SOSP 2023, Koblenz, Germany, October 24, 2023

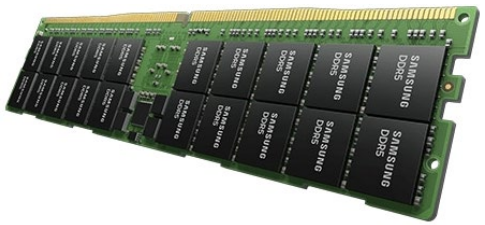
MEMTIS: Efficient Memory Tiering with Dynamic Page Classification and Page Size Determination

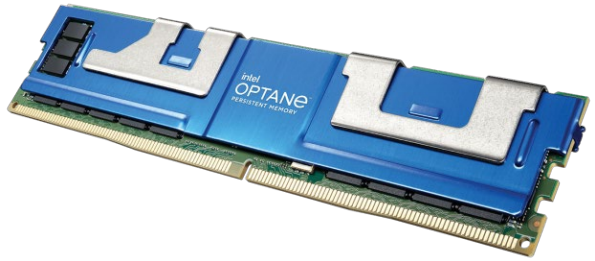
Taehyung Lee, *Sumit Kumar Monga,*
Changwoo Min, and Young Ik Eom

igalia

Tiered main memory in OS

Physical memory space





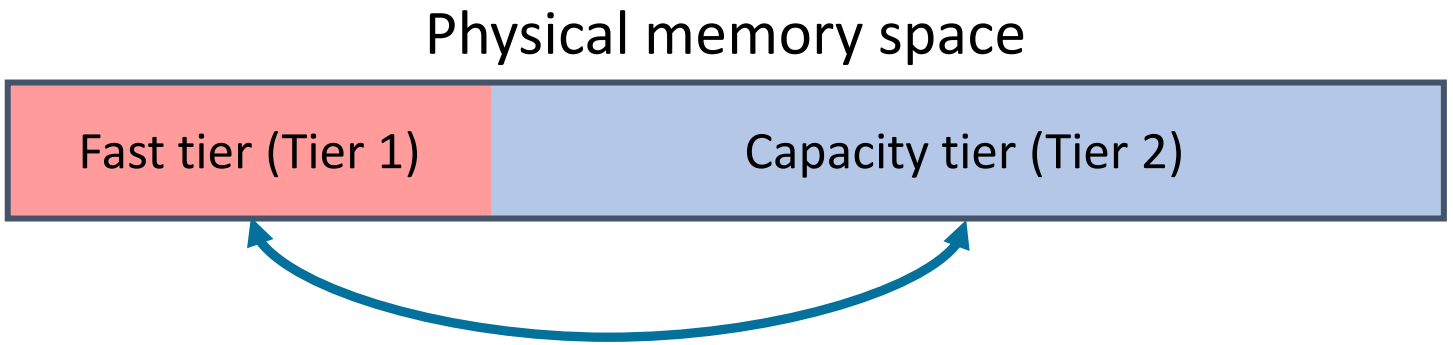
DRAM

- High $/GB
- High load/store latency
- Low capacity

CXL memory expander / Intel Optane DC PMM

- Low $/GB
- Low load/store latency
- High capacity

Tiered main memory in OS



Monitor memory accesses

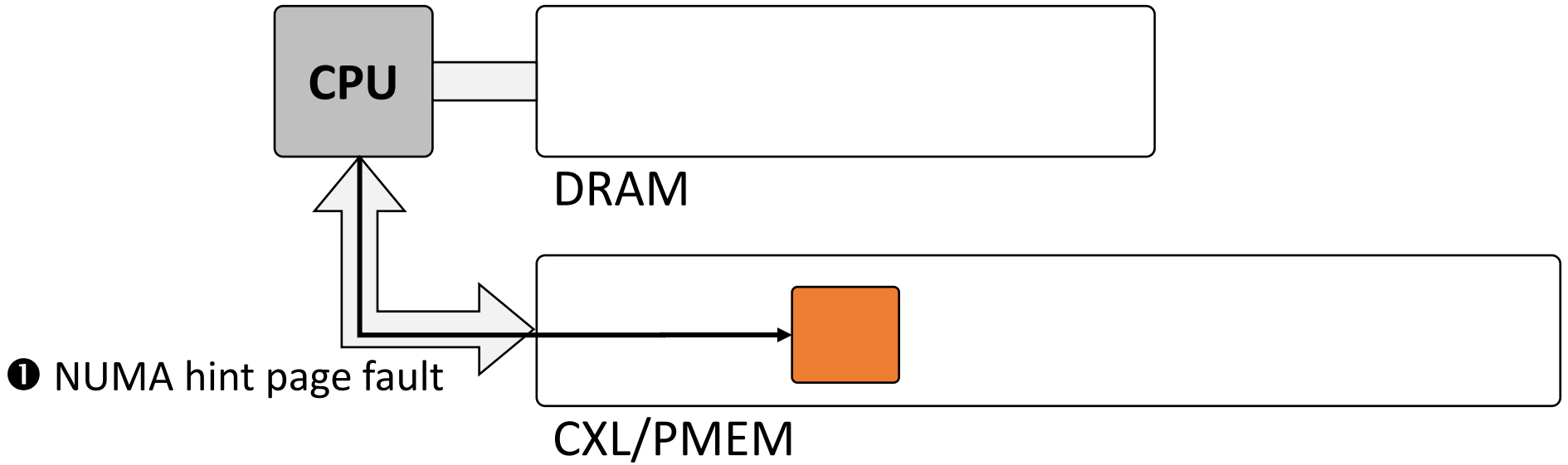
Decide which pages are hot

Migrate hot pages to the fast tier

Goal: maximize the utilization of *fast tier* memory with *hot* pages

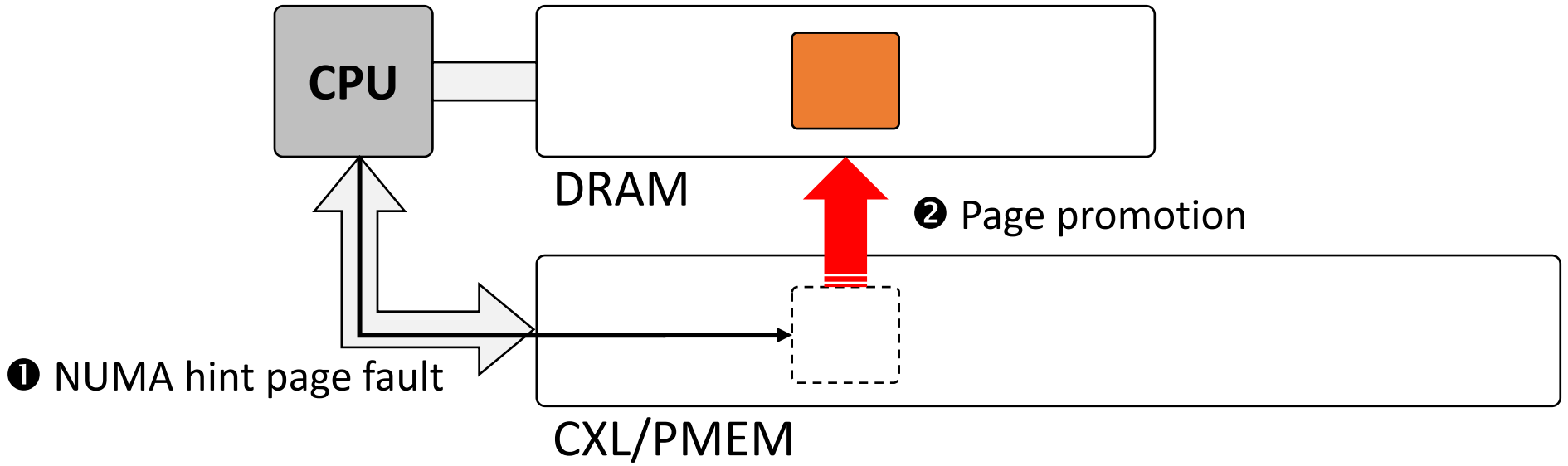
Which pages are hot?

- Static access counts as a threshold for *hot* pages
  - ✓ AutoNUMA:



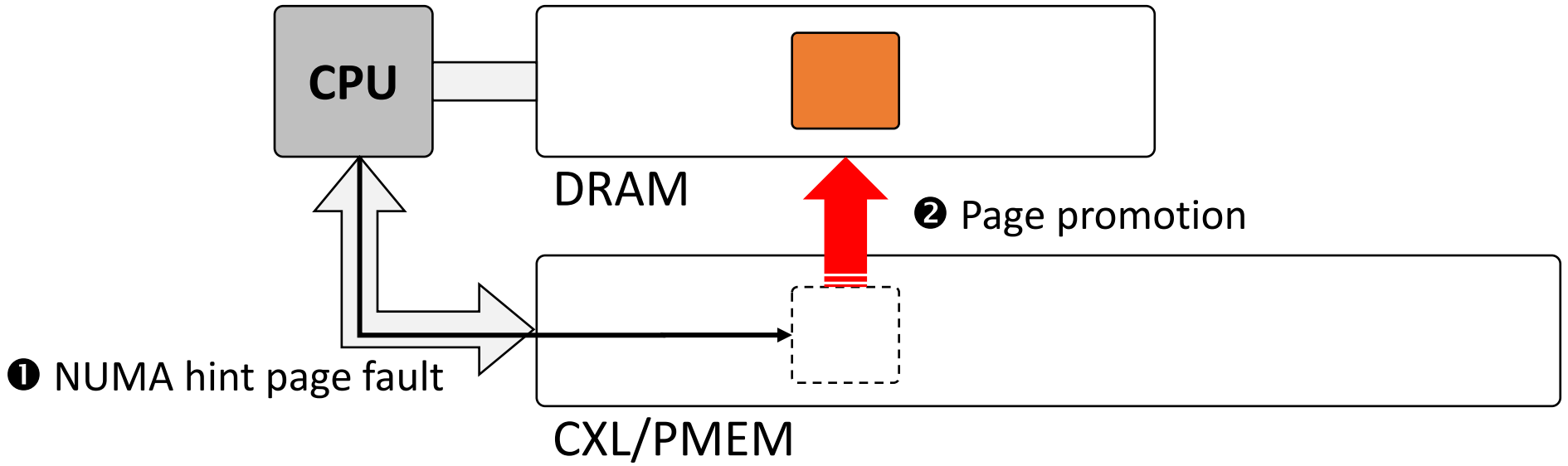
Which pages are hot?

- Static access counts as a threshold for *hot* pages
  - ✓ AutoNUMA:



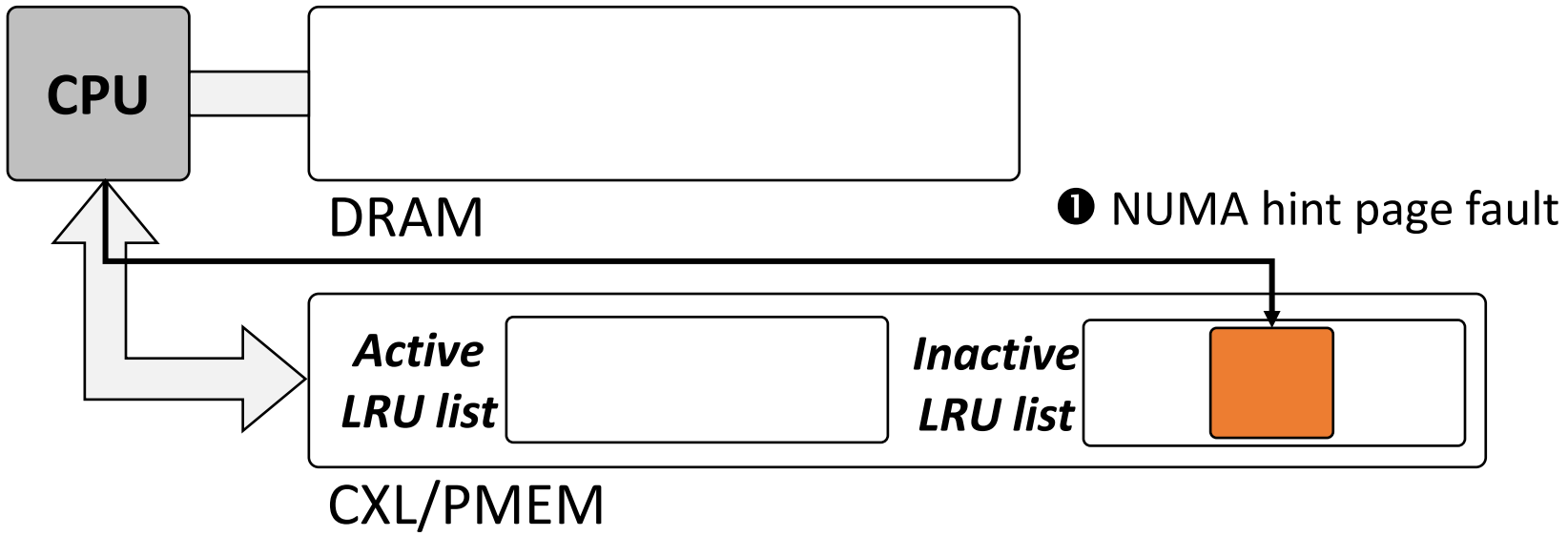
Which pages are hot?

- Static access counts as a threshold for *hot* pages
  - ✓ AutoNUMA: 1 access (considering only access recency)



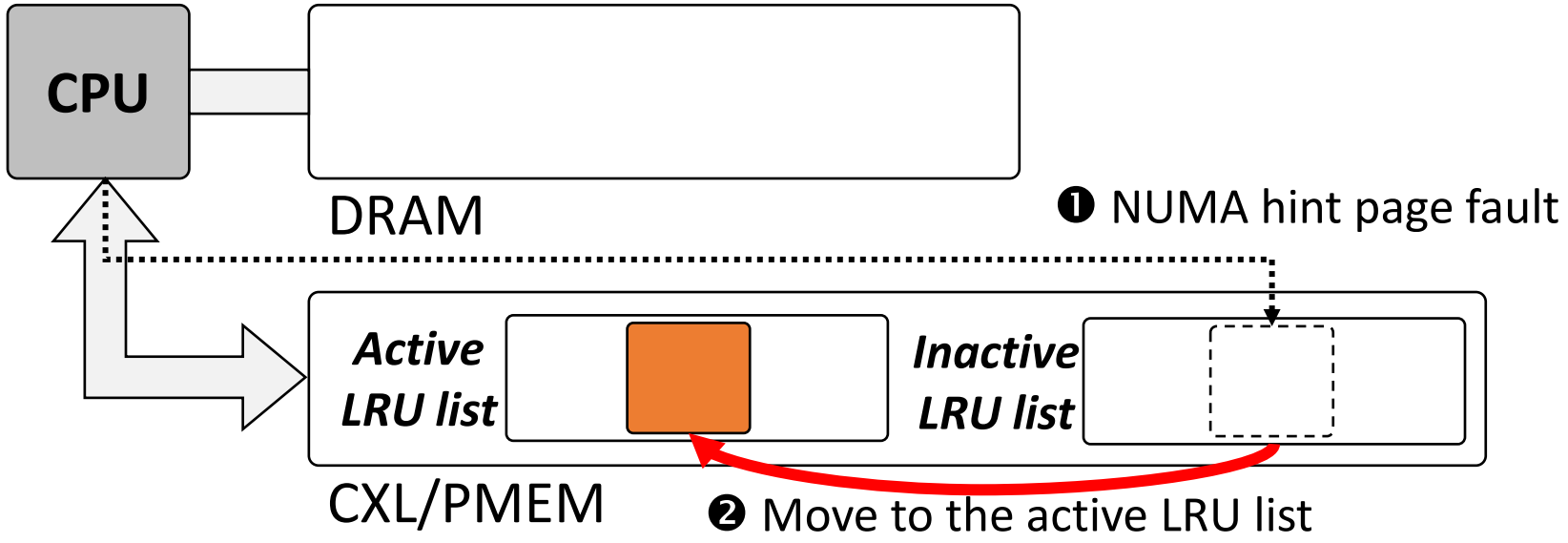
Which pages are hot?

- Static access counts as a threshold for *hot* pages
  - ✓ AutoNUMA: 1 access (considering only access recency)
  - ✓ TPP [ASPLOS 2023]:



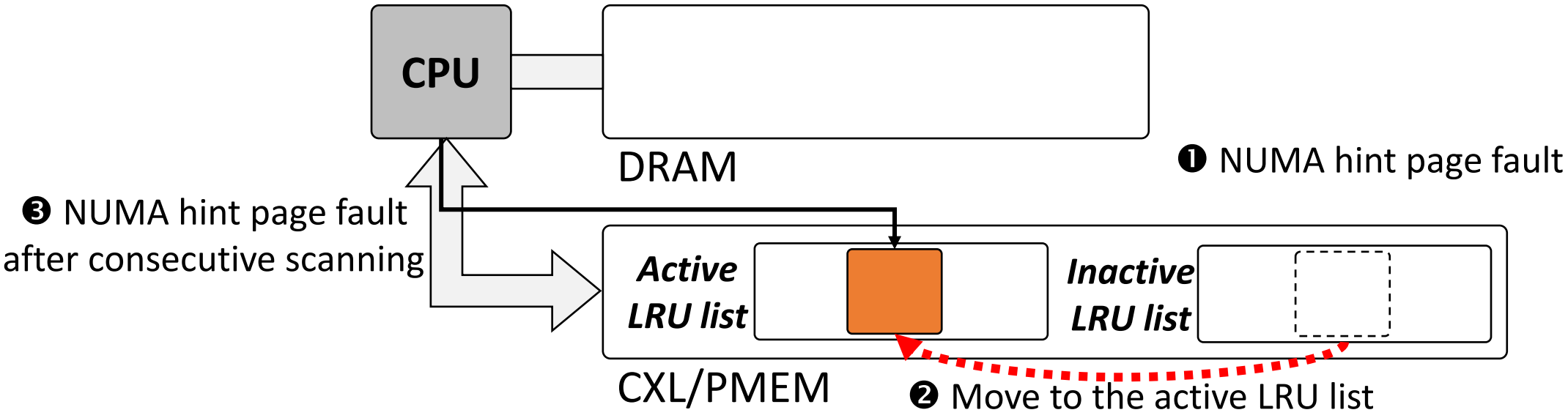
Which pages are hot?

- Static access counts as a threshold for *hot* pages
  - ✓ AutoNUMA: 1 access (considering only access recency)
  - ✓ TPP [ASPLOS 2023]:



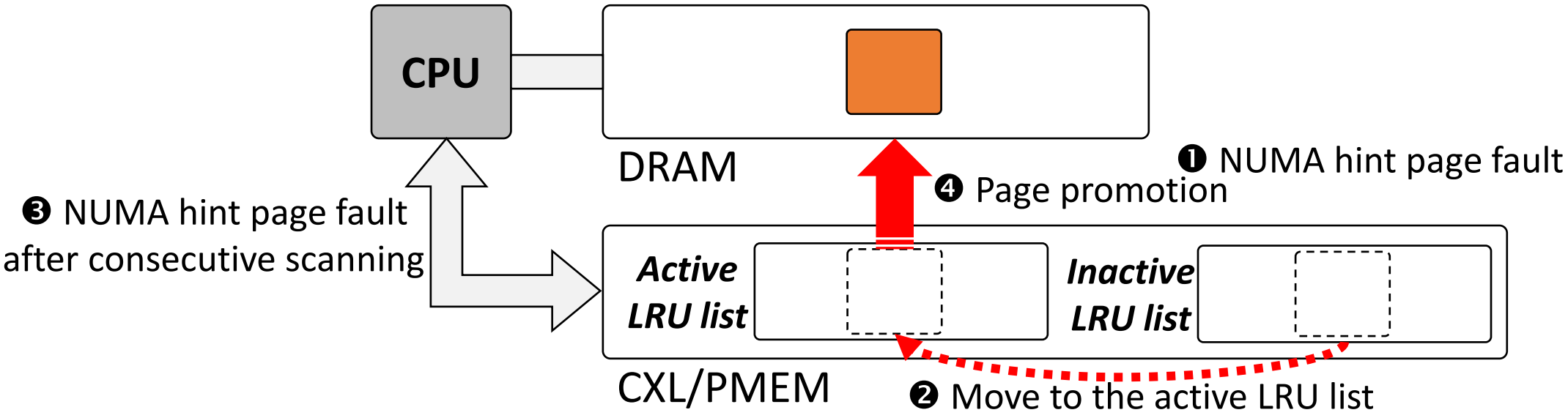
Which pages are hot?

- Static access counts as a threshold for *hot* pages
  - ✓ AutoNUMA: 1 access (considering only access recency)
  - ✓ TPP [ASPLOS 2023]:



Which pages are hot?

- Static access counts as a threshold for *hot* pages
  - ✓ AutoNUMA: 1 access (considering only access recency)
  - ✓ TPP [ASPLOS 2023]: 2 accesses



Which pages are hot?

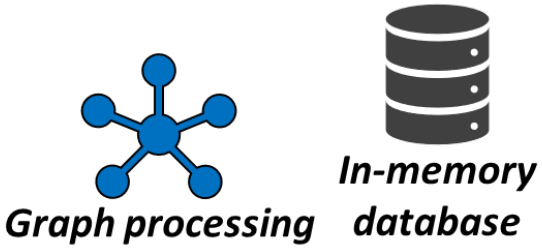
- Static access counts as a threshold for *hot* pages
  - ✓ AutoNUMA: 1 access (considering only access recency)
  - ✓ TPP [ASPLOS 2023]: 2 accesses
  - ✓ HeMem [SOSP 2021]: hot threshold → 8 load accesses or 4 store accesses
    cooling threshold → 18 accesses
    (monitored through PEBS)

Which pages are hot?

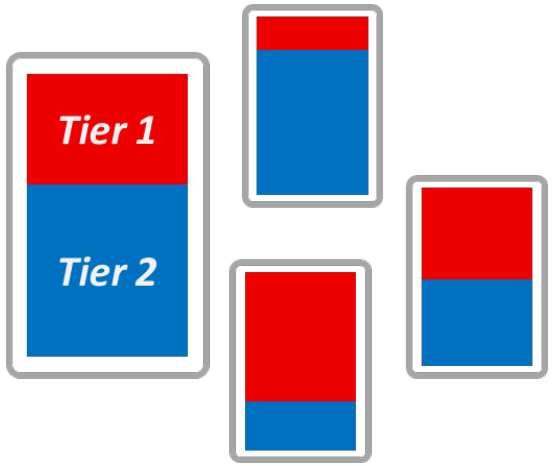
- Static access counts as a threshold for *hot* pages
  - ✓ AutoNUMA: 1 access (considering only access recency)
  - ✓ TPP [ASPLOS 2023]: 2 accesses
  - ✓ HeMem [SOSP 2021]: hot threshold → 8 load accesses or 4 store accesses
    cooling threshold → 18 accesses
    (monitored through PEBS)

Are such static approaches sufficient for *transparent* management of tiered memory?

Criticality of hotness detection

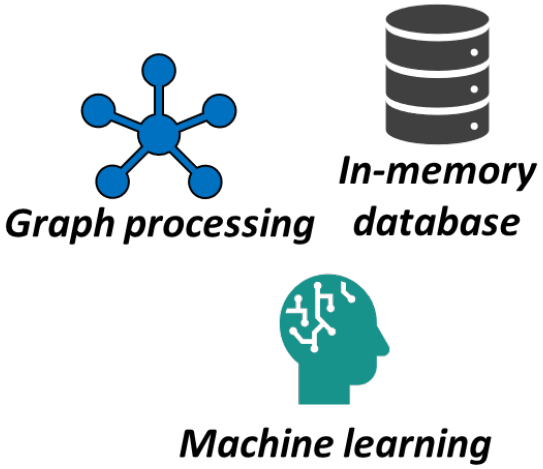


Diverse set of applications

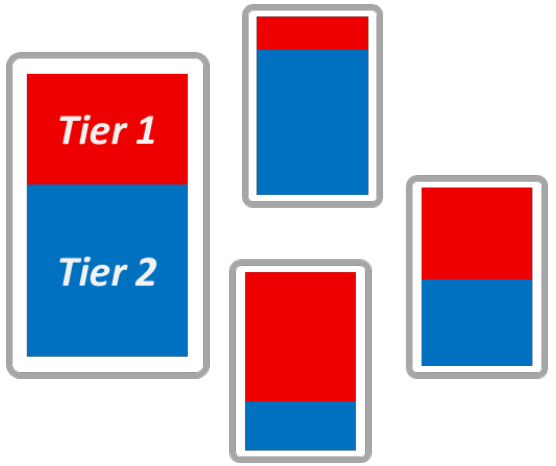


Various memory configurations

Criticality of hotness detection

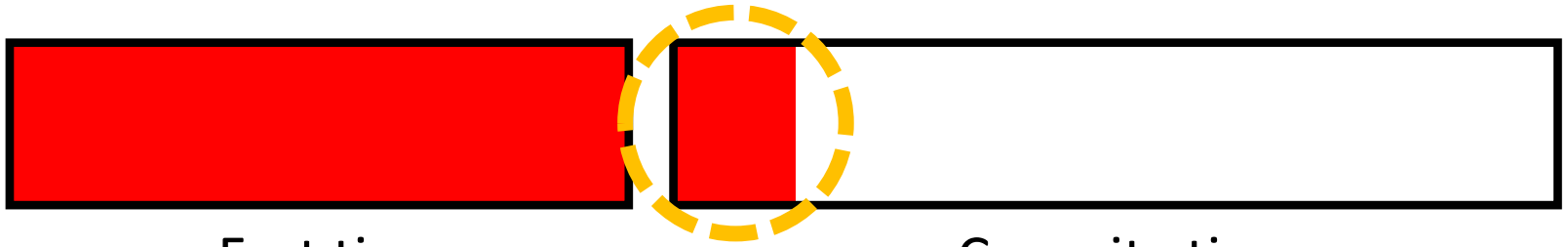


Diverse set of applications

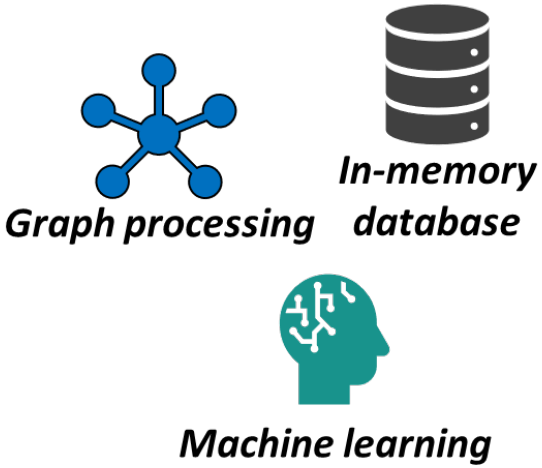


Various memory configurations

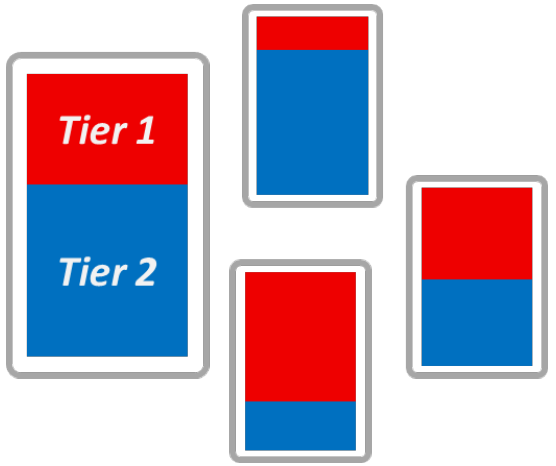
- Case 1: hot set size > fast tier size



Criticality of hotness detection

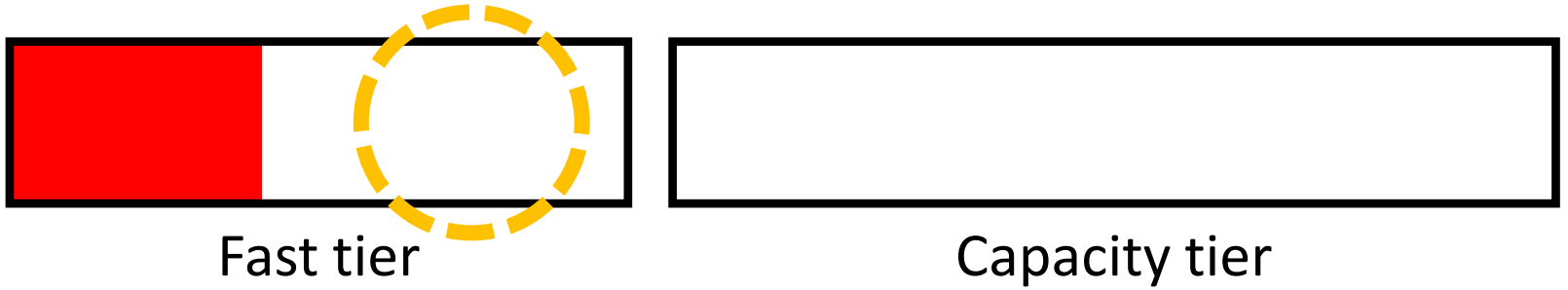


Diverse set of applications



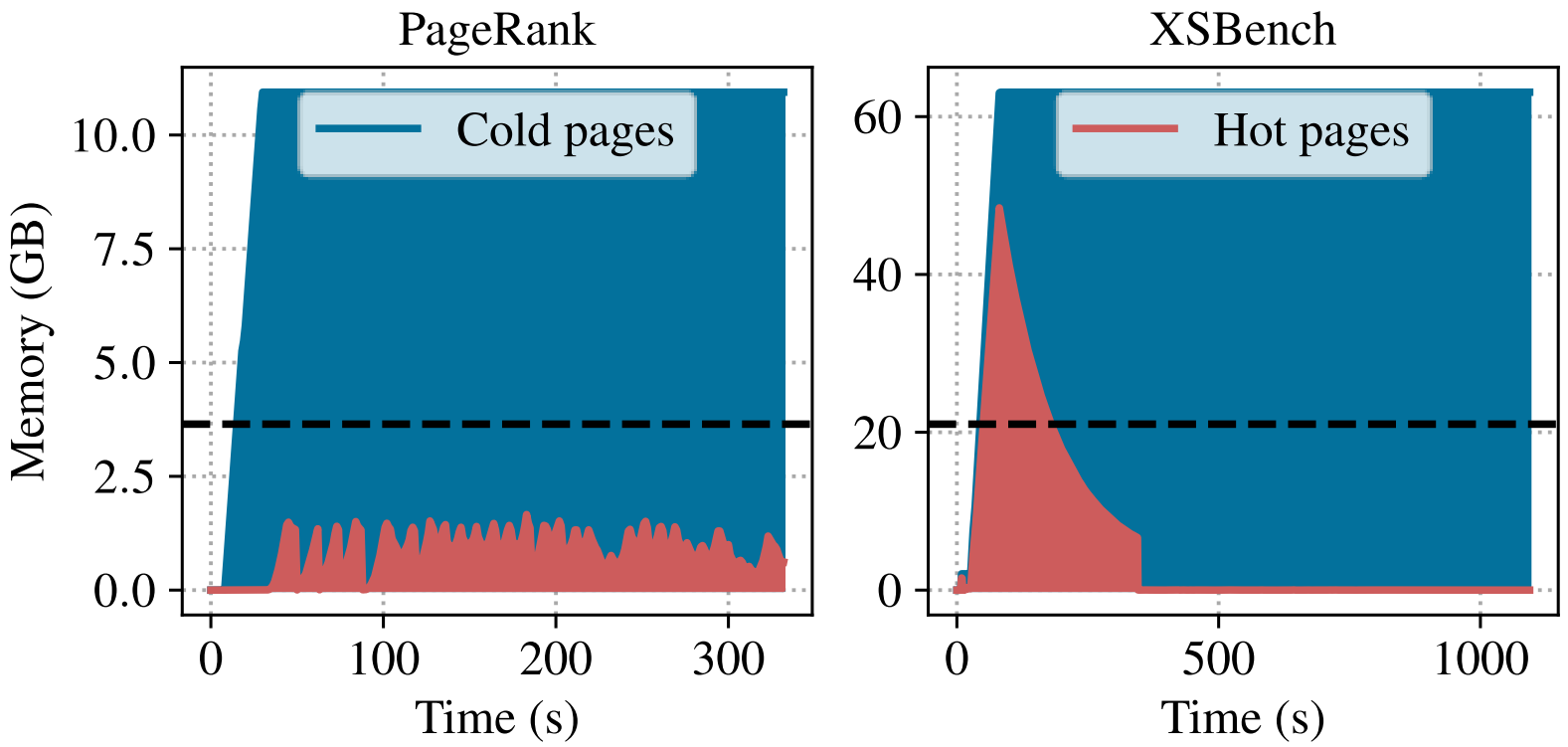
Various memory configurations

- Case 1: hot set size > fast tier size
- Case 2: hot set size < fast tier size



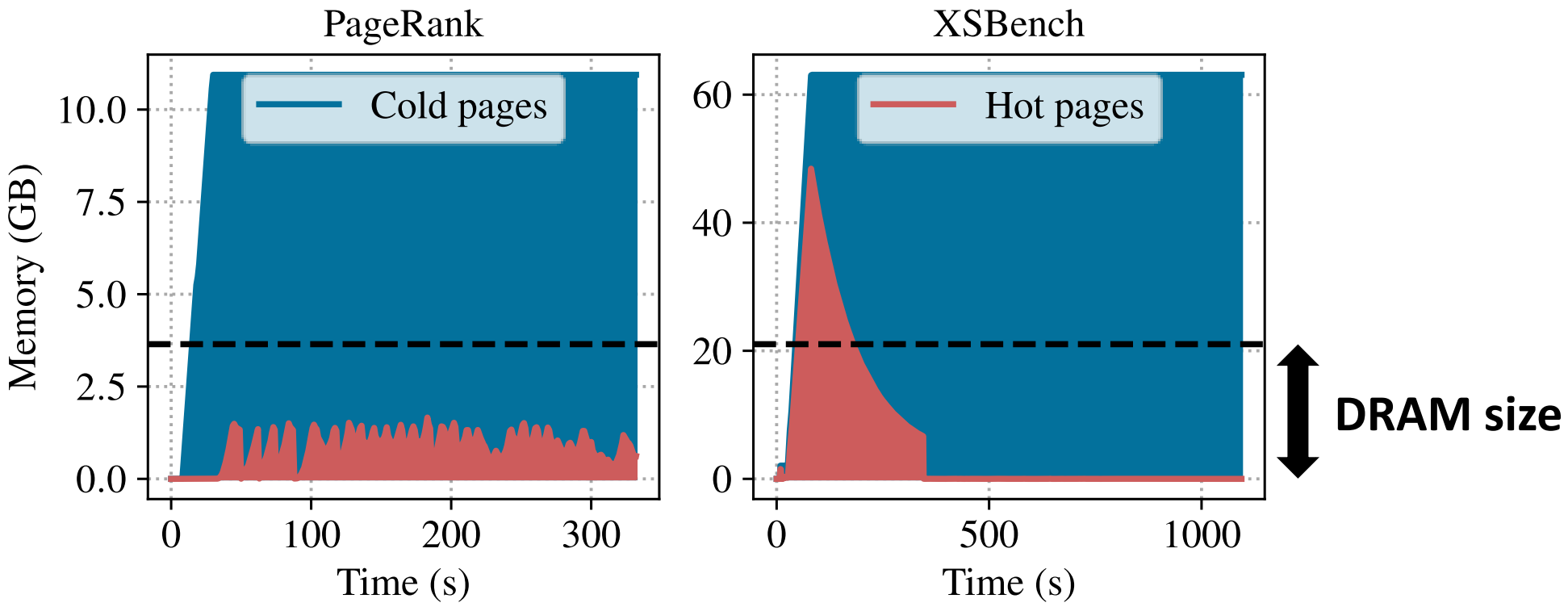
Criticality of hotness detection

- HeMem: Hot/cold memory footprint
- DRAM + NVM tiered memory system



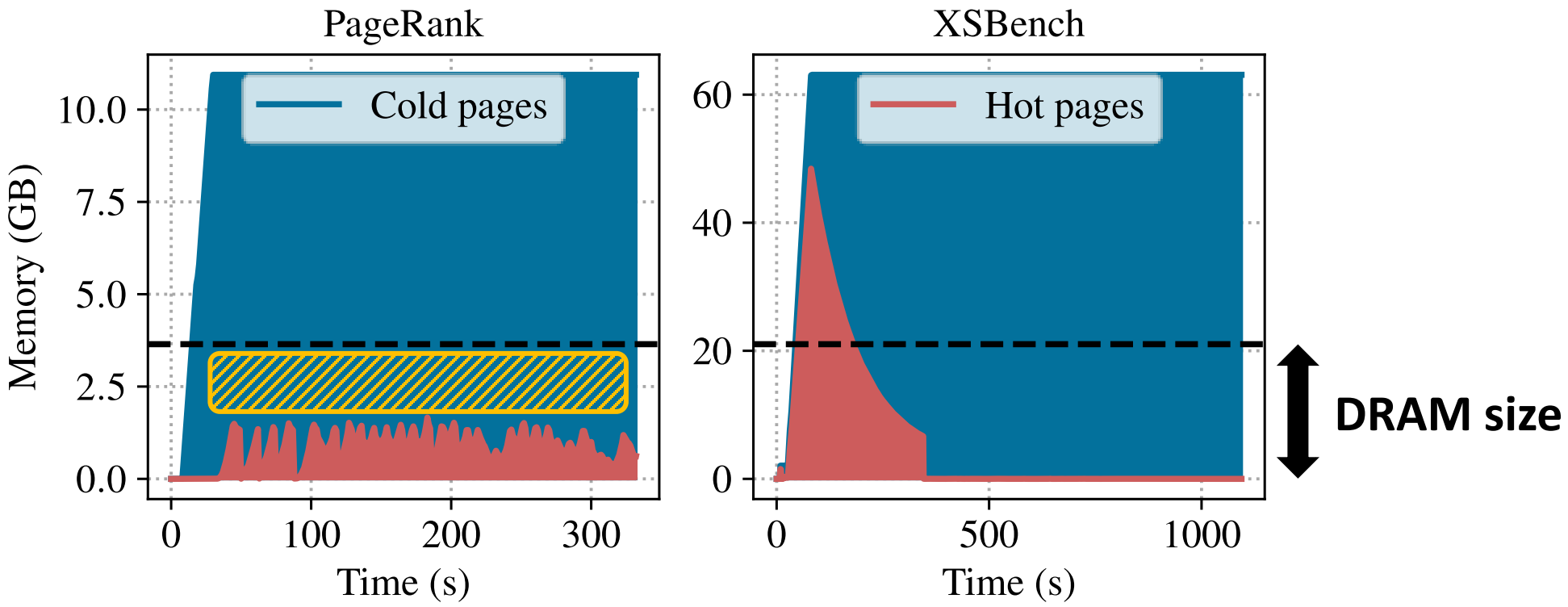
Criticality of hotness detection

- HeMem: Hot/cold memory footprint
- DRAM + NVM tiered memory system



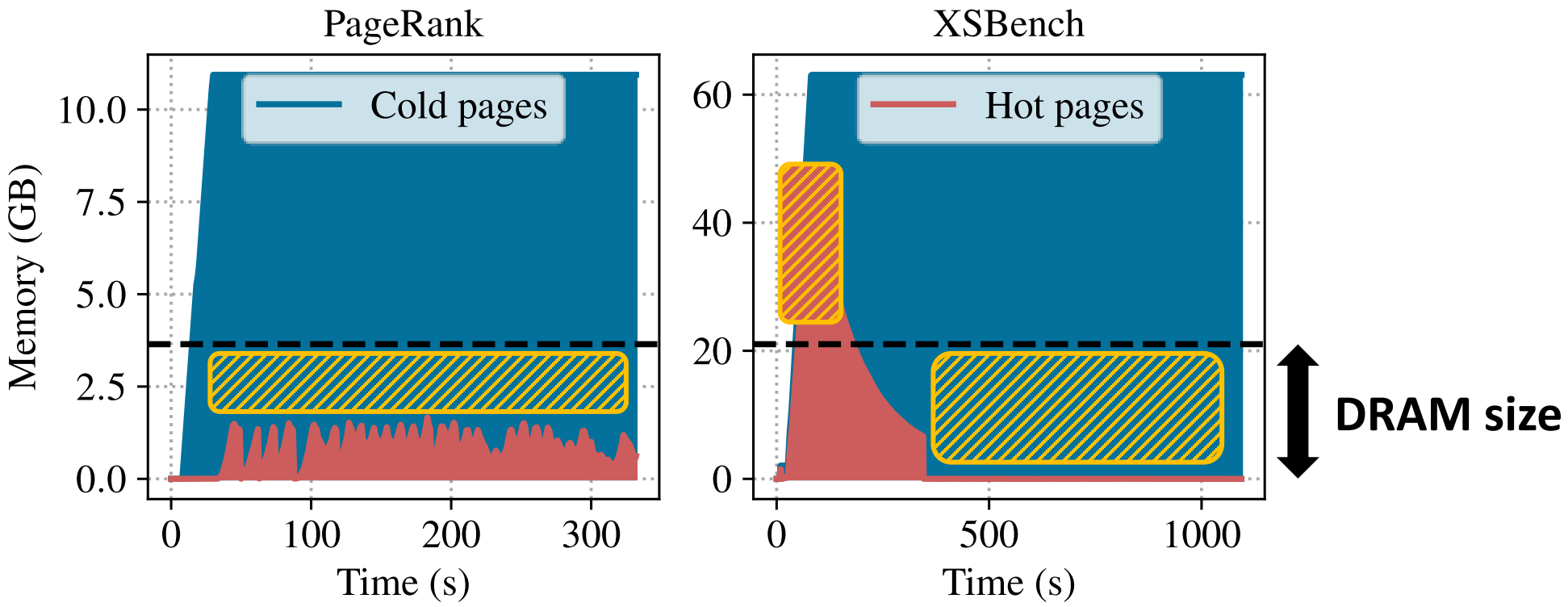
Criticality of hotness detection

- HeMem: Hot/cold memory footprint
- DRAM + NVM tiered memory system



Criticality of hotness detection

- HeMem: Hot/cold memory footprint
- DRAM + NVM tiered memory system

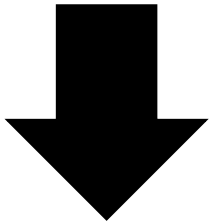


Our solution: MEMTIS

- Goals
  - ✓ Maximize the fast tier utilization with ***truly hot*** pages
  - ✓ Work well for diverse set of applications and memory configurations

Our solution: MEMTIS

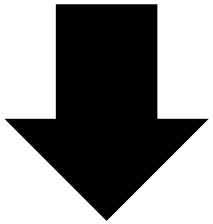
- Goals
  - ✓ Maximize the fast tier utilization with ***truly hot*** pages
  - ✓ Work well for diverse set of applications and memory configurations



Fine-grained, lightweight access tracking

Our solution: MEMTIS

- Goals
  - ✓ Maximize the fast tier utilization with ***truly hot*** pages
  - ✓ Work well for diverse set of applications and memory configurations

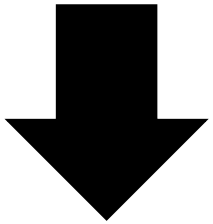


Fine-grained, lightweight access tracking

Histogram-based hot set classification

Our solution: MEMTIS

- Goals
  - ✓ Maximize the fast tier utilization with ***truly hot*** pages
  - ✓ Work well for diverse set of applications and memory configurations



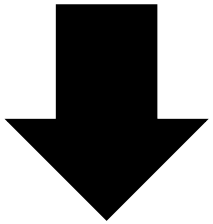
Fine-grained, lightweight access tracking

Histogram-based hot set classification

Skewness-aware page size determination

Our solution: MEMTIS

- Goals
  - ✓ Maximize the fast tier utilization with ***truly hot*** pages
  - ✓ Work well for diverse set of applications and memory configurations



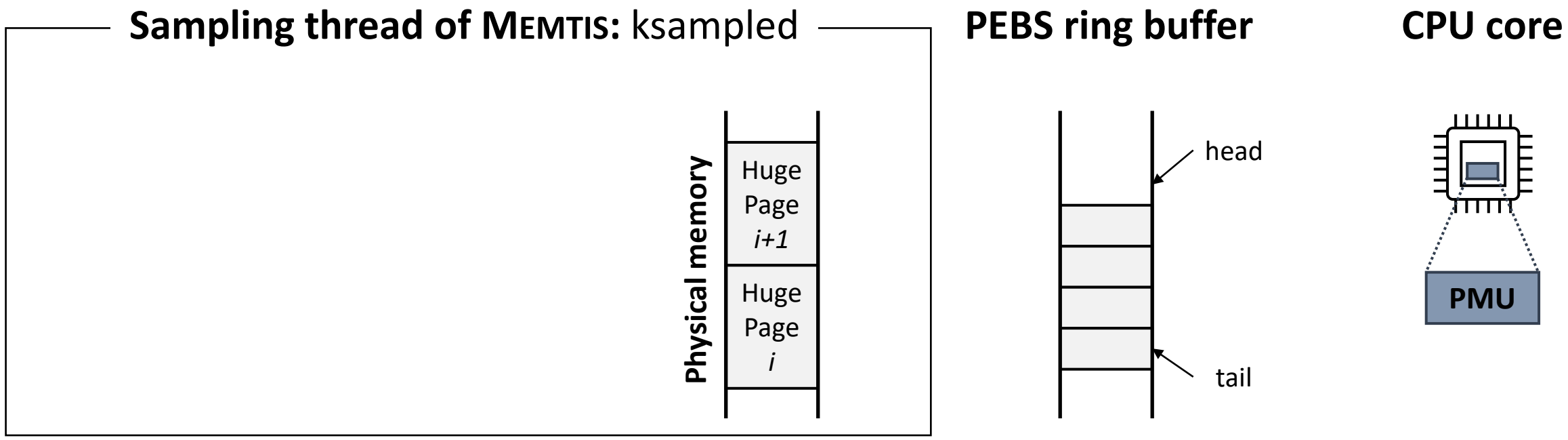
Fine-grained, lightweight access tracking

Histogram-based hot set classification

Skewness-aware page size determination

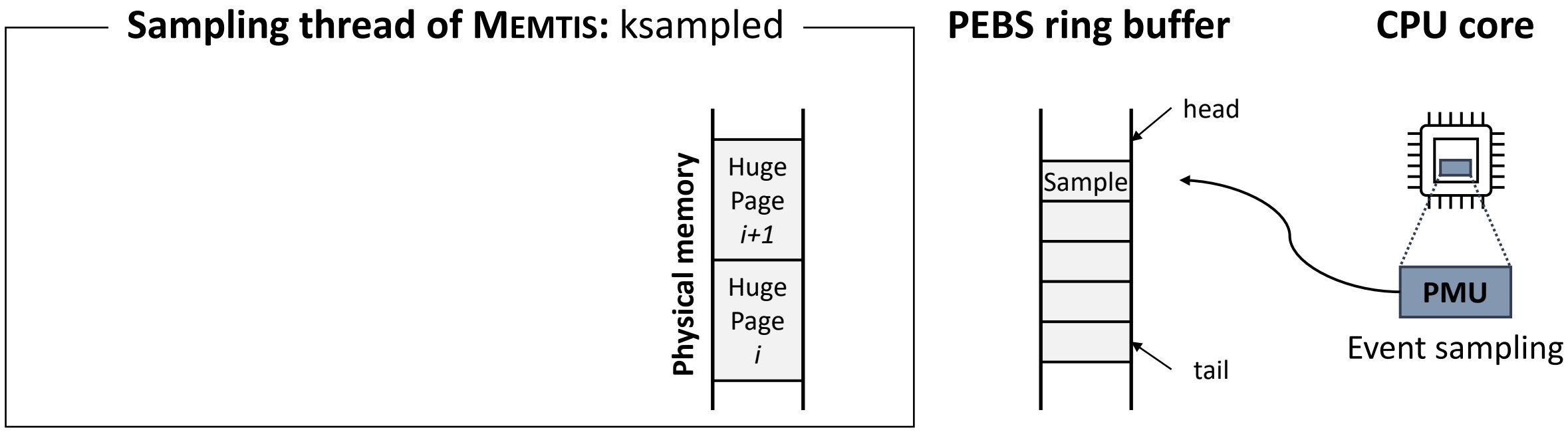
Fine-grained, Lightweight memory access sampling

- Using processor event-based sampling (PEBS): LLC load miss and store inst.



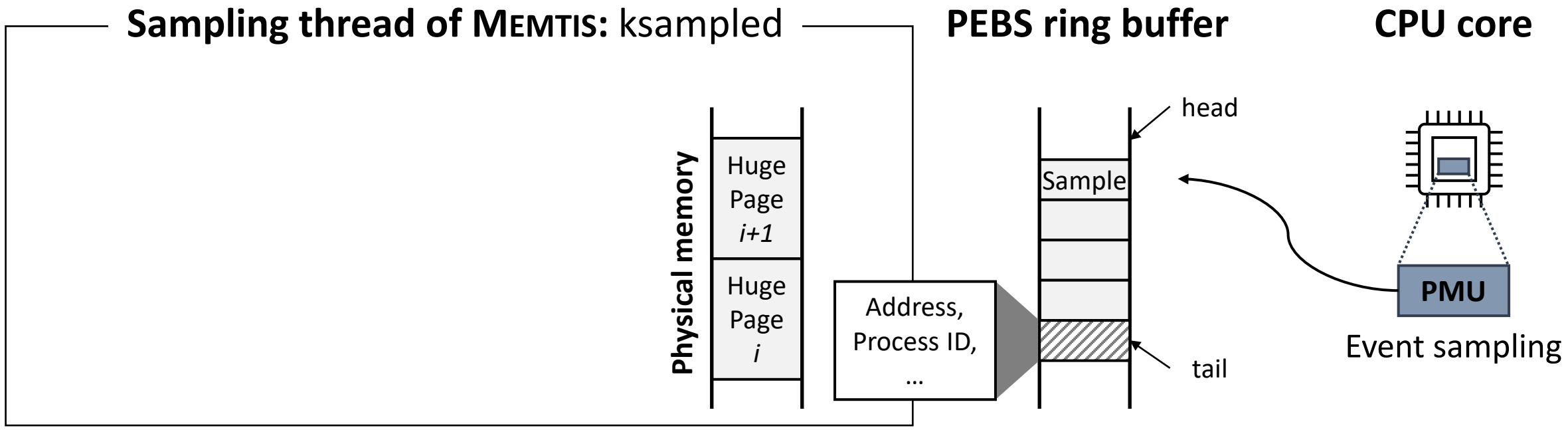
Fine-grained, Lightweight memory access sampling

- Using processor event-based sampling (PEBS): LLC load miss and store inst.



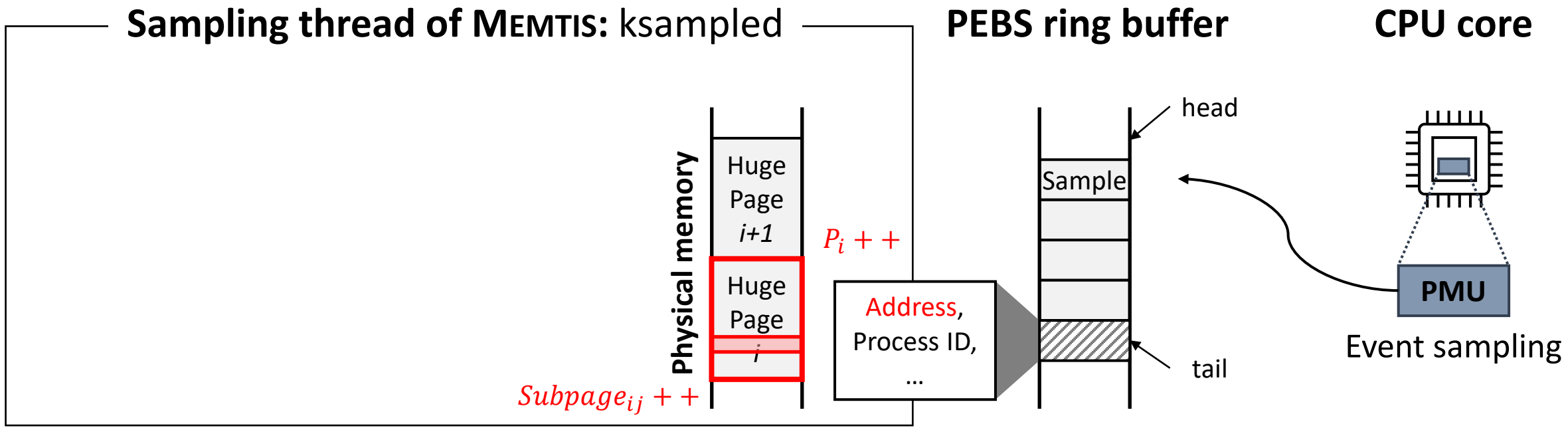
Fine-grained, Lightweight memory access sampling

- Using processor event-based sampling (PEBS): LLC load miss and store inst.



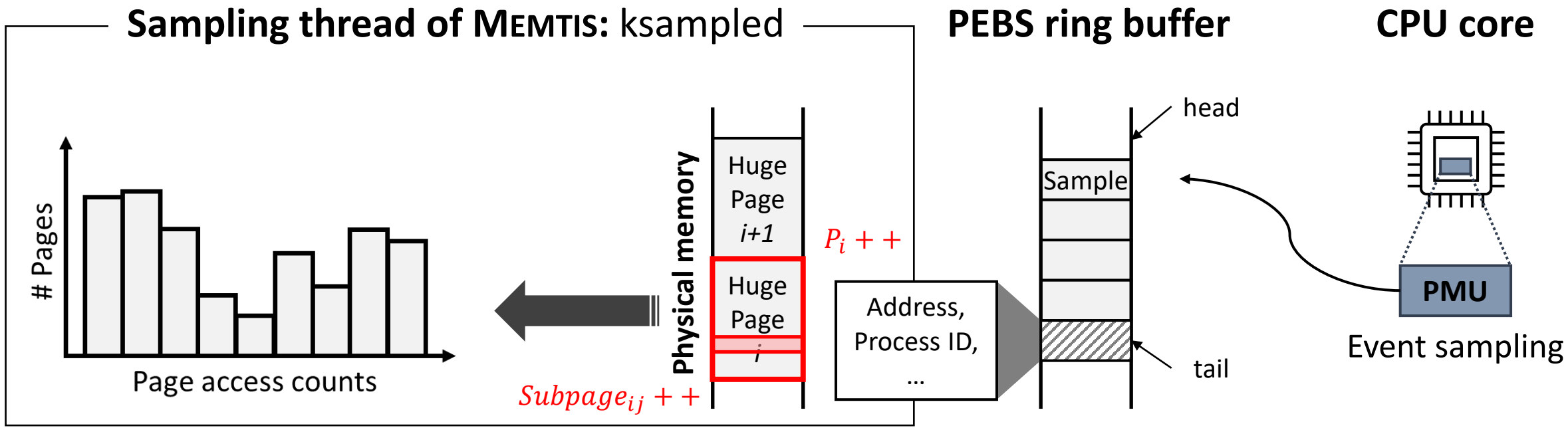
Fine-grained, Lightweight memory access sampling

- Using processor event-based sampling (PEBS): LLC load miss and store inst.
- Fine-grained access tracking for huge pages



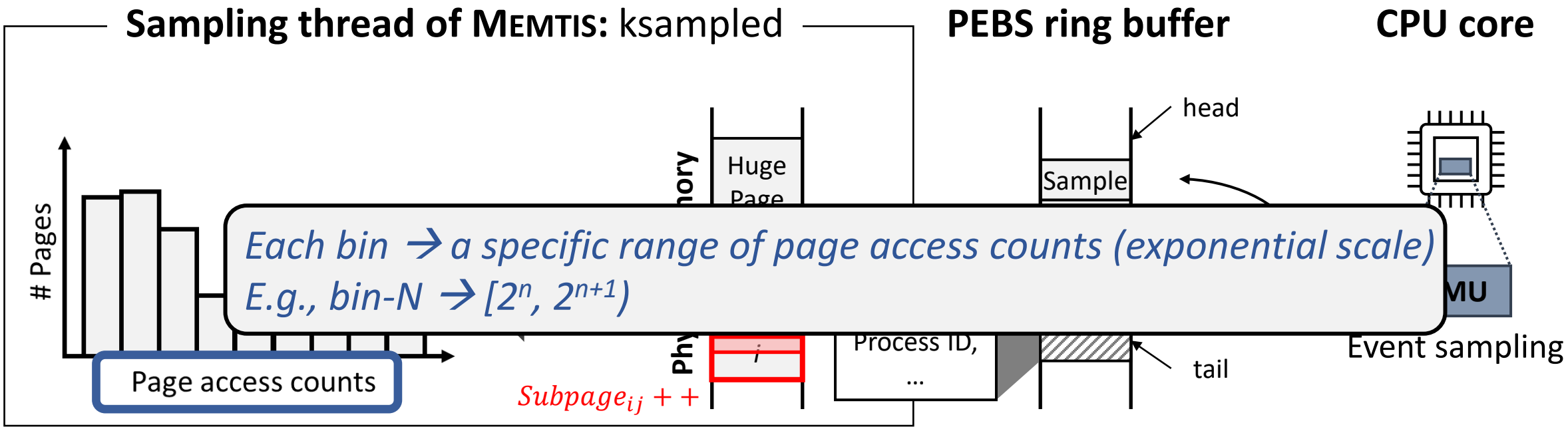
Fine-grained, Lightweight memory access sampling

- Using processor event-based sampling (PEBS): LLC load miss and store inst.
- Fine-grained access tracking for huge pages
- Building page access histogram



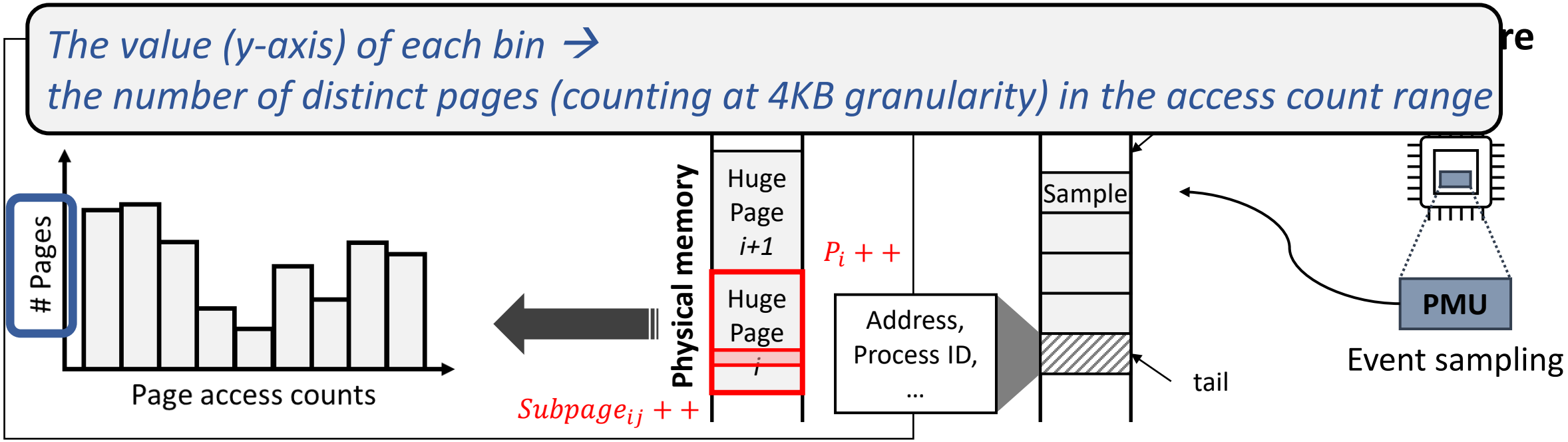
Fine-grained, Lightweight memory access sampling

- Using processor event-based sampling (PEBS): LLC load miss and store inst.
- Fine-grained access tracking for huge pages
- Building page access histogram



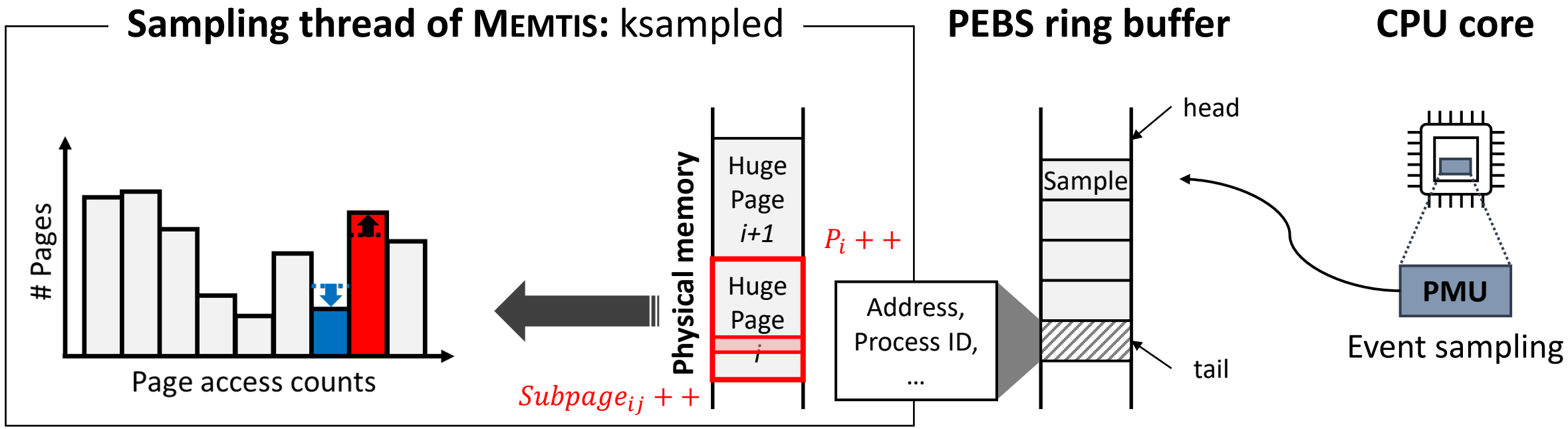
Fine-grained, Lightweight memory access sampling

- Using processor event-based sampling (PEBS): LLC load miss and store inst.
- Fine-grained access tracking for huge pages
- Building page access histogram



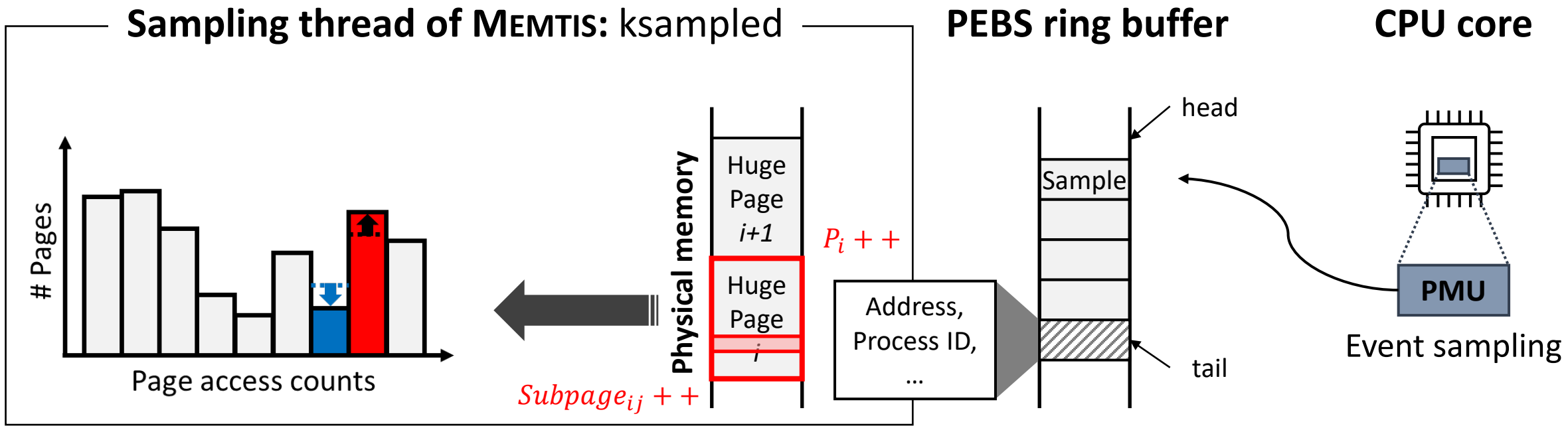
Fine-grained, Lightweight memory access sampling

- Using processor event-based sampling (PEBS): LLC load miss and store inst.
- Fine-grained access tracking for huge pages
- Building page access histogram



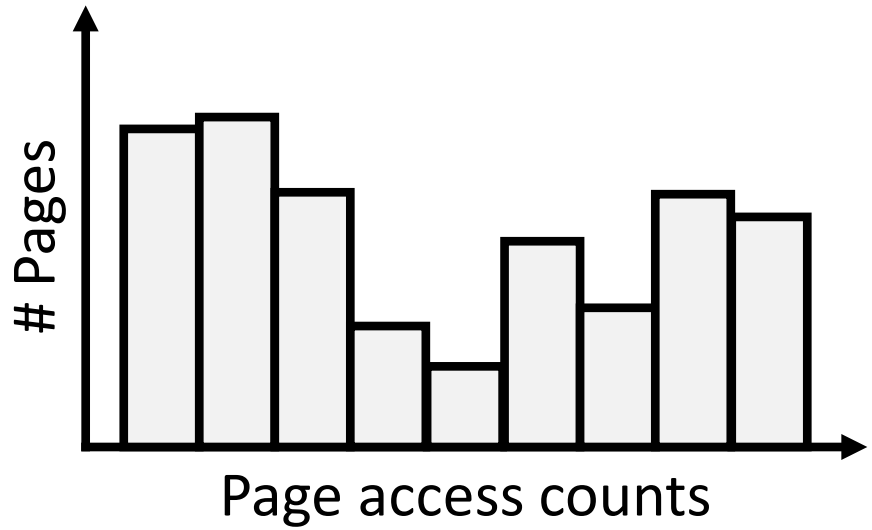
Fine-grained, Lightweight memory access sampling

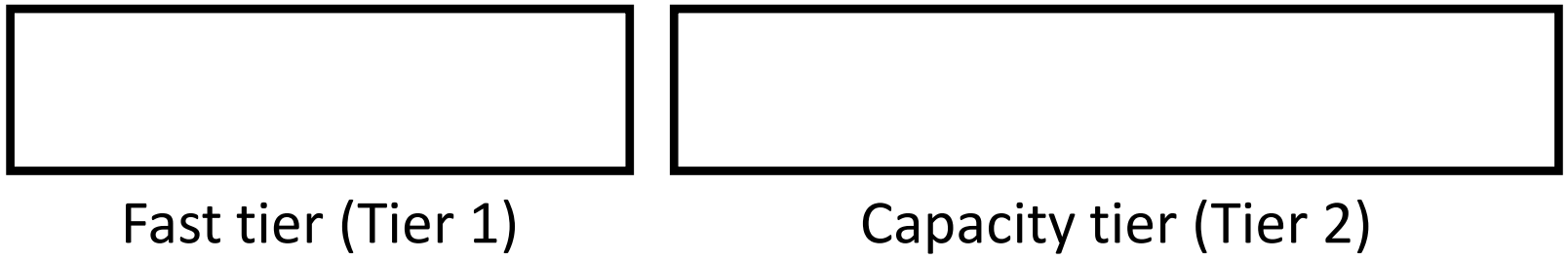
- Using processor event-based sampling (PEBS): LLC load miss and store inst.
- Fine-grained access tracking for huge pages
- Building page access histogram
- Dynamically adjusts the sampling period → keep the CPU usage < 3%



Histogram-based hot set classification

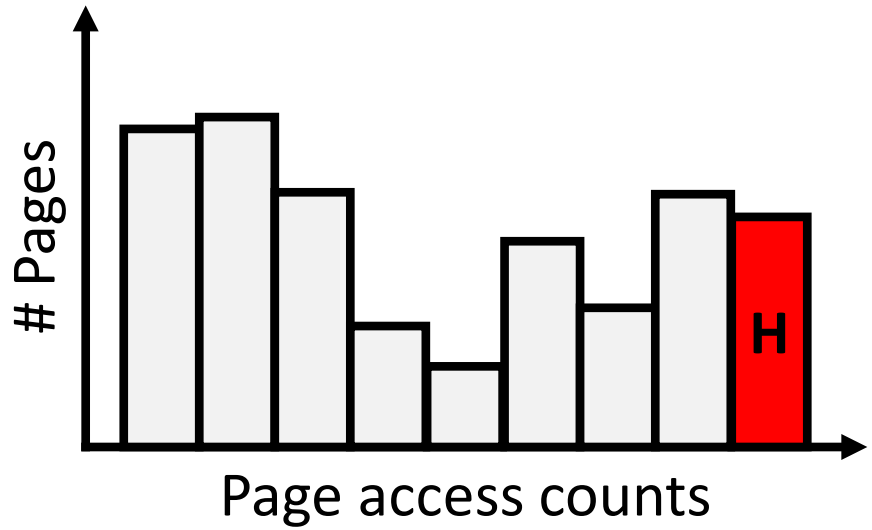
- Determining hot/warm/cold thresholds

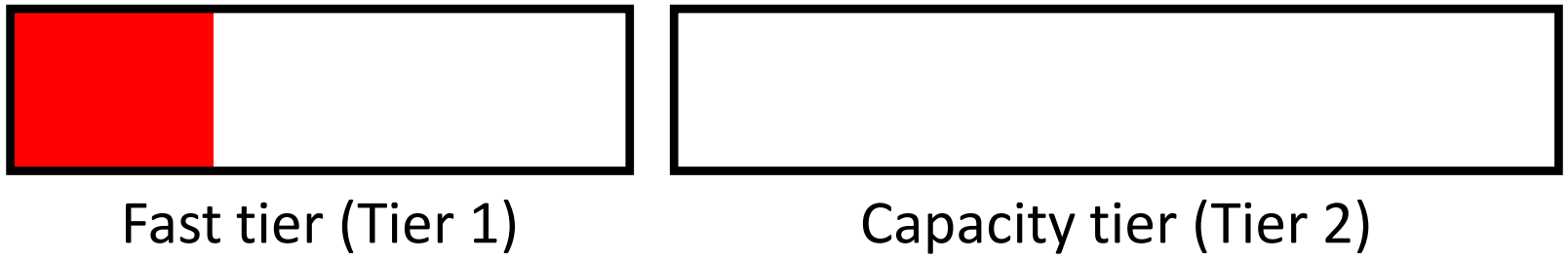




Histogram-based hot set classification

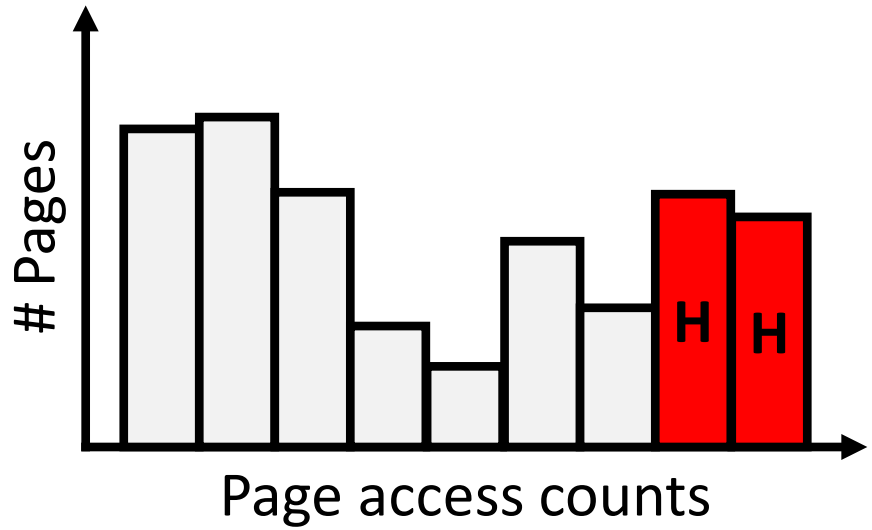
- Determining hot/warm/cold thresholds

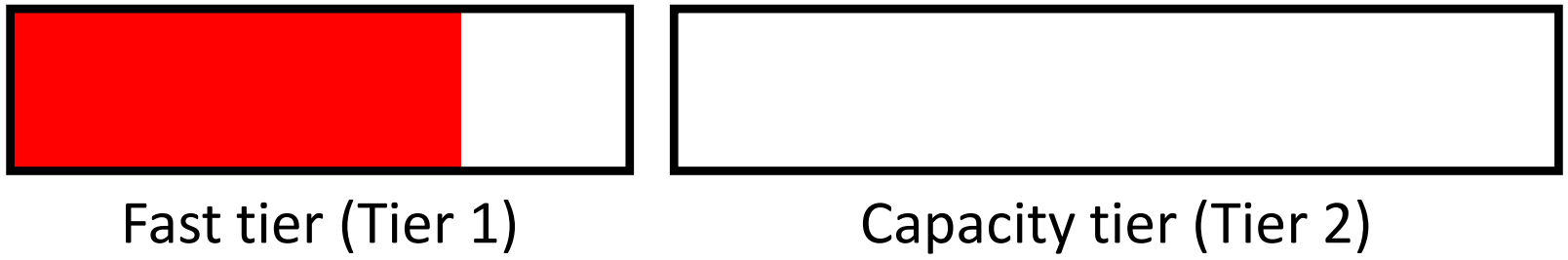




Histogram-based hot set classification

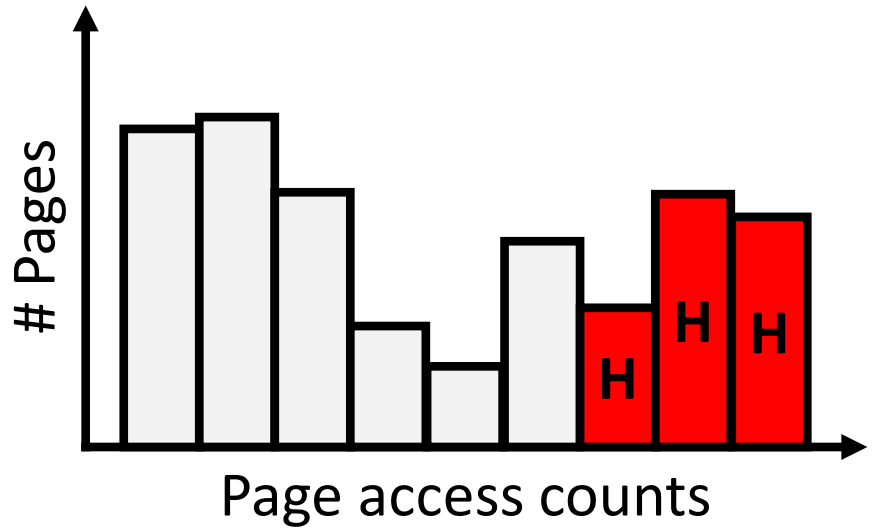
- Determining hot/warm/cold thresholds

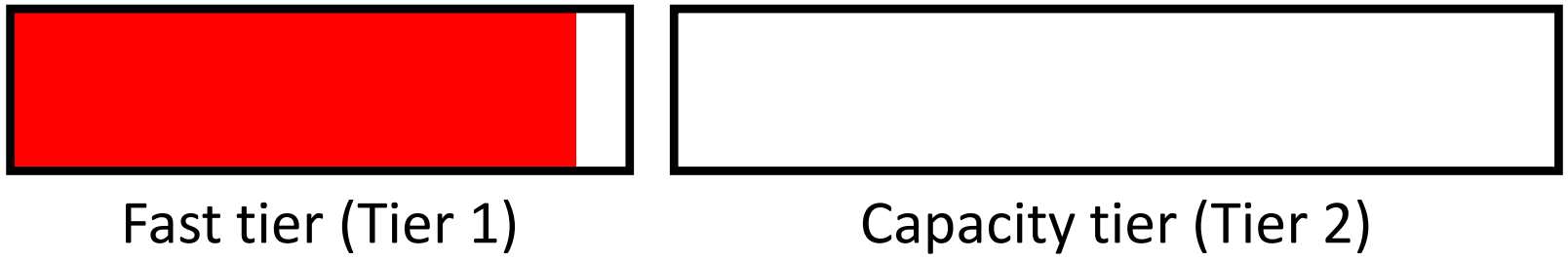




Histogram-based hot set classification

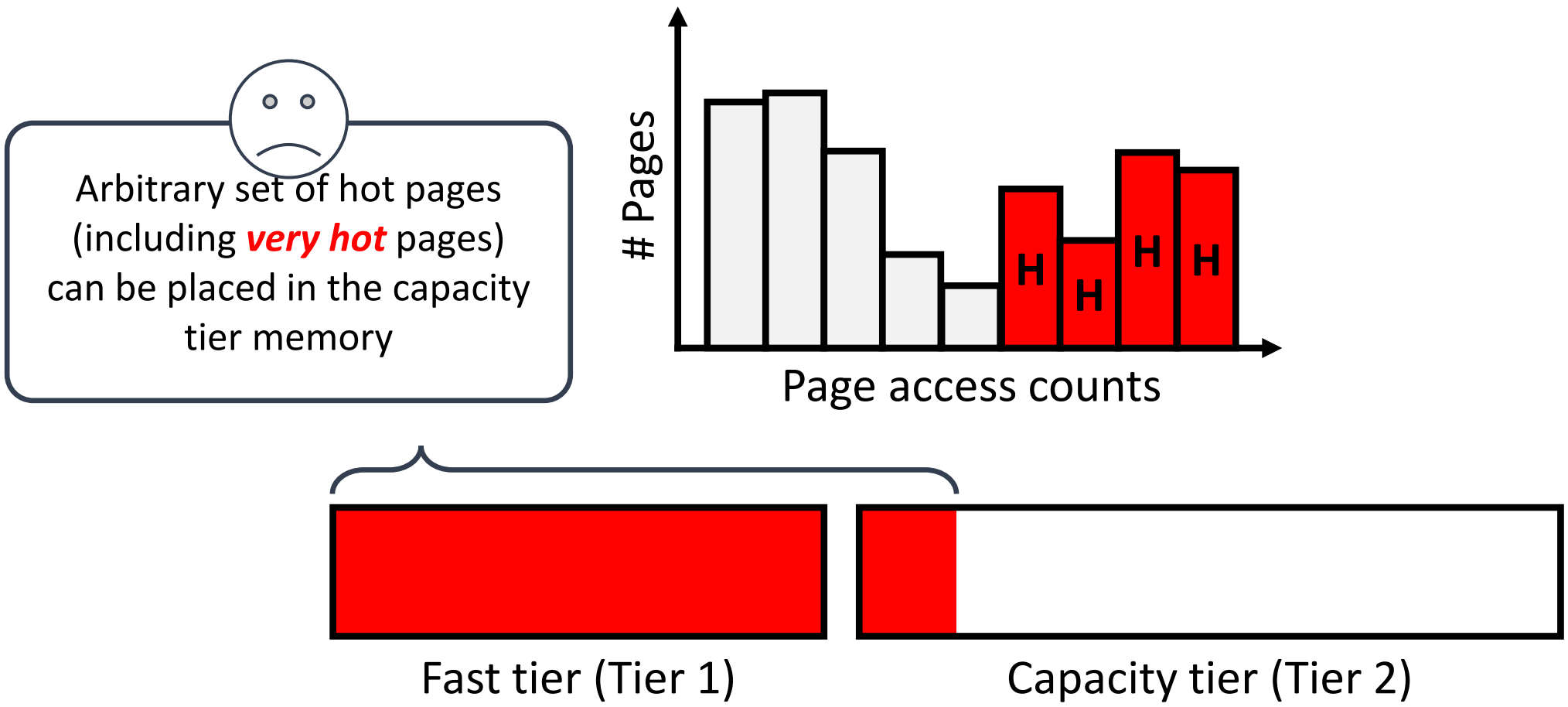
- Determining hot/warm/cold thresholds





Histogram-based hot set classification

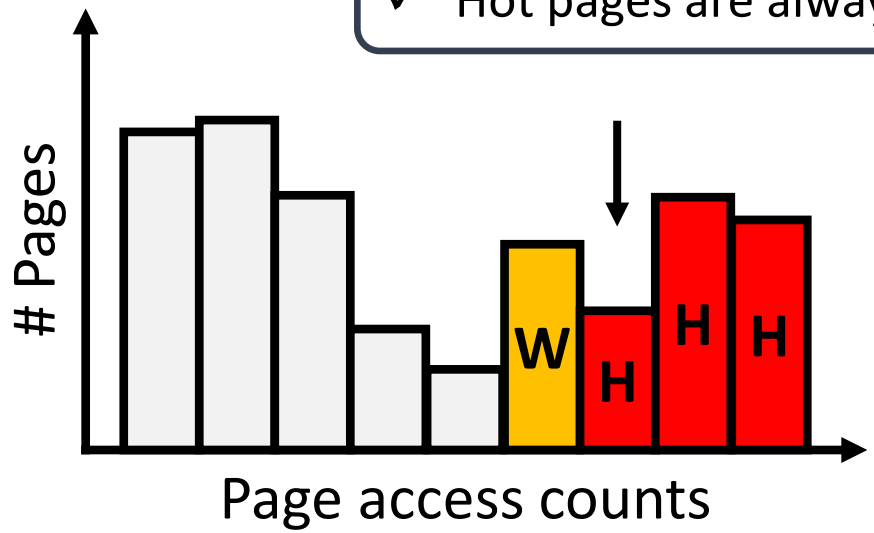
- Determining hot/warm/cold thresholds



Histogram-based hot set classification

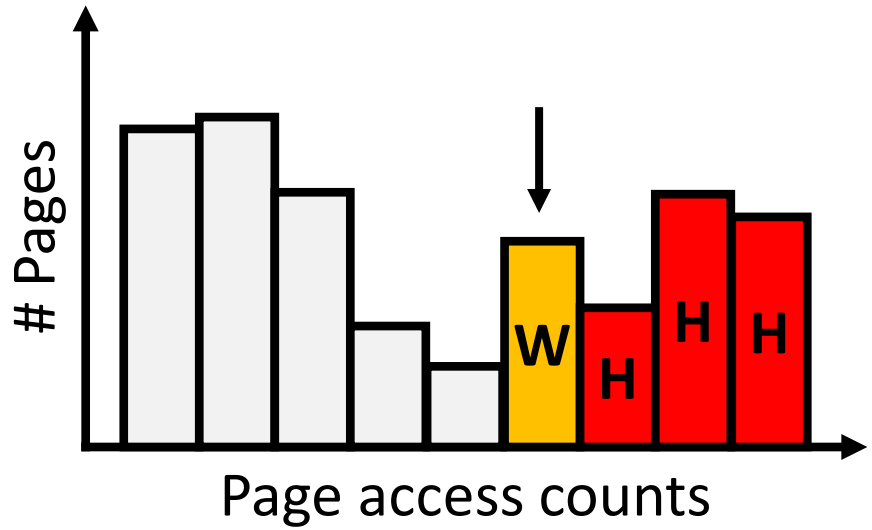
- Determining hot/warm/cold thresholds

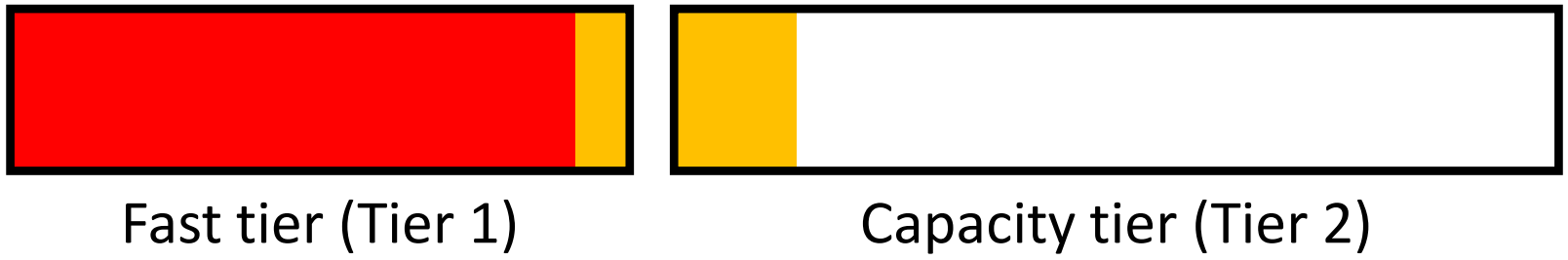
✓ Hot pages are always placed in fast tier memory



Histogram-based hot set classification

- Determining hot/warm/cold thresholds

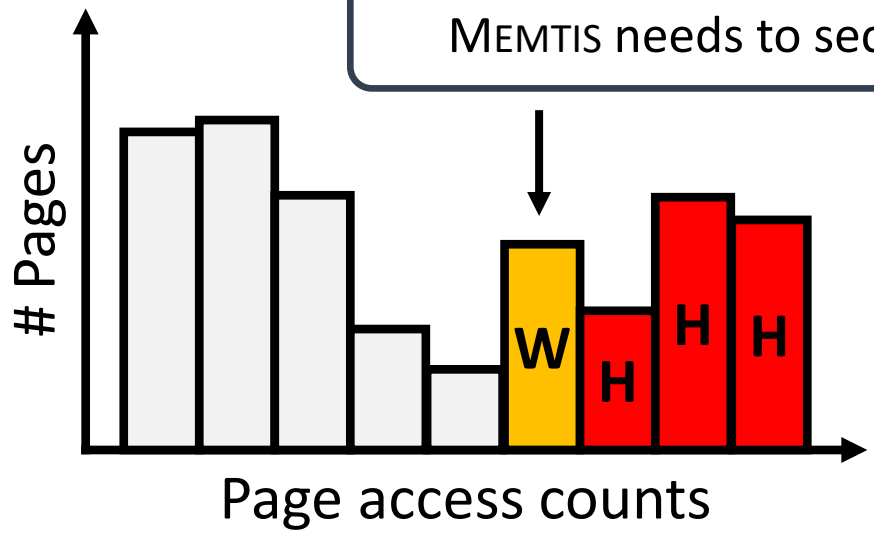




Histogram-based hot set classification

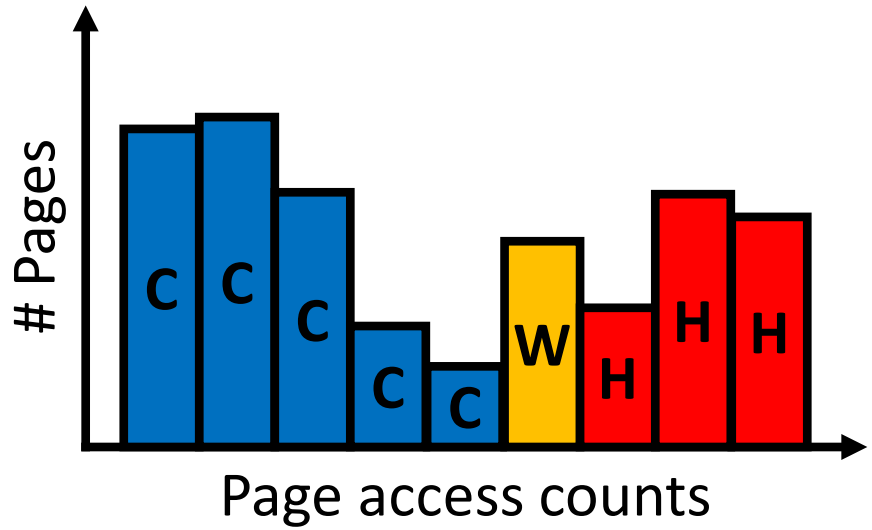
- Determining hot/warm/cold thres

✓ No promotion for warm pages
✓ No demotion unless there are cold pages in fast tier and MEMTIS needs to secure some free space



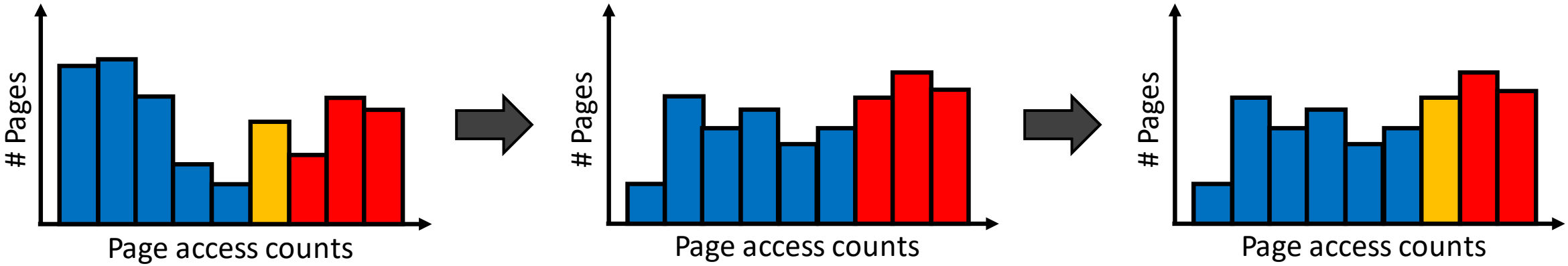
Histogram-based hot set classification

- Determining hot/warm/cold thresholds



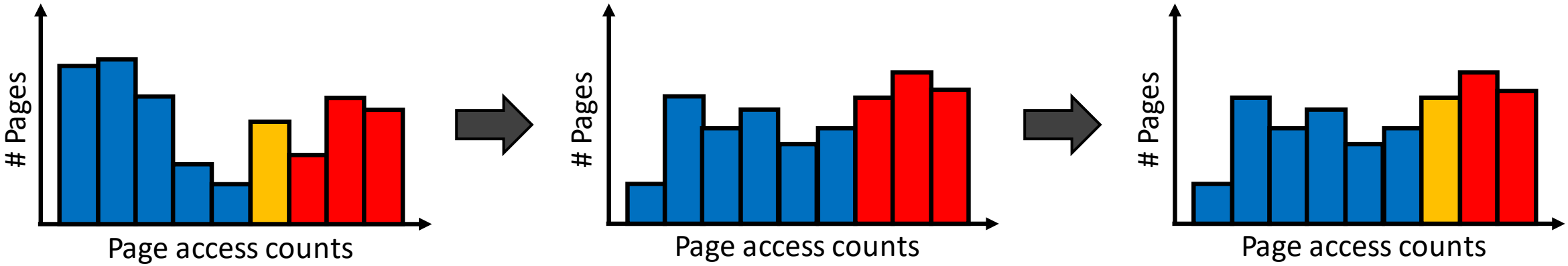
Histogram-based hot set classification

- Threshold adaptation
  - ✓ Maintain the hot set size not to exceed the size of fast tier memory



Histogram-based hot set classification

- Threshold adaptation
  - ✓ Maintain the hot set size not to exceed the size of fast tier memory



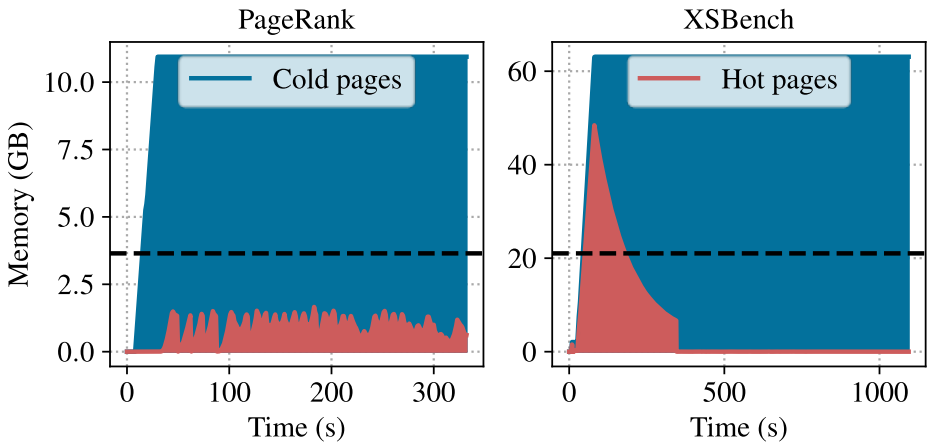
- Periodic cooling
  - ✓ Decay the impact of old accesses and give more weight to recent accesses
  - ✓ Exponential moving average of page access counts with a decay factor of 0.5 (halves every page's access count)

Evaluation setup

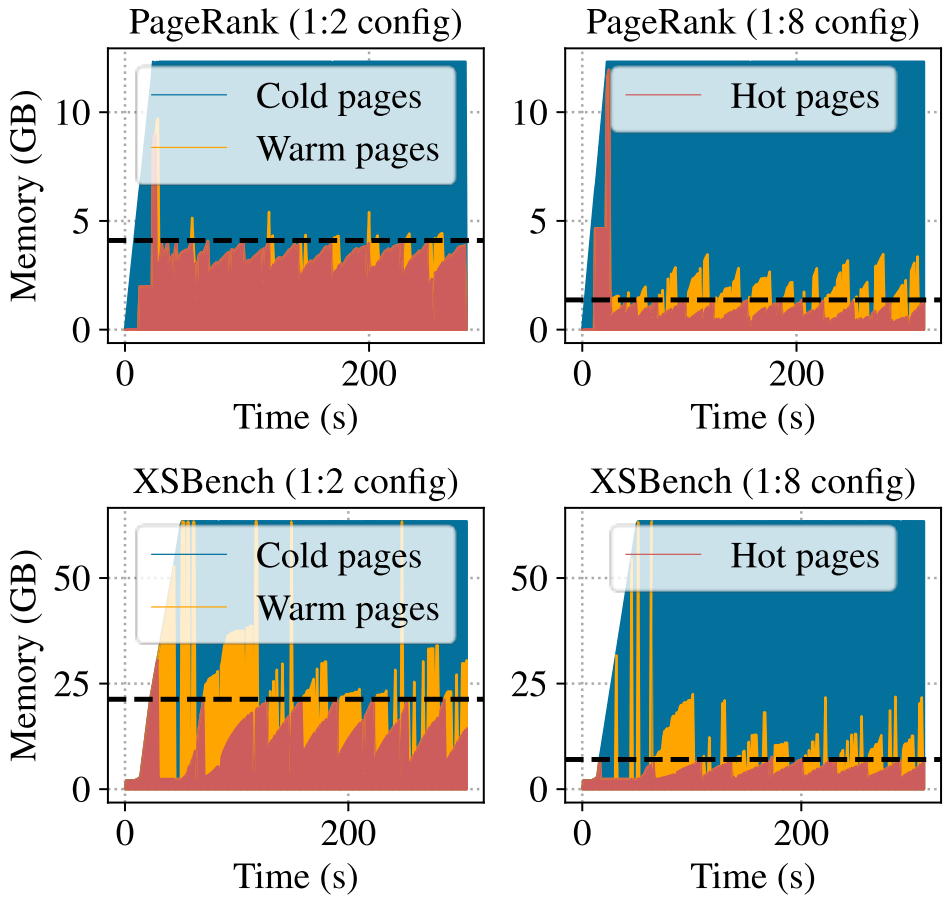
- Hardware environment
  - ✓ Intel Xeon 5218R @ 2.10Hz (Cascade Lake, 20 cores)
  - ✓ All DIMMs populated: [6 × 16GB DRAM] + [6 × 128GB Intel Optane DC PMM]
- Tiering configuration (fast tier size vs. capacity tier size)
  - ✓ Three configurations: 1:2, 1:8, 1:16
  - ✓ E.g., 1:2 config. → fast tier size is set to 33% of the RSS for each benchmark
- Competitors
  - ✓ AutoNUMA (Vanila Linux), HeMem [SOSP'21], TPP [ASPLOS'23]
  - ✓ Nimble [ASPLOS'19], AutoTiering [ATC'21], Tiering-0.8 in the paper

Page hotness identification

HeMem

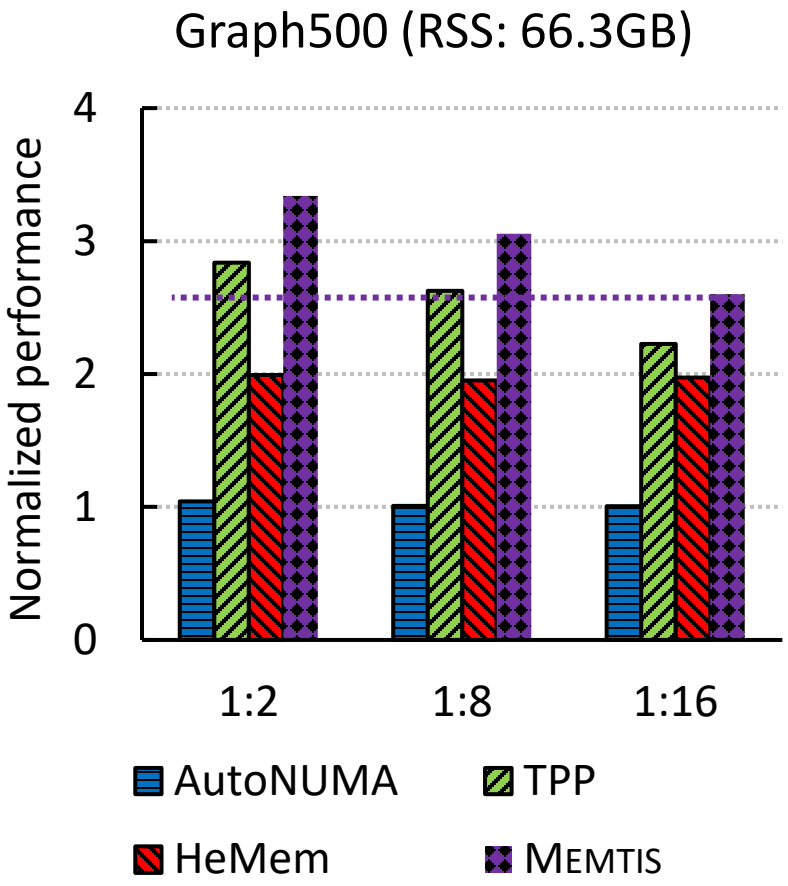


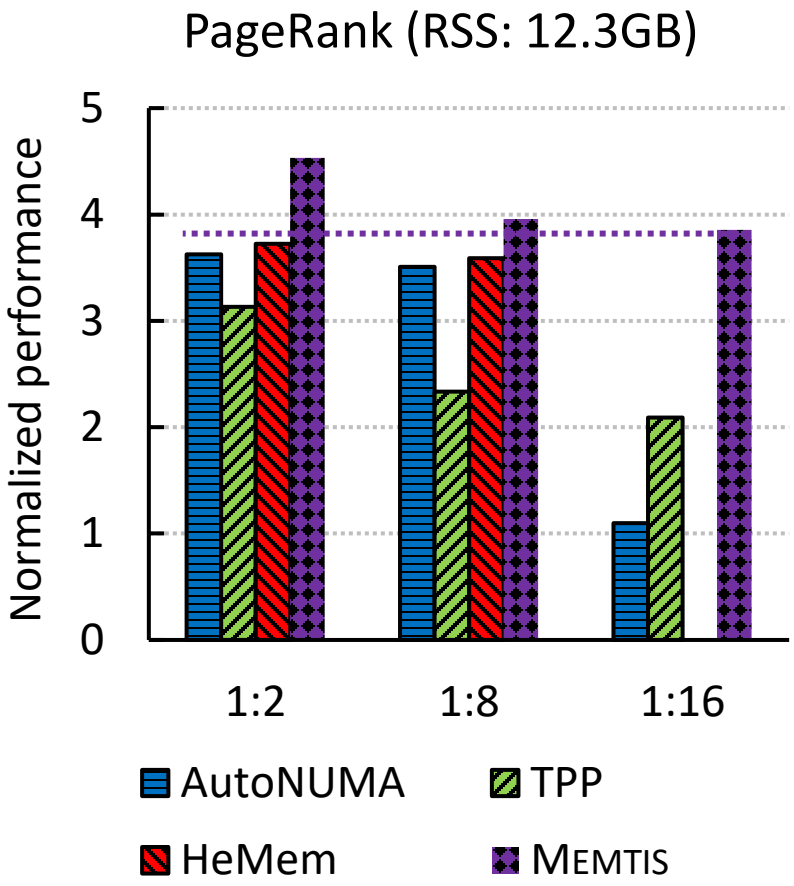
MEMTIS

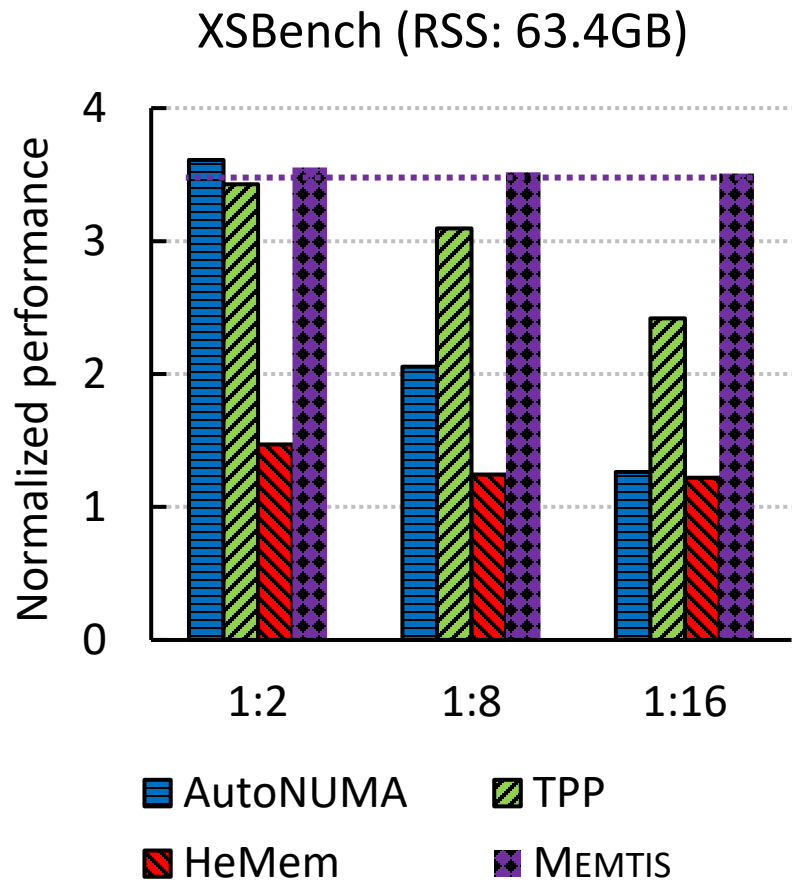


Performance comparison

- Normalized to all-NVM performance

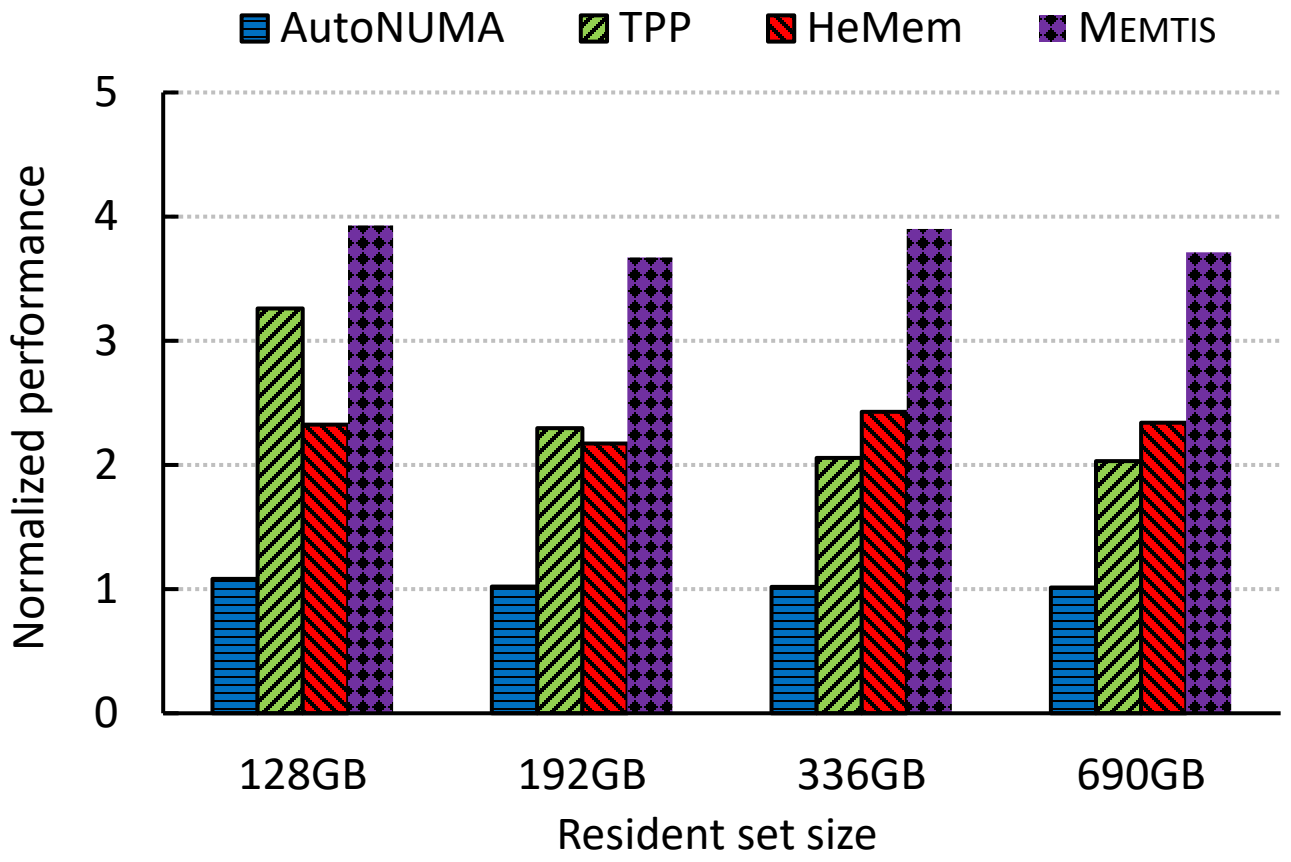






Scalability to memory sizes

- Increasing the RSS of Graph500 from 128GB to 690GB (Fast tier size: 64GB)
- PEBS-based systems become more effective as the RSS increases



Conclusion

- Efficient and transparent management of tiered memory should
  - ✓ Track memory access in a scalable way
  - ✓ Consider both diverse memory access patterns and memory configurations
  - ✓ Maintain the hot set size as close as possible to the fast tier size
- **MEMTIS**
  - ✓ Performs memory access tracking in a lightweight, fine-grained manner
  - ✓ Adjusts hotness thresholds based on the page access distribution
  - ✓ (Dynamically decides page size for better utilization of fast tier memory)

Thank you!