

Finding NEMO:
Nimble and Expressive Memory Observability

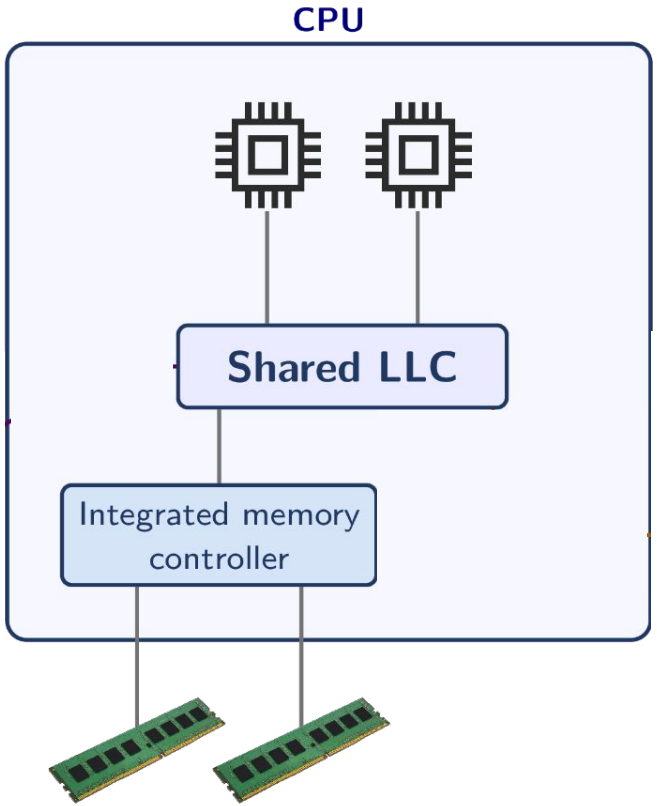
Shihang (Vic) Li*, Matthew Giordano*, Tushar Garg, Rohan Kadekodi, Daniel S. Berger, Baris Kasikci, Thomas Anderson, Simon Peter

* Equal Contribution



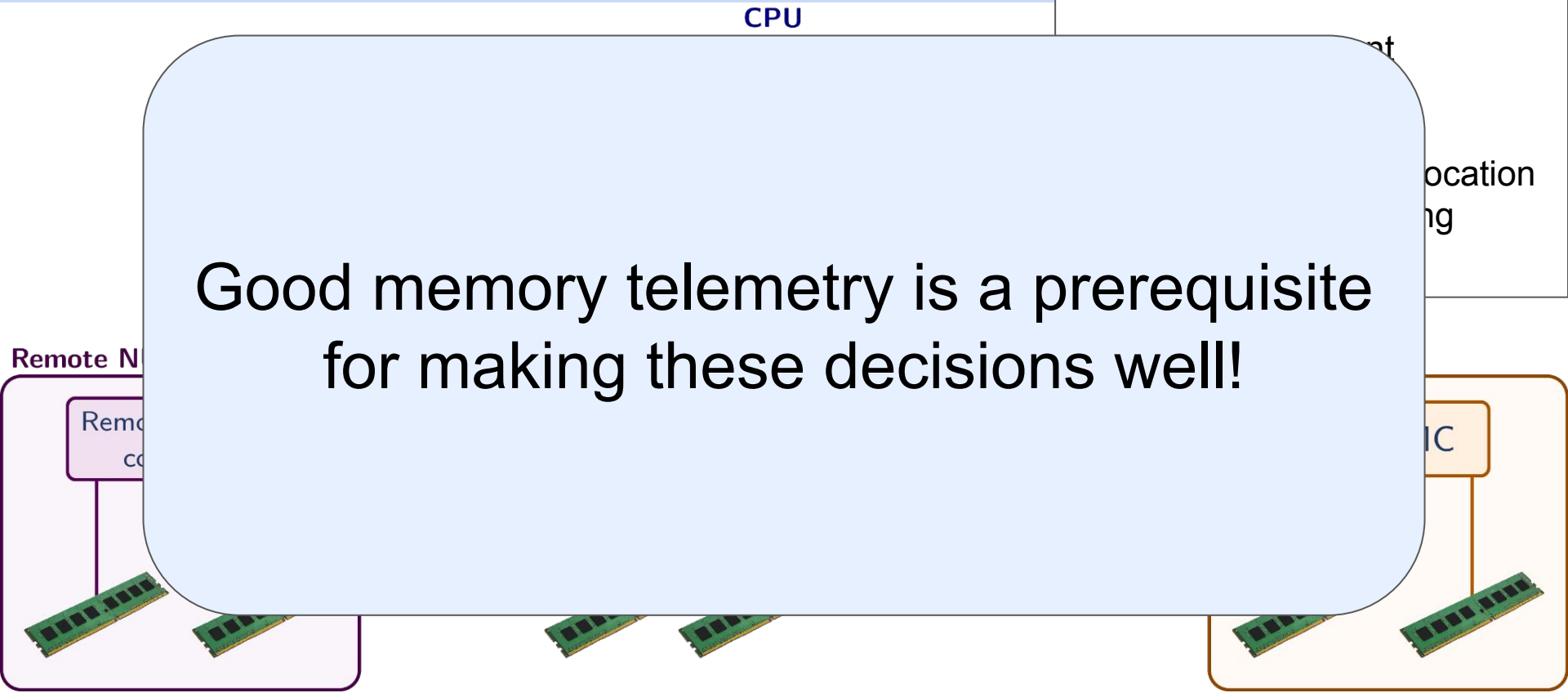


Memory is Complex



Memory is Complex

Memory Management Responsibilities



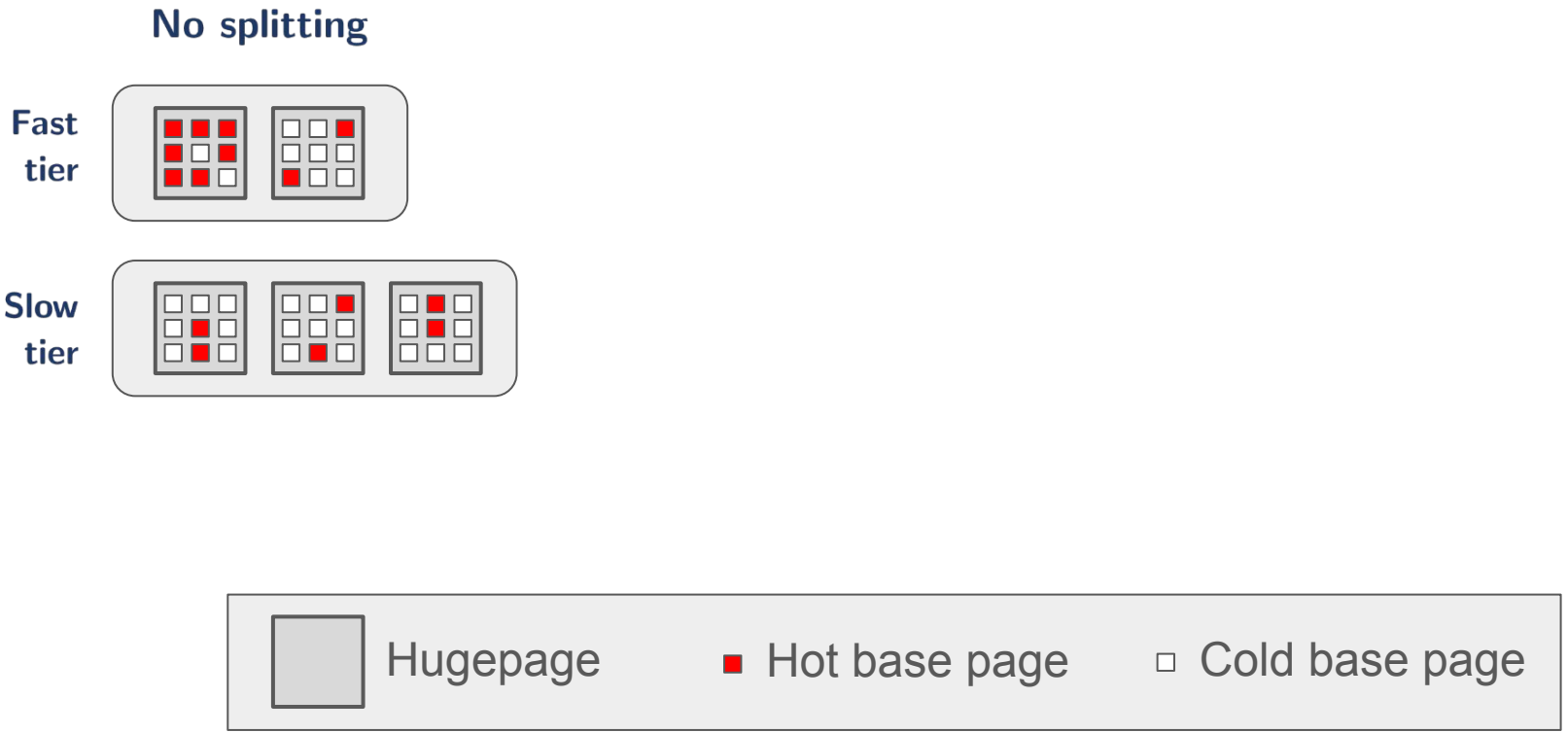
Good memory telemetry is a prerequisite for making these decisions well!

Telemetry Mechanisms Today Tradeoff Fidelity & Overhead

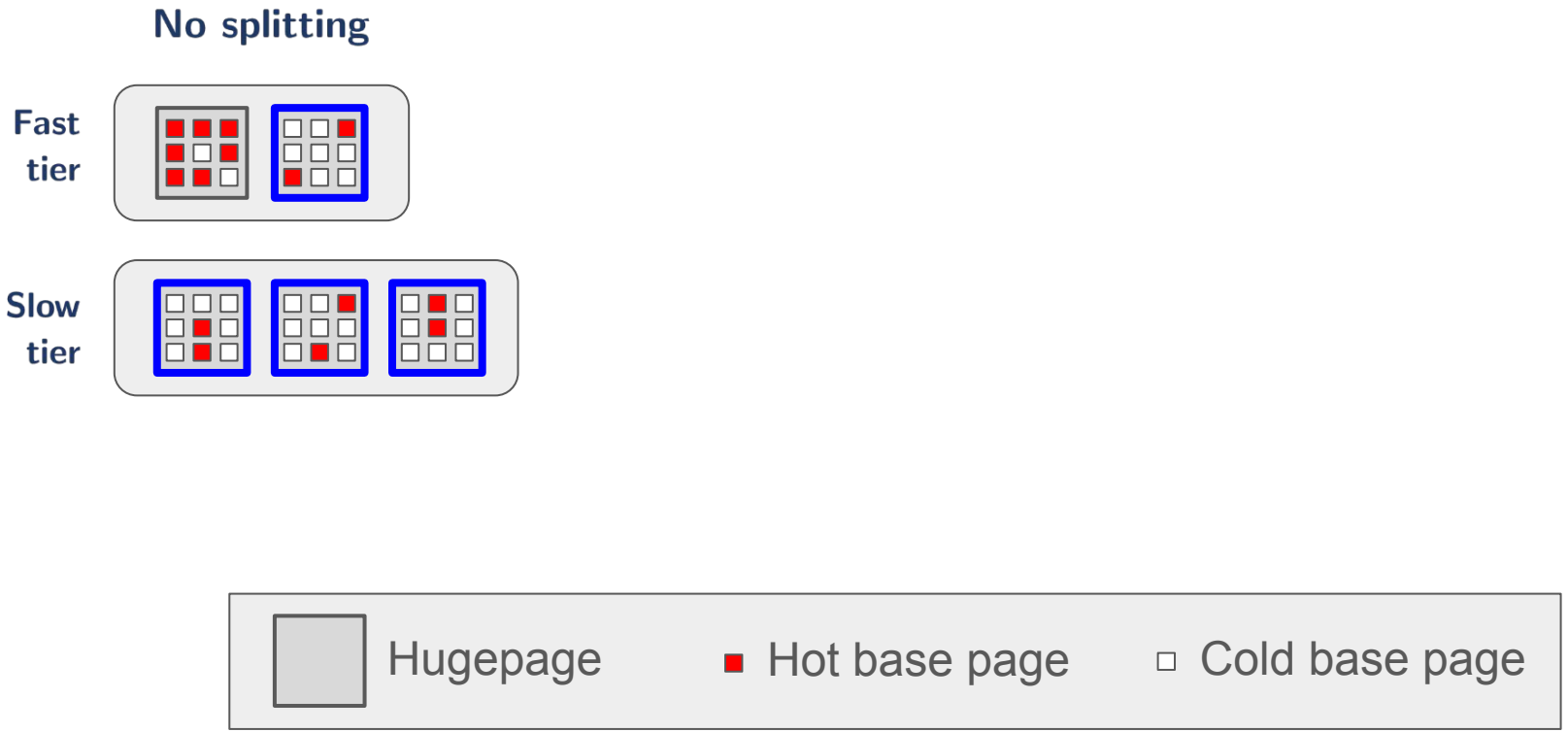
- **Soft page faults** (unmap pages)
- **Page Table Scanning** (access / dirty bits)
- **Load/Store Sampling** (via PEBS/IBS)

	High Sample Rate	Low Sample Rate
Flexibility	✅	✅
Fidelity	✅	❌
Overhead	❌	✅

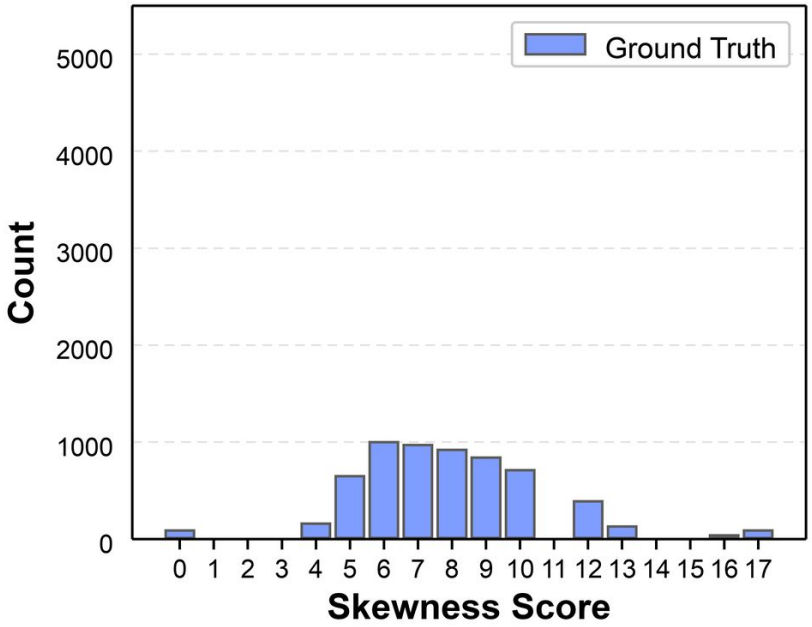
Example: Hugepage Splitting in Tiered Memory

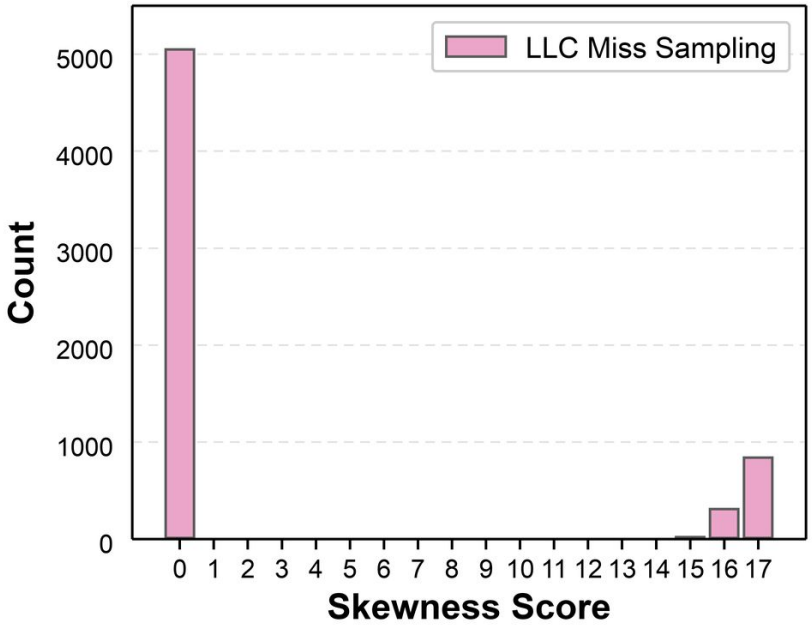


Example: Hugepage Splitting in Tiered Memory



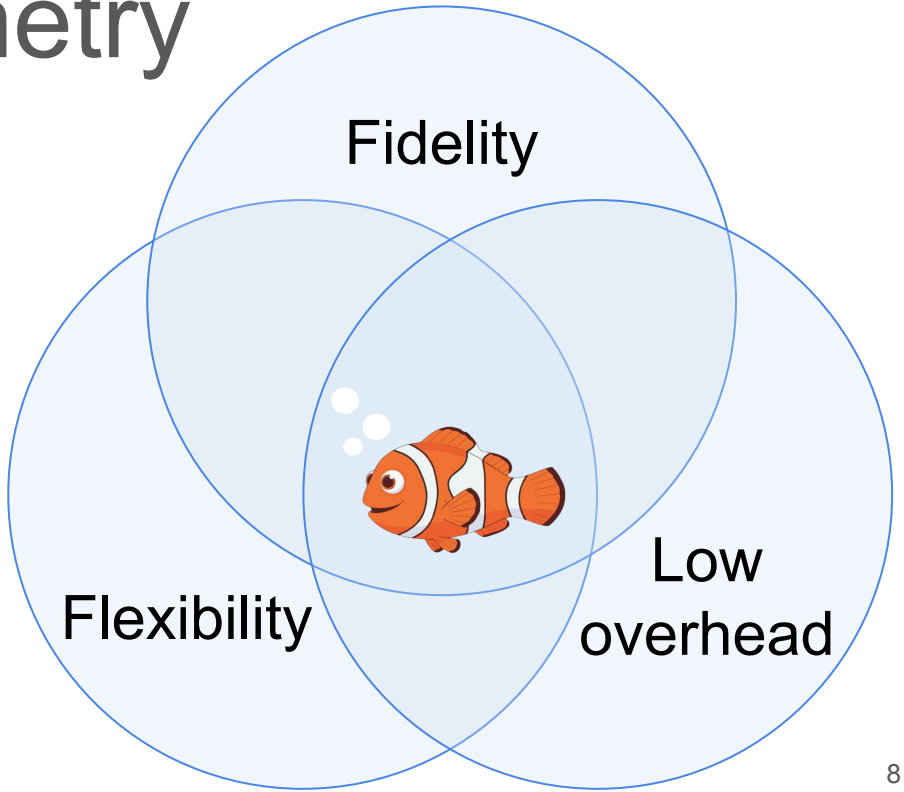
Sampled Telemetry Misleads the OS



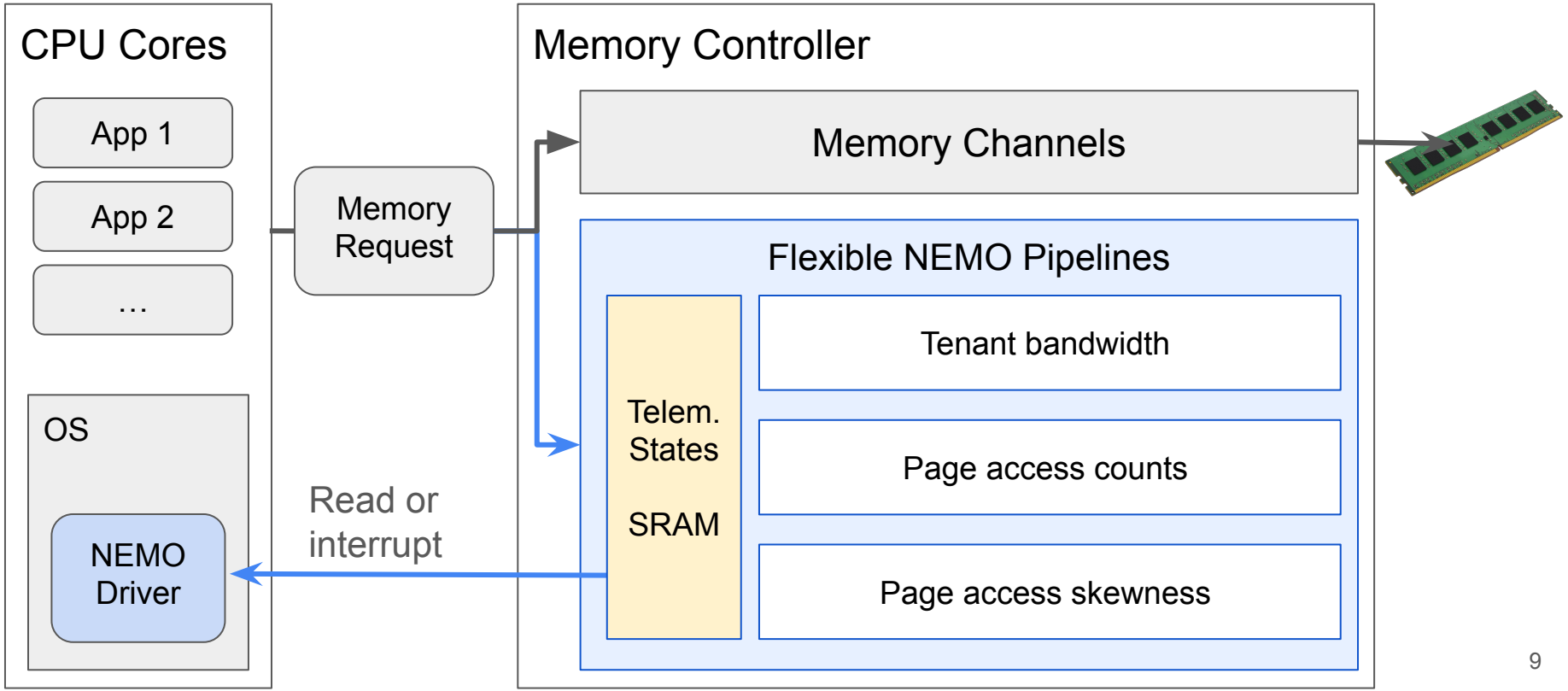


Skewness scores as defined by Memtis (SOSP ‘23)

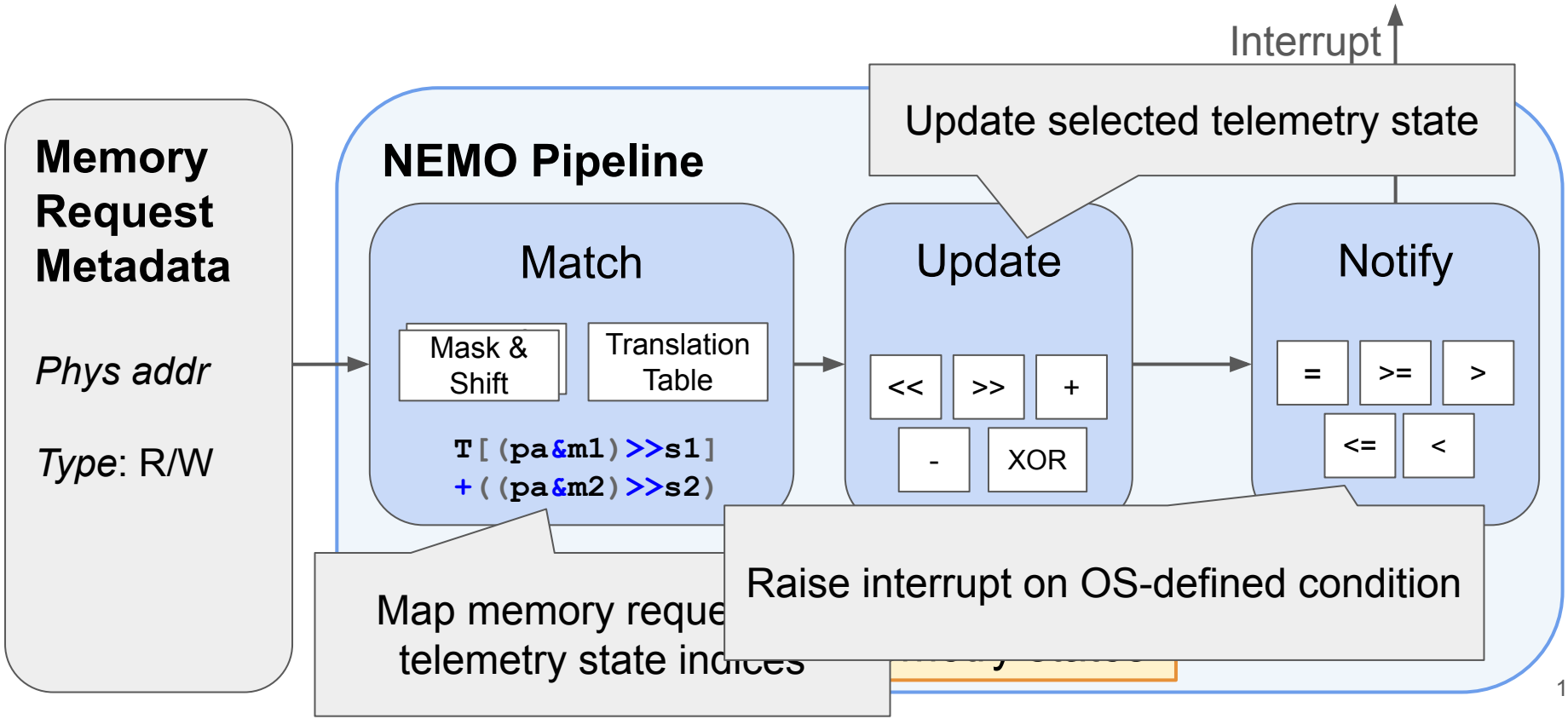
Can Memory Telemetry Deliver Fidelity, Flexibility, and Low Overhead?



NEMO: Nimble and Expressive Memory Observability



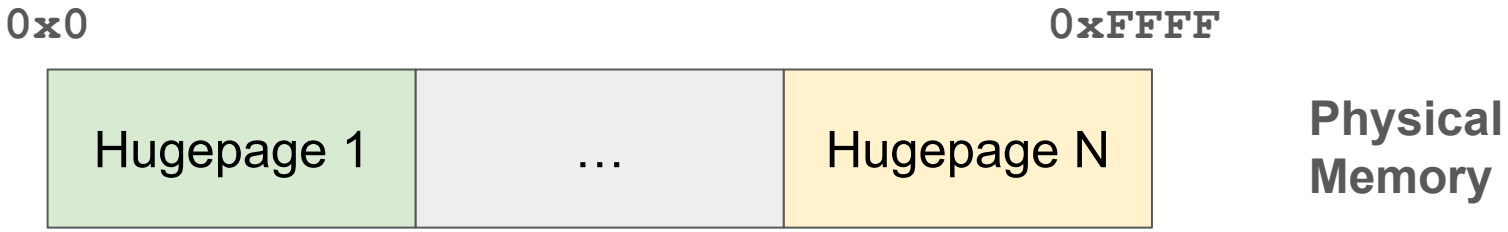
Flexible NEMO Pipeline Architecture



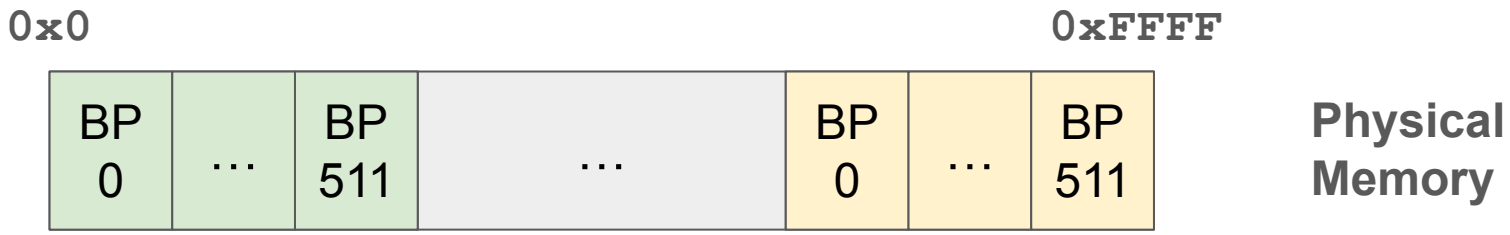
Example: Skewness in Hugepage



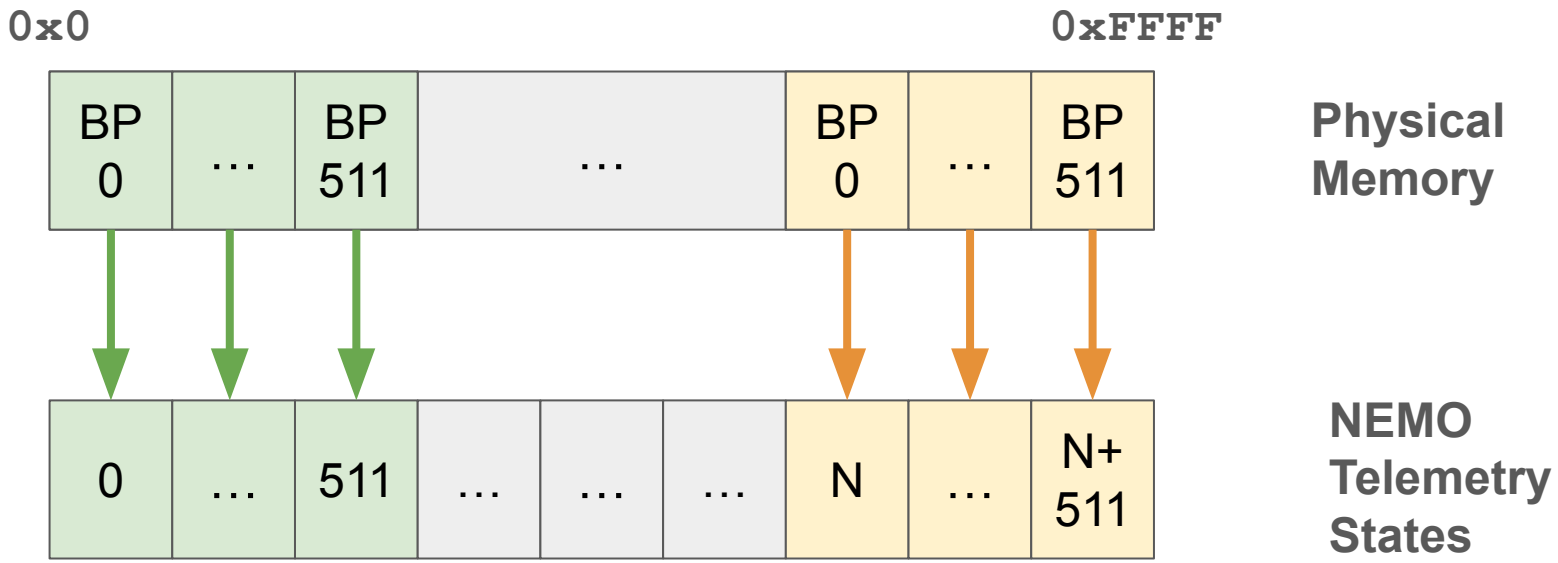
Example: Skewness in Hugepage



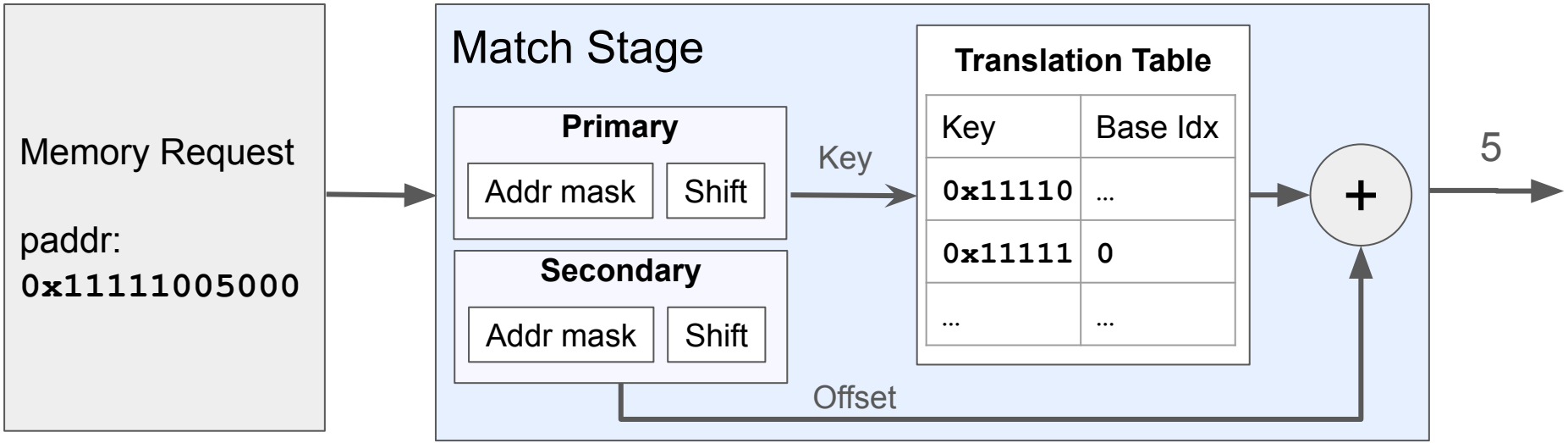
Example: Skewness in Hugepage



Example: Skewness in Hugepage



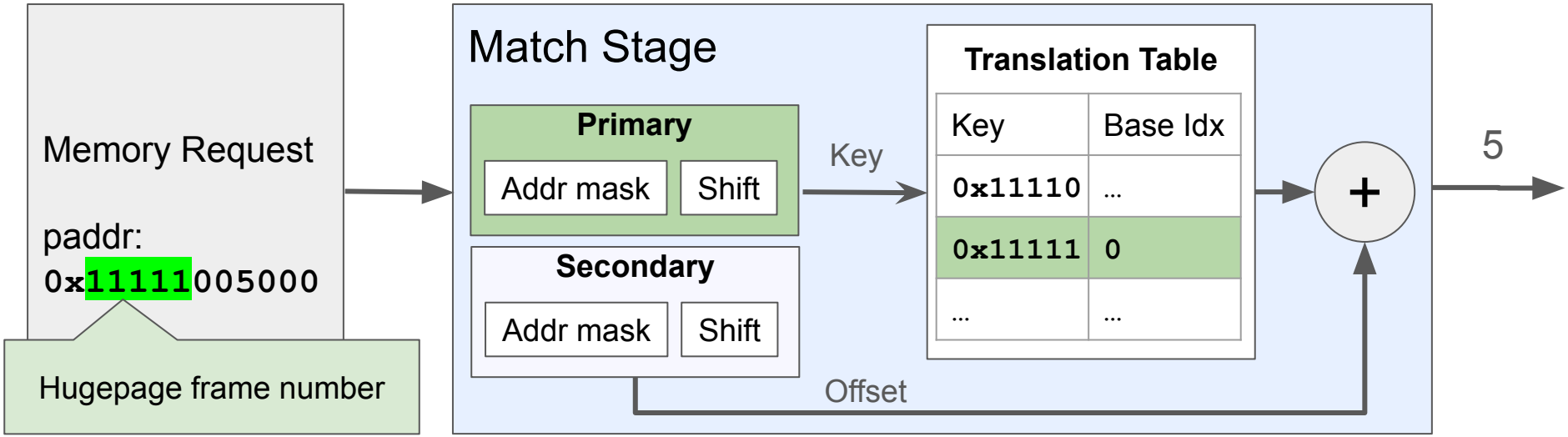
Match Stage Example



Telemetry States

Index	0	1	2	3	4	5	…	509	510	511	…
Value											…

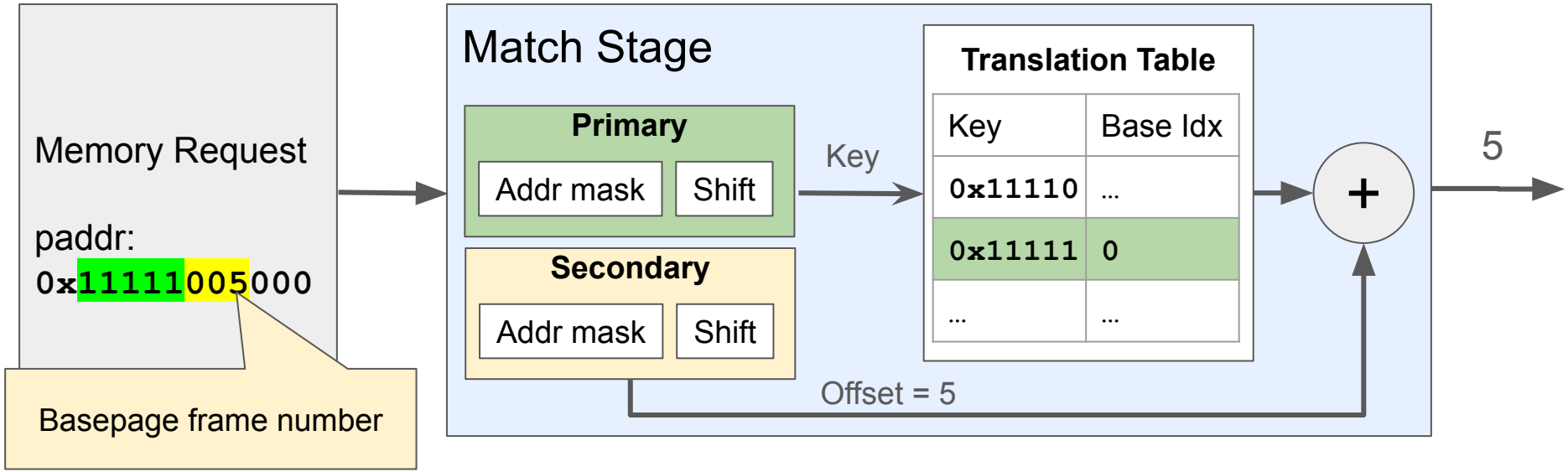
Match Stage Example



Telemetry States

Index	0	1	2	3	4	5	…	509	510	511	…
Value											…

Match Stage Example

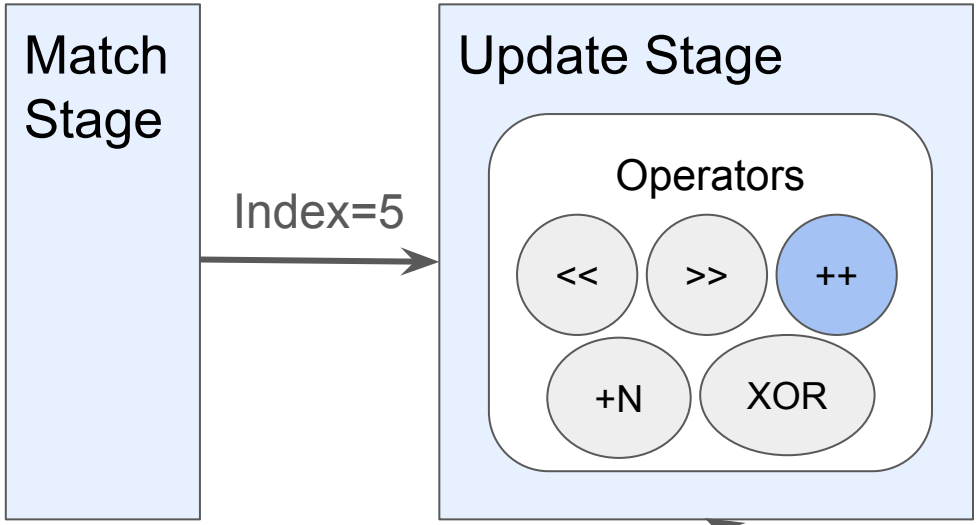


Telemetry States

Add offset

Index	0	1	2	3	4	5	…	509	510	511	…
Value											…

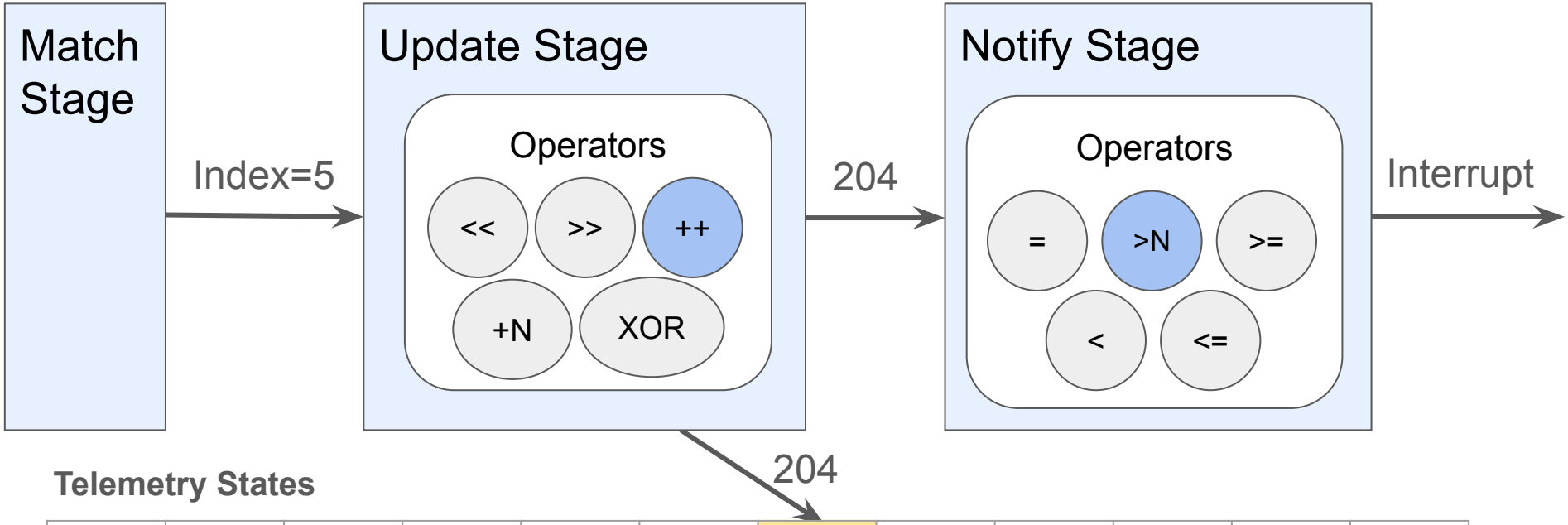
Update And Notify Stage



Telemetry States

Index	0	1	2	3	4	5	…	509	510	511	…
Value						203					…

Update And Notify Stage



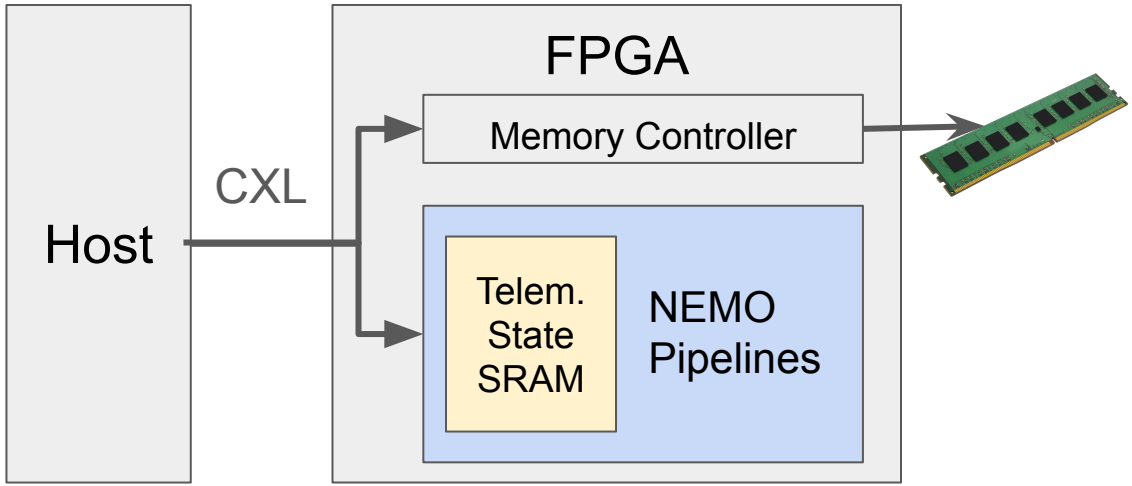
Telemetry States

Index	0	1	2	3	4	5	…	509	510	511	…
Value						204					…

CXL Memory Controller Prototype on FPGA

- Altera Agilex 7 I-Series Dev Kit
- CXL 2.0 Type-3 hard IP
- 16 GiB onboard DDR

	Full Bitstream	NEMO
Logic	17%	4.8%
Regs	9.3%	1.7%
SRAMs	14%	4.4% (1.4MiB)

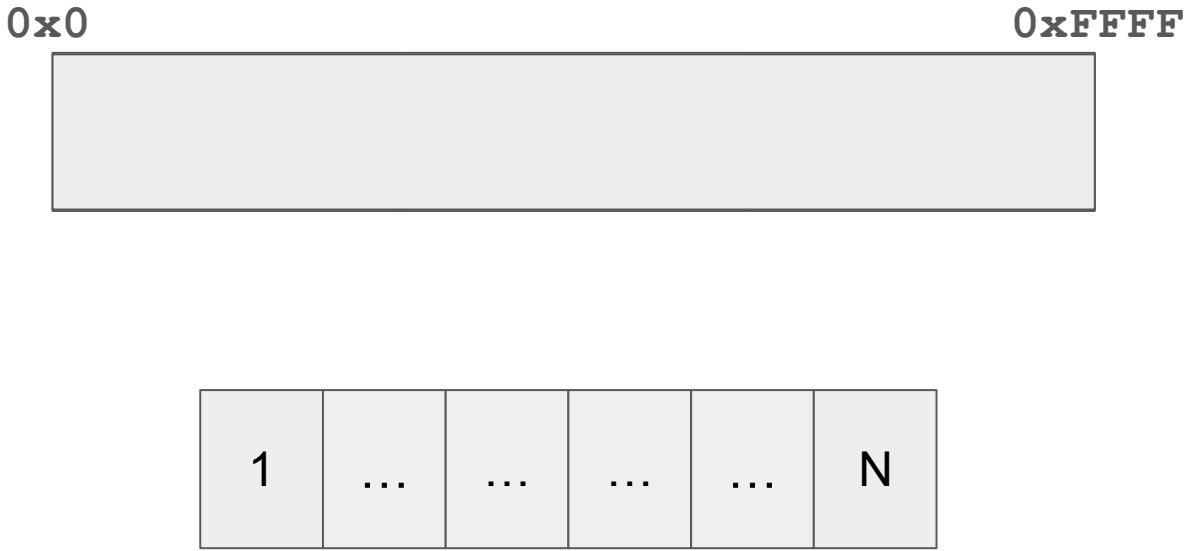




Evaluation Use Cases

1. Detect Hot Set Shifts
2. Split Partially Hot Hugepages
3. Performance Isolation Across Tenants

Use Case 1: Hot Set Tracking



Use Case 1: Hot Set Tracking

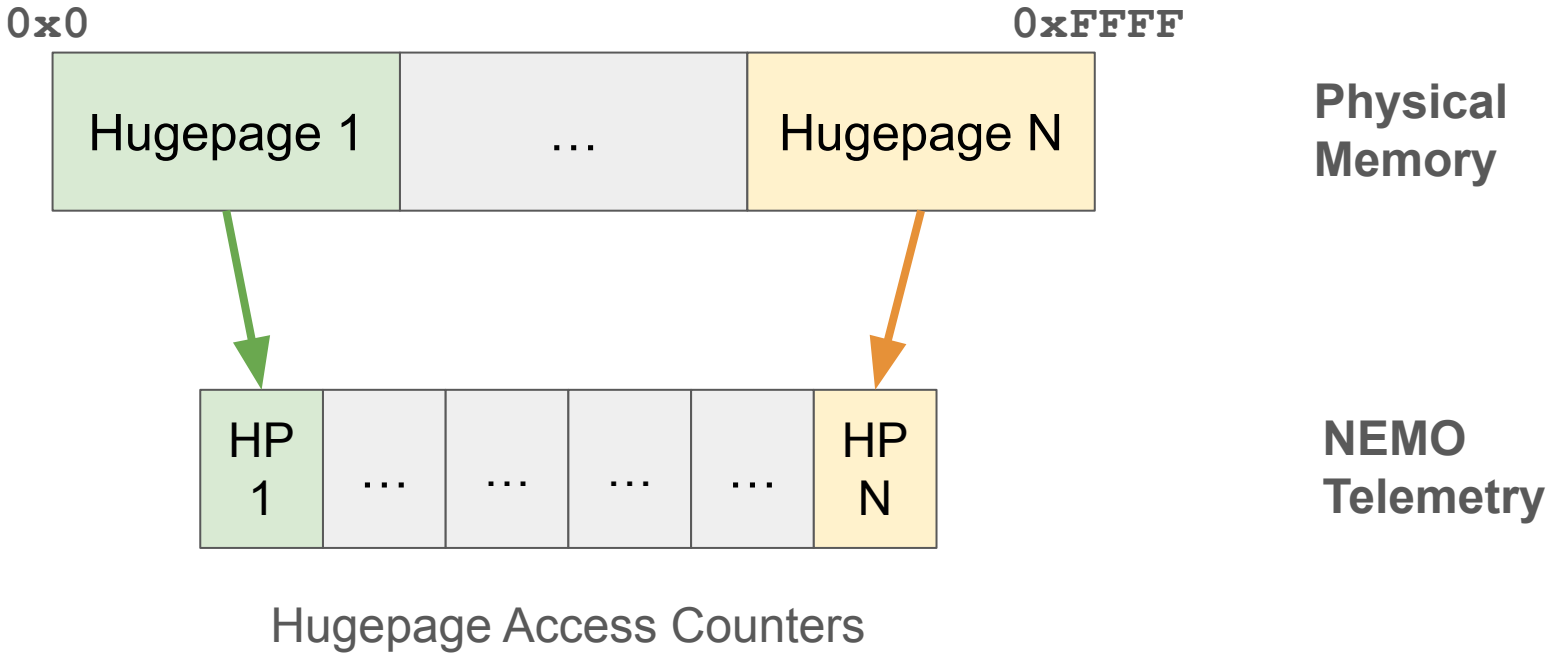


Physical Memory



NEMO Telemetry

Use Case 1: Hot Set Tracking



Hot Set Tracking: Evaluation Setup

System Integration: HeMem, a SOTA memory tiering system

Baseline Telemetry: PEBS

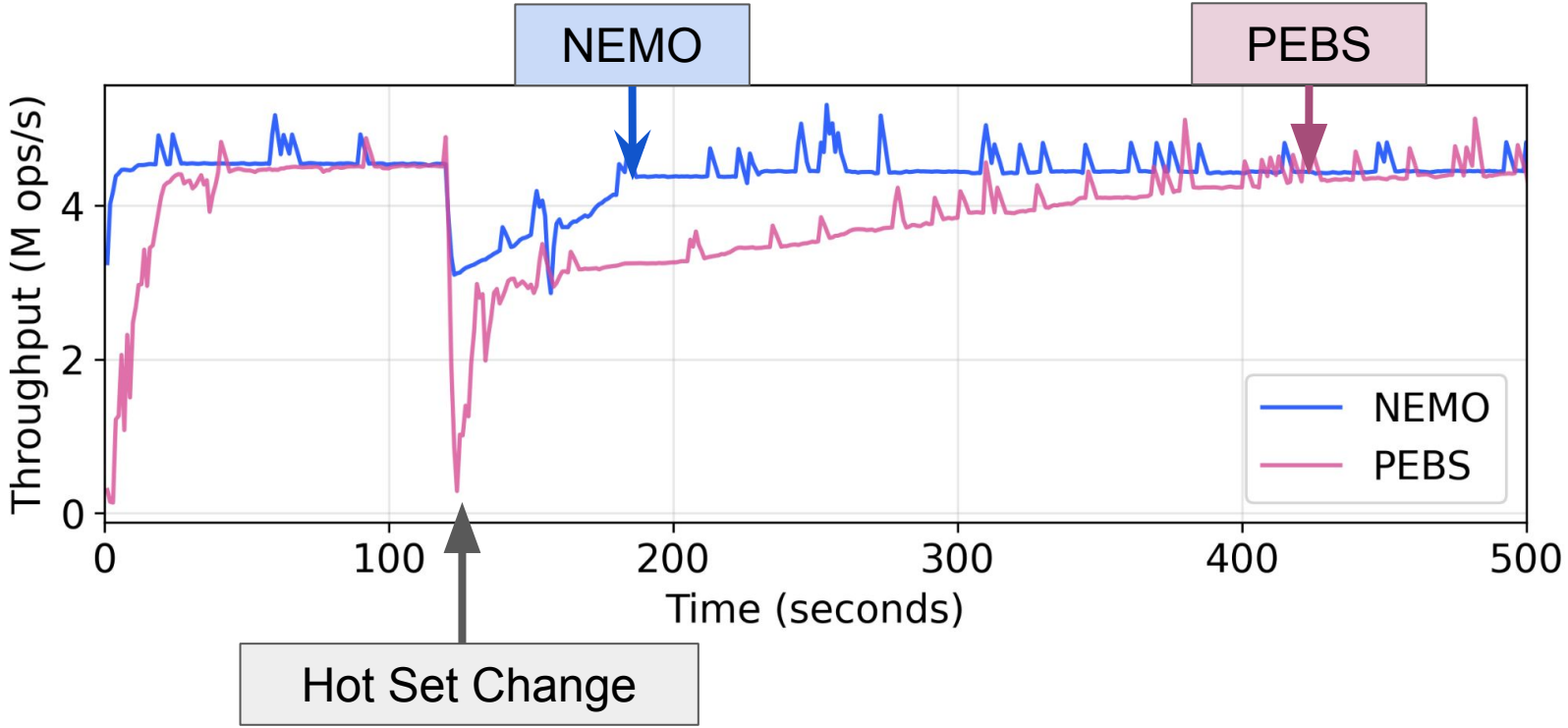
Application: FlexKVS, a key-value store

Synthetic workload: 20% hot set shifts to a disjoint 20% hot set mid-run

Eval Q: On hot set shift, how quickly does application performance recover?

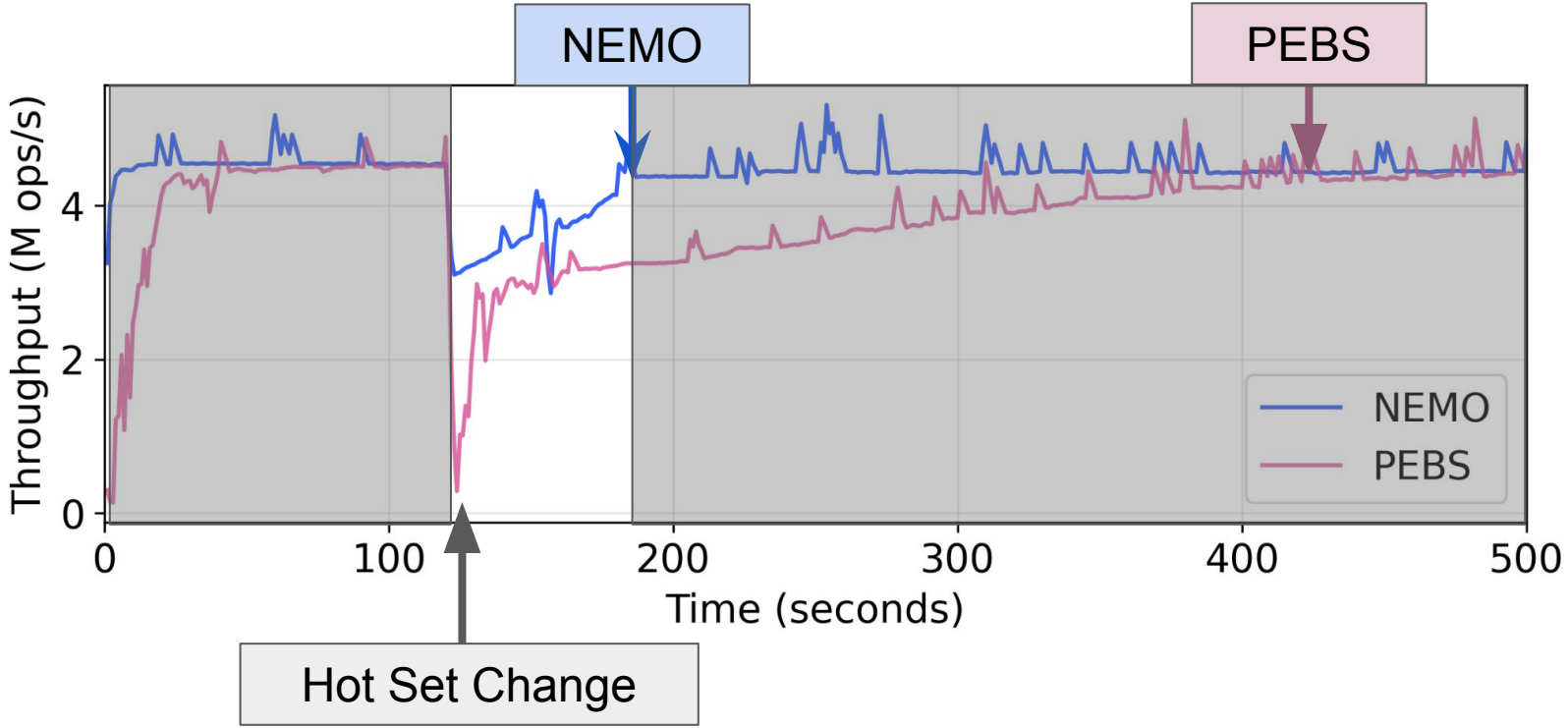
NEMO: Faster Hot Set Change Detection

Takeaway: NEMO enables 5x faster performance recovery than PEBS



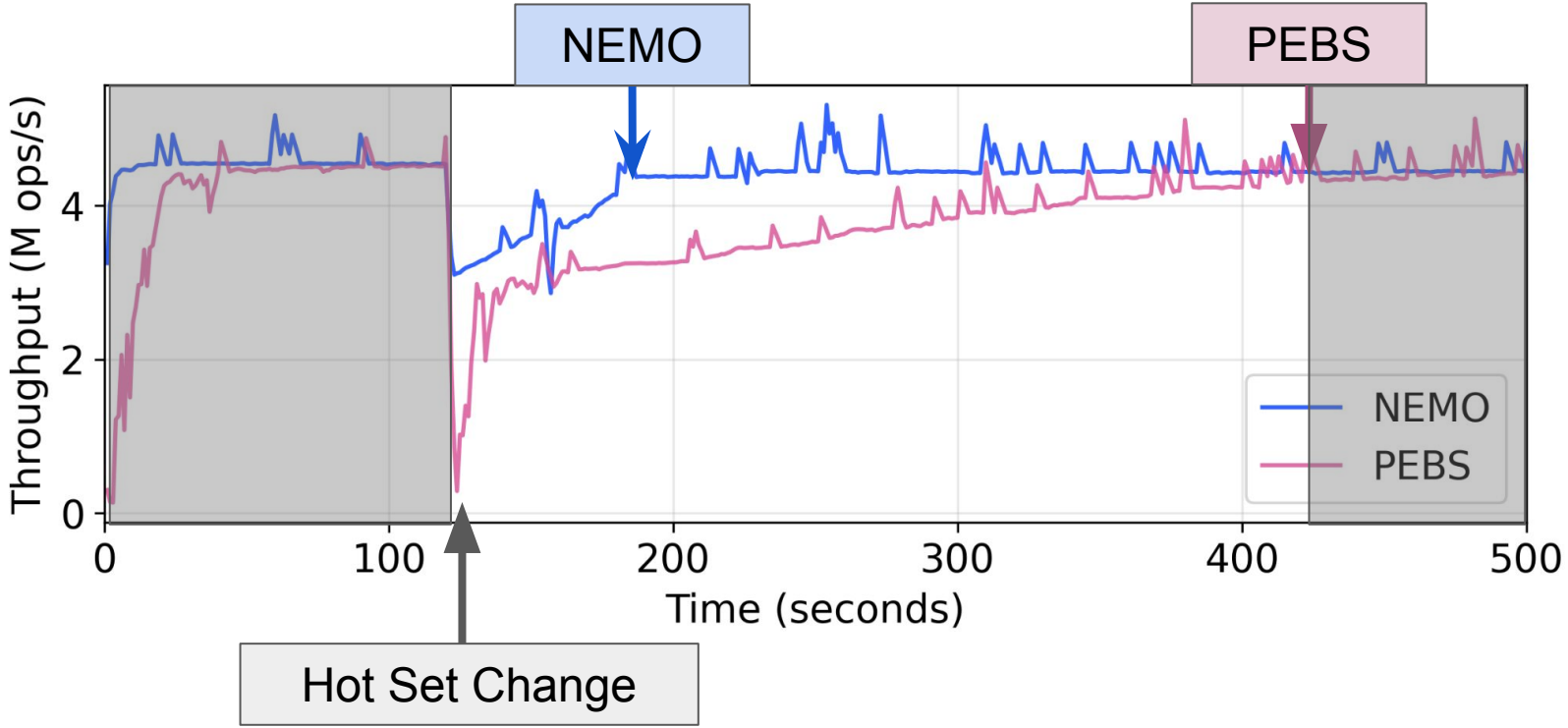
NEMO: Faster Hot Set Change Detection

Takeaway: NEMO enables 5x faster performance recovery than PEBS



NEMO: Faster Hot Set Change Detection

Takeaway: NEMO enables 5x faster performance recovery than PEBS



Splitting Hugepages: Evaluation Setup

System Integration: Memtis, a SOTA memory-tiering system that splits hugepages

Baseline Telemetry: PEBS

Setup: sweep fast:slow memory ratio from 1:2 (Azure) to 1:8 (future CXL setups)

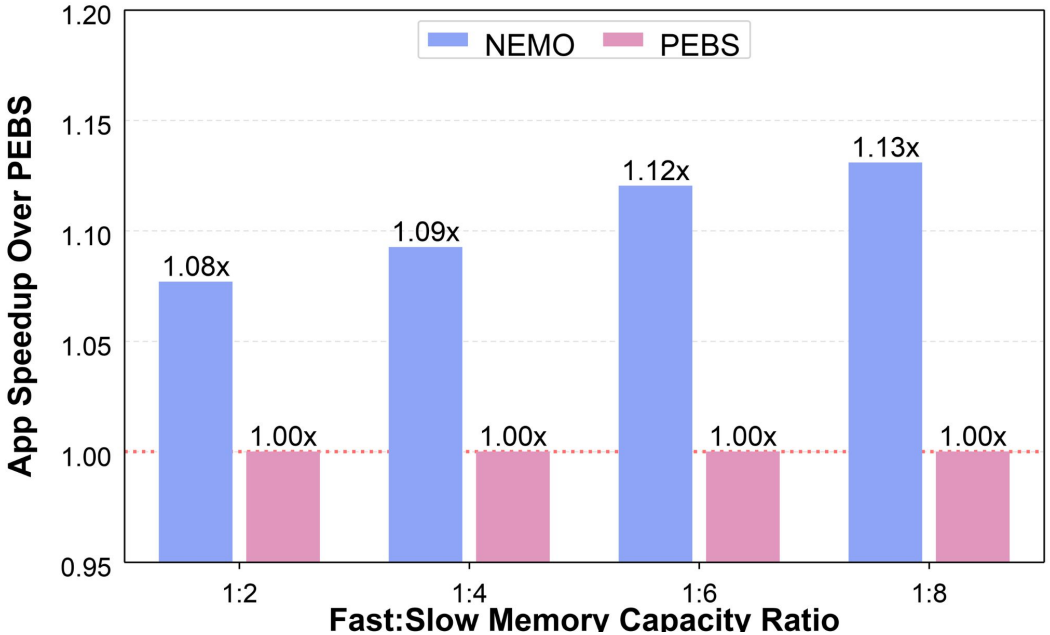
Application: Silo, an in-memory database engine, on YCSB-C (readonly)

Eval Q: Can NEMO's higher-fidelity telemetry improve app performance?

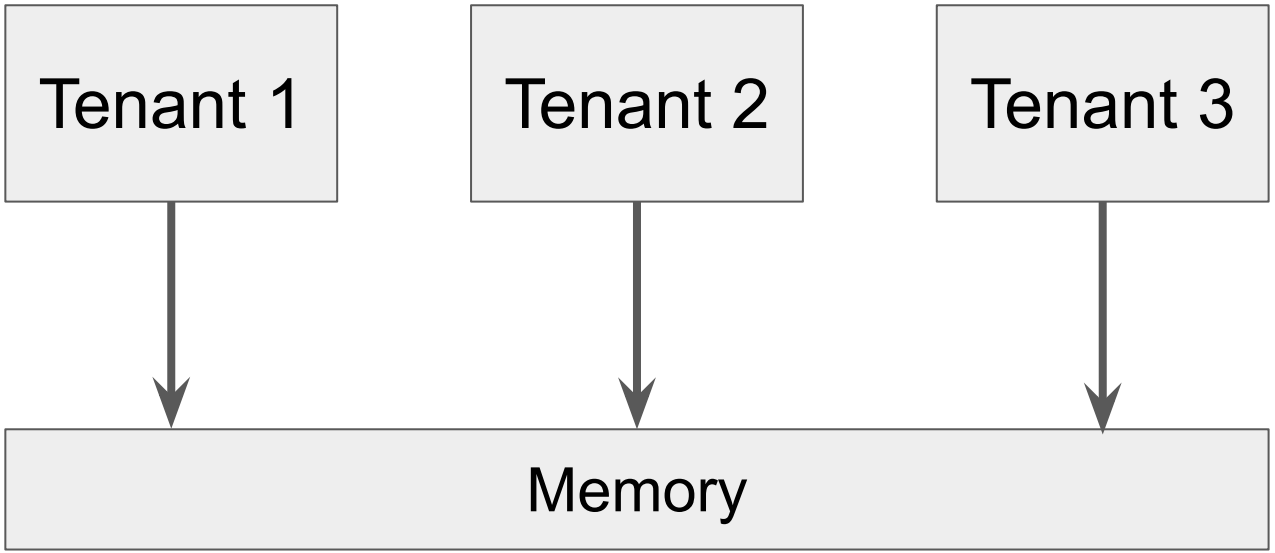
NEMO: Higher Page-Split Decision Quality

Takeaways:

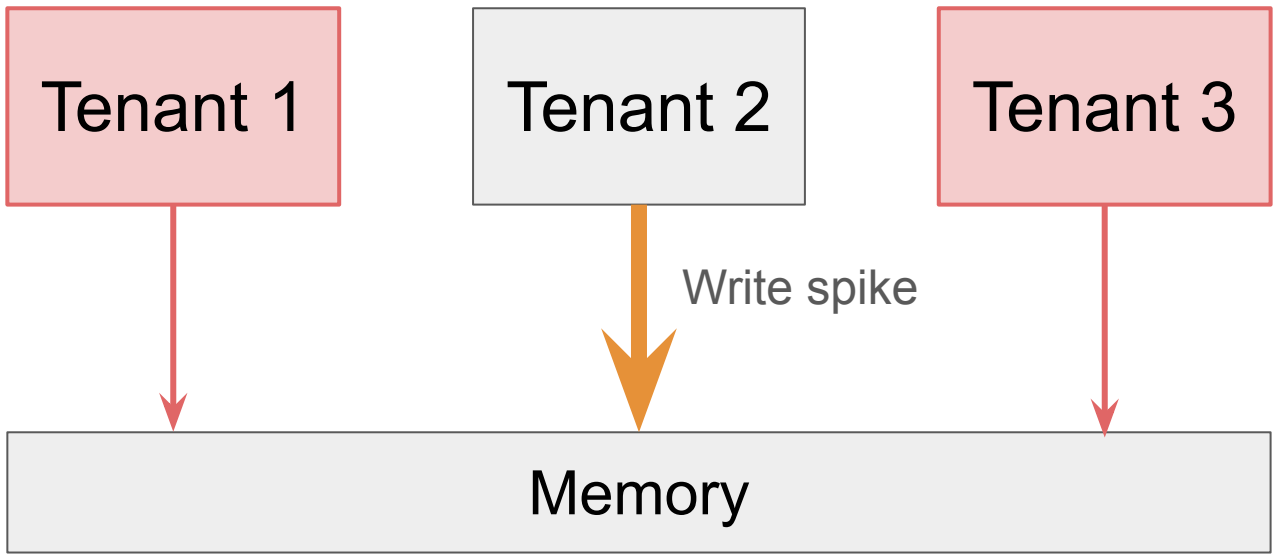
- NEMO improves Silo throughput relative to PEBS
- Silo throughput improvement widens as fast tier shrinks



Use Case 3: Performance Isolation



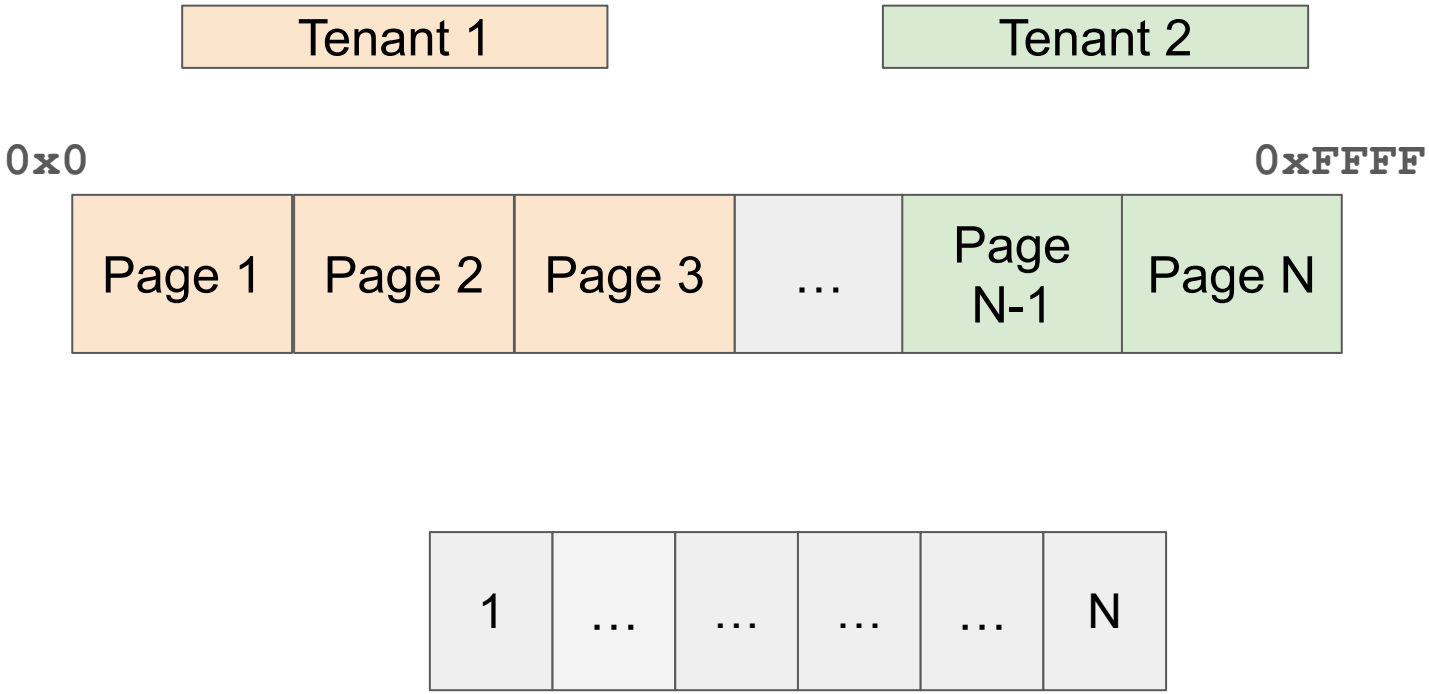
Use Case 3: Performance Isolation



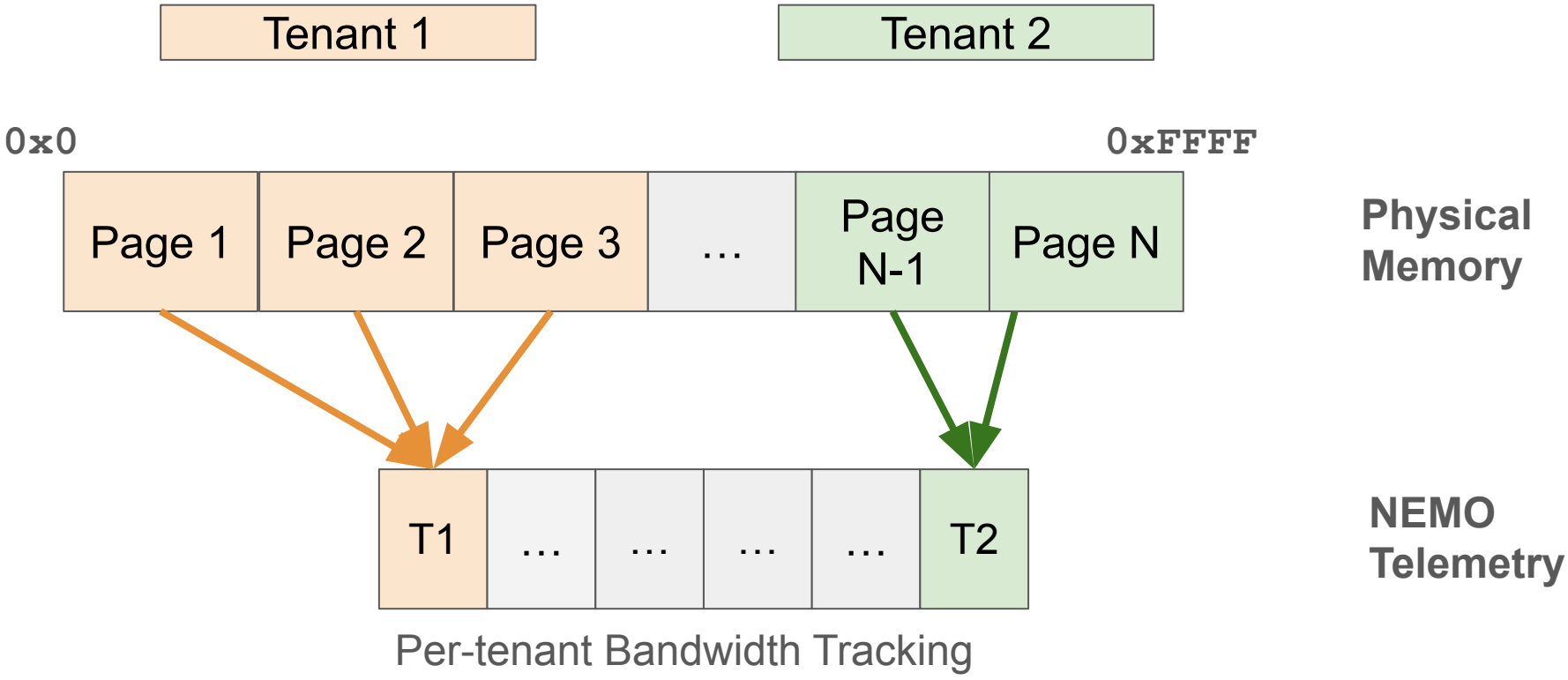
NEMO Telemetry: Tenant Bandwidth Tracking



NEMO Telemetry: Tenant Bandwidth Tracking



NEMO Telemetry: Tenant Bandwidth Tracking



Performance Isolation: Evaluation Setup

System Integration: userspace controller using cgroups to allocate CPU time

Applications:

- ***Victim***: FlexKVS, a key-value store
- ***Antagonist***: a process that consumes high memory bandwidth every 0.5s

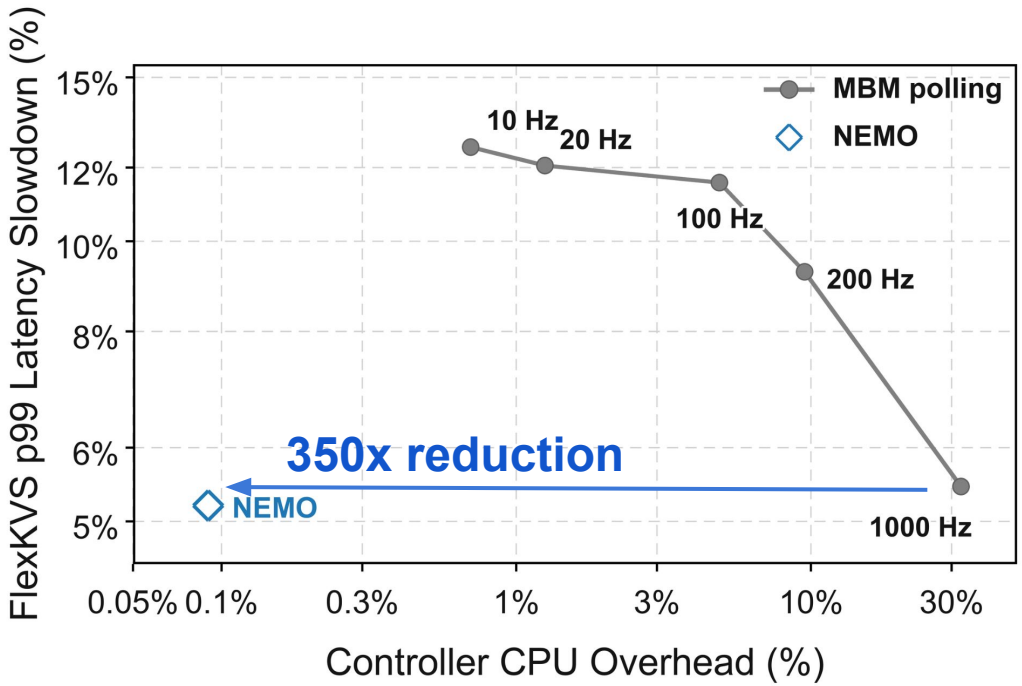
Baseline Telemetry: Intel Memory Bandwidth Monitoring (MBM)

- MBM requires polling => fidelity-overhead trade-off

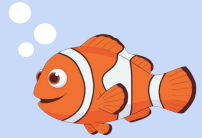
Eval Q: Can NEMO eliminate the trade-off, keeping high fidelity?

NEMO: Low-Overhead Performance Isolation

Takeaway: NEMO has 350x lower overhead than MBM, same isolation



Nimble and Expressive Memory Observability



- Today's OS memory telemetry trades between fidelity & overhead
- NEMO: a programmable telemetry engine on memory controllers
- Match-update-notify: **flexible**, **high-fidelity** telemetries **w/o overhead**
  - Hot set tracking: 5x faster recovery than HeMem+PEBS
  - Hugepage splitting: 1.13x higher app throughput than Memtis+PEBS
  - Perf. isolation: 350x overhead reduction than Cgroup+MBM

HW + SW source code: https://github.com/vic-lsh/nemo

Backup Slides

Use Case 1: Hot Set Tracking

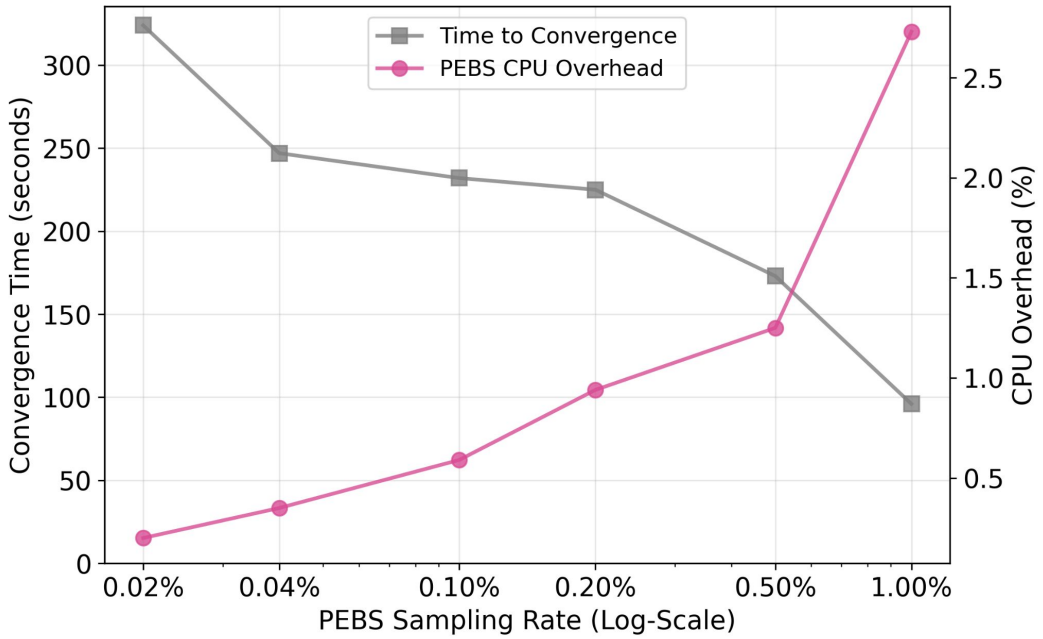


Fig. 4: PEBS sampling rate v.s. performance recovery time

Use Case 1: Hot Set Tracking (FASTER)

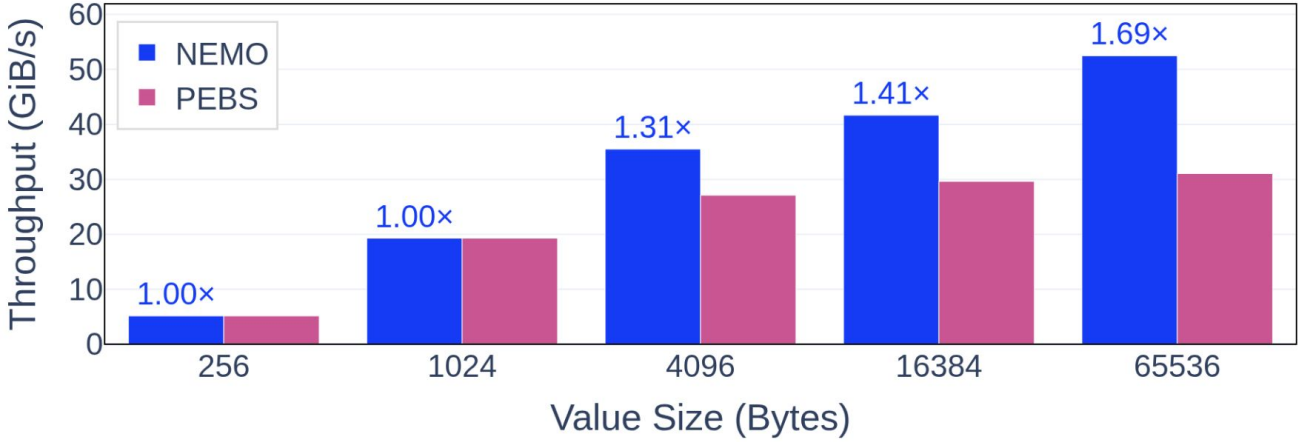


Figure 5: FASTER KV throughput across various value sizes. HeMem-NEMO identifies the hot set with larger value sizes, which PEBS is unable to do because of the prefetcher. There is no hot set with smaller value sizes.

Use Case 2: Hugepage Splitting

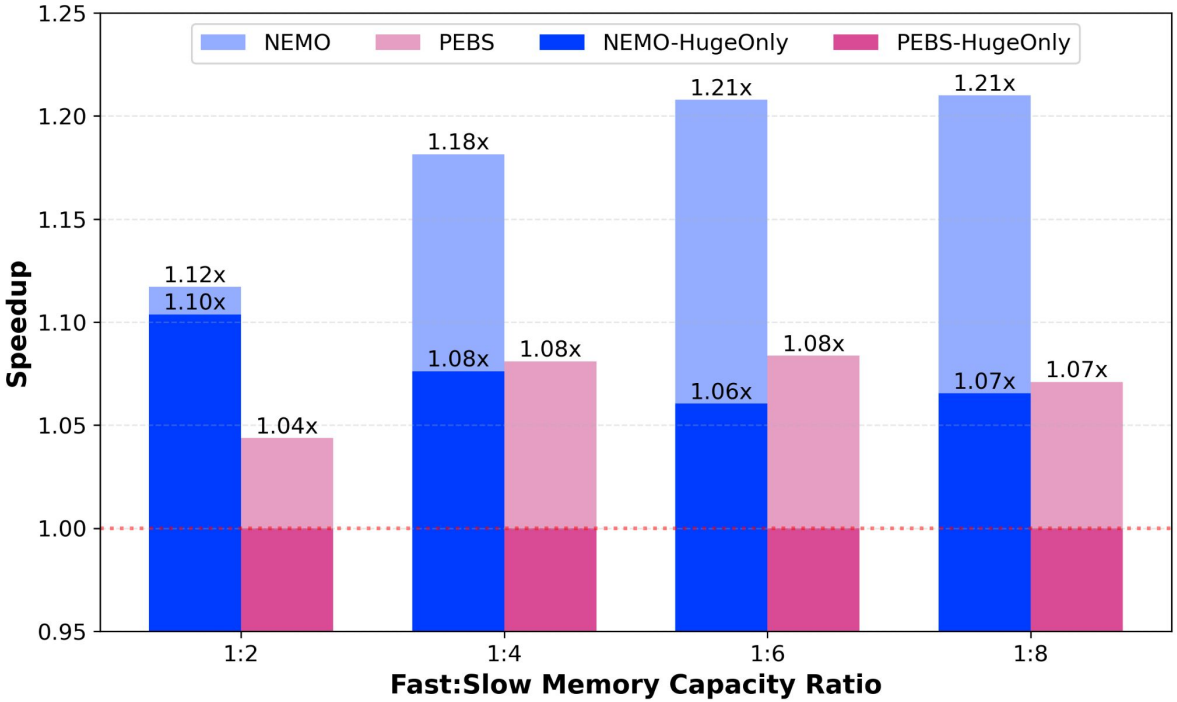


Fig. 6a: Silo throughput under various configurations

Use Case 2: Hugepage Splitting

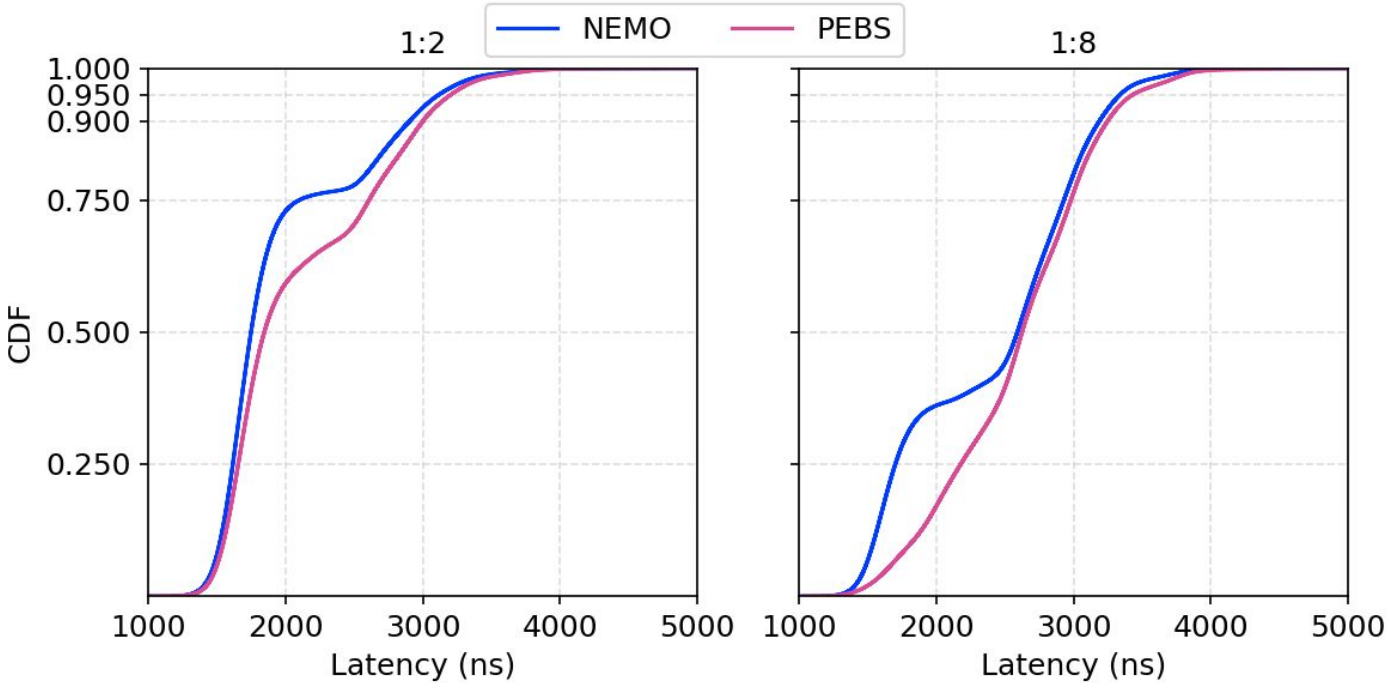


Fig. 6b: Silo latency CDF under 1:2 and 1:8 fast:slow memory configurations

Use Case 2: Hugepage Splitting

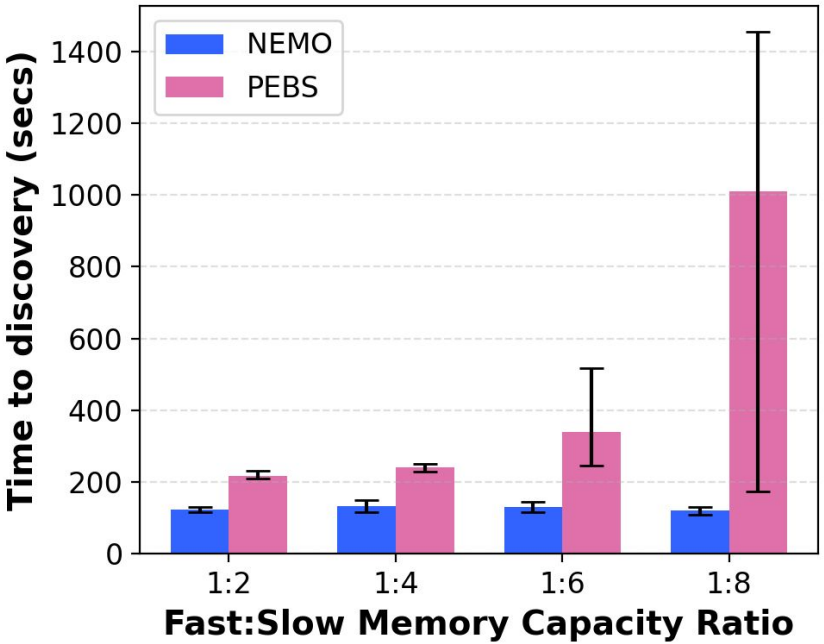


Fig. 7b: median time to identify 95% of page-split candidates over 5 runs; error bars show min/max