European Commission | HORIZON EUROPE

MLSysOps

DPUConfig: Optimizing ML Inference in FPGAs Using Reinforcement Learning

Alexandros Patras, Spyros Lalis, Christos D. Antonopoulos, Nikolaos Bellas

Dept. Electrical and Computer Engineering,
University of Thessaly

Volos, Greece

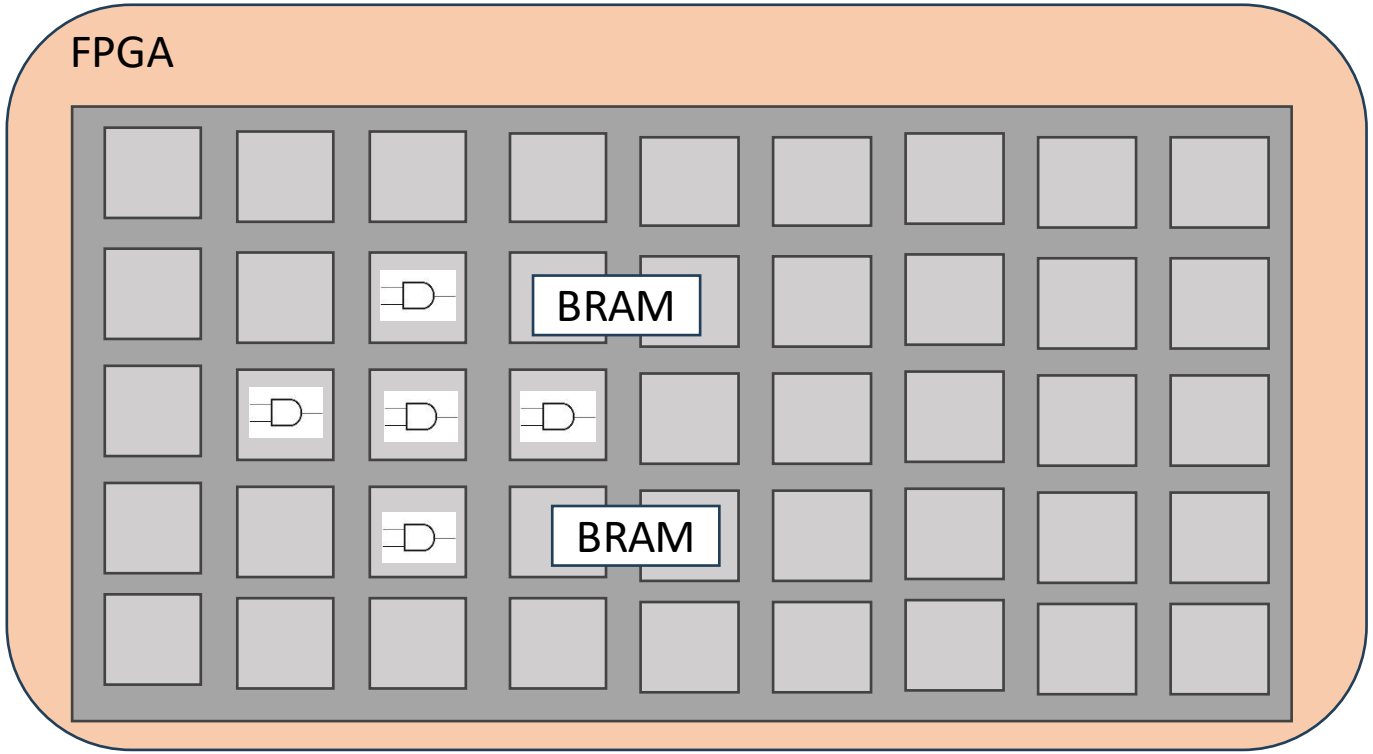
CSL

HiPEAC

UNIVERSITY OF THESSALY

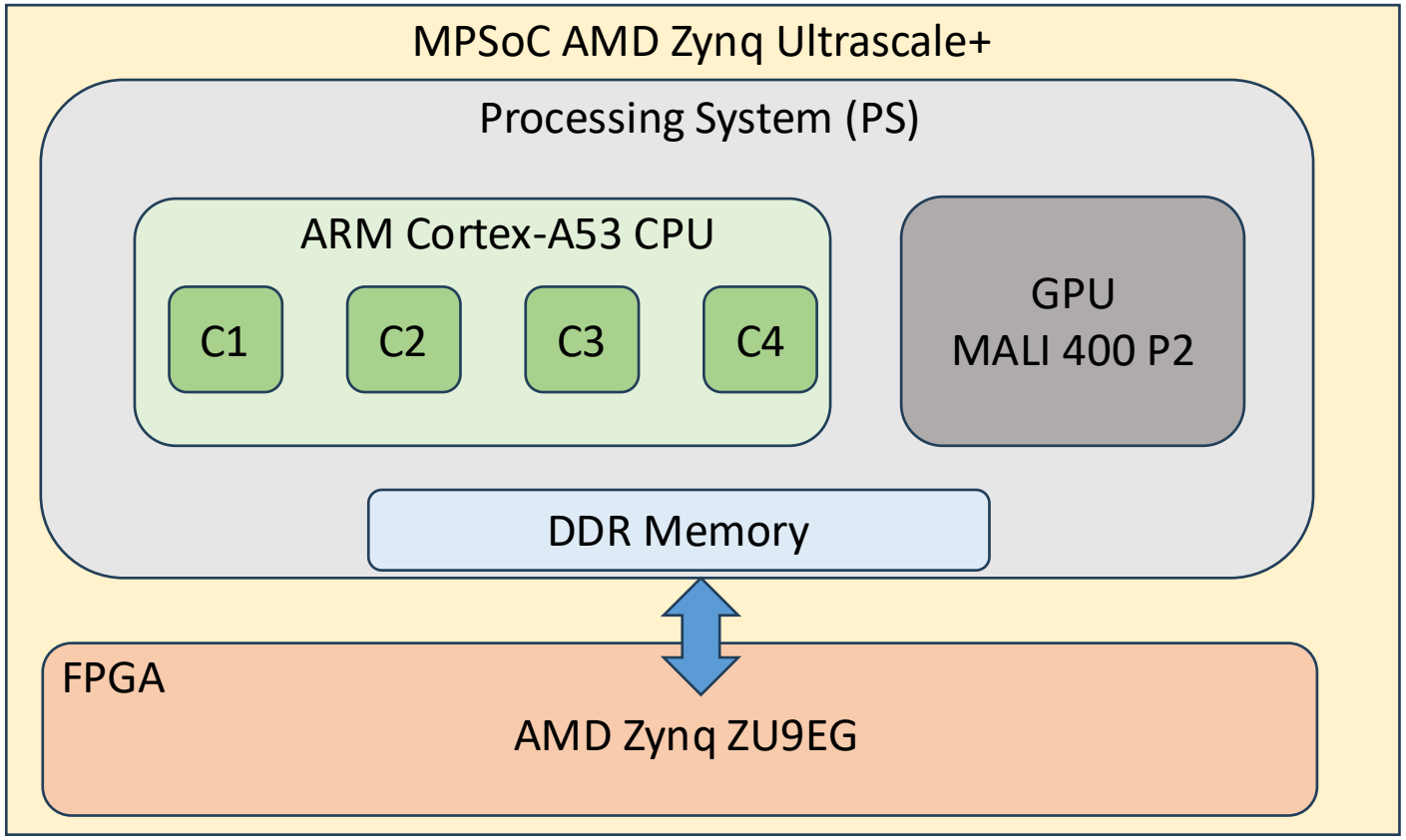
FPGAs

FPGA (Field-Programmable Gate Array) is a reconfigurable hardware device that can be programmed to implement custom digital circuits.



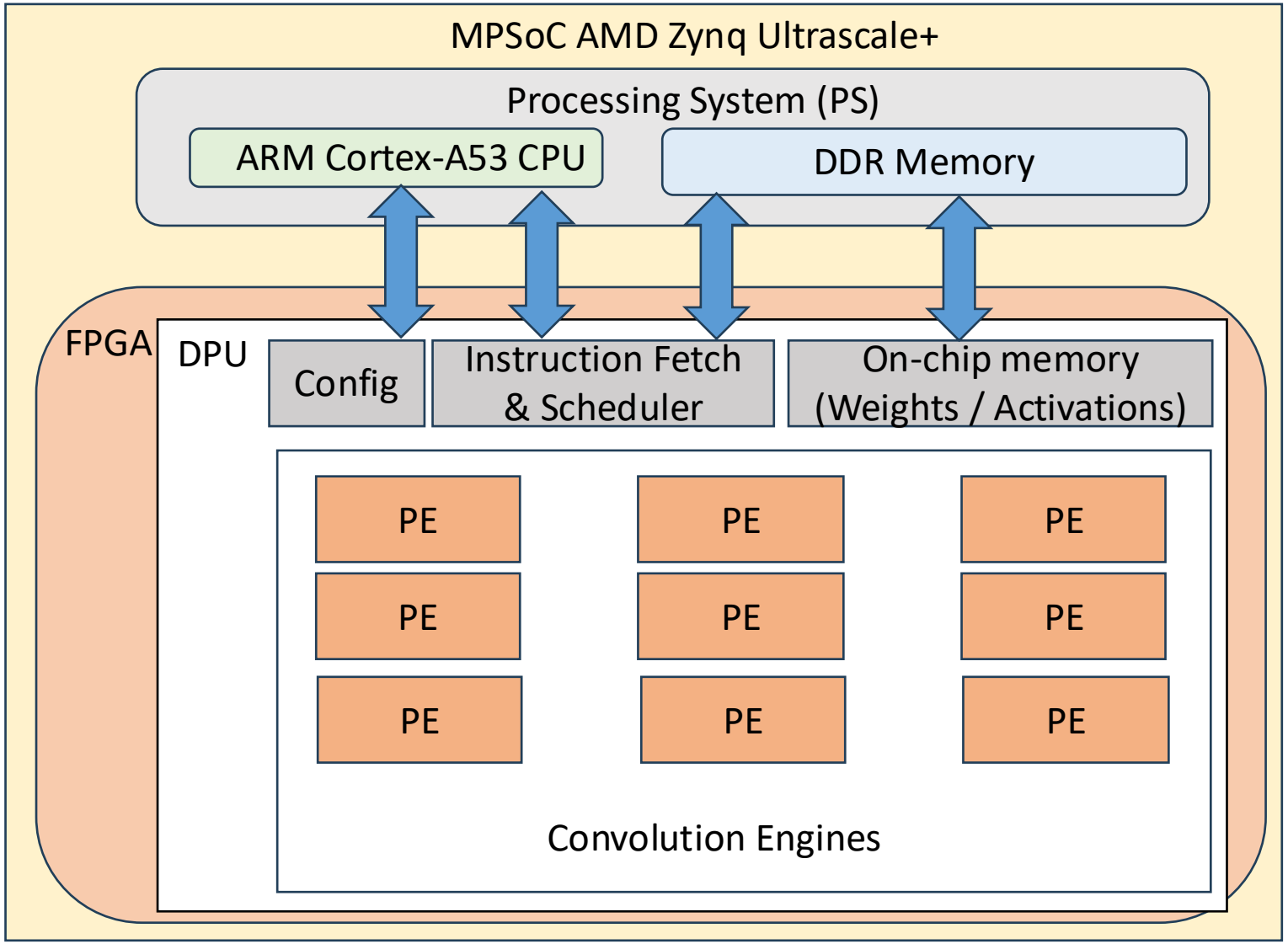
Experiment Testbed

We use the AMD ZCU102 (Zynq UltraScale+ MPSoC) which combines ARM CPUs (PS) + FPGA logic (PL) for parallel computations.



AMD DPU

- ✓**DPU (Deep Learning Processing Unit):** AMD Vitis AI accelerator IP for efficient DNN inference on FPGAs.
- ✓Supports key operations.
- ✓It can be configured in 8 sizes (e.g. 512, 4096 ops/cycle) and support multiple instances.
- ✓Vitis AI compiles pretrained models into DPU-executable instructions.



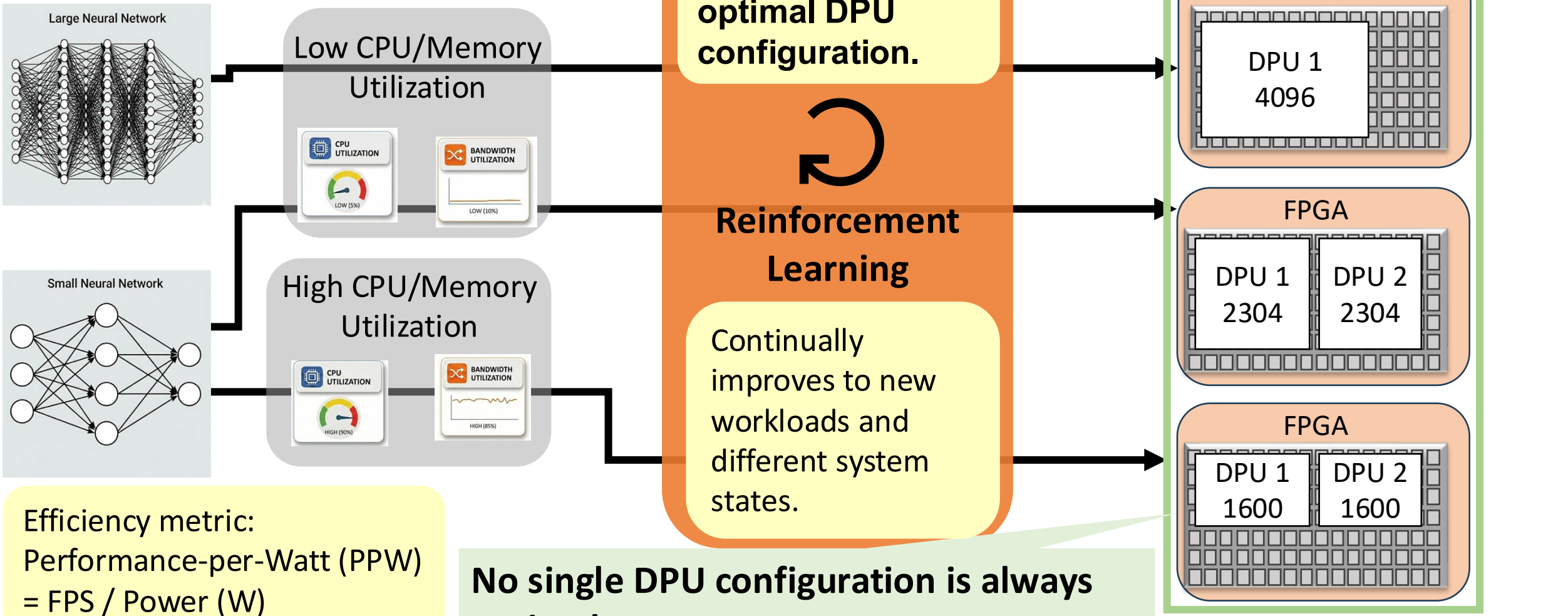
Objective & Motivation

ML Model (Workload)

System State

Framework

FPGA DPU Configurations

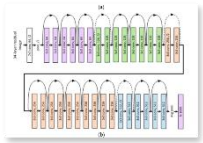


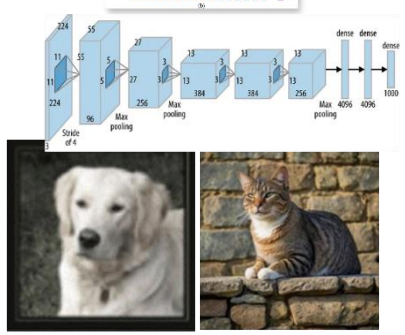
Efficiency metric:
Performance-per-Watt (PPW)
= FPS / Power (W)

No single DPU configuration is always optimal.

DPUConfig design

ML Workload

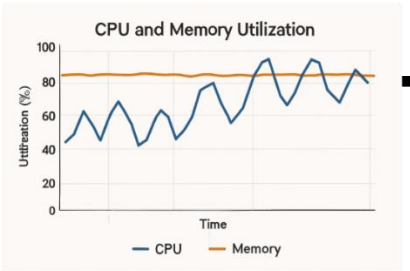


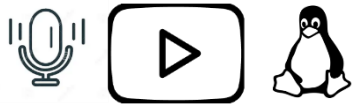


Model Static Features

(1) Observe State S

Dynamic System Features





Co-executing CPU apps

RL Agent Policy

(2) Select Action A

(3) Reconfigure FPGA and Load Instructions

(4) Execute Inference

MPSoC

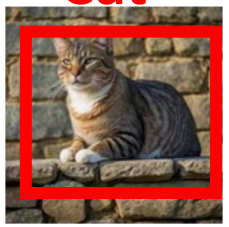
CPU

C1 C2 C3 C4

FPGA PL

DPU#1 DPU#2 DPU#3 DPU#4 DPU#5 DPU#6 DPU#7 DPU#8

Cat



(5) Compute Reward R

Telemetry: Latency, Power

RL Agent Design

RL Agent Policy

State S

Model Static Features

- # operations (GMACs)
- Weight data and memory transfers (bytes)
- # parameters

Dynamic System Features

- CPU Utilization (%)
- Memory Bandwidth Utilization (MB/s)
- Power consumption (W)

Actions S

1: DPU Size = 512, #Instances = 1
2: DPU Size = 512, #Instances = 4

….

25: DPU Size = 4096, #Instances = 2
26: DPU Size = 4096, #Instances = 3

Reward

- If **FPS < target → negative reward**
- Compute PPW = FPS / Power
- Reward = **PPW improvement** over a baseline based in system load and model characteristics.

Training/Test ML Models

Train/Test	Model	# GMAC Operations	Data I/O (MB)
Train	ResNet18	1.82	12.13
	ResNet50	4.10	38.94
	MobileNetV2	0.30	5.74
	DenseNet121	2.86	43.74
	InceptionV4	12.3	89.00
	RepVGG A0	1.52	11.84
	ResNext-50 32x4d	11.41	95.85
	YOLOv5s	8.26	159.80
Test	RegNetX 400MF	1.57	24.33
	InceptionV3	5.74	43.13
	ResNet152	11.54	76.52

Prepared 2 pruned versions for each model, with pruning ratios of 25% and 50%.

Total: 12 x 3 = 36 models

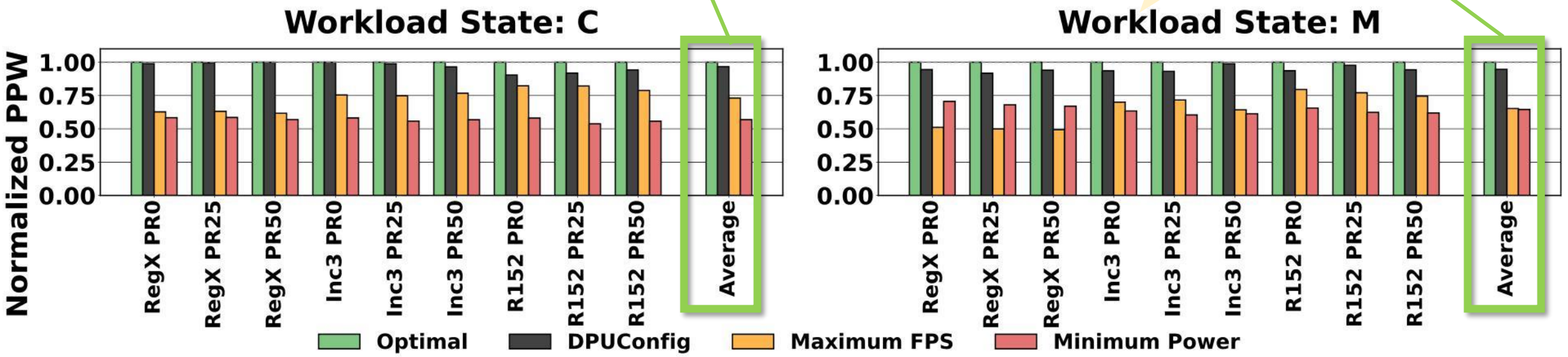
RL Accuracy

Compute-bound computations running on CPU

97 % of the optimal DPU configuration

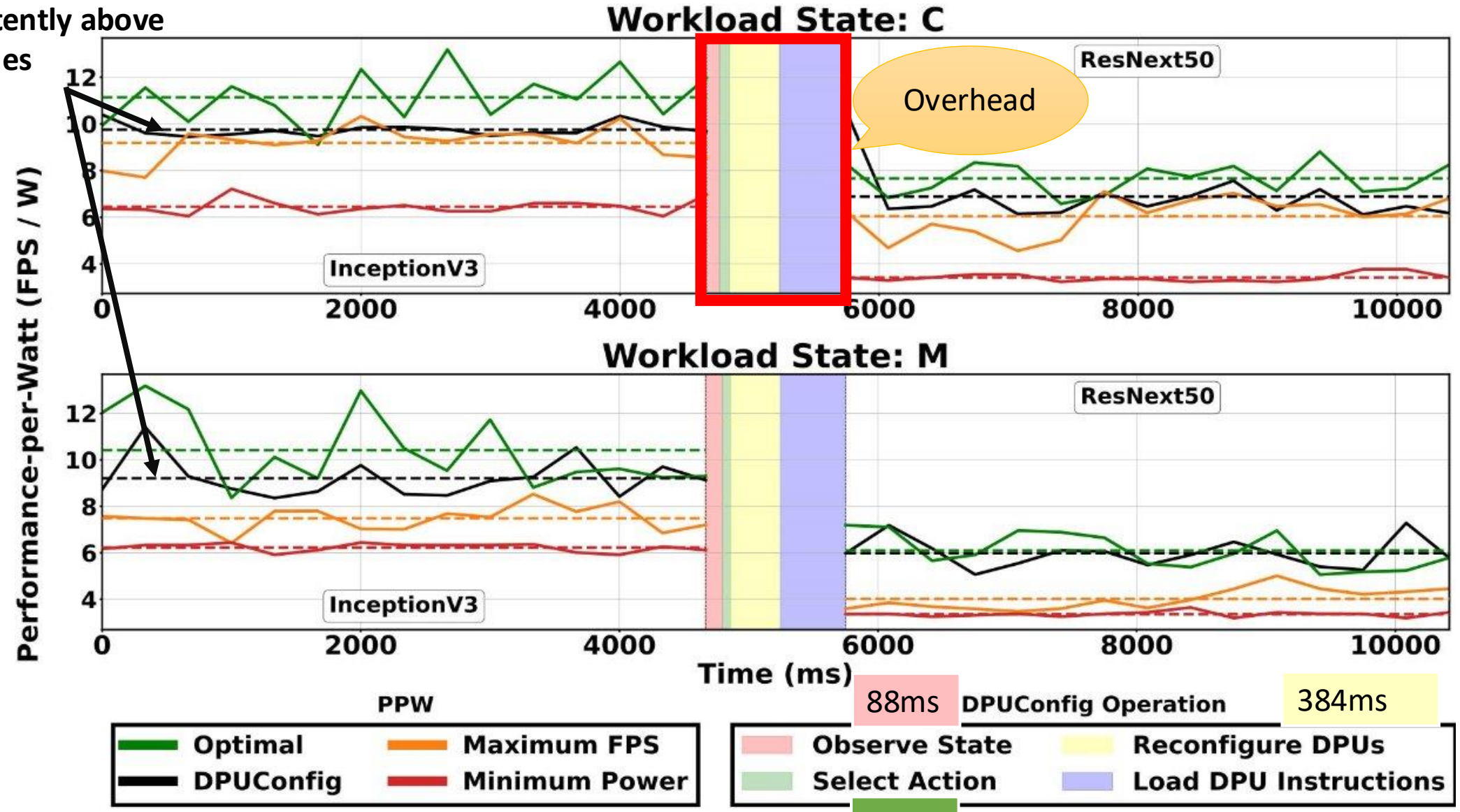
Memory-bound computations running on CPU

95 % of the optimal DPU configuration



DPUConfig Timeline

Consistently above baselines



Thank you!

Questions?

Acknowledgments

The research is funded by the European Union's Horizon Europe Programme under the "MLSysOps" Project (Grant Agreement No. 101092912).