The 3rd Real-Time And intelliGent Edge computing workshop (RAGE 2024)

When AI Meets Real-Time: Real-Time Scheduling Framework for Multi-DNN Inference

DGIST

Hoon Sung Chwa

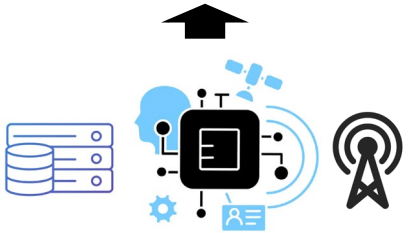
DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

Real-Time Computing Research

New Software

System Software

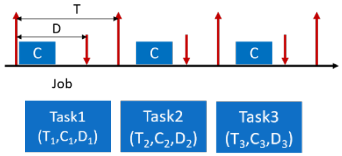
- ❑ Timing guarantee
- ❑ Resource efficiency



New Hardware

'70-'80

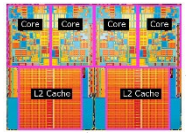
Single-core processor



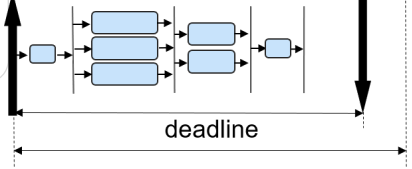
Periodic task model
(T,C,D): Period, Execution time, Deadline

'90-'00

Multi-core processor

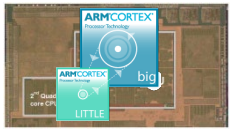


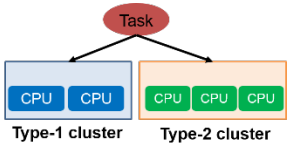
Parallel task model OpenMP



'10-'20

Heterogeneous processor



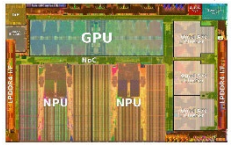


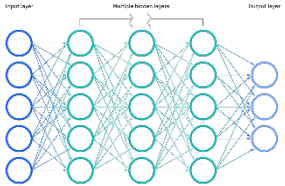
Fully-migrative model

'18-

Integrated SoC

Deep neural network





Real-Time Computing Research

System Software

- ❑ Timing guarantee
- ❑ Resource efficiency

Q. How to extend real-time computing techniques to support new software & hardware?

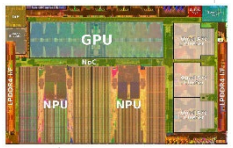
Opportunities

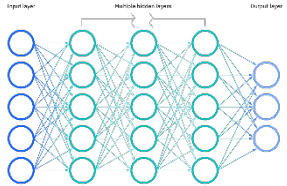
- Diverse optimization techniques
- Deterministic execution behavior

Challenges

- AI accelerators
- Memory-intensive workloads
- Timing guarantee + Accuracy

Deep neural network

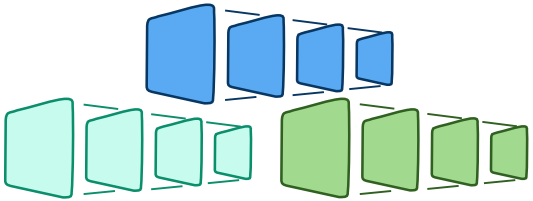




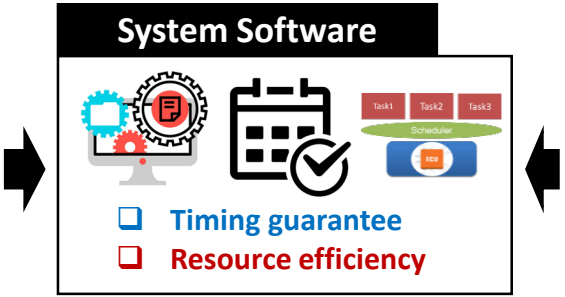
Integrated SoC

’18-

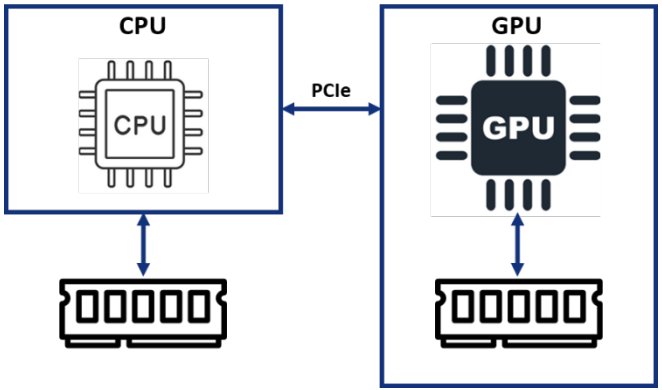
Today's Talk



DNN models



Real-Time Scheduling Framework

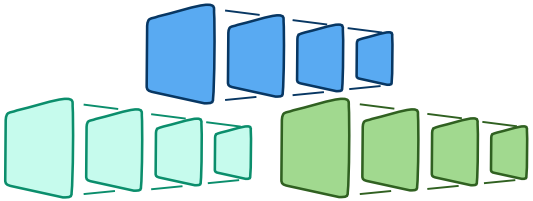


Modern AI Computing Platform

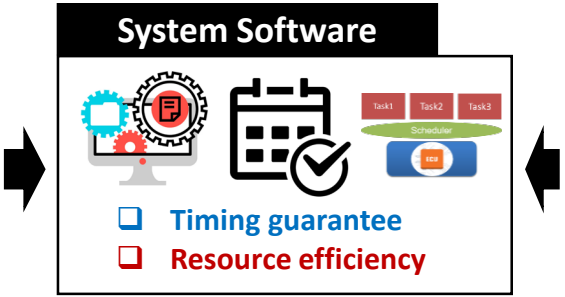
1 **CPU-GPU Performance Imbalance**
[LaLaRAND, RTSS 2021]

2 **GPU Memory Capacity Bottleneck**
[RT-Swap, RTAS 2024]

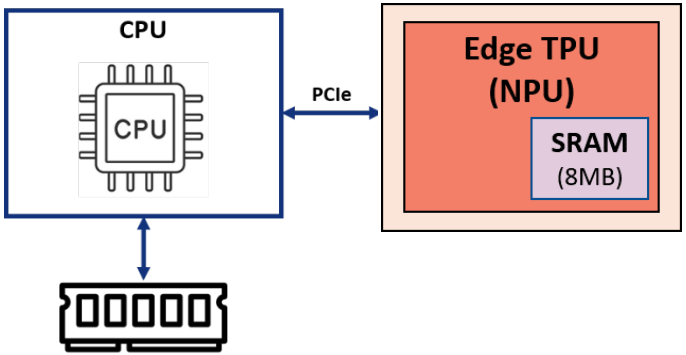
Today's Talk



DNN models



Real-Time Scheduling Framework



Modern AI Computing Platform

1 **CPU-GPU Performance Imbalance**
[LaLaRAND, RTSS 2021]

2 **GPU Memory Capacity Bottleneck**
[RT-Swap, RTAS 2024]

3 **SRAM Caching**
[SPET, DAC 2023]

LaLaRAND: Flexible Layer-by-Layer CPU/GPU Scheduling for Real-Time DNN Tasks

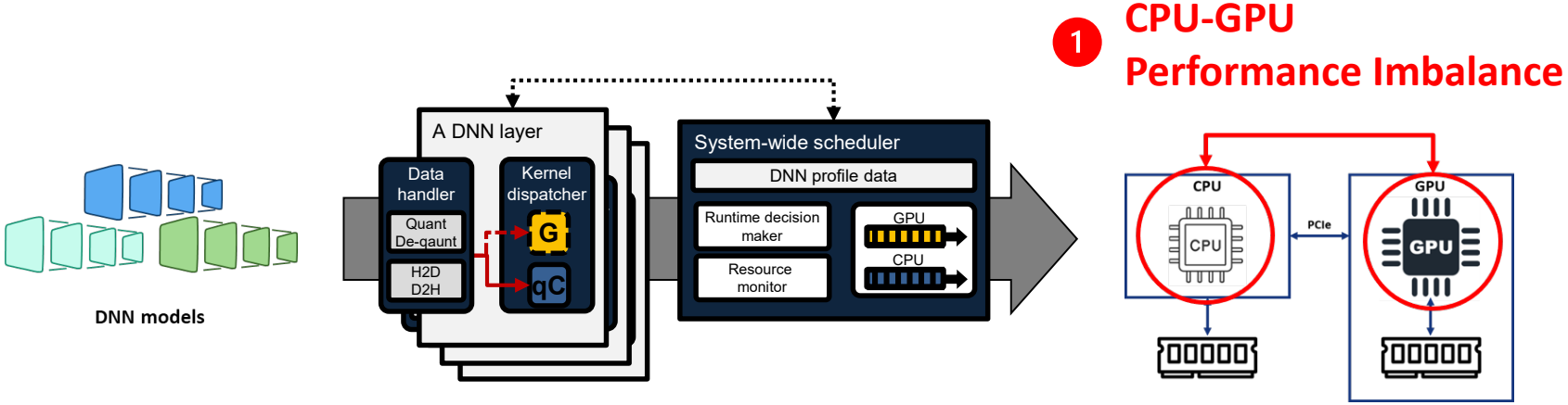
Woosung Kang[*], Kilho Lee[†], Jinkyu Lee[‡], Insik Shin[§], and Hoon Sung Chwa[*]

42nd IEEE Real-Time Systems Symposium (RTSS 2021)

LaLaRAND: Flexible Layer-by-Layer CPU/GPU Scheduling for Real-Time DNN Tasks

Woosung Kang*, Kilho Lee†, Jinkyu Lee‡, Insik Shin§, and Hoon Sung Chwa*

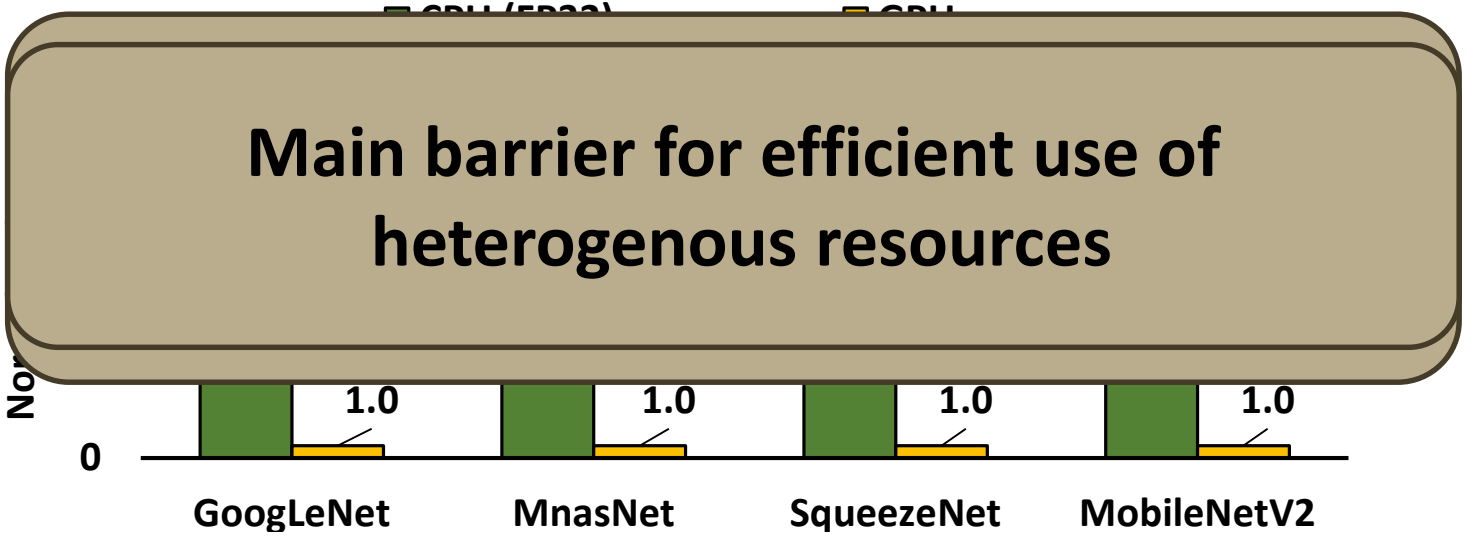
42nd IEEE Real-Time Systems Symposium (RTSS 2021)



Layer-level CPU/GPU Scheduling Framework with Quantization

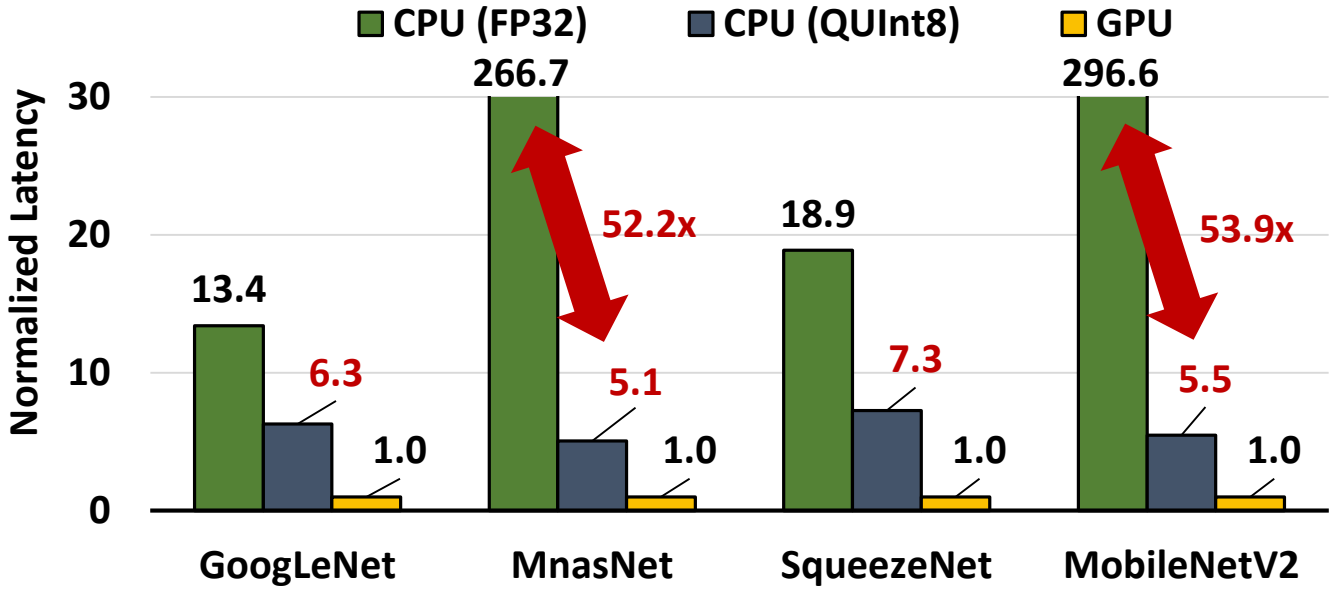
CPU/GPU Performance Imbalance

- DNNs **prefer GPU resource** due to the nature of **parallel execution characteristics**
- DNNs execute up to **296.6x slower on CPU** resource than GPU resource



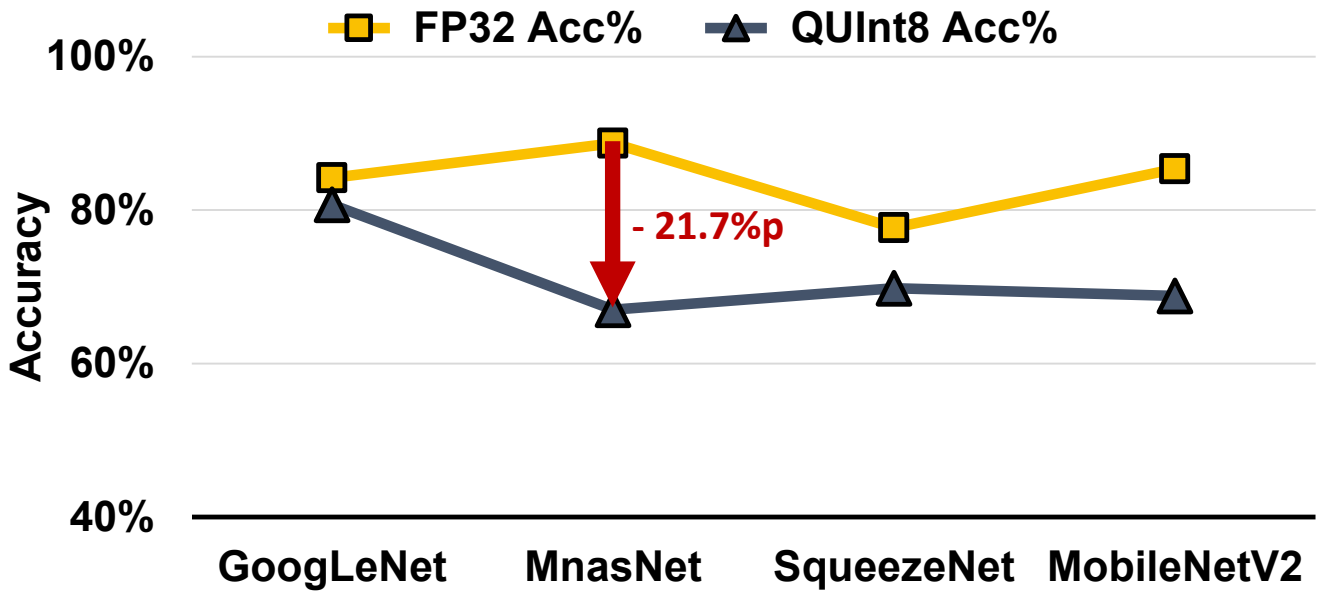
CPU-friendly 8-bit Quantization

- 8-bit quantization leads a great **reduction in execution time**
  - CPU execution times become **comparable with GPU execution times**



CPU-friendly 8-bit Quantization

- 8-bit quantization leads a great **reduction in execution time**
  - CPU execution times become **comparable with GPU execution times**
- For trade-off, 8-bit quantization causes an **accuracy drop**



Research Goals

- By utilizing quantization applied CPU along with GPU,

G1. Providing timing guarantees for real-time DNN tasks and improving schedulability performance

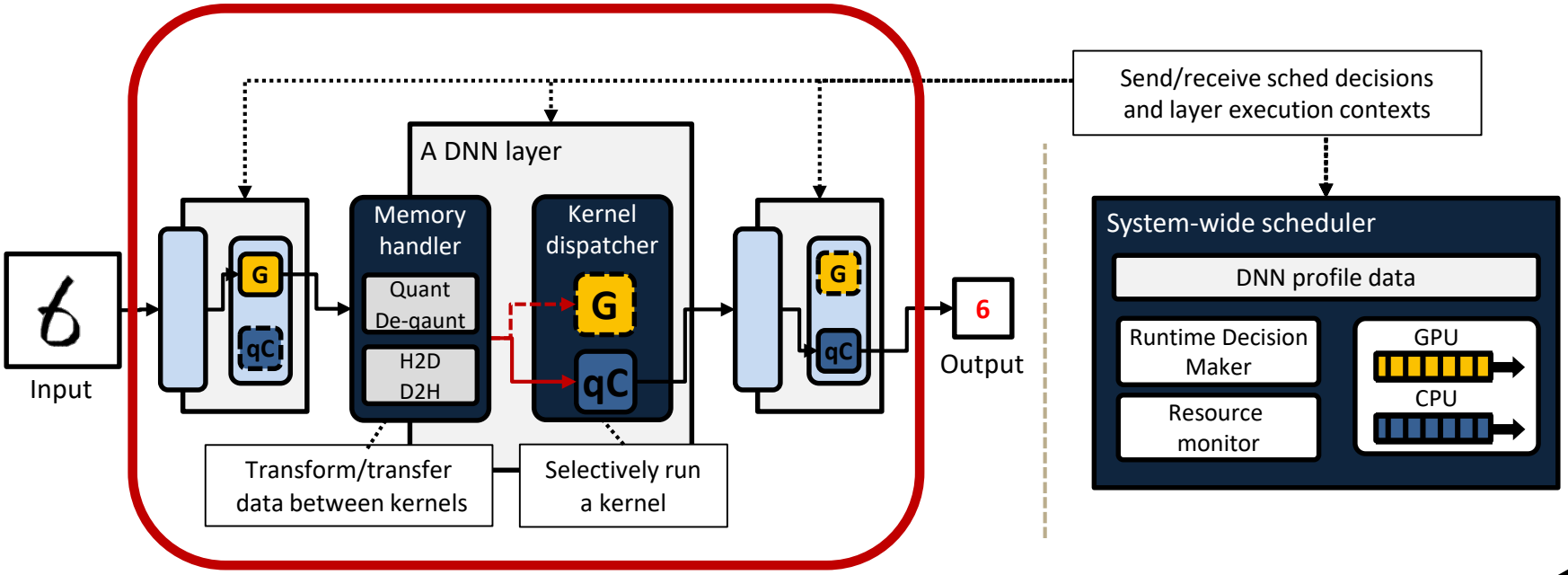
⇒ Q1. How to utilize heterogeneous resources?

G2. Minimizing the overall accuracy loss due to CPU-friendly quantization without compromising timing guarantees

⇒ Q2. How to utilize quantization?

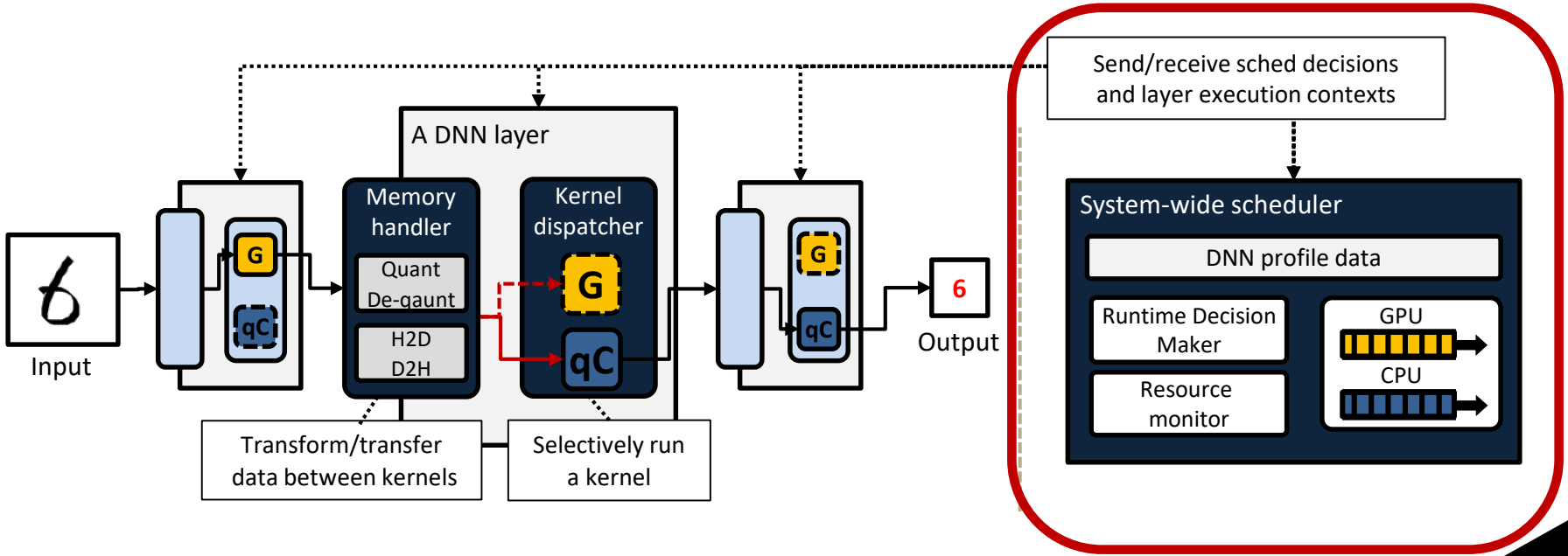
LaLaRAND Overview

- **LaLaRAND** : **La**yer-by-**La**yer **R**esource **A**llocation to **N**-**D**NNs
  - DNN layer for **layer-level CPU/GPU resource allocation**



LaLaRAND Overview

- **LaLaRAND** : **La**yer-by-**La**yer **R**esource **A**llocation to **N**-**D**NNs
  - DNN layer for **layer-level CPU/GPU resource allocation**
  - System-wide scheduler for **layer-level scheduling**

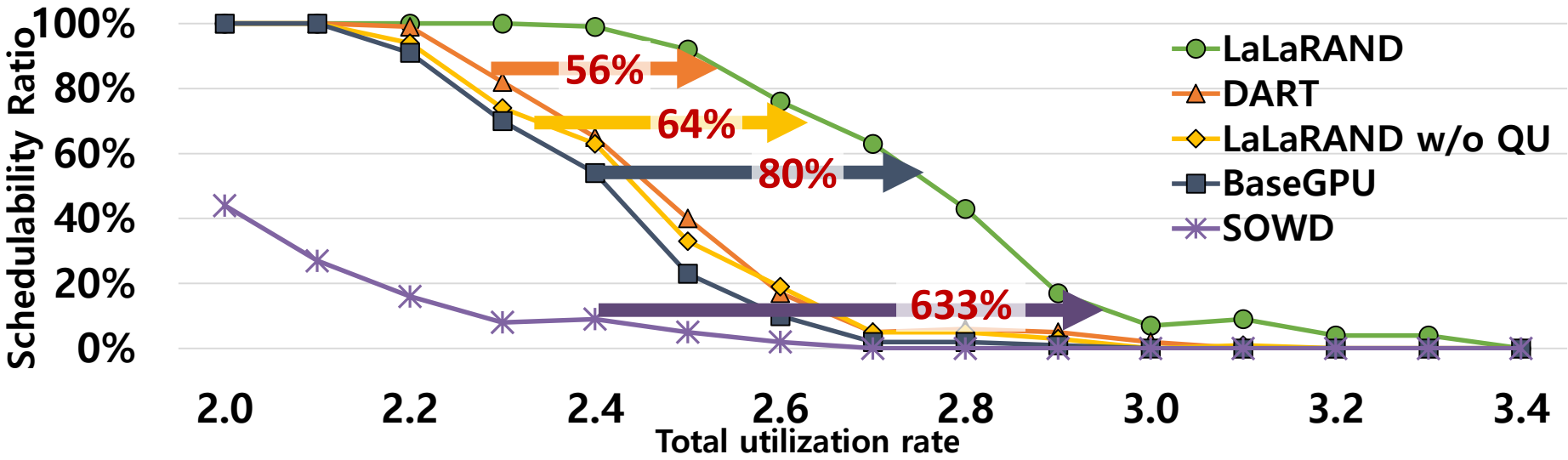


Evaluation

- **Schedulability Ratio**
  - **Schedulability ratio with varying total utilization rate**

configuration

Total utilization rate = $\sum_i(\sum_j(C^C_{i,j} + C^G_{i,j})/2)/T_i$
Tasks in 1 taskset = 10
Priority assignment = rate-monotonic (RM)



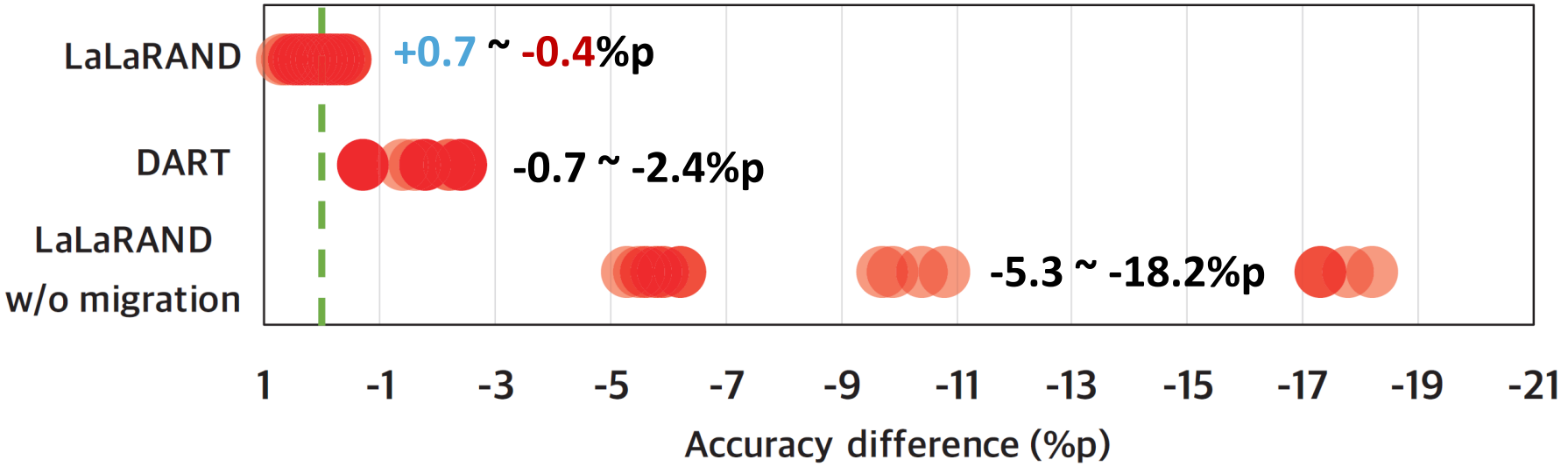
LaLaRAND exhibits high capability in finding schedulable task set

Evaluation

- **Accuracy**
  - **Distribution of accuracy difference**

configuration

Total utilization rate = 2.5 ~ 3.0
Number of tasks = 5
Dataset = ImageNet 2014 1000 images



LaLaRAND minimizes the accuracy drop or even improves the accuracy

RT-Swap: Addressing GPU Memory Bottlenecks for Real-Time Multi-DNN Inference

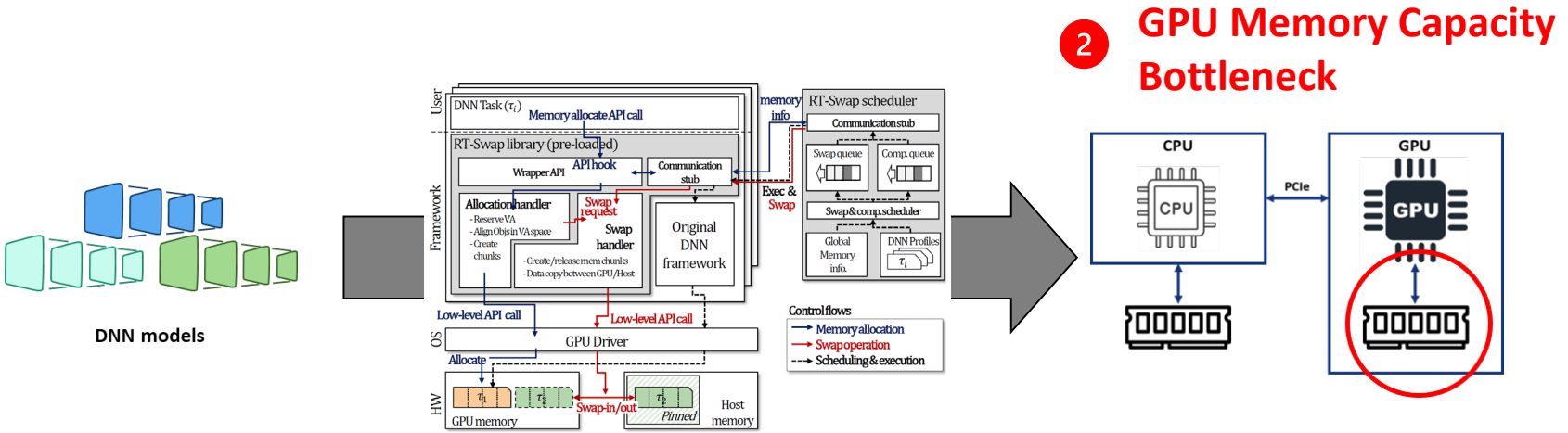
Woosung Kang[1], Jinkyu Lee[2], Youngmoon Lee[3], Sangeun Oh[4], Kilho Lee[5], and Hoon Sung Chwa[1]

30th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2024)

RT-Swap: Addressing GPU Memory Bottlenecks for Real-Time Multi-DNN Inference

Woosung Kang[1], Jinkyu Lee[2], Youngmoon Lee[3], Sangeun Oh[4], Kilho Lee[5], and Hoon Sung Chwa[1]

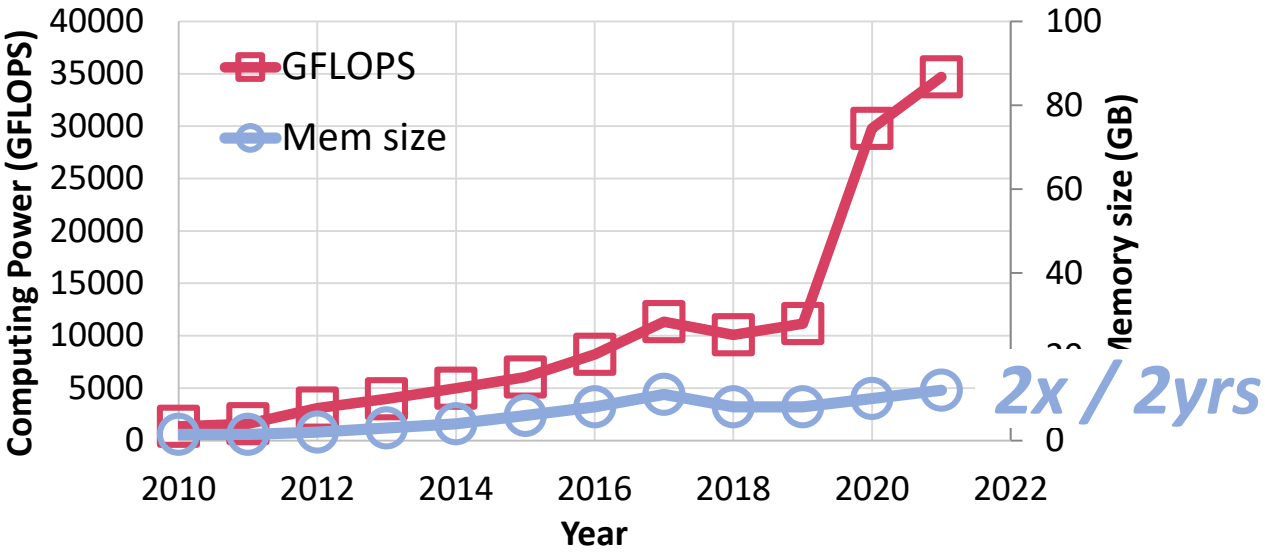
30th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2024)

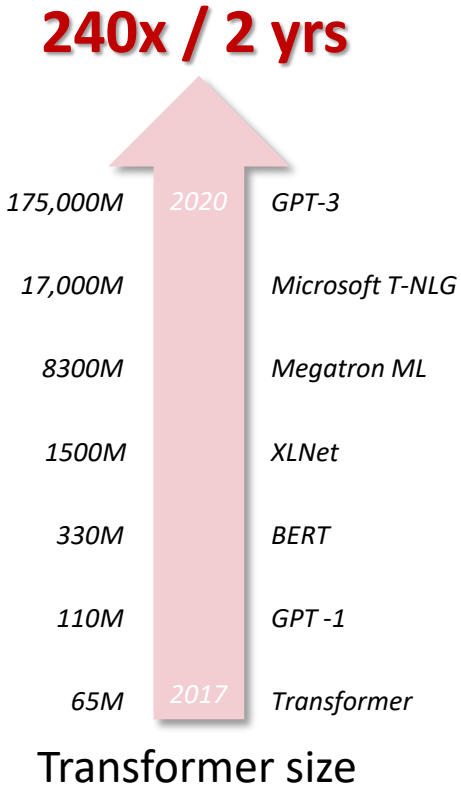


GPU Memory Management Framework

Increasing Memory Demands of AI Models

- ***Ever-increasing memory demands of AI models***
  - *The growth in GPU memory capacity lags behind the rising memory requirements of AI models.*





New Bottleneck, GPU Memory Capacity

- ***Memory issues are waiting for you just around the corner***
  - *For instance, GPT-3 requires 175B parameters,* ***700GB***
  - *Most high-end GPU, NVIDIA A100, has* ***80GB GPU memory***

GPT3

175B parameters

Roughly ***700GB***

X 9

NVIDIA A100, **80GB**

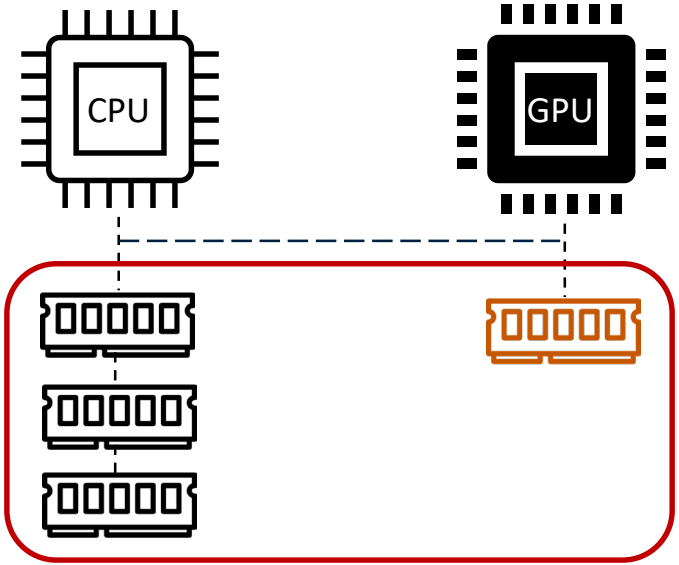
$17,000 for single A100

$153,000

Overcome the Limited GPU Memory Capacity

- CPU memory has a relatively large capacity and is easily extensible
- ***Co-utilizing CPU memory along with GPU memory***

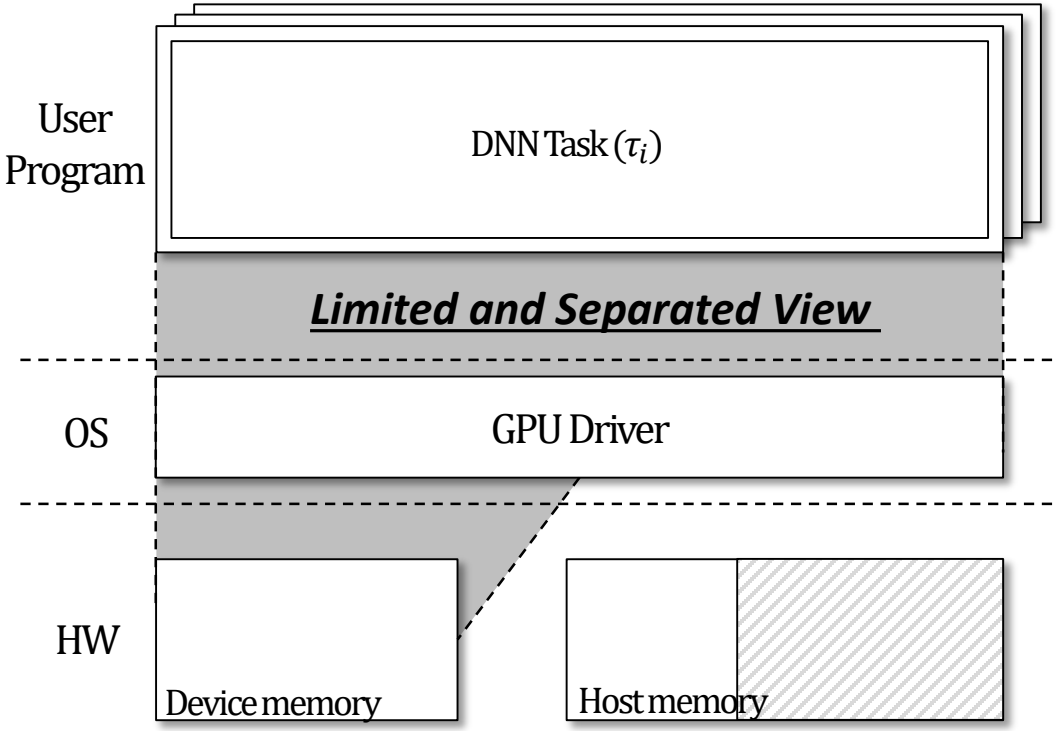
Transfer overhead exist!



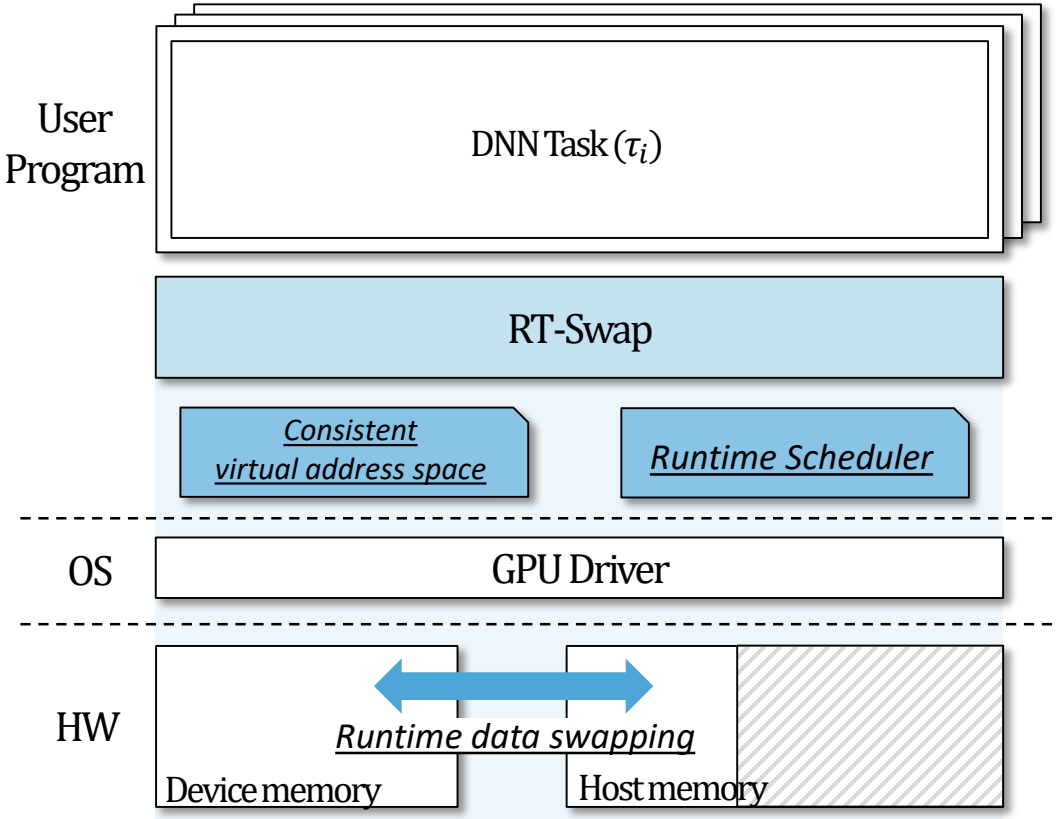
Easily extensible!

Co-utilize both memories by **transferring data between them**

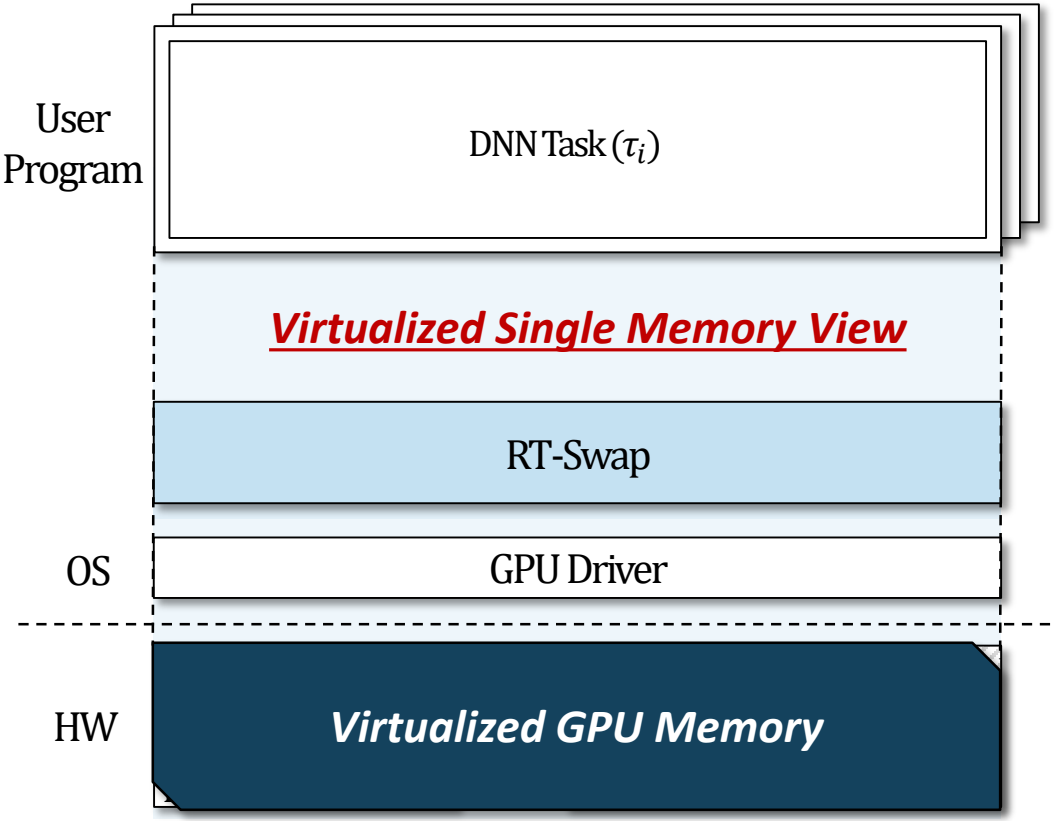
RT-Swap Overview



RT-Swap Overview

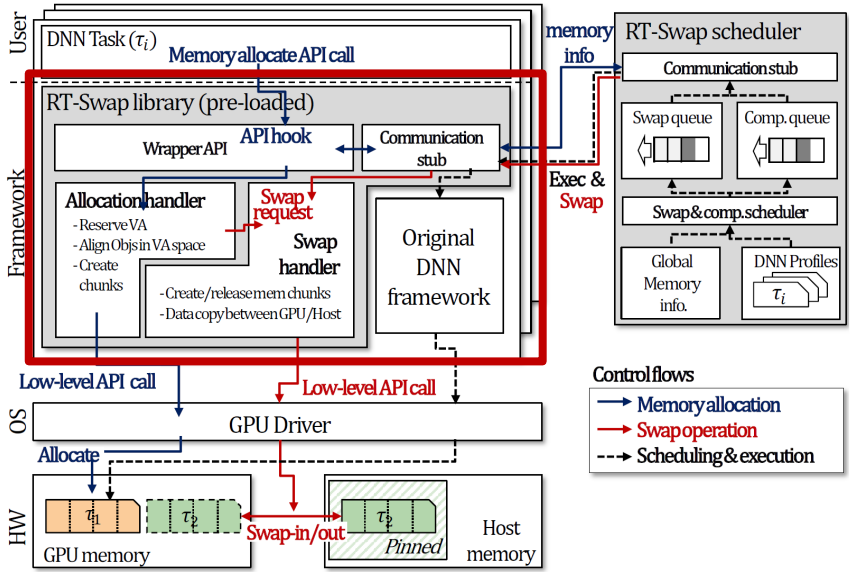


RT-Swap Overview

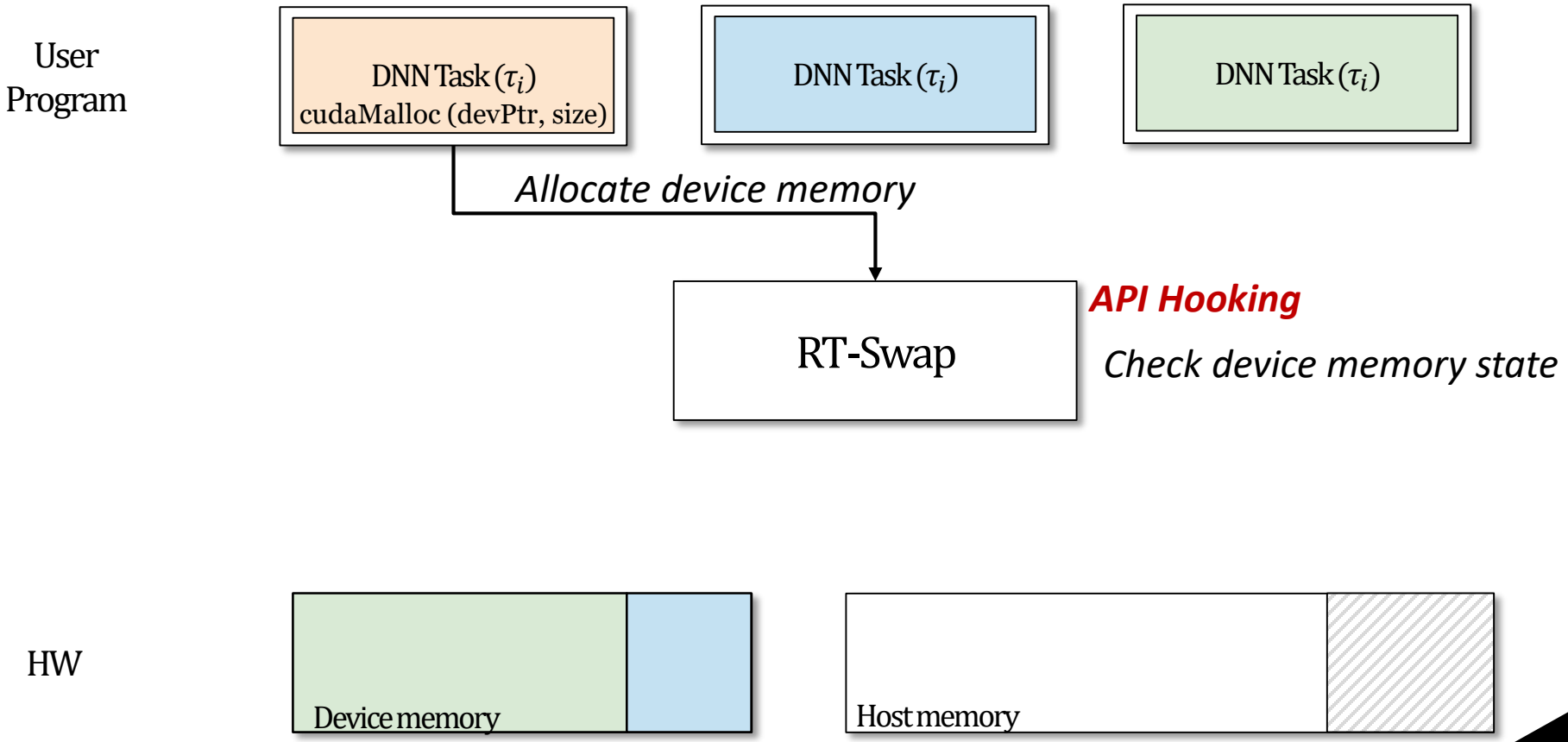


RT-Swap: Beyond the Memory Wall of GPU

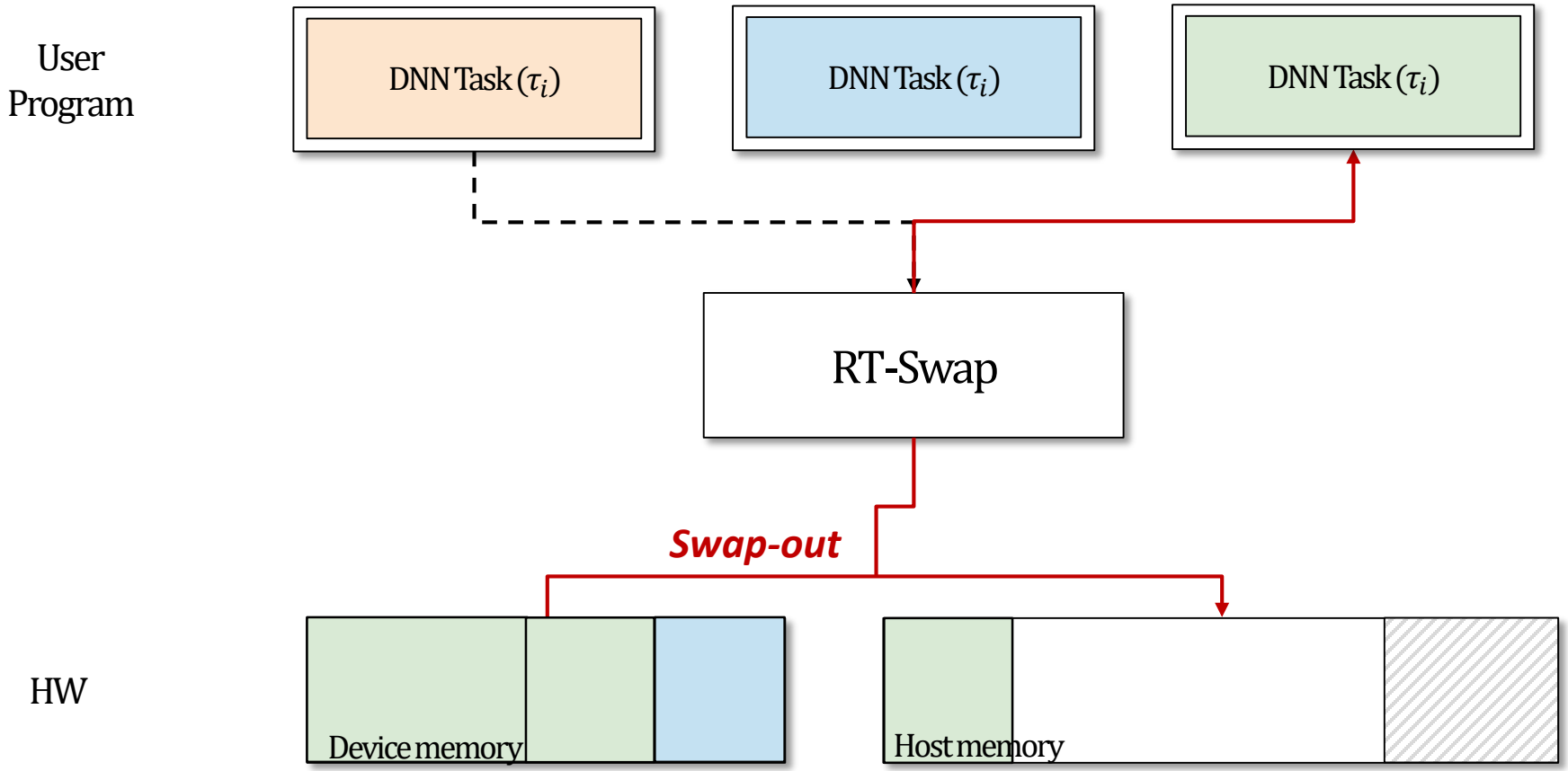
- A runtime memory management framework
  - **RT-Swap library** – Transparent swapping operations
    - Swap handler and VMM (Virtual Memory Management) handler



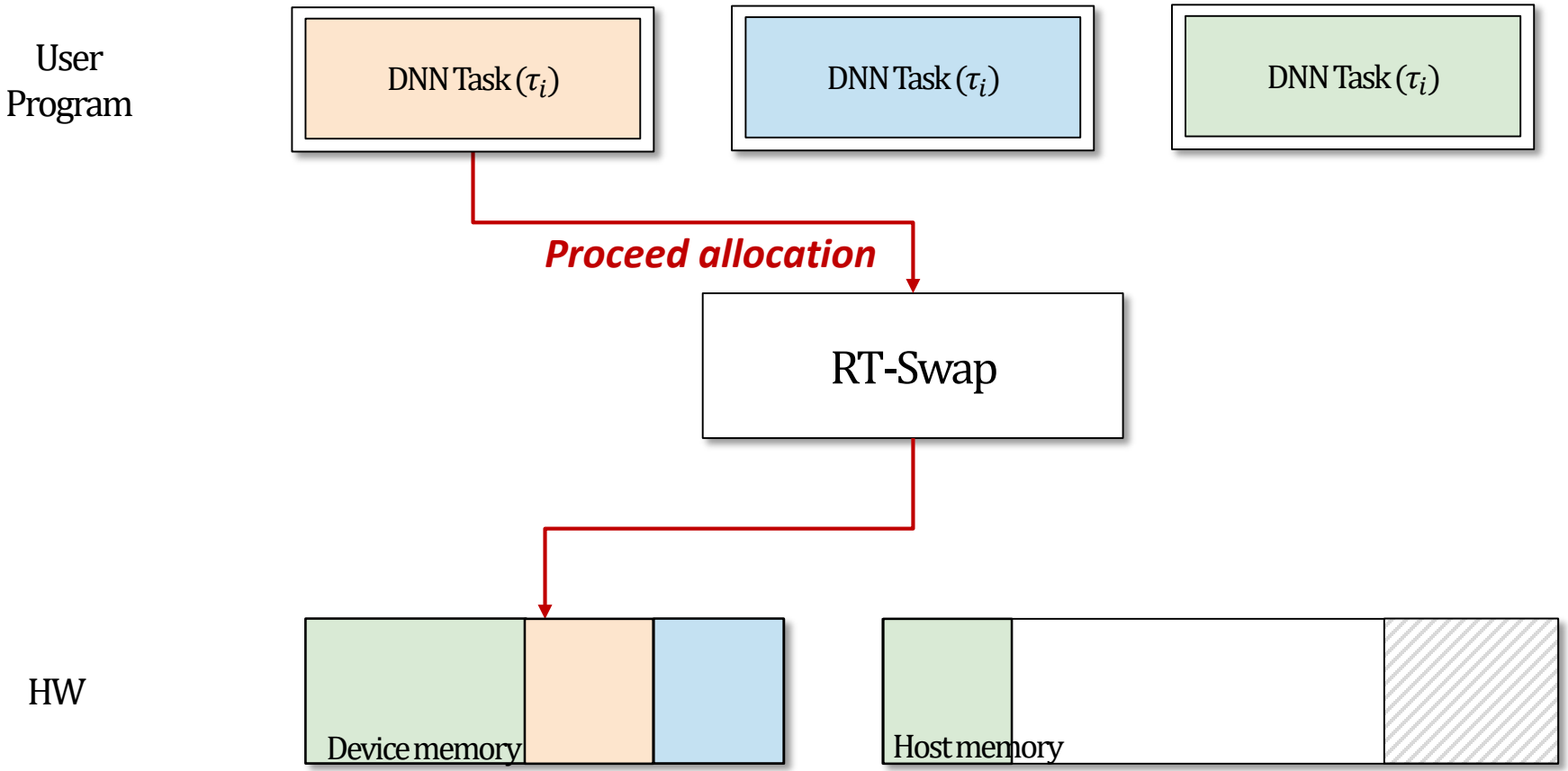
Runtime Memory Swapping



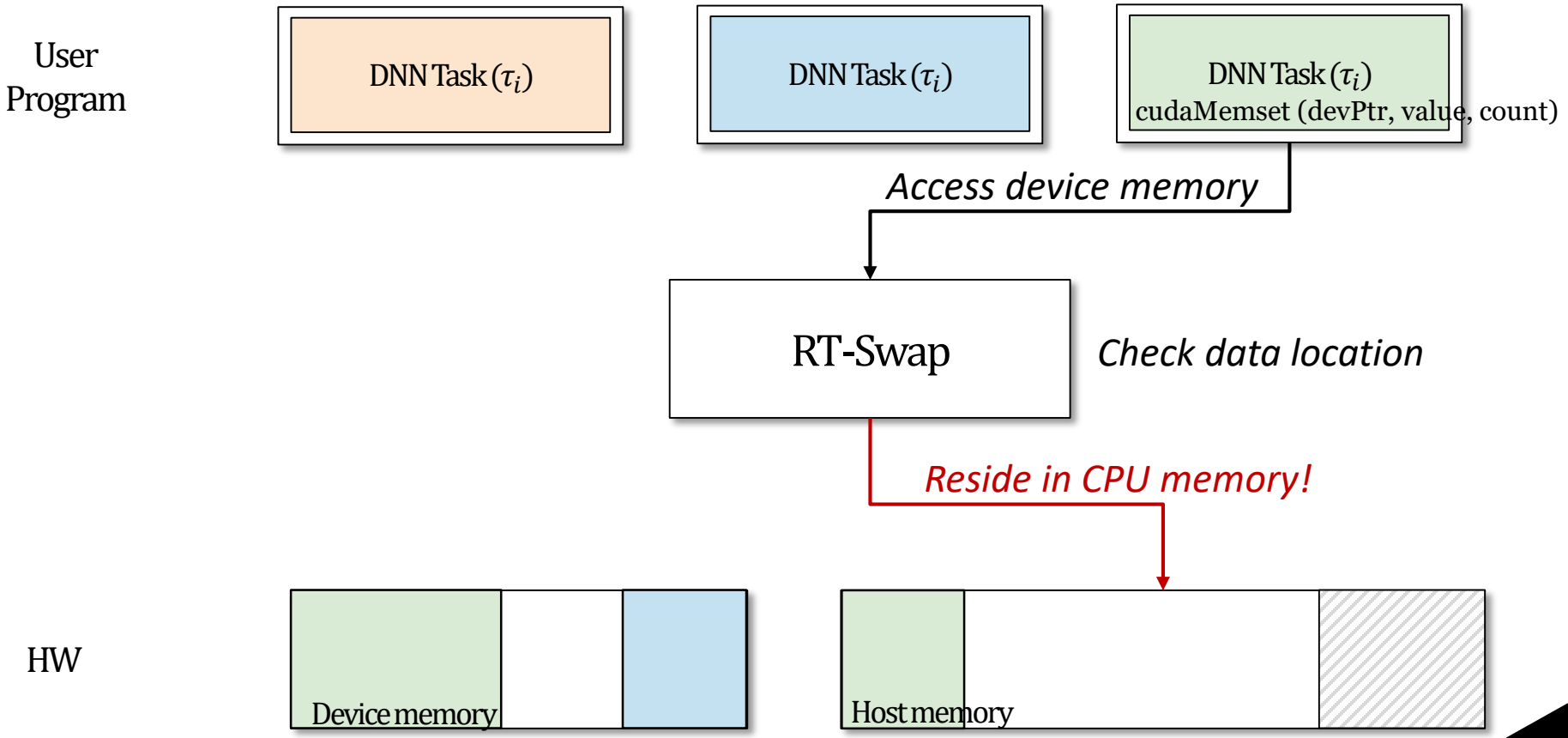
Runtime Memory Swapping



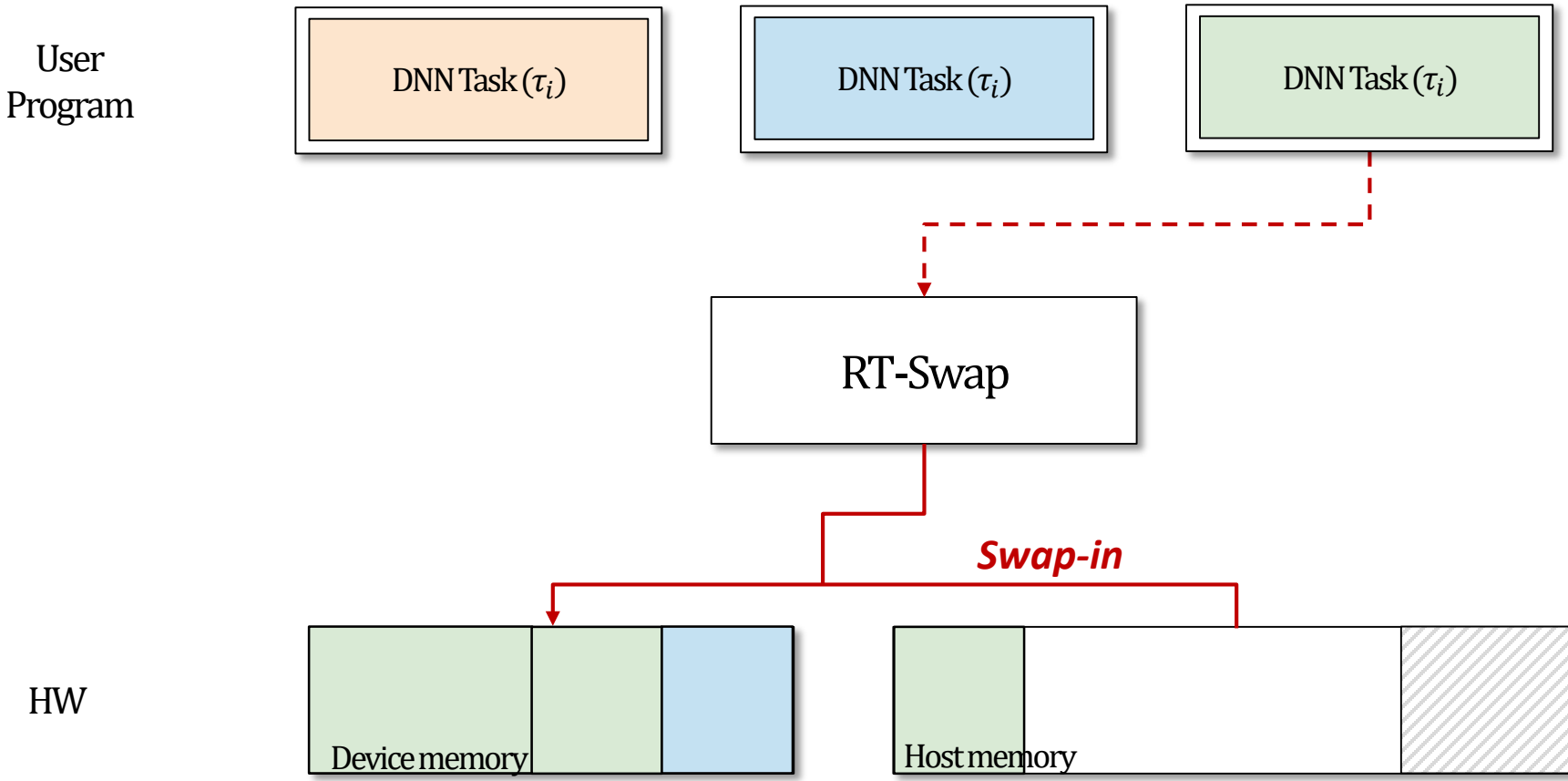
Runtime Memory Swapping



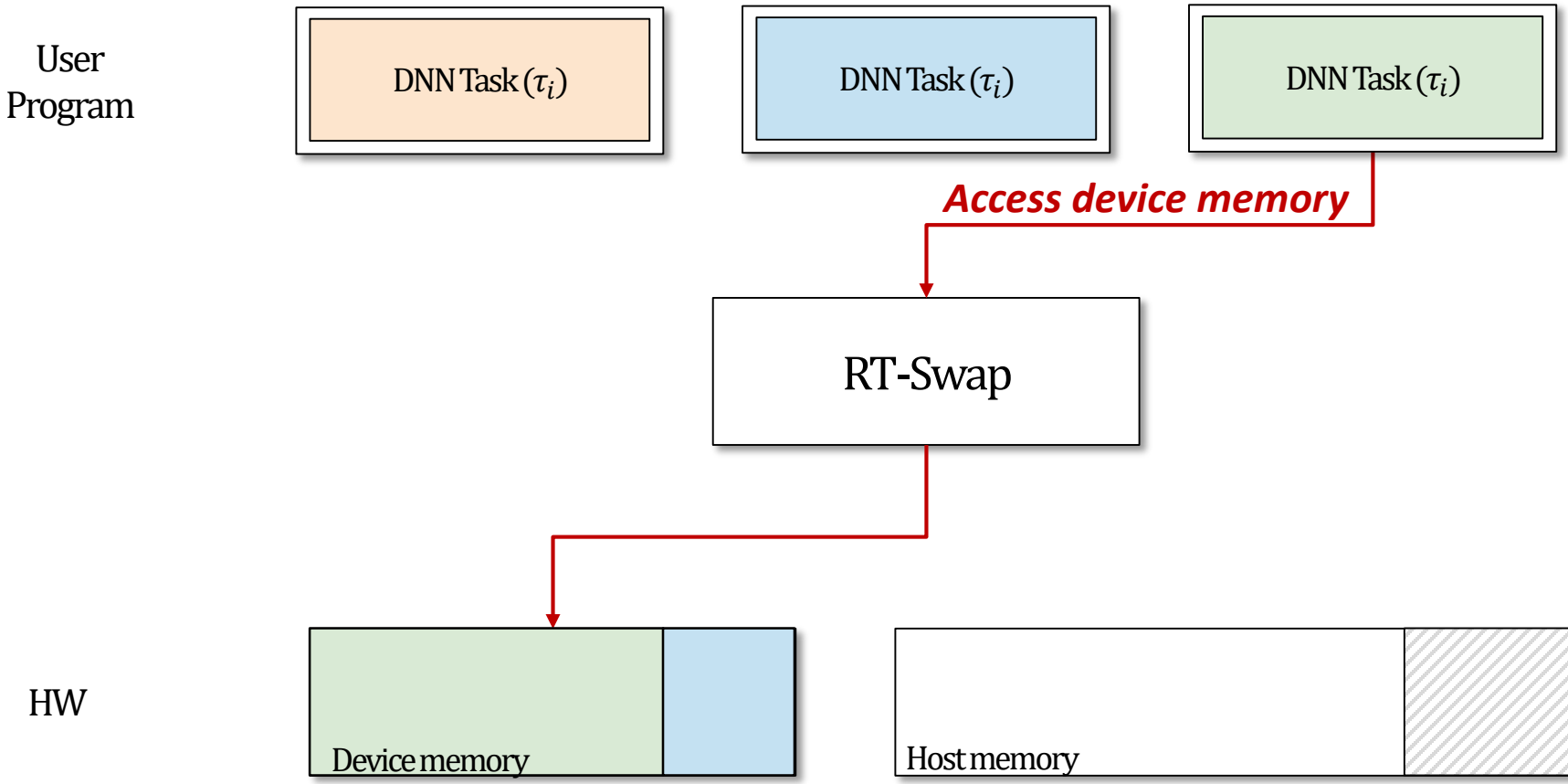
Runtime Memory Swapping



Runtime Memory Swapping

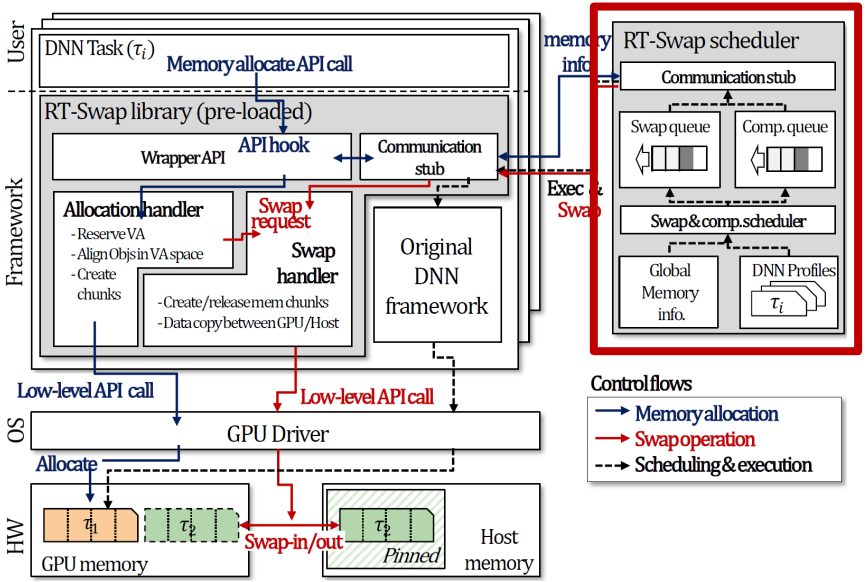


Runtime Memory Swapping



RT-Swap: Beyond the Memory Wall of GPU

- A runtime memory management framework
  - **RT-Swap library** – Transparent swapping operations
    - Swap handler and VMM (Virtual Memory Management) handler
  - **RT-Swap scheduler** – Priority-based scheduling of execution and swap operations



Runtime Results

- A case study with six DNN tasks
  - Cumulative memory footprint is **25.6GB (1.6GB overflow)**
  - Tasks are scheduled under **EDF**
  - Swap chunk size is set **as 32MB**

  - **RT-Swap is compared with Base (NVIDIA oversubscription)**

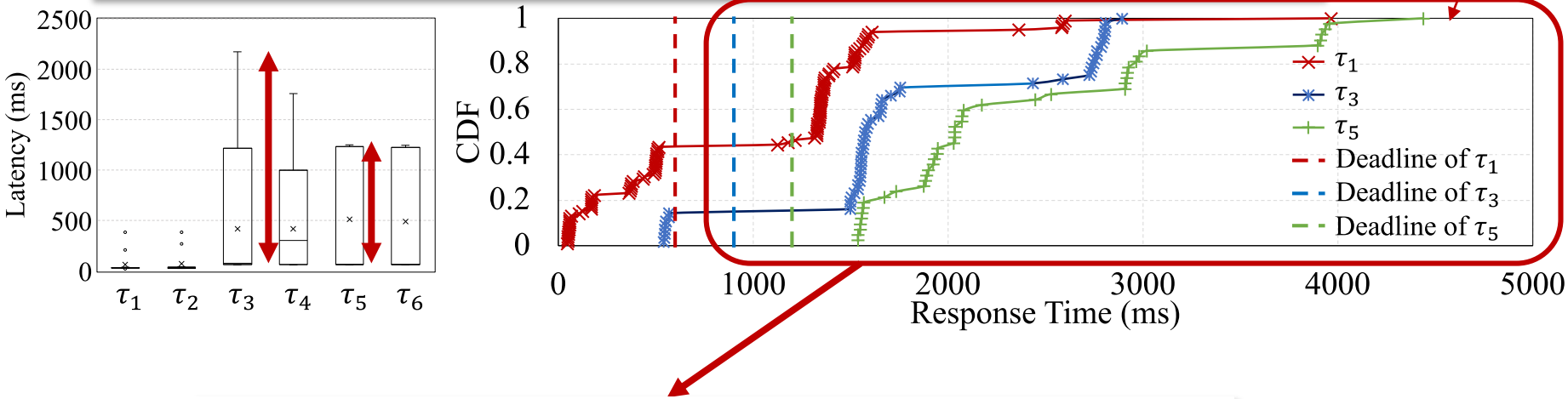
Task	Deadline (ms)	DNN model	Max. swap volume (MB)
τ_1	600	DenseNet_416	0
τ_2	600	ResNet_256	0
τ_3	900	ResNext_608	576
τ_4	900	ResNext_608	576
τ_5	1200	ResNext_608	576
τ_6	1200	ResNext_608	576

[Task set information]

Runtime Results

- Base (NVIDIA oversubscription)
  - Latency consists of the execution time of DNN and swapping overheads

large and long execution time variations due to the ***LRU eviction policy***

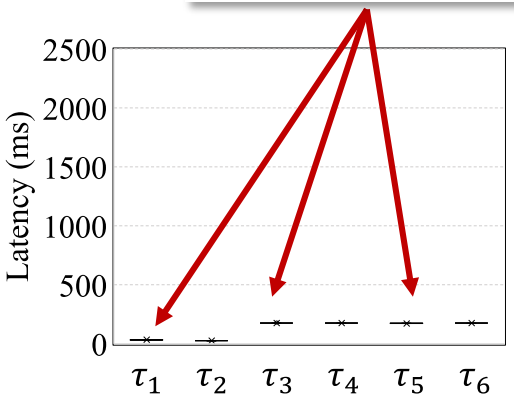


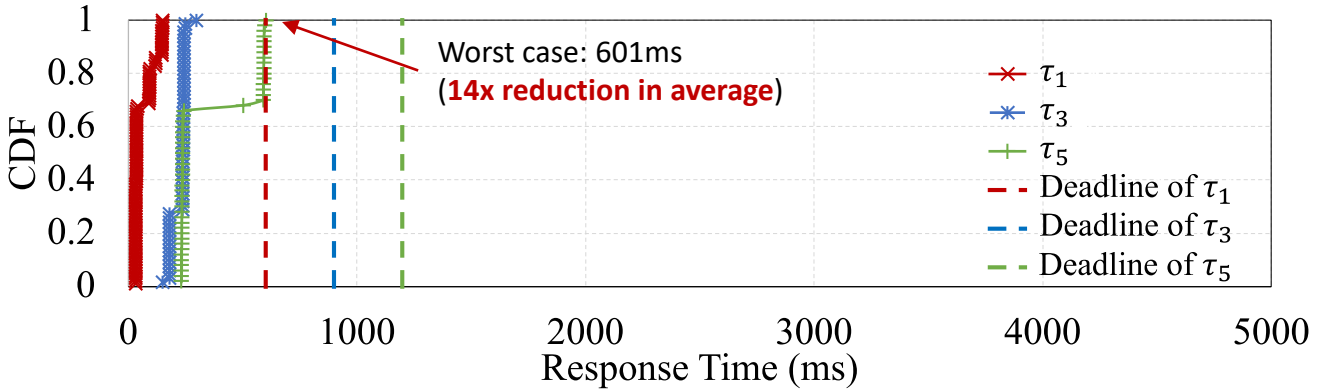
Deadline misses due to its ***large variation in execution time***

Runtime Results

- RT-Swap
  - Latency consists of the execution time of DNN and swapping overheads

Predictable and fast execution time thanks to ***fixed and optimized swap operation***





No deadline misses thanks to ***swap-aware real-time scheduling***

SPET: Transparent SRAM Allocation and Model Partitioning for Real-Time DNN Tasks on Edge TPU

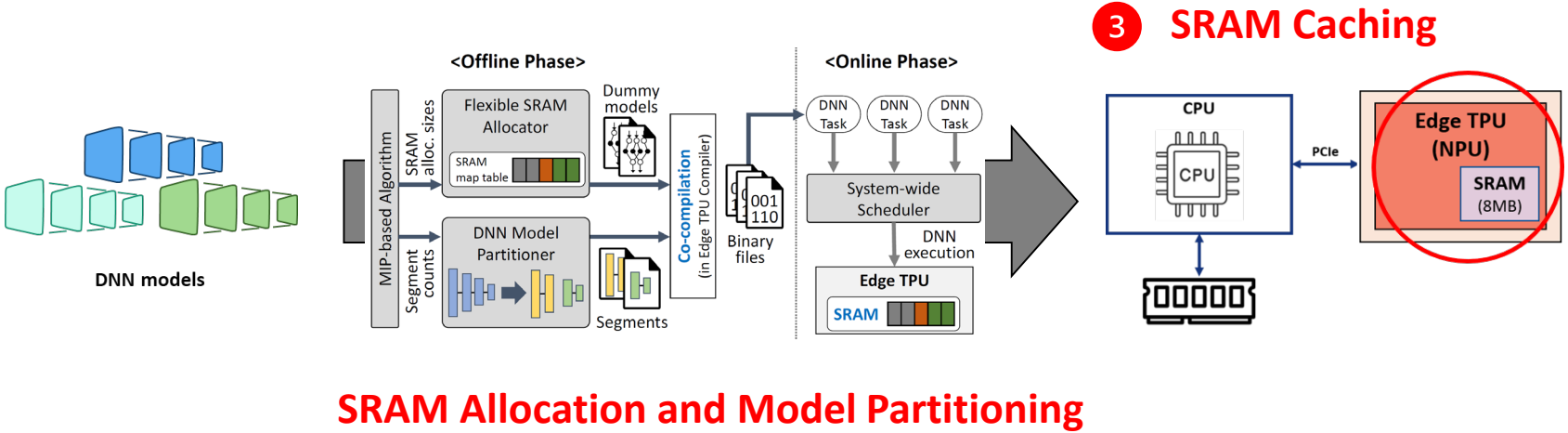
Changhun Han[†], Hoon Sung Chwa[‡], Kilho Lee[§], and Sangeun Oh[†]

60th ACM/IEEE Design Automation Conference (DAC 2023)

SPET: Transparent SRAM Allocation and Model Partitioning for Real-Time DNN Tasks on Edge TPU

Changhun Han[†], Hoon Sung Chwa[‡], Kilho Lee[§], and Sangeun Oh[†]

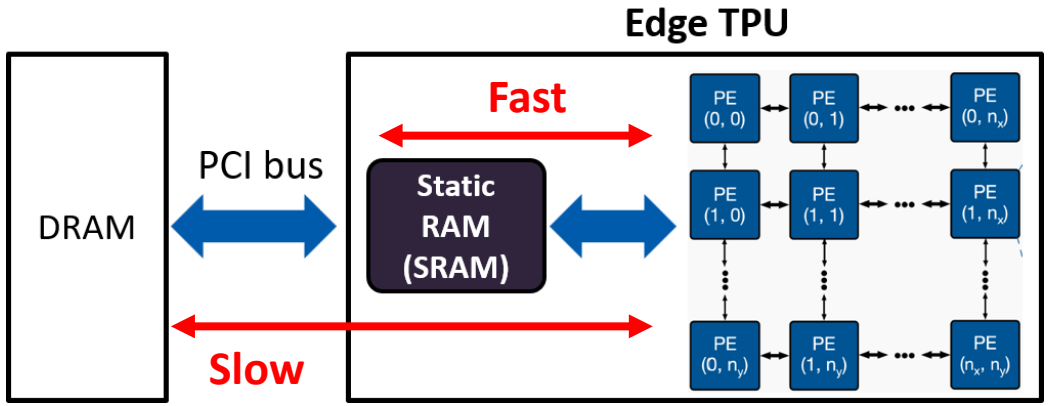
60th ACM/IEEE Design Automation Conference (DAC 2023)



SRAM Allocation and Model Partitioning

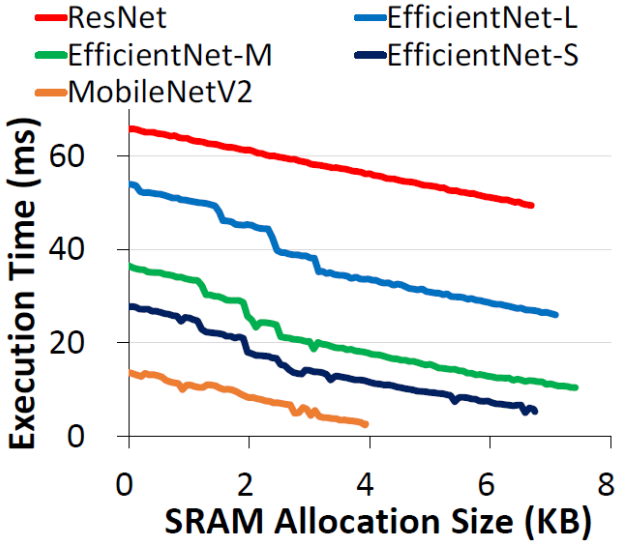
Edge TPU

- Architecture
  - Processing elements (PEs) array
  - **Static RAM (SRAM)**
    - Internal cache memory for storing parameter data of DNN (8-MB size)
    - Enables faster inference than fetching the parameters from the host memory (DRAM)



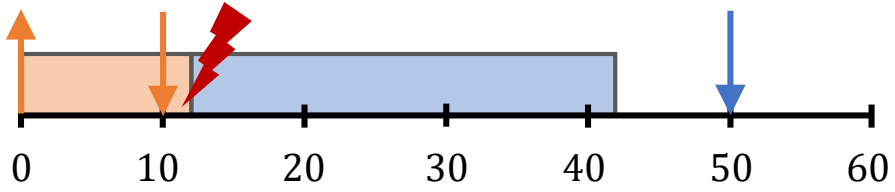
Problems in Edge TPU

- **P1. Deadline-oblivious SRAM allocation**
  - SRAM allocation↑ ➔ Execution time of a DNN task ↓
  - The compiler assigns SRAM accroding to <u>the memory intensiveness of each model</u>



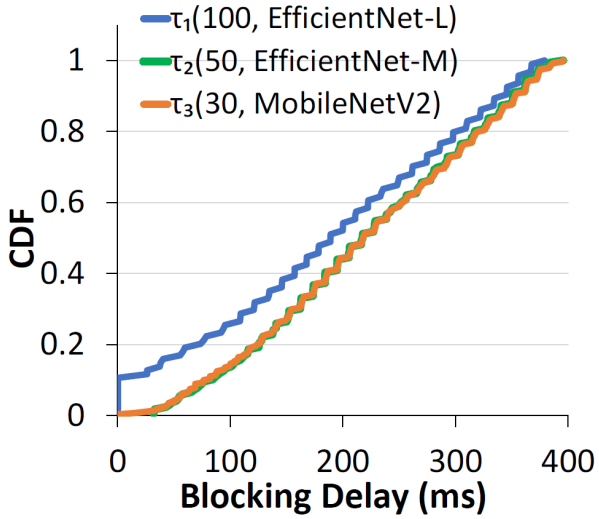
	SRAM size (exec. time)	Deadline
MobileNetV2	1.6 MB (12 ms)	10 ms
EfficientNet-L	5.5 MB (30 ms)	50 ms

Deadline missed by insufficient SRAM!



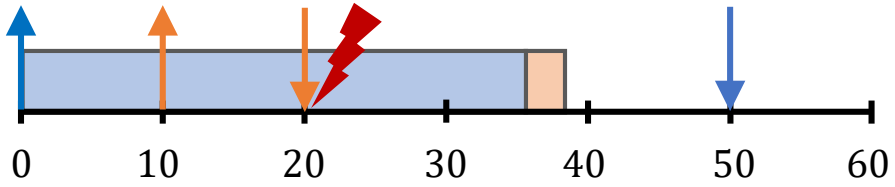
Problems in Edge TPU

- **P2. Non-preemptive nature of Edge TPU**
  - Edge TPU schedules DNN tasks in a non-preemptive FIFO manner
  - Impose unpredictable and non-trivial blocking delays on high-priority tasks



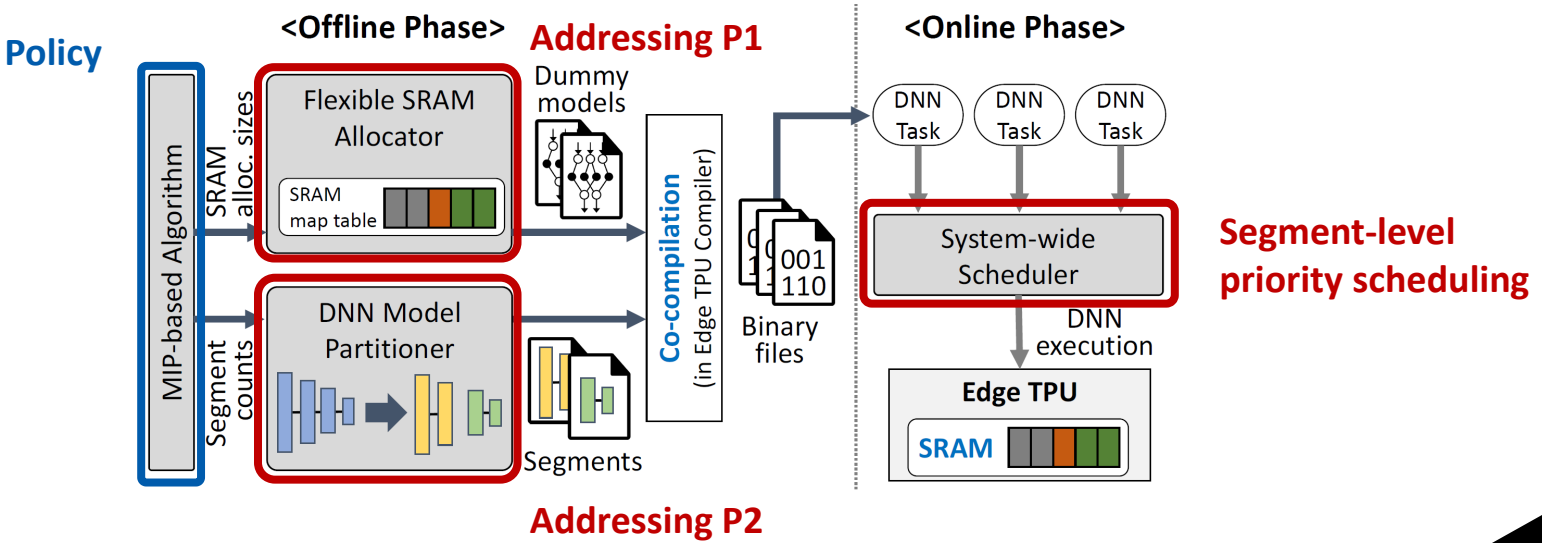
	Prio.	Exec. time	Deadline
MobileNetV2	High	3 ms	10 ms
EfficientNet-L	Low	35 ms	50 ms

Deadline missed by long blocking



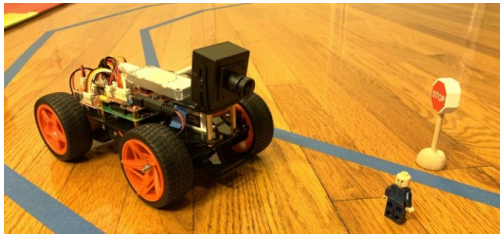
SPET Framework

- A novel DNN framework that supports the real-time guarantee of multiple DNN tasks on Edge TPU
  - **Policy** ➔ MIP-based algorithm for finding proper SRAM allocation & segment counts
  - **Mechanism** ➔ transparent SRAM allocation & model partitioning

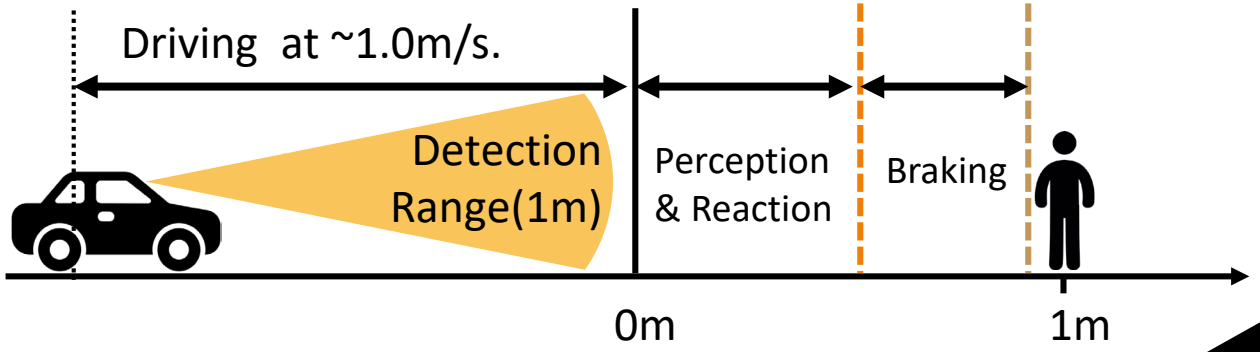


Case Study

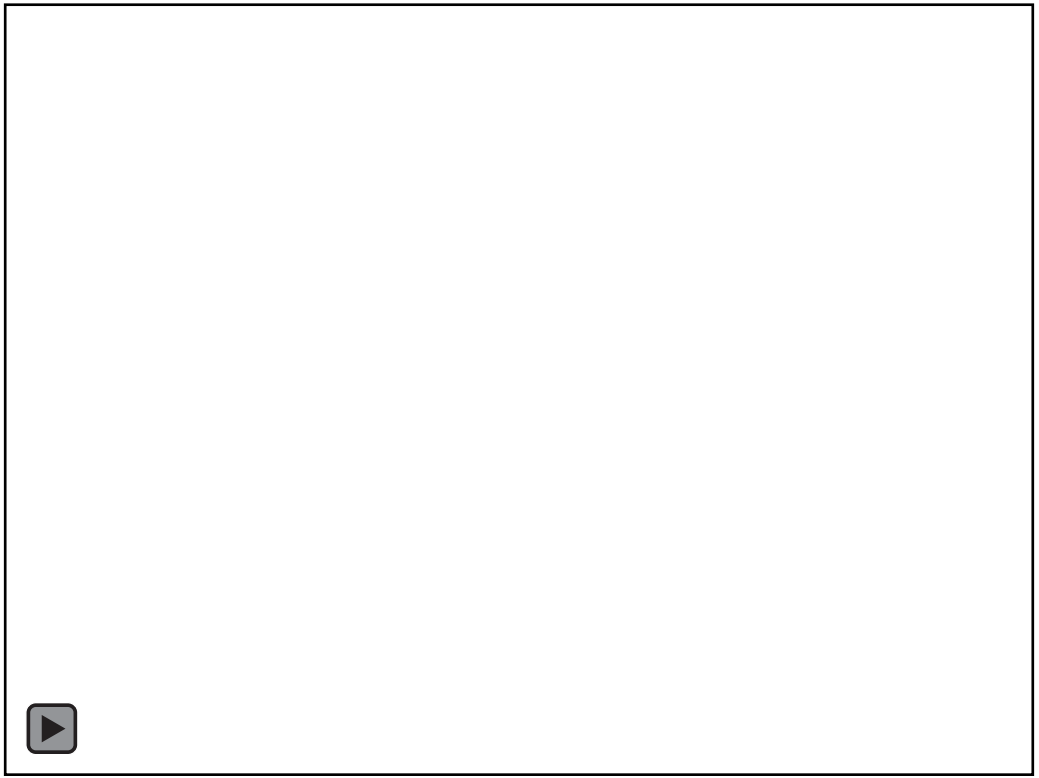
- Emergency braking system of 1/10 scale self-driving car with Edge TPU
  - Three DNN tasks
    - $\tau_1(Period = 250ms$, EfficientDet) for **object detection**
    - $\tau_2(Period = 330ms$, DAVE=2) for **lane tracking**
    - $\tau_3(Period = 1{,}000ms$, InceptionV2) for **image classification**
  - Emergency braking when a pedestrian is detected within 1m



1/10 scale self-driving car
(PiCar-V)

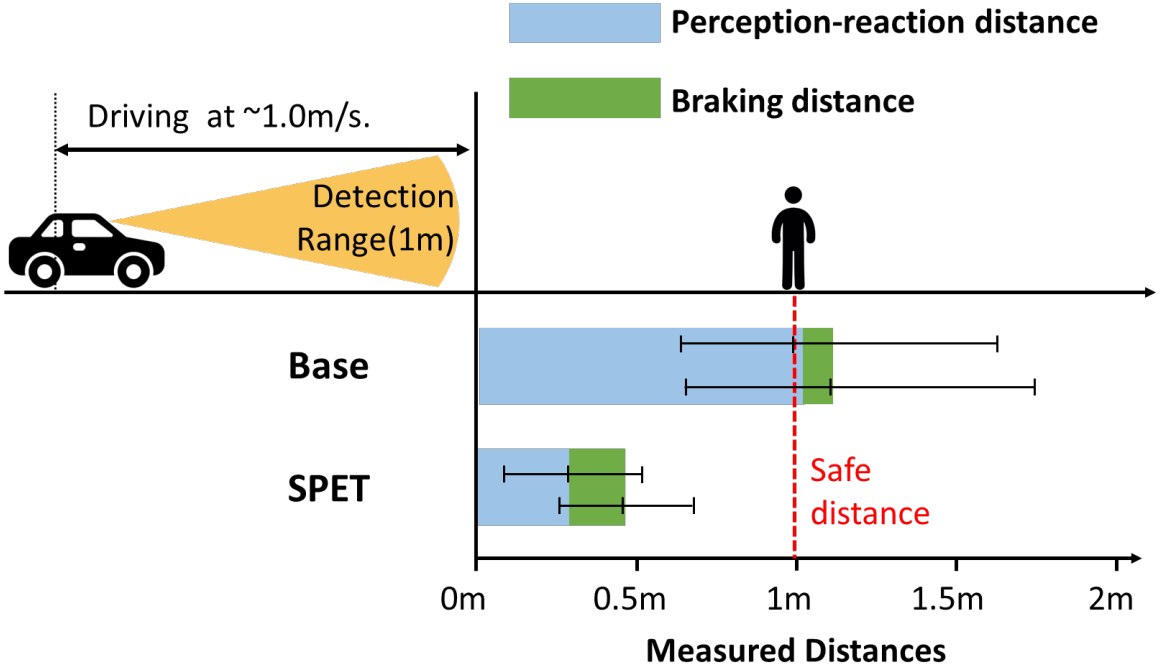


Case Study

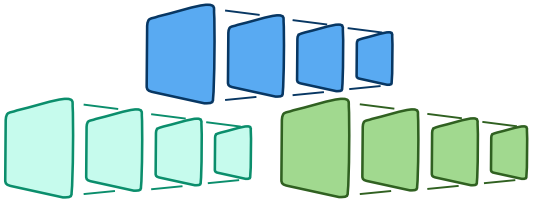


Case Study

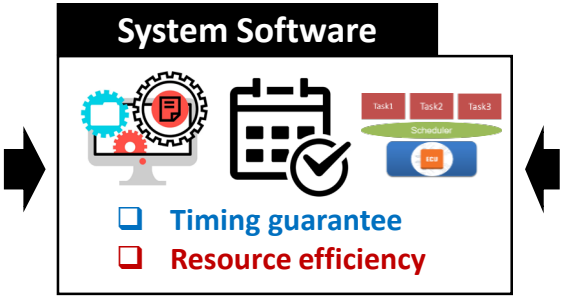
- Perception-reaction distance of SPET is **reduced by 70%** compared to Baseline



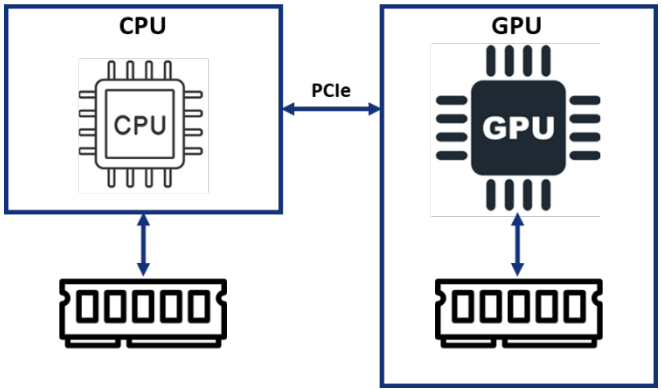
Summary and Future Directions



DNN models



Real-Time Scheduling Framework



Modern AI Computing Platform

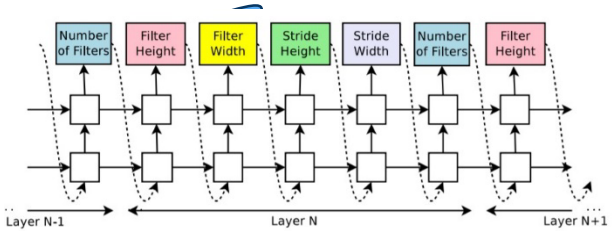
1 **CPU-GPU Performance Imbalance**
[LaLaRAND, RTSS 2021]

2 **GPU Memory Capacity Bottleneck**
[RT-Swap, RTAS 2024]

3 **SRAM Caching**
[SPET, DAC 2023]

4 **Different Criticality**
[DNN-SAM, RTAS 2022]
[RT-MOT, RTSS 2022]

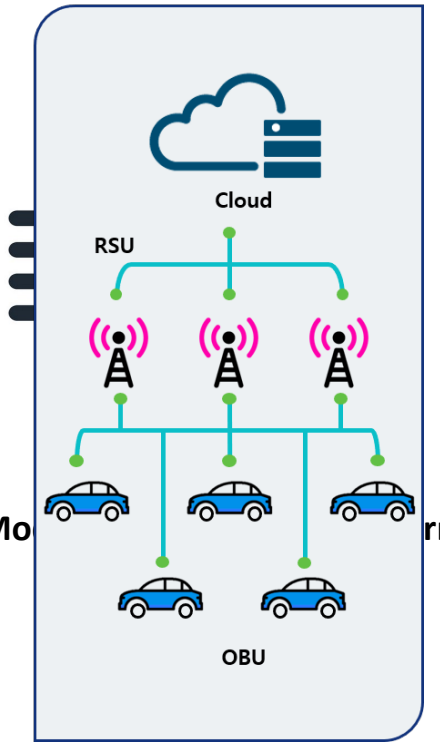
Summary and Future Directions



New DNN models

DNN Models

Real-Time Scheduling Framework



Mo rm

Cloud-Edge Computing

Thank you!

Real-Time Computing Lab.

Hoon Sung Chwa (좌훈승)

https://sites.google.com/view/chwahs

DGIST 대구경북과학기술원
Daegu Gyeongbuk
Institute of Science & Technology