Morpheus: An Adaptive DRAM Cache with Online Granularity Adjustment for Disaggregated Memory

Xu Zhang, Tianyue Lu, **Yisong Chang**, Ke Zhang, Mingyu Chen

State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences

University of Chinese Academy of Sciences

Executive Summary

- Existing hardware-DRAM-cache-based memory disaggregation system faces severe data over-fetching
  - Obs: An average of 82.6% of cached pages are only touched for <4 blocks (256B per block)
- We propose Morpheus DRAM cache
  - Dynamically selects coarse or fine cache granularity for each page
  - Adaptively adjusts DRAM space occupied by each granularity
- Morpheus exhibits 1.17-1.34x performance speedup and reduces the percentage of inefficient pages to 45.6%

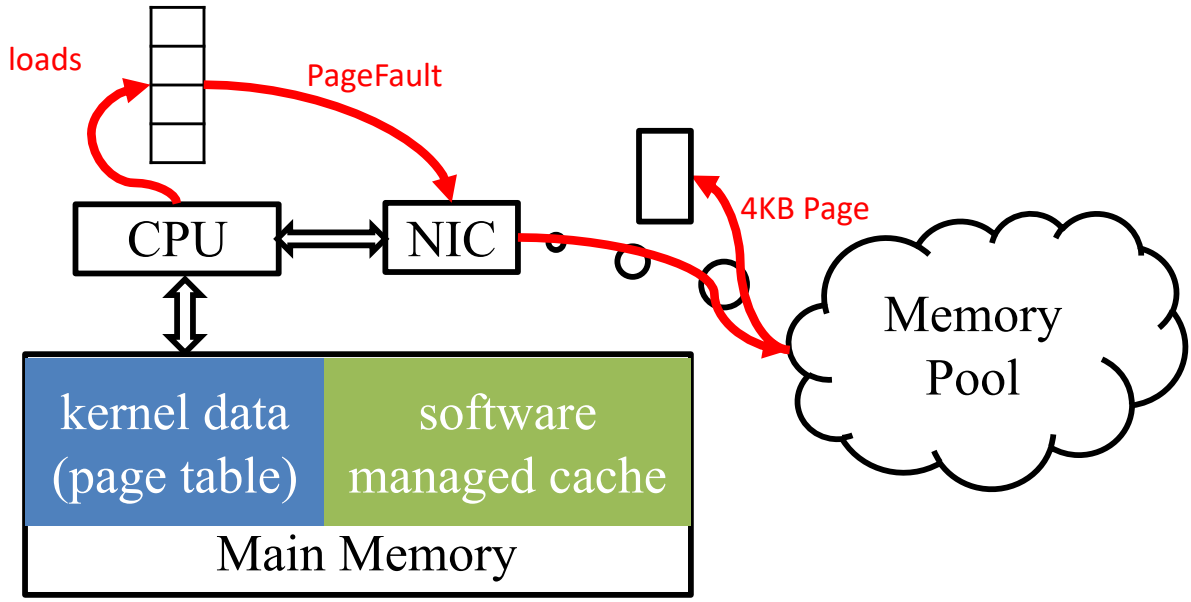
中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

Outline

- Background
- Motivation
- Morpheus
- Evaluation

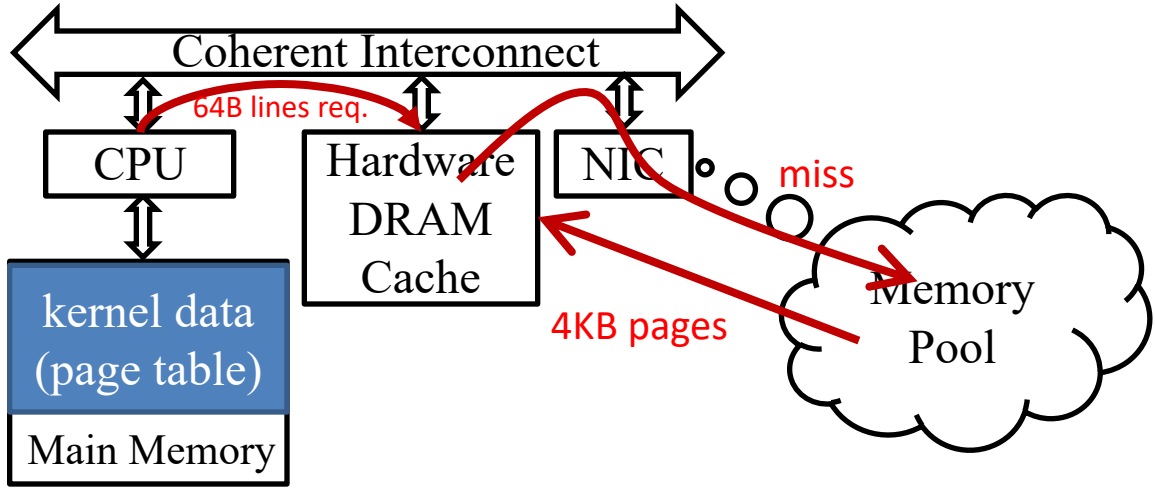
Partial memory disaggregation system

- Each CPU has a-few-GB local memory to alleviate latency overhead introduced by network
  - Storing kernel data
  - Caching hot data
- Software-managed cache
  - APIs-based at the user level
  - Paging-based at the kernel level



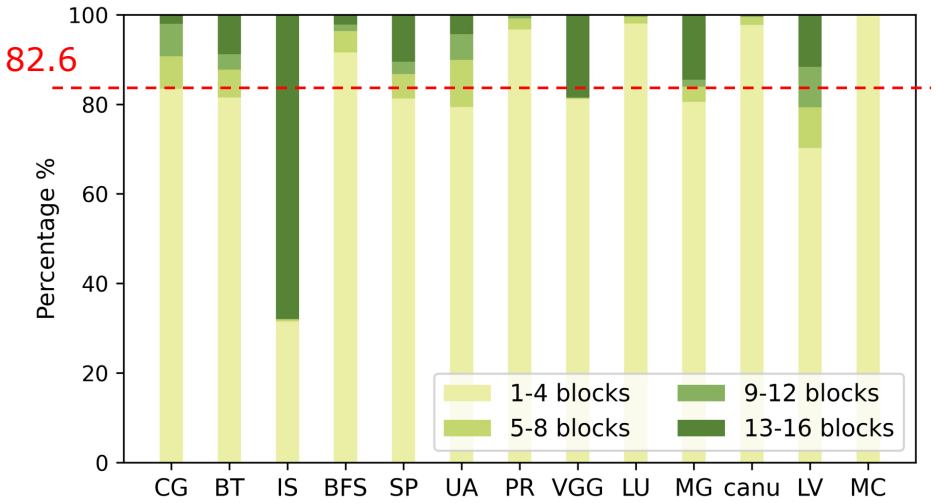
Key benefits of the hardware approach

- Coherence-based hardware-managed cache
  - CPUs connect to DRAM Cache Card via coherent interconnect
  - Exposing large logical address space backed up with memory pool
- A single LD/ST can initiate DRAM cache looking up and fetching remote data
  - Transparent to user applications
  - Avoiding severe page faults overhead



The Key challenge of the hardware approach

- Existing hardware-DRAM-cache-based memory disaggregation system[1] (named Kona) faces severe data over-fetching
  - Obs: An average of 82.6% of cached pages are inefficient
    - only touched for <4 blocks (256B per block)
  - Causes
    - To leverage spatial locality, DRAM Cache fetches pages instead of lines in case of cache misses



[1] I. Calciu, M. T. Imran, I. Puddu *et al.*, "Rethinking Software Runtimes for Disaggregated Memory," in *Proc. of ASPLOS*, 2021.

Outline

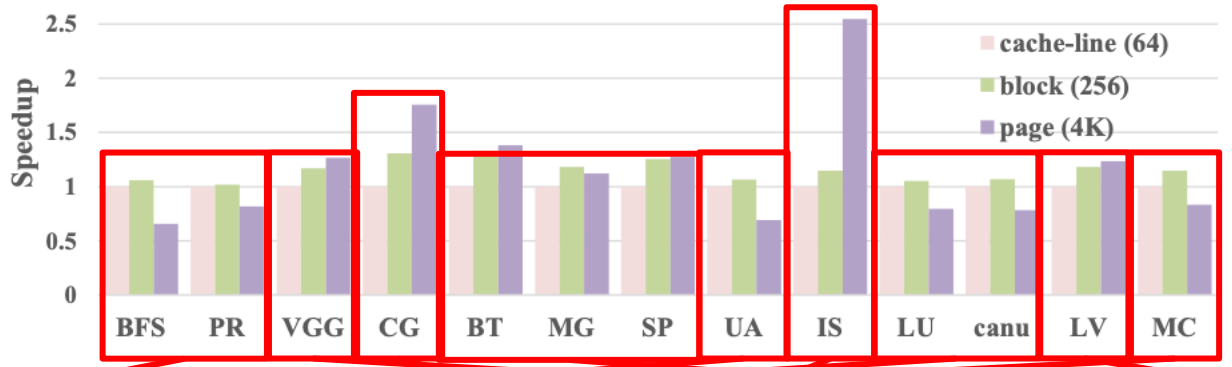
- Background
- Motivation
- Morpheus
- Evaluation

Preferred cache granularities of various apps.

- Finer granularity will avoid over-fetching
- To answer whether there is one-size-fits-all granularity
- We select 13 memory-intensive applications and 3 cache granularities
  - Machine Learning, Graph Processing, Database, etc.
  - Fine (64B), Coarse (4KB), Moderate (256B)

Clustering apps. into 3 categories

- There is no one-size-fits-all granularity
  - But there is one-size-fits-one-category granularity



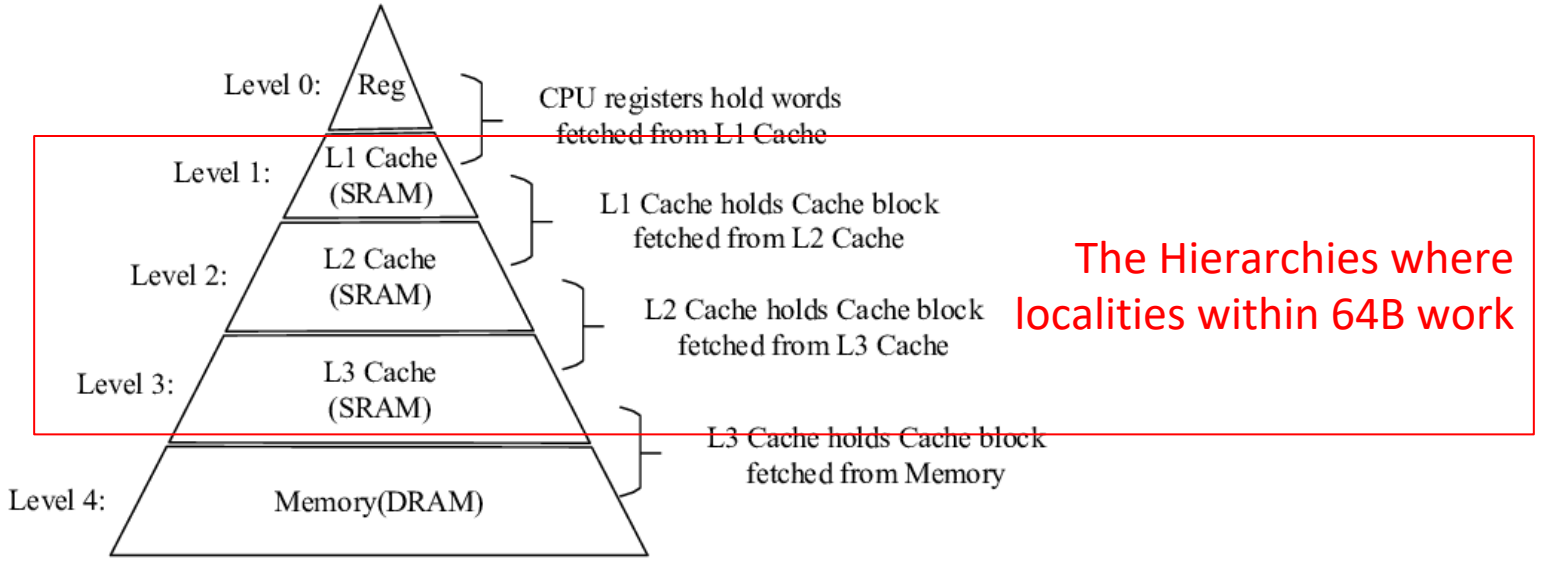
Temporary-locality-only
Preferring block (256B) granularity as they have hot spots in pages.

Spatial-locality-only
Preferring page granularity as they tend to sequentially access large continuous data.

Both-localities
There exist both hot spots and continuous data in these applications.

How to design for 3 categories

- Key observation 1: DRAM cache does not need fine granularity
  - block size (256B) is always better than conventional cache-line size (64B)
  - Localities within 64B are fully exploited before the DRAM cache



X. Gao, L. Huang, J. Jiang and F. Qi, "CSPM: A Coordinated Software Prefetching Mechanism For Multi-Level Caches," ICCCS`22

How to design for 3 categories

- Key observation 2: both moderate and coarse granularities are useful
  - The apps. preferred moderate granularity
    - Presenting high temporal locality
    - Remote fetching latency is reduced, and network bandwidth is saved
  - The apps. preferred coarse granularity
    - Presenting high spatial locality
    - Demanding one remote fetching for each page
  - The management granularity should be switchable

How to design for 3 categories

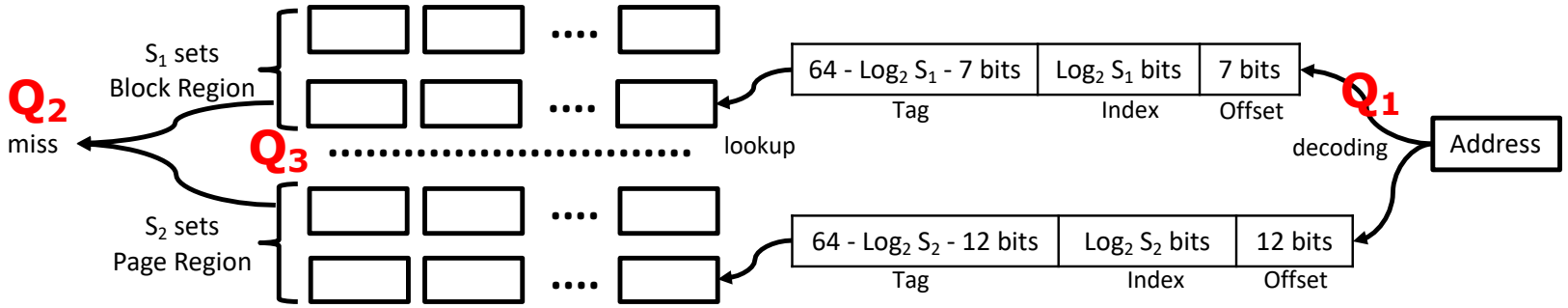
- Key observation 3: There is no consistent granularity for each page
  - Both-locality apps. have pages preferring different granularity
  - Even one page may present different granularity at different time epochs
  - 2 granularities should be allowed to exist at the same time

Outline

- Background
- Motivation
- Morpheus
- Evaluation

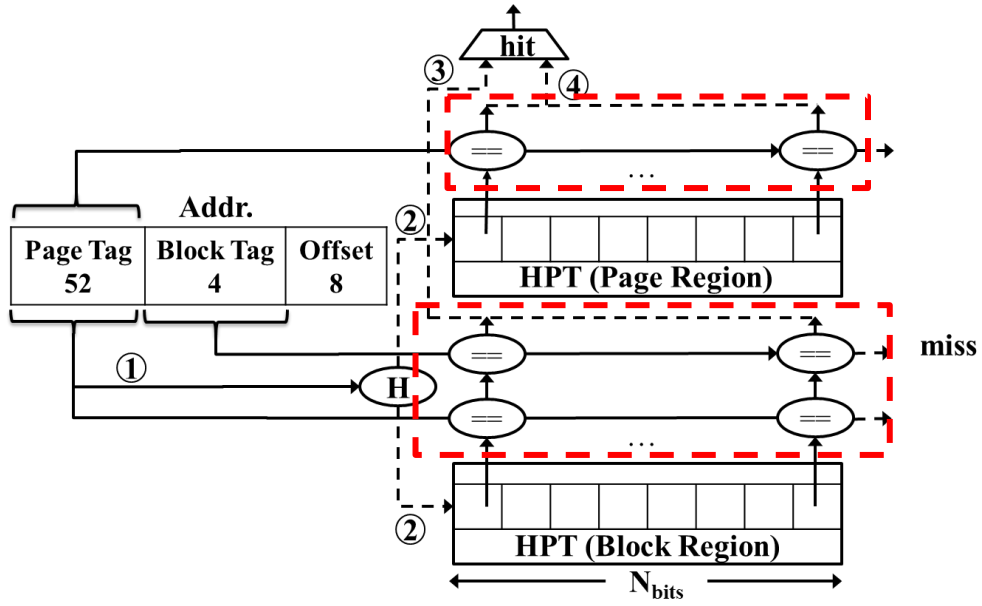
Challenges

- Morpheus aims to fulfill the above key observations
  - 2 regions managed with block and page respectively
  - 3 challenges need to be tackled
- Q1: How to identify cache tag/index in the address?
- Q2: How to decide the granularity for one cache miss?
- Q3: How to dynamically adjust the capacity?



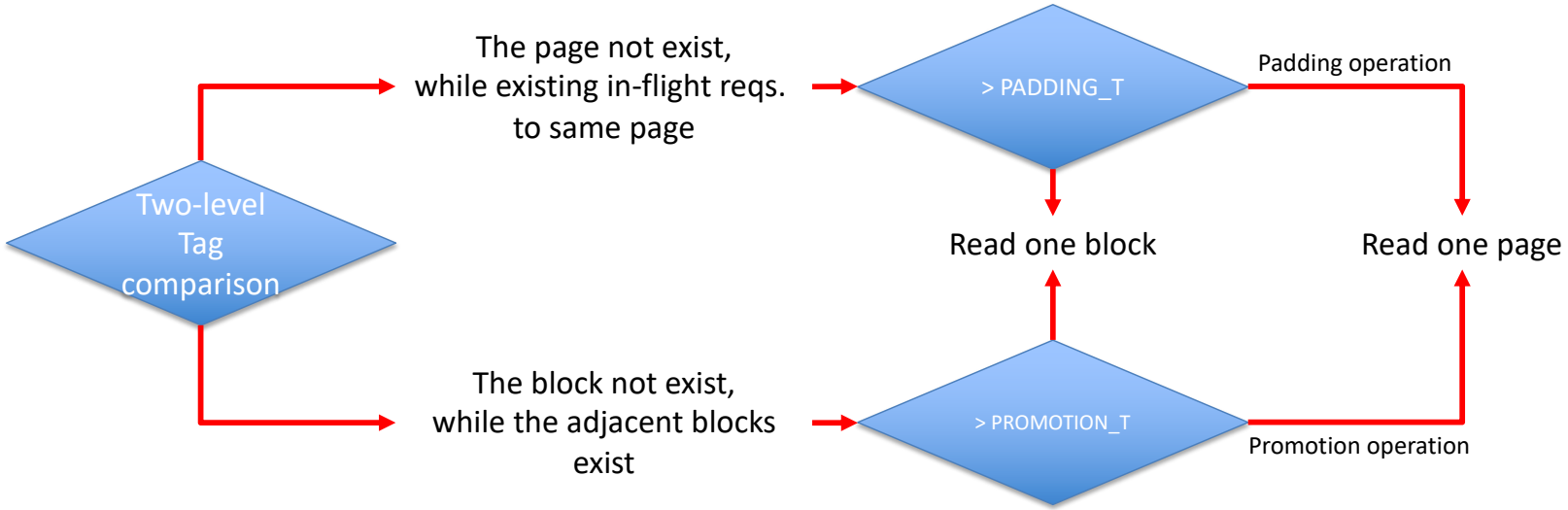
Q1: How to identify cache tag/index in the address?

- level-1: hash page table
  - Indexed with hashed **Page Tag**
    - fixed 52 most significant bits of physical address
  - Storing tag and unfixed address to the region
- level-2: tag comparison
  - Generate hits only if there exists mapping in the hash page table



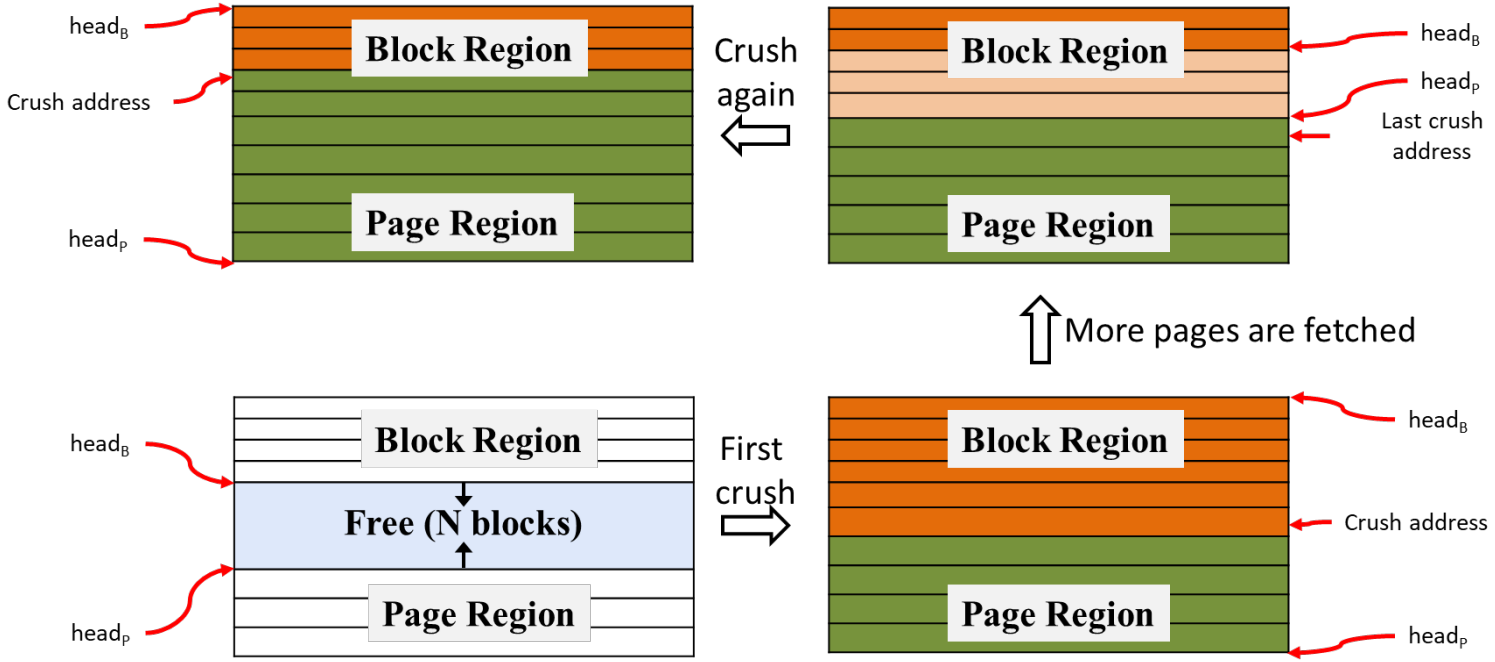
Q2: How to decide the granularity for one cache miss?

- Applications are inclined to access adjacent data within the high-spatial-locality page in a relatively short time
  - Based on history statistics from tag comparison and MSHRs
  - Two adjustable thresholds



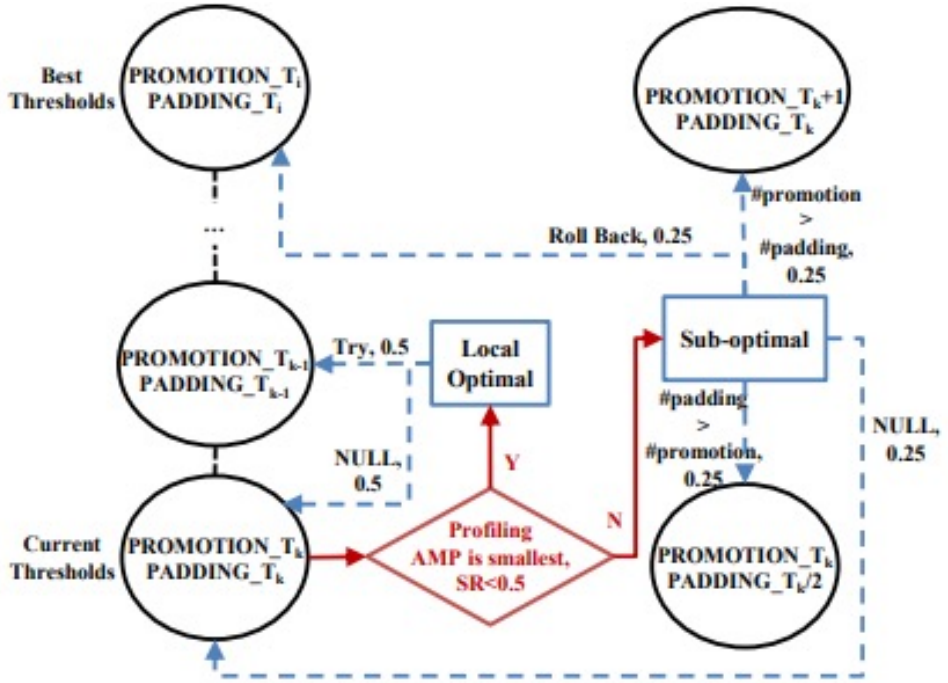
Q3: How to dynamically adjust the capacity?

- The fast-growing region evicts the space occupied by the other region
  - Two thresholds have an impact on the growth speed



Heuristic searching for thresholds

- Optimizing for fewer fetched data and higher performance
  - sparsity ratio (SR)
    - ratio of evicted pages with less than 2K bytes that have been touched
  - average miss penalty (AMP)
    - cache miss rate × average remote memory accessing latency



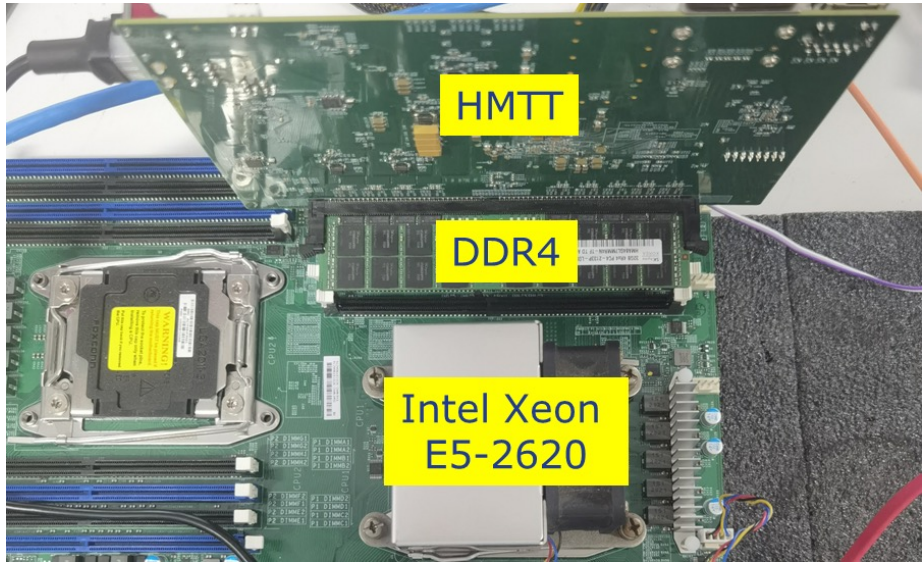
Outline

- Background
- Motivation
- Morpheus
- Evaluation

Porotype and parameters

- 5 kinds of applications
  - Memory intensive
  - 6 real applications and 1 benchmark
- HMTT: hybrid memory trace toolkit
  - Collecting memory accessing traces on the Intel Xeon E5-2620 CPU

Workload	Domain	Input Data	Trace Size (GiB)
BFS [23]	Graph Processing	32M vertices × 512M edges	13
PageRank (PR) [23]			31
VGG [24]	Machine Learning	Flickr1024 [25]	47
LeViT (LV) [26]		imagenet200 [27]	151
Memcached (MC) [28]	Database	memtier [29]	4
canu [30]	Bioinformatics	Nanopore [31]	87
CG, BT, SP UA, LU	NPB	Class D	59, 51 59, 81 57
MG, IS		Class C	50, 62



Yongbing Huang, Licheng Chen, Zehan Cui, Yuan Ruan, Yungang Bao, Mingyu Chen, and Ninghui Sun. 2014. HMTT: A hybrid hardware/software tracing system for bridging the DRAM access trace's semantic gap. ACM Trans. Archit. Code Optim. 11, 1, Article 7 (February 2014), 25 pages.

Porotype and parameters

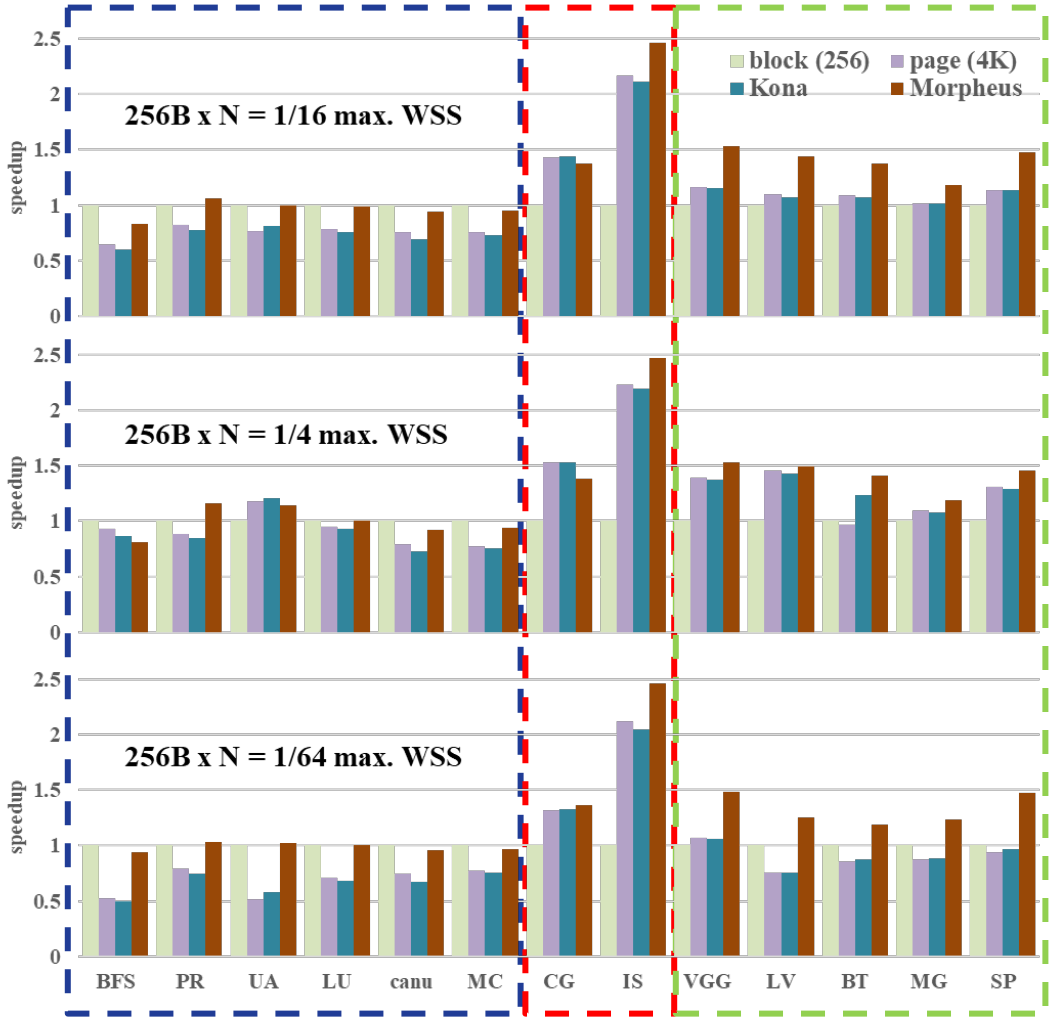
- 5 kinds of applications
  - Memory intensive
  - 6 real applications and 1 benchmark
- HMTT: hybrid memory trace toolkit
  - Collecting memory accessing traces on the Intel Xeon E5-2620 CPU
- DRAMsim 3: memory simulator
  - Replaying traces
- Baseline
  - Direct-mapped block- or page-based cache
  - Kona

Workload	Domain	Input Data	Trace Size (GiB)
BFS [23]	Graph Processing	32M vertices × 512M edges	13
PageRank (PR) [23]			31
VGG [24]	Machine Learning	Flickr1024 [25]	47
LeViT (LV) [26]		imagenet200 [27]	151
Memcached (MC) [28]	Database	memtier [29]	4
canu [30]	Bioinformatics	Nanopore [31]	87
CG, BT, SP UA, LU	NPB	Class D	59, 51 59, 81 57
MG, IS		Class C	50, 62

Trace Scanner	time window	256 ns
	max. outstanding	64
DRAM	DDR4,8Gbx4,3200MHz	
	64-entry read queue, 64-entry write queue	
	CL-tRCD-tRP-tRAS: 22-22-22-52	
	N_{bits}	512
Network	propagation delay	800 ns
	Bandwidth	100 Gbps
Coherent Interconnect	propagation delay	100 ns
	Bandwidth	500 Gbps
Morpheus	Block Region	HPT: N entries, 4-way. TLB: 1 MB
	Page Region	HPT: $N/16$ entries, 4-way. TLB: 64 KB
	Reversed Page Table	N entries
	Thresholds History Stack	16 entries
	monitoring window	10 ms
	Thresholds initial value	PROMOTION_T = 0 PADDING_T = 2000

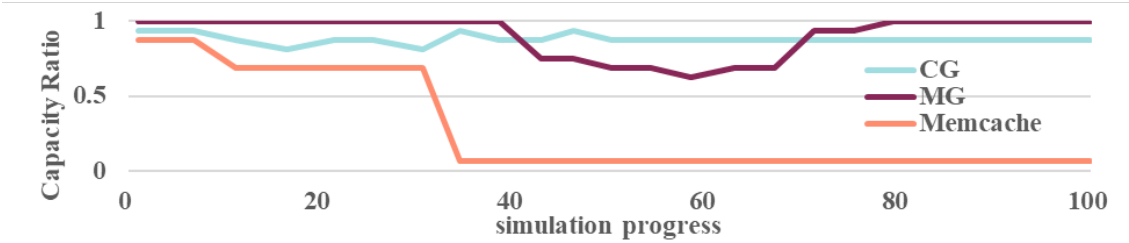
Results: single application

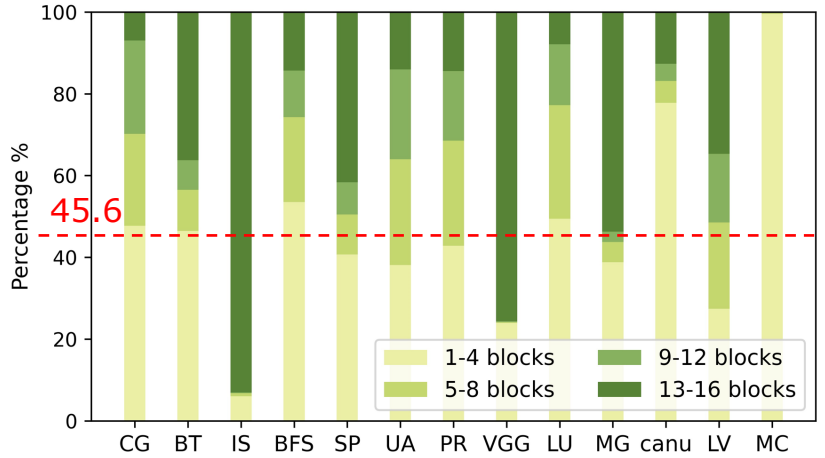
- Varying DRAM cache capacity
  - 1/4, 1/16, 1/64 of the application's footprint
- Performance in each category
  - Temporary locality only
    - Similar with the block (256)
  - Spatial locality only
    - Similar with the page (4K)
  - Both locality
    - Geometric mean of 1.28x speedup compared to Kona



Results

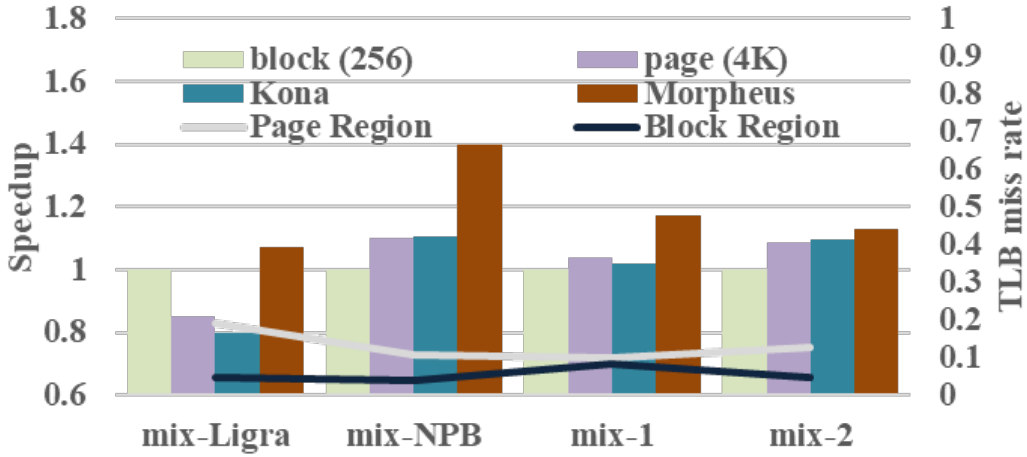
- Different region growth rate for different categories
  - Adjustable capacity affects the number of fetched data
- Decreasing the percentage of inefficient pages
  - From 82.6% to 45.6%
  - Saving network bandwidth by 24.5% to 94.7% compared to Kona





Performance on multi-apps.

- 2 mixing methods
  - one spatial-locality-only with one temporal-locality-only
  - randomly choosing applications from Ligra and NPB
- Morpheus presents a geometric mean of 1.19x speedup compared to the Kona



Takeaway

- Coherence-based DRAM cache card
  - A typical use case of CXL-enabled enhanced memory functions
  - Flexibility to deploy complex hardware logic
- Morpheus presents an adaptive DRAM cache with online granularity adjustment
- Morpheus presents a 1.17x to 1.34x speedup compared to the Kona and remarkably saves network bandwidth by 24.5% to 94.7%.
  - Please refer to the paper for more results

Morpheus: An Adaptive DRAM Cache with Online Granularity Adjustment for Disaggregated Memory

Xu Zhang, Tianyue Lu, Yisong Chang, Ke Zhang, Mingyu Chen

Thanks for Listening!

Please contact us for more implementation details