Semiconductor Research Corporation

A Systematic Approach to Scalably Saturate PM Bandwidth

Soujanya Ponnapalli, Sekwon Lee, Rohan Kadekodi, Vijay Chidambaram

soujanya.ponnapalli@utexas.edu

TEXAS
The University of Texas at Austin

Task/Theme: 3135.003

Persistent Memory

Persistent
Memory (PM)

Persistent Memory

Persistent Memory (PM)

Non-volatile

Persistent Memory

	Persistent Memory (PM)
	
Non-volatile	✓
Byte-addressable	✓

Persistent Memory

	DRAM	Persistent Memory (PM)	Solid-state Drive (SSD)
			
Non-volatile	✗	✓	✓
Byte-addressable	✓	✓	✗

Persistent Memory

	DRAM	Persistent Memory (PM)	Solid-state Drive (SSD)
			
Latency	< 100ns (1x)	2—3x	1000x
Non-volatile	✗	✓	✓
Byte-addressable	✓	✓	✗

Persistent Memory

	DRAM	Persistent Memory (PM)	Solid-state Drive (SSD)
			
Latency	< 100ns (1x)	2—3x	1000x
Non-volatile	✗	✓	✓
Byte-addressable	✓	✓	✗

PM offers durability and byte-addressability at low latency

Persistent Memory

Durable writes

Byte-addressable

High capacity and density

PM is a promising building block for large-scale kv-stores

Persistent Memory

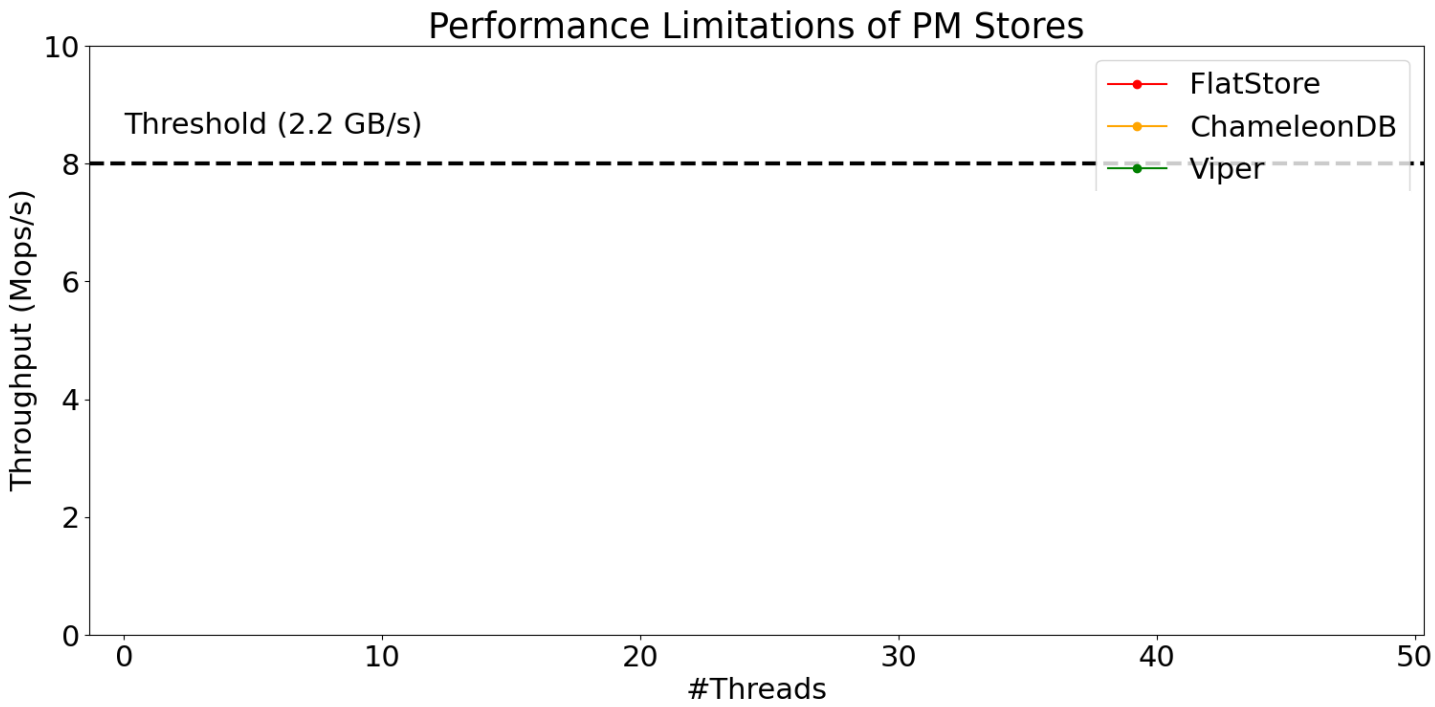
Durable writes	Higher cost relative to SSD
Byte-addressable	Lower bandwidth relative to DRAM
High capacity and density	Nuanced Performance Characteristics [1,2,3]*

*[1] Yang, Jian, et al. "**An empirical guide to the behavior and use of scalable persistent memory.**" 18th USENIX Conference on File and Storage Technologies (FAST 20). 2020.*
*[2] Daase, Björn, et al. "**Maximizing persistent memory bandwidth utilization for OLAP workloads**." Proceedings of the 2021 International Conference on Management of Data. 2021.*
*[3] Izraelevitz, Joseph, et al. "**Basic performance measurements of the intel optane DC persistent memory module**." arXiv preprint arXiv:1903.05714 (2019).*

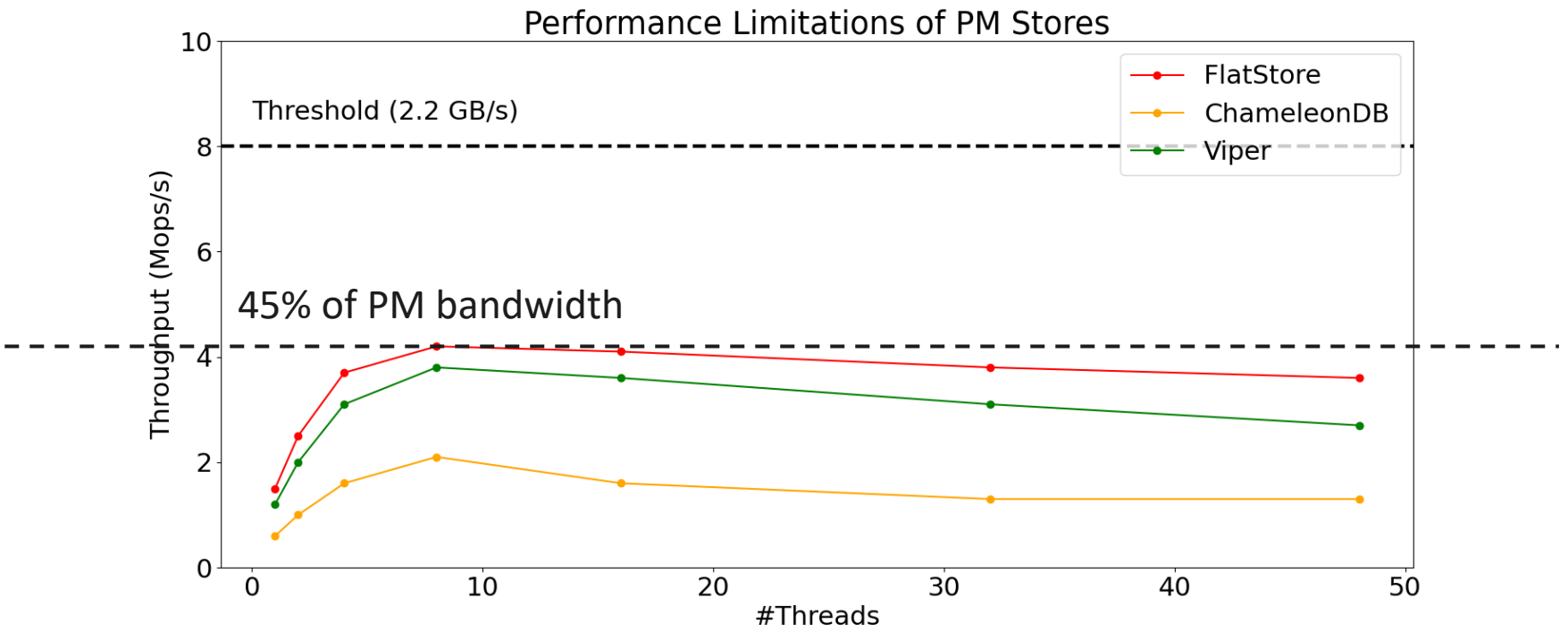
PM Key-Value Stores

- State-of-the-art PM key-value stores
  - FlatStore
  - Viper
  - ChameleonDB
- Underutilize available PM bandwidth
- Low throughput
- Poor scalability

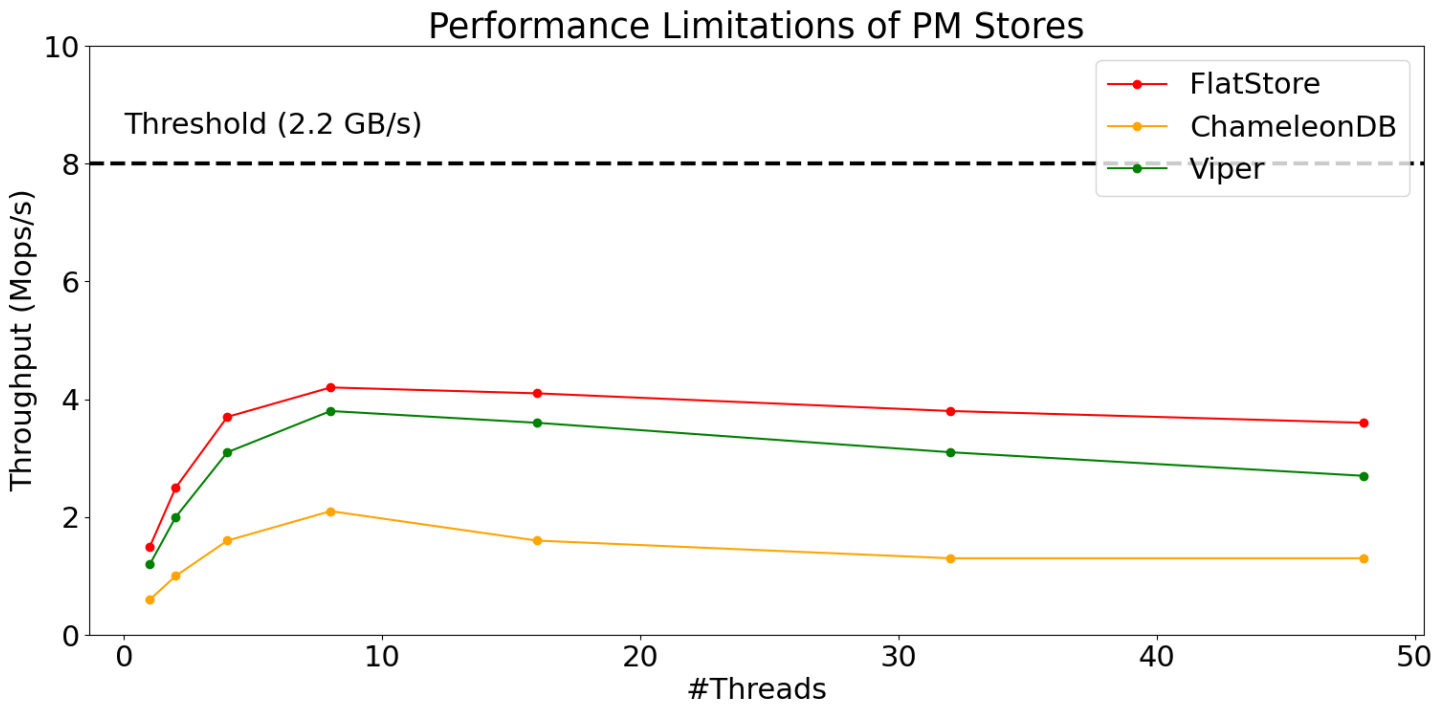
PM Key-Value Stores



PM Key-Value Stores



PM Key-Value Stores



PM stores suffer from low throughput and poor scalability

Motivation: Efficient PM Stores

- Write-optimized key-value stores are popular storage engines for distributed systems
- Compute Express Link (CXL) and PM expansion: Scalable PM stores
- Disaggregated PM: Improving the utilization of resources is crucial for cost efficient datacenters

High bandwidth utilization is crucial for efficient PM stores

Scalably Saturating PM Bandwidth

- TyrantKV improves PM bandwidth utilization by retaining fine-grained control over all PM accesses
- In-Direct access for applications
- Manages PM as individual NVDIMMs
- Leverages multiple media to avoid overloading PM
- Obtains 2.5 – 5x higher throughput than state-of-the-art PM stores on standard YCSB workloads

Outline

- I/O bottlenecks in PM stores
  - Strawman solution and drawbacks
- Design and architecture of TyrantKV
- Performance and Scalability of TyrantKV

PM Stores: Design for Low Latency

PM Stores: Design for Low Latency

- Direct-access for applications
  - Application threads directly perform I/O on NVDIMMs
  - PM stores do not control the #threads concurrent threads reading or writing to NVDIMMs

PM Stores: Design for Low Latency

- Direct-access for applications
  - Application threads directly perform I/O on NVDIMMs
  - PM stores do not control the #threads concurrent threads reading or writing to NVDIMMs
- Hardware manages individual NVDIMMs
  - A large file round-robins at 4kB granularity across available NVDIMMs per NUMA node
  - Memory controllers place data on NVDIMMs

PM Stores: Design for Low Latency

- Direct-access for applications
  - Application threads directly perform I/O on NVDIMMs
  - PM stores do not control the #threads concurrent threads reading or writing to NVDIMMs
- Hardware manages individual NVDIMMs
  - A large file round-robins at 4kB granularity across available NVDIMMs per NUMA node
  - Memory controllers place data on NVDIMMs

PM stores let go off fine-grained control over PM accesses

PM Analysis: Insights

- Bounded thread scaling
- Non-interleaved NVDIMMs
- Concurrent reads and writes to NVDIMMs

PM Analysis: Insights

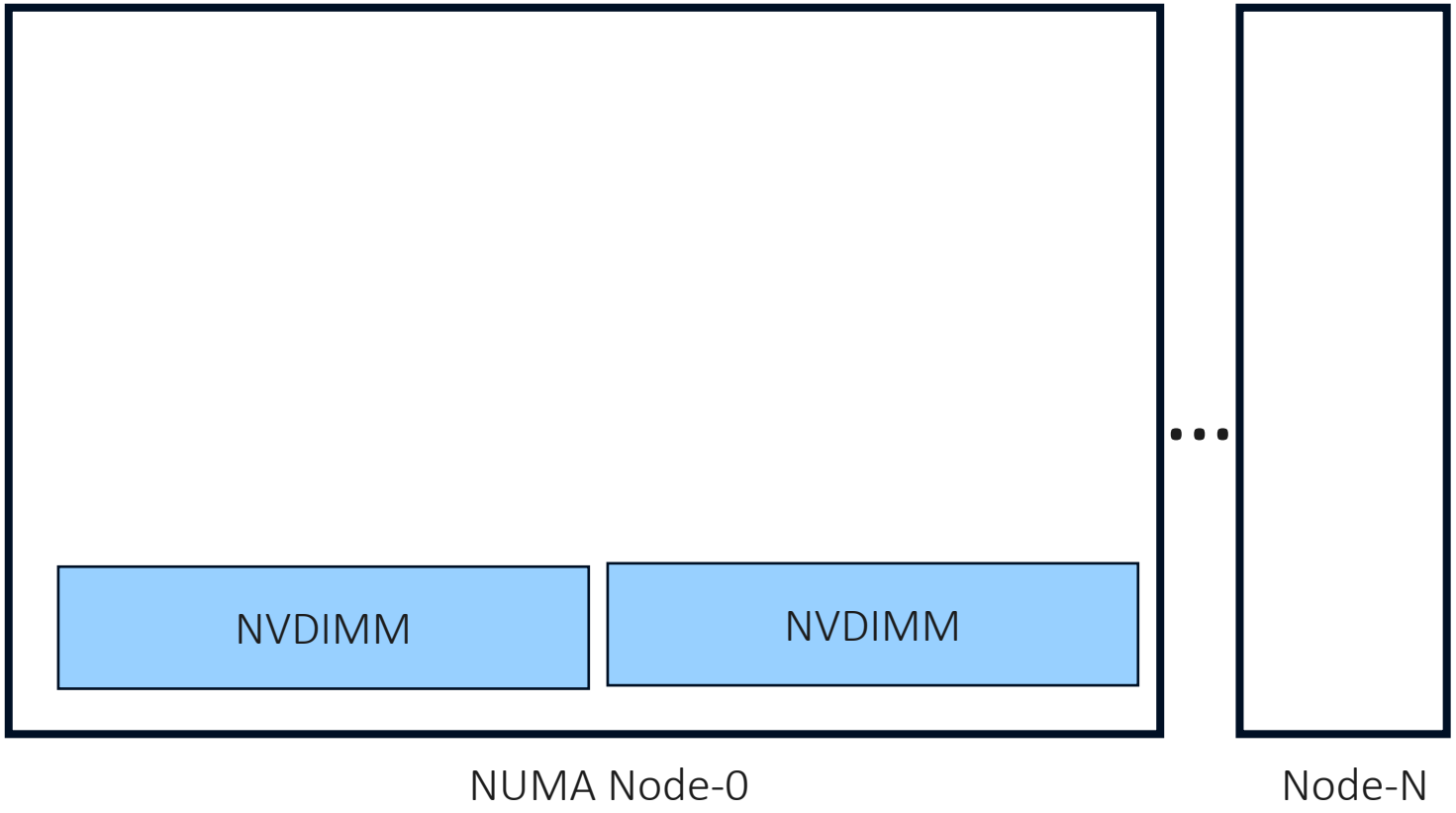
- Bounded thread scaling
  - Scaling threads beyond a threshold hurts PM bandwidth
- Non-interleaved NVDIMMs
  - To balance the load across NVDIMMs and avoid overwhelming PM controllers
- Concurrent reads and writes to NVDIMMs
  - Provides peak throughput with lower #threads *([3])

PM requires fine-grained control to achieve high utilization

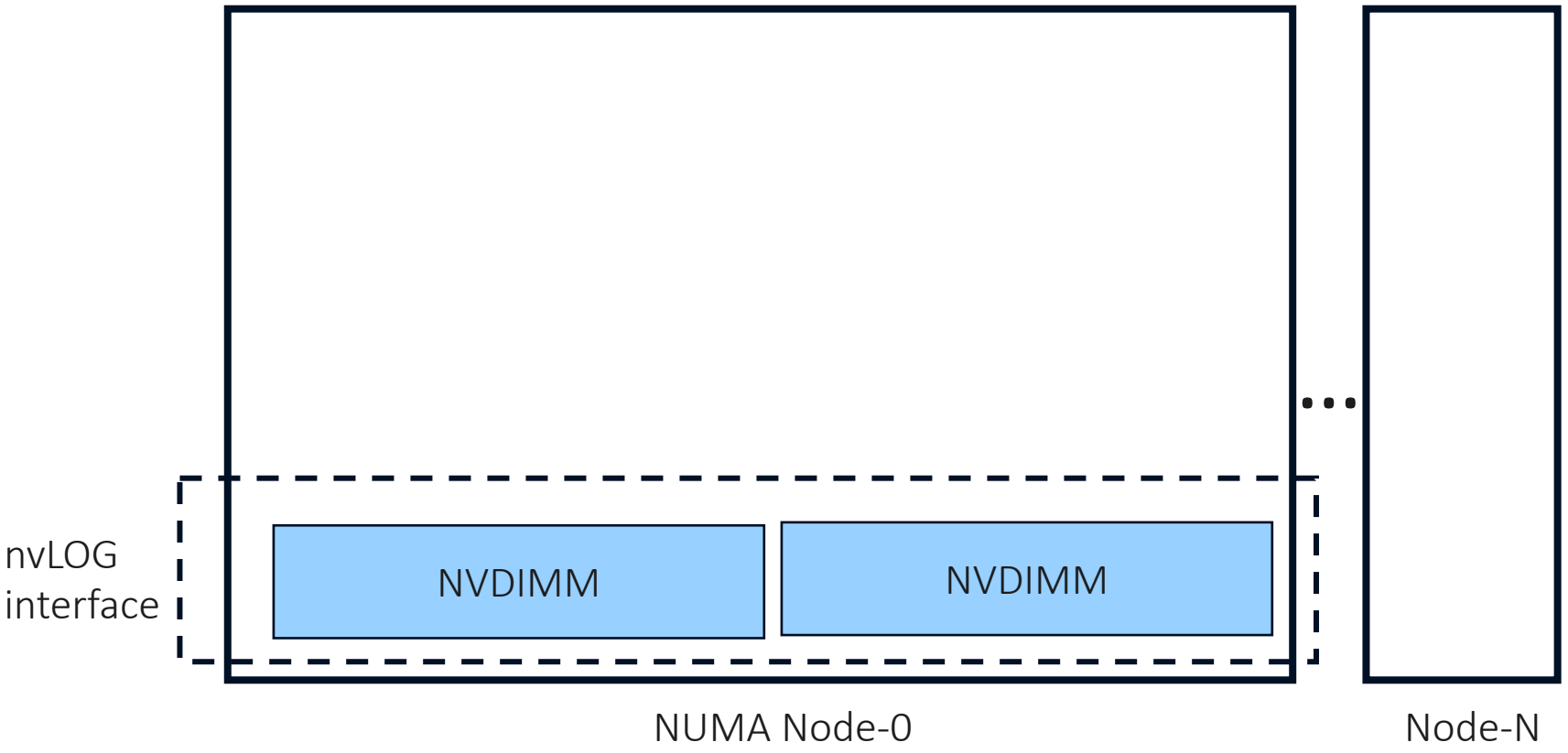
I/O Bottlenecks: PM-agnostic Design

- Direct-access for applications
  - Applications determine the #threads per NVDIMM
- Hardware-managed NVDIMMs
  - Memory-controllers determine data placement
- Overheads from PM agnostic design choices
  - Strawman solution: PM-aware FS
  - Overheads from system calls

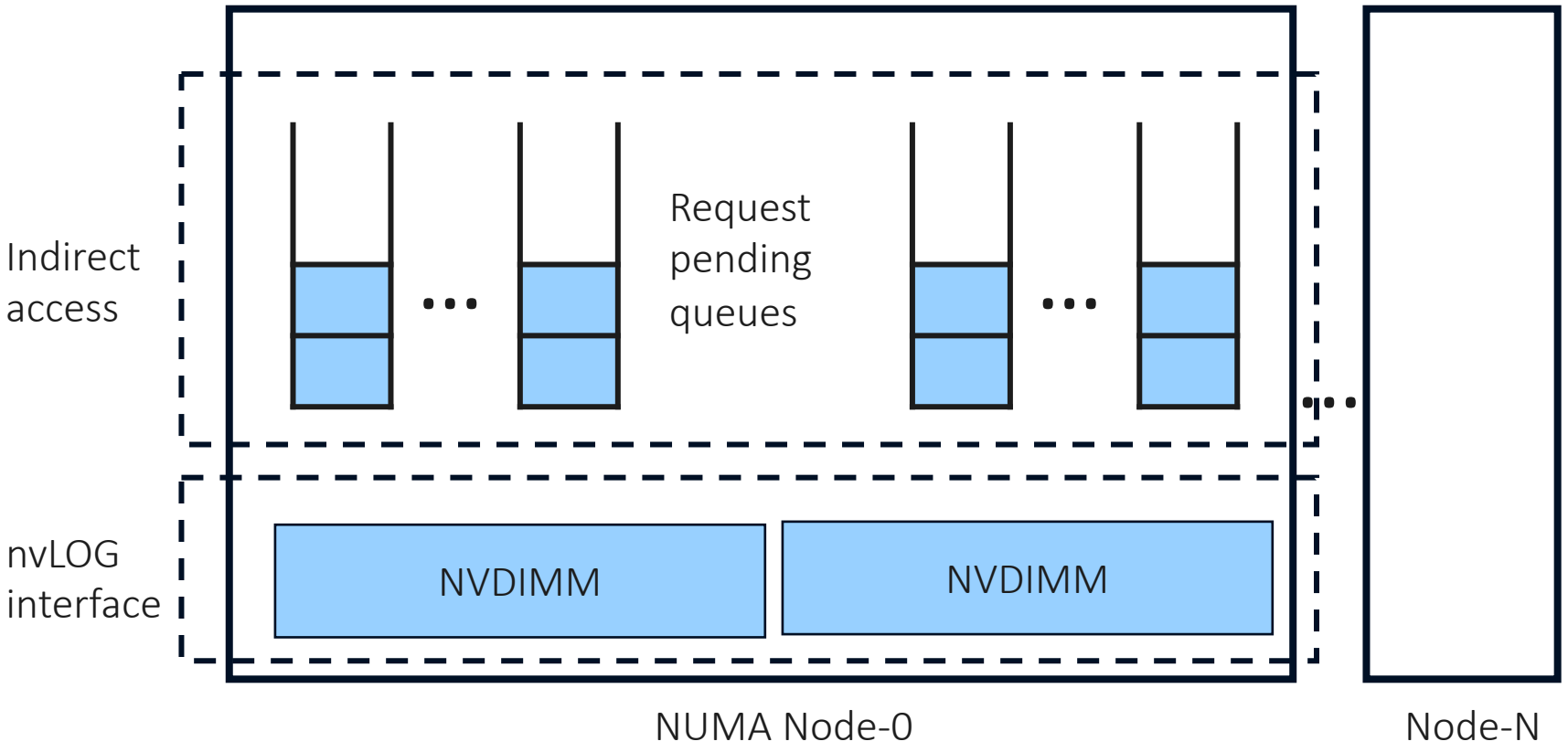
TyrantKV



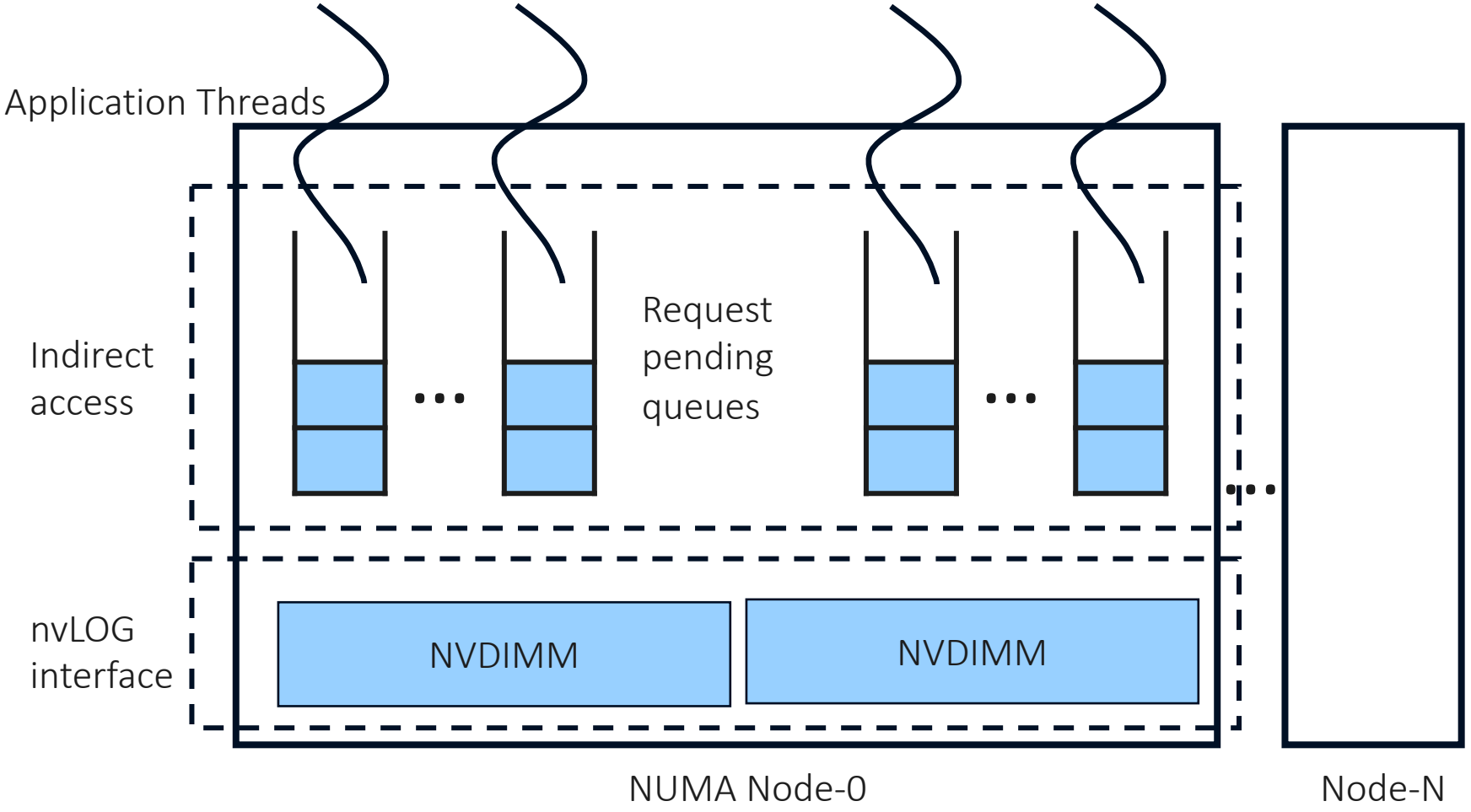
TyrantKV: nvLOG interface



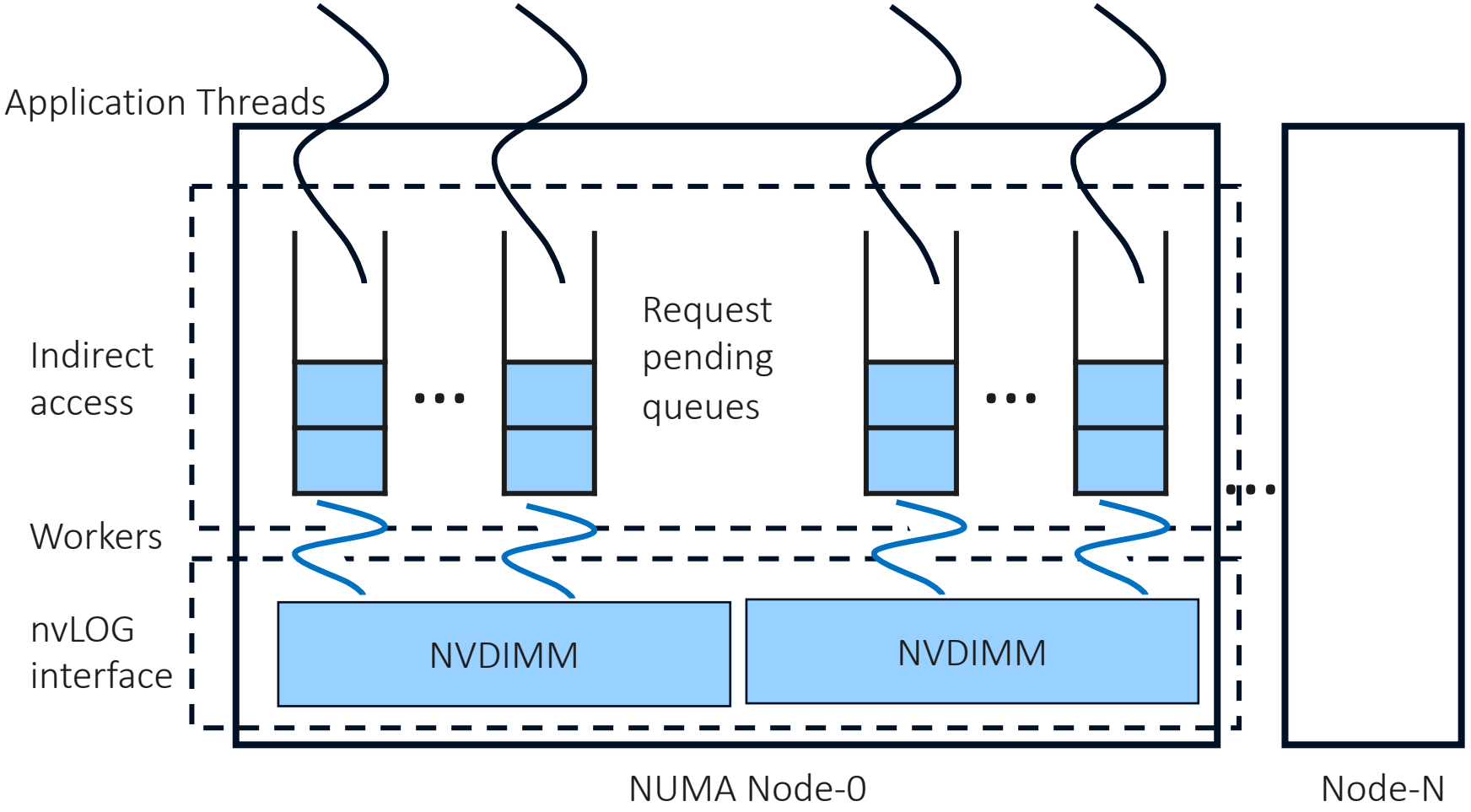
TyrantKV: Indirect Access



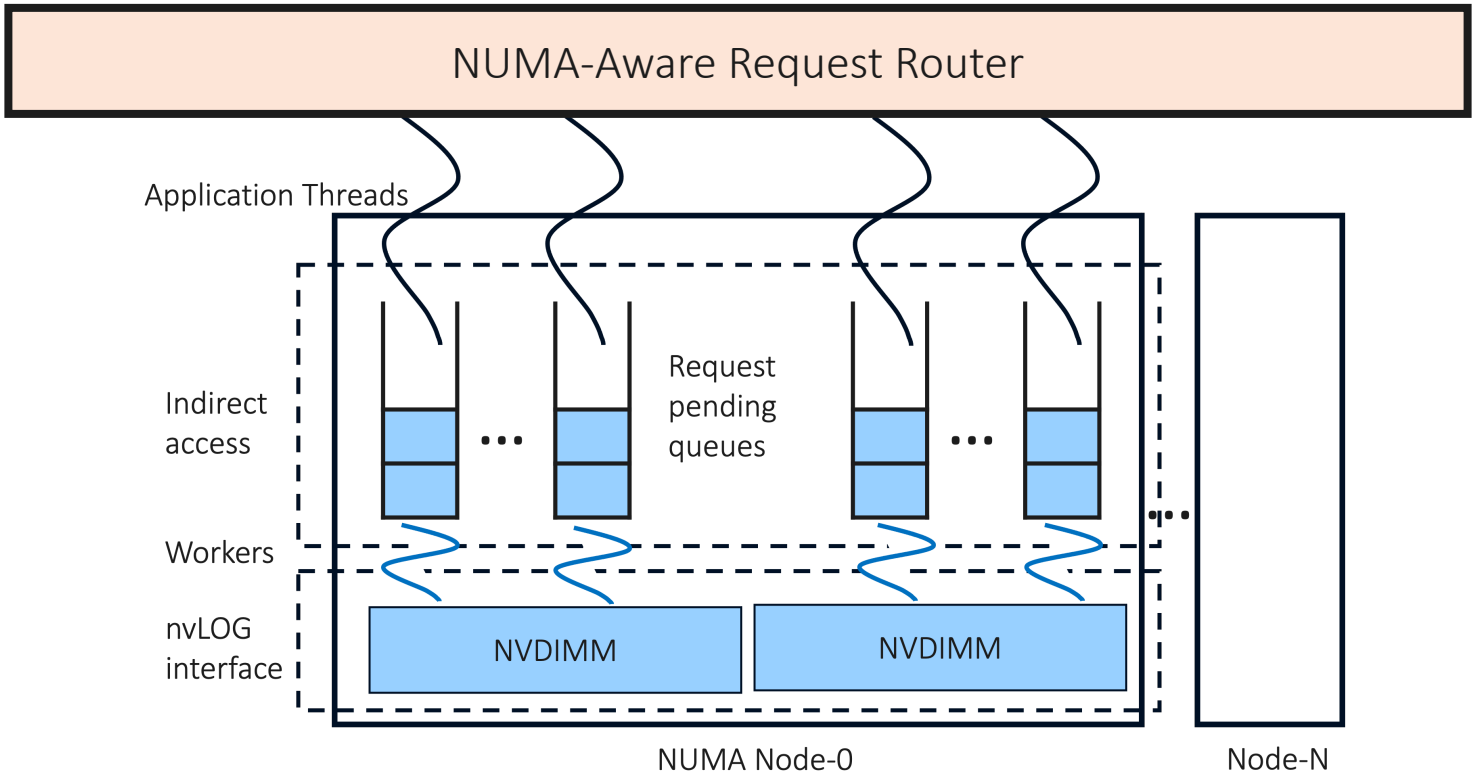
TyrantKV: Indirect Access



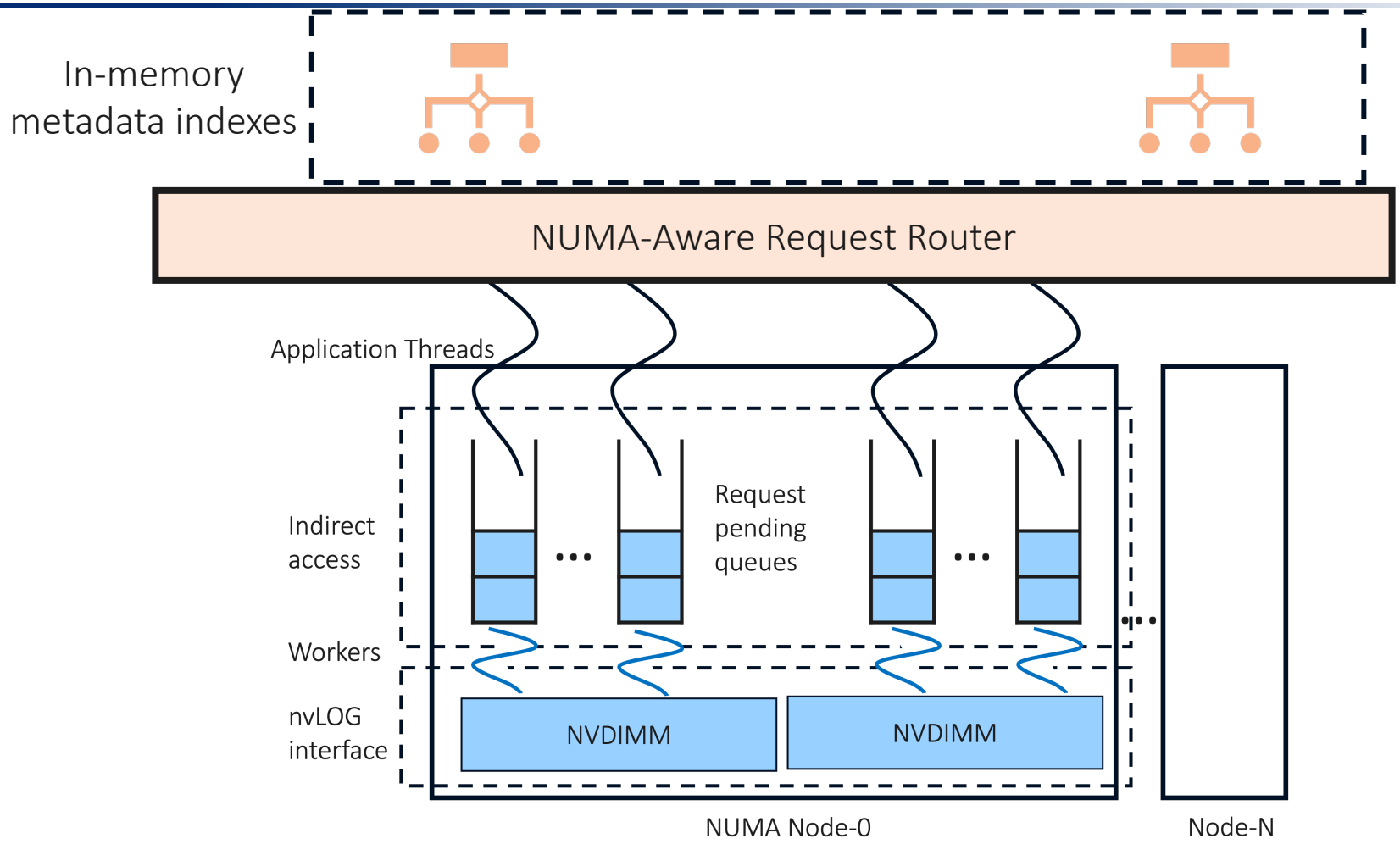
TyrantKV: Indirect Access



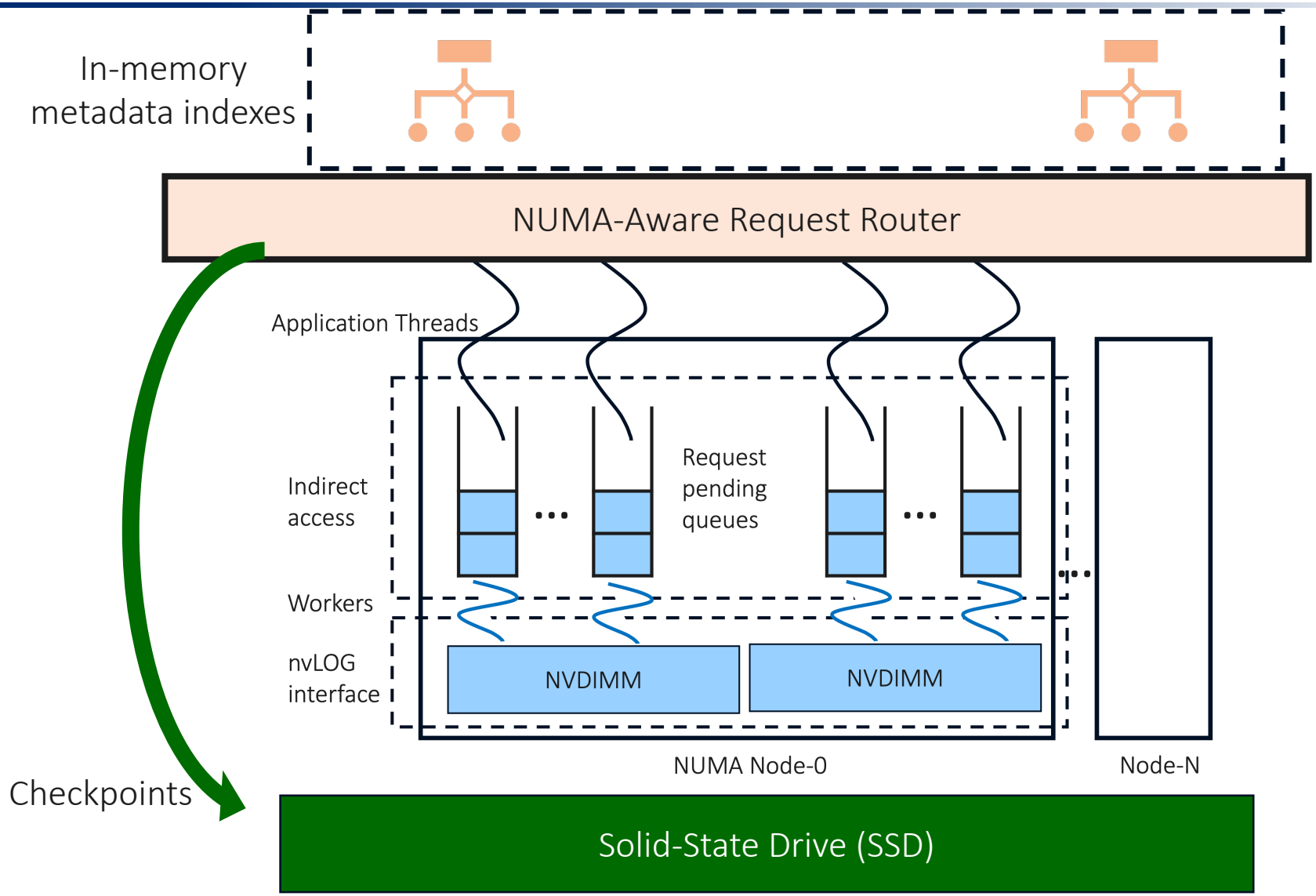
TyrantKV: Multiple Media



TyrantKV: Multiple Media



TyrantKV: Multiple Media



TyrantKV: Fine-Grained Control

Avoids I/O overheads from PM

- nvLOGs: DIMM-aware data placement
- Indirect-access: Bounded thread scaling
- Multi-media design: Avoids overloading PM

Evaluation: Setup

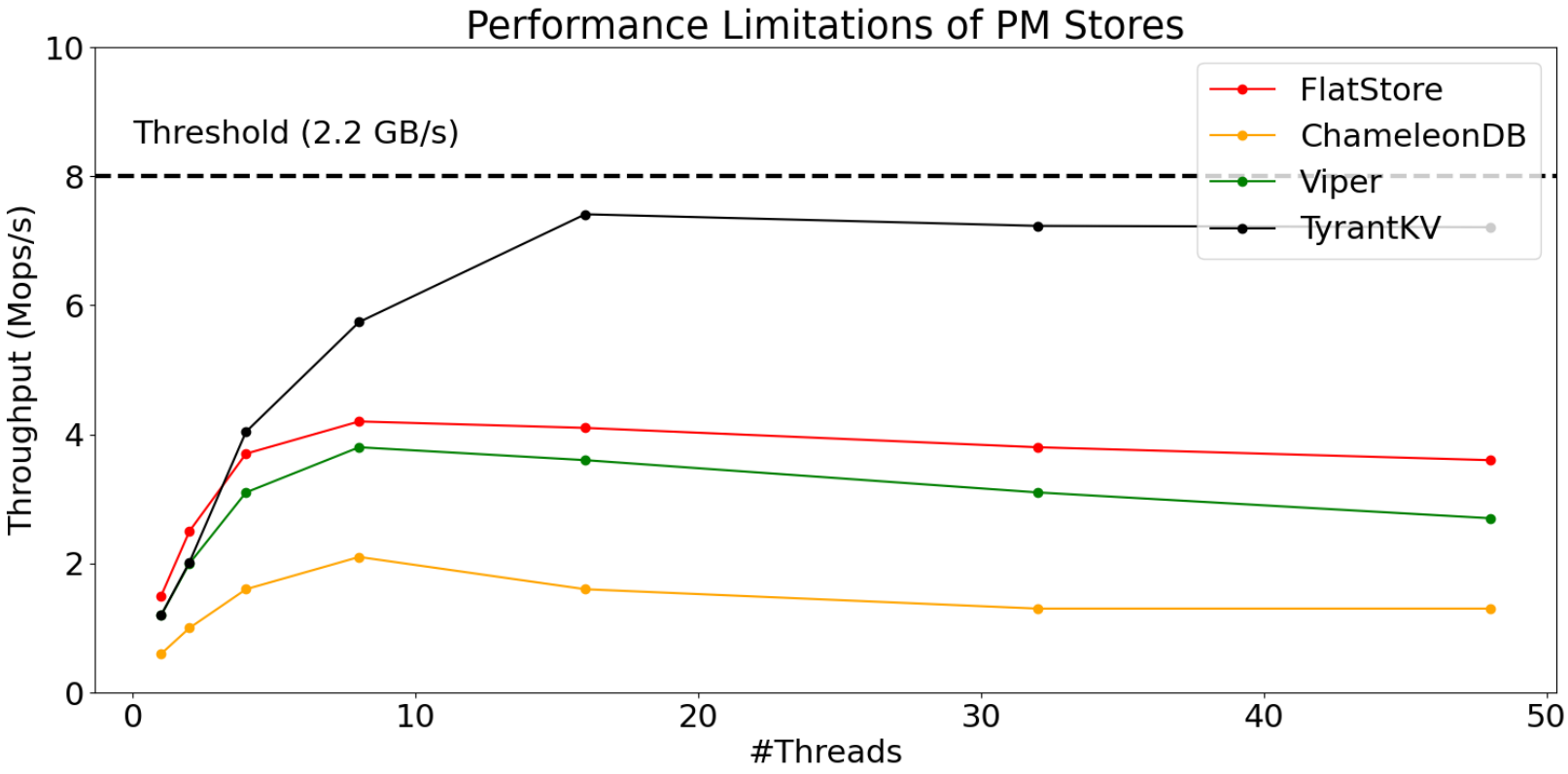
Experimental Setup

- compute_nvdimm server: Chameleon cloud
  - 4-socket NUMA server
  - 56 cores per CPU
  - 3 TB PM

Yahoo-Cloud Serving Benchmark (YCSB)

- 12GB Load-A workload
- 8B keys, 256B values

High Throughput



Scalability

- State-of-the-art PM key-value stores
  - FlatStore, Viper, and ChameleonDB
- Achieves up to 88% write bandwidth utilization
- Increasing the number of NVDIMMs from 1 to 4, TyrantKV's throughput scales by 3.9x

Evaluation: Summary

- State-of-the-art PM key-value stores
  - FlatStore, Viper, and ChameleonDB
- Achieves up to 88% write bandwidth utilization
- Increasing the number of NVDIMMs from 1 to 4, TyrantKV's throughput scales by 3.9x
- On YCSB workloads, TyrantKV achieves higher throughput and outperforms PM stores by 2.5 – 5x

Future work

- Disaggregated PM stores
  - CXL PM expansion
  - PM bandwidth utilization

Summary

cs.utexas.edu/~soujanya
soujanya.ponnapalli@utexas.edu

- Achieving high, scalable PM bandwidth utilization is crucial for cost efficiency and practical adoption of PM
- Fine-grained control over all accesses is fundamental for obtaining high, scalable PM bandwidth utilization
- TyrantKV retains fine-grained control over all PM accesses and saturates 86% of PM write bandwidth
- TyrantKV has 2—5x higher throughput to PM stores on YCSB and scales to NVDIMMs across NUMA nodes