TPage: Transparent Page Management for CXL-Based Tiered Memory Systems

Maoxin Ye ★, Suzhen Wu ★, Jie Zhang $, Bo Mao ★

★Xiamen University $ Peking University

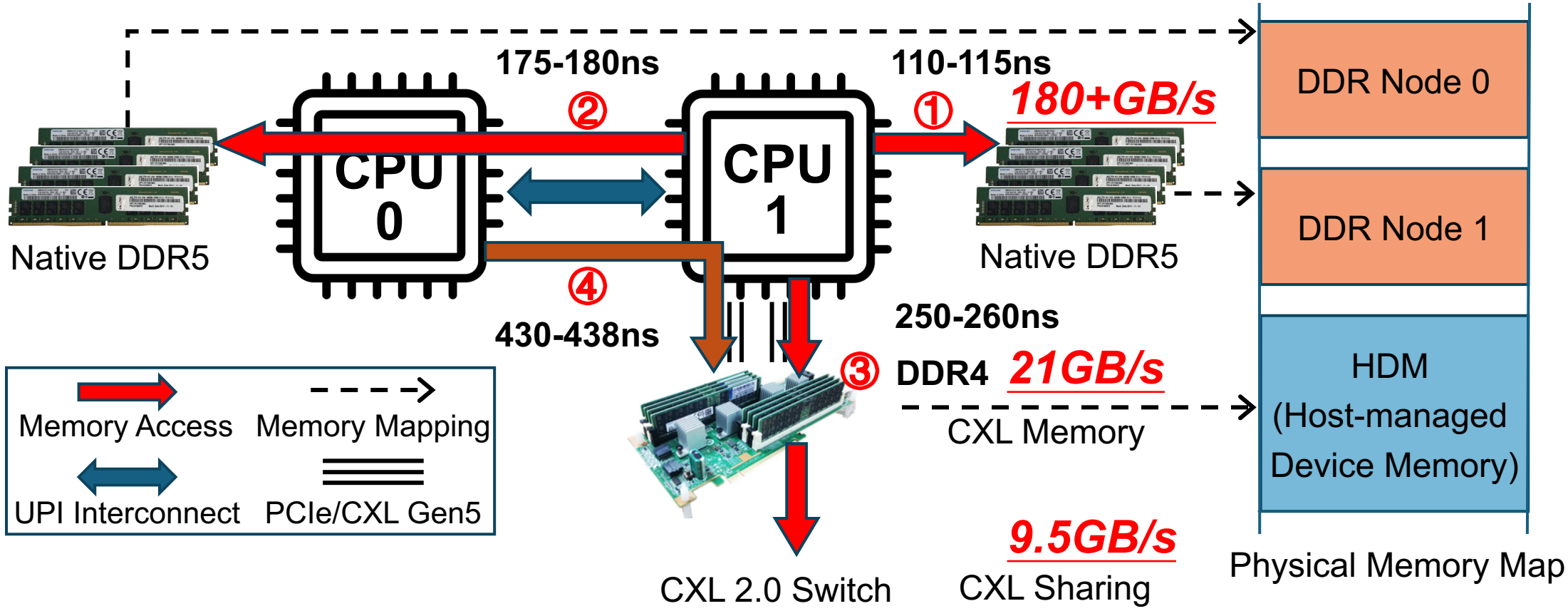
SC'26 | November 15-20, 2026, Chicago, Illinois, USA.

Outline

- Introdution
- Motivation
- TPage
- Evaluation

Introduction

- Memory-intensive applications (HPC, LLMs) are driving rapid growth in memory demand
- Traditional DDR faces capacity scaling limits
- ✓ CXL enables memory expansion and forms a tiered memory system
- CXL memory has higher latency and lower bandwidth than local DDR (**180 +GB/s** vs. **21 GB/s**)



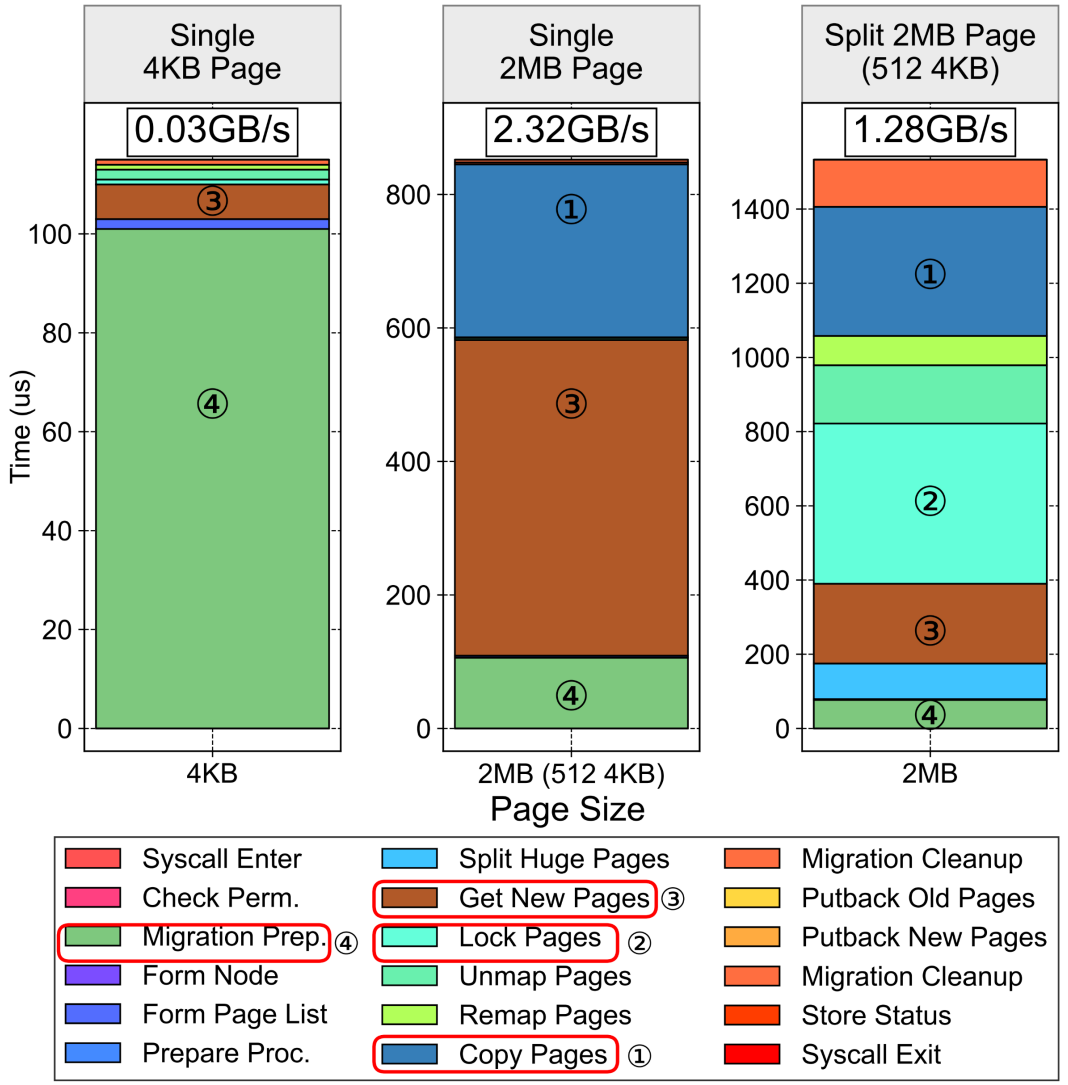
Motivation: Existing Solutions Fall Short

Problem ① — Slow Migration Throughput

✗ TPP's page-fault-based migration achieves only 4–16 MB/s

We benchmark three migration strategies on our CXL platform (Linux 6.9.0):

- At 4KB granularity , over 83% of migration time is spent on kernel preparation (LRU disable, synchronization), not actual data copying
- Batch migration (512 pages at once) amortizes preparation overhead, achieving hugepage-level throughput
- Split-then-migrate incurs additional locking cost (1.28 GB/s), but enables fine-grained subpage migration — a necessary trade-off for avoiding memory bloat

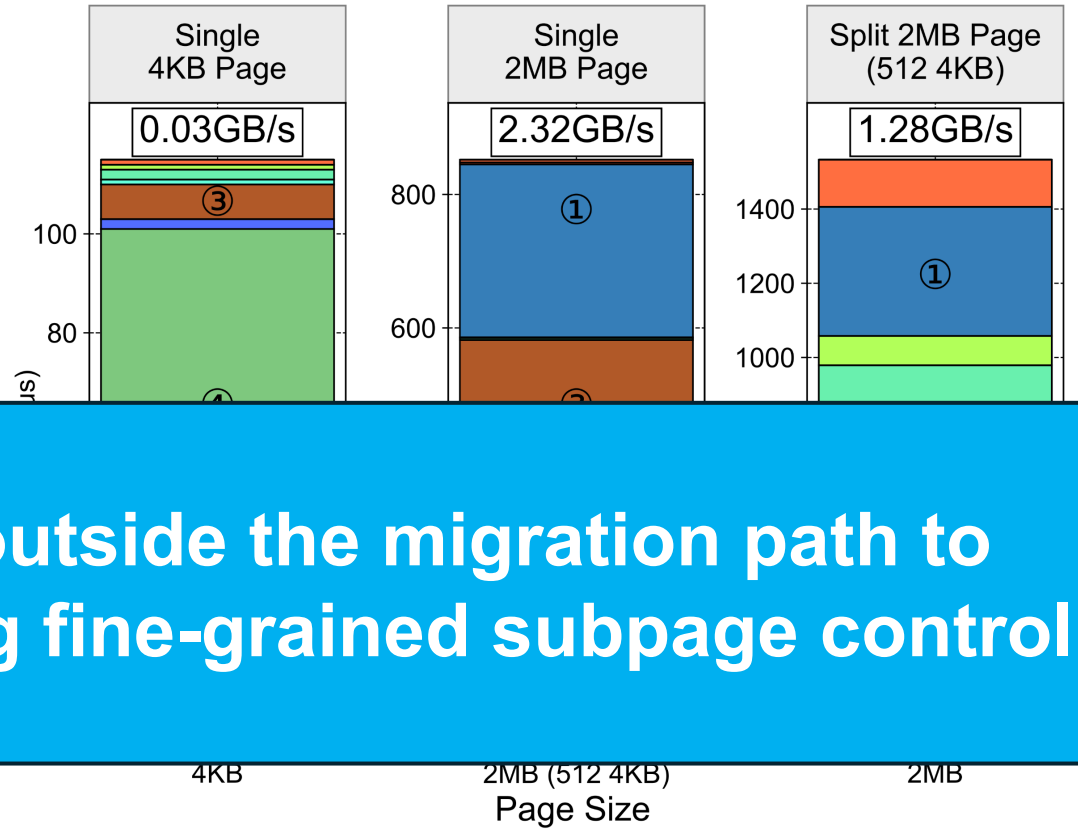


Motivation: Existing Solutions Fall Short

Problem ① — Slow Migration Throughput

✗ TPP's page-fault-based migration achieves only 4–16 MB/s

We benchmark three migration strategies on our CXL platform (Linux 6.9.0):



The insight: Pre-split hugepages outside the migration path to recover split overhead while retaining fine-grained subpage control.

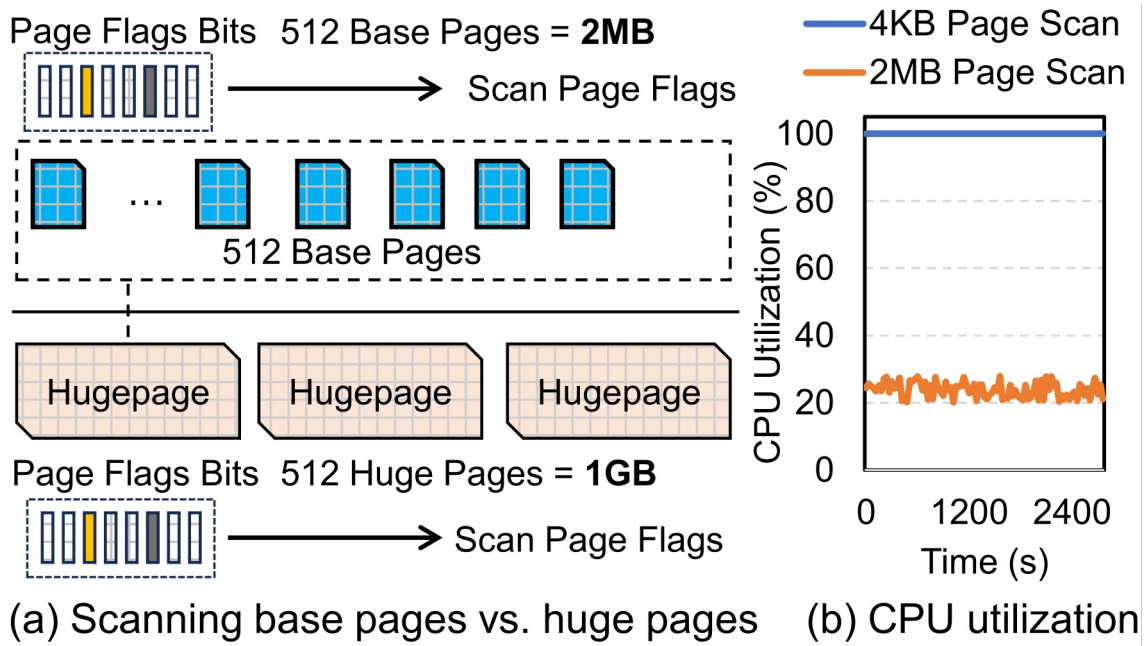
level throughput

- Split-then-migrate incurs additional locking cost (1.28 GB/s), but enables fine-grained subpage migration — a necessary trade-off for avoiding memory bloat

Motivation: Existing Solutions Fall Short

Problem ② — Hugepage-induced Memory Bloat & Scanning Trade-off

- ✗ 2MB THP improves migration throughput (2.3 GB/s) but brings a new problem: not all 512 subpages within a hugepage are hot
- ✗ THP causes memory bloat: 2MB hugepages migrate cold subpages to DDR, wasting fast-tier DDR space
- ✗ CXL lacks PEBS support: Systems like Memtis and FlexMem rely on PEBS, which is infeasible on CXL memory
- ✗ No existing CXL-compatible solution achieves subpage-level hotness tracking



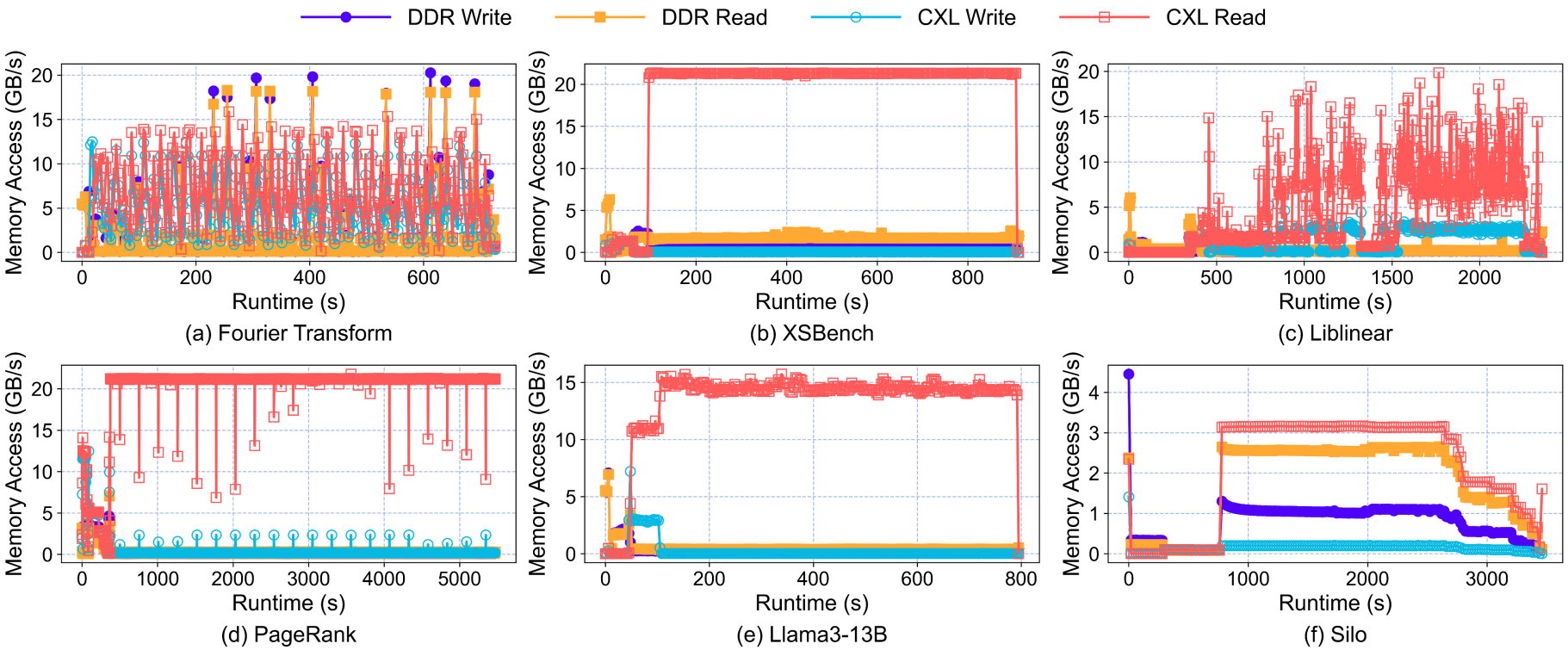
(a) Scanning base pages vs. huge pages (b) CPU utilization

The dilemma: 4KB scanning can track subpage hotness but is prohibitively expensive; 2MB scanning is lightweight but blind to subpage-level access pattern

Motivation: Existing Solutions Fall Short

Problem ③ — TPP's Inability to Correct Placement

- ✗ Under 1:1 DDR-CXL configuration, CXL handles the majority of read traffic despite DDR's 6× bandwidth advantage
- ✗ TPP fails to proactively promote hot pages to DDR, leaving bandwidth imbalance unresolved



Motivation: Key Insights

To solve the above problems, we need a design that simultaneously achieves:

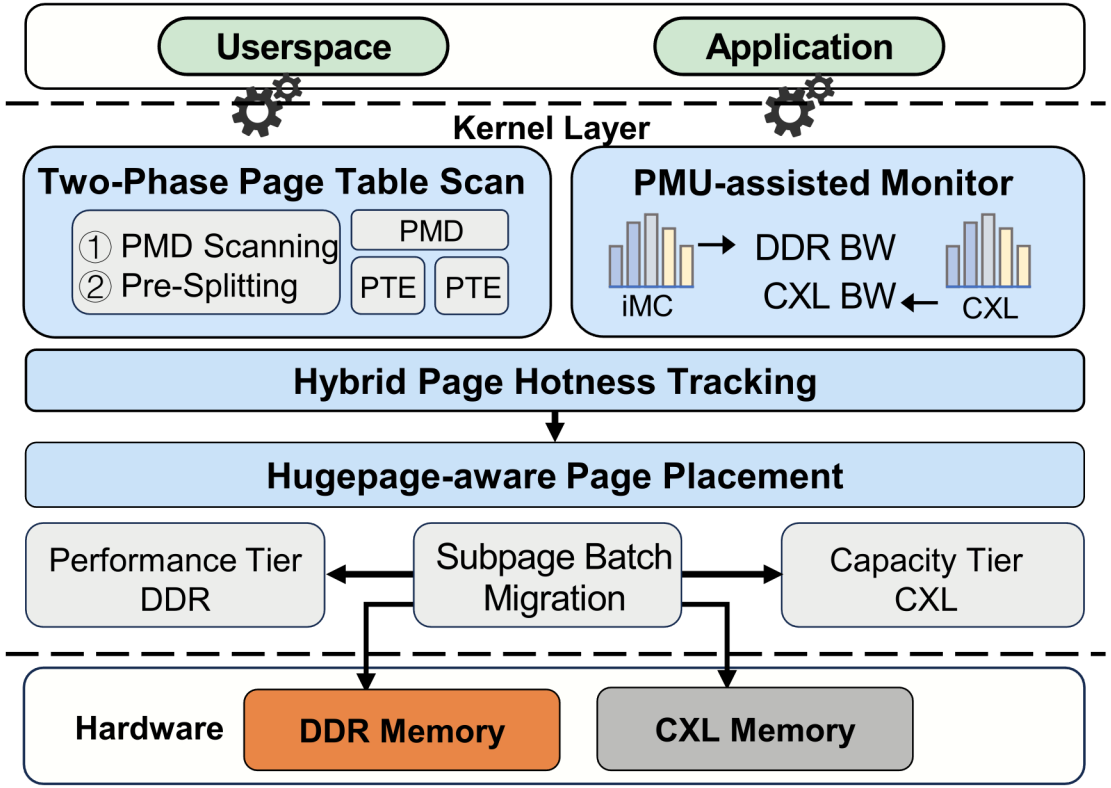
- ✓ Fast migration — batch subpage migration and support hot-only migration
- ✓ Fine-grained tracking — subpage-level hotness without PEBS
- ✓ Low overhead — avoid continuous 4KB PTE scanning costs

Core idea: Combine coarse hugepage scanning (efficient) with selective fine-grained scanning (precise), and use PMU feedback to adapt scanning intensity

Design Overview

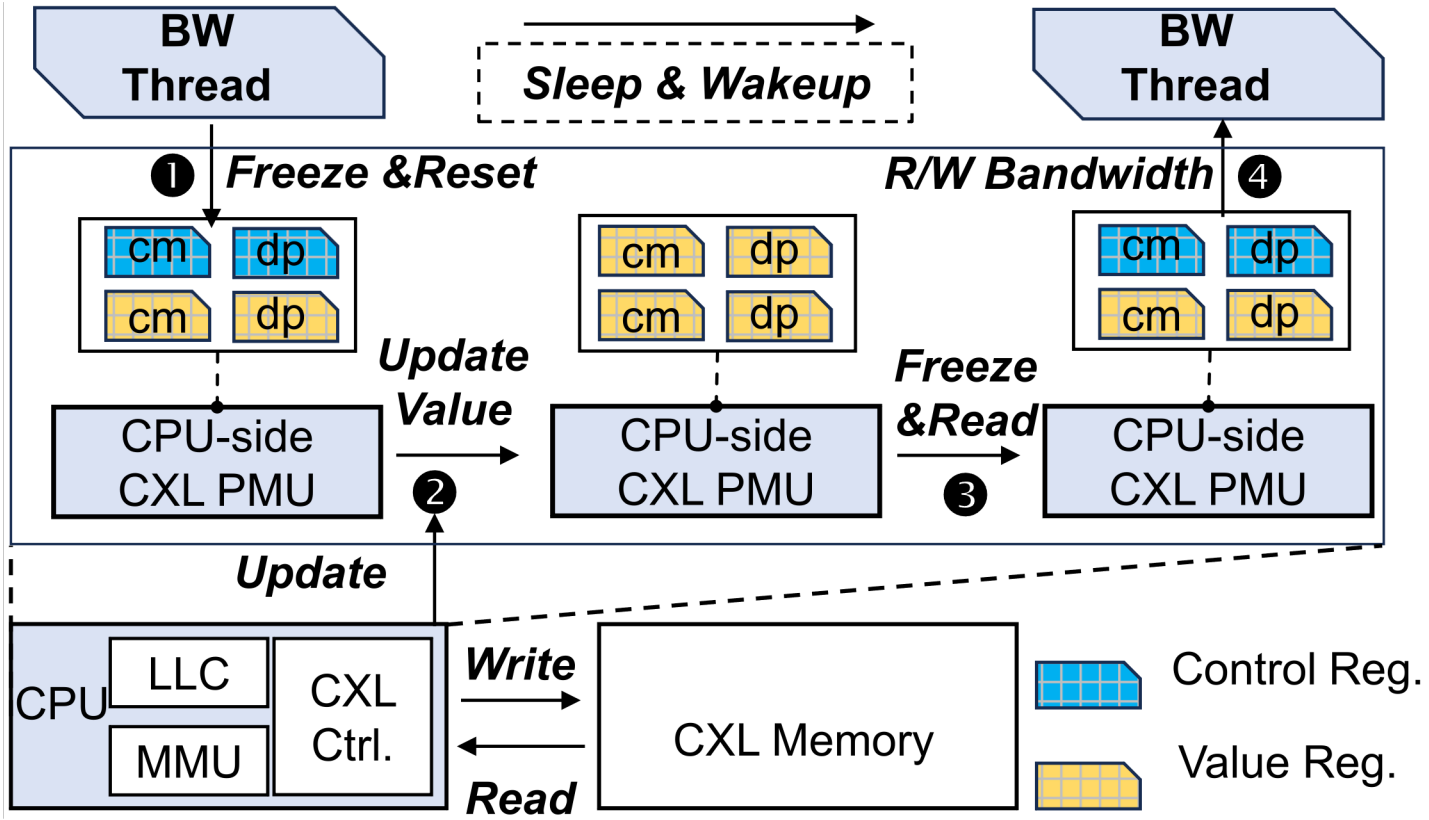
TPage: Transparent Page Management for CXL-Based Tiered Memory Systems

1. PMU-assisted Memory Imbalance Monitor
   — when to scan
2. Two-phase Page Table Scan
   — how to scan efficiently
3. Hybrid Page Hotness Tracking
   — how to measure hotness
4. Hugepage-aware Page Placement
   — how to migrate smartly



PMU-assisted Memory Imbalance Monitor

- CPU-side iMC PMUs monitor local DDR bandwidth;
- CPU-side CXL PMUs (CM/DP registers) monitor CXL bandwidth
- Define pressure ratio $R_B = B_{DDR}/B_{CXL}$ and space ratio $R_S = S_{\mathrm{DDR}}/S_{\mathrm{CXL}}$

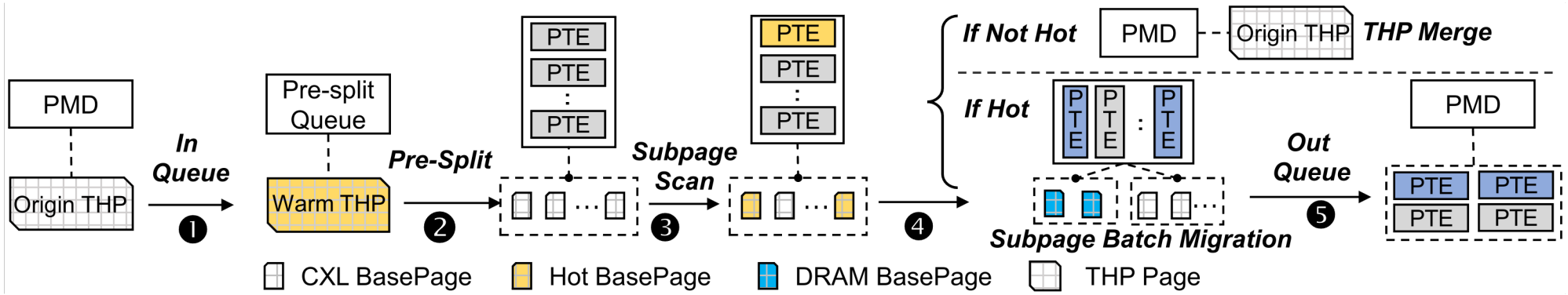


Key Advantages:

- Off critical path
- Minimal overhead
- CXL-agnostic
- Enables adaptive scanning

Two-phase Page Table Scan

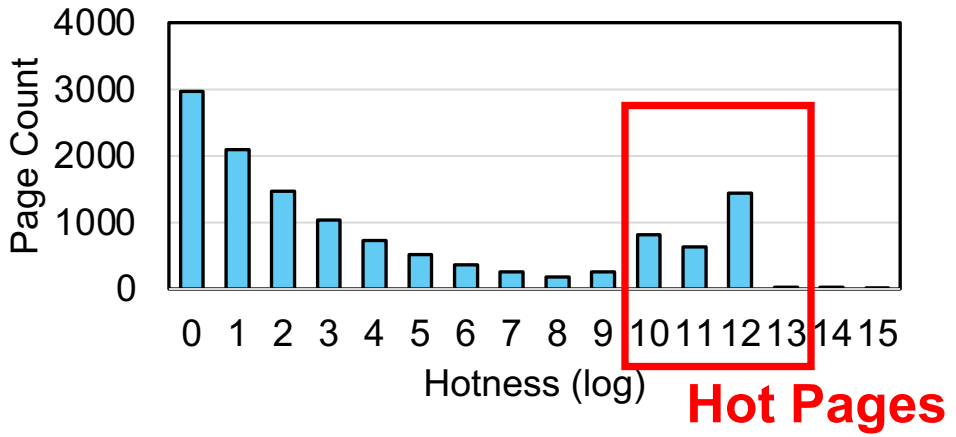
- Phase 1 (Coarse Scan) ❶: Scan at 2MB hugepage granularity via PMD access bits — quickly locates hot hugepage candidates with low CPU overhead
- Pre-splitting (❷) is executed proactively on the pre-split queue, decoupled from the critical migration path
- Phase 2 (Fine Scan) ❸: Pre-split hot candidates, then perform PTE-level scanning within a time window to capture per-subpage hotness



Hybrid Page Hotness Tracking

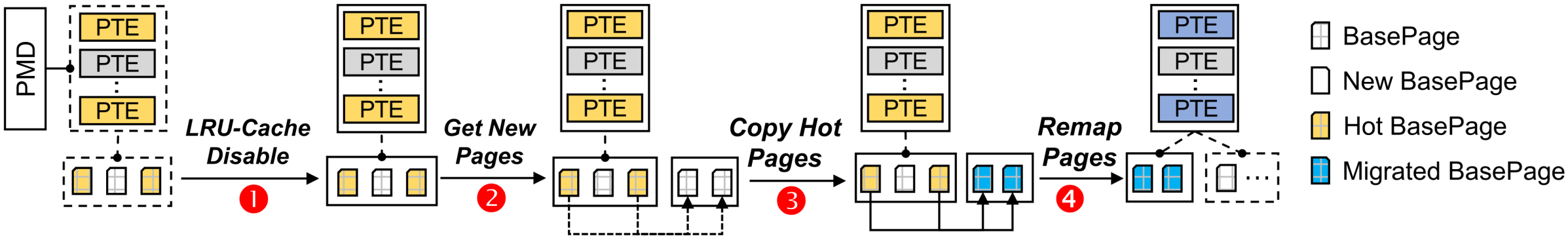
- Hotness metadata stored in unused fields of struct page entries; Phase 2 subpages use page_ext mechanism
- Access frequency histograms with logarithmically scaled bins (2^n boundaries)
- **Full-coverage scanning** — every page is observed in each scan cycle, so no histogram decay is needed (unlike PEBS-based Memtis/FlexMem which only sample partial accesses)
- DDR and CXL maintain independent histograms, providing direct basis for migration decisions





Hugepage-aware Page Placement

- Integrates **two-phase scan** results and **hotness tracking** to orchestrate tiered page movement
- CXL → DDR (promotion): **Hot subpages batch-migrated** to DDR; cold regions stay on CXL
- DDR → CXL (demotion): Evaluate hot/cold ratio — high hot fraction stays in DDR; cold subpages demoted to CXL
- Lazy hugepage recovery: After cold subpages are demoted, the 2MB region becomes physically contiguous on CXL, enabling re-collapse into a hugepage



Evaluation Setup

- Platform: Intel Xeon Gold 6430 (2×32 Core), DDR5 4800MHz, ASIC-based CXL Expander (PCIe 5.0), 21 GB/s CXL bandwidth per CPU
- Software: Linux 6.9.0, ~4,100 lines of kernel code
- Benchmarks:
  - PageRank (40GB)
  - Fourier Transform (80GB)
  - Silo (80GB)
  - Llama3-13B (30GB)
  - XSBench (63.4GB)
  - Liblinear (68GB)
- Baselines: CXL-Mem, AutoNUMA, TPP, Tiering-0.8, Nomad, Colloid

TABLE II: The Configuration of System Environment.

CPU	Intel Xeon Gold 6430	2 × 32 Core	
Memory	8 × 32GB / DDR5	4800 MHz	
ASIC-based CXL Expander	8 × 32GB / DDR5	PCIe 5.0 / MCIO Interface	
OS	Ubuntu v22.04	Kernel	v6.9.0

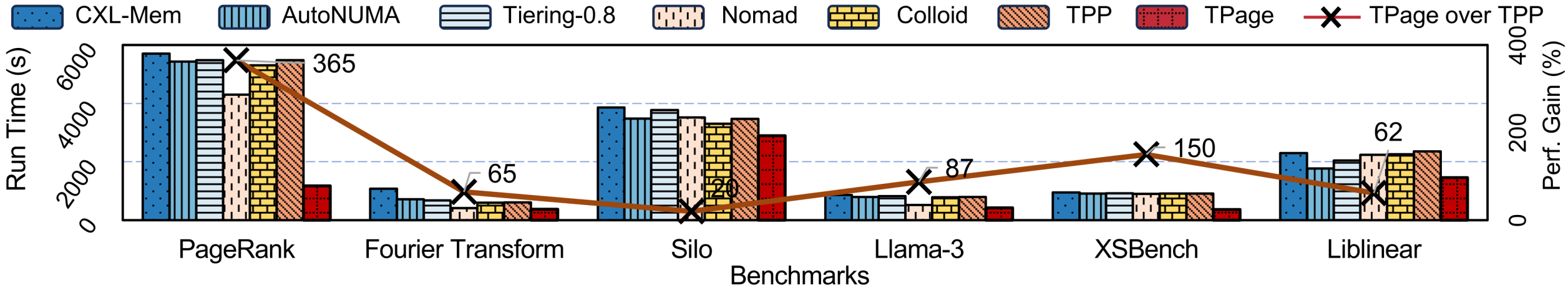
TABLE III: Benchmark Characteristics.

Benchmark	RSS	Description
PageRank [3]	40GB	Compute PageRank score of a graph
Fourier Transform [30]	80GB	Bandwidth-consuming matrix transpose
Silo [46]	80GB	In-memory database engine
Llama3 [13]	30GB	Large language model (Llama3-13B)
XSBench [45]	63.4GB	Computational kernel of the Monte Carloneutron transport algorithm
Liblinear [11]	68GB	Linear classification of a large data set

RSS : Resident Set Size in Memory

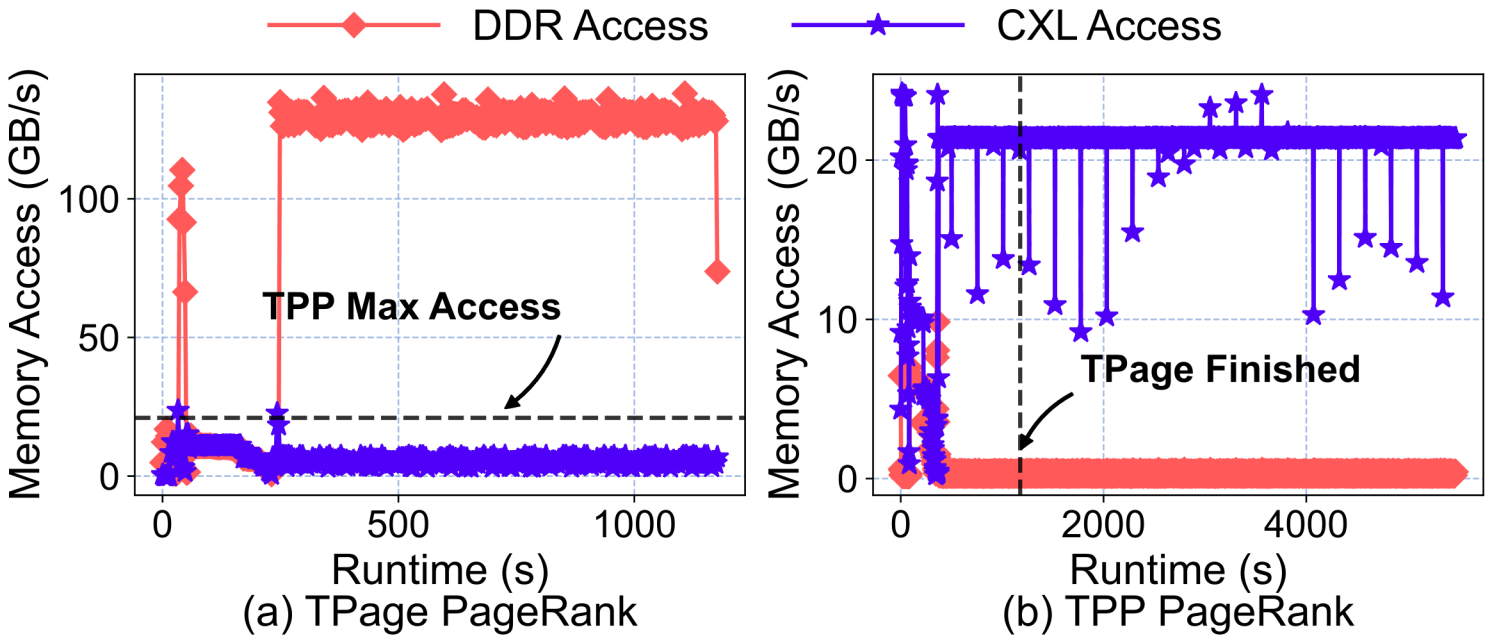
Overall Performance

- TPage consistently outperforms all baselines across all workloads
- Speedup over TPP (mainline Linux): PageRank 4.65×, Fourier 1.65×, Silo 1.2×, Llama3 1.87×, XSBench 2.5×, Liblinear 1.62×
- Nomad outperforms TPP on several workloads but requires 4-level page tables, conflicting with default 5-level paging on modern servers



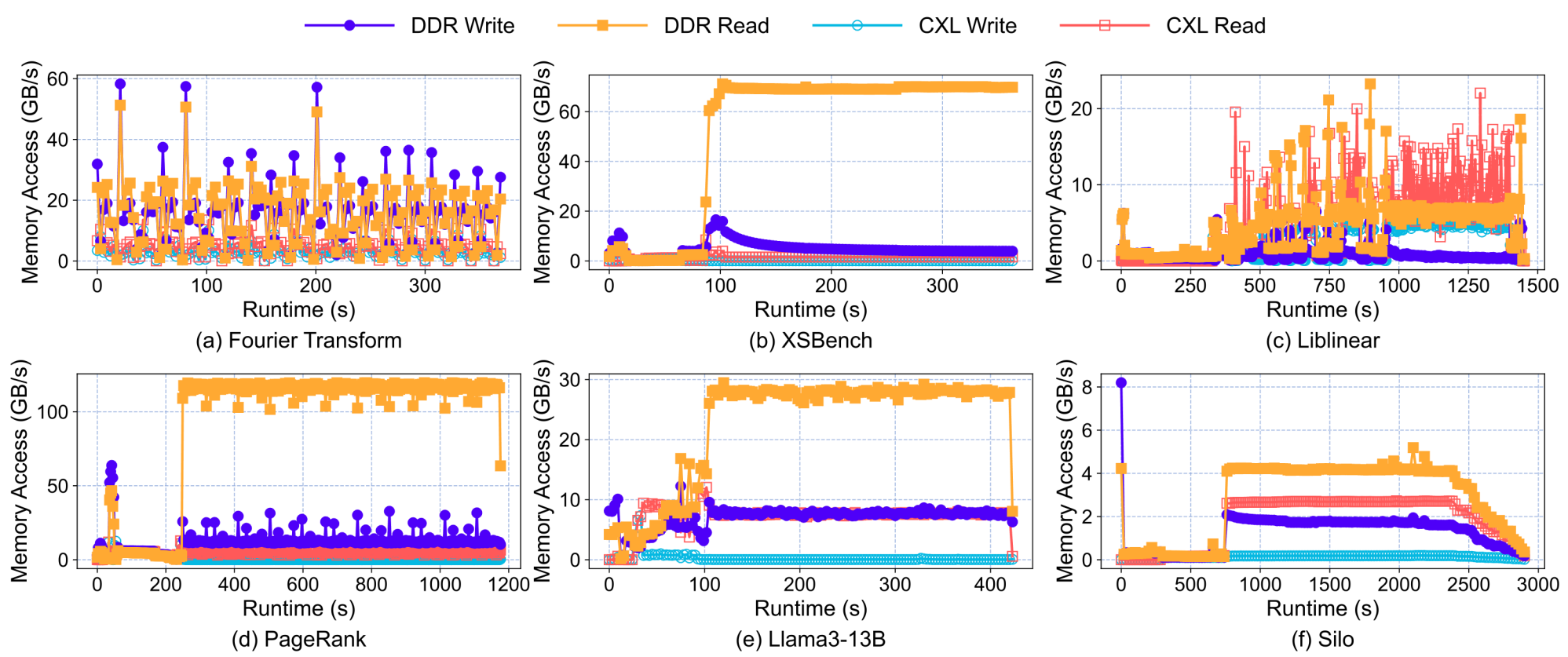
Case Study — PageRank Deep Dive

- PageRank shows the most pronounced gain (4.65× over TPP)
- TPage completes most migrations within the first 60 seconds (graph construction phase), corresponding to the first DDR access peak
- TPP still shows a dominant CXL access peak during this stage
- In the subsequent training phase, TPage sustains high DDR access with no further migrations
- TPP fails to migrate critical pages, resulting in persistent CXL access and suboptimal bandwidth u



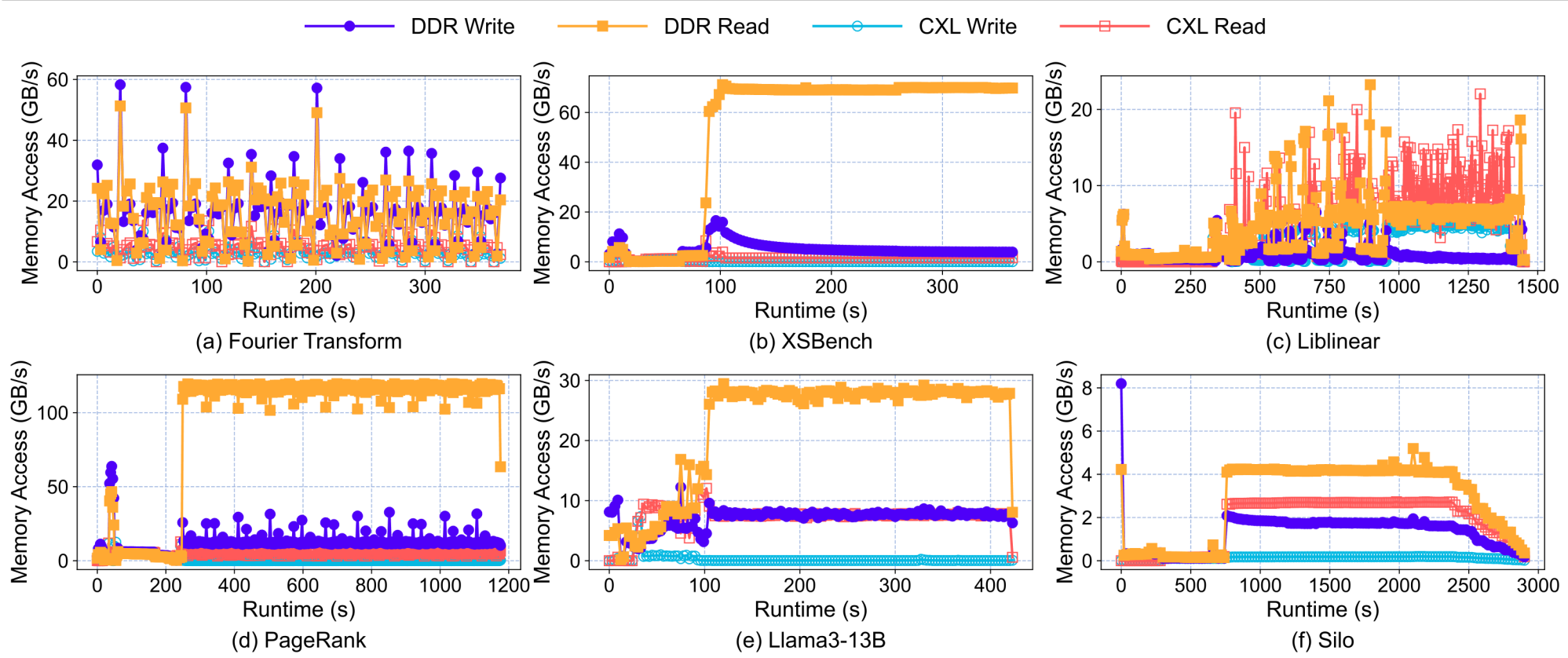
Memory Access Analysis

- Compared to TPP (Slide 7), TPage successfully redirects most read/write traffic to DDR
- CXL primarily serves as a capacity tier with supplementary bandwidth
- Validates TPage's hot page promotion strategy works in practice



Memory Access Analysis

- Compared to TPP (Slide 7), TPage successfully redirects most read/write traffic to DDR
- CXL primarily serves as a capacity tier with supplementary bandwidth
- Validates TPage's hot page promotion strategy works in practice



Limitations and Discussion

- **PEBS Compatibility**: TPage can seamlessly integrate PEBS as it becomes available on future CXL devices; PEBS + PTE-scan + PMU-based adaptation can further reduce overhead
- **Multi-Node Scalability**: CPU-side PMUs only, natively compatible with CXL 2.0 multi-host (MLD + CXL Switch provide per-host hardware isolation)
- **Hybrid Mapping TLB Pressure**: Mitigated by batch migration, lazy re-collapse, PMUs-driven Scan

Conclusion

- TPage: transparent page management for CXL-based tiered memory

- Four innovations:
  - ✓ PMU-assisted imbalance monitor → adaptive scanning frequency
  - ✓ Two-phase PTE scan → efficient + fine-grained hotness detection
  - ✓ Hybrid hotness tracking → full coverage, no decay
  - ✓ Hugepage-aware placement → batch subpage migration, no bloat

- 3.6× speedup over TPP, 2.44 GB/s migration throughput on real CXL hardware

SC26 CHICAGO NOV 15–20

TPage: Transparent Page Management for CXL-Based Tiered Memory Systems

Thank you!

Maoxin Ye ★, Suzhen Wu ★, Jie Zhang $, Bo Mao ★

★ *Xiamen University* $ *Peking University*

Advanced Storage Technology Lab (ASTL): https://astl.xmu.edu.cn/index.htm