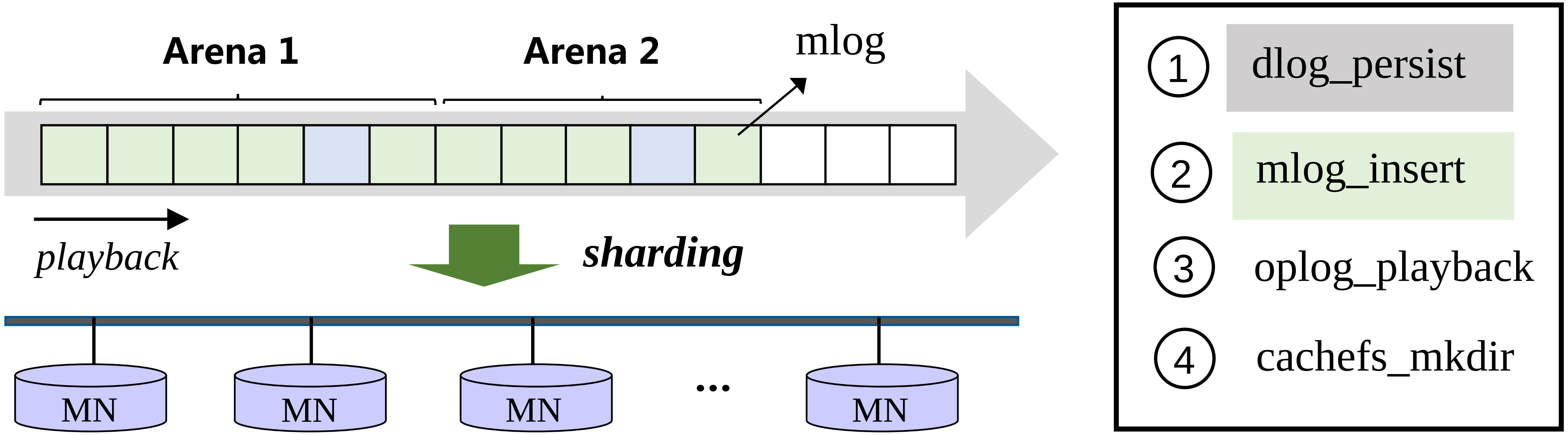
Disentangles the data access from coupled operations to reap the large aggregated bandwidth.

Control-plane FS Design

■ CacheFS structure



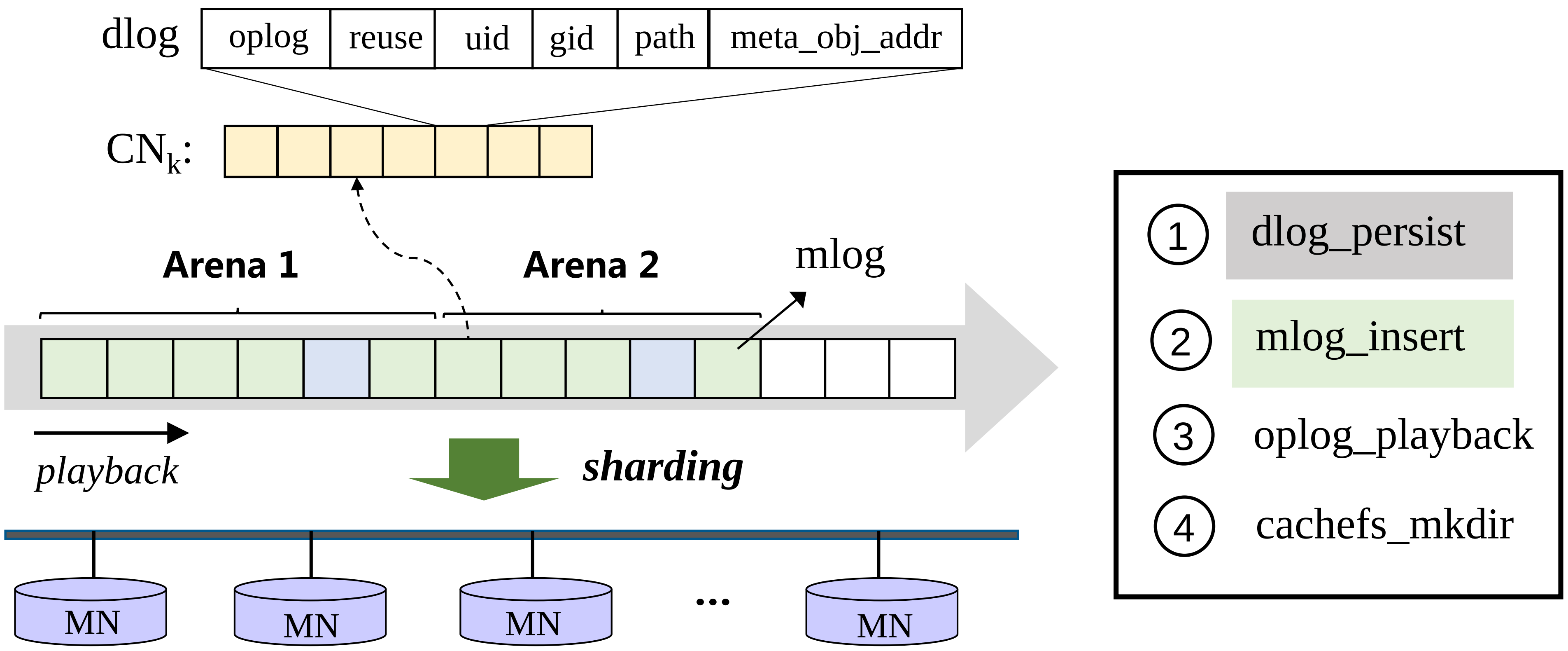
■ Shared log based CacheFS Design



Control-plane FS Design

- **Delegating Data Durability to Log Persistence**

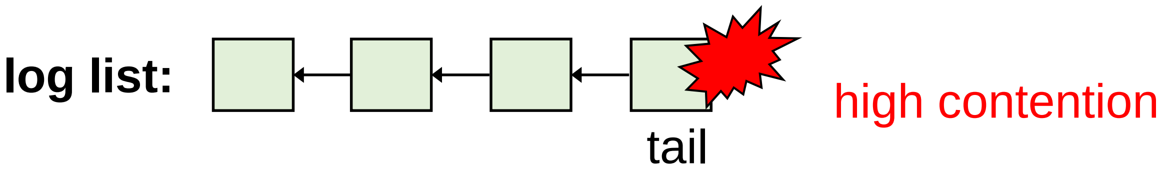
① dlog persist : *RDMA_WRITE + RDMA_READ*



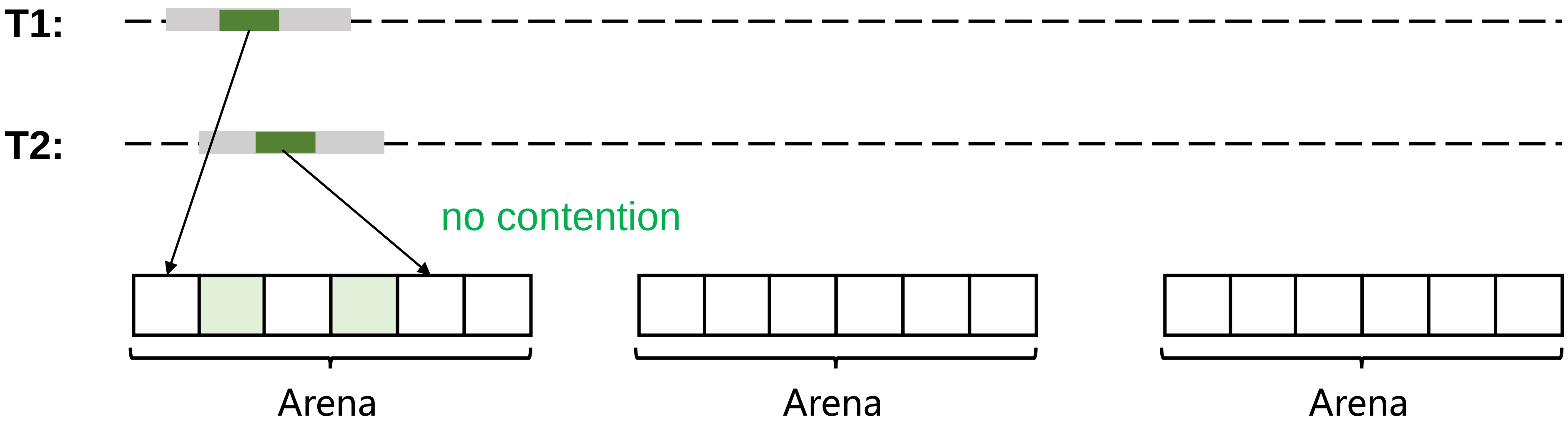
Control-plane FS Design

■ Delegating Syscall Linearizability to Log Ordering

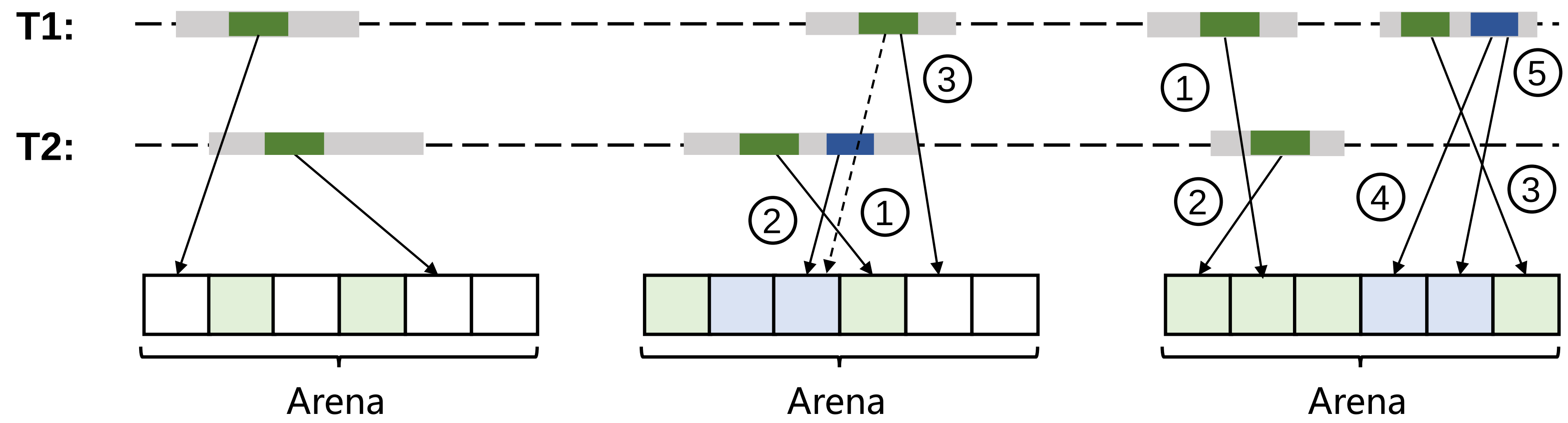
Naïve solution: using *RDMA_CAS* to append mlogs to a list one by one



Key insight: producing a sequence of mlogs for linearizable syscalls does not require linearizable mlog append.



Control-plane FS Design



T1: ③ fail and re-insert mlog

T2: ① insert mlog

② scan & insert pseudo mlogs

T1: ① insert mlog

③ insert mlog

④ ⑤ insert pseudo mlogs

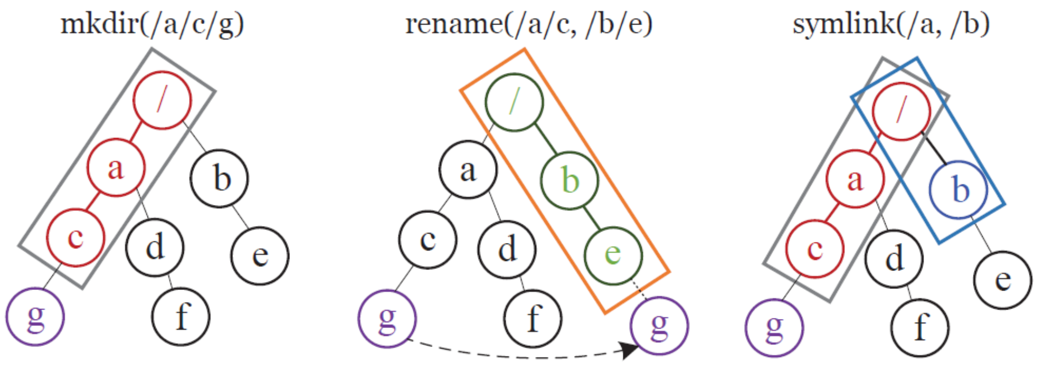
complete log history

T2: ② insert mlog

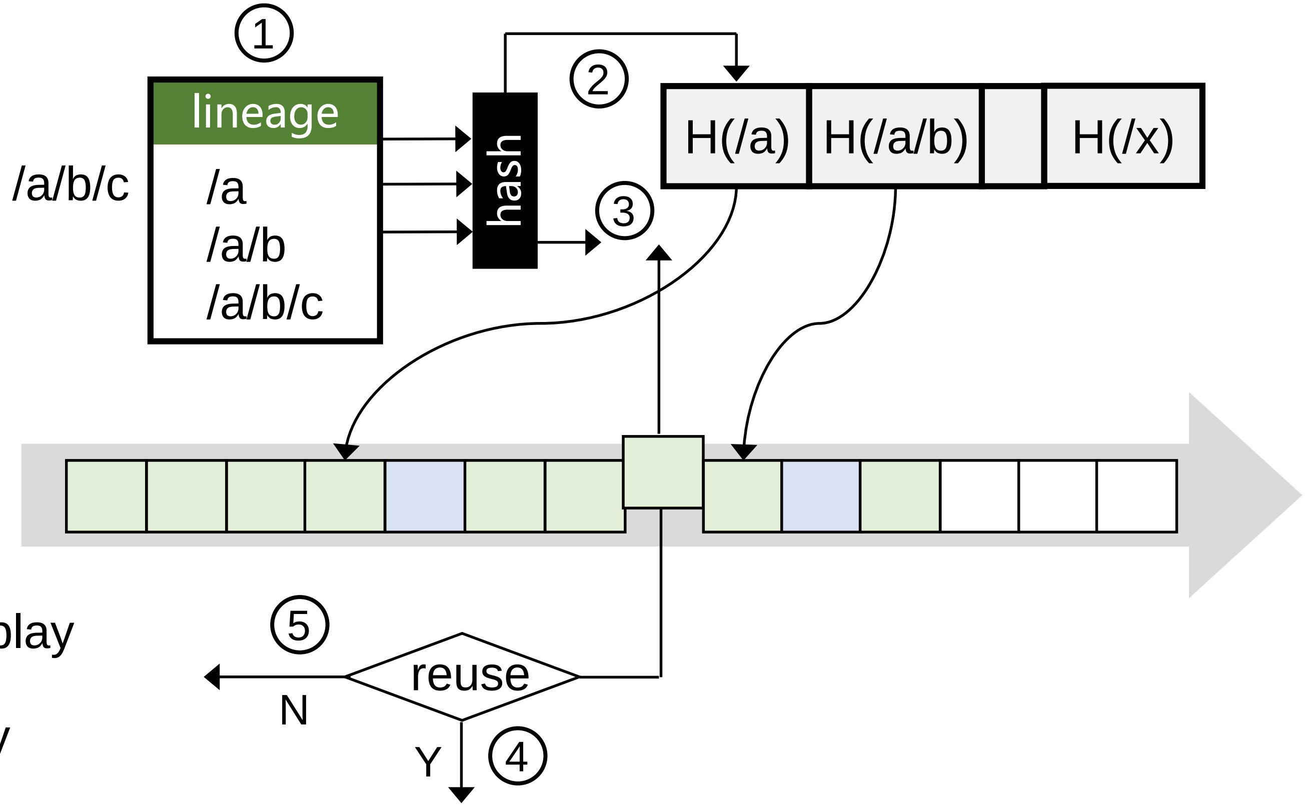
Control-plane FS Design

■ Delegating CacheFS Coherence to Log Playback

Reducing playback latency via 1) file-lineage-based dependence check and 2) collaborative log playback



(a) Direct Lineage (b) Step Lineage (c) Remote Lineage



① calculate file lineage

② query skip table

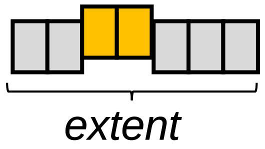
③ read mlog & check dependence

④ partial replay

⑤ full replay

Data-plane FS Design

■ Data Storage Paradigm

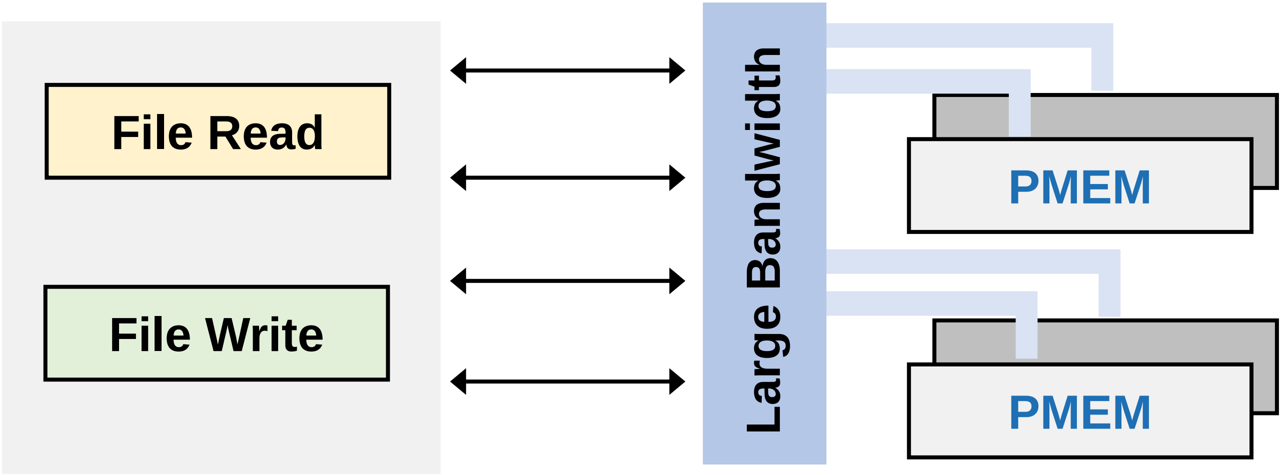
Type	Key	Value	
meta object	/a	[obj_1_addr, obj_0_addr]	obj_1
	/a/b/c	[obj_2_addr, obj_1_addr]	obj_2
	/a/hardlink	[obj_2_addr, obj_0_addr]	
	/a/b/symlink	[obj_3_addr, obj_1_addr]	obj_3
data section	[obj_2_addr, start_addr, section_size]	extent_addr →	

Access Interface

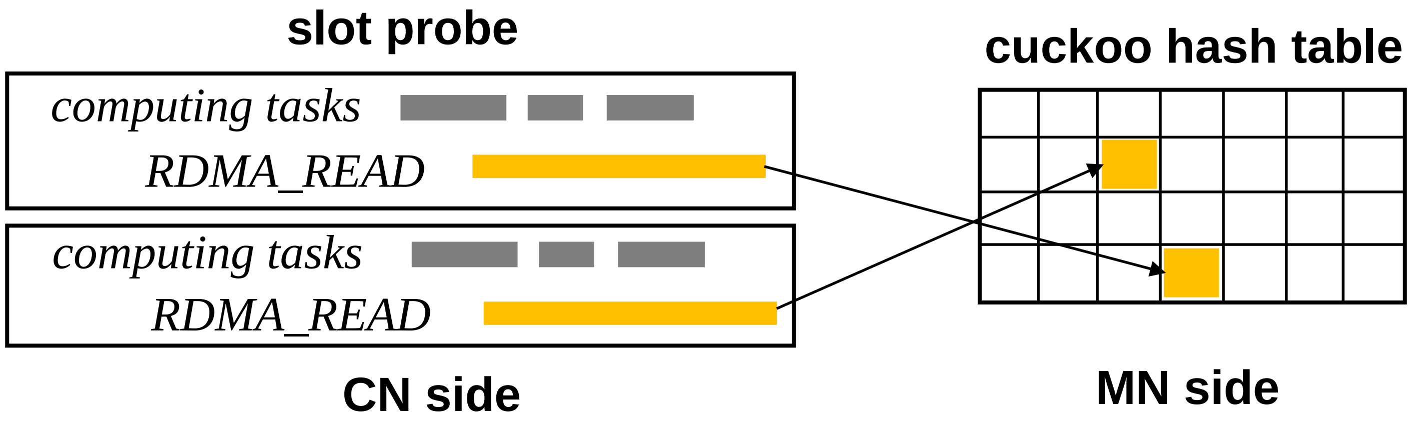
- int vec_kv_get(key_t *k_vec, val_t *v_vec)
- int vec_kv_put(key_t *k_vec, val_t *v_vec)

Data-plane FS Design

- Reap Large Aggregated PM Bandwidth

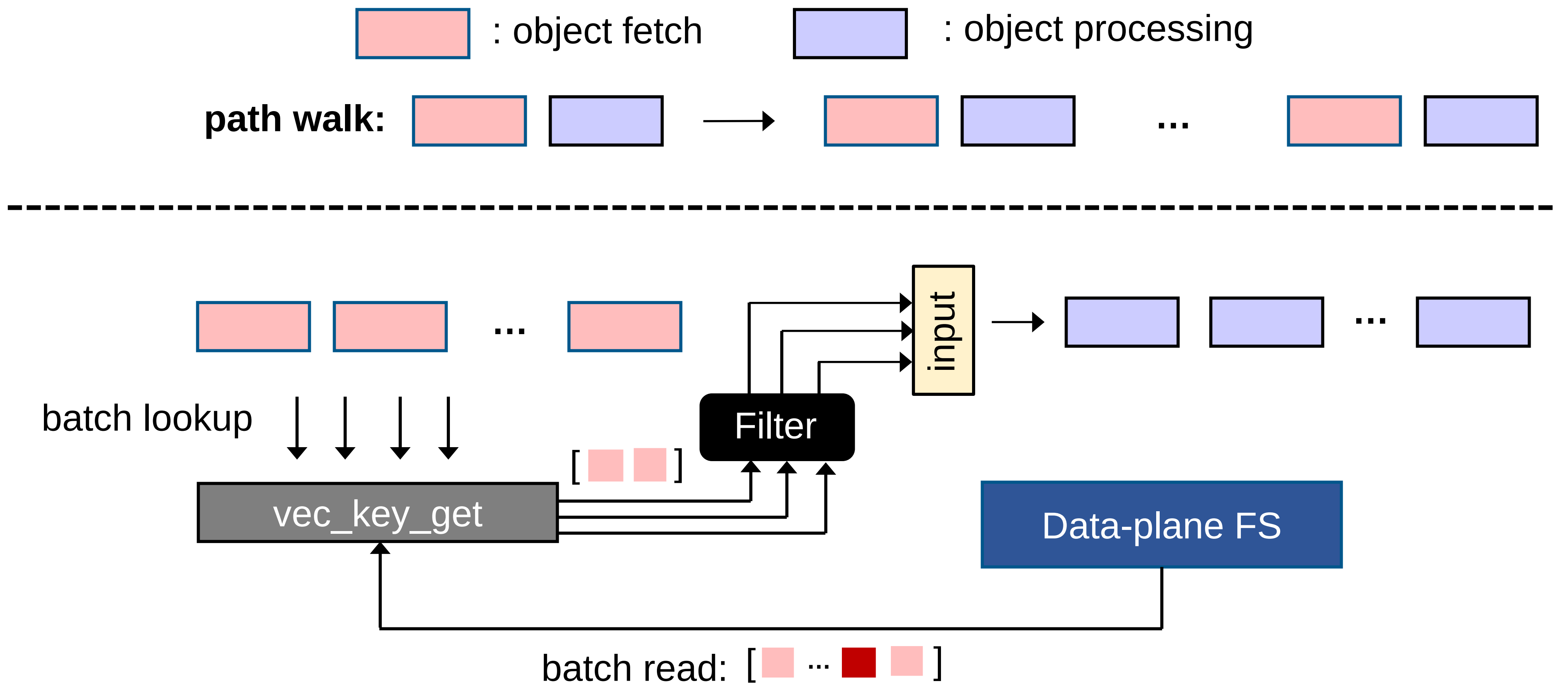


Pipelined hash lookup



Data-plane FS Design

- **Example: file path walk**



Evaluation

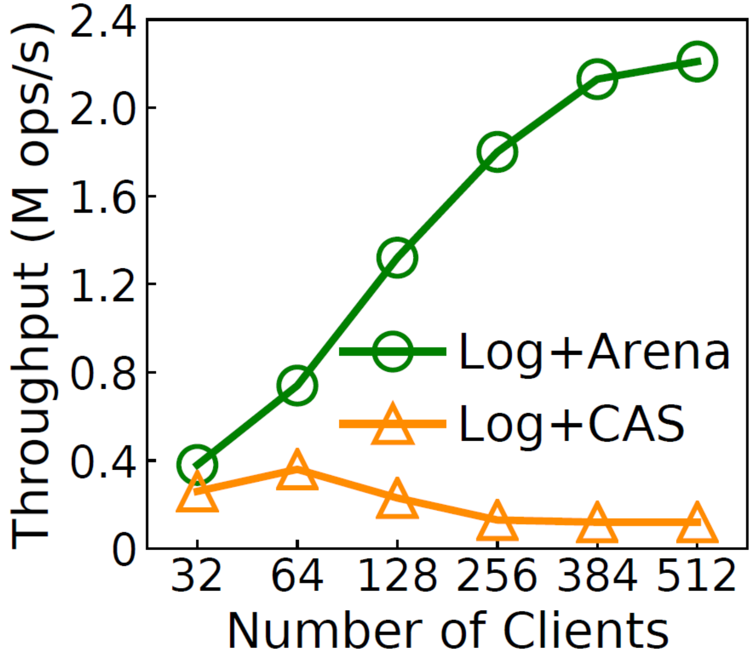
■ Node Configuration

Processor	2 × Intel Xeon Gold 5220 (24 cores)
Memory	128 GB (4 × 32 GB) DRAM + 512 GB (4 × 128 GB) NVM
Storage	512 GB NVMe SSD
Network	2 × Mellanox ConnectX-6 100 GbE NICs

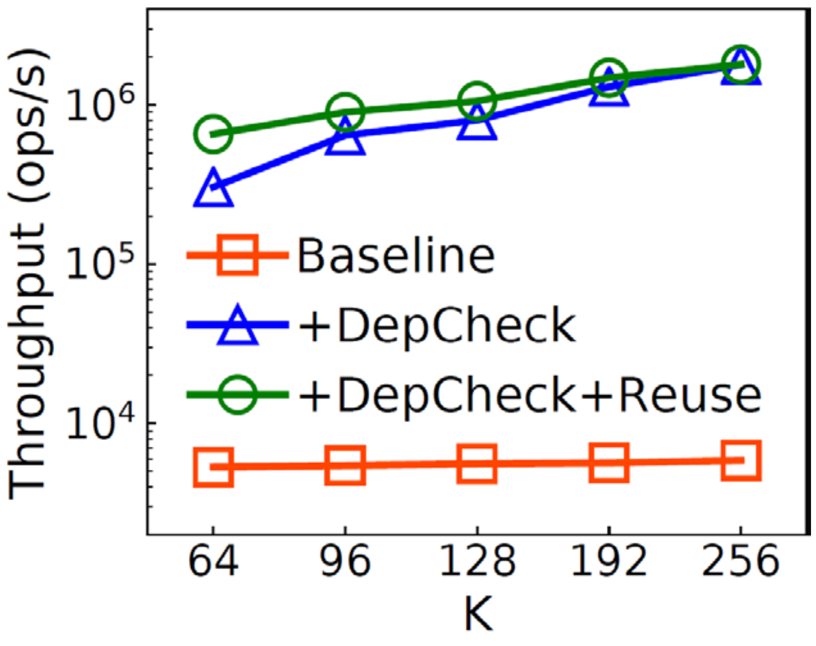
■ Disaggregated and Symmetric PM systems

	CPU	DRAM	PMEM	NIC	Price
CN	32 cores	8GB DDR4		2 × ConnectX-6 NIC	$3919
MN	1 cores	8GB DDR4	4 × 128GB DCPMM	2 × ConnectX-6 NIC	$3463
SN	16 cores	2 × 32GB DDR4	2 × 128GB DCPMM	ConnectX-6 NIC	$3789

Control-plane FS Evaluation



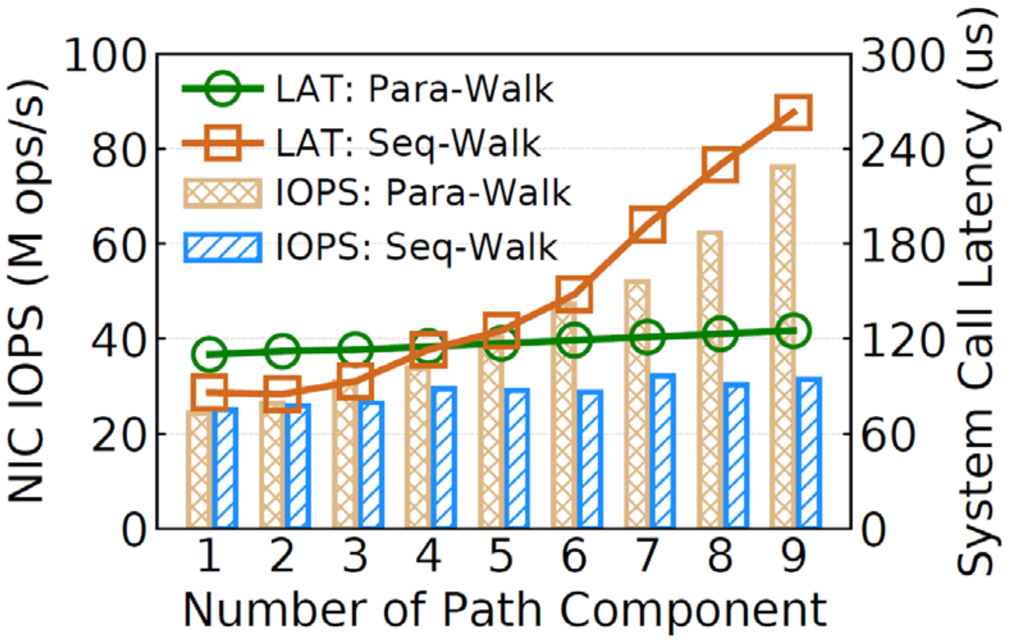
(a) Log Arena Scalability



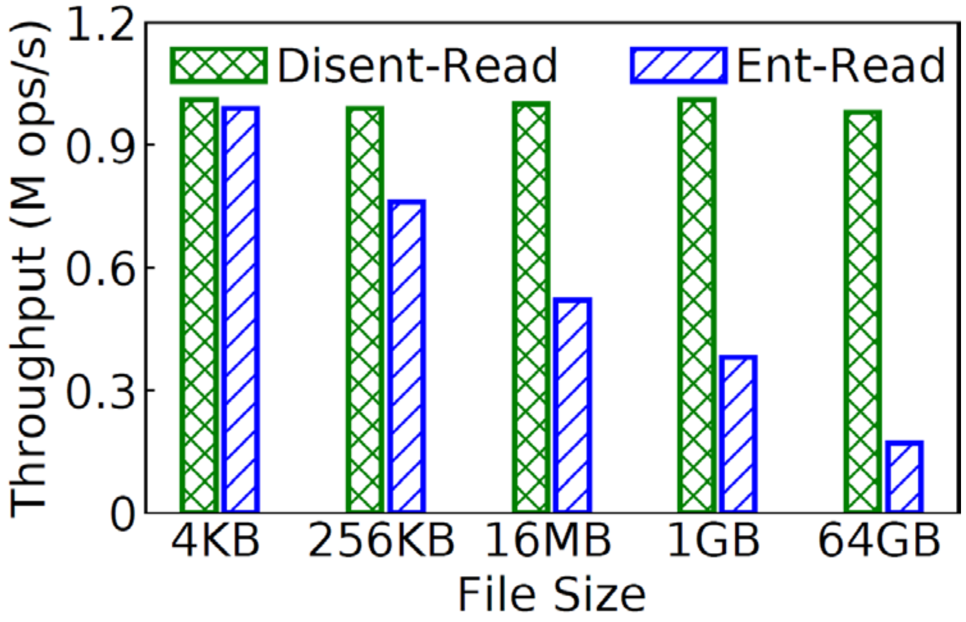
(b) Log Replay Performance

- **Log+Arena scales much better than Log+CAS due to small RNIC contention**
- **+DepCheck+Reuse performs 42%/209x better than +DepCheck and baseline**

Data-plane FS Evaluation



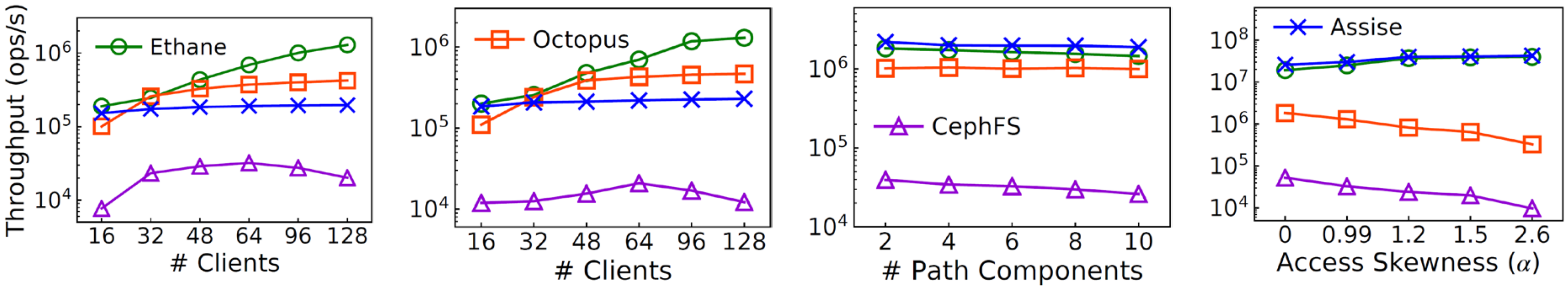
(a) Path Walk Latency



(b) Data Read Throughput

- **Para-Walk delivers a much more stable latency than that of Seq-Walk**
- **Ent-Read throughput is higher than Disent-Read by up to 5.76x**

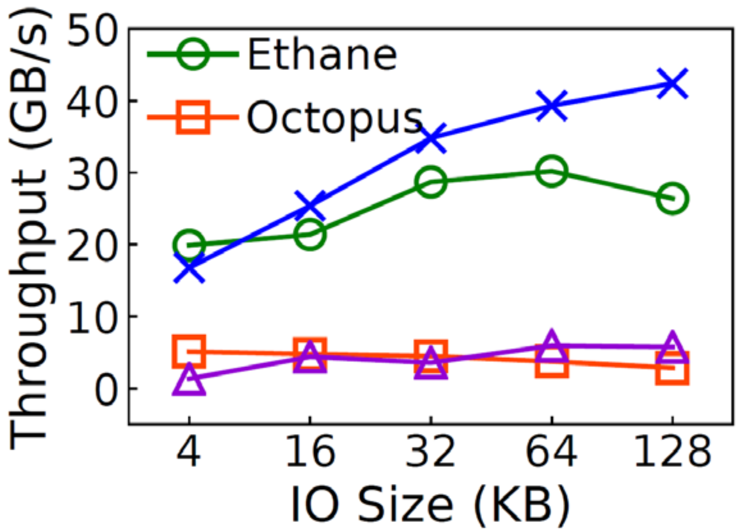
Macrobenchmark Performance



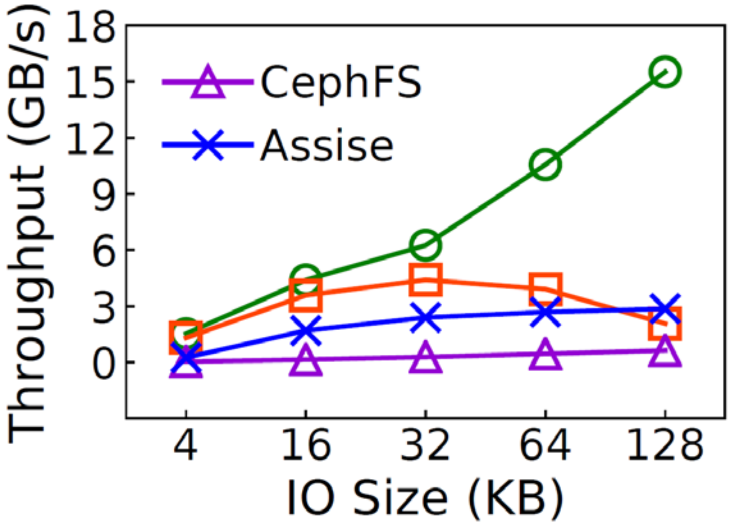
(a) creat (b) unlink (c) stat (d) stat (skew)

- **Frequent cross-node communication degrades CephFS throughput**
- **The single-threaded MDS prevents Octopus scalability improvement**
- **The chain replication protocol incurs excessive remote writes for Assise**

Macrobenchmark Performance



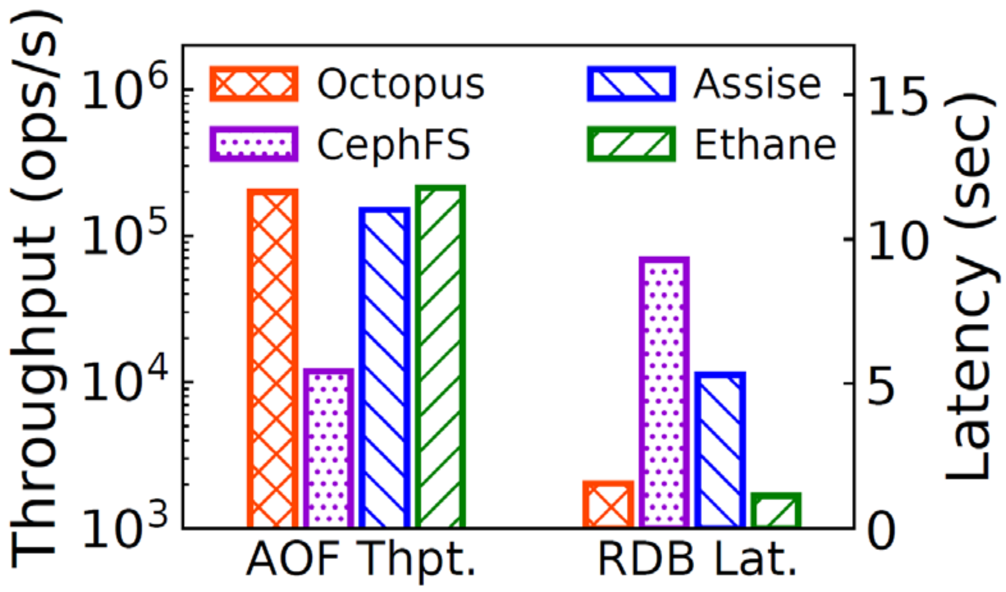
(a) read



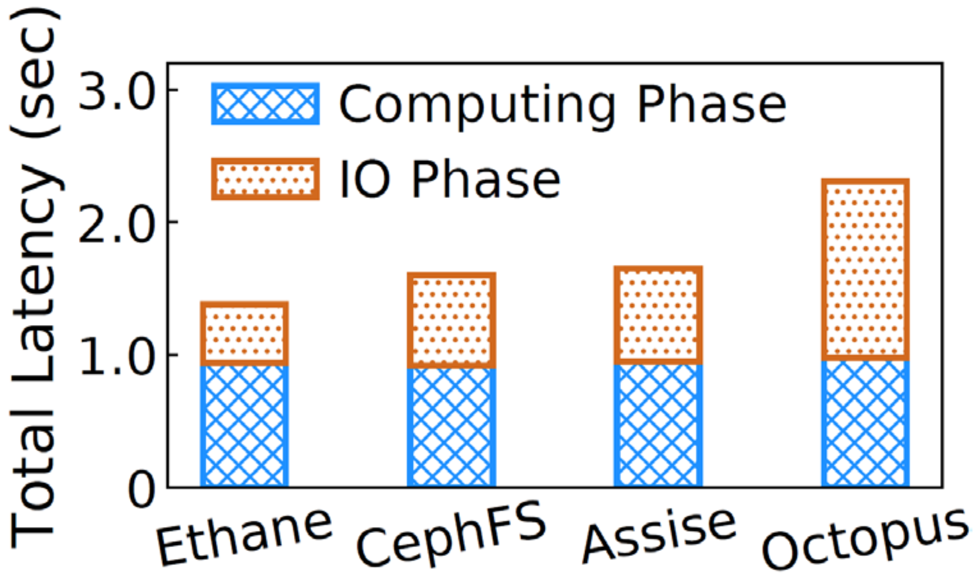
(b) write

- **The load imbalance issue decreases Octopus throughputs**
- **Client-local NVM design improves Assise performance**

Application Performance



(a) Redis Cluster

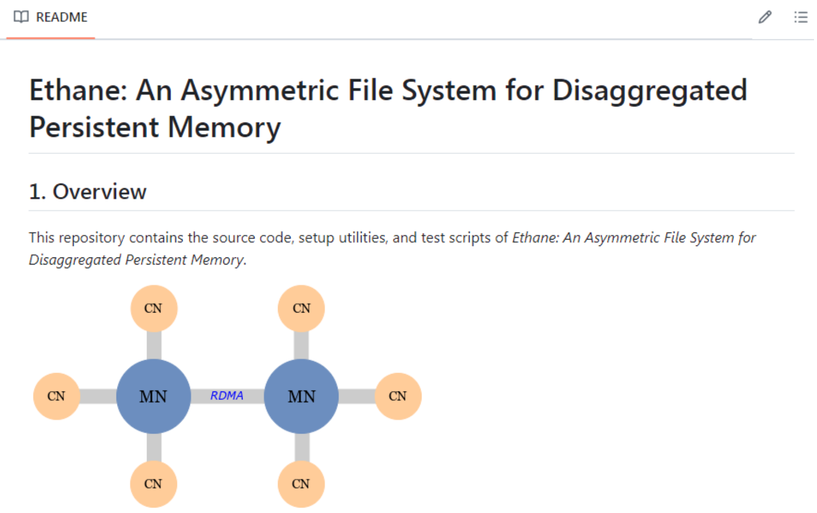


(a) Metis

- **Ethane achieves 6.77%/17.98×/41.55% higher AOF throughputs than Octopus/CephFS/Assise**
- **Ethane yields superior performance than others with the same hardware budget.**

Conclusion

- **We reveal three performance and cost issues in distributed PM file system**
- **We propose a novel asymmetric NVM file system architecture**
- **Ethane consists of a control plane and a data plane**
  - **Shared-log-based control-plane FS**
  - **Access-disentangled data-plane FS**
- **Ethane improves file system performance with low hardware costs**


README

Ethane: An Asymmetric File System for Disaggregated Persistent Memory

1. Overview

This repository contains the source code, setup utilities, and test scripts of *Ethane: An Asymmetric File System for Disaggregated Persistent Memory*.

https://github.com/miaogecm/Ethane.git

USENIX ATC '24

Ethane: An Asymmetric File System for Disaggregated Persistent Memory

Miao Cai, Junru Shen, Baoliu Ye

miaocai@nuaa.edu.cn

Thanks!

usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION