KarmaPM: Reward-Driven Power Manager

Author Name: **Sunil Kumar** and Vivek Kumar

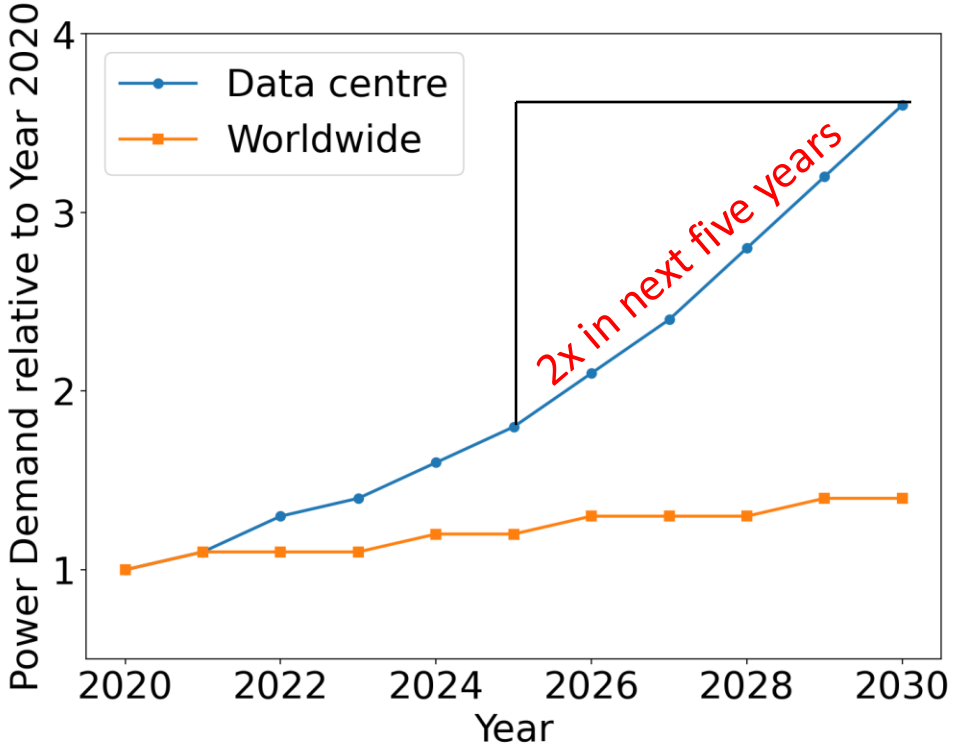
IIIT Delhi, India

Outline

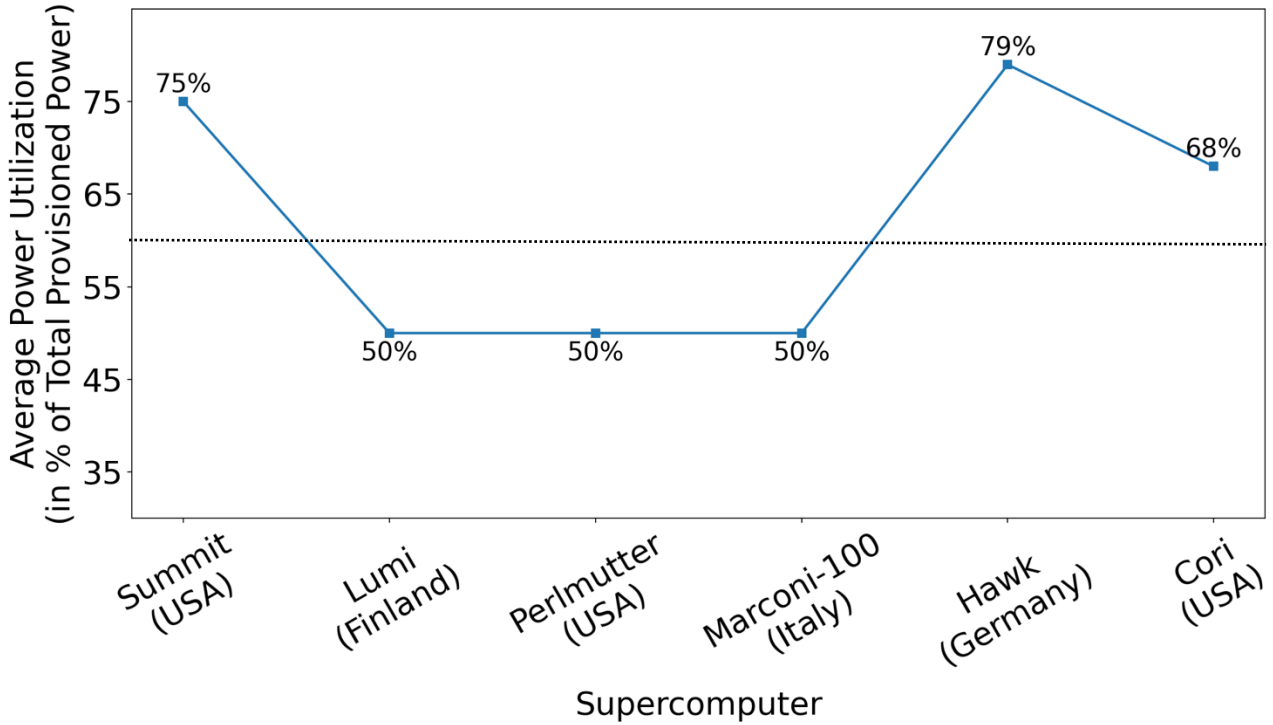
- ✓Introduction
- ✓Motivation
- ✓Related Work
- ✓Contribution
- ✓Implementation
- ✓Results
- ✓Summary

Global Perspective on Computing Power

Power demand in the data centres



Power usage at supercomputers

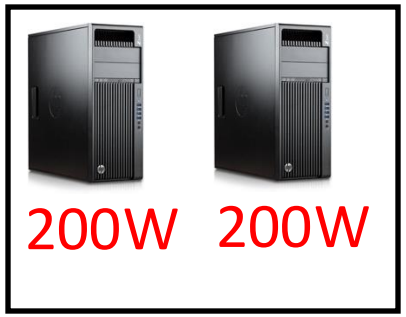


1. https://www.iea.org/reports/energy-and-ai
2. Patki et.al. [ICS2025]

It is extremely essential to improve power efficiency

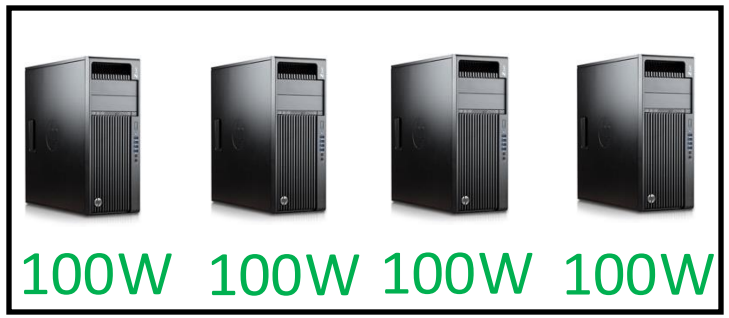
Hardware Overprovisioning using Power cap

400W



400W

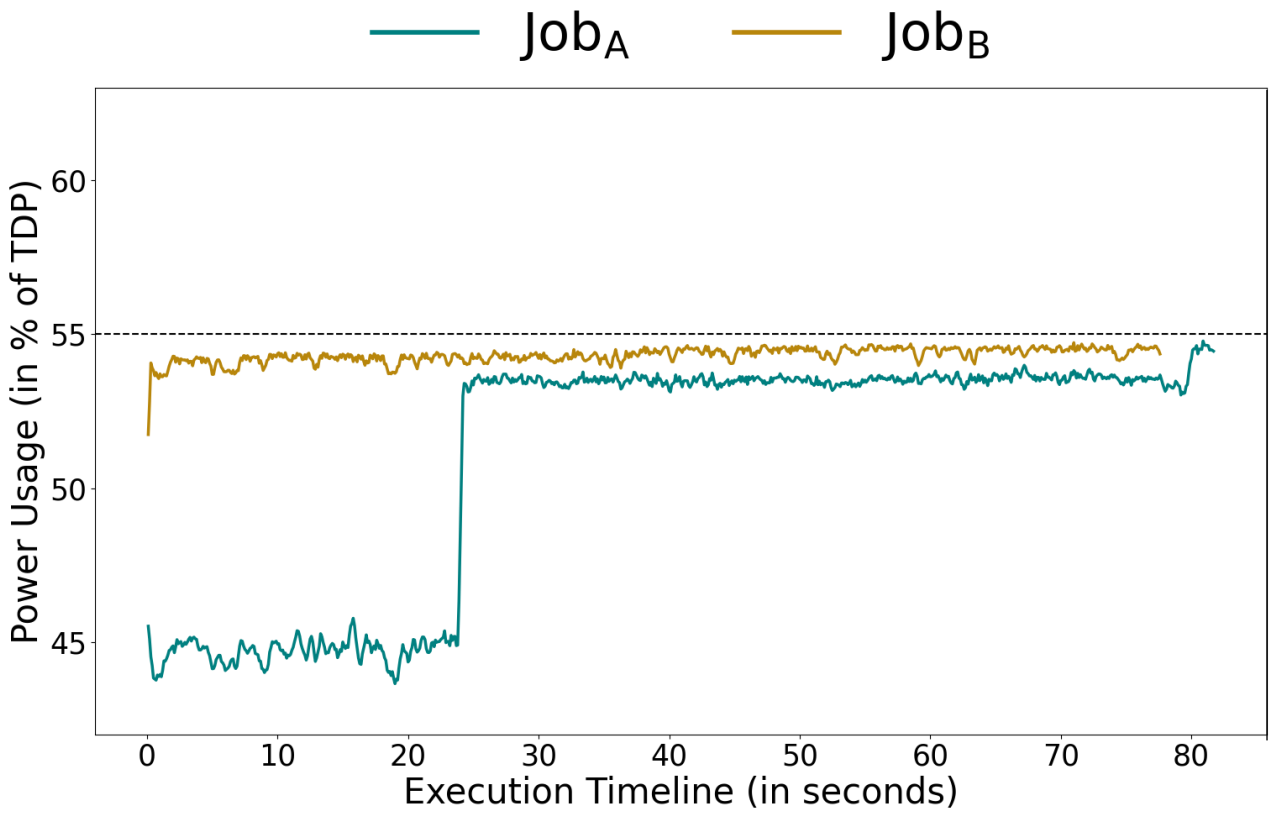
System running under **TDP**



System running under **50% of TDP**

- Servers are designed to operate within the Thermal Design Power (TDP) limit
  - TDP is the maximum power limit
- Power capping (PCAP) restricts power usage below TDP
  - Allows using more servers within the same power budget

Issues with Power Capping



Power usage changes throughout the application execution

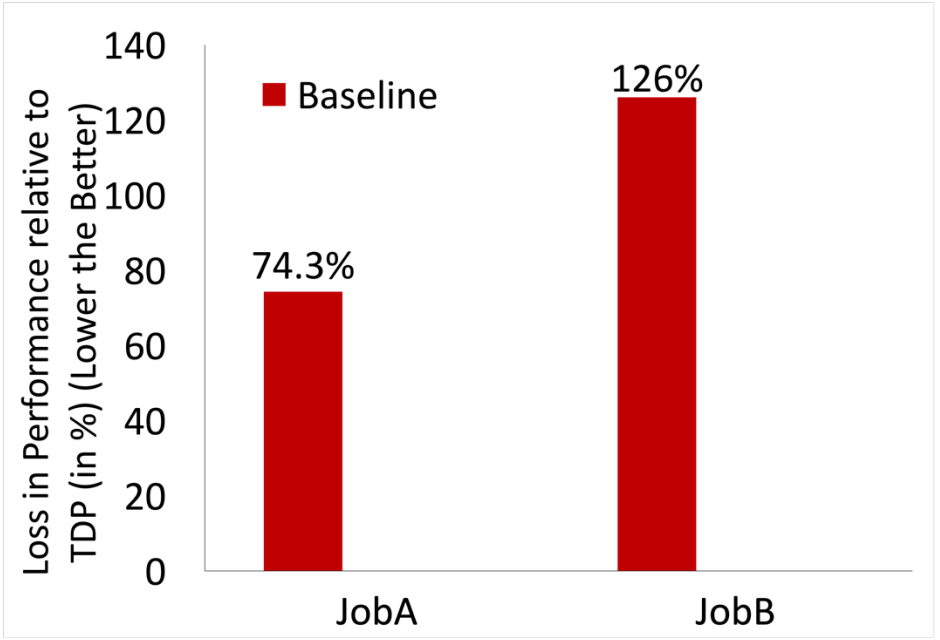
Job$_B$ fully utilizes the available power, whereas Job$_A$ only partially utilizes it

Issues with Power Capping

Power usage changes throughout the application execution

Job$_B$ fully utilizes the available power, whereas Job$_A$ only partially utilizes it

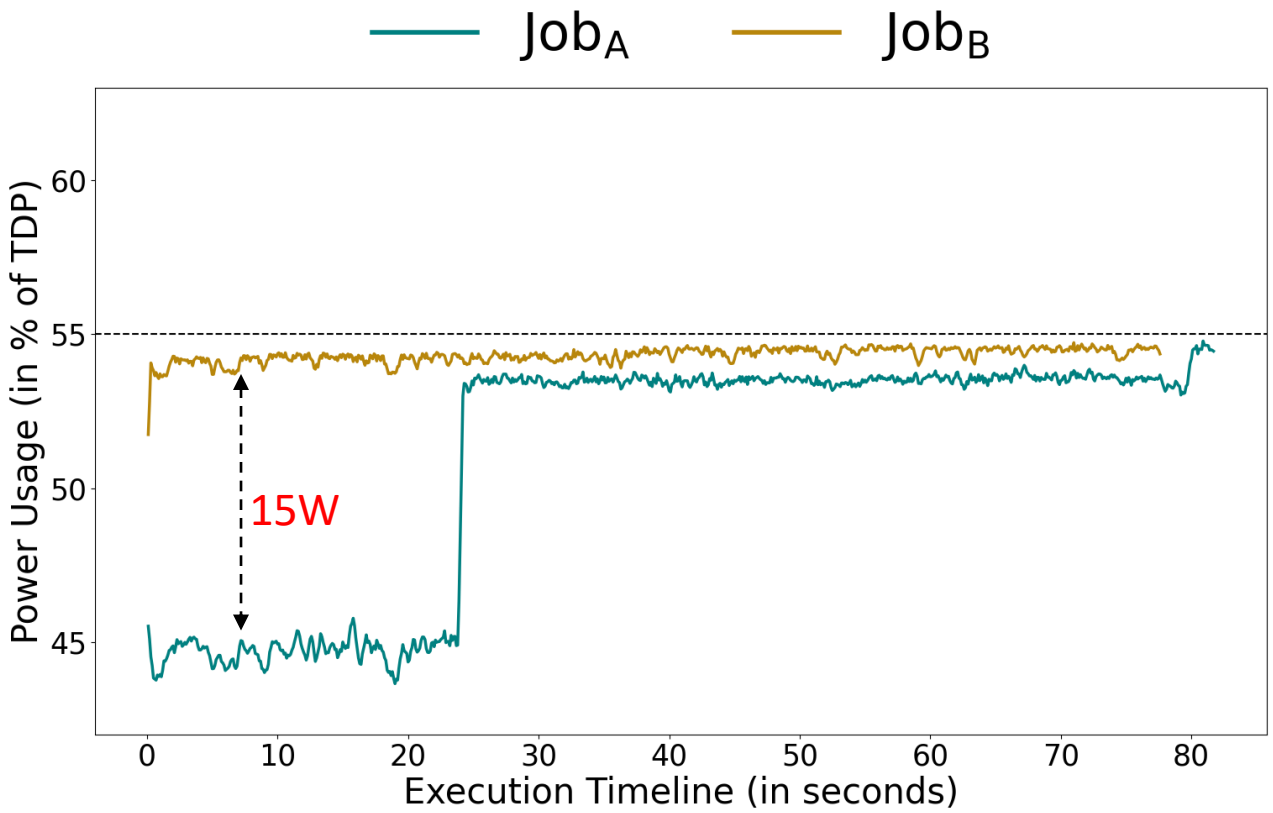
Issues with Power Capping



Power usage changes throughout the application execution

Job$_B$ fully utilizes the available power, whereas Job$_A$ only partially utilizes it

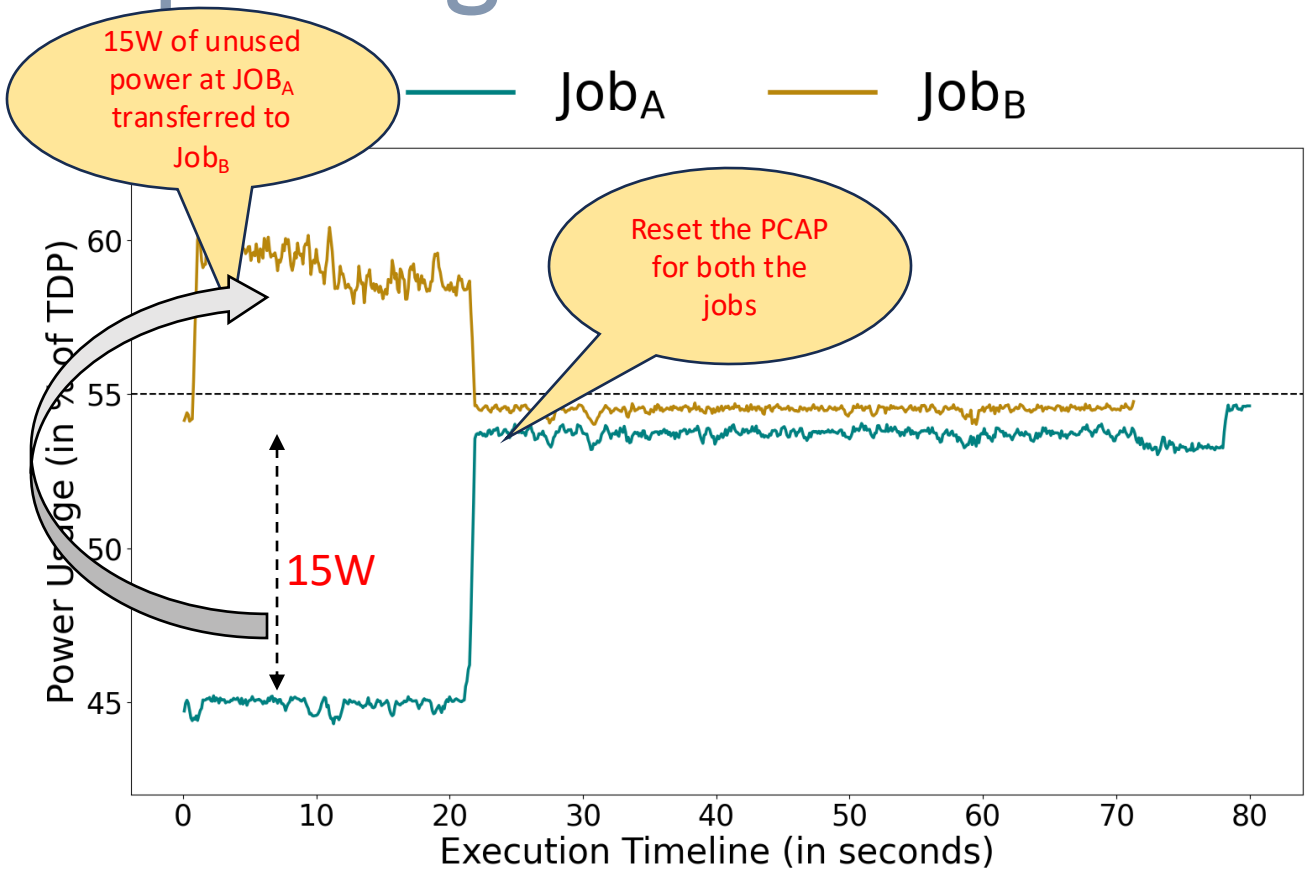
Improving Performance under PCAP



Overall system throughput improved by 3.3% (geometric mean of speedup of each application over baseline)

Co-running applications on multi-socket servers provides an opportunity to reduce power wastage

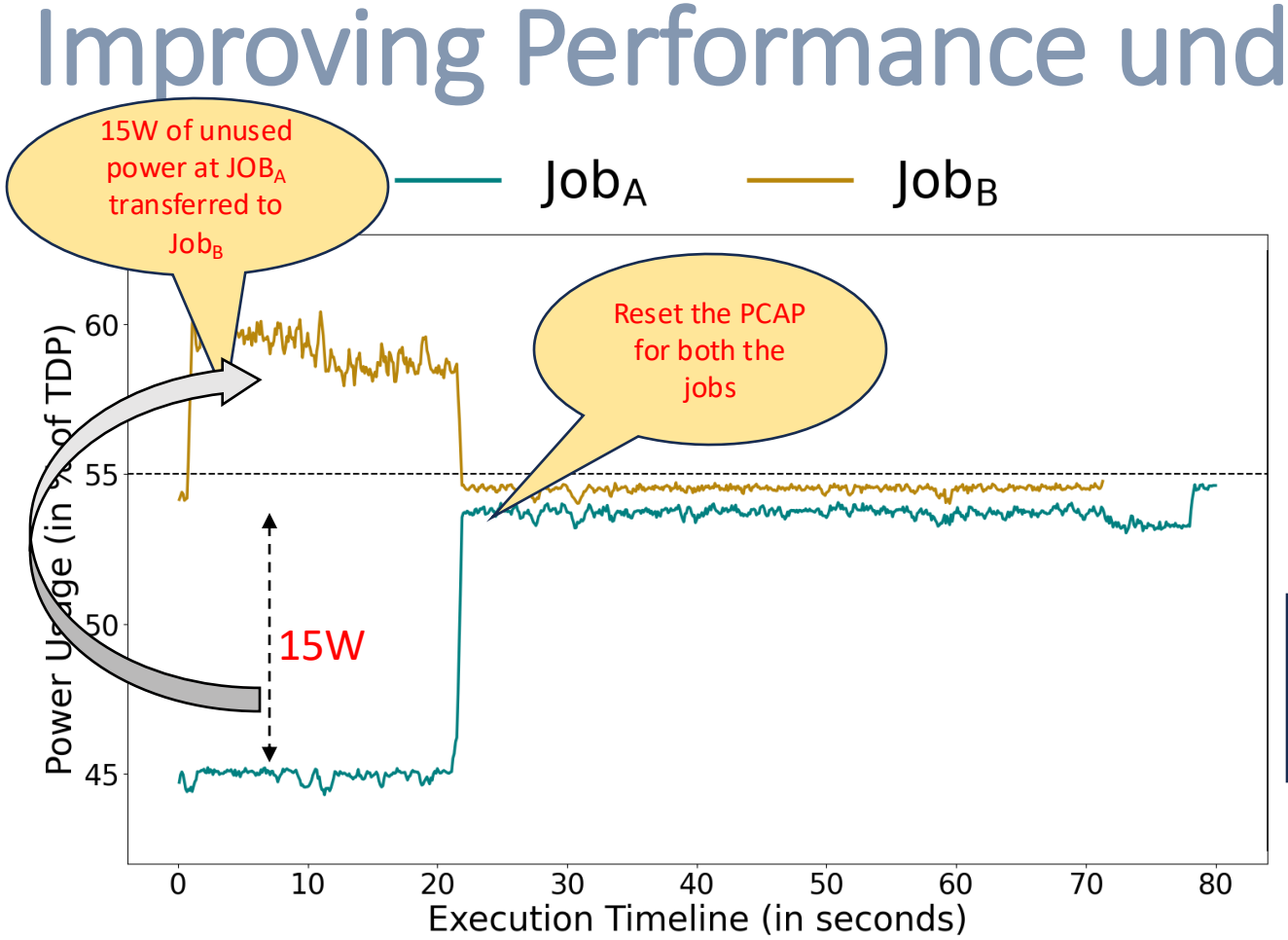
Improving Performance under PCAP

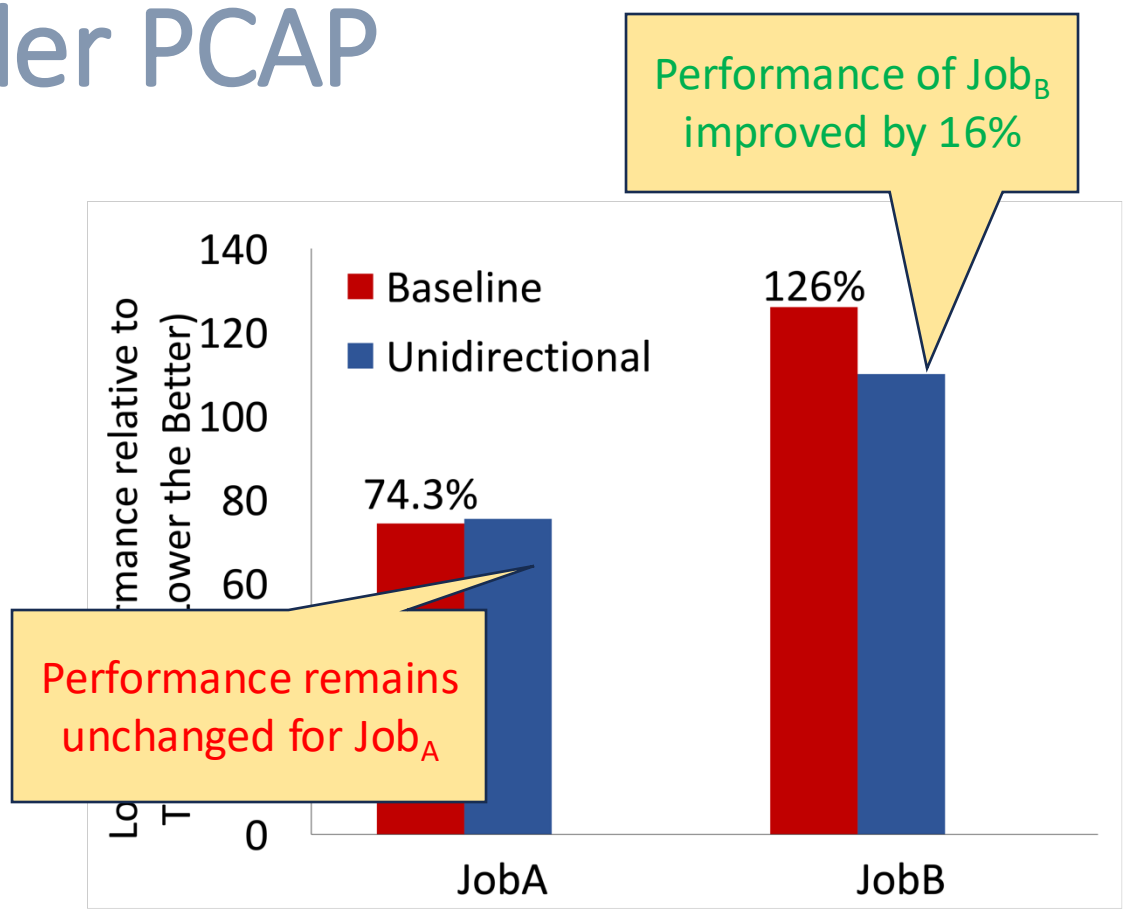


Overall system throughput improved by 3.3% (geometric mean of speedup of each application over baseline)

Co-running applications on multi-socket servers provides an opportunity to reduce power wastage

Improving Performance under PCAP

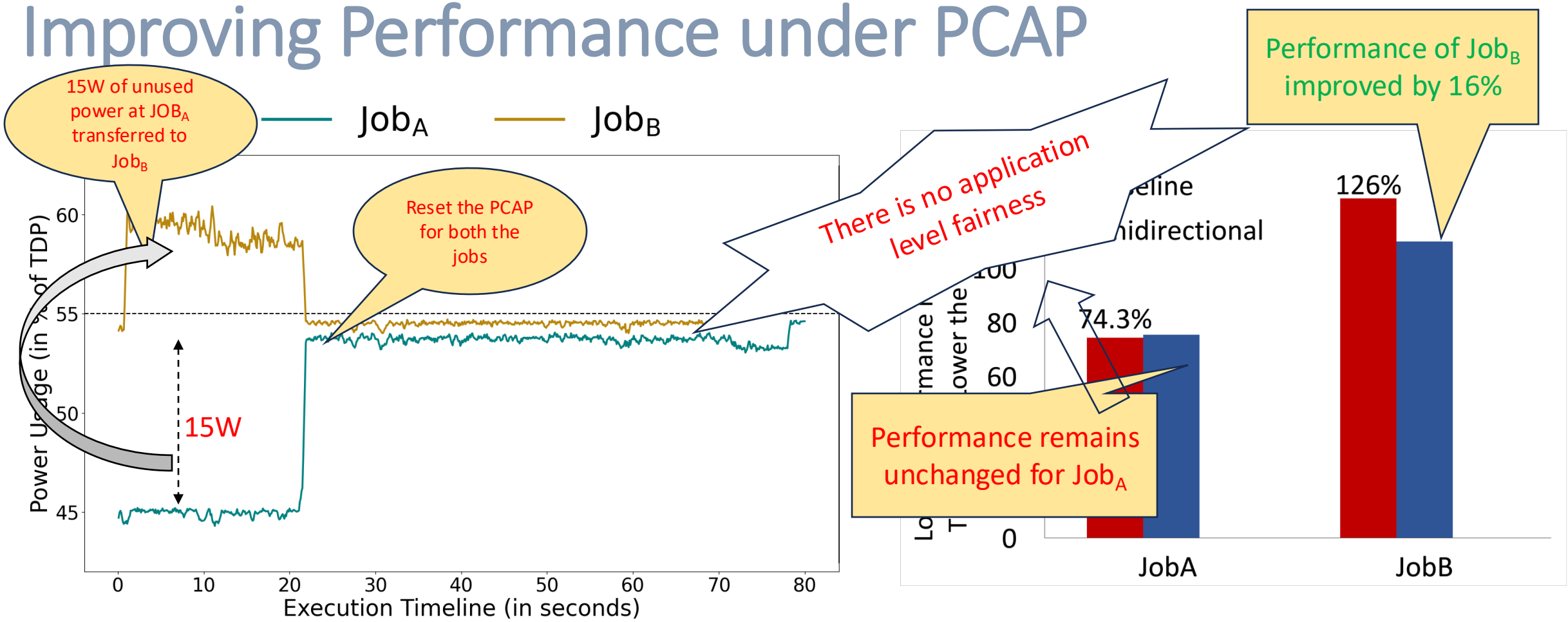




Overall system throughput improved by 3.3% (geometric mean of speedup of each application over baseline)

Co-running applications on multi-socket servers provides an opportunity to reduce power wastage

Improving Performance under PCAP



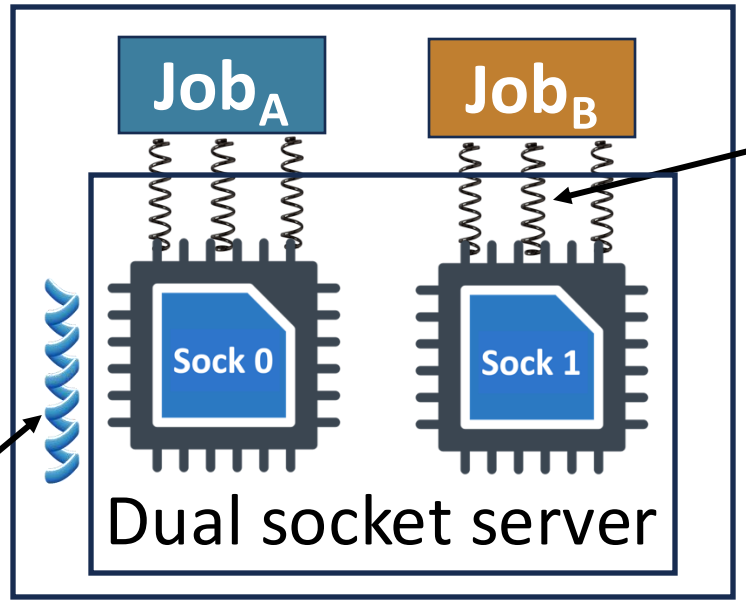
Overall system throughput improved by 3.3% (geometric mean of speedup of each application over baseline)

Co-running applications on multi-socket servers provides an opportunity to reduce power wastage

Contributions

- ✓ KarmaPM: A library-based power management system
  - ✓ A light-weight daemon that dynamically reallocates power by profiling hardware performance counters
  - ✓ ML model-free and oblivious to the parallel programming models
- ✓ Enables bi-directional power transfer between co-running jobs
  - ✓ A novel reward mechanism that improves both throughput and fairness
- ✓ Experimental evaluations on a quad-socket 72-core Intel Xeon processor
  - ✓ Using several exascale proxy applications (MPI, OpenMP and Kokkos)
- ✓ Results
  - ✓ Our results show that KarmaPM can substantially improve the system throughput and application-level fairness.

High-level Architecture of KarmaPM



Start KarmaPM daemon process

Set the user PCAP on each socket of the server

Launch Job$_A$ on Sock0 & Job$_B$ on sock1

Both Job$_A$ & Job$_B$ are running?

Yes

Record power usage and active thread count of Job$_A$ & Job$_B$

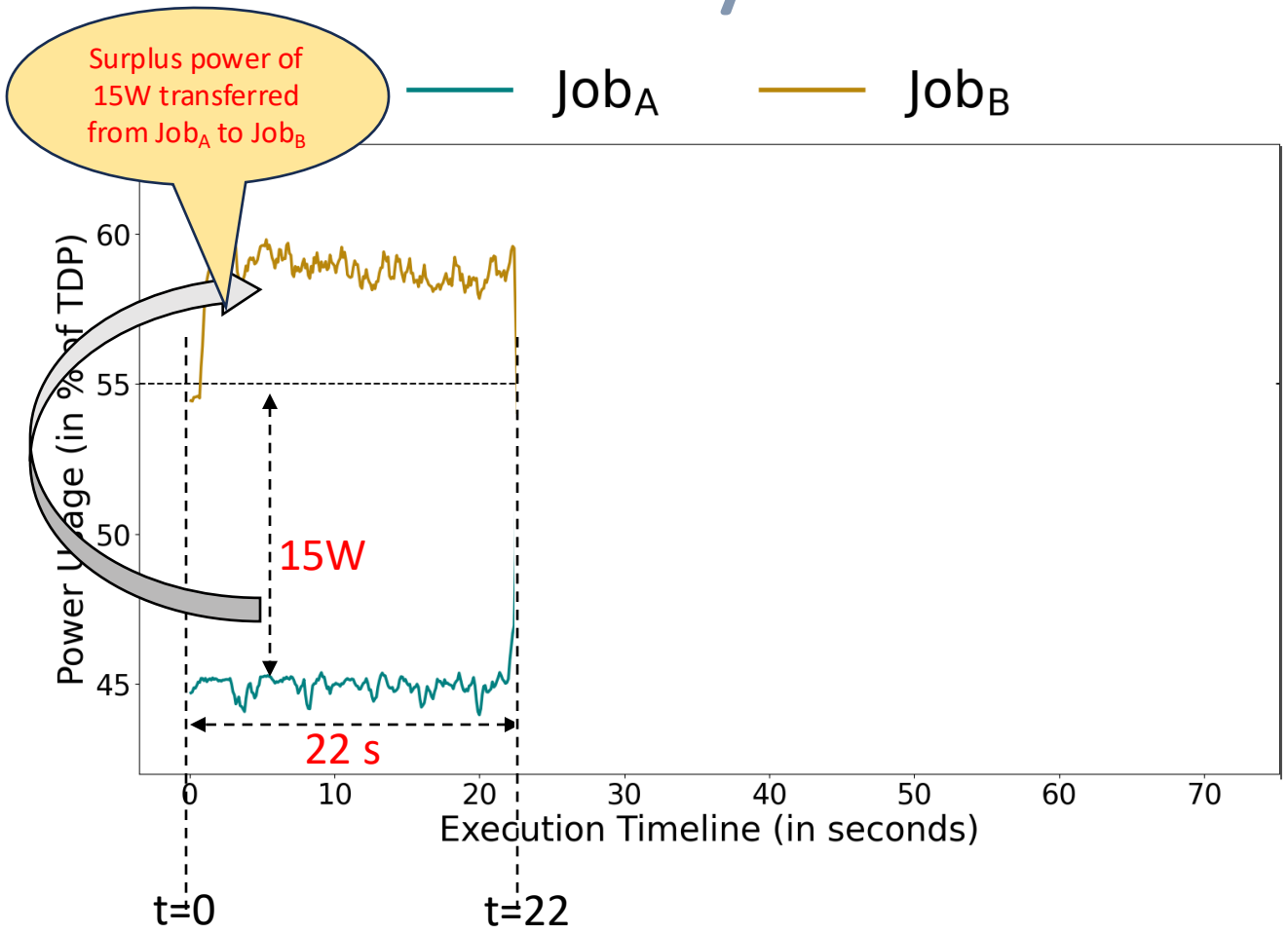
Run KarmaPM Policy

Sleep for 100 ms

No

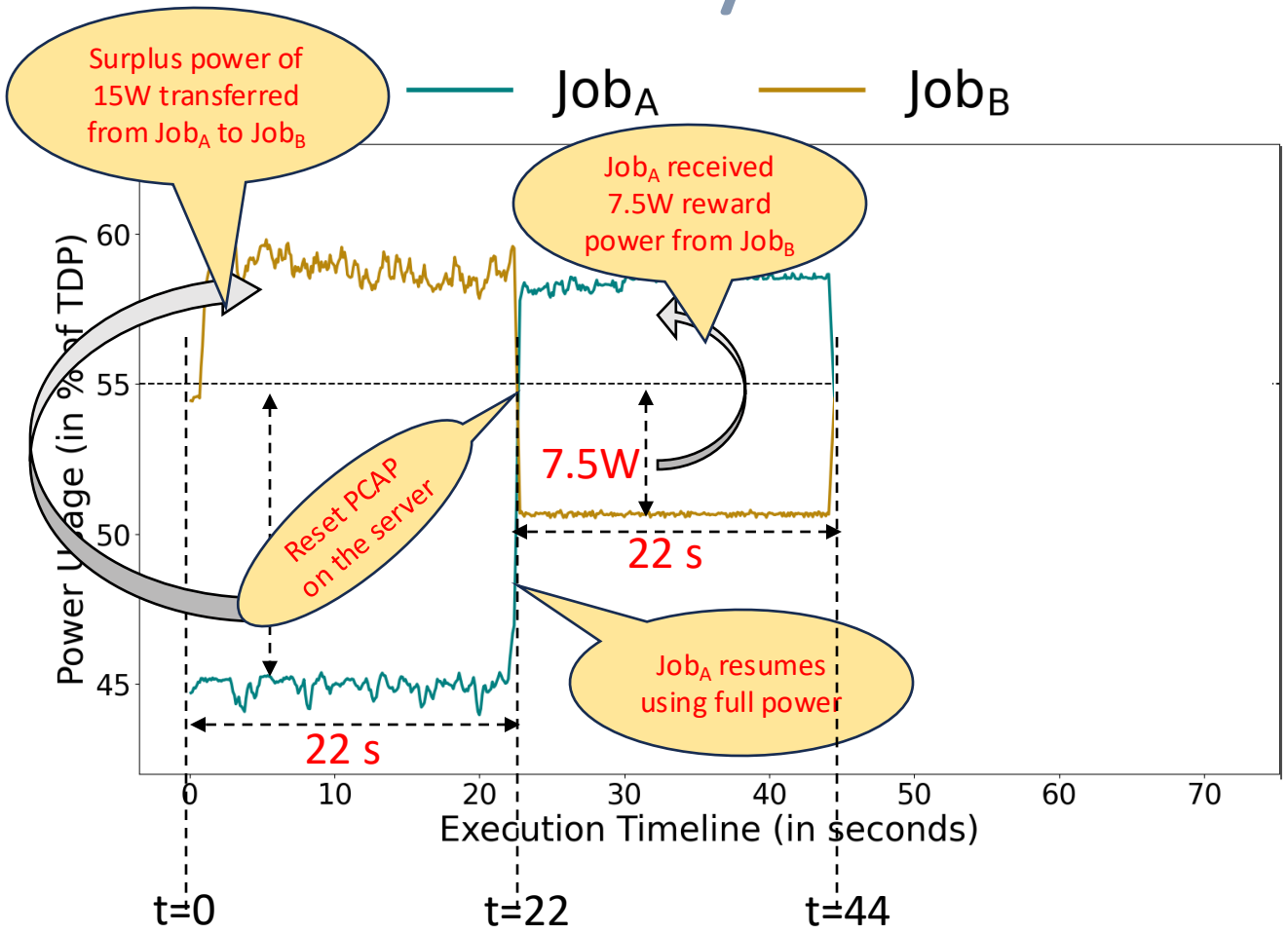
Terminate KarmaPM daemon process

KarmaPM Policy



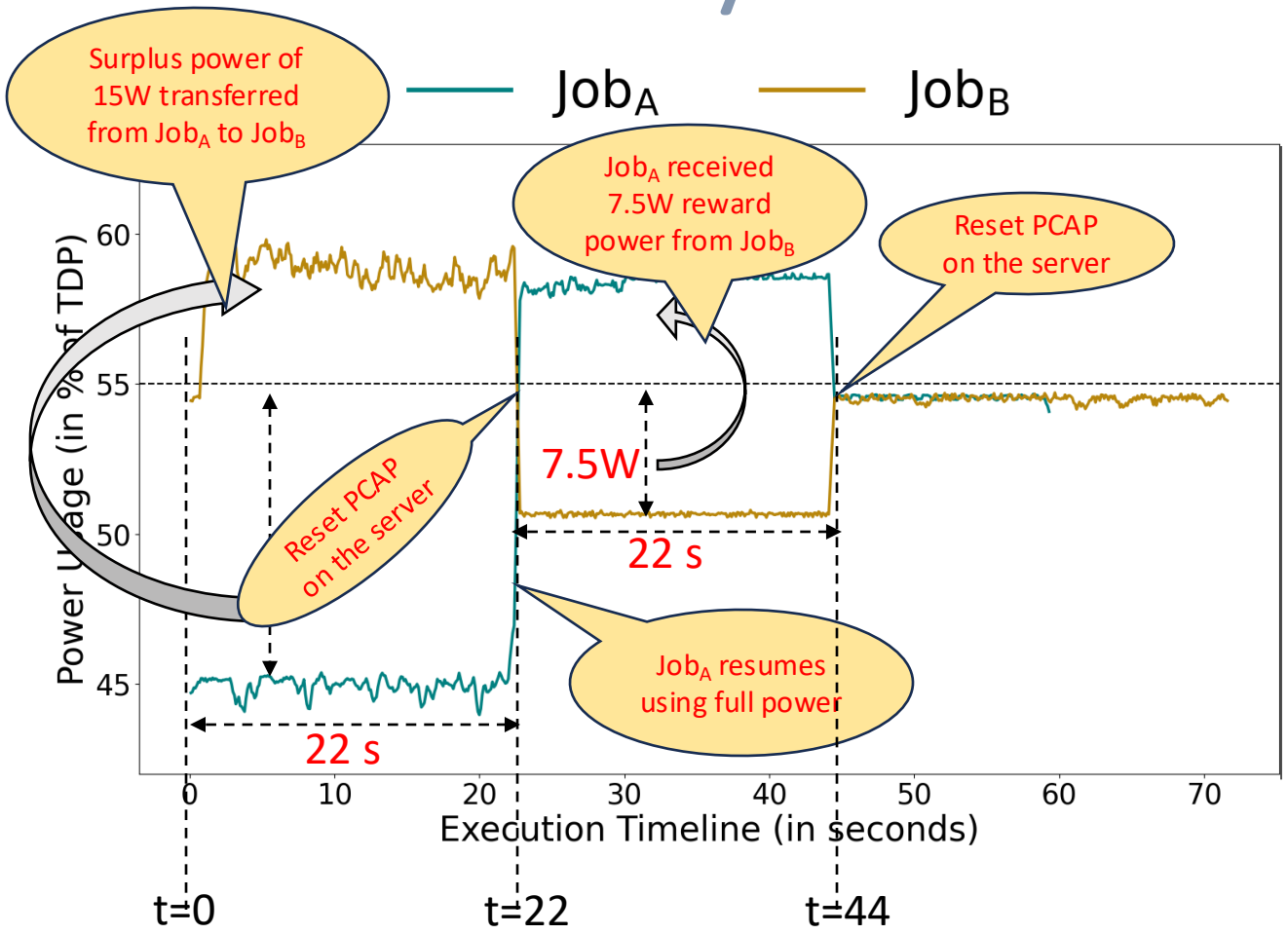
- Job$_A$ starts with low power usage, Job$_B$ with high power usage
- KarmaPM transfers **15W** surplus power from Socket0 to Socket1 at *t=0*
  - *Similar to existing approaches*

KarmaPM Policy



- Job$_A$ starts with low power usage, Job$_B$ with high power usage
- KarmaPM transfers **15W** surplus power from Socket0 to Socket1 at *t=0*
  - *Similar to existing approaches*
- After **22s**, Job$_A$ resumes using full power
  - KarmaPM resets PCAP on the server
- KarmaPM rewards Job$_A$ by returning 50% of the previously transferred power to Job$_B$ for the same duration (next 22s)
  - Provides application-level fairness

KarmaPM Policy



- Job$_A$ starts with low power usage, Job$_B$ with high power usage
- KarmaPM transfers **15W** surplus power from Socket0 to Socket1 at *t=0*
  - *Similar to existing approaches*
- After **22s**, Job$_A$ resumes using full power
  - KarmaPM resets PCAP on the server
- KarmaPM rewards Job$_A$ by returning 50% of the previously transferred power to Job$_B$ for the same duration (next 22s)
  - Provides application-level fairness
- Execution continues with the user-set PCAP at JobA and JobB after t=44

Experimental Methodology

Exascale OpenMP proxy applications

- ✓Pennant
- ✓MiniFE (Kokkos)
- ✓Quicksilver
- ✓CoMD (Kokkos)
- ✓SimpleMOC (MPI)
- ✓PathFinder
- ✓RSBench
- ✓CG (NPB suite)

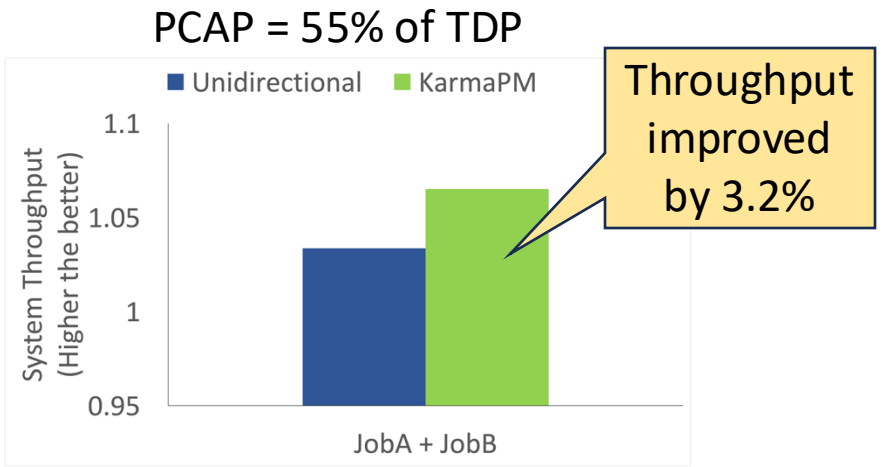
Hardware platform

- ✓Quad socket Intel Xeon Cooper Lake
  - ✓**18** cores per socket
  - ✓TDP per socket = 150 Watts

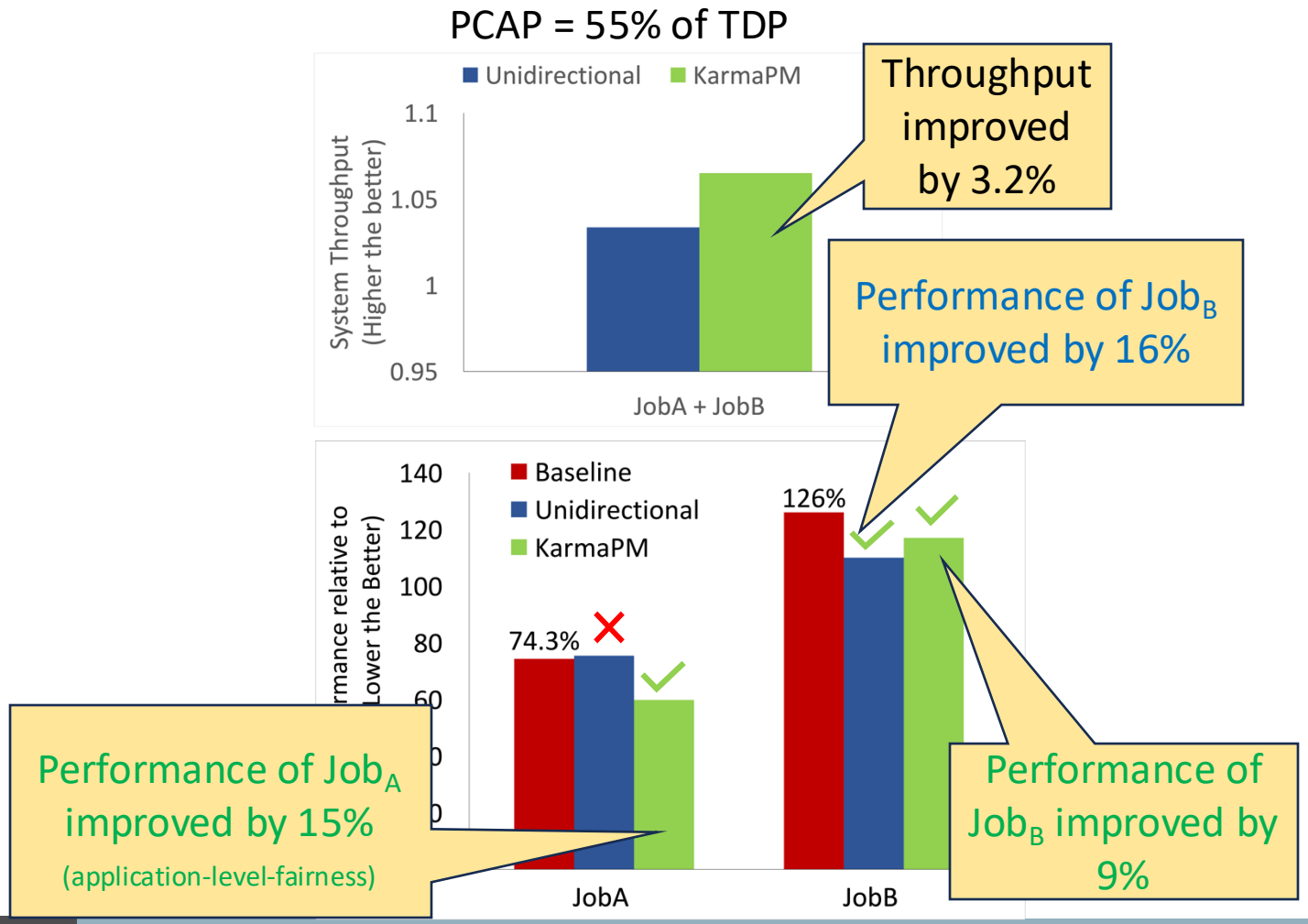
Mix Type	Number of Mixes	Socket Binding
4 Applications	5	Each application uses one socket
2 Applications	1	Each application uses two socket

Evaluated using three PCAP settings, 55%, 65%, and 75% of TDP

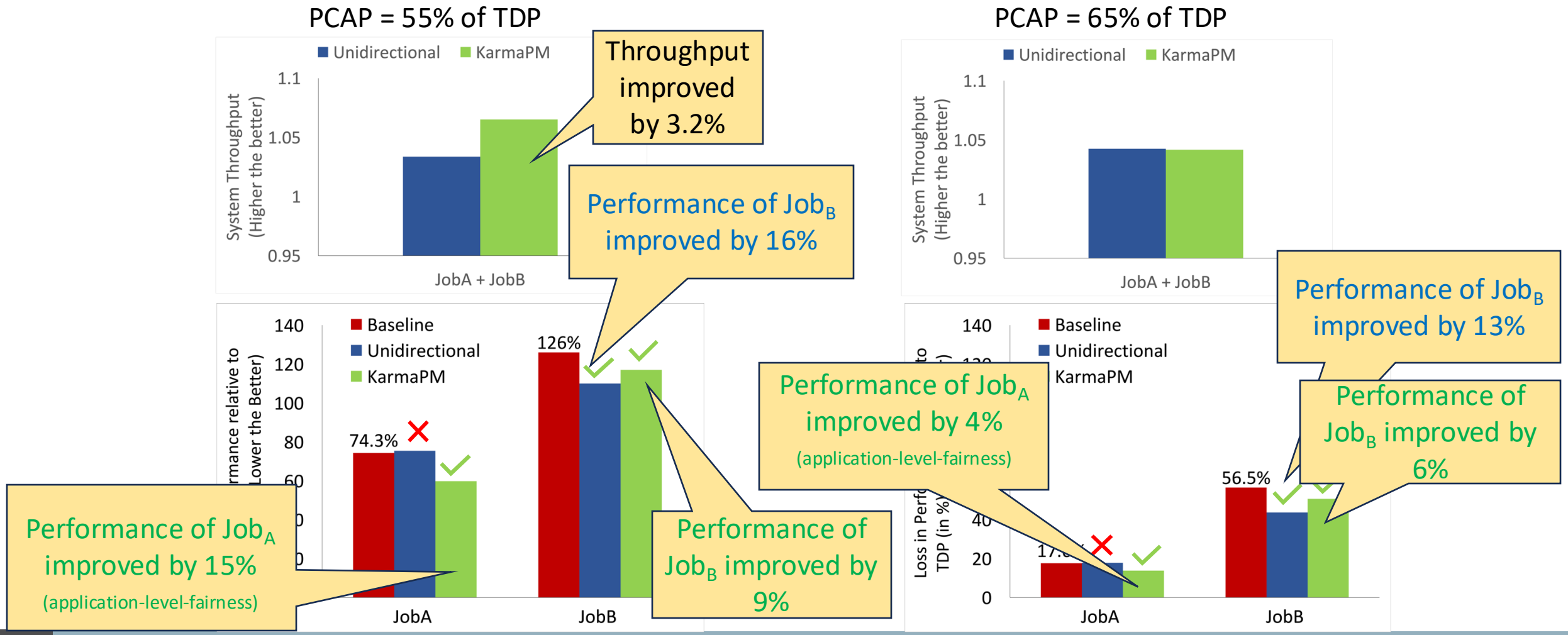
Throughput and Fairness from KarmaPM



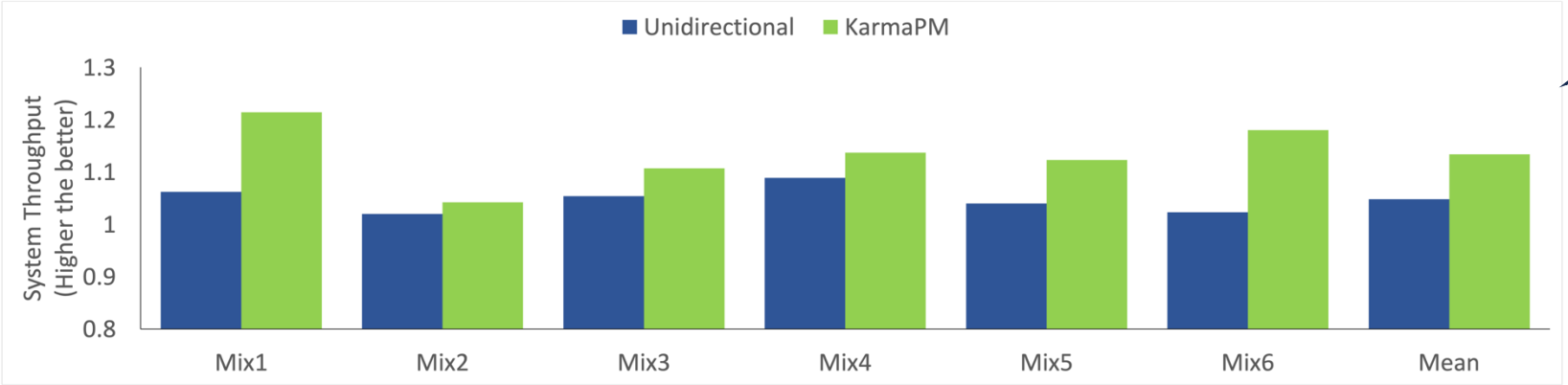
Throughput and Fairness from KarmaPM



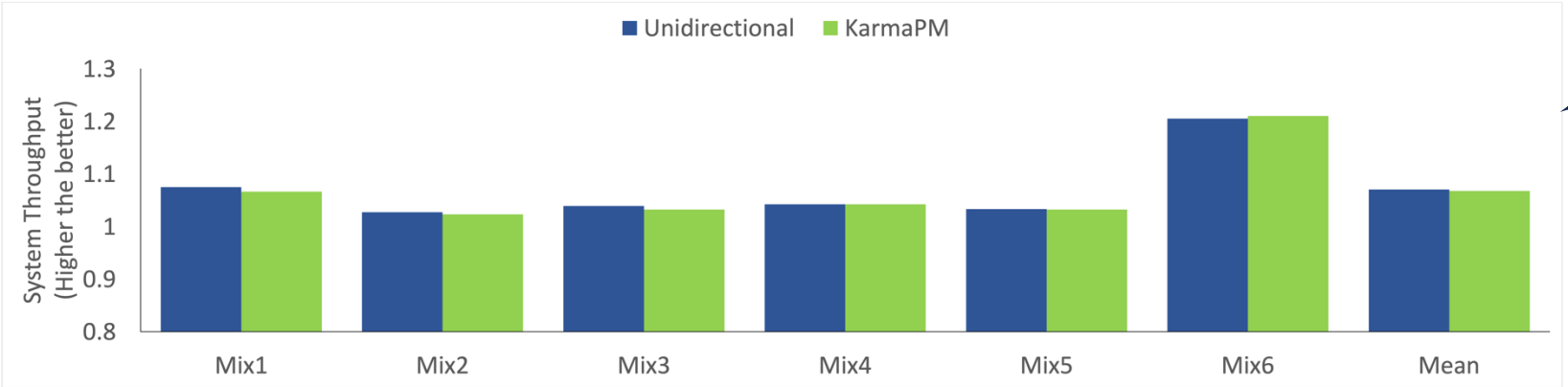
Throughput and Fairness from KarmaPM



System Throughput from KarmaPM



PCAP=55% of TDP



PCAP=65% of TDP

- KarmaPM improves both throughput and fairness at low PCAP
- At higher PCAPs, KarmaPM improves fairness without affecting throughput

Summary

- Hardware overprovisioning using power capping addresses the increasing computing power demand
  - However, PCAP degrades the application performance
- Running applications in pairs on a single server provides an opportunity to reduce power wastage by transferring unused power from one application to other
  - However, this approach does not support application-level fairness
- **KarmaPM** uses a novel reward-driven bi-directional power transfer mechanism that improves both throughput and application-level fairness
- In future, we plan to extend KarmaPM for heterogeneous architecture (CPU+GPU)

Q&A

Acknowledgement

This research is supported by Google PhD Fellowship 2022