P-MOSS: Scheduling Main-Memory Indexes Over NUMA Servers Using Next Token Prediction

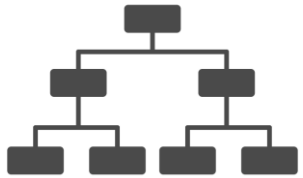
Yeasir Rayhan and Walid G. Aref
Purdue University
West Lafayette, Indiana, USA

PURDUE UNIVERSITY

Overview

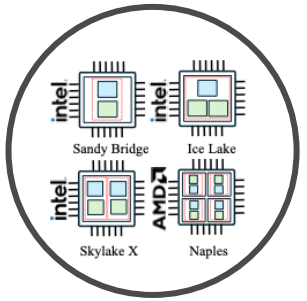
P-MOSS A Learned Performance MOnitoring Unit (PMU)-driven Spatial Query Scheduling Framework.

What Problem Does P-MOSS Solve?



Spatial Scheduling of a Main-Memory Index

CPU Vendor



NUMA / Chiplet Architecture

Data & Query Workload

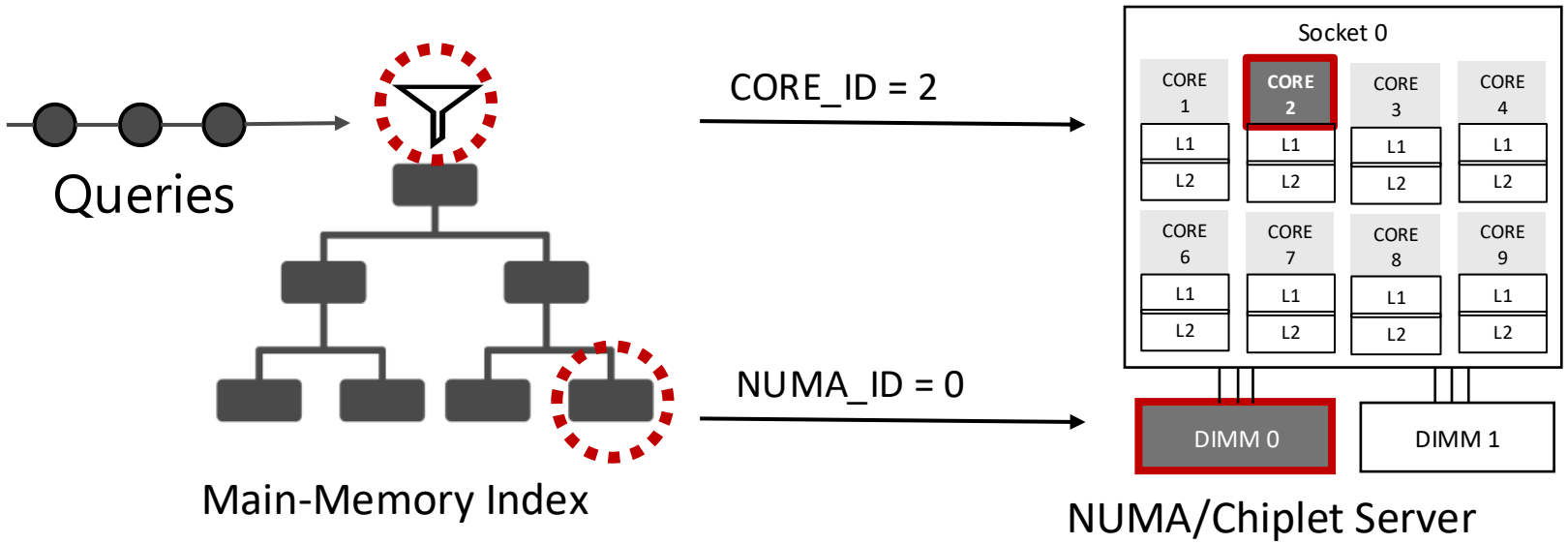
Seen and Unseen

The LLM Recipe	Profile-Guided Optimization	Offline RL
Next-token prediction, GPT backbone and 2-stage training.	HW performance statistics from HW PMU guide the scheduling decision.	Decouples training from the DMBS critical path.

What is Spatial Scheduling?

Spatial Scheduling [1] co-optimizes the following.

1. Core Scheduling Strategy
   Selects the CORE_ID that executes a query for a given index partition.
2. Data Partitioning Strategy
   Assigns each index partition to a NUMA_NODE (its memory location).

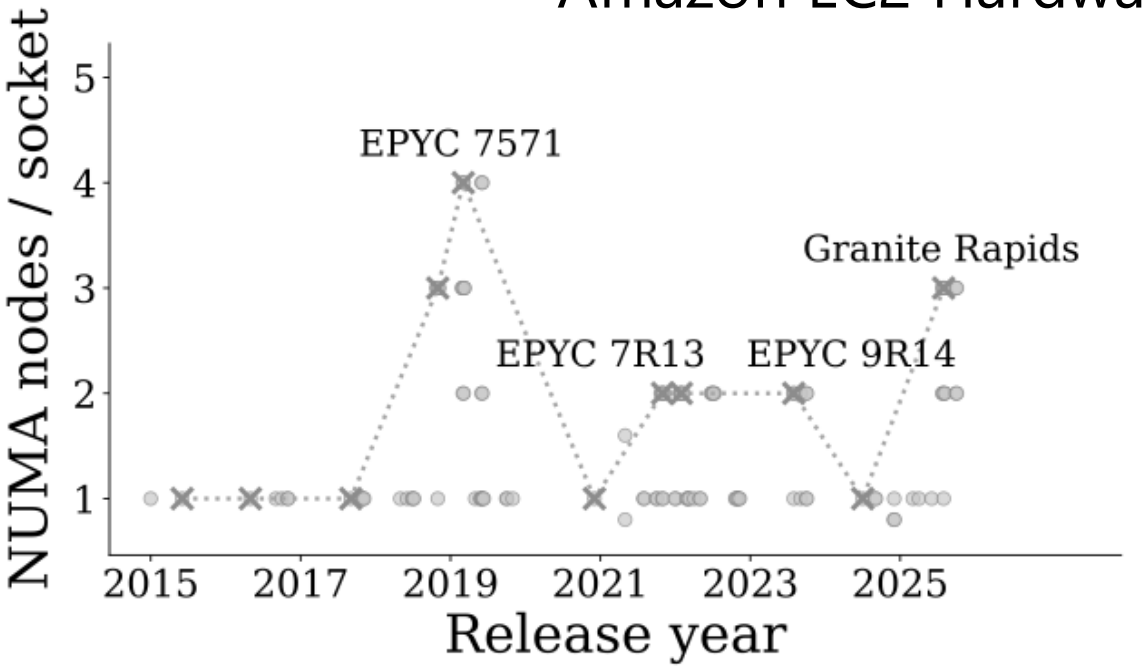


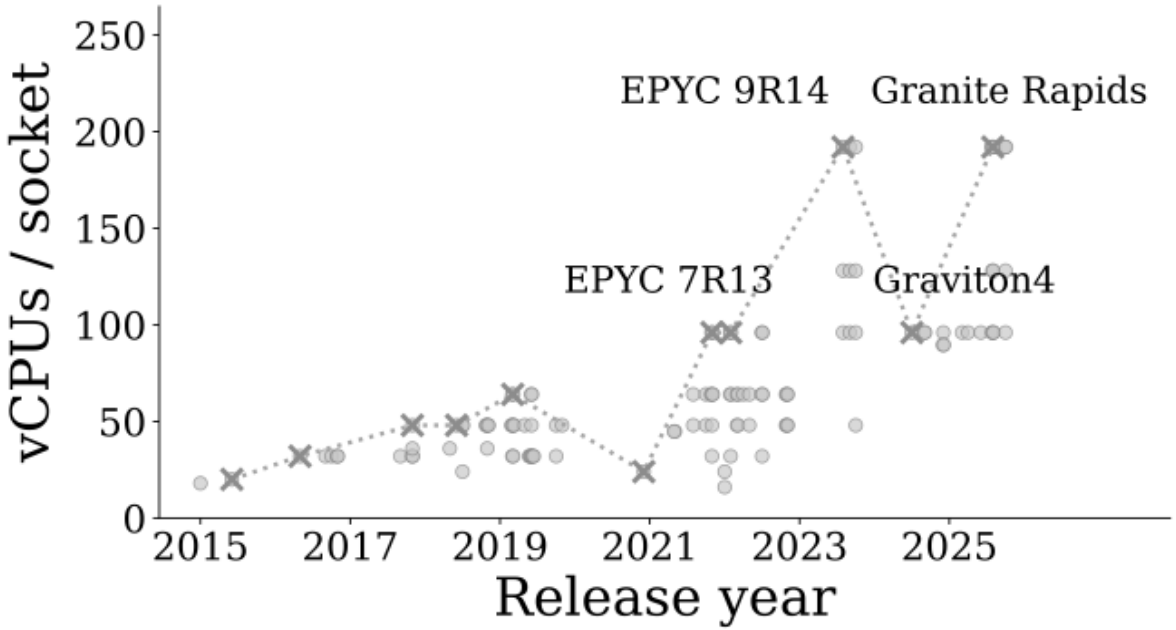
[1] Jana Giceva, Gustavo Alonso, Timothy Roscoe, and Tim Harris. Deployment of Query Plans on Multicores. In VLDB, 2014.

Motivation: Hardware Trends Are Shifting

1. Chiplet Era
   4× more NUMA domains per socket.
2. Many-Core Era
   8× more vCPUs per socket.

Amazon EC2 Hardware Landscape [2, 3]

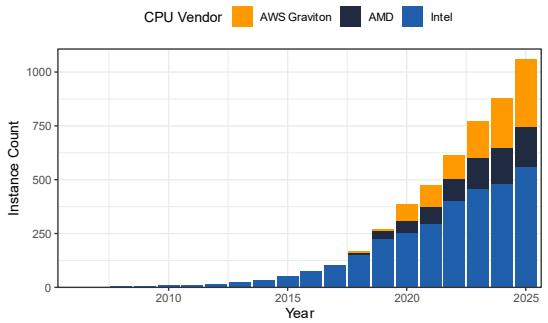




[2] Till Steinert, Maximilian Kuschewski, Viktor Leis. Cloudspecs: Cloud Hardware Evolution Through the Looking Glass. In CIDR, 2026.
[3] https://instances.vantage.sh/

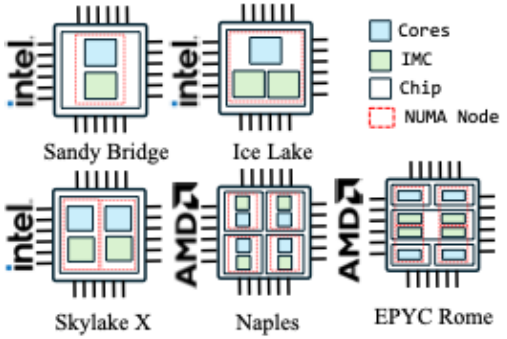
Challenges

1. Diverse CPU vendors with an exponential number of instance choices



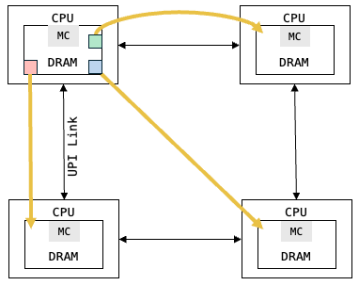
> 1K Amazon EC2 instance choices by 2025 [2].

2. Diverse CPU Architecture with different CPU topology



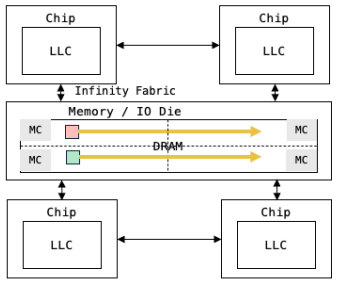
3. Hardware heterogeneity*

NUMA: Cross-Socket Transfer



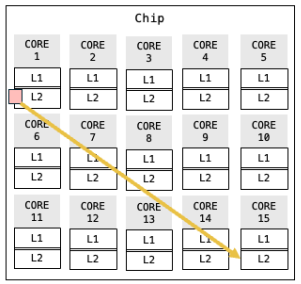
10× Expensive

Chiplet: Cross-Chip Transfer



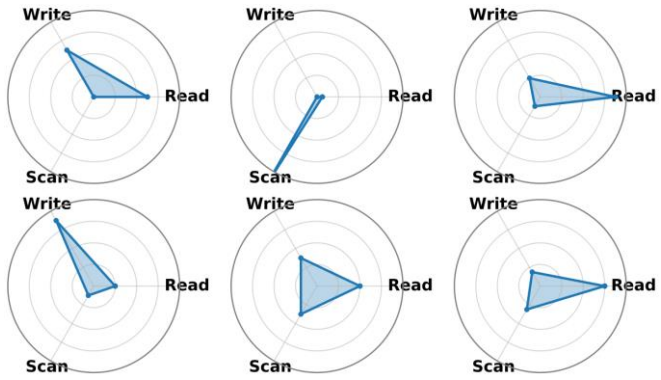
4× Expensive

100+ Core Server: Cross-Core Transfer



1.5× Expensive

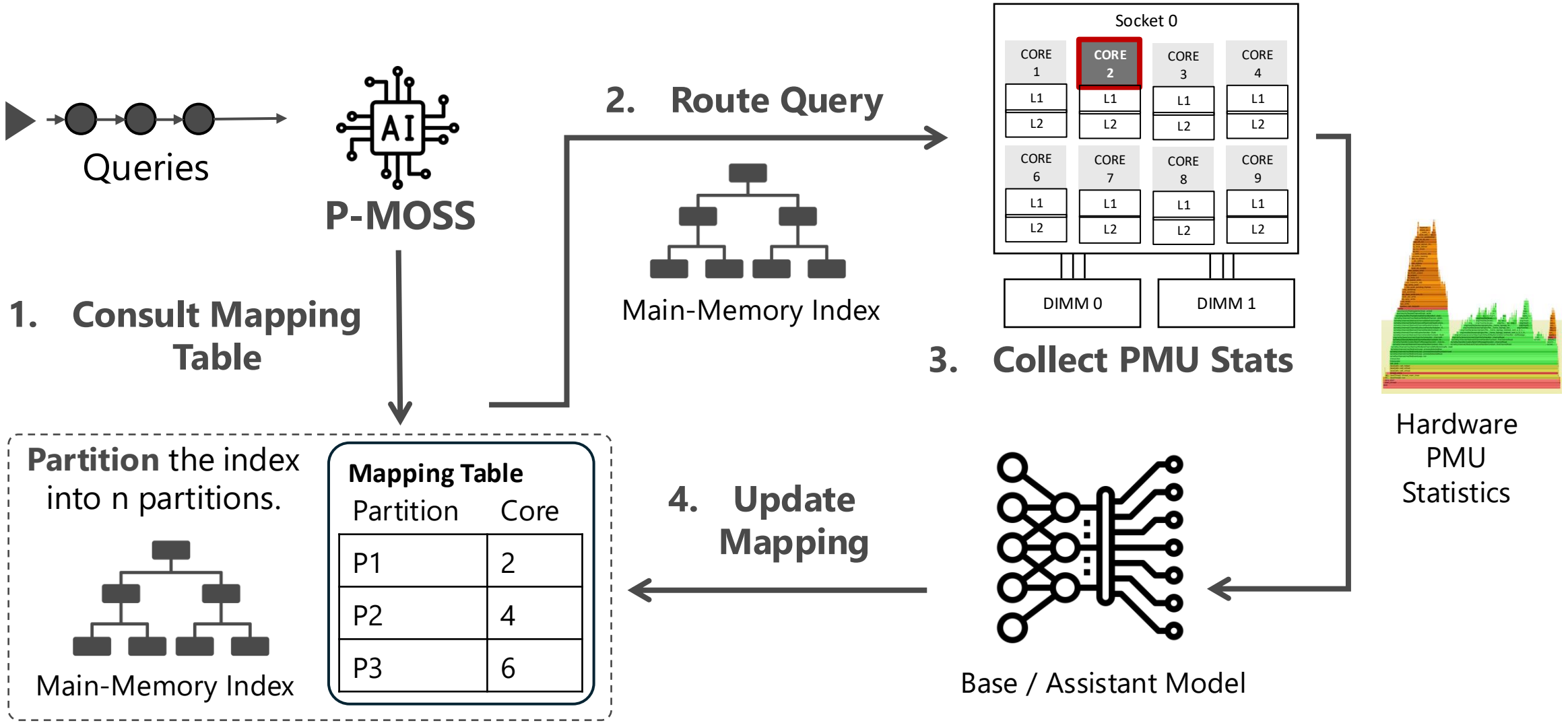
4. Diverse data and query workload with varying access patterns.



*All numbers are calculated in our own testbed.

PURDUE UNIVERSITY

P-MOSS Scheduling Loop



Core Learning Principles of P-MOSS

The LLM Recipe
Next-token prediction, GPT backbone
and 2-stage training

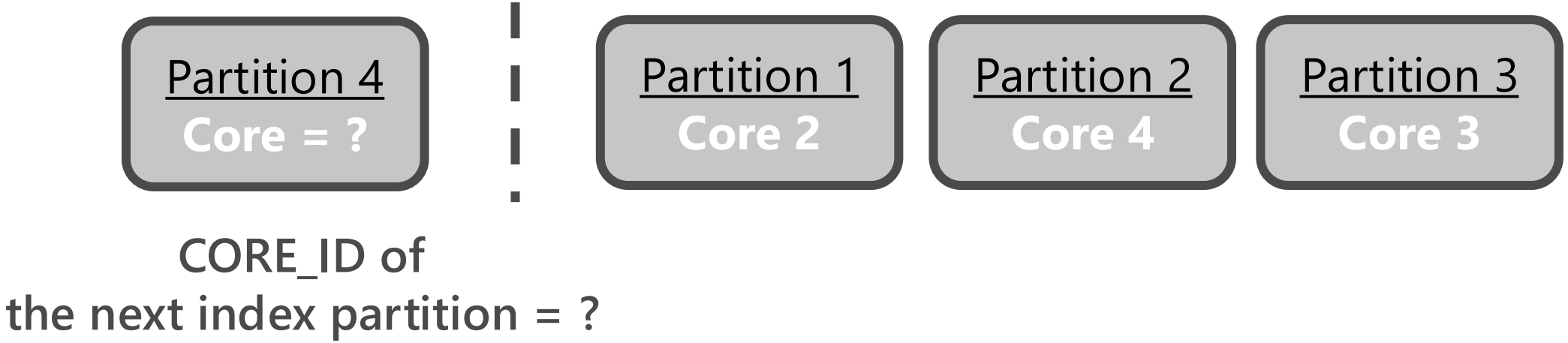
Profile-Guided Optimization
HW performance statistics from HW PMU guide the scheduling decision.

Offline RL
Decouples training from the system's critical path.

The LLM Recipe: Next Token Prediction (NTP)

The LLM Recipe	Profile-Guided Optimization	Offline RL
Next-token prediction, GPT backbone and 2-stage training	HW performance statistics from HW PMU guide the scheduling decision	Decouples training from the system's critical path

1. Next Token Prediction (NTP).
   - Formulate spatial scheduling as an NTP problem.
   - Predict the CORE_ID of the next index slice.
     - Data Partitioning is implicit. The corresponding NUMA NODE of the core stores the data.



The LLM Recipe: Two-Stage Training Pipeline

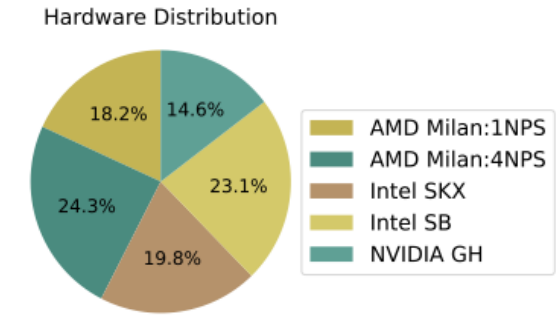
The LLM Recipe
Next-token prediction, GPT backbone
and 2-stage training

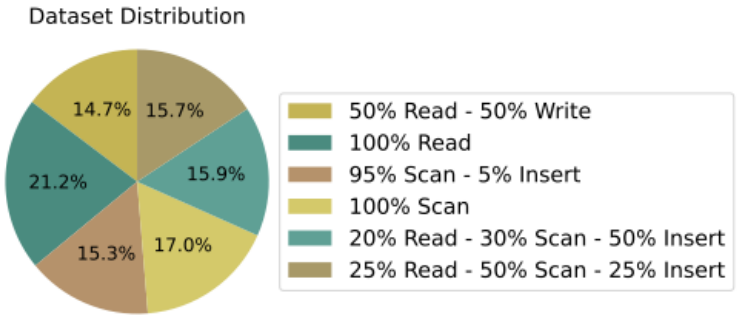
Profile-Guided Optimization
HW performance statistics from HW PMU guide the scheduling decision

Offline RL
Decouples training from the system's critical path

2. Two-stage training pipeline
   - Pre-Training → Base Model
     - Train on 8K distinct scheduling policies across diverse hardware and query workload.
     - Equivalent to internet-scale pre-training for LLMs.
   - Post-Training → Assistant Model
     - Train on 200-300 scheduling policies from the target hardware and query workload.
     - Equivalent to task-specific LLM fine-tuning.

Pre-Training Corpus



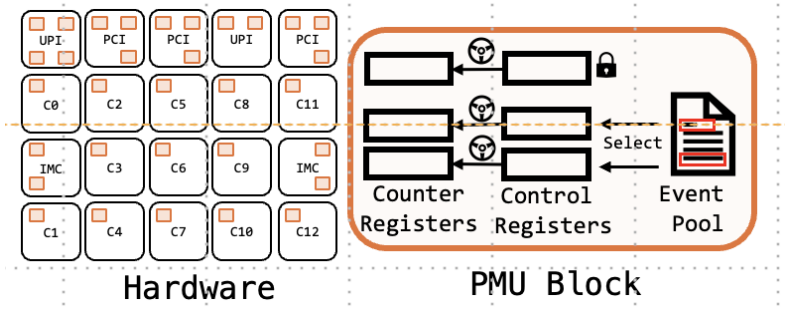


Profile-Guided Optimization (PGO)

The LLM Recipe	**Profile-Guided Optimization**	Offline RL
Next-token prediction, GPT backbone ~~and 2-stage training~~	HW performance statistics from HW PMU guide the scheduling decision.	Decouples training from the system's critical path.

- PGO. Use execution profiles collected from prior runs to optimize future execution.
  - A standard compiler optimization technique.
- Why PGO?
  - **Provides the abstraction across diverse hardware and workload, enabling transferable scheduling policies to unseen environment**.

- P-MOSS profiles 39 HW PMU counters from the HW PMU
  - The number of instructions executed, L1-I (L1-Instruction) cache misses, L1D (L1-Data) cache misses, LLC (Last Level Cache) cache misses, …
- Low overhead profiling
  - rdpmc instruction overhead: 9 – 27 clock cycles [4]



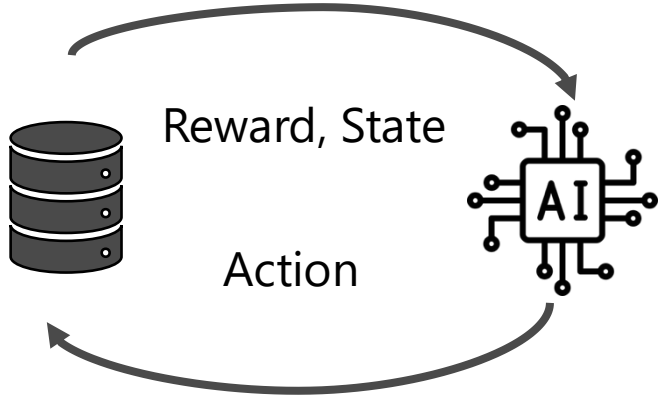
Each hardware component contains PMU blocks with 1 - 8 counters.

[4] Agner Fog. 2025. Software Optimization Resources. https://www.agner.org/optimize/instruction_tables.pdf

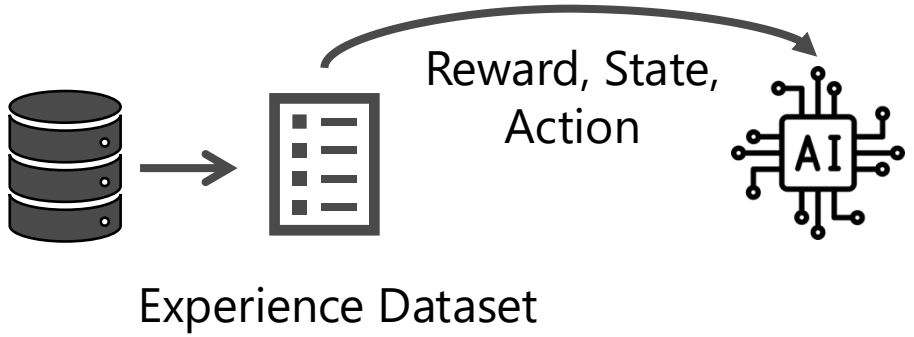
Offline Reinforcement Learning

The LLM Recipe	Profile-Guided Optimization	**Offline RL**
Next-token prediction, GPT backbone and 2-stage training	HW performance statistics from HW PMU guide the scheduling decision.	Decouples training from the system's critical path.

- The agent learns a policy from an experience dataset, without any active interaction with the database.
  - Decouples training from the DBMS critical path.
  - Enables scalable training over large datasets.

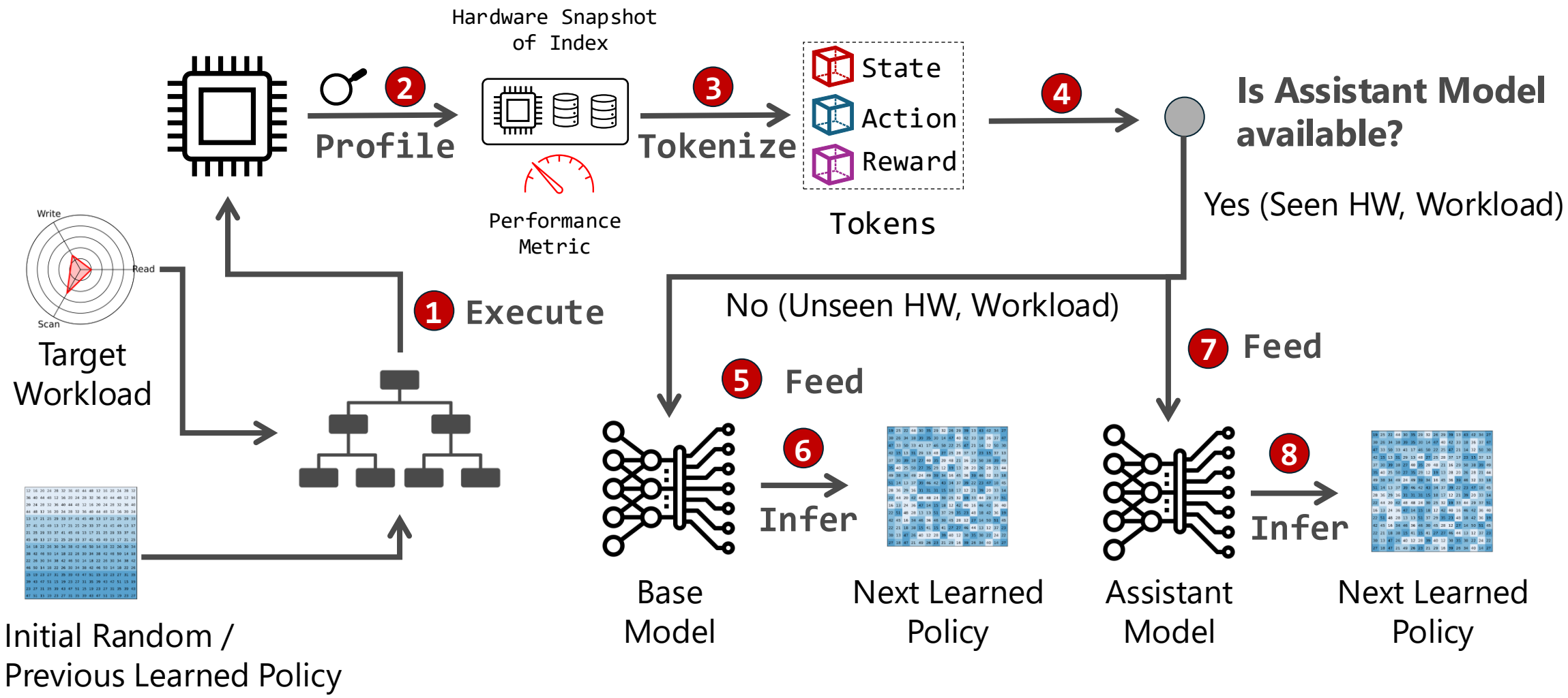


Online RL
(Feedback Loop)



Offline RL
(No Feedback Loop)

P-MOSS Runtime



Experiments: Setting

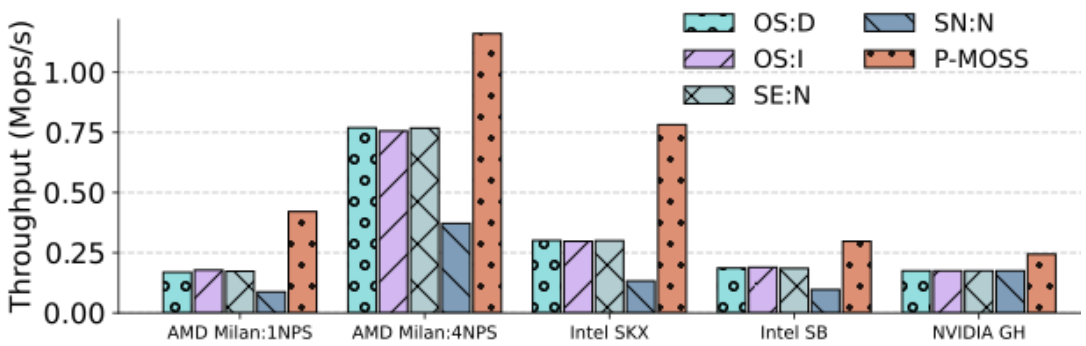
- Dataset
  - YCSB benchmark with **1000M** records.
- Hardware
  - AMD Milan (1NPS)
  - AMD Milan (4NPS)
  - Intel Skylake X
  - Intel Sandy Bridge
  - NVIDIA H200 Grace Hopper
  - IBM Power System S822LC
- Baselines
  - OS Default (OS:D). OS-managed core scheduling and first-touch data placement.
  - OS Interleave (OS:I). OS-managed scheduling with interleaved data placement.
  - Shared Everything-NUMA (SE:N) [5]. OS scheduling with NUMA-aware data placement.
  - Shared Nothing-NUMA (SN:N) [5]. NUMA-aware core scheduling and data placement.

[5] Danica Porobic, Ippokratis Pandis, Miguel Branco, Pinar Tözün, and Anastasia Ailamaki. OLTP on Hardware Islands. In VLDB, 2012

Experiments: Static YCSB Workload

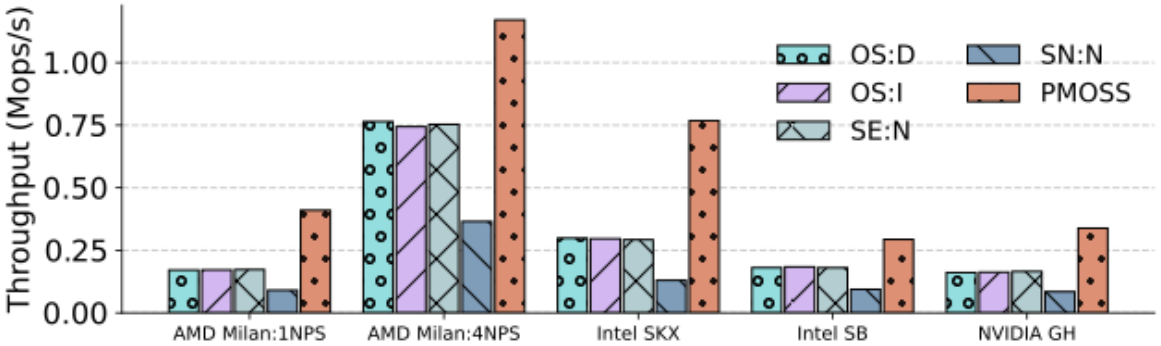
YCSB-A 50% Read 50% Write

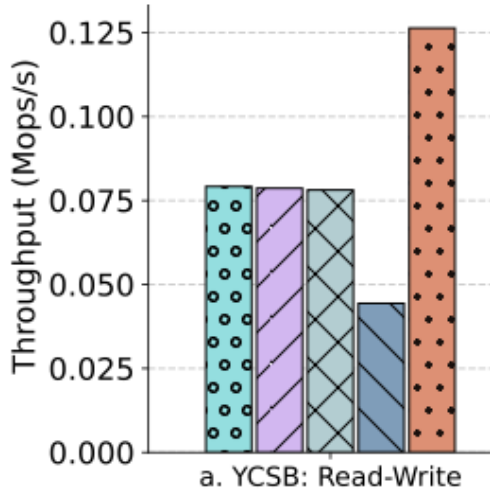
1.90× speedup (Seen HW)



YCSB-C: 100% Read

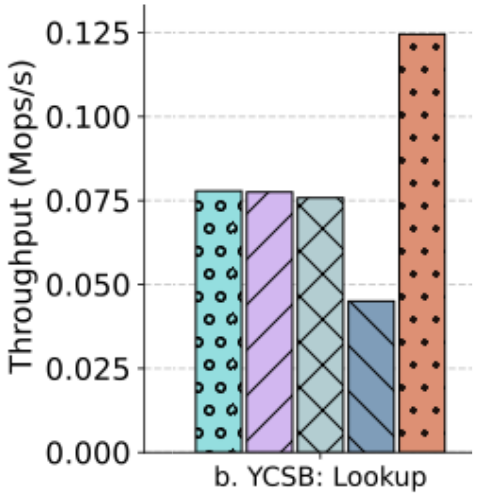
2.57× speedup (Seen HW)





1.46× speedup (Unseen HW)

IBM Power System S822LC

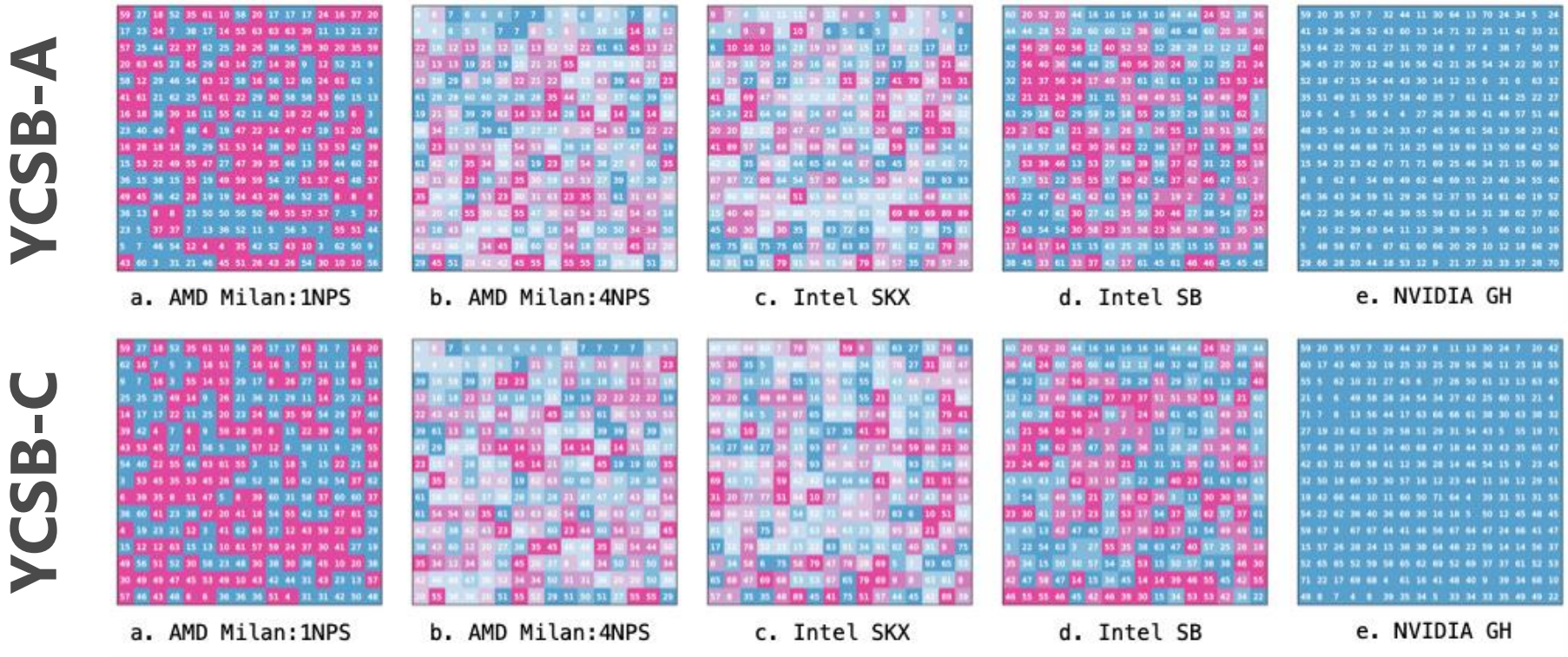


1.46× speedup (Unseen HW)

IBM Power System S822LC

Snippet of P-MOSS's Learned Scheduling Policies

- Learned Policies are different from the observed policies in the training set.
- Different HW, workload → Different learned policies



*Cells with the same color are scheduled on the same NUMA node.

Takeaways

1. Spatial scheduling matters in modern hardware!
2. No single heuristic generalizes across all hardware and workloads.
3. Next Token Prediction (NTP) enables offline RL and data-driven learning with GPT-style models.
4. Hardware PMU statistics provide effective signals for learned scheduling (ML4DB in general)!

Thank You!

Questions?

Read our paper!

Get in touch!

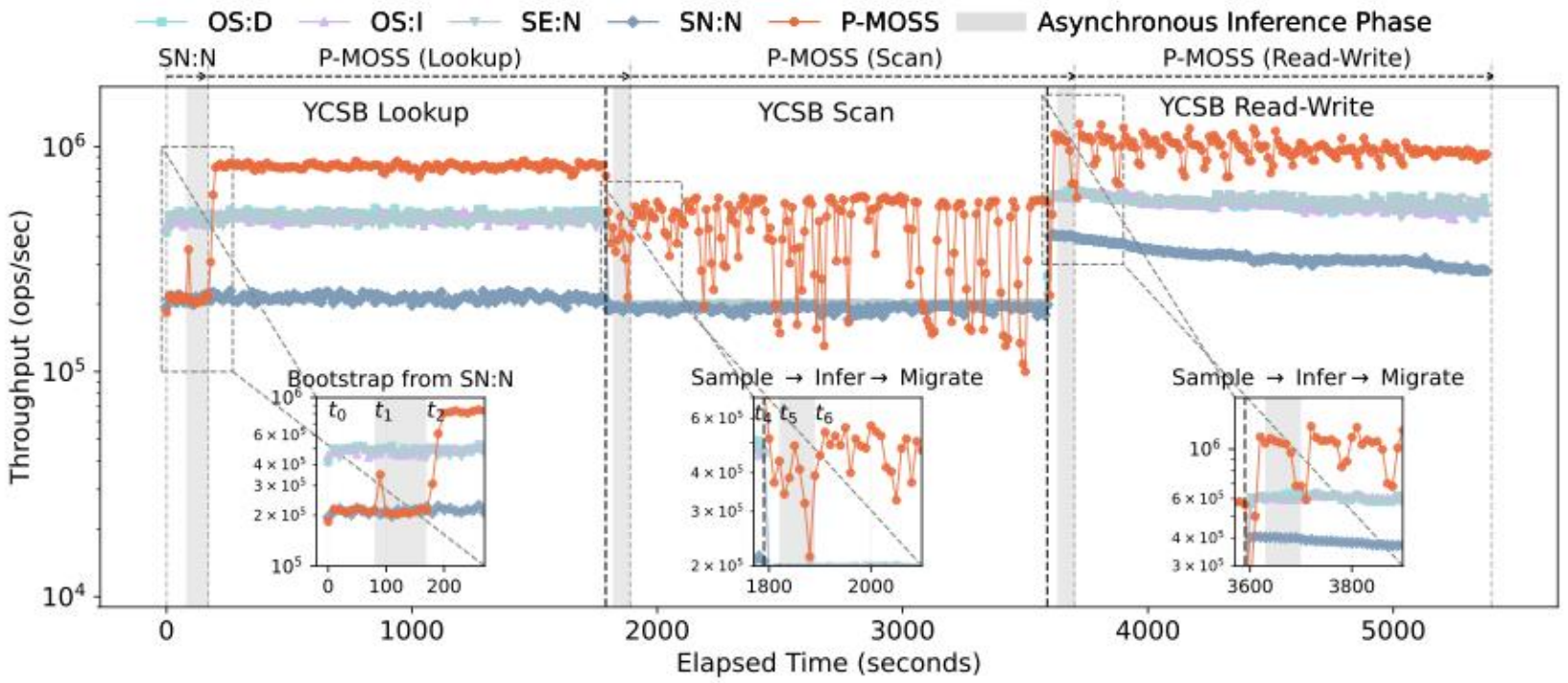
Backup Slides

PURDUE UNIVERSITY

Experiments: Dynamic YCSB Workload

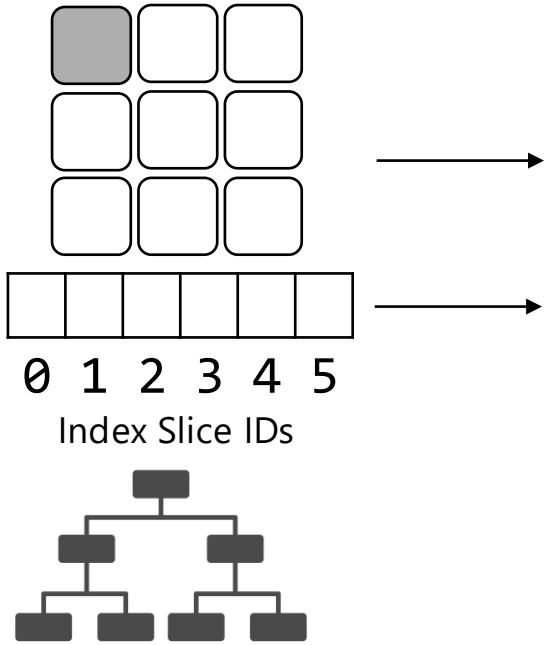
YCSB-C → YCSB-E → YCSB-A

1.72× speedup



The LLM Recipe: Next Token Prediction

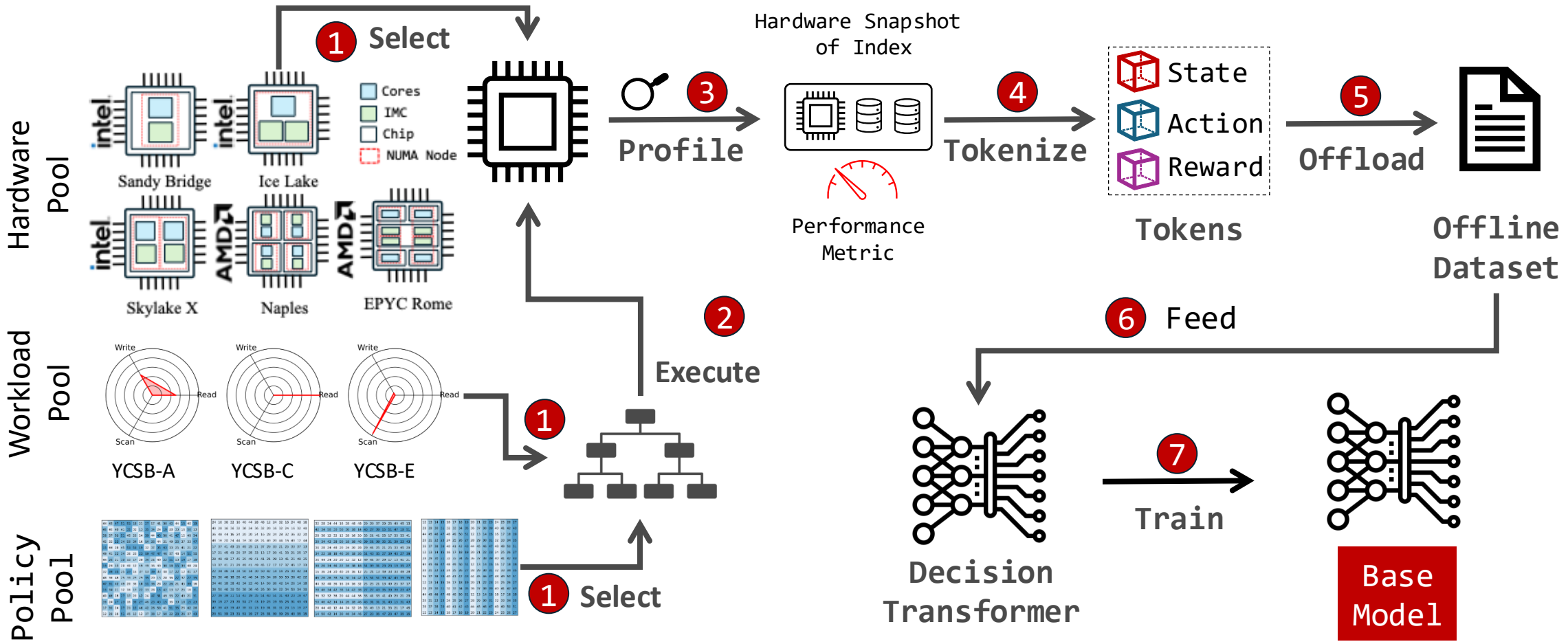
- Predict the CORE_ID of the next slice.
  - Data Partitioning is implicit. The corresponding NUMA NODE of the core stores the data.



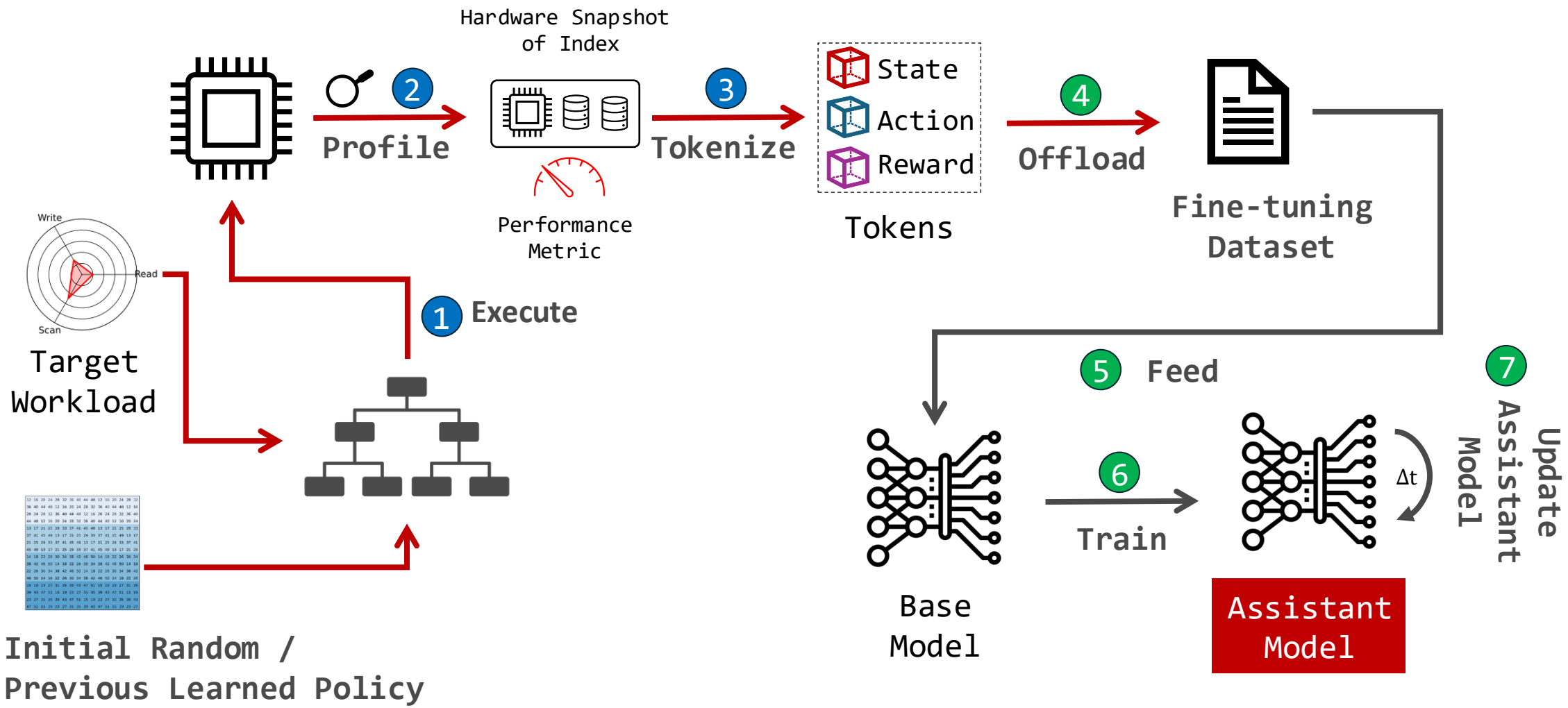
The hardware is represented as a 2D-die, where each cell represents a core with its own CORE_ID.

Each index is logically partitioned into multiple index slices based on the index key.

The LLM Recipe: Pre-Training



The LLM Recipe: Post-Training



Decision Transformer (DT)

- Decision Transformer (DT) treats RL as a supervised conditional sequence modeling problem, i.e., Next Token (Action) Prediction.
- DT represents each policy in the experience dataset as a sequence of ⟨Reward-to-go, State, Action⟩ tuples.
  - Reward-to-go Token (RTG Token)
    - Expected future cumulative reward, e.g., expected query throughput.

An example trajectory