Exploring Next Token Prediction For Optimizing Databases

Yeasir Rayhan and Walid G. Aref

Purdue University

West Lafayette, Indiana, USA

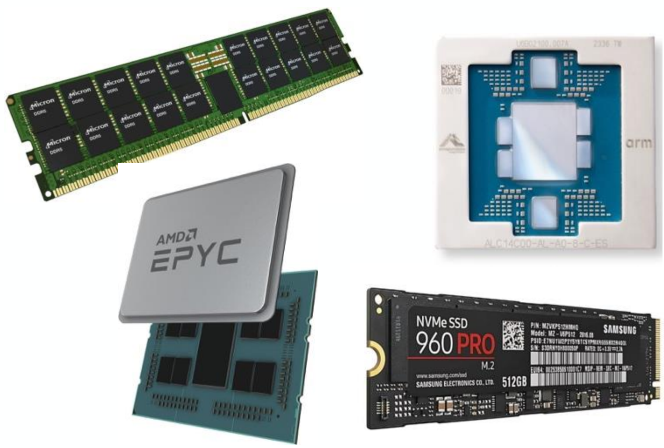
PURDUE UNIVERSITY®

Roadmap

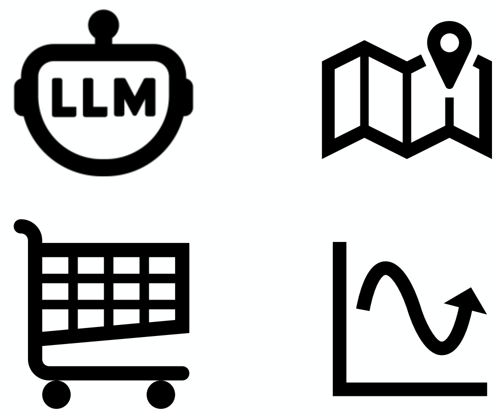
- The current landscape of Database Systems design space
  - Hardware and Workload
- The LLM recipe
  - Next Token Prediction (NTP): First step towards adopting the LLM recipe
- The building blocks to adopt Next Token Prediction (NTP) in databases
  - Decision Transformers
  - DB-Tokens
- The Probe and Learn (PoLe) framework
  - Preliminary case study
    - Index Scheduling

The Game Changers in Databases [1]

Hardware



Workload



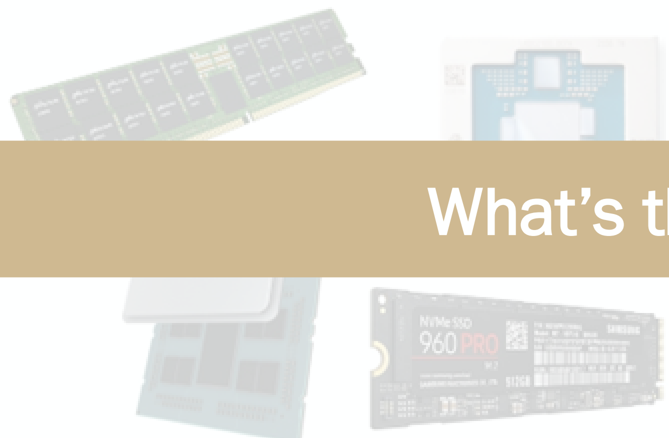
[1] Anastasia Ailamaki. 2021. Accelerated Data Management Systems Through Real-Time Specialization. Keynote at MICRO.

PURDUE UNIVERSITY

The current state

Hardware

Workload

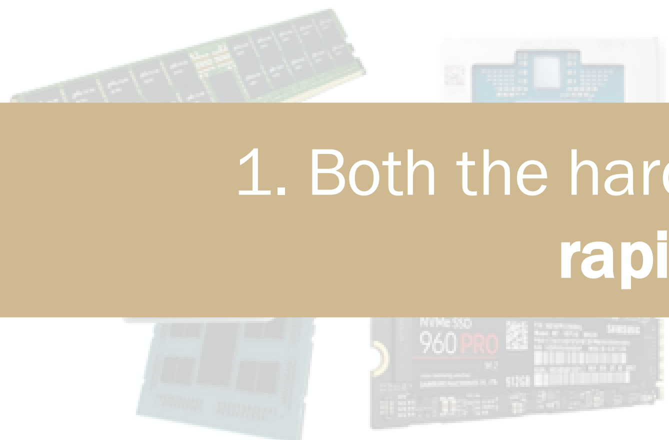


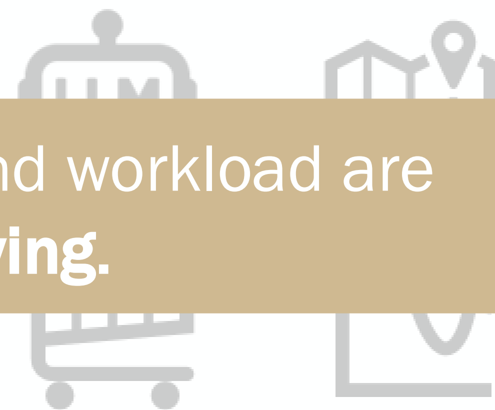
What's the current state?

The Current State: Observation 1

Hardware

Workload





1. Both the hardware and workload are **rapidly evolving.**

Observation 1: Rapid Evolution of Hardware

- Every few years, there is a new technology:
  - Compute, Memory and Storage
- Within the past 6 years, we have seen the commercial introduction of
  - Processing In Memory (PIM) chips by UPMEM
  - Compute Express Links (CXL)
  - ARM-based server processors

AMD Opteron
NUMA Architecture
2003

2008
Intel Nehalem
NUMA Architecture

Intel Haswell
CoD Architecture
2013

2015
Intel Skylake X
SNC Architecture

AMD Zen
Chiplet Architecture
2017

2018
AWS Graviton
ARM Server

Compute Express Link
CXL
2019

2019
Intel Optane DC
PM Module

UPMEM
PIM Architecture
2019

2022
Samsung **CXL**
Memory Expander

PURDUE UNIVERSITY

Observation 1: Rapid Evolution of Workload

- The applications that databases need to support are continuously growing.
- Within the past 3-4 years, we have seen a drastic shift in the ML workloads.
  - Large Language Models (LLM)

2023
Amazon Redshift ML
LLM invocation

2023
BigQuery ML
SQL-only LLM Text Generation

2024
Databricks
ai-query() function with foundation models

2024
Oracle Heatwave Gen AI
In-Database LLM

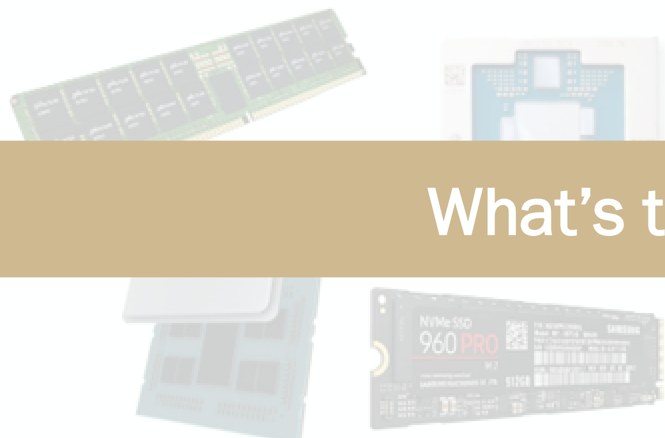
2024
SAP HANA vector engine
Data Analysis with LLMs

PURDUE UNIVERSITY

The current state

Hardware

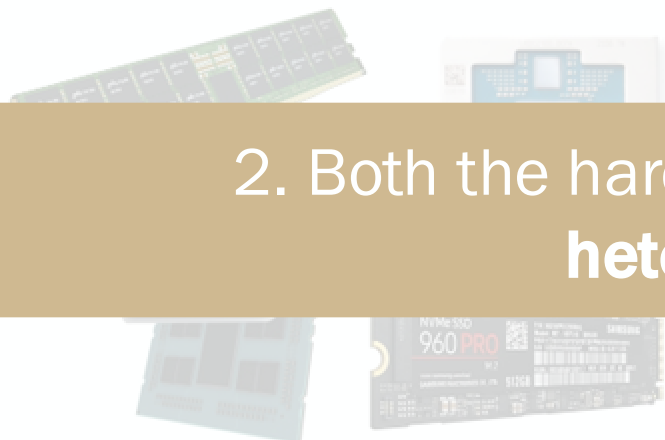
Workload



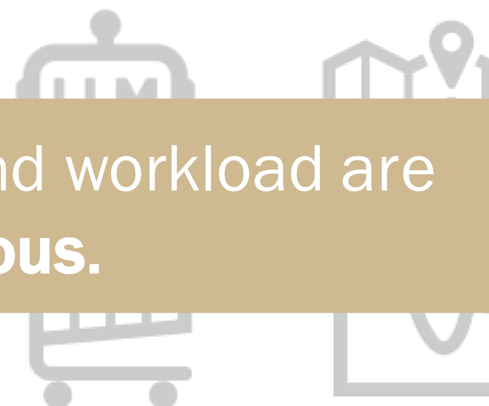
What’s the current state?

The Current State: Observation 2

Hardware

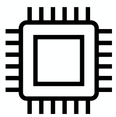


Workload



2. Both the hardware and workload are **heterogeneous.**

Heterogeneity in Hardware



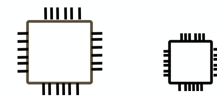
intel
NVIDIA
AMD
aws

Different Vendors

SAMSUNG
ASUS
micron

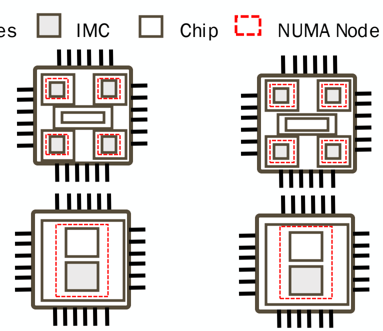
x86
RISC IBM Power RISC-v

Different Processor Architecture

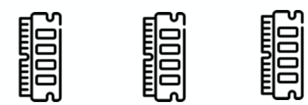


P-core E-core

Compute Cores



NUMA VS Chiplet Architecture



DRAM PMEM CXL Memory

Memory

PURDUE UNIVERSITY

Heterogeneity in Workload

ML Workload
LLM Queries
Training Data Preparation
Real-time Prediction

Spatial Workload
Location based Services
Geographic Information Systems
Moving Objects

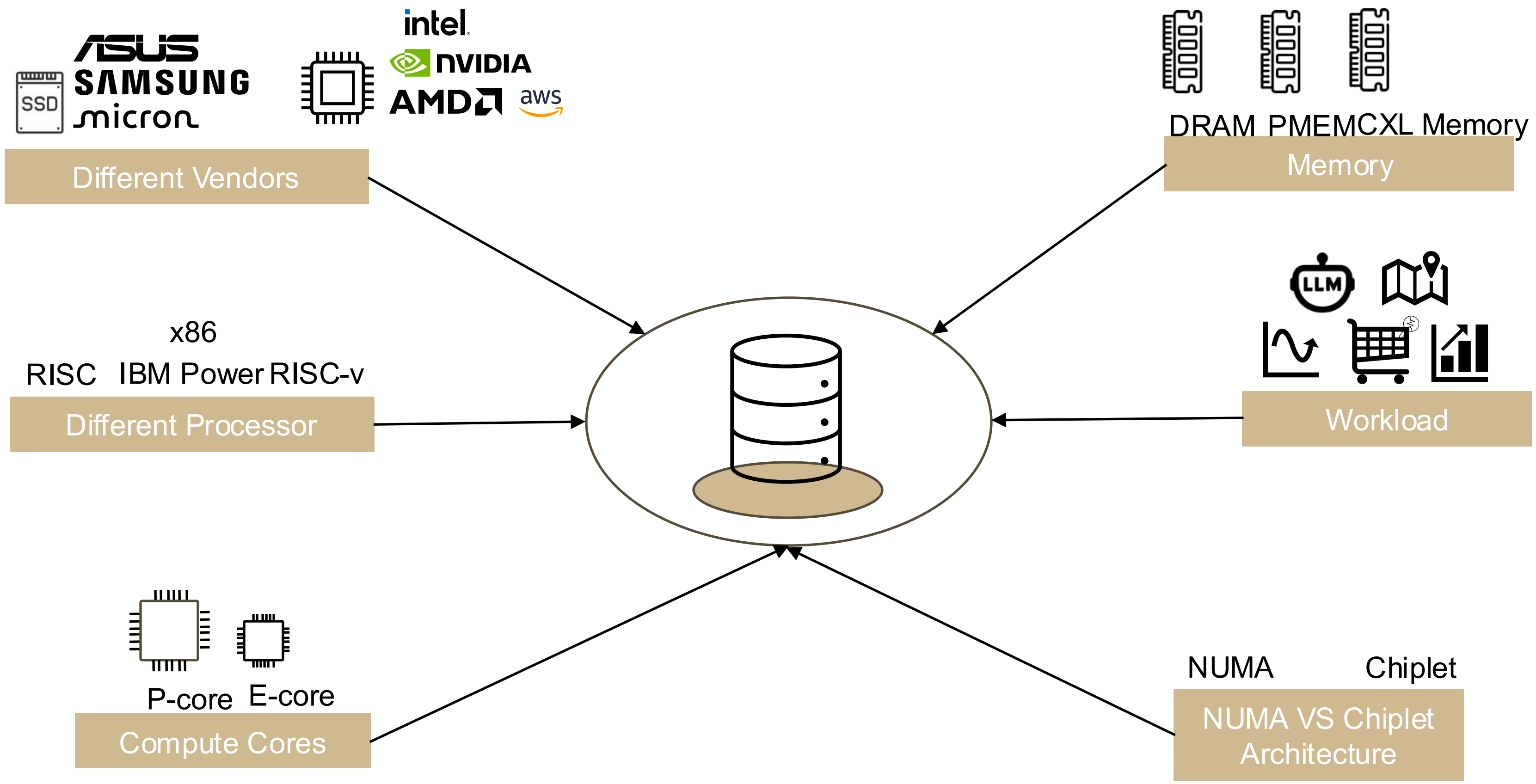
OLTP

HTAP

OLAP

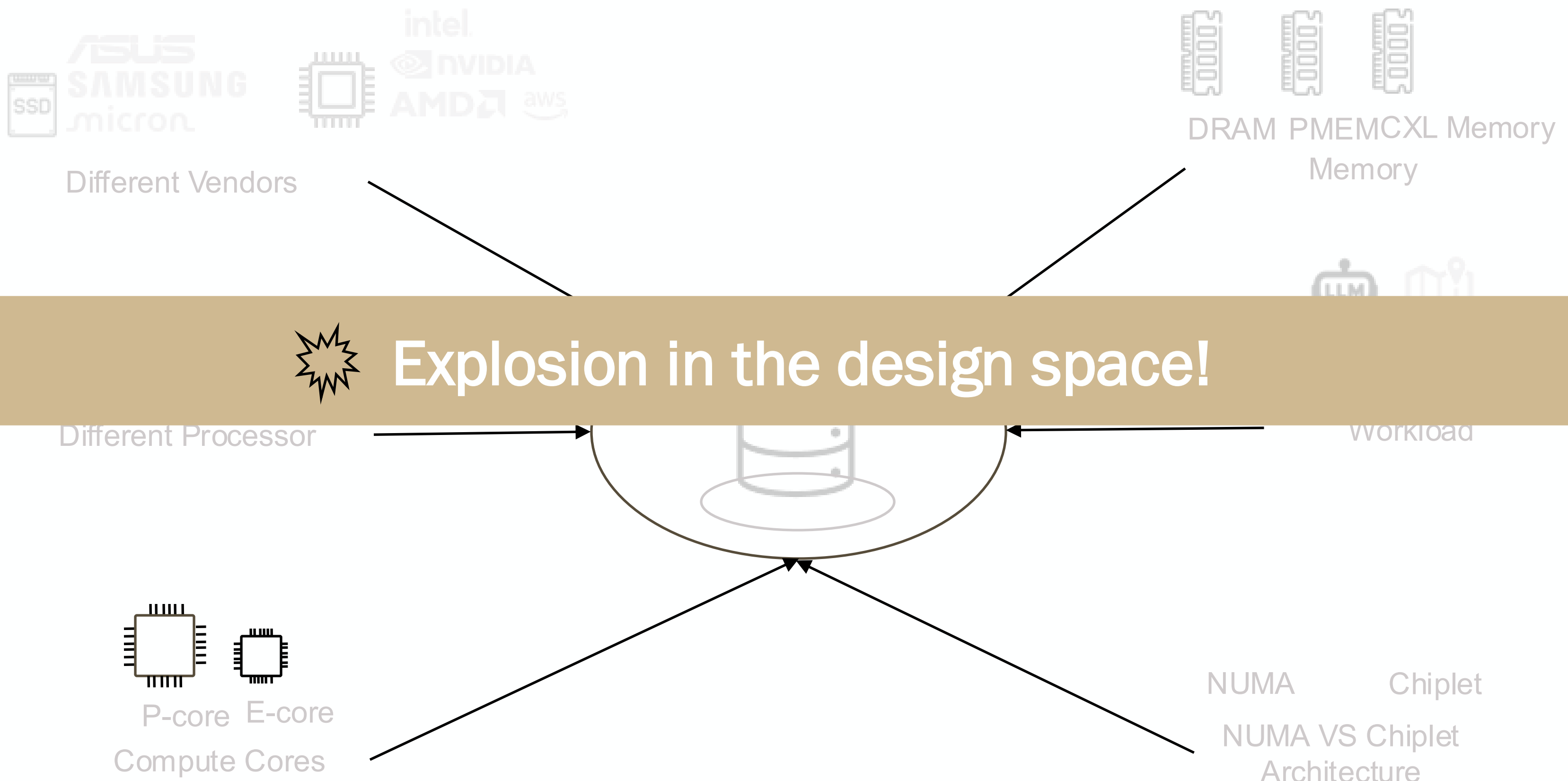
PURDUE UNIVERSITY

Designing Databases for the Modern Era



PURDUE UNIVERSITY

Designing Databases for the Modern Era



Explosion in the design space!

What's Required?

Databases that are **generalizable** across heterogeneous hardware and workload without sacrificing performance.

What's Required?

Databases that are **generalizable** across heterogeneous hardware and workload without sacrificing performance.

Large Language Models (LLM)

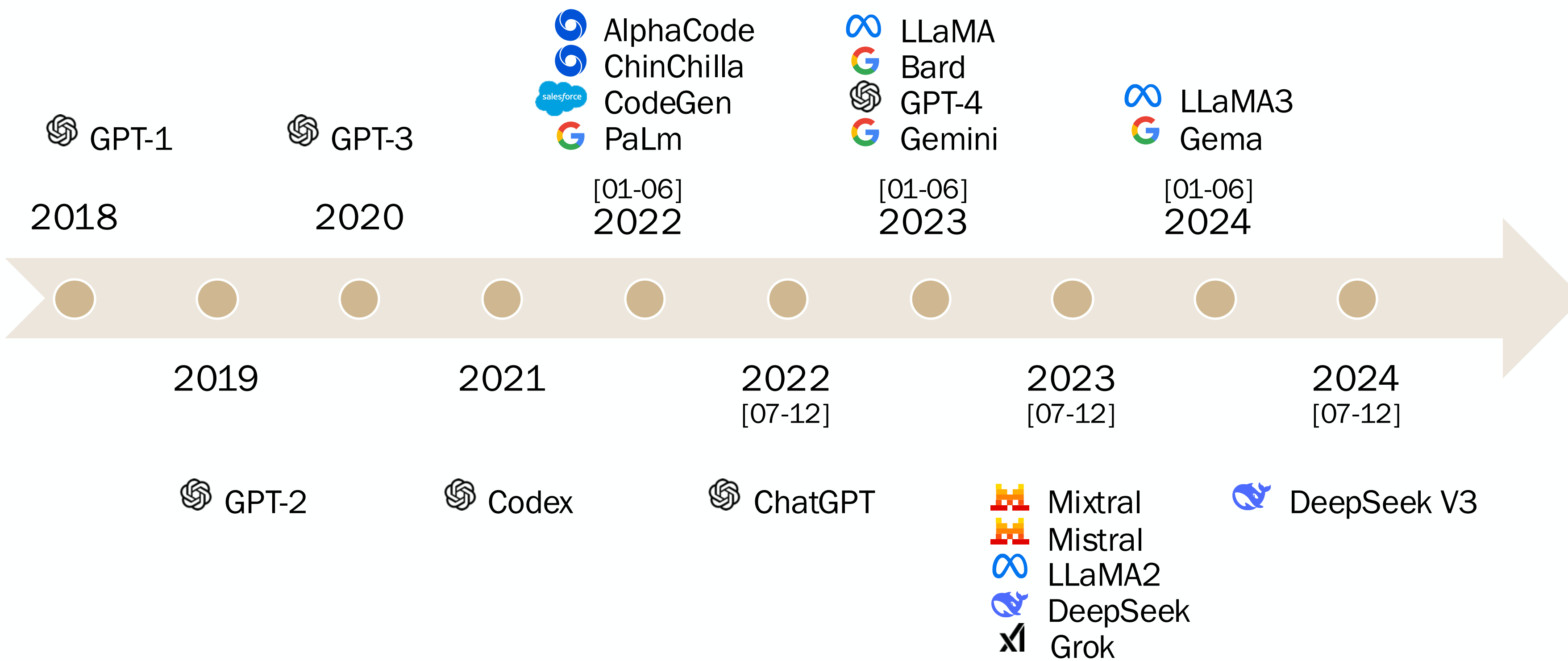
Large Language Models (LLM)

and

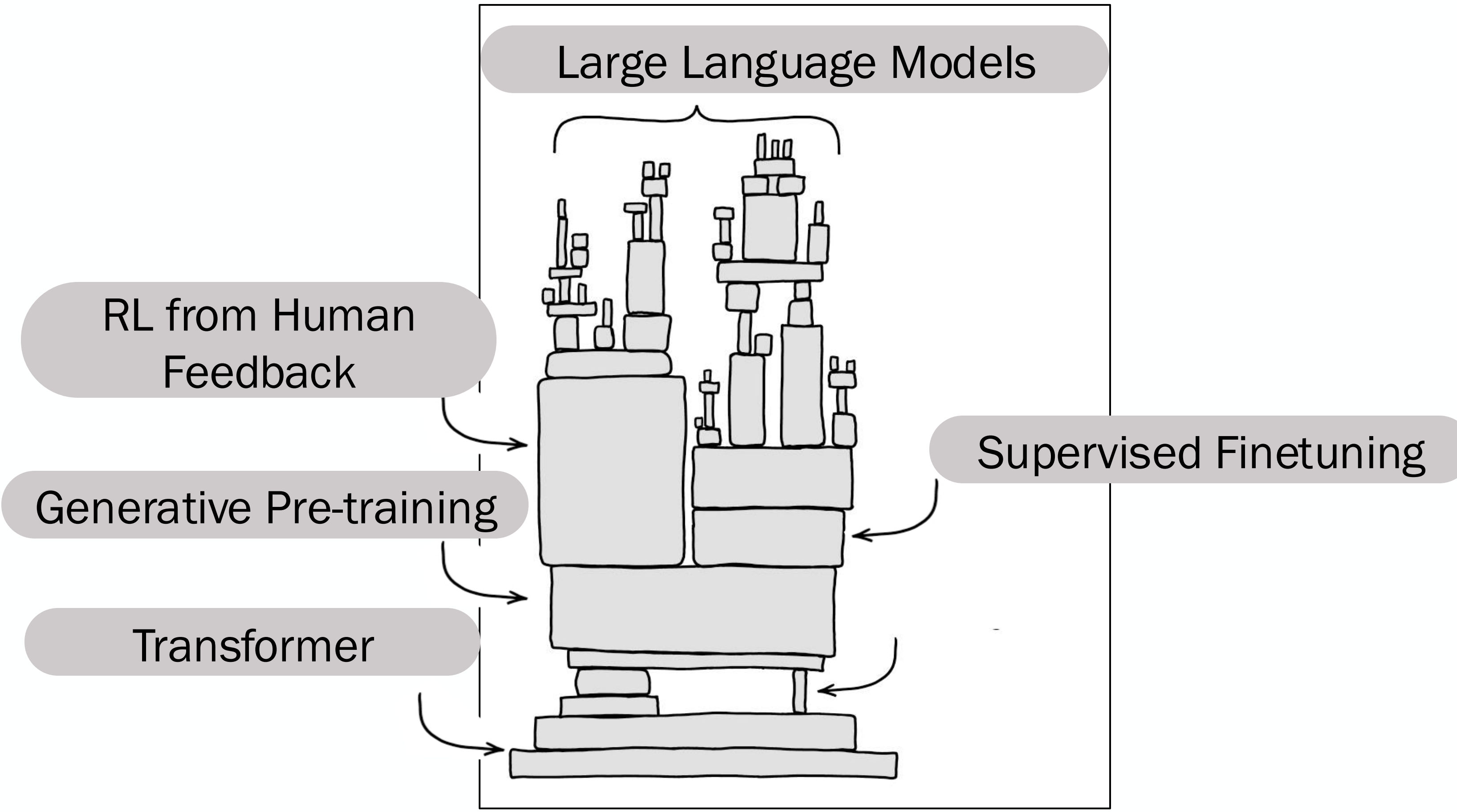
Next Token Prediction (NTP)

PURDUE UNIVERSITY

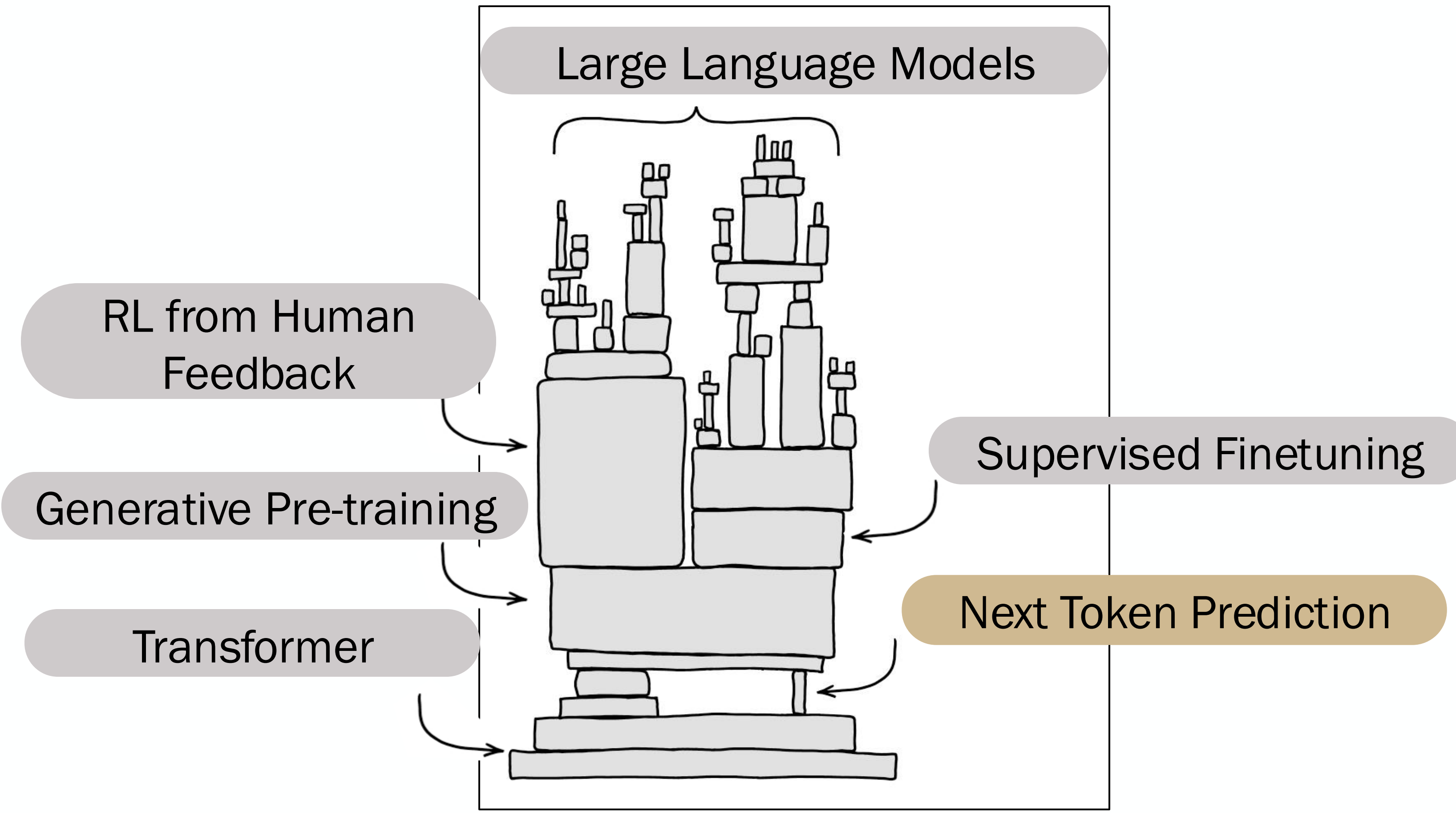
LLMs are everywhere!



The LLM Recipe



First Step Towards Adopting the LLM Recipe



What is Next Token Prediction?

- Next Token Prediction (NTP):
  - Model the probability of the next token in a given sequence based on the past tokens.
  - Token: a discrete unit, i.e., a word, a sub-word, or a character
- For any sequence $\boldsymbol{\tau}$
  - τ_i denotes the i-th token, and
  - $\boldsymbol{\tau}_{<\boldsymbol{i}}$ denotes the $i - 1$ tokens preceding τ_i
- Goal of NTP
  - Estimate the probability distribution of τ_i:

$$\mathbb{P}(\tau_i \,|\boldsymbol{\tau}_{<\boldsymbol{i}})$$

PURDUE UNIVERSITY

Generating a Sequence via NTP

SIGMOD | is | being | 25 | held | Berlin | in | Germany

Vocabulary: Possible set of tokens

SIGMOD

First Token

Generating a Sequence via NTP

SIGMOD | is | being | 25 | held | Berlin | in | Germany

Vocabulary: Possible set of tokens

SIGMOD

First Token

Next Token = ?

Generating a Sequence via NTP

SIGMOD is being 25 held Berlin in Germany

Vocabulary: Possible set of tokens

SIGMOD

First Token

Next Token = ?

PURDUE UNIVERSITY

Generating a Sequence via NTP

SIGMOD | is | being | 25 | held | Berlin | in | Germany

Vocabulary: Possible set of tokens

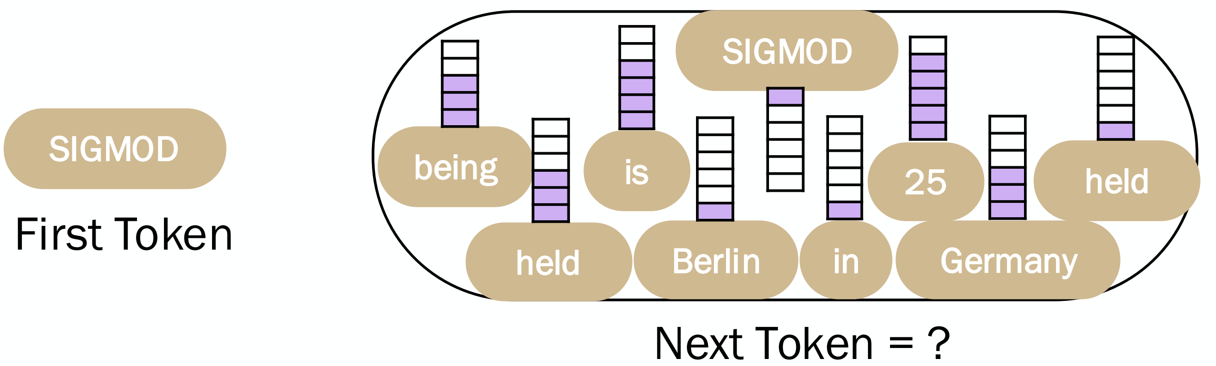
SIGMOD

First Token

Next Token = ?

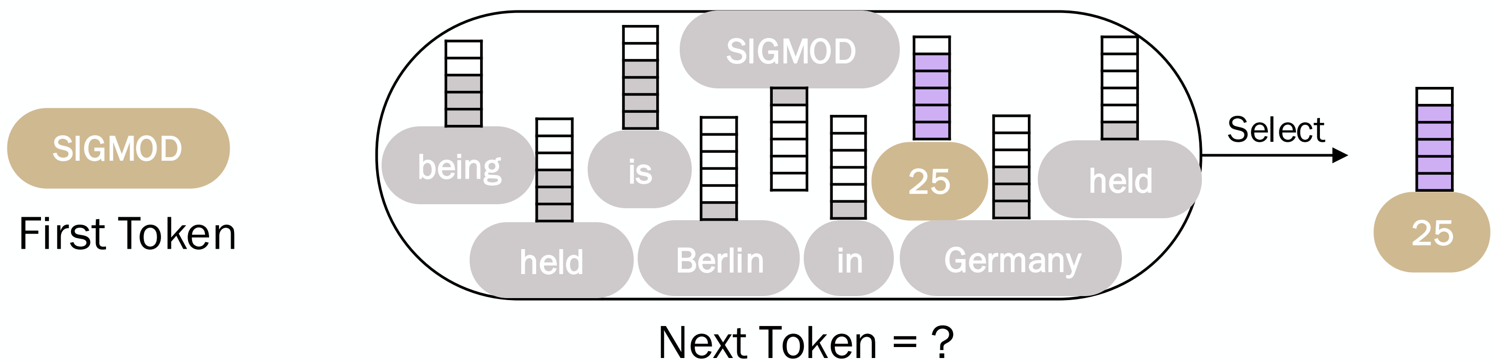
Generating a Sequence via NTP

- Estimate the probability distribution of each possible next tokens.



Generating a Sequence via NTP

- Select the token with the highest probability distribution as the next token.



Generating a Sequence via NTP

SIGMOD | is | being | 25 | held | Berlin | in | Germany

Vocabulary: Possible set of tokens

SIGMOD
First Token

25
Next Token

PURDUE UNIVERSITY

Generating a Sequence via NTP

SIGMOD is being 25 held Berlin in Germany

Vocabulary: Possible set of tokens

SIGMOD 25 is being held in Berlin Germany

Full Sentence

Next Token Prediction (NTP)

and

Database Systems

PURDUE
UNIVERSITY

Translating NTP into Database Systems

- **Join Order Selection** ($A \bowtie B \bowtie C \bowtie D$)
  - Tokens: The set of tables: A, B, C, D

Predicting the next table to join

A C D B

- **Transaction scheduling** (T1, T2, T3, T4)
  - Tokens: The set of transactions: T1, T2, T3, T4

Predicting the next transaction to schedule

T4 T3 T2 T1

Hindrances of Translating NTP into DBMS

1. The mismatch in objectives between the domain of NLP and Databases

NLP tasks	Database optimization tasks
Generative LLMs generate coherent sequences.	Goal-oriented
In Generative Pre-training phase, the goal is to compress a significant amount of world knowledge into the LLM by training on a diverse internet-scale corpus.	Improve query performance, scalability, and resource utilization.

Hindrances of Translating NTP into DBMS

2. The mismatch in the notion of Tokens in NLP and Databases

NLP tasks	Database optimization tasks
The notion of token is fixed for a particular tokenizer.	The notion of token is diverse and irregular. • In JOS, tables are tokens. • In scheduling, transactions are tokens.

NLP Tokens VS Database Tokens

SIGMOD | 25 | is | being | held | in | Berlin | Germany

- Syntactic regularity:
  - The verb follows the subject.

☒ SIGMOD | 25 | in | Germany | Berlin | held | is | being

- Contextual meaning:
  - The mention of Germany implicitly excludes other locations, e.g., USA.

☒ SIGMOD | 25 | is | being | held | in | Chicago | USA

NLP Tokens VS Database Tokens

A C D B

Join Order Selection ($A \bowtie B \bowtie C \bowtie D$)

- Lack of syntactic regularity:
  - Two different database instances can have tables with same name but with different attributes.
- Lack of contextual meaning:
  - A C D B does not eliminate the possibility of D A C B

PURDUE UNIVERSITY

Contribution of this Paper

Database Systems Meet Next Token Prediction

Building Blocks

Database Systems

Next Token Prediction

DB-Tokens

Decision Transformer

1. **Decision Transformer:** Sequence Modeling
2. **DB-Tokens:** Generalization

Building Block 1: Decision Transformer

- Goal-directed RL:
  - Treats RL as a supervised sequence modeling problem
  - Predict the next action (token) by conditioning on a desired reward, e.g., query throughput, latency, scalability etc.

$$\mathbb{P}(a_i \mid s_i, \hat{R}_i, a_{<i}, s_{<i}, \hat{R}_{<i})$$

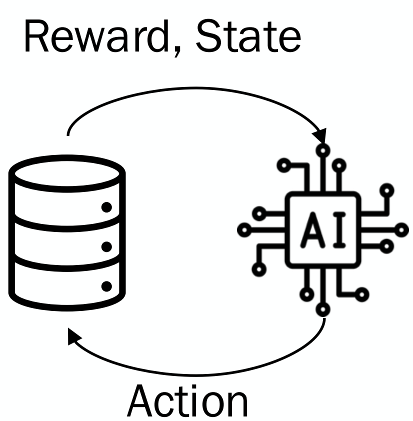
- Each trajectory (policy) is represented as a sequence of (return-to-go, state, action) tuples.

$$\tau = \hat{R}_1 \; s_1 \; a_1 \quad \hat{R}_2 \; s_2 \; a_2 \quad \hat{R}_T \; s_T \; a_T$$

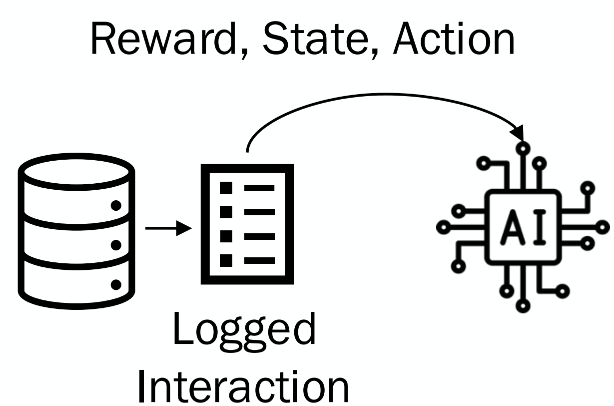
Reward-to-go token
Future cumulative reward expected from a given timestep onward.

Building Block 1: Decision Transformer

- Scalable across large datasets:
  - The underlying model is a Transformer architecture.
  - It follows the Offline Reinforcement Learning Paradigm
    - The model is trained on an offline dataset.
    - The model does not interact with the environment.



Online RL



Offline RL

Building Block 2: DB-Tokens

- Hardware profiles generated from hardware Performance Monitoring Units (PMU, for short)
  - Computationally inexpensive to retrieve from the hardware registers
  - **Generalizable** across different hardware and workload applications
  - Can provide accurate hardware context that the DBMS is running on
  - Can mimic the data distribution and query workload

PURDUE UNIVERSITY

The Framework

Database Systems

Next Token Prediction

DB Tokens

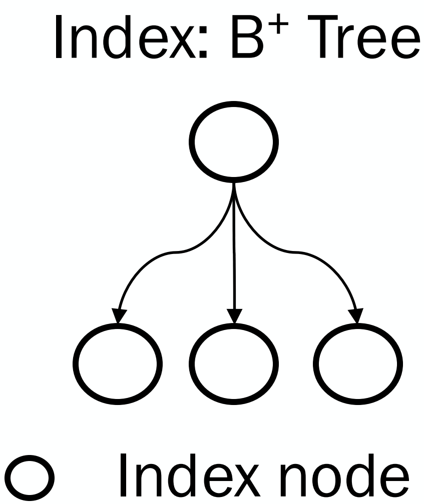
Decision Transformer

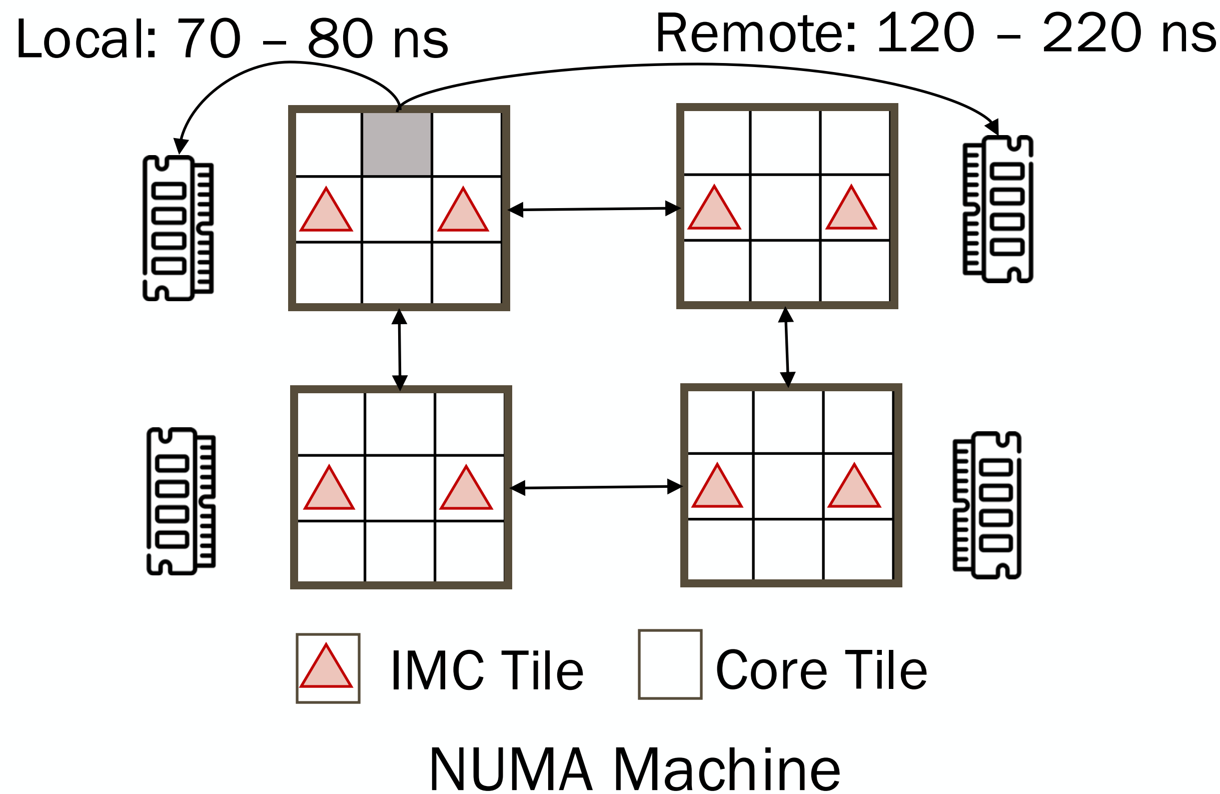
Probe and Learn

(Po Le)

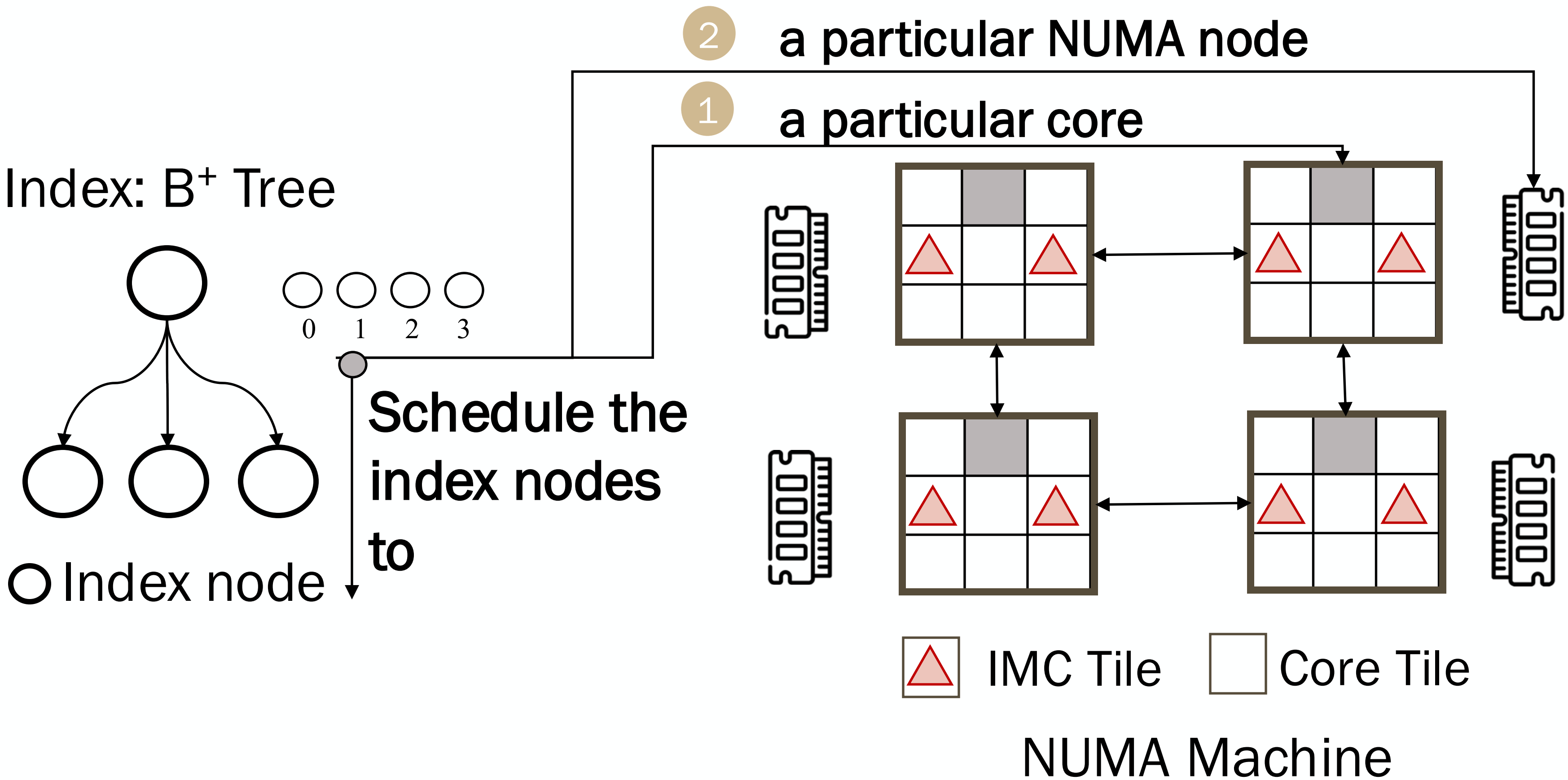
PURDUE UNIVERSITY

Case Study: Index Scheduling



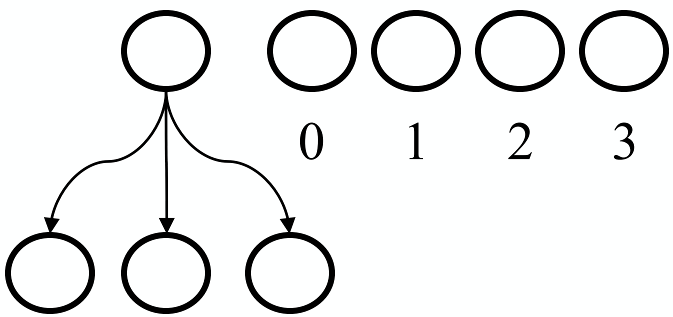


Case Study: Index Scheduling

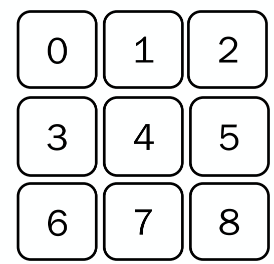


Index Scheduling: NTP Formulation

- Predict the next core to place the i-th index node
  - A policy in index scheduling = Sequence of core IDs

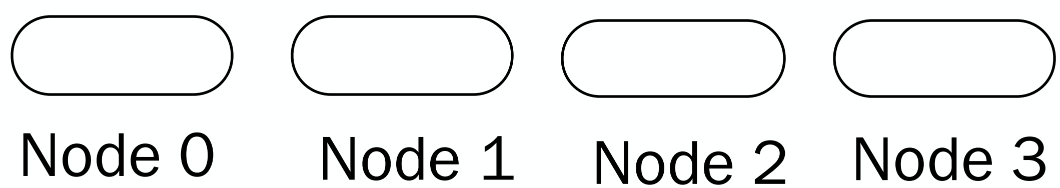


Index: B+ Tree



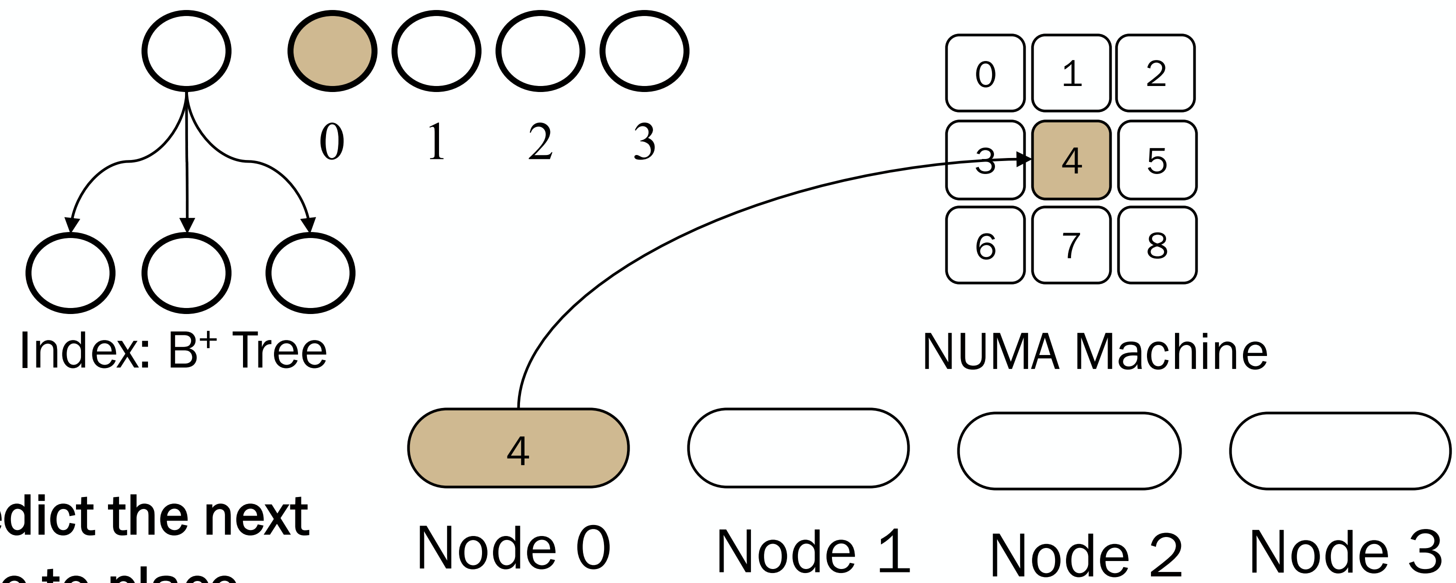
NUMA Machine

Predict the next core to place



Index Scheduling: NTP Formulation

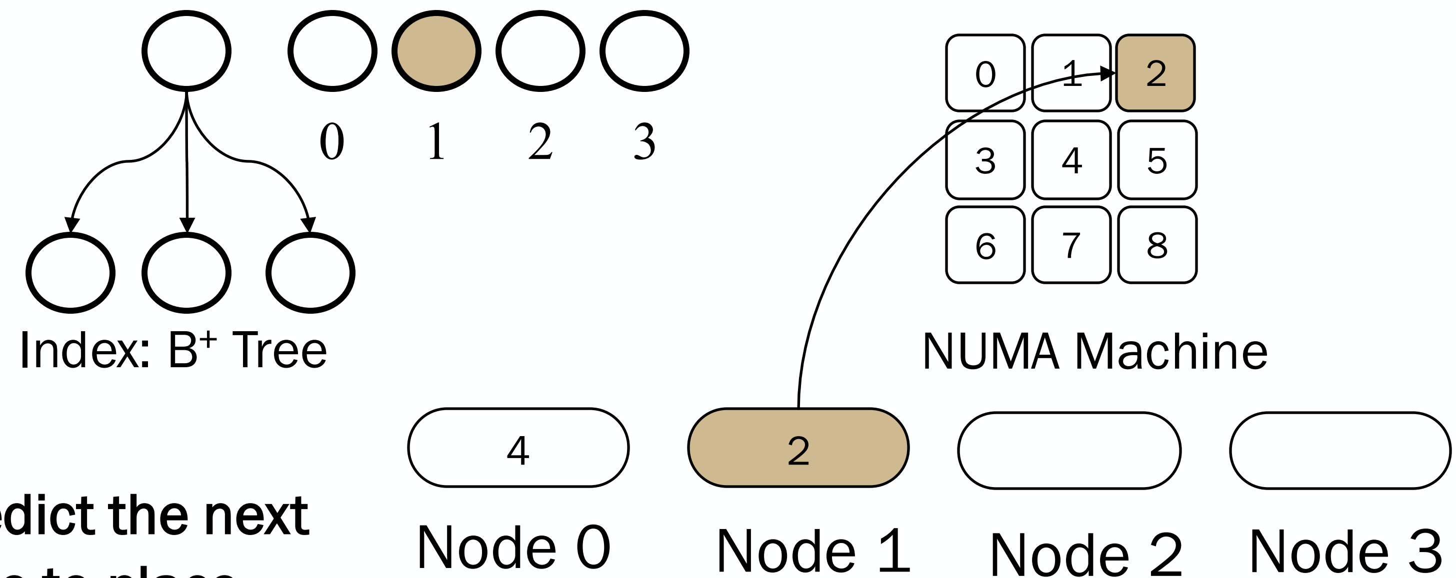
- Predict the next core to place the i-th index node
  - A policy in index scheduling = Sequence of core IDs



Predict the next core to place

Index Scheduling: NTP Formulation

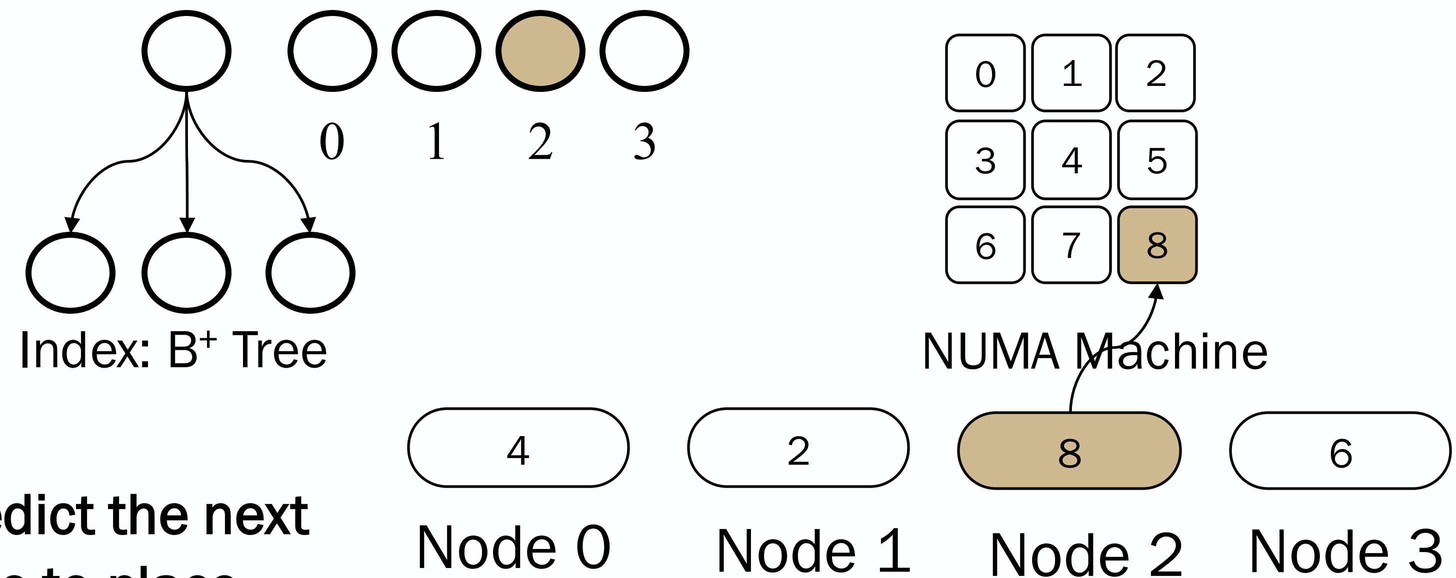
- Predict the next core to place the i-th index node
  - A policy in index scheduling = Sequence of core IDs



Predict the next core to place

Index Scheduling: NTP Formulation

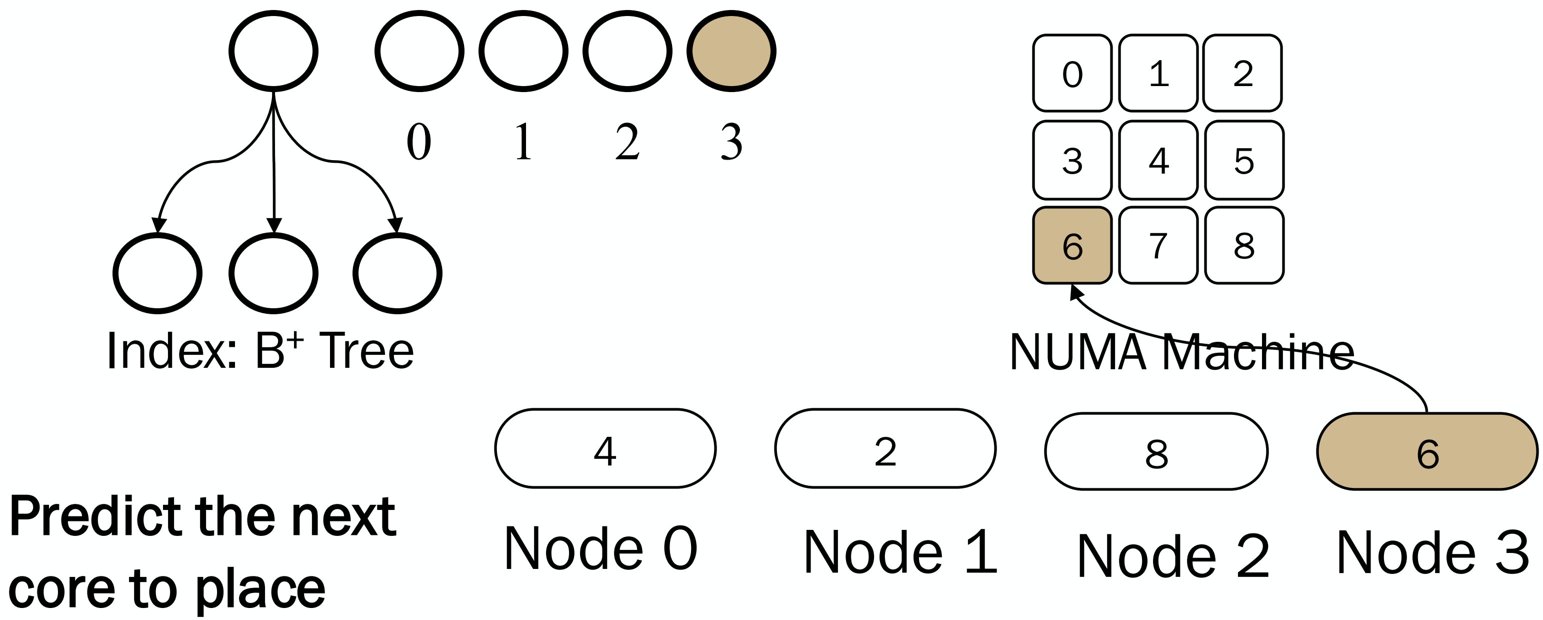
- Predict the next core to place the i-th index node
  - A policy in index scheduling = Sequence of core IDs



Predict the next core to place

Index Scheduling: NTP Formulation

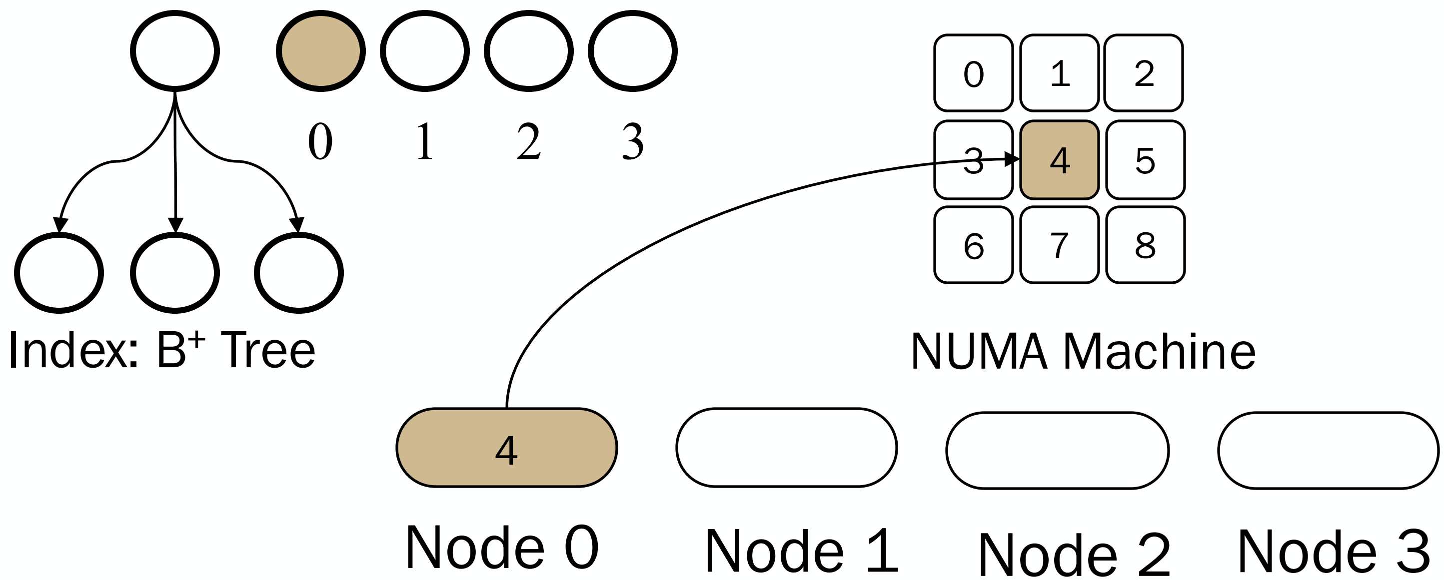
- Predict the next core to place the i-th index node
  - A policy in index scheduling = Sequence of core IDs



PURDUE UNIVERSITY

Index Scheduling: DB-Tokens

- Predict the next core to place the i-th index node
  - A policy in index scheduling = Sequence of core IDs



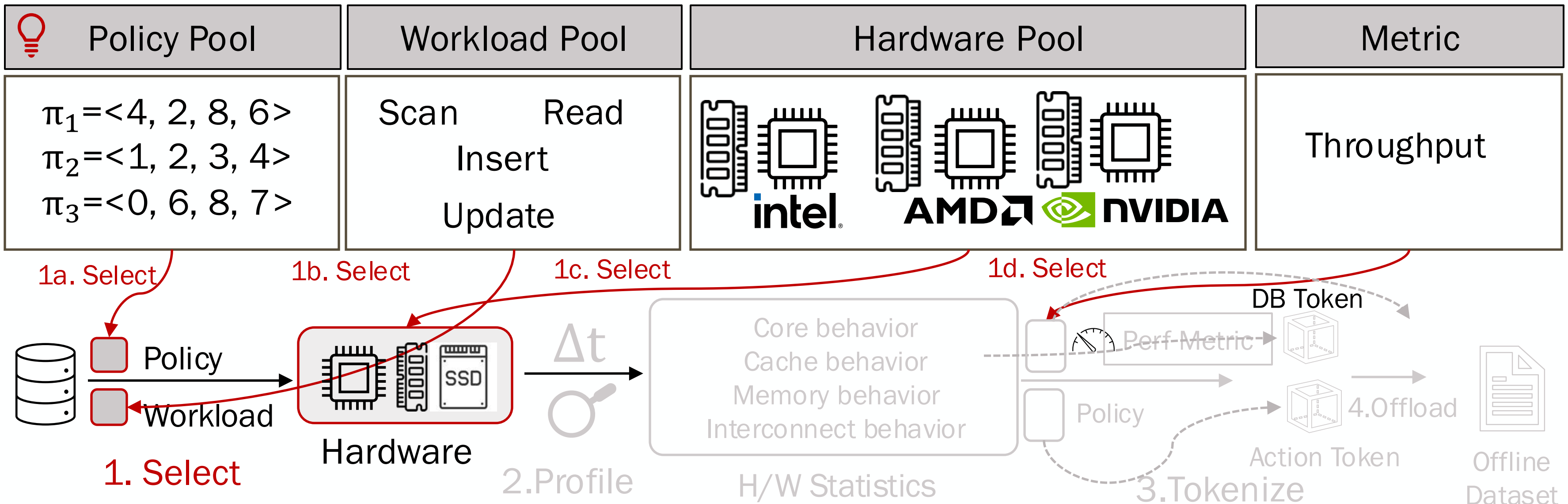
DB-Tokens: The L1, L2, LLC cache misses, branch misses, TLB misses, local and remote memory accesses of <Core 4>, query throughput of <Core 4>.

PURDUE UNIVERSITY

Probe Phase of PoLe Framework
Index Scheduling

PURDUE
UNIVERSITY

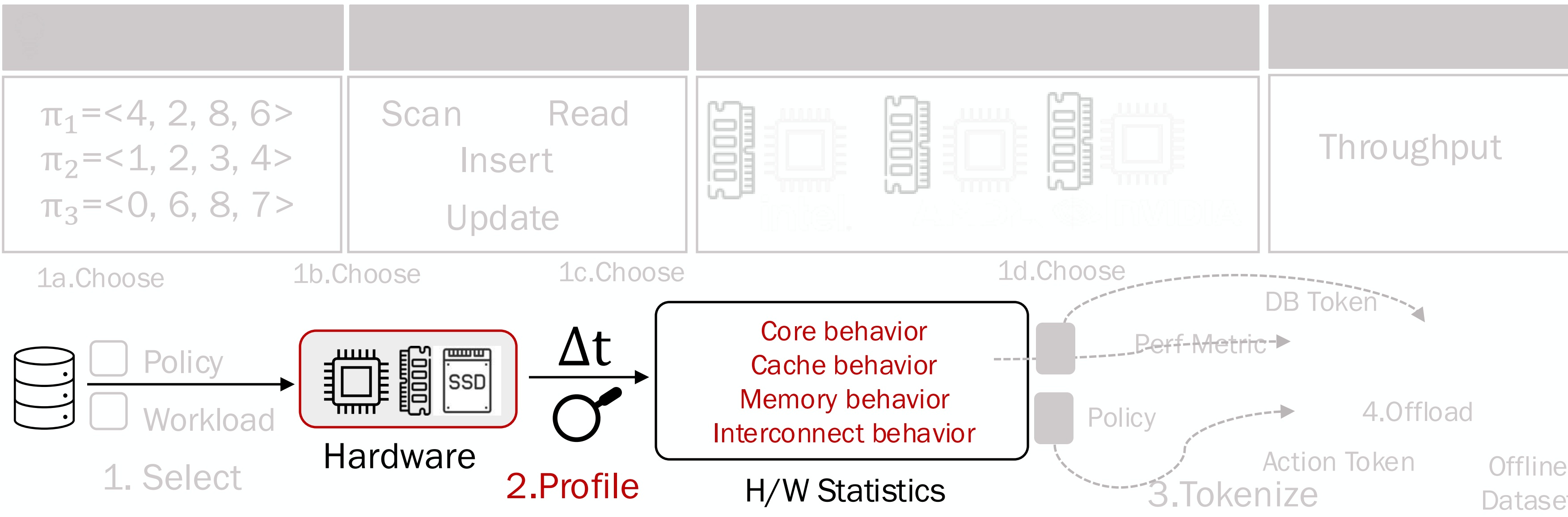
Probe Phase: Select (Step 1)



1. **Select:** Execute different policies across various hardware configurations and workloads, drawn from a diverse set of policy, hardware, and workload pools.

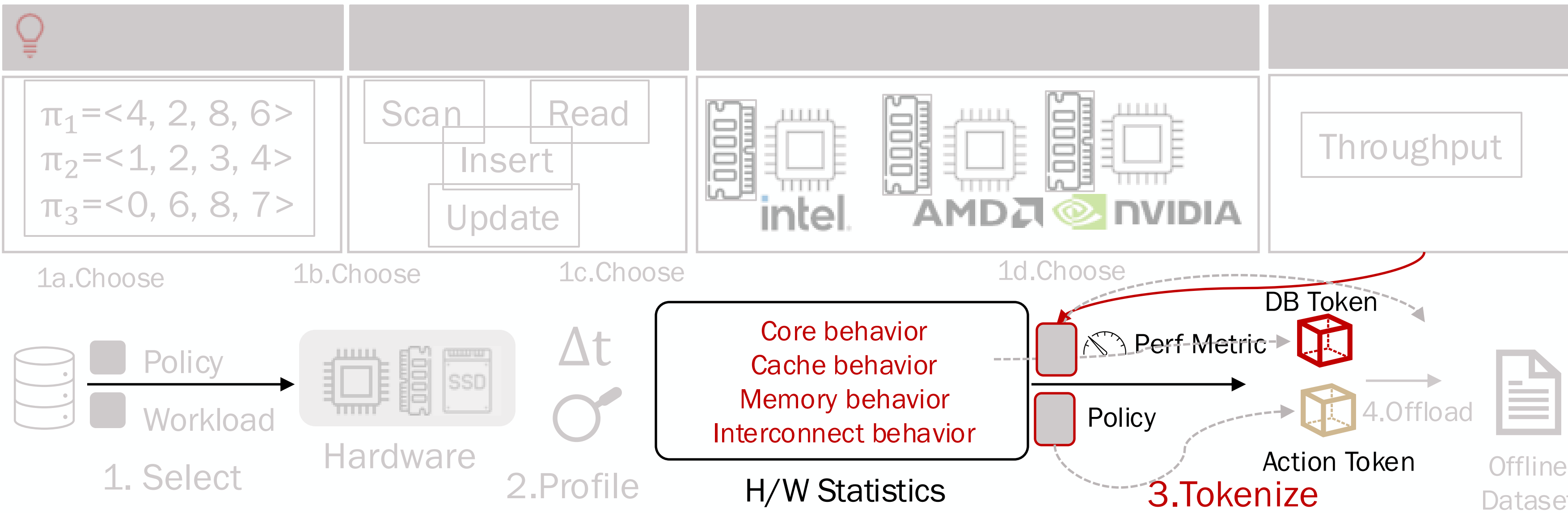
PURDUE UNIVERSITY

Probe Phase: Profile (Step 2)



2. **Profile:** During query execution, periodically profile the hardware to capture the behavior of crucial hardware components.

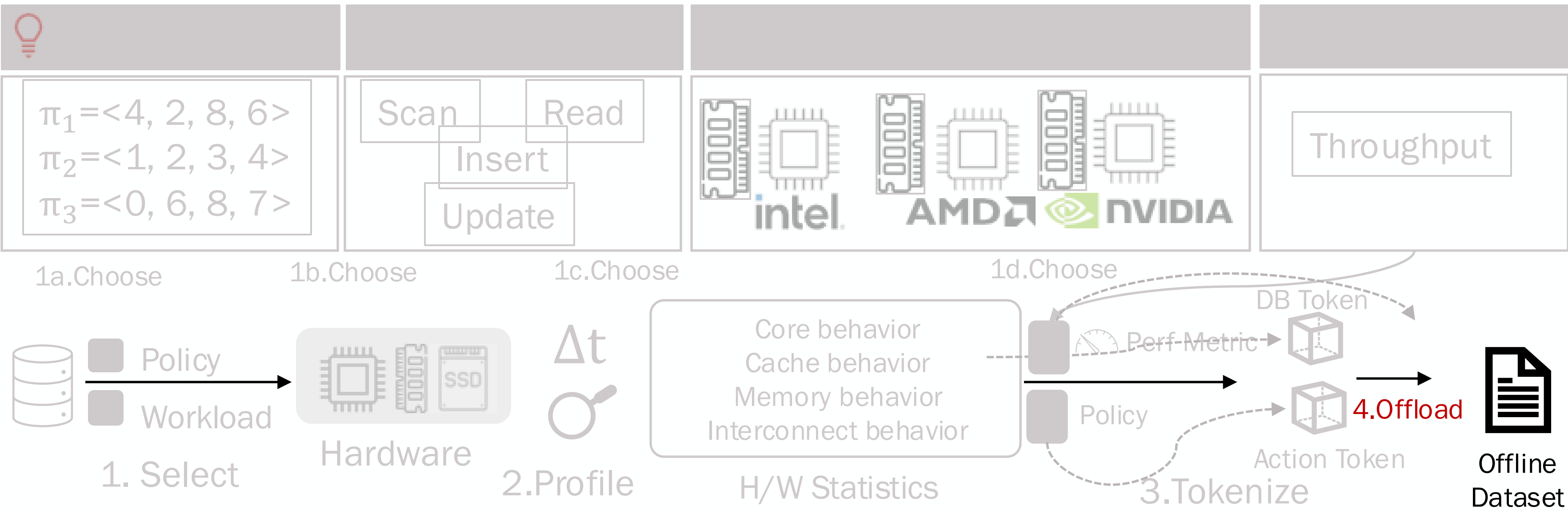
Probe Phase: Tokenize (Step 3)



3. **Tokenize:** Tokenize the hardwire profiles, alongside the desired performance metric and the current policy.

PURDUE UNIVERSITY

Probe Phase: Offload (Step 4)

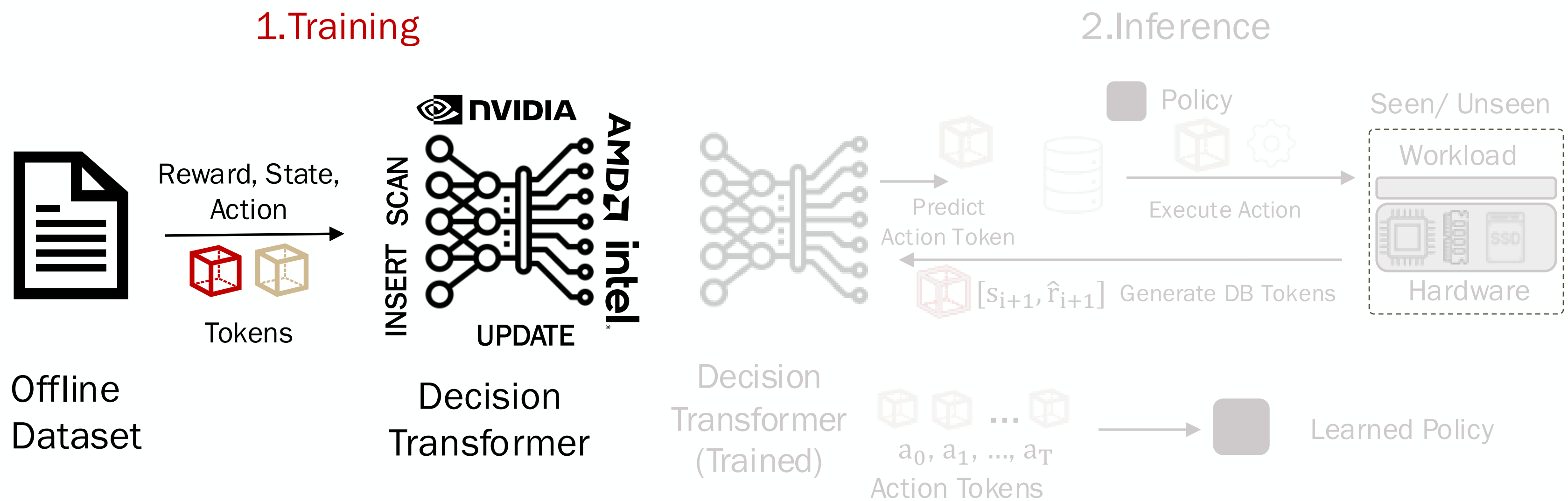


4. **Offload:** Offload the action tokens along with the associated DB-tokens to an offline dataset. This is provided to the Decision Transformer during the learning phase.

PURDUE UNIVERSITY

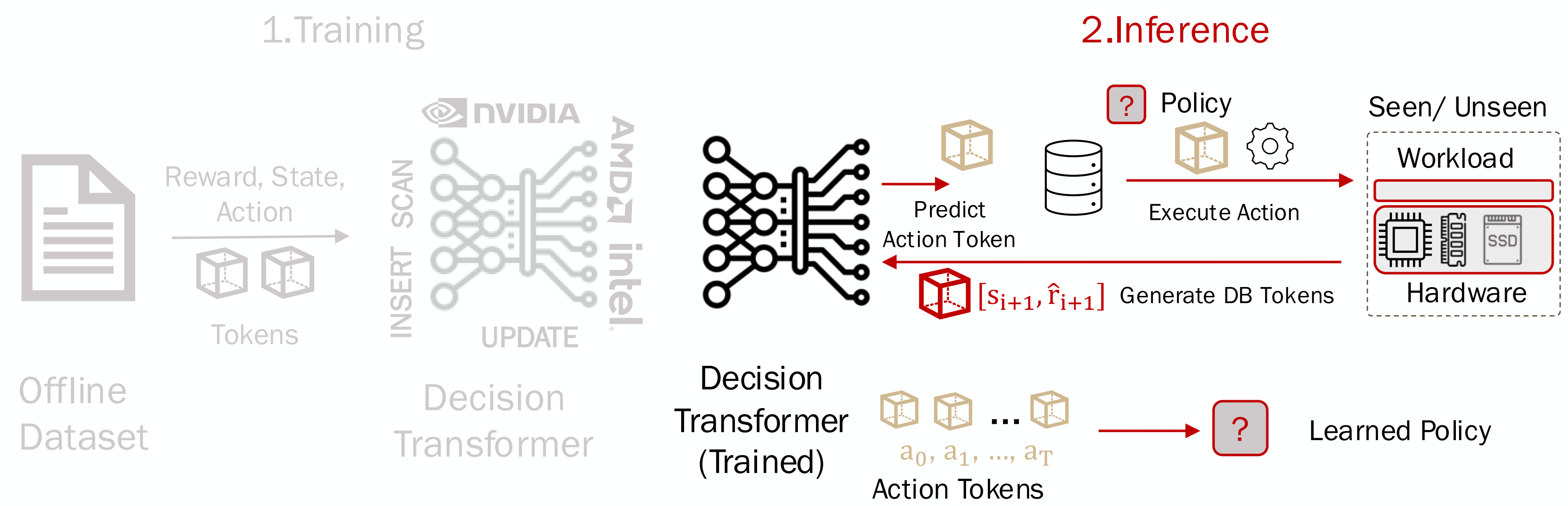
Learn Phase of PoLe Framework
Index Scheduling

Learn Phase: Training (Step 1)



1. **Training:** Train a Decision Transformer (DT) on the collected offline dataset in a supervised manner.

Learn Phase: Inference (Step 2)



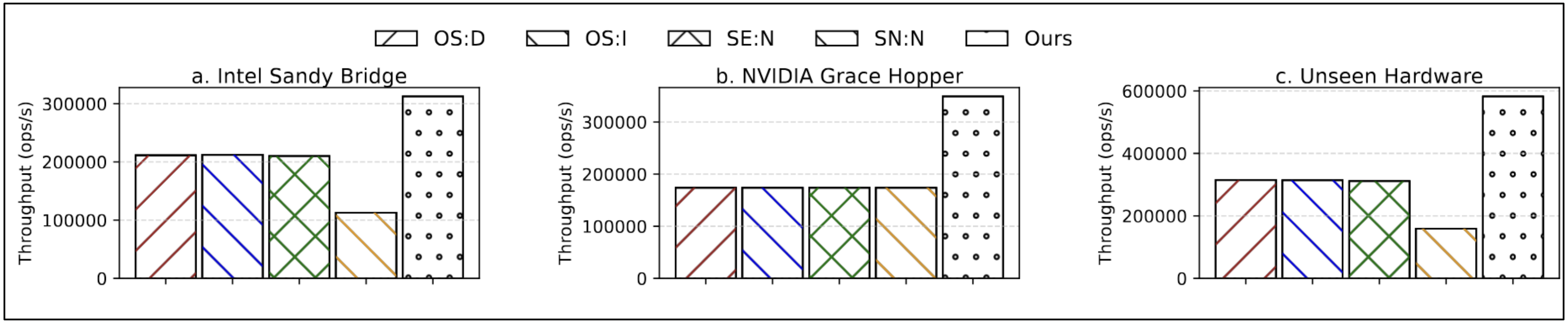
2. **Inference:** Infer a new policy via auto-regression using the trained Decision Transformer.

Experiment Settings

- Index: A main-memory B^+-Tree
- Workload: YCSB-A workload
- Baselines
  - OS Scheduling Policies
    - Default, Local, Interleaved
    - OS handles core scheduling.
  - Heuristics:
    - Shared Everything NUMA
      - Nearby index nodes are placed on the same NUMA node.
      - OS handles scheduling.
    - Shared Nothing
      - Nearby index nodes are placed on the same NUMA node.
      - Core Scheduling follows the data placement strategy.

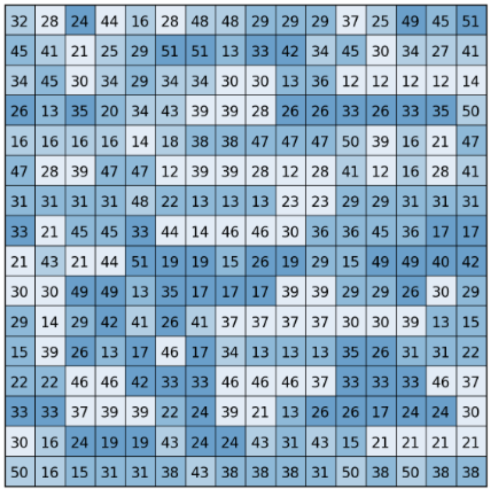
Preliminary Results

- On seen hardware (Intel Sandy Bridge and NVIDIA Grace Hopper)
  - PoLe outperforms the baselines by up to 2.78×
- On unseen hardware (Intel Skylake X)
  - PoLe outperforms the baselines by up to 3×

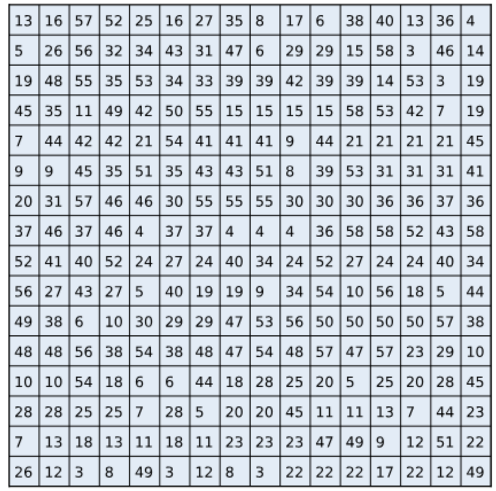


PoLe's Learned Scheduling Policies

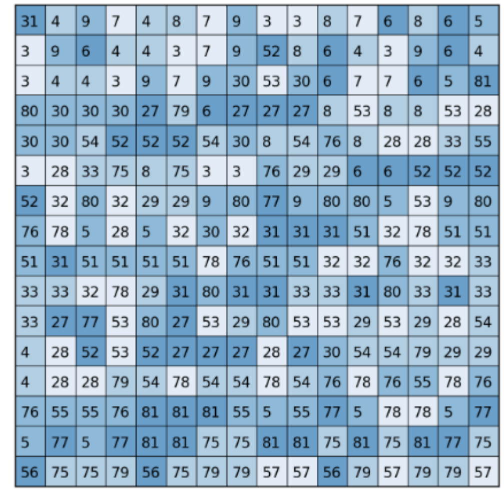
Intel Sandy Bridge (π_l)



NVIDIA Grace Hopper (π_l)



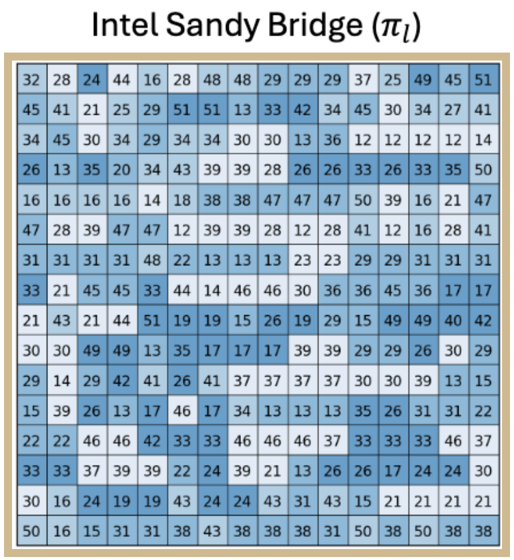
Intel Skylake X (π_l)

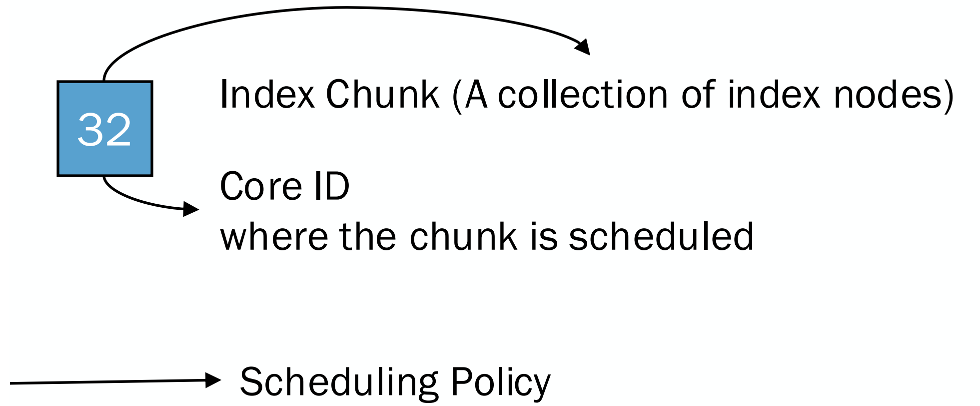


Cells sharing the same color indicate that the associated index chunks are scheduled on the same NUMA server.

PURDUE UNIVERSITY®

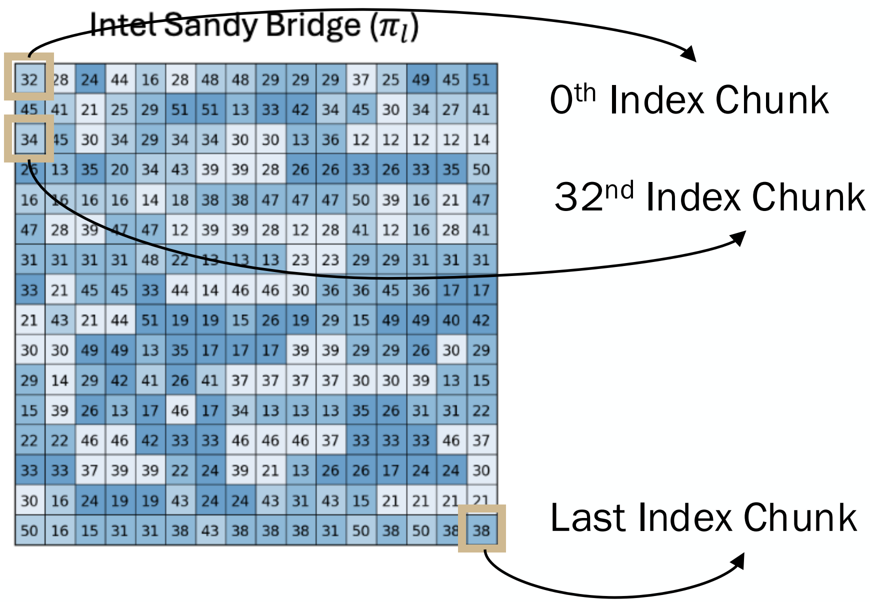
PoLe's Learned Scheduling Policies





Cells sharing the same color indicate that the associated index chunks are scheduled on the same NUMA server.

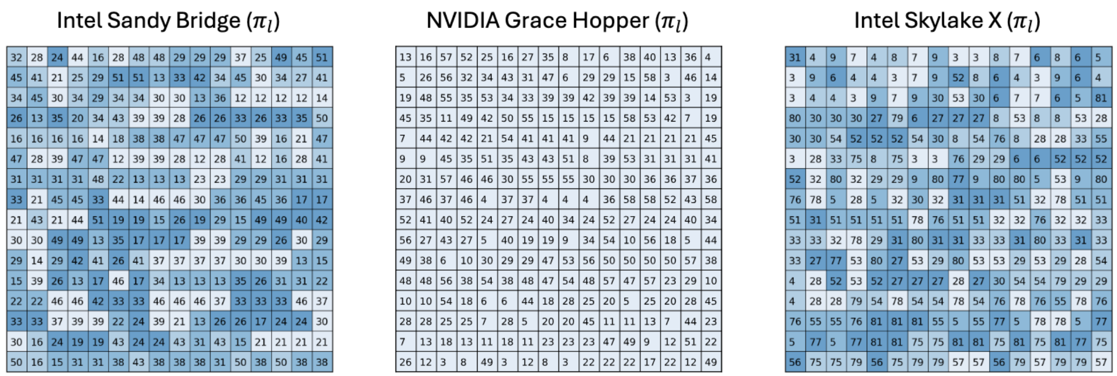
PoLe's Learned Scheduling Policies



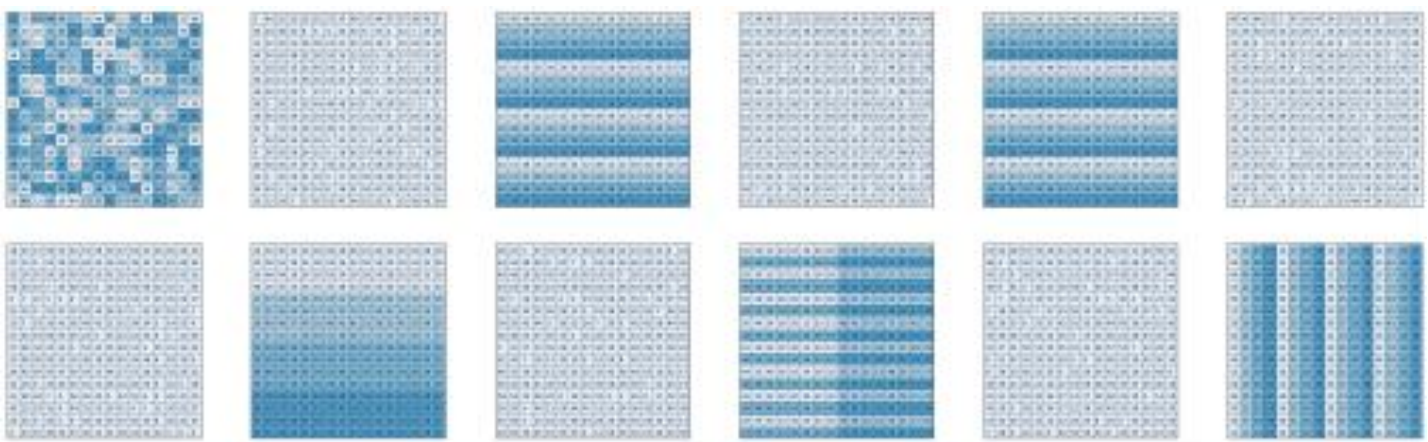
Cells sharing the same color indicate that the associated index chunks are scheduled on the same NUMA server.

PoLe's Scheduling Policies

PoLe's Learned Scheduling Policies

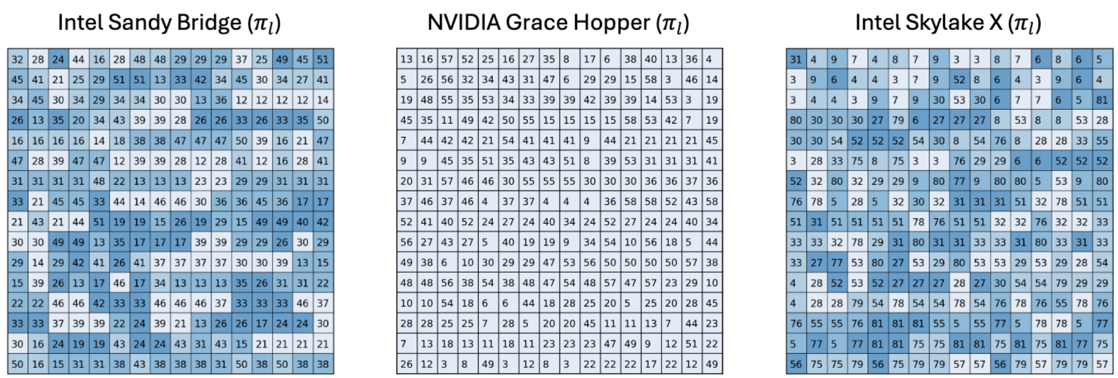


Scheduling Policies in the training set

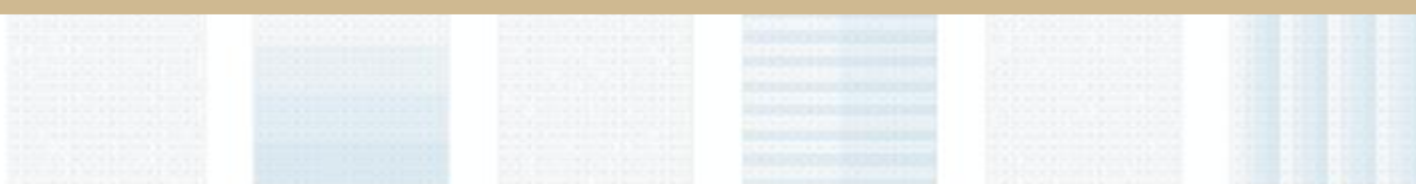


PoLe's Learned Scheduling Policies

PoLe's Learned Scheduling Policies



1. Learned Policies are different from the observed policies in the training set.
2. Different hardware ➡ Different learned policies.



Conclusion

- PoLe brings **Next Token Prediction** into the world of database optimization.
- It leverages **offline RL and DB-Tokens** to learn generalizable scheduling strategies.
- What's next?
  - Categorize the optimization tasks that can benefit from the Next Token Prediction (NTP) paradigm
  - Assess to what extent the PoLe framework can provide consistent performance and adaptivity guarantees

Thank You!

Questions?

Read our paper!

Get in touch!