Interference-Aware DNN Serving on Heterogeneous Processors in Edge Systems

Yeonjae Kim, Igjae Kim, Kwanghoon Choi (KAIST), Jeongseob Ahn (Korea Universit

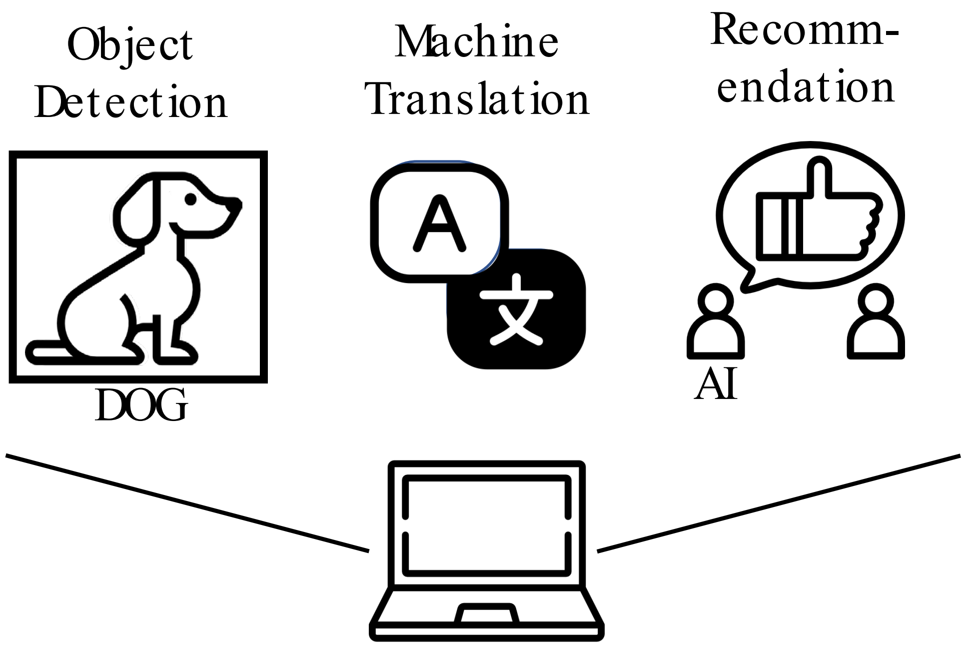
Jongse Park, Jaehyuk Huh (KAIST)

CASYS
Computer Architecture and Systems Lab

KAIST

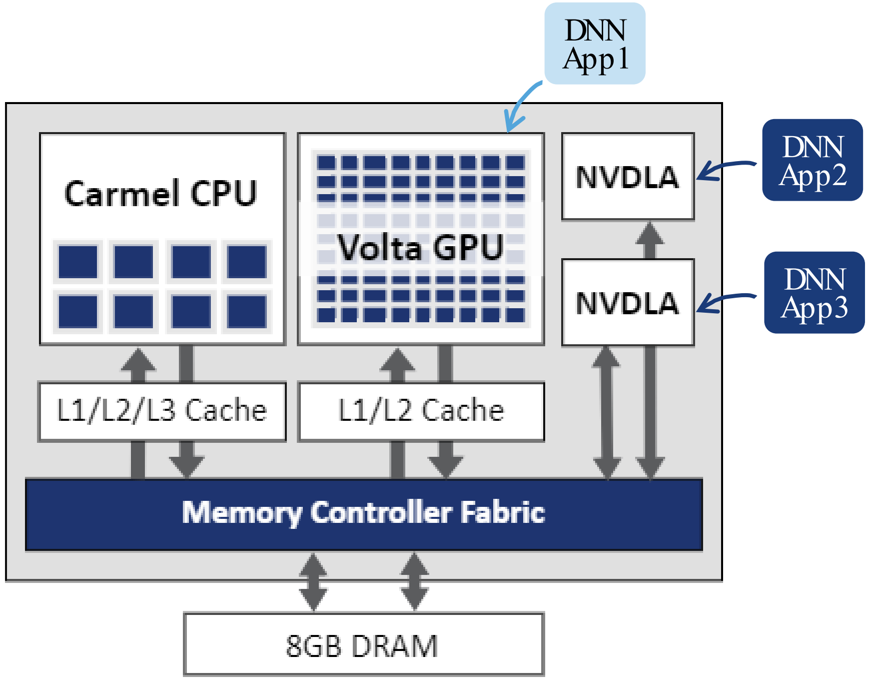
KOREA UNIVERSITY

Integrating heterogeneous devices for ML computing

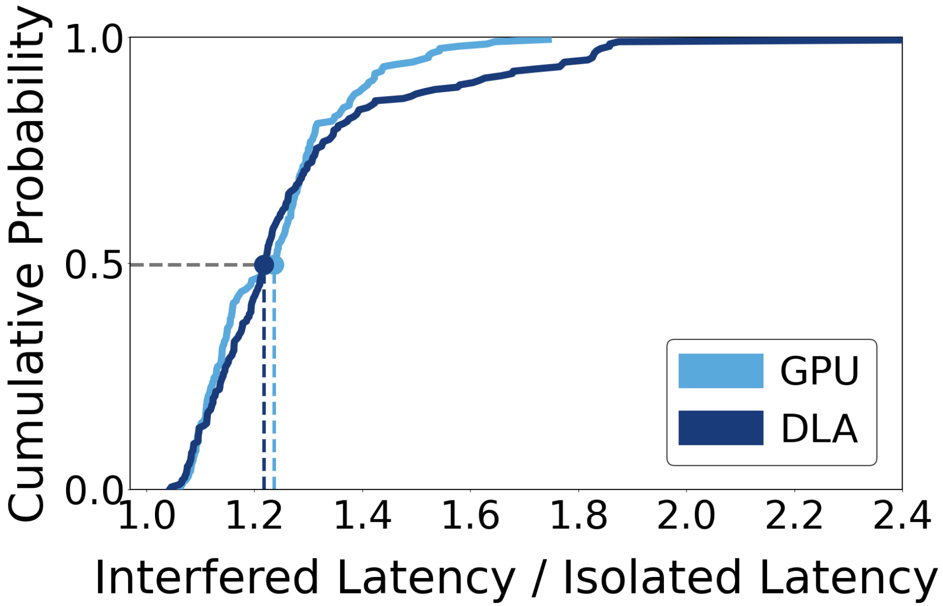


- Deep Learning Applications
  - Object detection
  - Speech cognition
  - Recommendation
- Heterogeneous Processors
  - Xavier, Apple M1, Samsung Exynos 9820

Performance Interference is not negligible



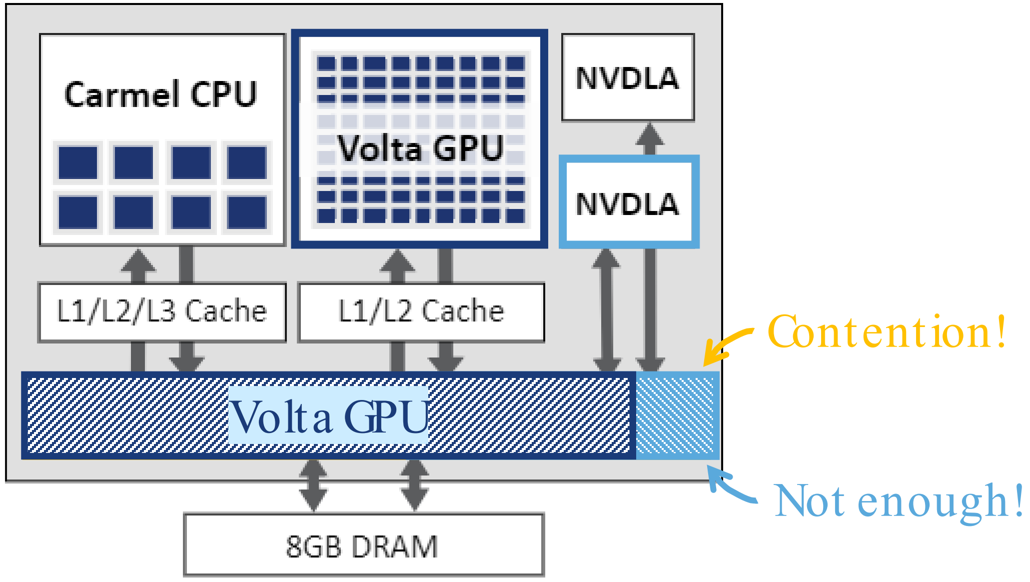
Example of heterogeneous edge device:
NVIDIA AGX Jetson Xavier



In 50% mappings,
GPU tasks: 24%↑ perfermonce degradation
DLA tasks: 22%↑ perfermonce degradation

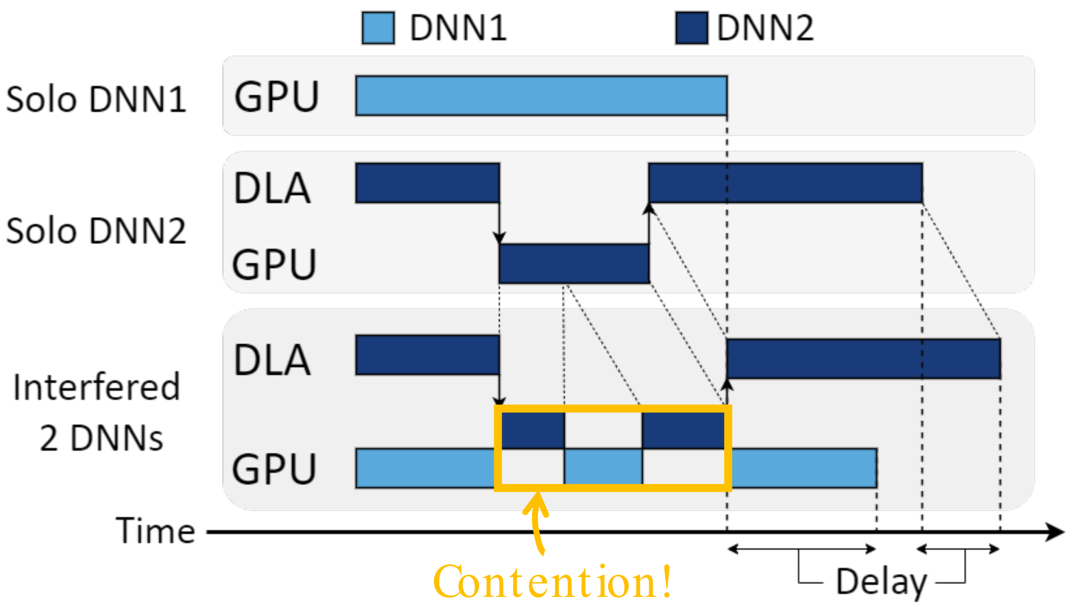
What are the sources of the interference?

“Memory bandwidth utilization”



Architecture of NVIDIA Xavier platform

“Limitation of DLA capability”



Related work

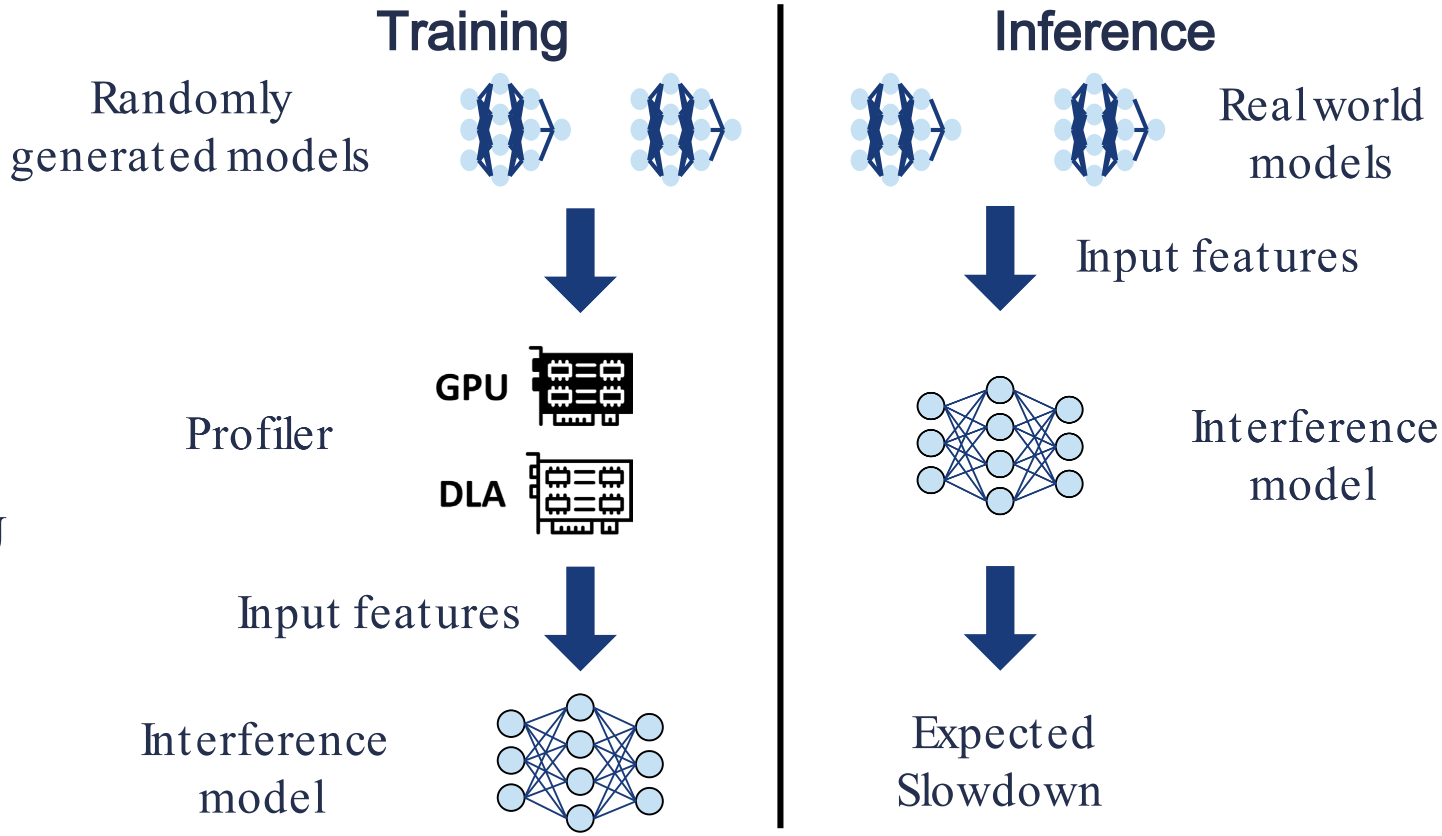
- Heterogeneous ML Schedulers
- **None** of these schedulers supports interference modeling

	MOSAIC[1]	SLO-PMAEL[2]	Gavel[3]	Our work
Heterogeneity support	✓	✓	✓	✓
Multi-model support	✗	✓	✓	✓
Customizable goal	✗	✗	✓	✓
Inference Tasks	✓	✓	✗	✓
Interference Modeling	✗	✗	✗	✓

[1] M. Han et al., Mosaic: Heterogeneity-, communication-, and constraint-aware model slicing and execution for accurate and efficient inference, PACT 2019.
[2] Seo et al., SLO-aware Inference Scheduler for Heterogeneous Processors in Edge Platforms, TACO 2021.
[3] D. Narayanan et al., Heterogeneity-aware cluster scheduling policies for deep learning workloads, OSDI 2020.

Interference Modeling

- For each co-located application,
  - Memory bandwidth utilization
  - GPU utilization
  - Average layer execution time on GPU
  - Latency slowdown by stress



- Consist of 4 submodels:

	Model1	Model2	Model3	Model4
Interfered processor	GPU	DLA	GPU	DLA0
Co-running Processor	DLA	GPU	DLA0, DLA1	DLA1, GPU
Accuracy	97.8%	94.3%	97.4%	94.0%

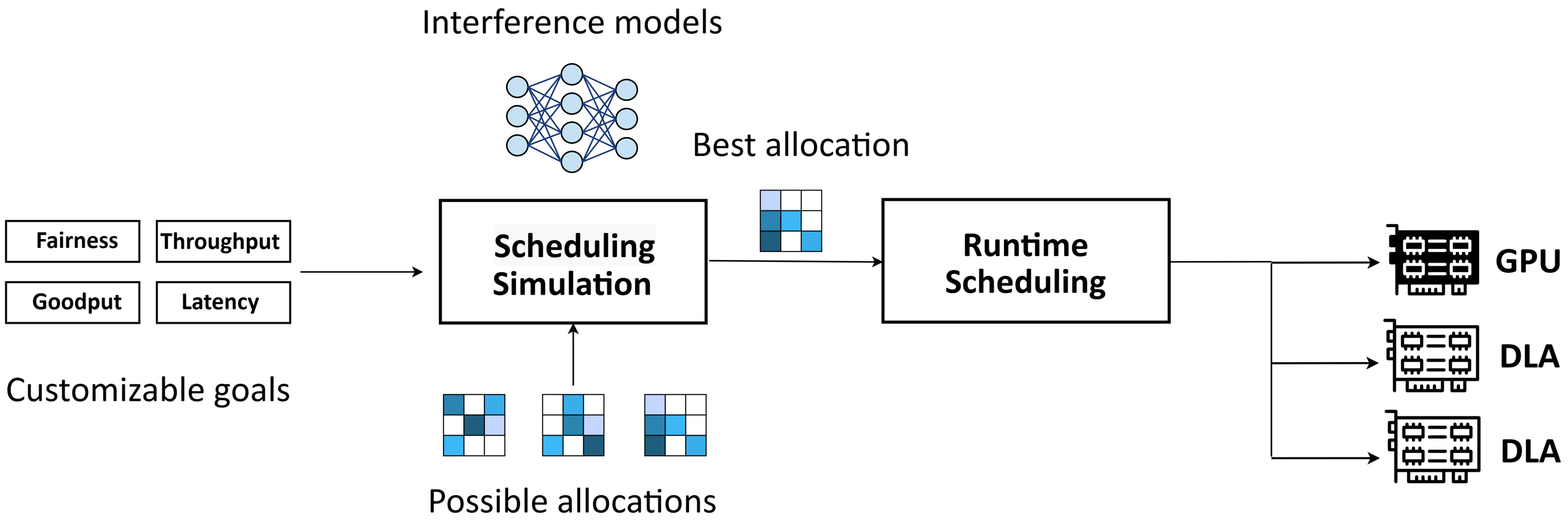
Interference Modeling

- Interference models are built with Multi-Layer Perceptron (MLP).
  - MLP models show the highest accuracy among several regression models.

Model	Model1	Model2	Model3	Model4
MLP	97.8%	**94.3%**	**97.4%**	**94.0%**
Kneighbor	97.7%	91.8%	95.6%	91.2%
Random forest	98.5%	92.7%	92.7%	88.5%
Decision Tree	97.5%	92.0%	87.4%	77.6%
SVR	92.8%	92.7%	94.6%	86.5%

Goal-Independent Scheduling Framework

- Overview

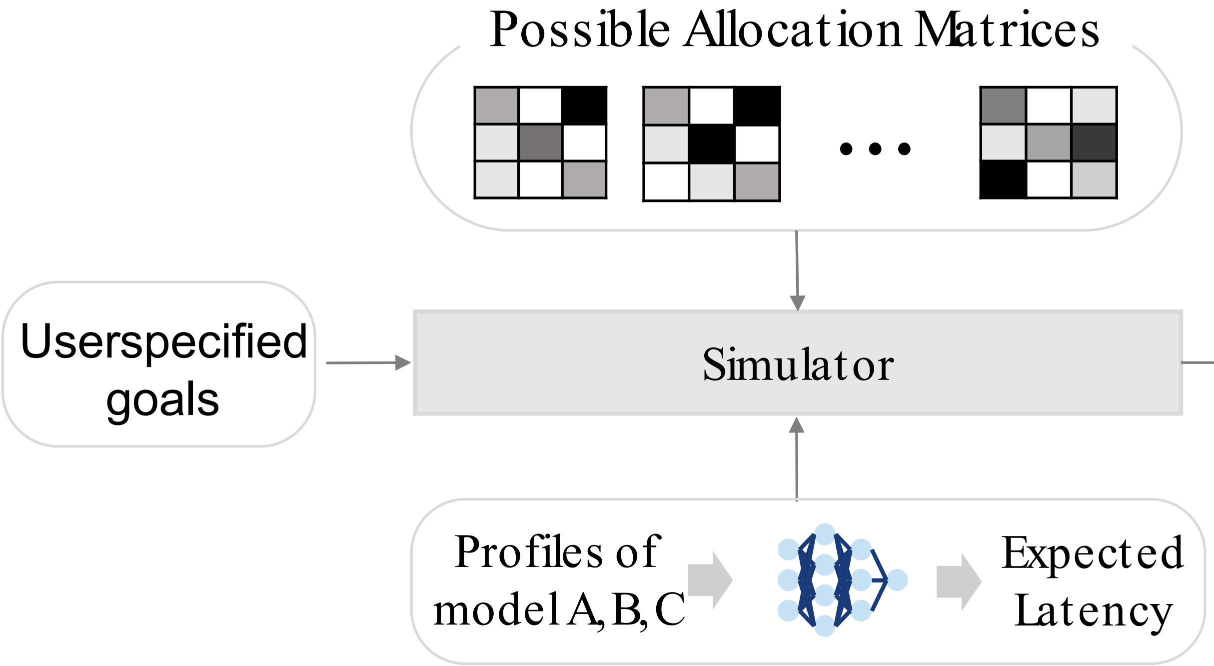


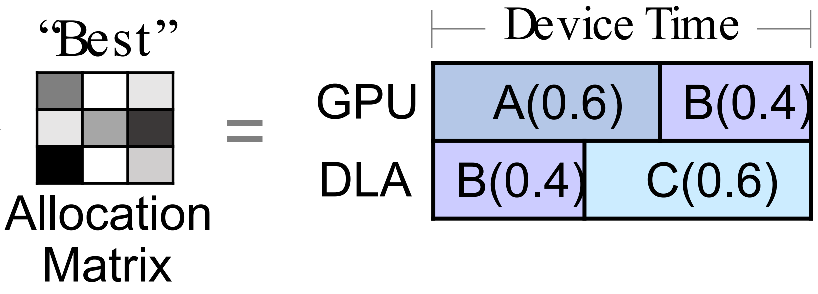
"Interference-Aware, Goal-Independent Scheduling Framework"

Goal-Independent Scheduling Framework

- Search for the best scheduling policy

With 1,000 samples there exists an allocation with 99% performance of the best allocation. [1]

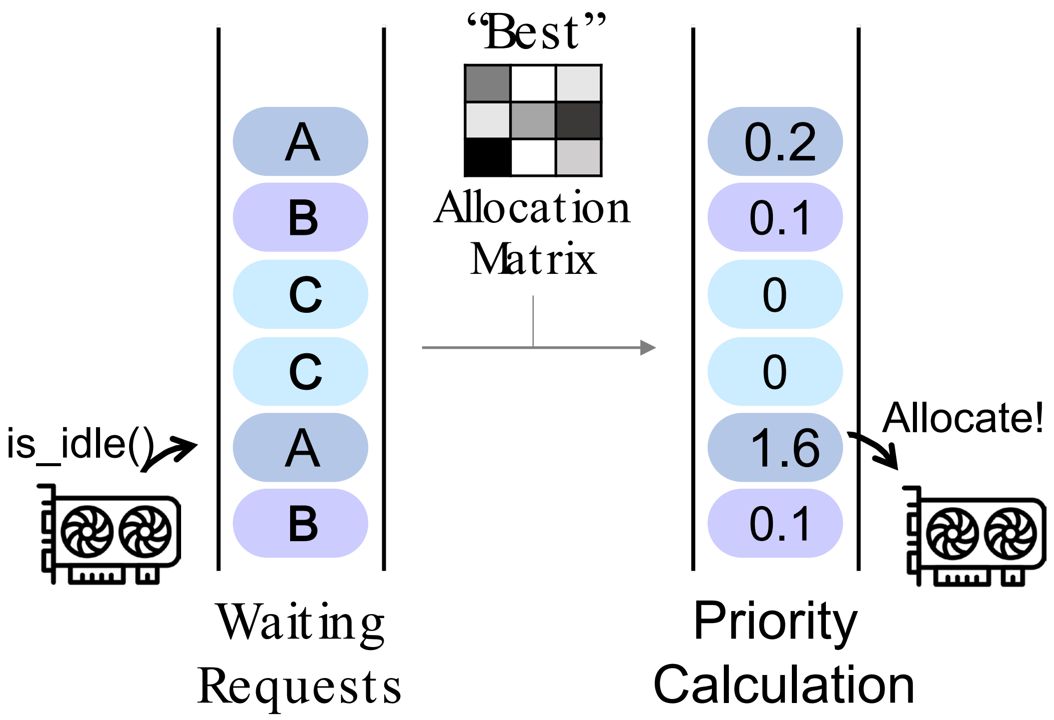




[1] P. Radojković et al. "Optimal Task Assignment in Multithreaded Processors: A Statistical Approach," ACM SIGPLAN Notices

Goal-Independent Scheduling Framework

- Priority-based scheduling



- Priority score

$$\textit{Priority Score} = \frac{\textit{Allocation Ratio}}{\textit{Consumed Allocation}}$$

- Routes requests to different devices depending on priority score

Evaluation Setting

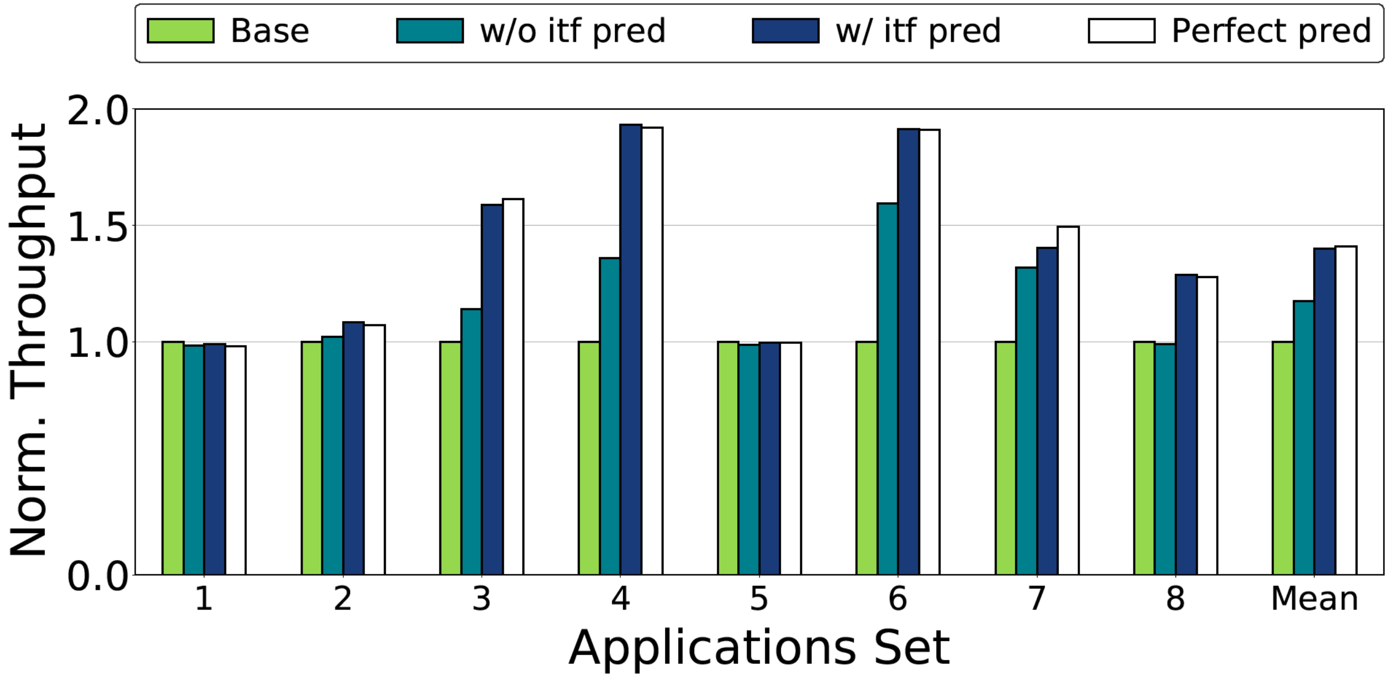
- Nvidia AGX Jetson Xavier

GPU	512-Core Volta GPU with Tensor Cores
DLA	(2x) NVDLA Engines

- TensorRT API
- Benchmarks
  - 40 (8x5) application scenarios with 14 DNN models from the torchvision
  - consist of 8 application sets, for each set we use 5 different request ratios
- Metrics
  - Goodput, Throughput with SLO 99%, Throughput, Fairness

Performance comparison

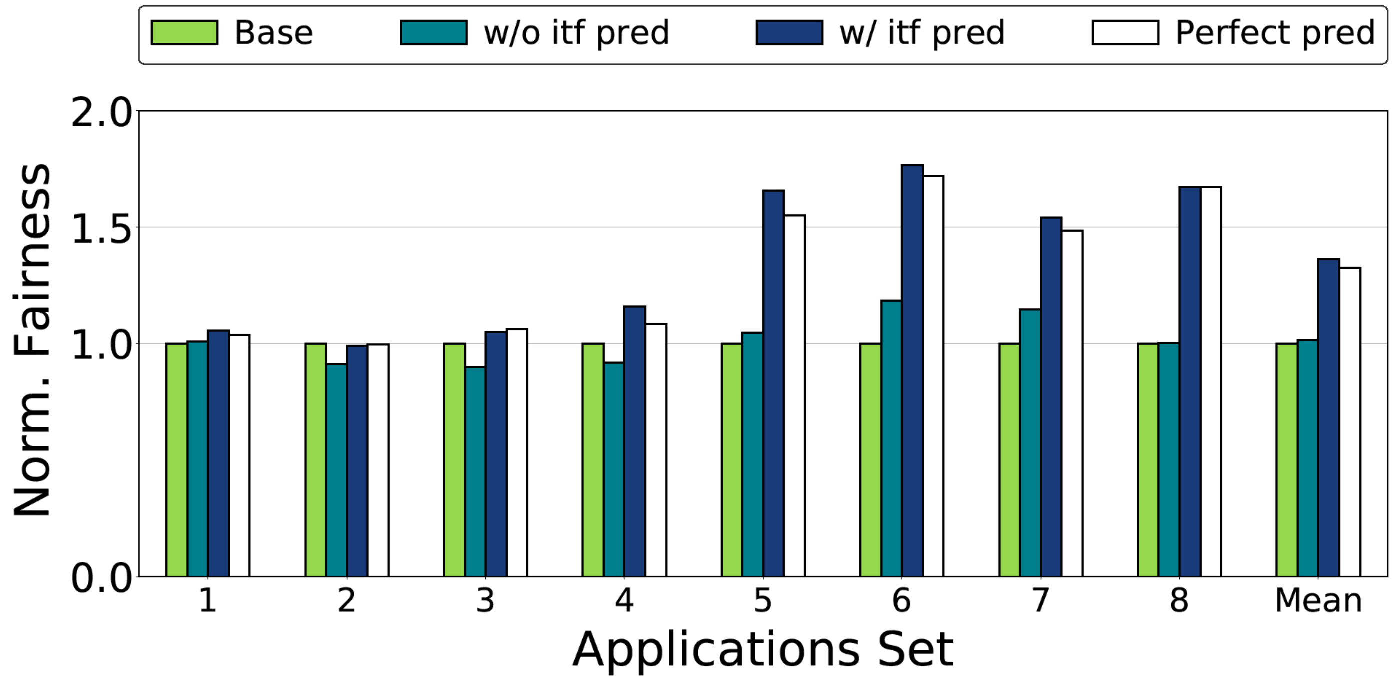
- Goodput: Throughput which satisfy target SLO.



Compared to w/o itf pred, w/ itf pred shows 18.1% average improvement
Compared to Base thpt, w/ itf pred shows 40.0% average improvement

Performance comparison

- Throughput under SLO satisfaction rate 99%



Compared to w/o itf pred, w/ itf pred shows 33% average improvement
Compared to Base thpt, w/ itf pred shows 36.1% average improvement

Conclusion

- Develop an MLP-based interference model, trained from randomly generated layers.
- Propose a goal-independent scheduling mechanism with sampled simulation.
- Achieve 40.0% higher goodput compared to baseline.

Thank you for listening!

Q&A

Computer Architecture and Systems Lab