ASPLOS 2026

PACT: A Criticality-First Design for Tiered Memory

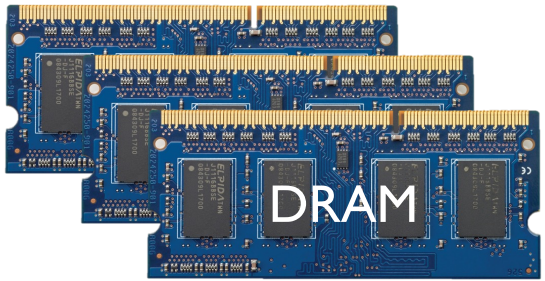
Hamid Hadian, Jinshu Liu*, Hanchen Xu*, Hansen Idden, Huaicheng Li

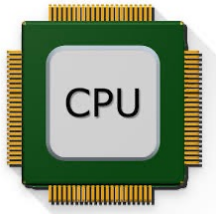
VIRGINIA TECH

* *Equal contribution*

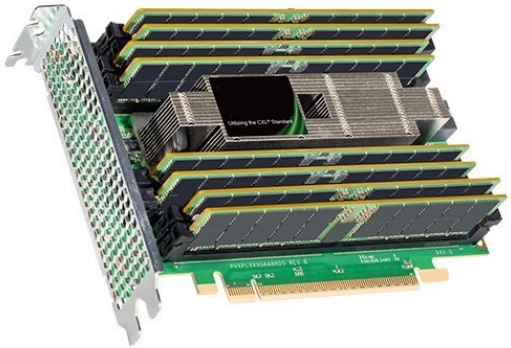
Modern workloads are memory-intensive

redis VOLTDB LLaMA C++ GAP APACHE Spark





CXL



Limited Size

Low memory access latency (~100ns)

Scaling

High capacity

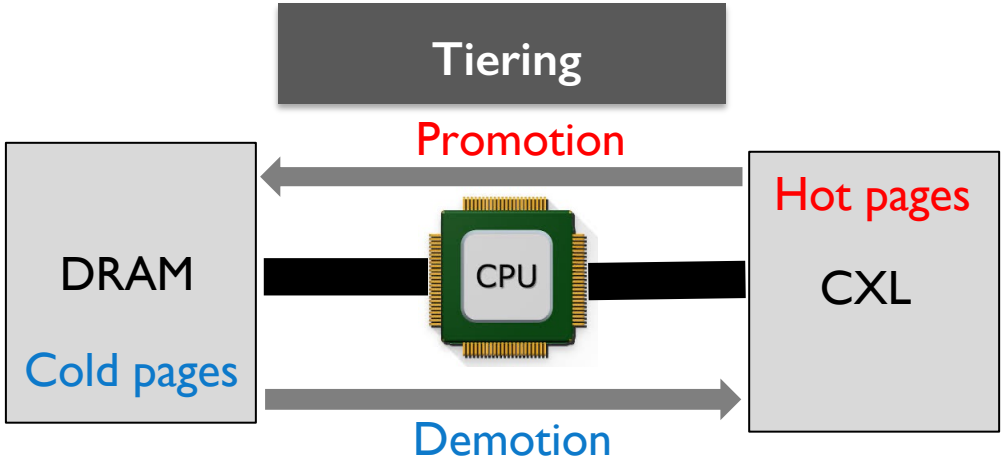
Slower (~2-3X)

CXL promises more capacity at the cost of more latency

Tiered Memory Architecture Becomes a New Norm

Workloads suffer from significant penalty while placing in CXL

- Dynamic and efficient placement
- Hotness is key foundation



Can hotness guide page migration efficiently ?

Pond [ASPLOS'23]

Hotness Cannot Represent True Performance Criticality

Common foundation of tiering: hotness (frequency/recency)

Hotness $\stackrel{?}{\neq}$ Performance criticality

Why hotness alone cannot capture true performance ?

Hotness alone fails to capture full spectrum of performance

→

Frequently accessed is not always performance critical

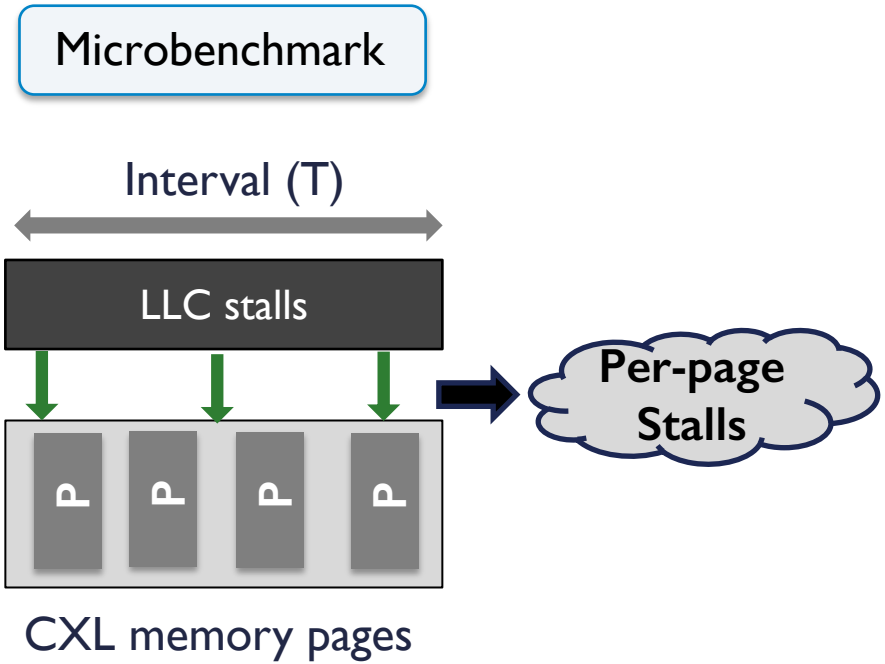
↓

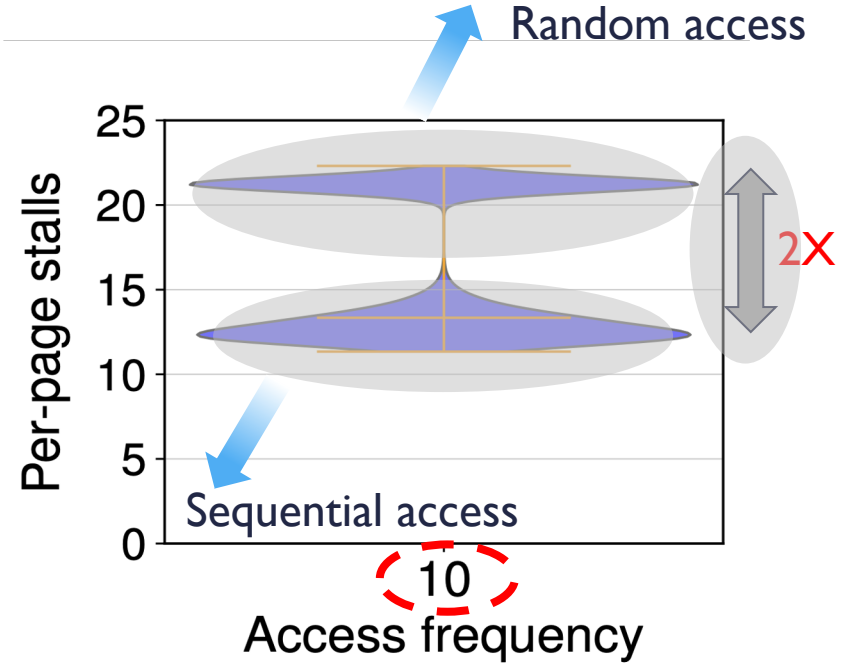
MLP can amortize latency (SoarAlto)

There is a need to move from frequency toward performance criticality

Quantifying Per-Page Access Criticality

Do all memory pages contribute the same in performance ?

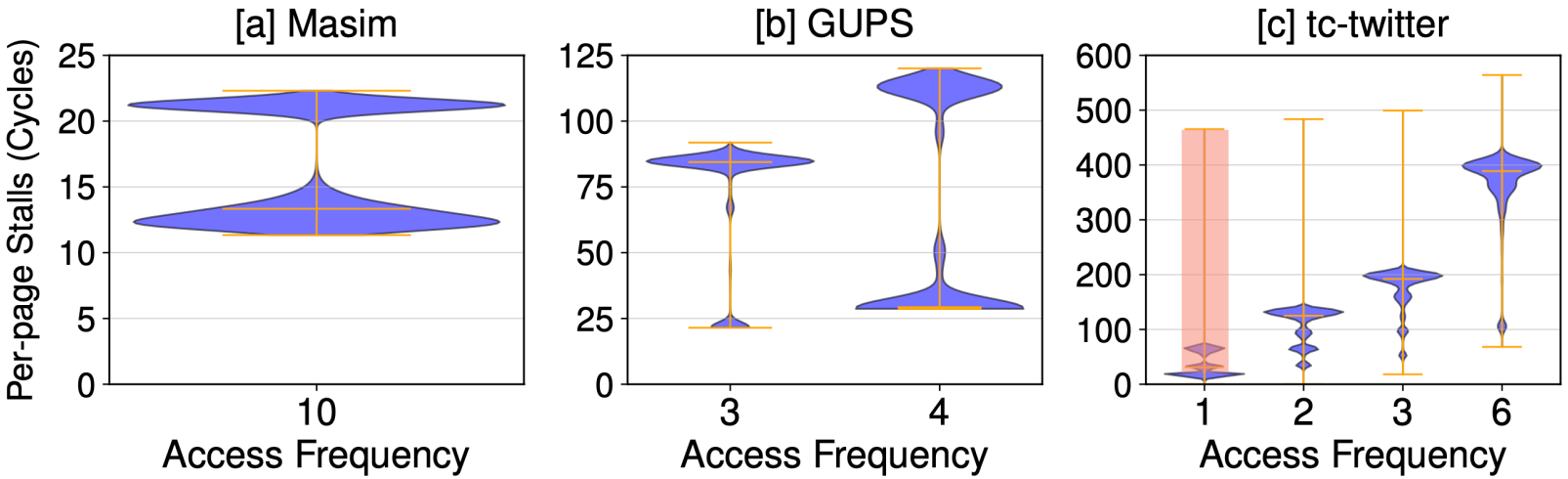




Pages with equal frequency contribute differently to performance

Quantifying Per-page Criticality

Spectrum of per-page stall costs for different workloads



Pages with the same access frequency can differ in stall cost by up to 65X

From Hotness to Criticality Driven Tiering

Can criticality be used as a new foundation to guide online memory tiering designs ?

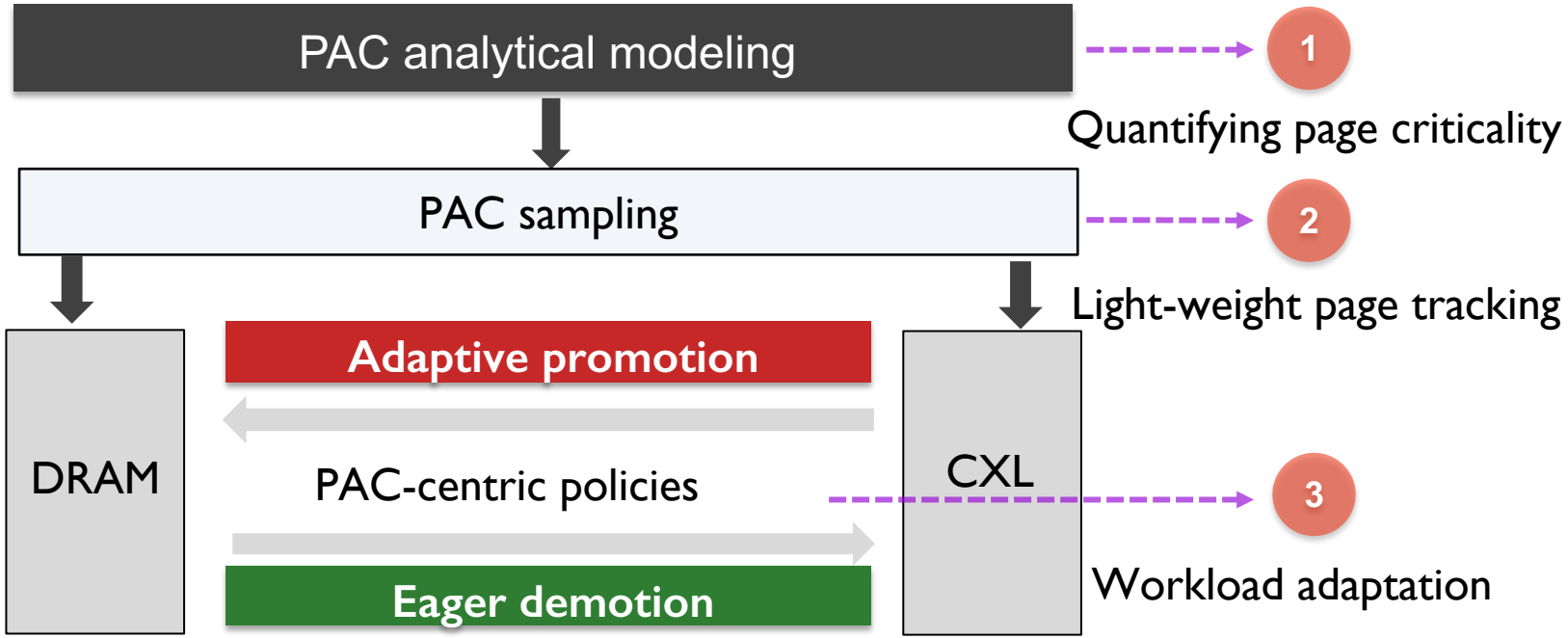
Research Challenges

PAC: Per-page access criticality;

Measuring per-page access contribution to workload slowdown

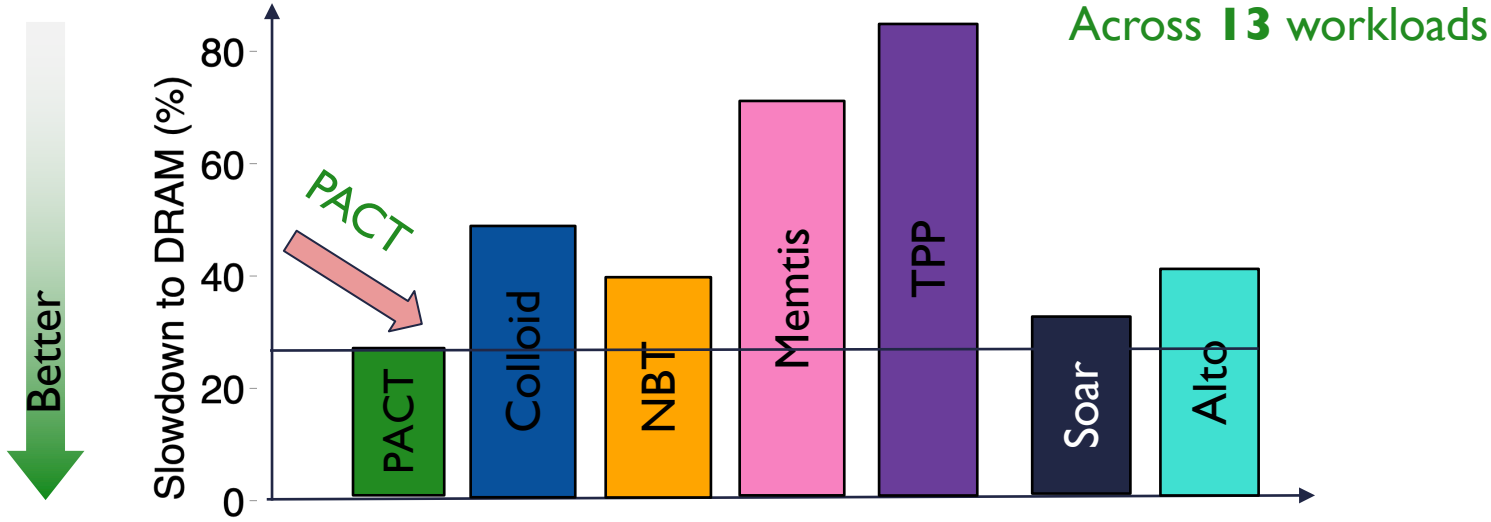
1. Quantifying PAC *(Coarse-grained counters)* → PAC analytical model → Per-page access criticality
2. PAC sampling *(Extra overhead)* → Light-weight tracking → From analytical model to light-weight sampling and tracking
3. Revisiting tiering policies *(Skewed PAC distribution)* → PAC-centric migrations → Adaptive promotion and eager demotion policies

PACT: PAC-Centric Tiered Memory Management



PACT Outperforms SOTA

Bc-kron workload: Improvements against other solutions



PACT improves workload performance over 2nd best tiering system by up to 61% and 50x fewer page migrations

Colloid [SOSP'24] Memtis [SOSP'23] TPP [ASPLOS'23] SoarAlto [OSDI'25]

Overview

PAC modeling & sampling

PAC-centric migration policies

Evaluation

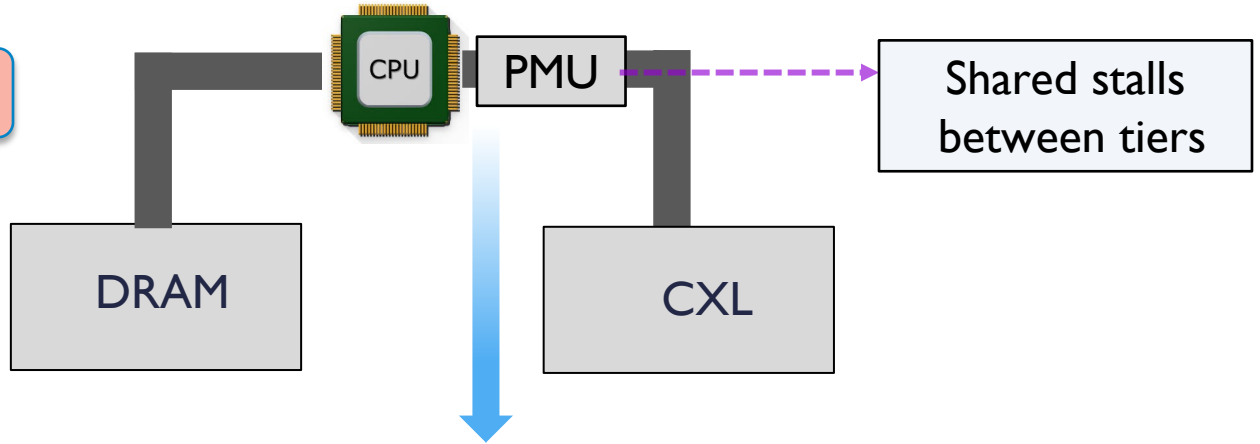
Conclusion

PAC Modeling: Quantifying Per-page Criticality

The observability gap:

To compute PAC, we need per-page accesses and stalls from the CXL

Lack of hardware support



$$Stall_{Total} = Stall_{DRAM} + Stall_{CXL}$$

It requires estimation in the absence of direct hardware support

PAC Modeling: Quantifying Per-Page Criticality

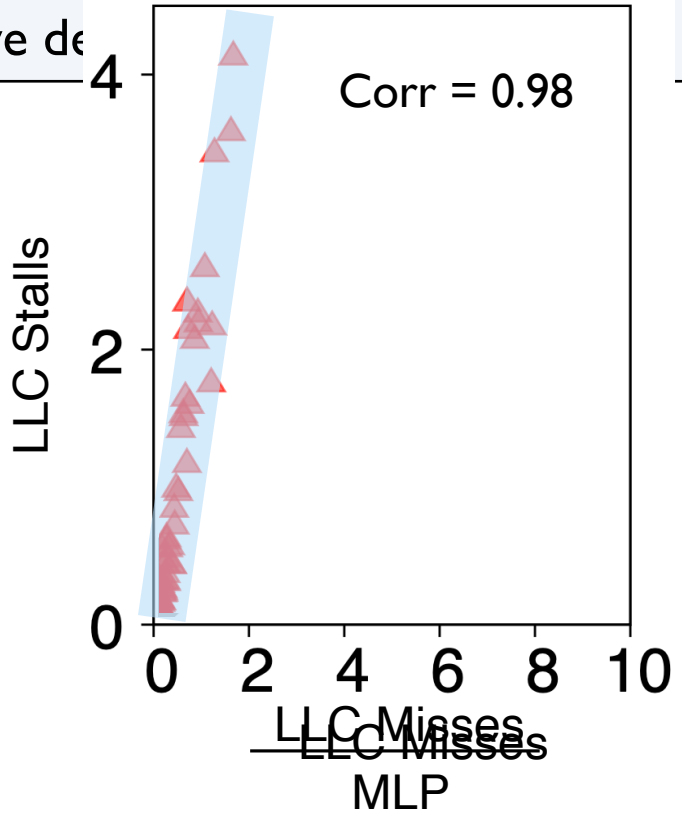
An extensive study with 96 workloads and 3 different setups

To observe CXL-induced stalls, we de

CXL induced stalls

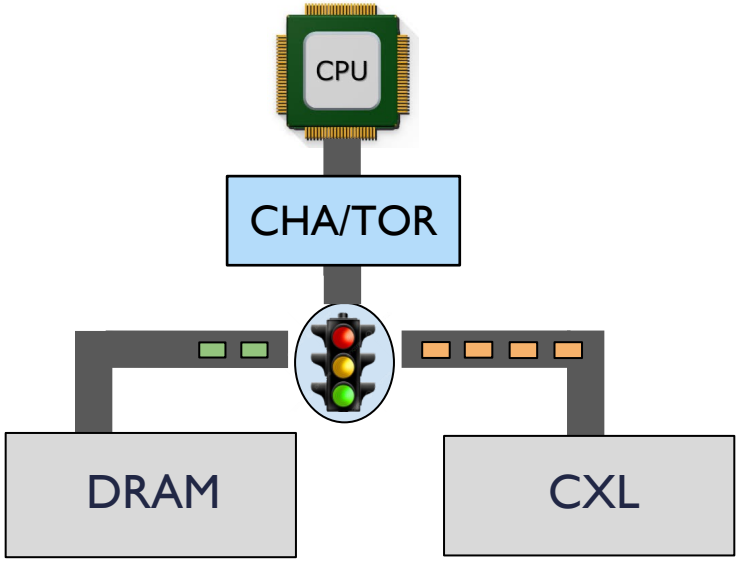
$$\text{Stall}_{\text{CXL}} = K \times \frac{\text{LLC-misses}_{\text{CXL}}}{\text{MLP}_{\text{CXL}}}$$

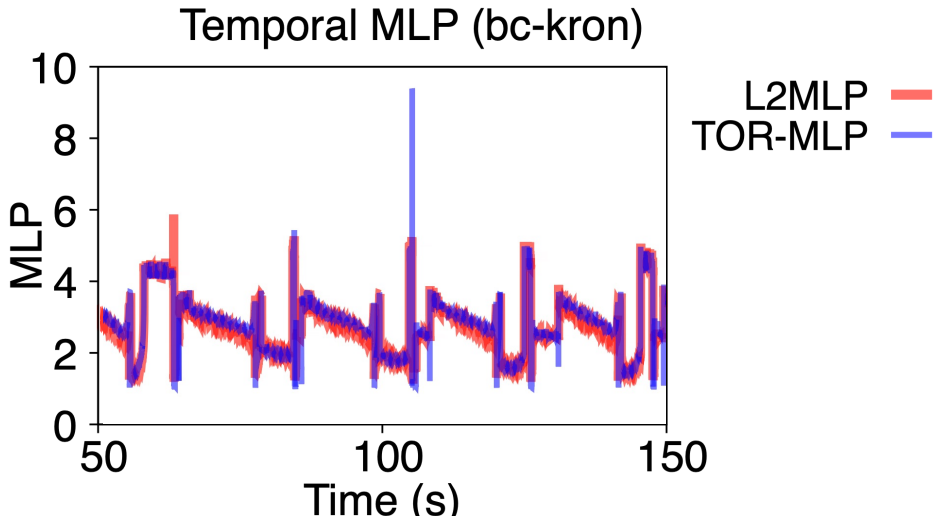
System-wide



PAC Modeling: Quantifying Per-Page Criticality

Measuring per-tier MLP using CHA (Cache Home Agent) metrics





$$MLP_{CXL} = \frac{\text{TOR occupancy}}{\text{Number of active cycles}}$$

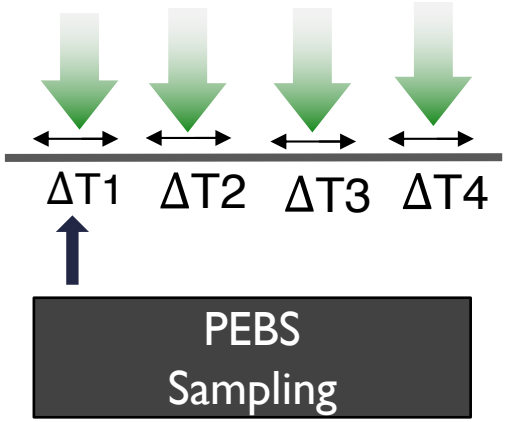
Only two TOR counters enable per-tier MLP estimation

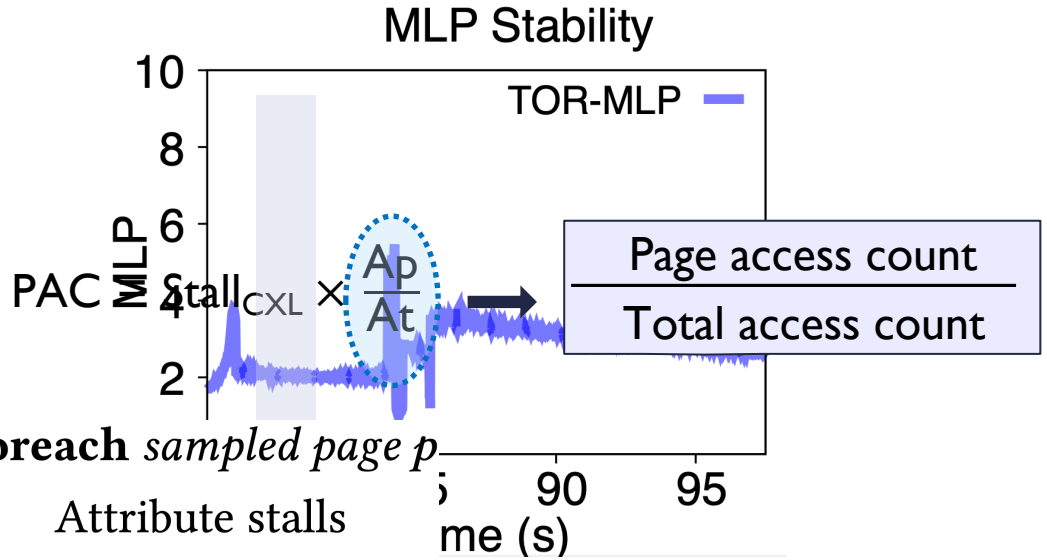
PAC Modeling: Quantifying Per-Page Criticality

Final step : Calculating per-page criticality (PAC)

MLP stability

$$Stall_{CXL} = K \times \frac{LLC\text{-}misses_{CXL}}{MLP_{CXL}}$$





PAC value is assigned to each page by an analytical model

Overview

PAC modeling & sampling

PAC-centric migration policies

Evaluation

Conclusion

PAC-Centric Policies: Adaptive Promotion

How can we redesign PAC-centric polices ?

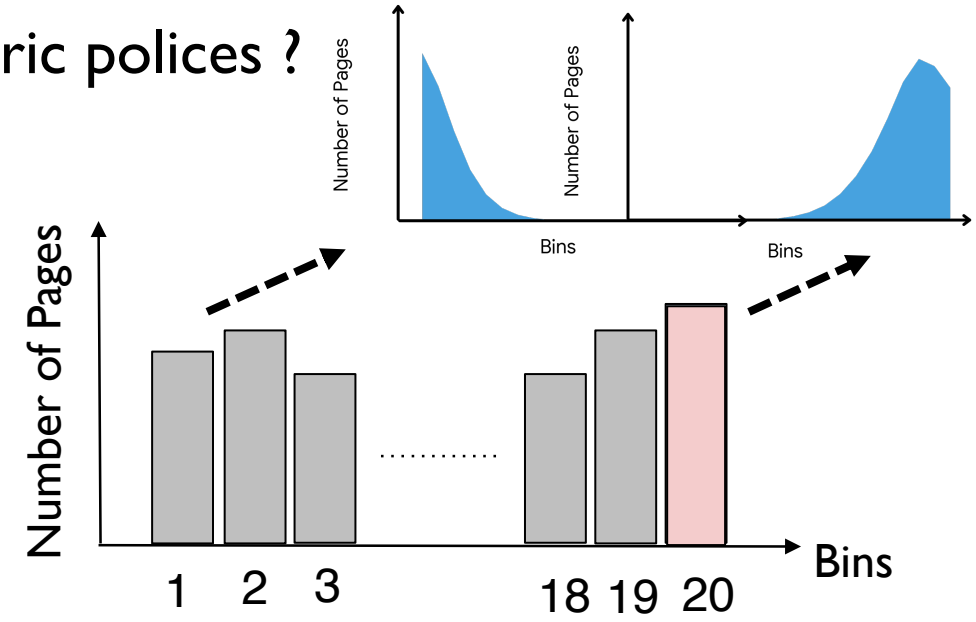
Skewed PAC values invalidate static threshold-based approach

Naïve static binning

PAC: [min . . . max]

Inaccurate promotion

Threshold-based

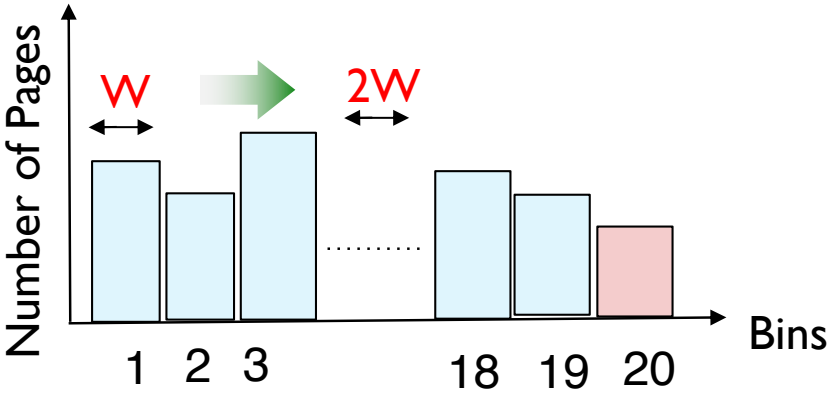


PAC-Centric Policies: Adaptive Promotion

From static binning to adaptive binning

Adaptive promotion

Adaptive binning



Bin width:
Freedman-Diaconis

PAC: [min . . . max]

Q1 Q3

Reservoir Sampling
First and third quartile of PAC values

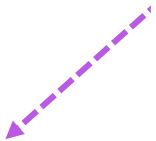
Integrating two mathematical methods lead to smooth distribution

Tiering Policies: Eager Demotion

To enable timely promotion, we proactively free fast-tier space

Balances need to free fast-tier

Fine-grained control over LRU



Maintains empty room

Demotion intensity could be configured

Eager demotion boosts functionality of adaptive promotion

Overview

PAC modeling & sampling

PAC-centric migration policies

Evaluation

Conclusion

Evaluation Setup

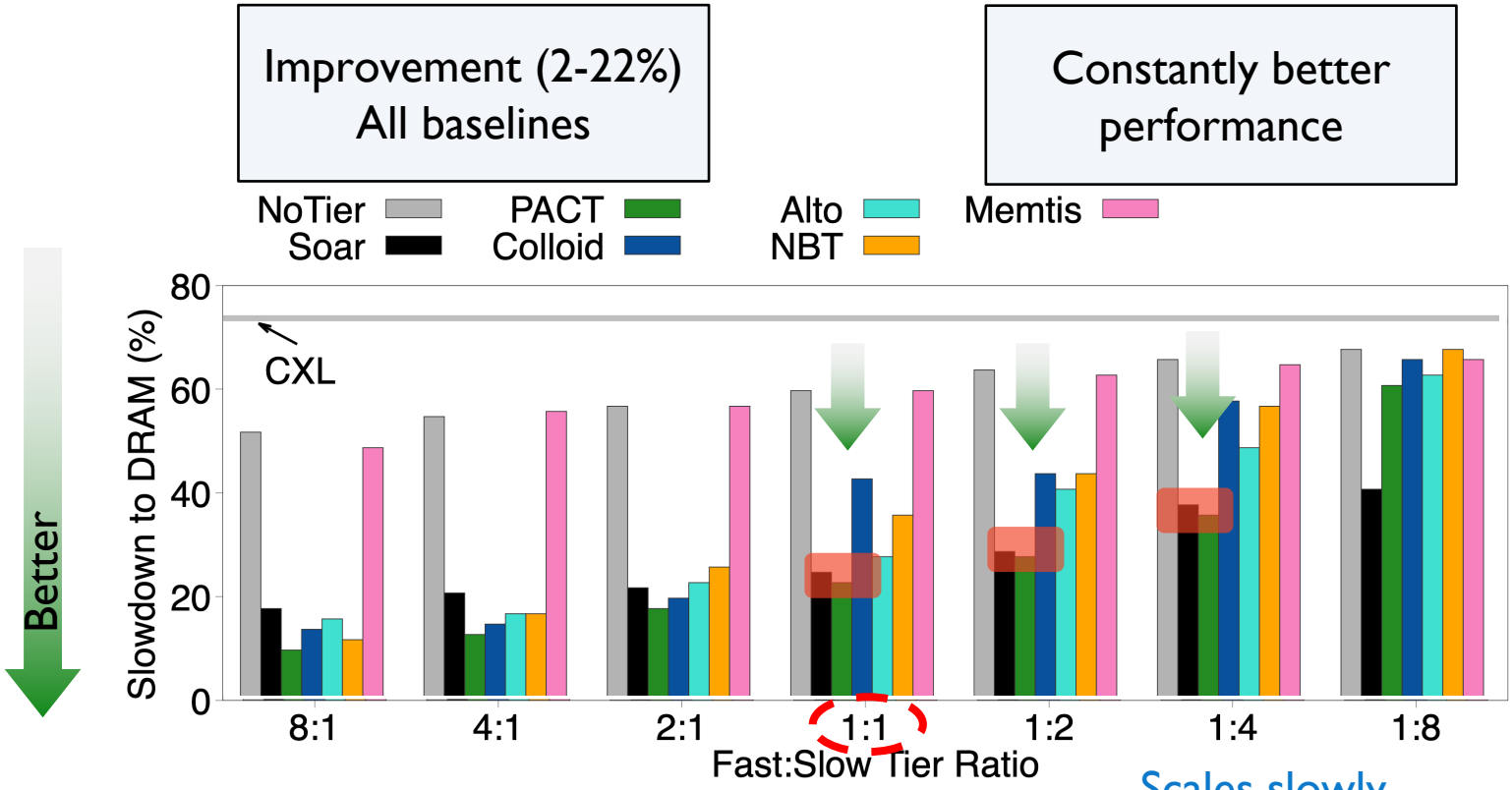
Local DRAM	90ns
Emulated CXL	190ns

7 state-of-the-art tiering designs:

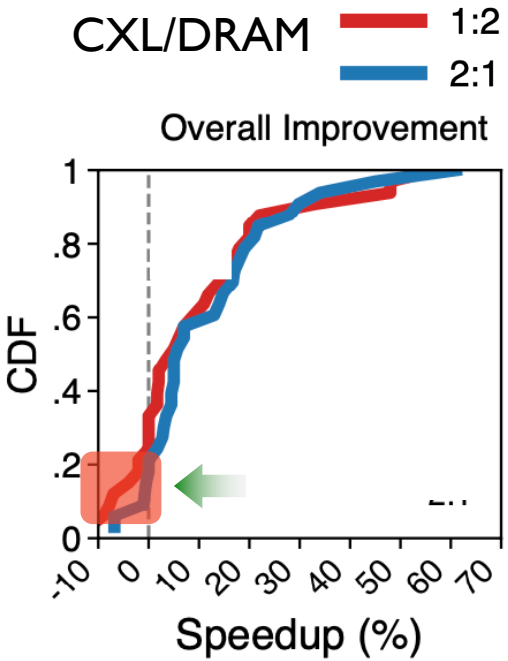
Soar	OSDI'25	Offline analysis
Alto	OSDI'25	MLP-aware
Memtis	SOSP'23	THP-Aware
Colloid	SOSP'24	Latency-balance
Nomad	OSDI'24	Replication
TPP	ASPLOS'23	NUMA balancing
NBT	Linux	NUMA balancing

PACT Performance Under Varying DRAM/CXL Ratios

Workload : bc-kron from GAPBS as a representative workload

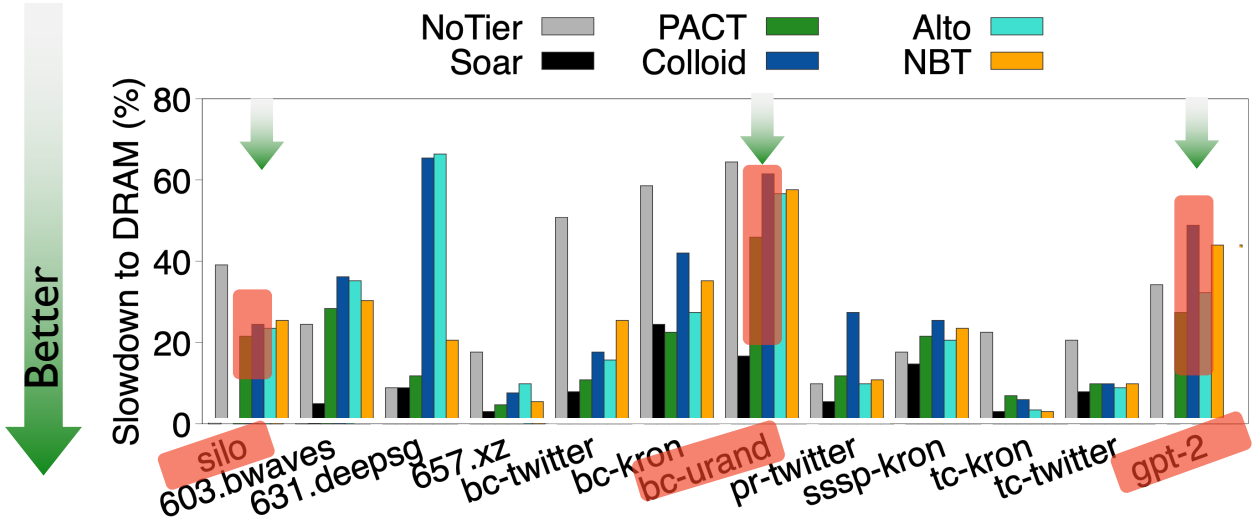


All Workloads: SPEC, GAPBS, key-value stores



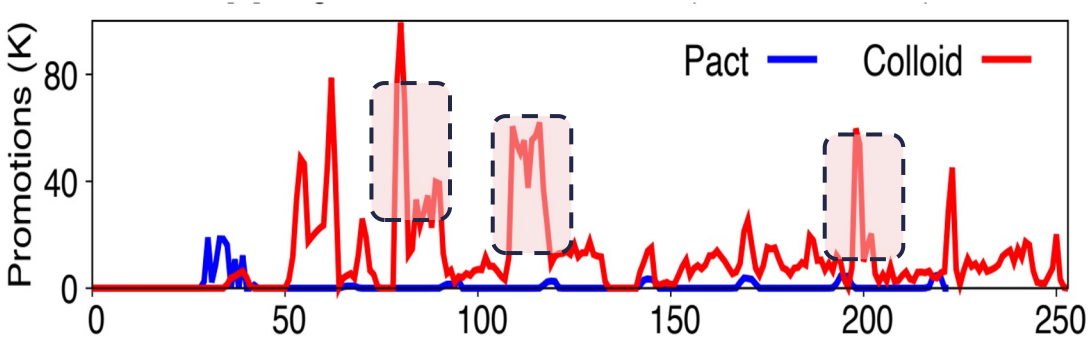
PACT Achieves better performance up to 61%

PACT shows a robust behavior

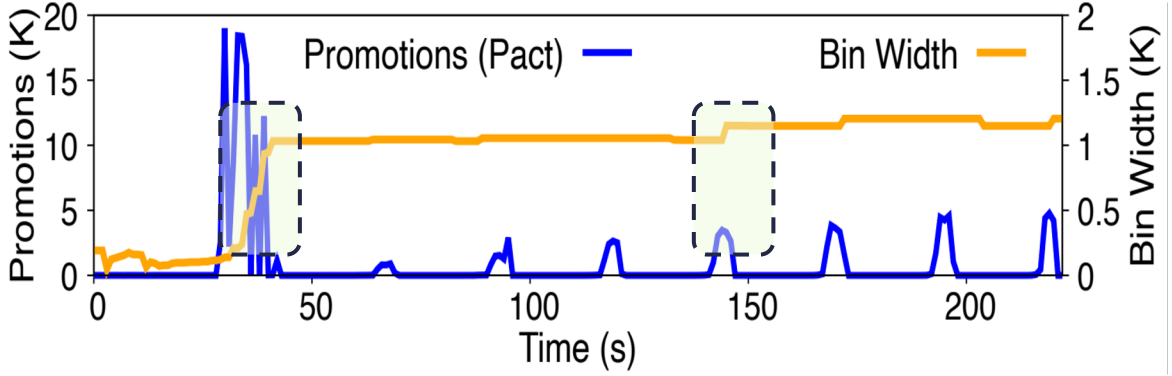


PACT Adapativity

How adaptive promotion policy can make online decisions ?

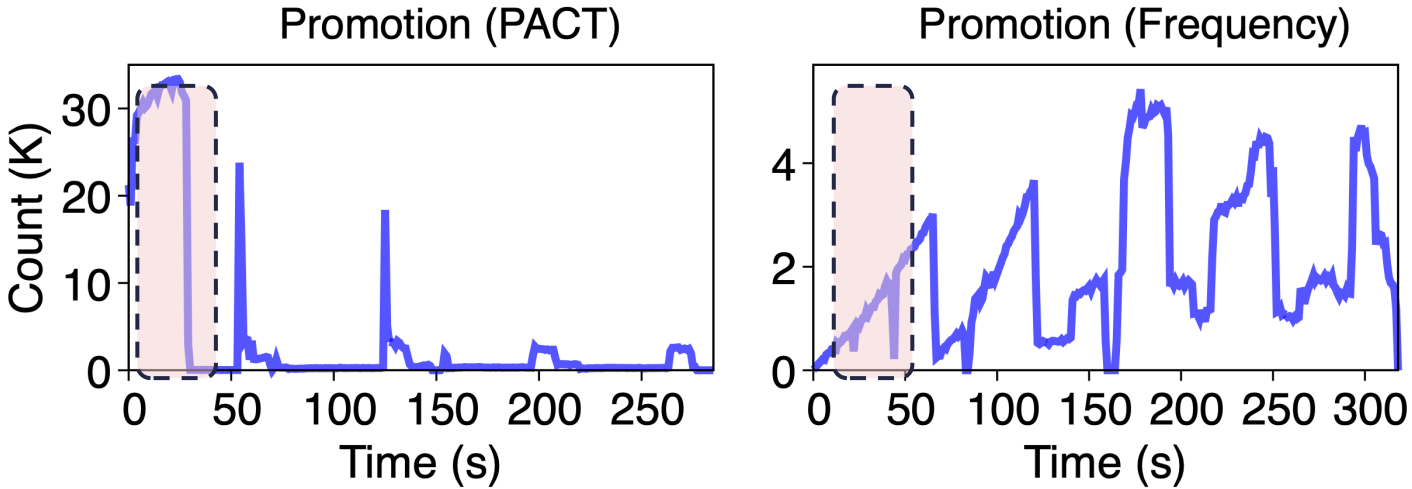


PACT: 500K migrations against 4.2 million migrations for Colloid



Sensitivity Study: PAC vs. Hotness

Differences of PAC-centric and hotness-based promotions



More in the Paper

- Workload colocation
- Bandwidth contention
- Mathematical methods
- PACT for THP pages

PACT: A Criticality-First Design for Tiered Memory

Hamid Hadian
Virginia Tech
Blacksburg, USA

Jinshu Liu*
Virginia Tech
Blacksburg, USA

Hanchen Xu*
Virginia Tech
Blacksburg, USA

Hansen Idden
Virginia Tech
Blacksburg, USA

Huaicheng Li
Virginia Tech
Blacksburg, USA

Abstract

Tiered memory systems typically place pages based on access frequency (hotness), yet frequency alone fails to capture the true performance impact. We present PACT, an online, page-granular tiered memory design that elevates performance criticality to a first-class design principle. At its core is Per-page Access Criticality (PAC), a fine-grained metric that quantifies each page's contribution to application performance rather than merely counting accesses. PACT profiles PAC online using a lightweight analytical model that uniquely decomposes per-tier *memory-level parallelism via hardware queue occupancy counters, enabling direct CPU stall attribution to individual pages. To handle highly skewed PAC distributions, PACT employs PAC-centric migration policies: eager demotion and adaptive promotion, to dynamically place performance-critical pages in DRAM. Across 13 workloads, PACT achieves up to 61% performance improvement over the best of 7 state-of-the-art tiering designs with up to 50× fewer migrations.*

CCS Concepts: • **Hardware → Emerging technologies**; • **Computer systems organization → Architectures**.

Keywords: Tiered Memory, Compute Express Link (CXL), Operating Systems

ACM Reference Format:
Hamid Hadian, Jinshu Liu, Hanchen Xu, Hansen Idden, and Huaicheng Li. 2026. PACT: A Criticality-First Design for Tiered Memory. In *Proceedings of the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '26), March 22–26, 2026, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3779212.3790198

1 Introduction

The growing gap between compute performance and memory capacity has made tiered memory architectures essential for modern datacenters. As DRAM scaling slows and memory-hungry workloads continue to grow, systems increasingly combine fast-tier memory (DRAM) with slow-tier alternatives (NUMA, persistent memory, and CXL) [23, 30, 37, 45]. Compute Express Link (CXL) accelerates this trend by enabling hardware-level memory disaggregation and pooling, but CXL access latencies remain 2–3× higher [3, 19, 21, 32, 47, 48]. This latency gap makes effective tiered memory management critical for performance-sensitive applications.

Existing tiered memory systems address this challenge through page sampling, allocation, and migration techniques that promote "hot" pages to fast-tier memory [17, 24, 25, 27, 29, 37, 44, 51–53, 56, 58, 59]. These systems rely on *hotness*, typically page-level access frequency, to guide placement decisions, assuming frequently accessed pages are performance-critical.

However, hotness is an unreliable proxy for performance impact [8, 22, 34]. Memory access criticality, defined as the performance cost an access imposes on the CPU, depends on many factors, such as access patterns, memory-level parallelism (MLP), and access latency, rather than access frequency alone [20, 22, 34, 51]. For instance, sequential accesses with high MLP (*e.g.*, array traversals) can tolerate slow-tier latency with minimal performance impact, while pointer-chasing operations with low MLP suffer proportional slowdowns [22, 34]. This fundamental disconnect, that frequency does not equal criticality, motivates our work.

While recent work has moved beyond access frequency toward criticality, these approaches either rely on offline, coarse-grained profiling (*e.g.*, object level) or incorporate criticality only as a reactive hint layered atop fundamentally hotness-driven policies [22, 34, 52]. Consequently, criticality is never elevated to a first-class design principle, leaving these systems without online adaptability or page-level precision necessary for effective tiered memory management.

Our key insight is that effective page placement demands a criticality-first redesign in which the runtime performance impact of each page access is directly quantified, rather than inferred through indirect proxies. To this end, we introduce **Per-page Access Criticality (PAC)**, a metric that quantifies each page's contribution to CPU stall time online through an accurate, *per-tier* MLP decomposition, derived from a study of 96 workloads. PAC provides fine-grained precision at both

*Equal contribution.

ASPLOS '26, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-2359-9/2026/03
https://doi.org/10.1145/3779212.3790198

Overview

PAC modeling & sampling

PAC-centric migration policies

Evaluation

Conclusion

Efficient tiered memory management is more critical than ever

- PACT: Fine-grained, online performance criticality
- PAC metric: A step toward criticality-aware systems
- Rethinking of tiering by using PAC-centric policies
- Possible use case of PACT is workload colocation

Thank you! Questions?