Lightweight AI for Efficient Resource Management in Heterogeneous-Core Architectures

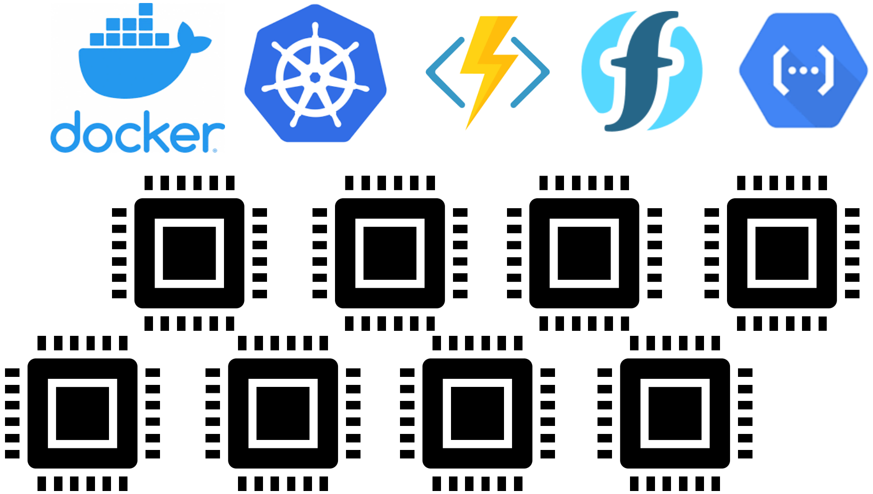
Joshua Kim[1], Chaojie Zhang[2], Íñigo Goiri[2], Christopher J. Rossbach[1,2], Jovan Stojkovic[1]

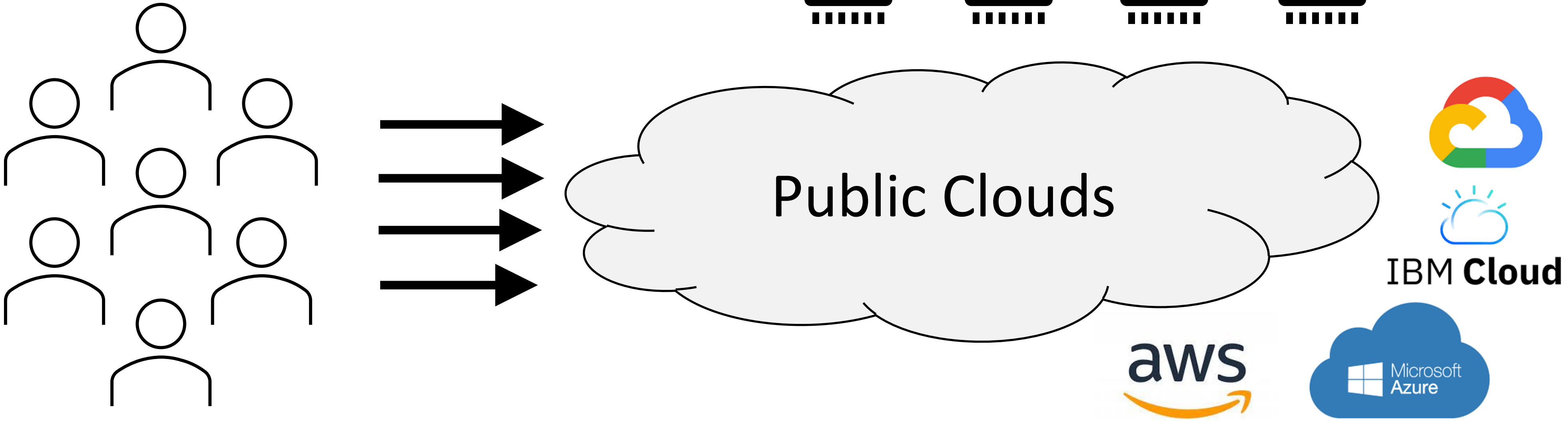
[1]The University of Texas at Austin, [2]Microsoft

The Growth of Cloud Computing

New Computing Paradigms:

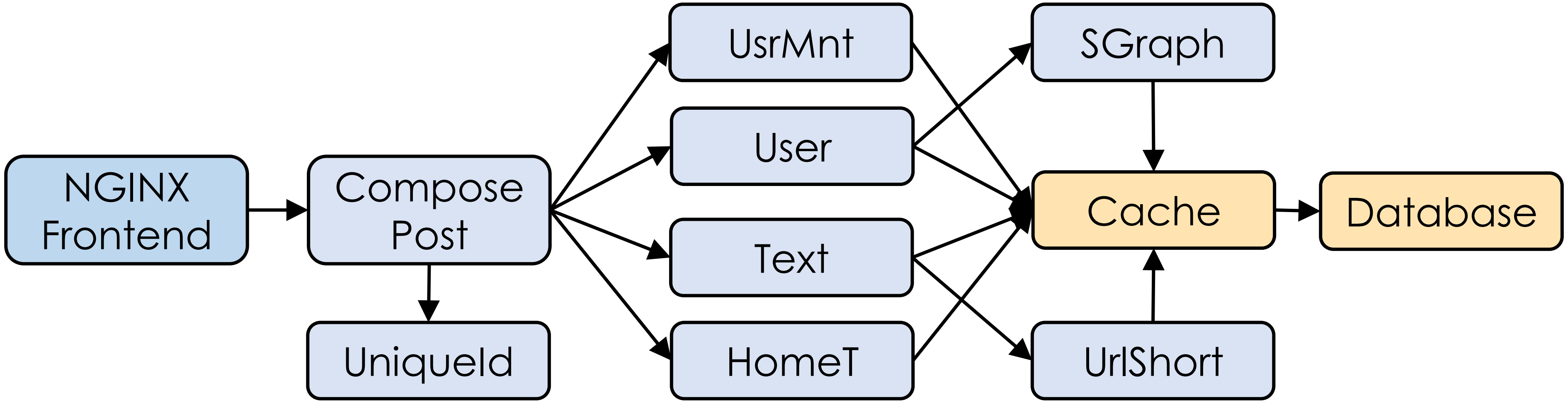
- ***Microservices***
- ***Serverless or Function-as-a-Service (FaaS)***





Microservices

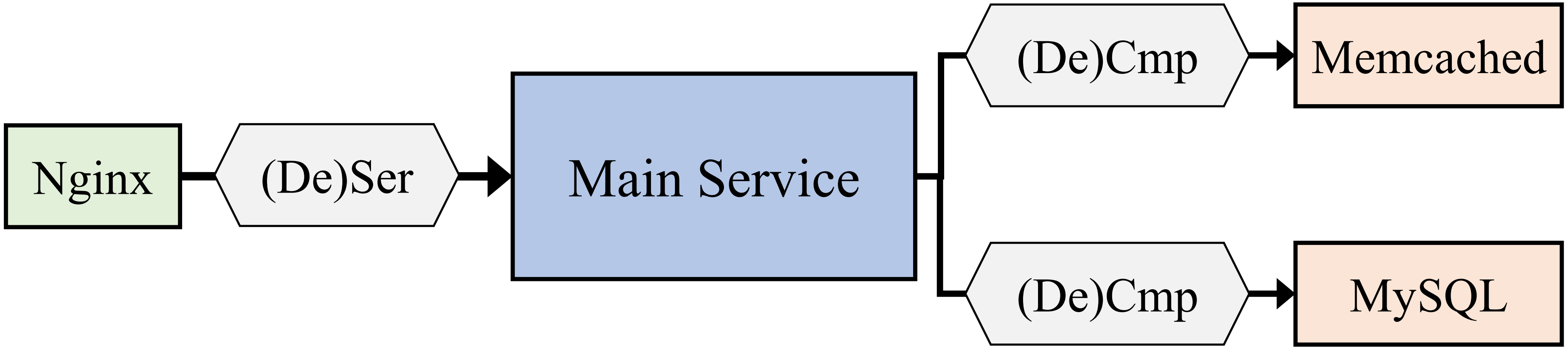
- Large monolithic applications decomposed into many small interdependent services
  - Each service implements separate functionality



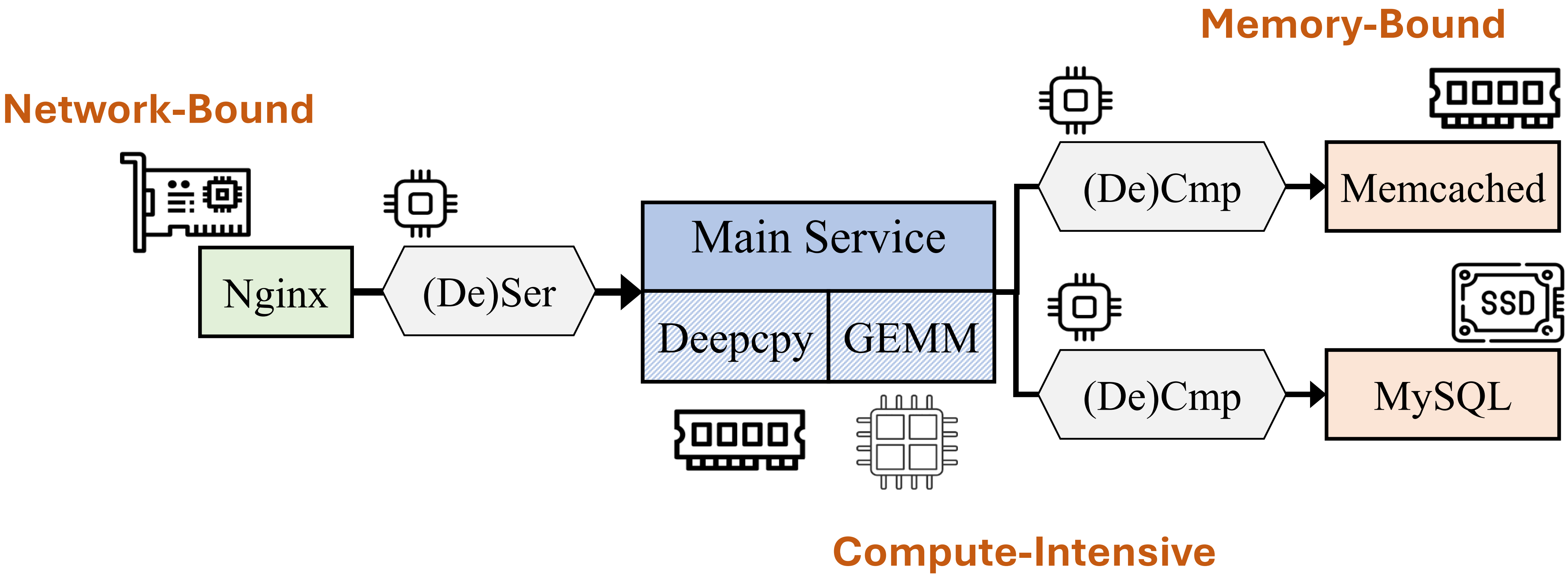
Benefits of Microservices

- Scalability
- Design simplicity
- HW management

Characterizing DC Workloads

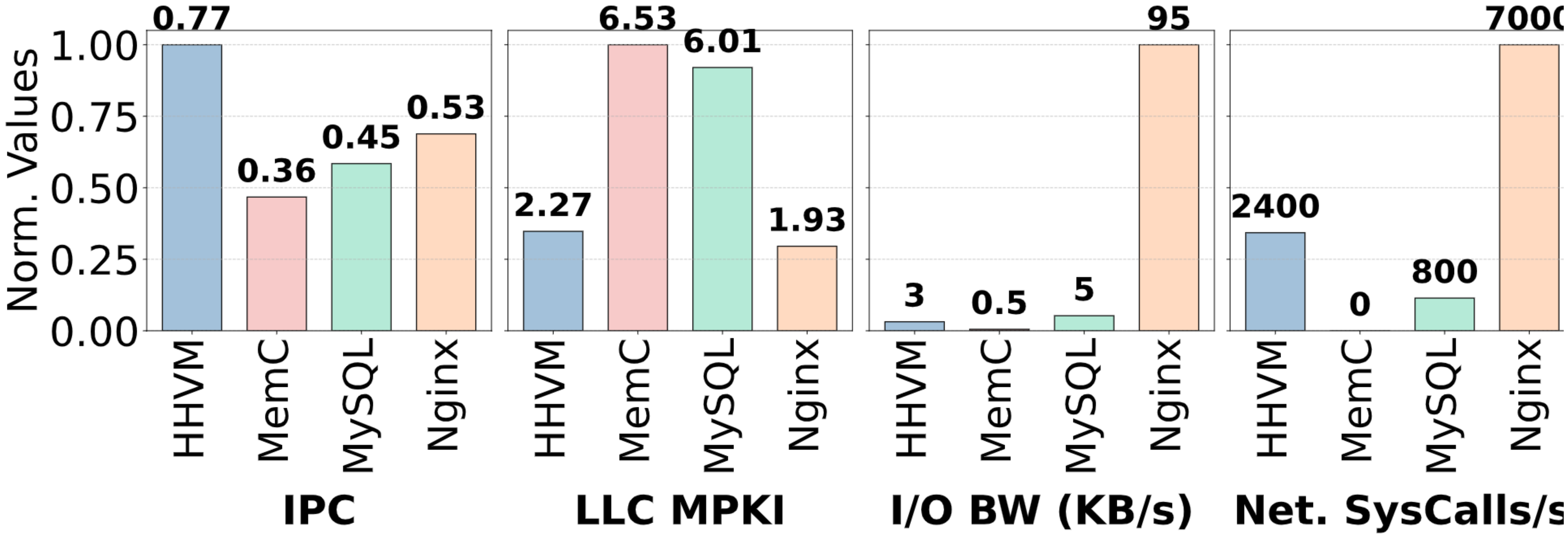


Characterizing DC Workloads



Heterogeneity Across Microservices

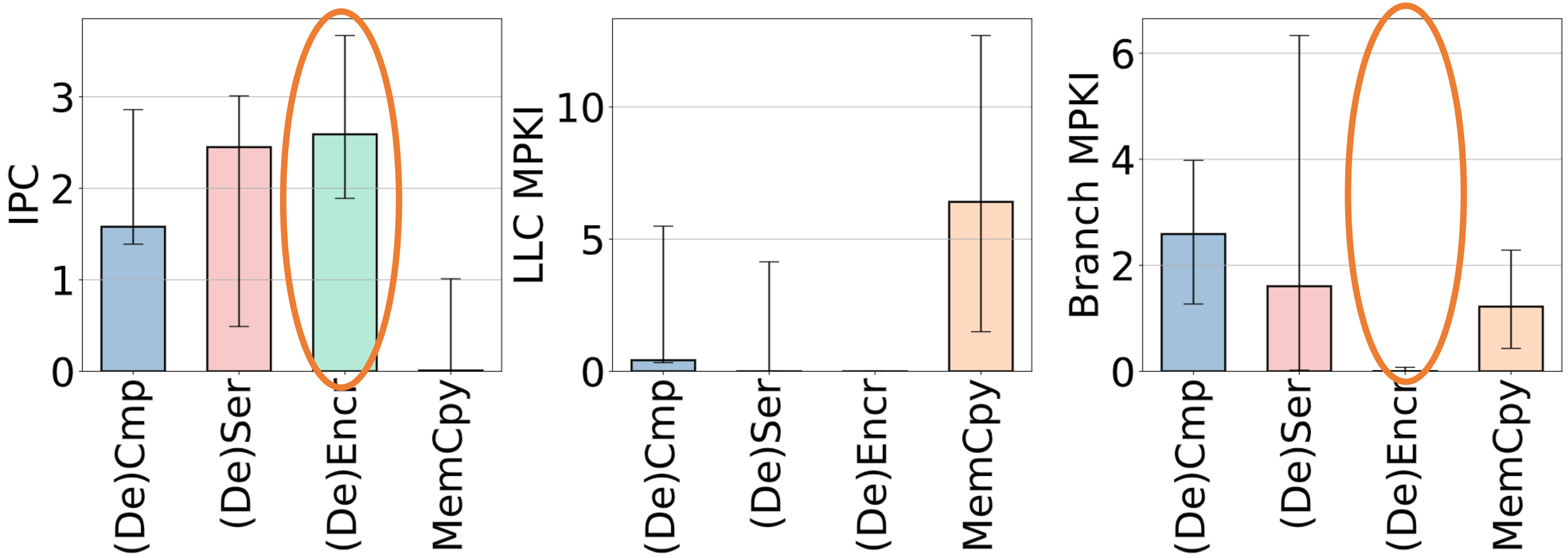
1. Microservice Software Architecture



Normalized metrics for different microservices that appear in Mediawiki

Heterogeneity Across Datacenter Tax

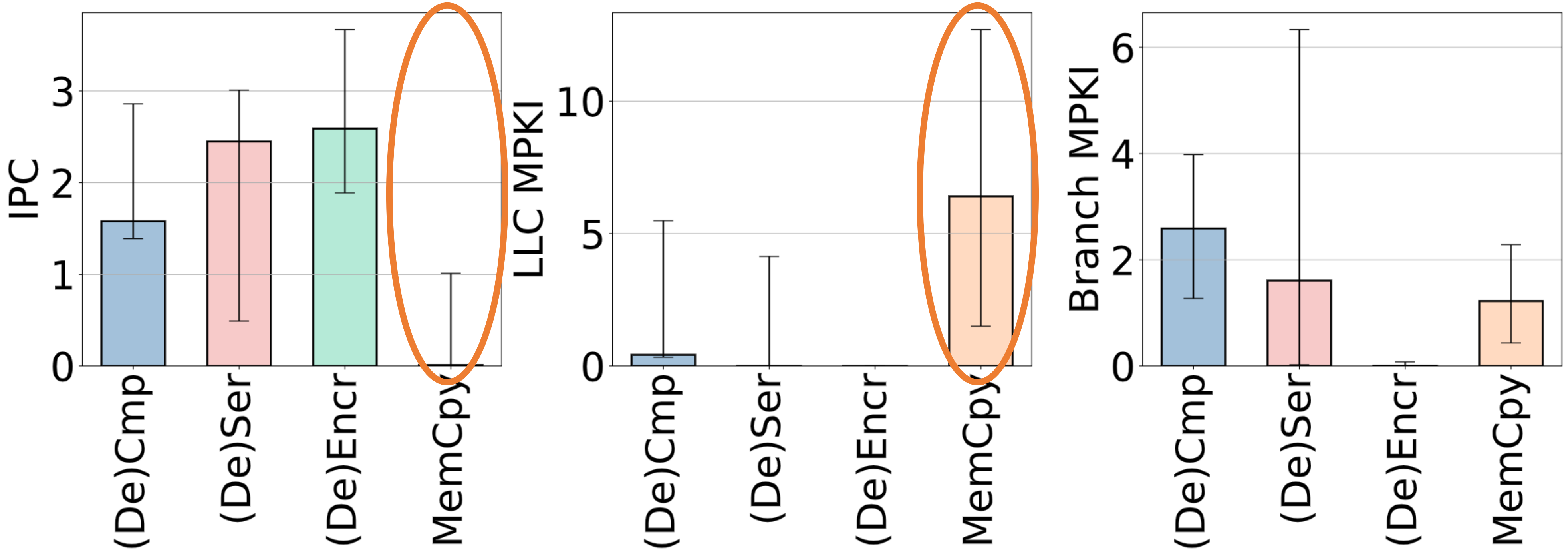
2. Datacenter Tax Operations



Distribution of metrics across datacenter taxes in *DCPerf* workloads

Heterogeneity Across Datacenter Tax

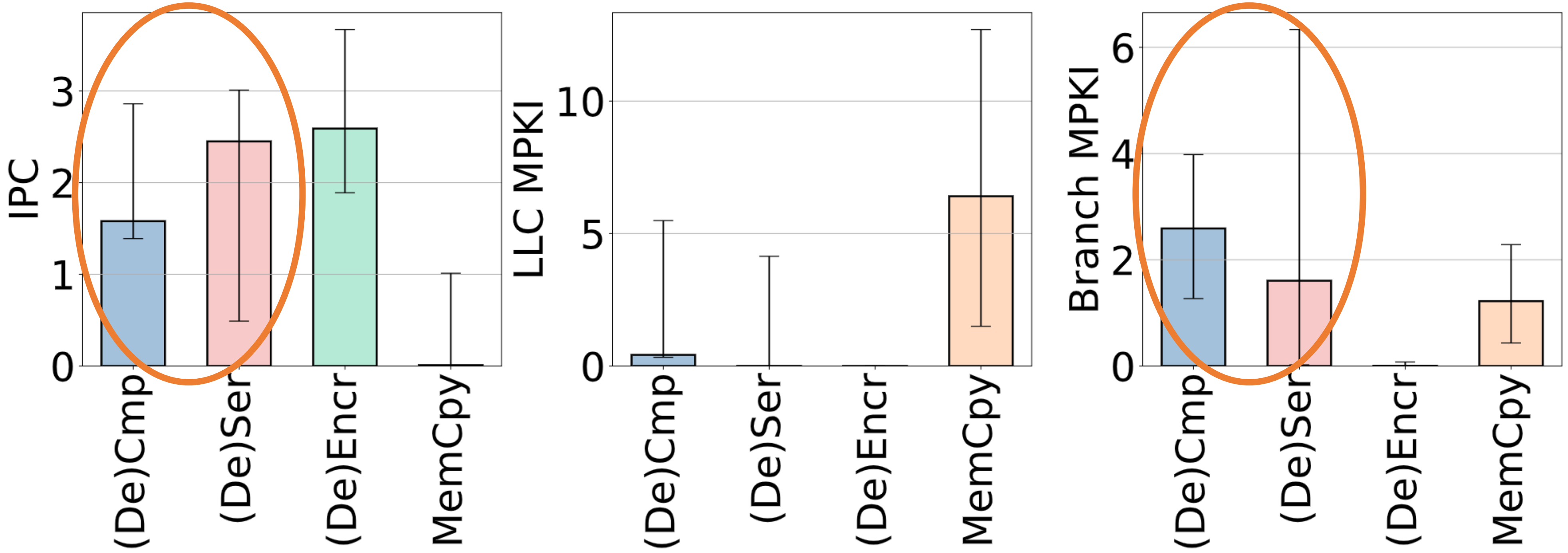
2. Datacenter Tax Operations



Distribution of metrics across datacenter taxes in *DCPerf* workloads

Heterogeneity Across Datacenter Tax

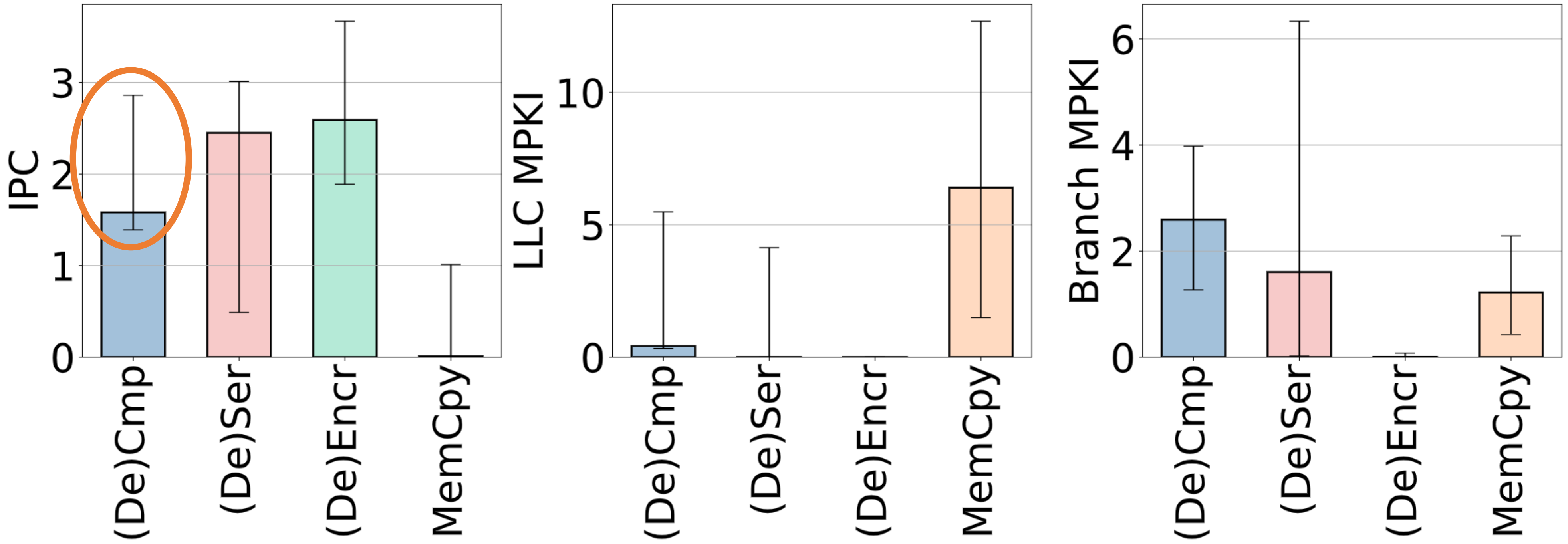
2. Datacenter Tax Operations



Distribution of metrics across datacenter taxes in *DCPerf* workloads

Heterogeneity Across Datacenter Tax

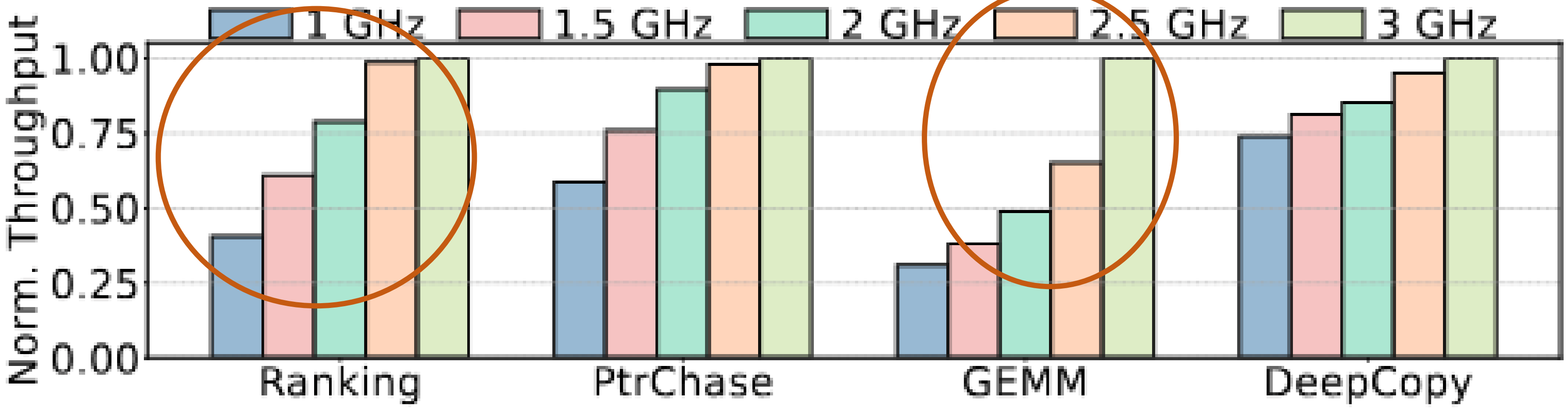
2. Datacenter Tax Operations



Distribution of metrics across datacenter taxes in *DCPerf* workloads

Heterogeneity in a Single Service

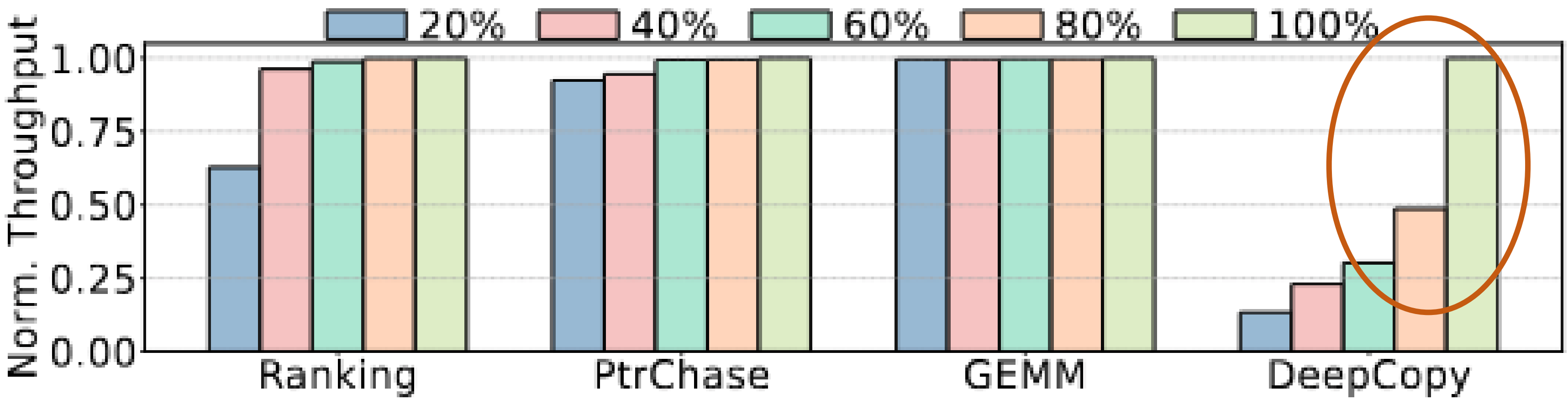
3. Multi-stage Internal Structure of Single Service



(a) Scaling CPU frequency from 3 GHz to 1 GHz.

Heterogeneity in a Single Service

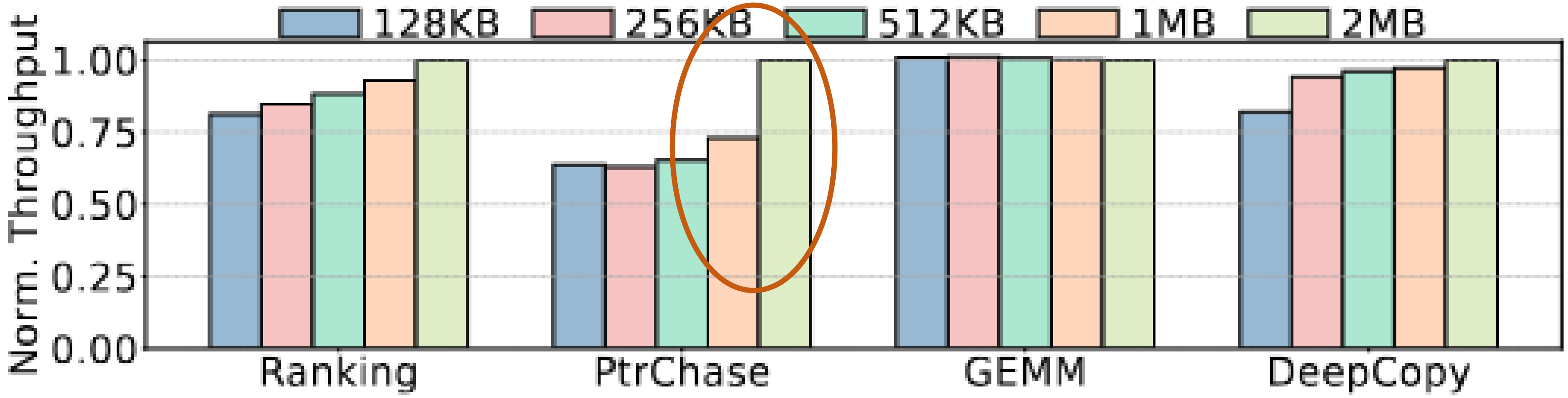
3. Multi-stage Internal Structure of Single Service



(b) Scaling memory bandwidth from 20% to 100%.

Heterogeneity in a Single Service

3. Multi-stage Internal Structure of Single Service



(c) Scaling L2 capacity from 128 KB (1 way) to 2 MB (16 ways).

High Heterogeneity Yields Inefficiencies

Case 1: Under-Provisioning

Configure for efficiency (e.g., less cores or lower frequency)

- ✅ Perf/Watt improves for memory- or network- bound phases
- ❌ Compute-intensive phases suffer from performance loss

Case 2: Under-Utilization

Match maximum resource requirement (e.g., compute)

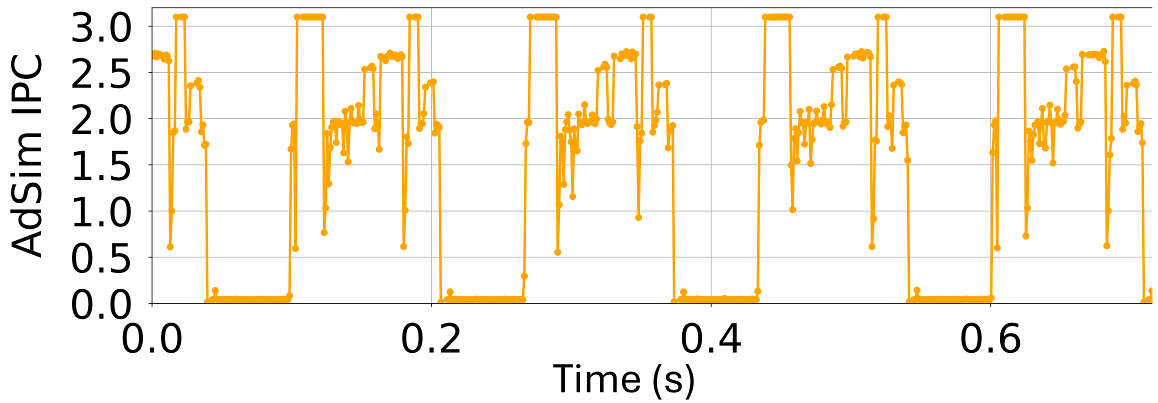
- ✅ Maximum performance for compute-intensive phases
- ❌ Resource-waste for less intensive memory- or network-bound phases

Phase Prediction is Critical

- To avoid inefficiencies, we must understand phase behavior

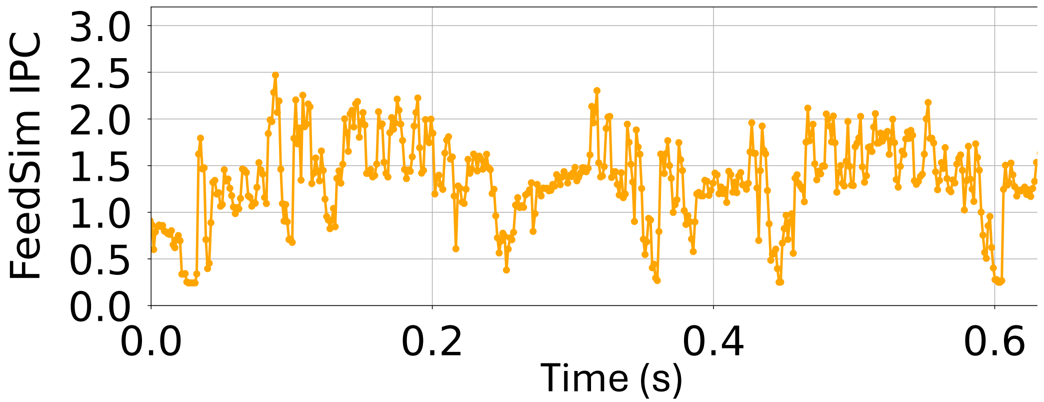
Challenge 1

- Phases are fine-grained and rapidly shifting



Challenge 2

- Phases duration and intensity varies across applications

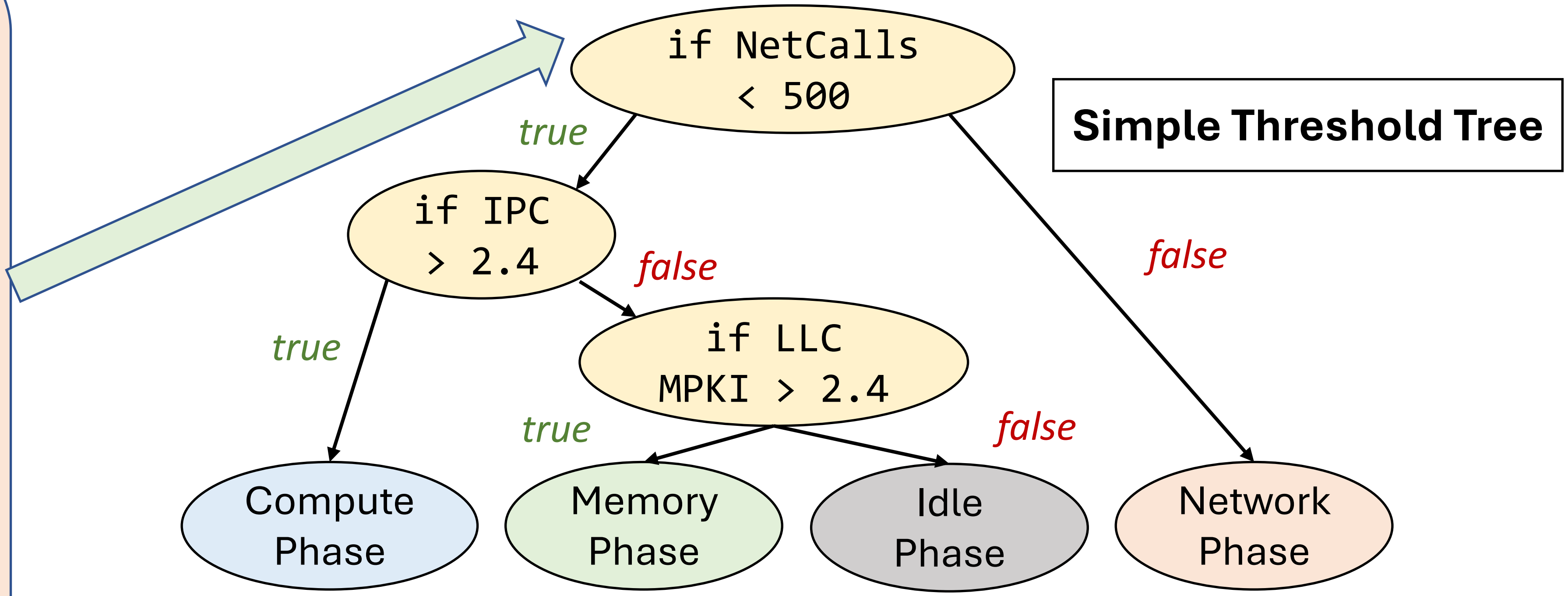


Static Prediction is Insufficient

Threshold-based Approach

Feature Vector

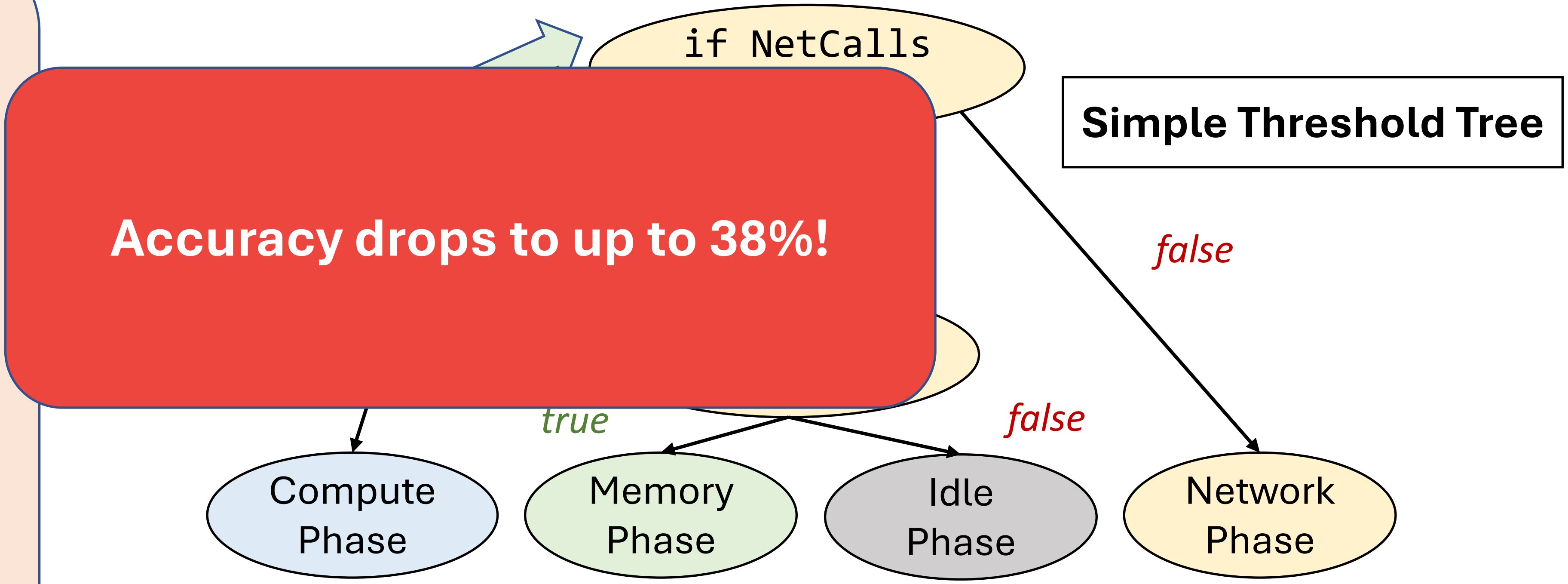
- IPC
- Cache MPKI
- TLB MPKI
- Branch MPKI
- I/O Bandwidth
- Frequency of Networking Syscalls
- …



Static Prediction is Insufficient

Threshold-based Approach

Feature Vector
- IPC
- Cache MPKI
- TLB MPKI
- Branch MPKI
- I/O Bandwidth
- Frequency of Networking Syscalls
- …

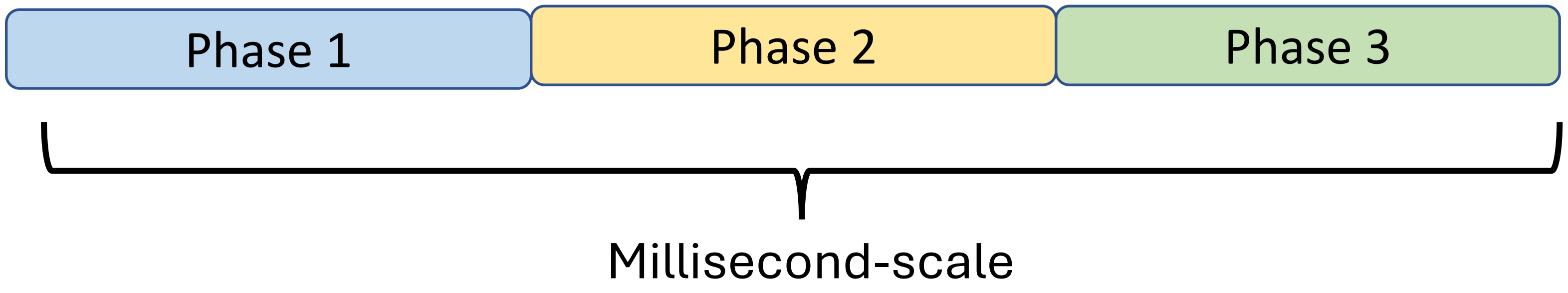


Lightweight AI for Phase Prediction

- AI provides strong pattern analysis on large, noisy datasets

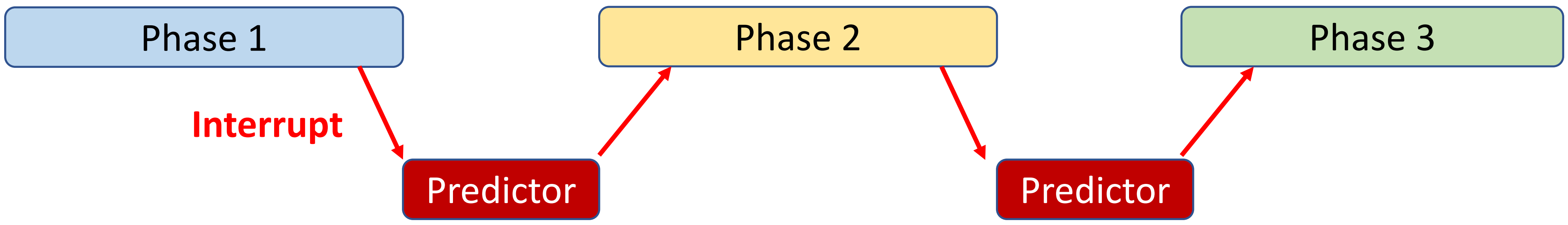
Lightweight AI for Phase Prediction

- AI provides strong pattern analysis on large, noisy datasets
- Needs to be lightweight to be able to place in hardware



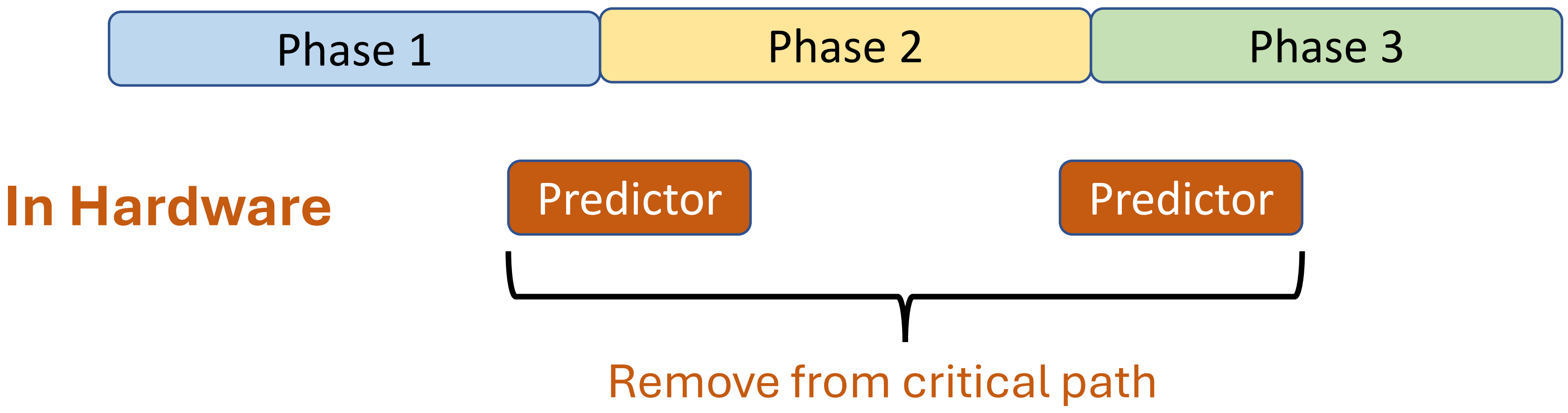
Lightweight AI for Phase Prediction

- AI provides strong pattern analysis on large, noisy datasets
- Needs to be lightweight to be able to place in hardware



Lightweight AI for Phase Prediction

- AI provides strong pattern analysis on large, noisy datasets
- Needs to be lightweight to be able to place in hardware



Candidate Lightweight AI Approaches

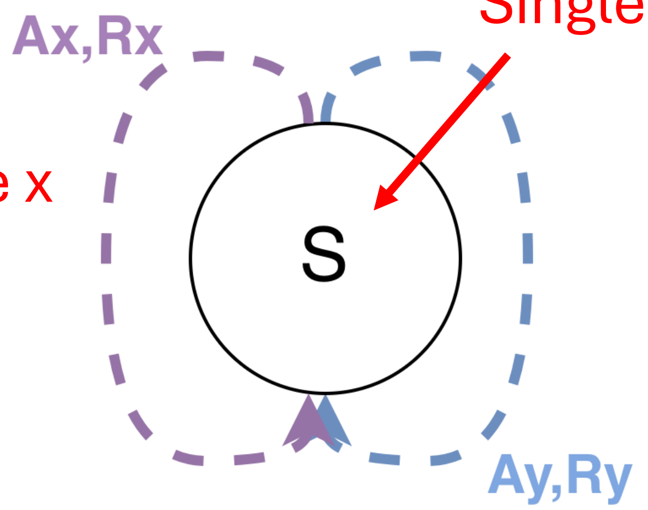
- Clustering-based Approaches

Candidate Lightweight AI Approaches

- Clustering-based Approaches
- Multi-Armed Bandits

Single State

Action = Labeling with Phase x

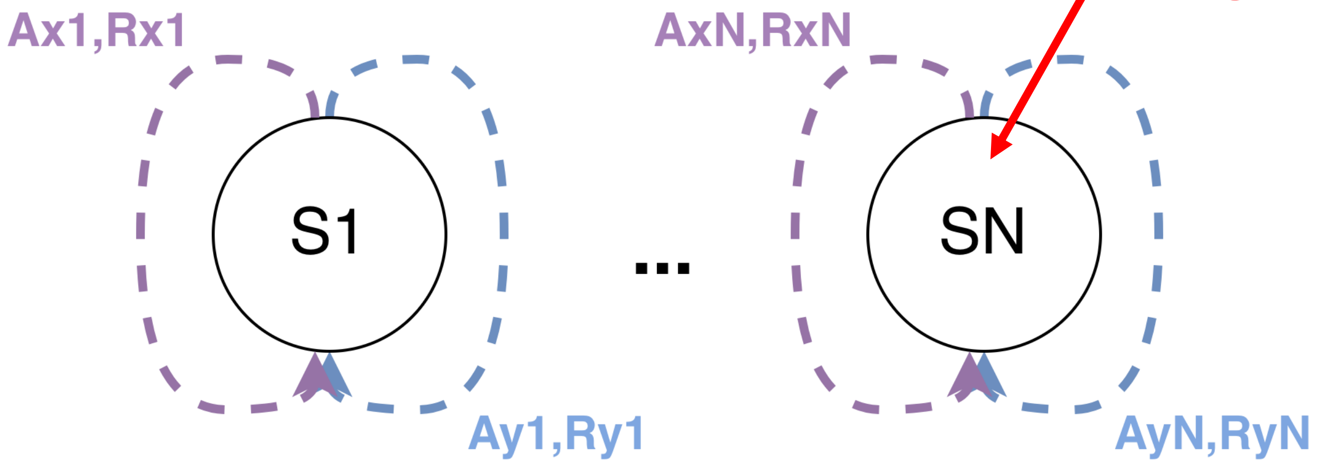


Distribution-based Reward

Candidate Lightweight AI Approaches

- Clustering-based Approaches
- Multi-Armed Bandits
- Contextual Bandits

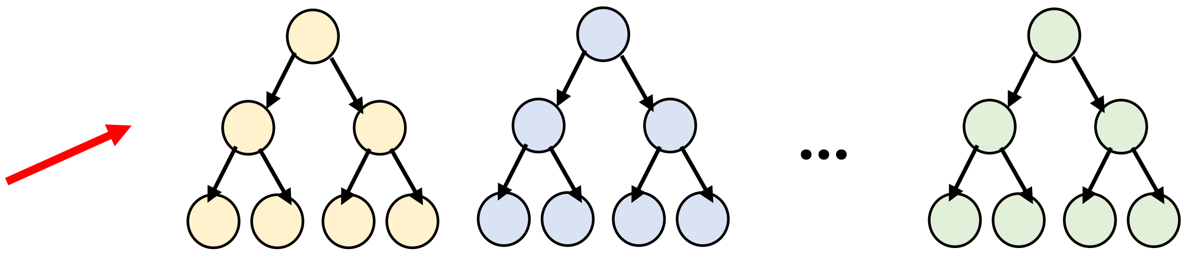
Multiple states for different contexts (e.g., feature vectors)



Candidate Lightweight AI Approaches

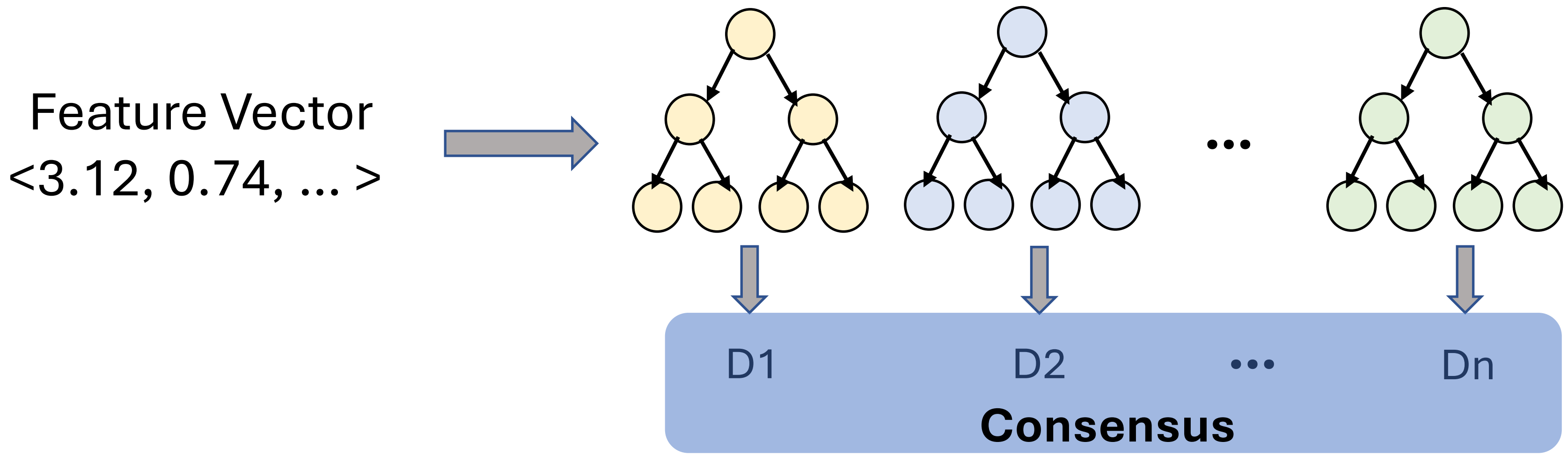
- Clustering-based Approaches
- Multi-Armed Bandits
- Contextual Bandits
- Random Forests

Pre-trained
Decision Trees

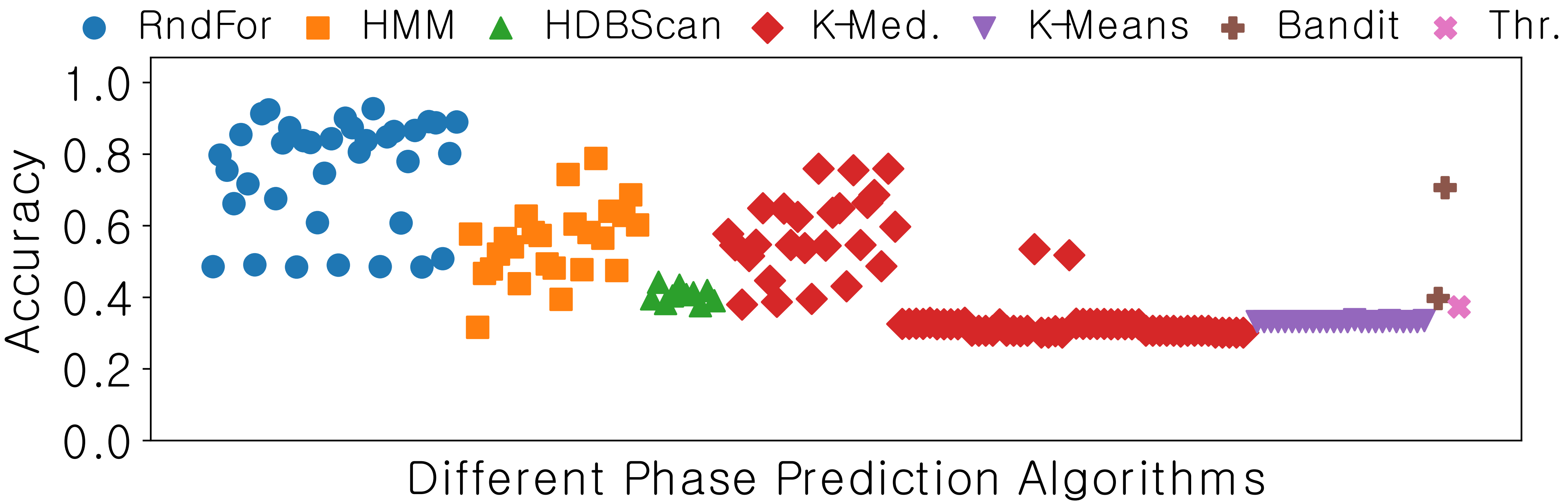


Candidate Lightweight AI Approaches

- Clustering-based Approaches
- Multi-Armed Bandits
- Contextual Bandits
- Random Forests

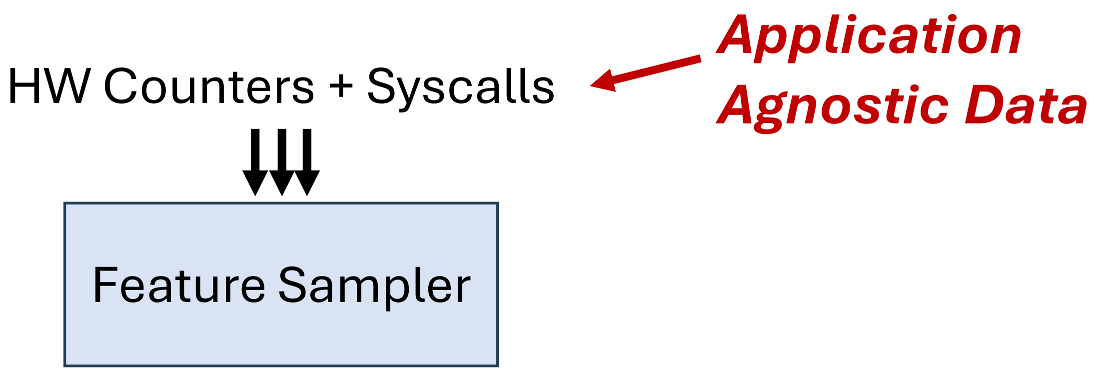


Prediction Accuracy Across Approaches



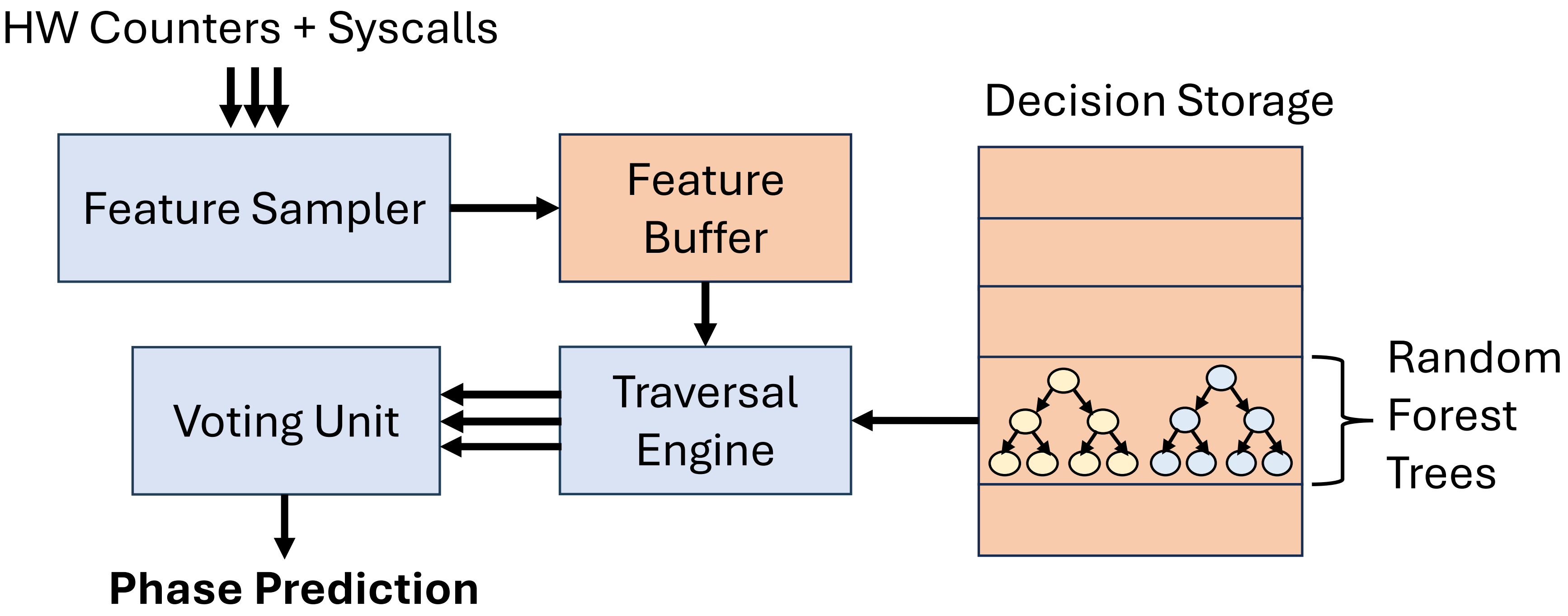
Phase Prediction – HW Support

- We implement in hardware to remove prediction from an application's critical path.



Phase Prediction – HW Support

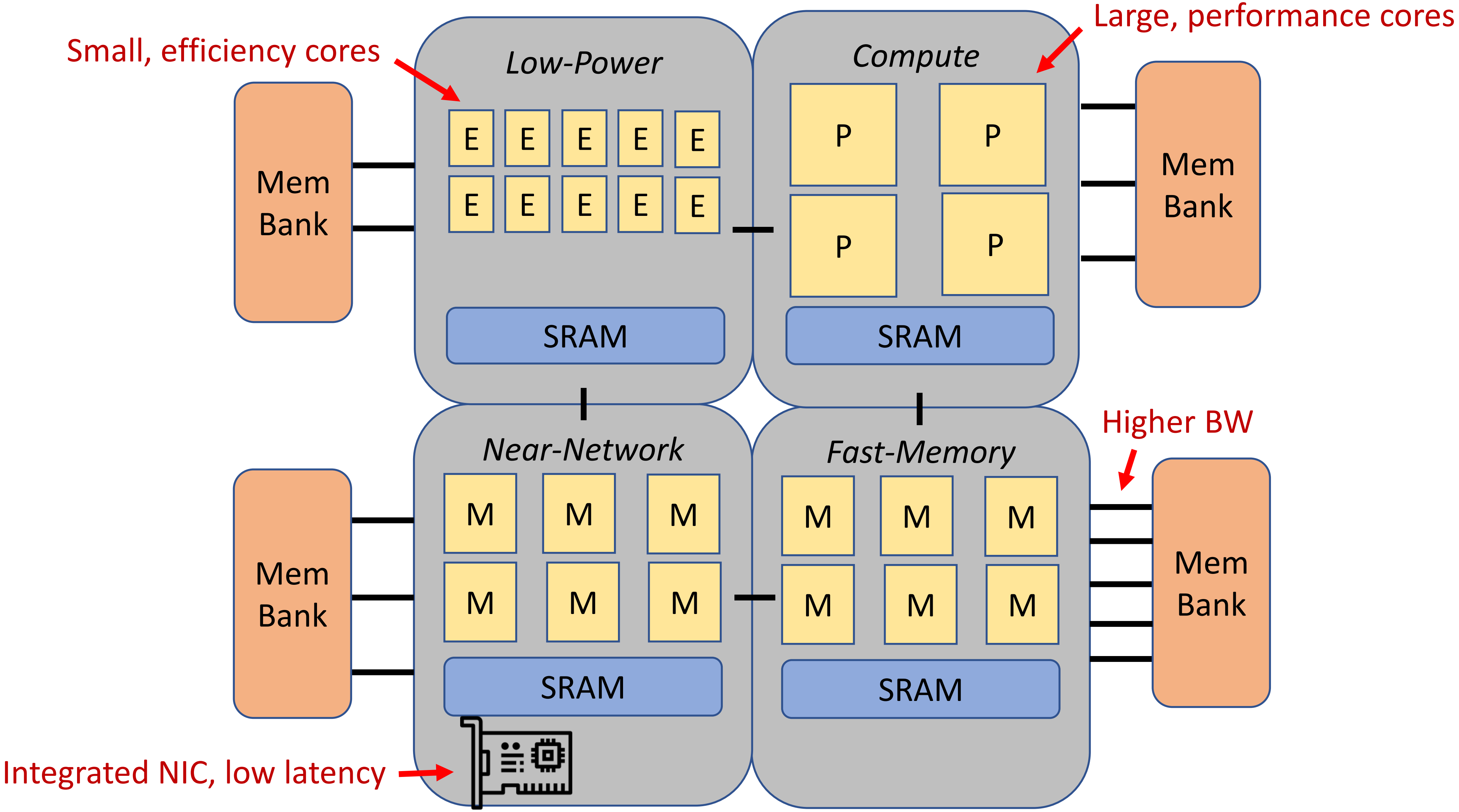
- We implement in hardware to remove prediction from an application's critical path.



Matching Workload Heterogeneity in Hardware

- Given a predicted phase, we realize the benefit by executing the thread on *specialized hardware* optimized for the phase.

Matching Workload Heterogeneity in Hardware

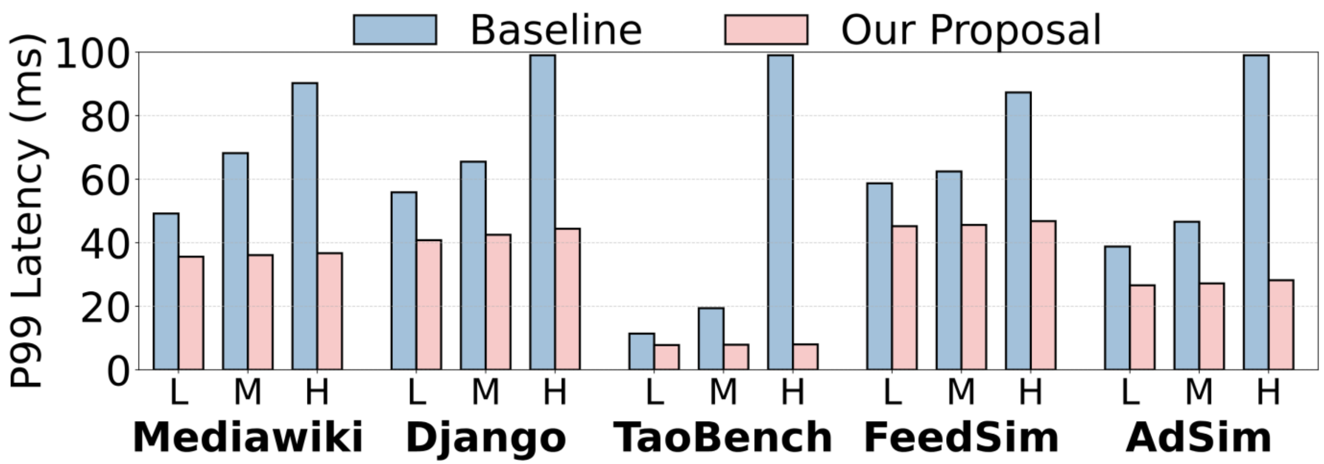


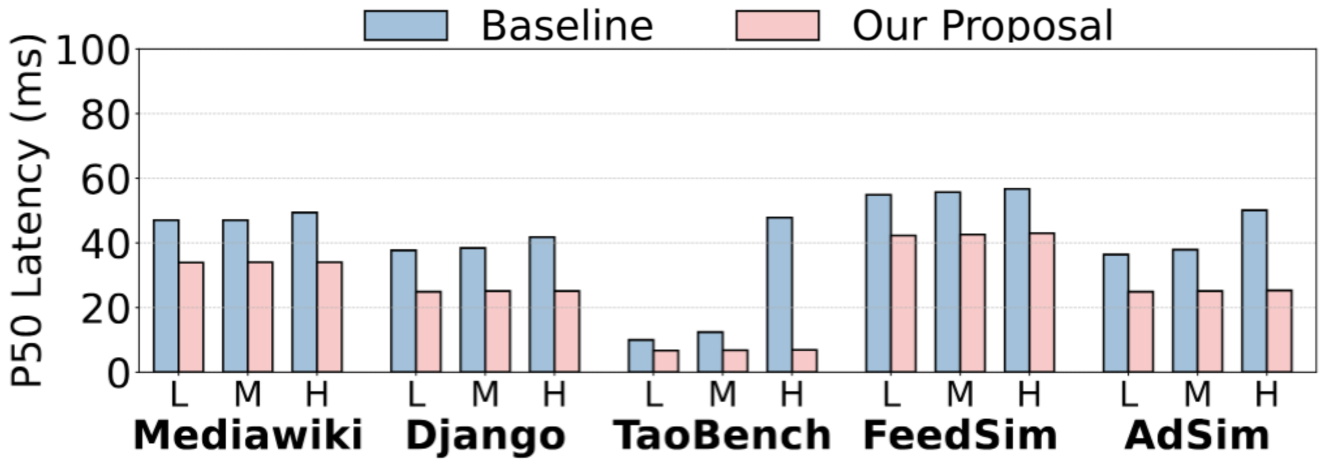
Evaluation

- Cycle-accurate full-system simulations: SST + QEMU
- DCPerf benchmark suite as evaluation workload with varying load levels (low, medium, high)
- Systems Evaluated:
    1. **Baseline**: 28-core Emerald Rapids Server
    2. **Our Proposal:** Iso-Area Heterogeneous Core Server

Evaluation

- Large reduction of P99 latency, especially at high loads
  - 28%, 42%, 65%
- Slight reduction in gain for P50 latency
  - 30%, 33%, 46%





Conclusion

In summary, we:

- Characterized phase-based execution heterogeneity in modern datacenter workloads
- Realized a need for ***lightweight AI*** for phase prediction, creating a HW implementation of Random Forests.
- Proposed a heterogeneous chiplet-based server architecture to match the workload heterogeneity
- Performed initial evaluation of our system, yielding a 65% reduction in tail latency.

Lightweight AI for Efficient Resource Management in Heterogeneous-Core Architectures

Joshua Kim[1], Chaojie Zhang[2], Íñigo Goiri[2], Christopher J. Rossbach[1,2], Jovan Stojkovic[1]

[1]The University of Texas at Austin, [2]Microsoft

TEXAS
The University of Texas at Austin
Microsoft