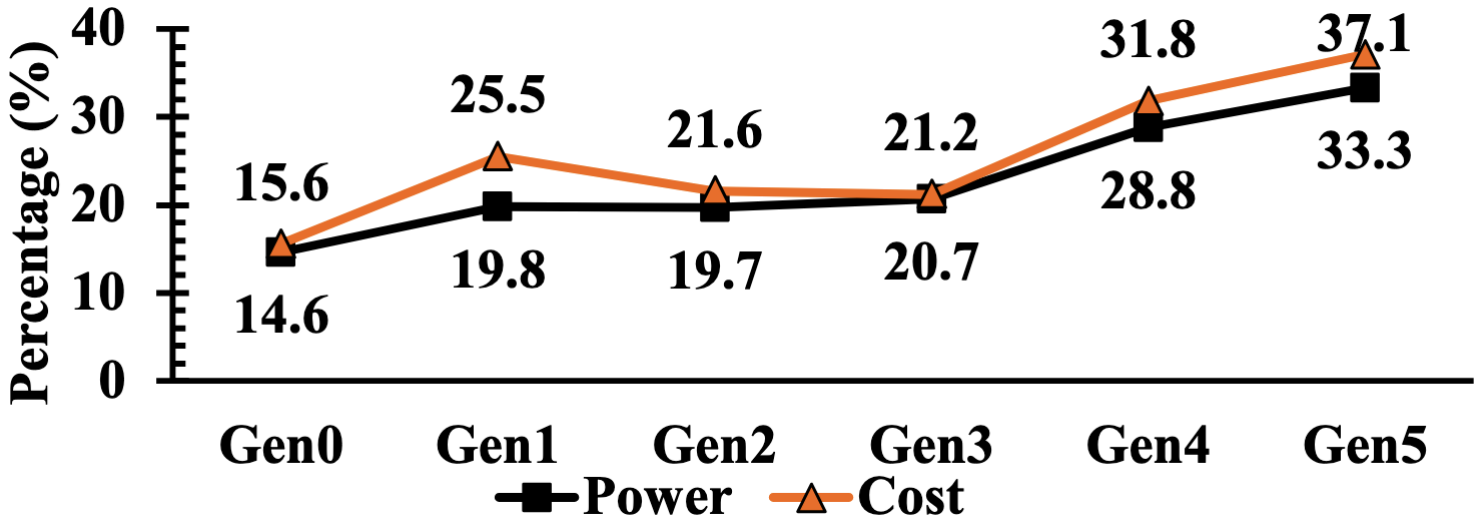
Pick Your Poison: Lightweight CXL Memory Tiering with ATLAS

TÉCNICO LISBOA

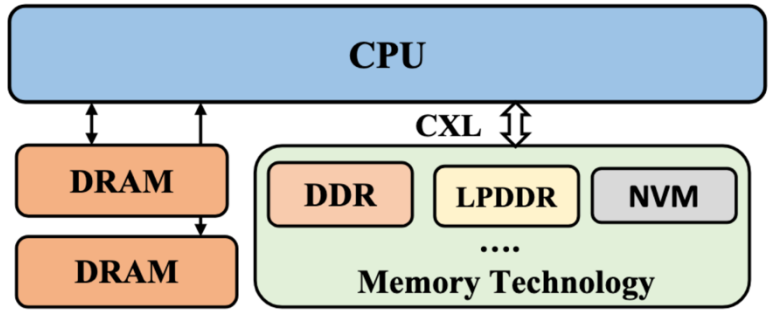
João Póvoas and João Barreto
INESC-ID & IST, Lisbon University

Towards tiered memory in the Cloud

Contribution of memory to a rack at Meta [Maruf et al., ASPLOS'23]



- VMs in data centers are demanding more and more memory
- DRAM-only hosts: limited capacity
- DRAM + CXL-memory hosts: an emerging trend to overcome this challenge



[Maruf et al., ASPLOS'23]

Tiered page placement

Which pages of an application's resident working set should be placed at the fast tier for optimal performance?

- An ideal solution:
  - Accurate
  - Responsive
  - Low-overhead, including tail latency overhead
- Hot topic in the OS community, many papers in top-tier confs.
  - Hemem [SOSP'21], TPP [ASPLOS'23], MEMTIS [SOSP'23], MDM [Eurosys'24], NOMAD [OSDI'24], Colloid [SOSP'24], Chrono [Eurosys'25], etc.

Anatomy of tiered page placement systems

- Page table entry (PTE)-based:
  - PTE scanning (access bits)
  - PTE poisoning
- ~~Access sampling (e.g., Intel PEBS)~~

Tiered AutoNUMA

- Today’s industry standard, based on TPP [ASPLOS’23]
- Pages in fast tier:
  - Linux’s existing LRU-based page reclamation daemon (kswapd)
  - Based on PTE scanning
  - When free space below *low watermark*, some *inactive* pages demoted to the slow tier

Tiered AutoNUMA

- Pages in slow tier:
  - Periodically, **PTE-poison** a random sample of pages in the slow tier
  - Access to poisoned page → page fault → access to poisoned page → page fault → migrate page up (*)

  - (*) if both accesses close in time and enough free space in fast tier

Tiered AutoNUMA

- Pages in slow tier:
  - Periodically, **PTE-poison** a random sample of pages in the slow tier
  - Access to poisoned page → page fault → access to poisoned page → page fault → migrate page up (*)
  - (*) if both accesses close in time and enough free space in fast tier

Sample may miss active pages

Susceptible to ping-pong migrations

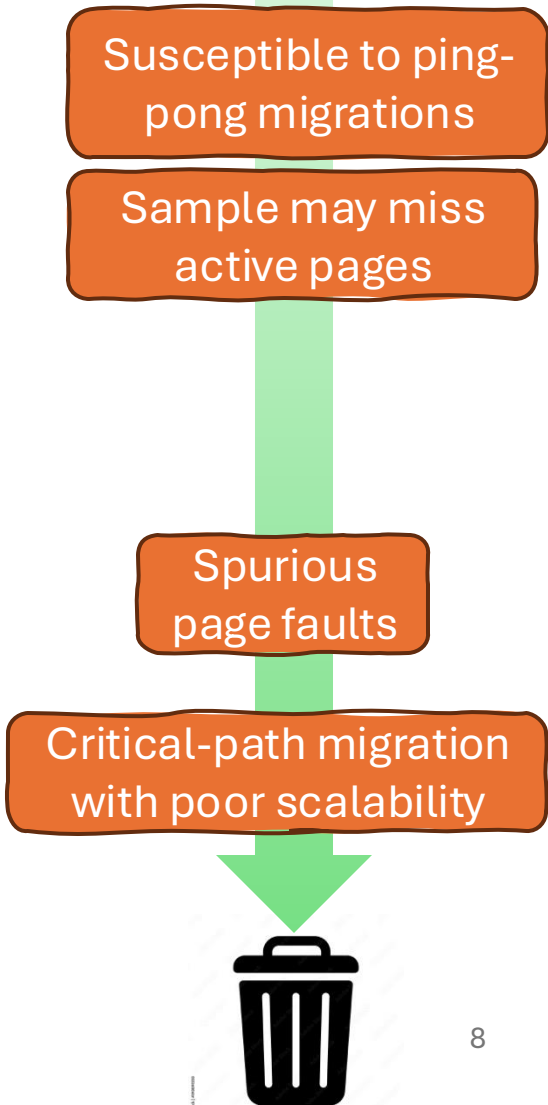
Critical-path migration with poor scalability

Spurious page faults

Aborted hot page promotions

ATLAS in one slide

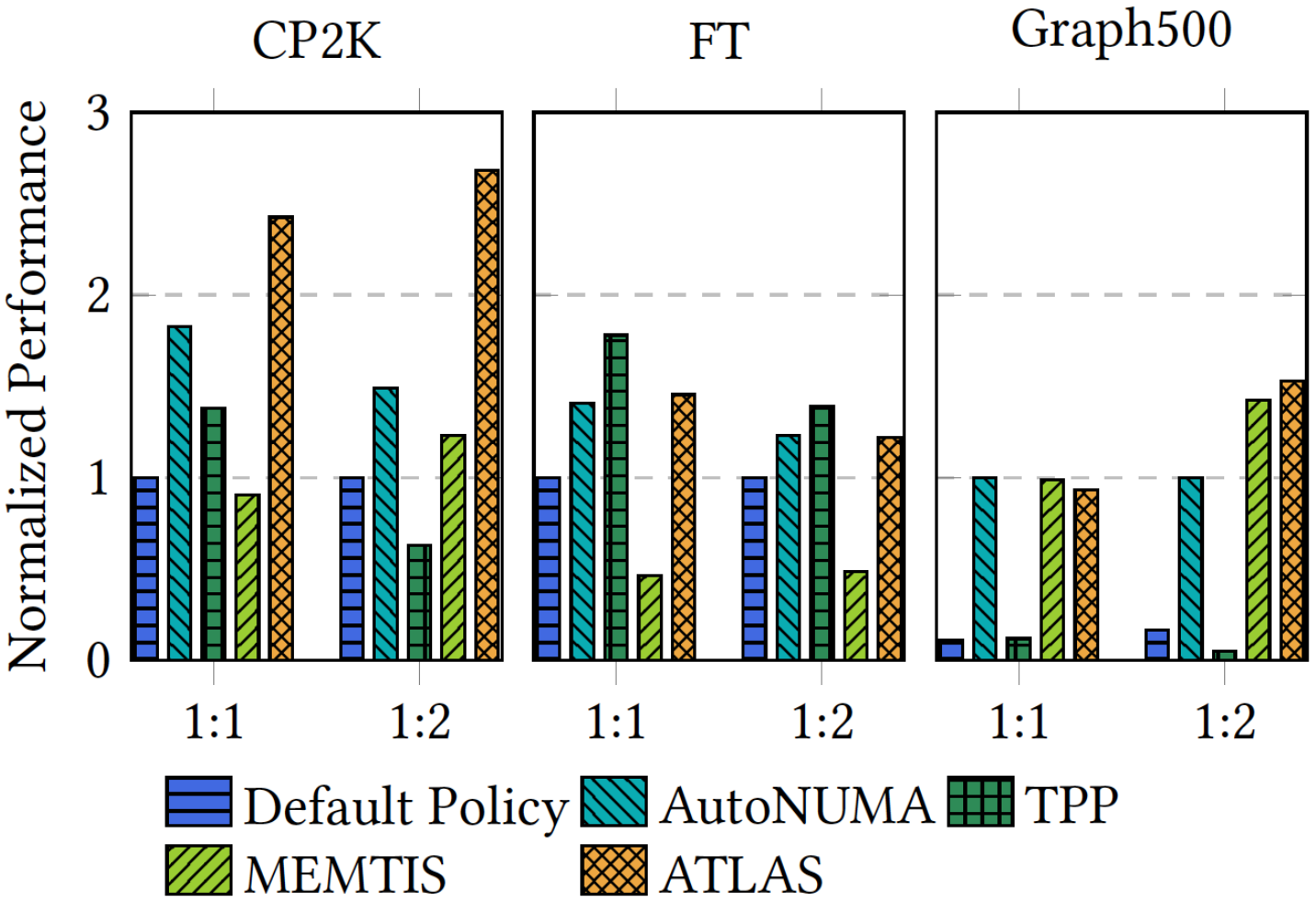
- Build an **tentative** hotness score via PMD scanning
- Poison pages with hotness score just below a dynamic promotion threshold
- When page fault, promote the page
  - This **accurately** confirms a true positive
- In background, we depoison inactive PTEs
  - I.e., false positives
- For poisoned pages, proactively execute the **unscalable** preparation steps of *migrate_pages*



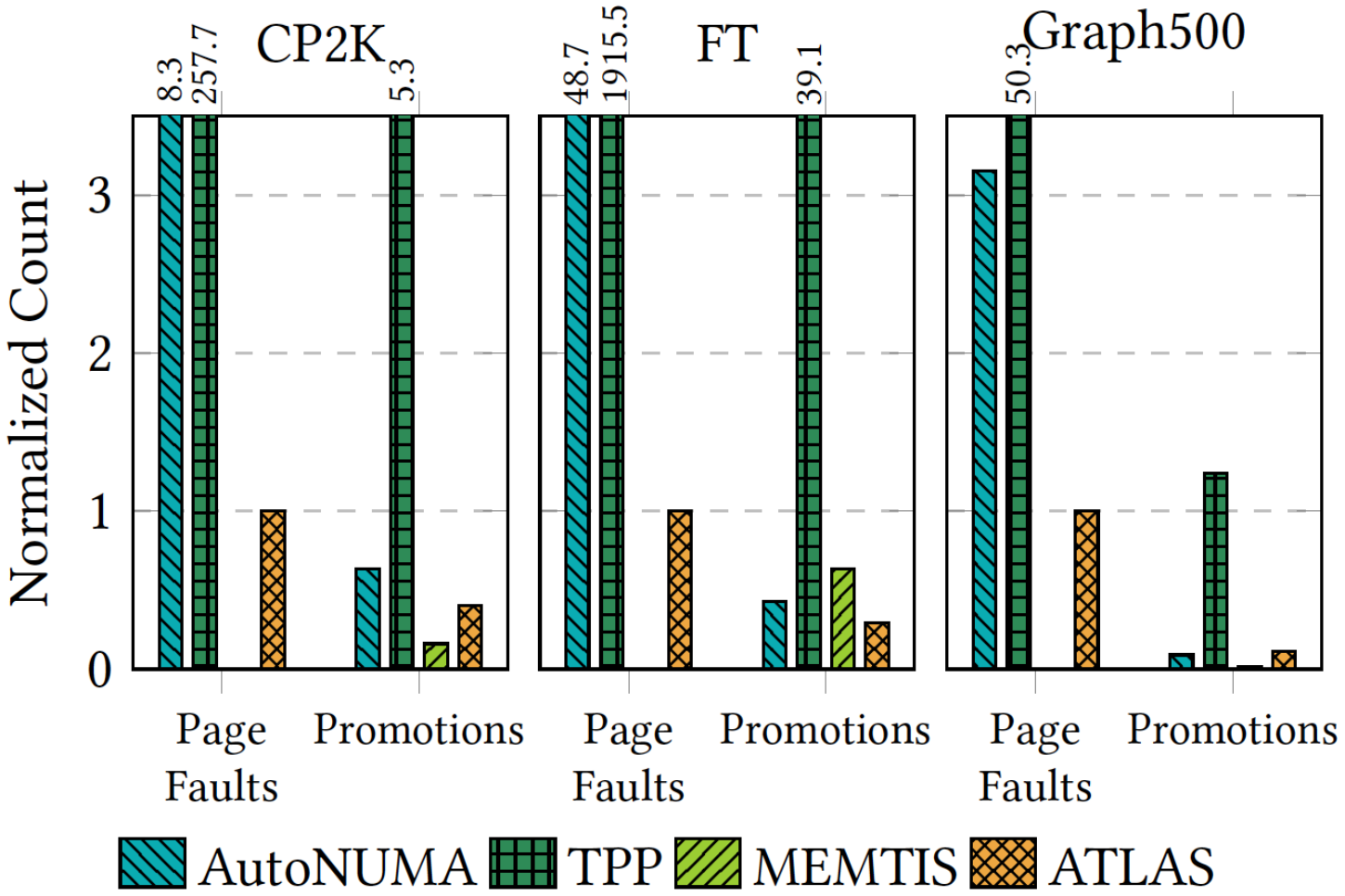
Early evaluation

- Prototype of ATLAS with the two-phase tracking/promotion policy
  - PMD scanning and proactive promotion preparation not considered
- “Baremetal” machine with DRAM + Optane, single-socket
  - To also compare with MEMTIS [SOSP’23], which is based on hw. event-based sampling

End-to-end performance



Migration volume



Take-away slide

- Tiered memory are emerging in the Cloud
- Tiered page placement should be accurate, responsive, low-overhead (including tail latency overhead)
- Today's standard solution, AutoNUMA, has important shortcomings
- ATLAS overcomes them with a novel **two-phase page promotion policy** and **scalable critical-path page promotion**

joao.barreto@tecnico.pt

Acknowledgements

- Supported by:
  - FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020 (DOI:10.54499/UIDB/50021/2020)
  - The Portuguese Recovery and Resilience Plan through project C645008882-00000055 (Center for Responsible AI)
  - European Commission through the Horizon Europe Programme, with the Grant Agreement GAP-101189689.