DASYA
Data-Intensive Systems and Applications
dasya.itu.dk

RAD
rad.itu.dk

IT UNIVERSITY OF COPENHAGEN
itu.dk

Taming GPU Interference: Safe Co-location and Fair Resource Management

Ehsan Yousefzadeh-Asl-Miandoab
ehyo@itu.dk

6th March 2026

GPU Underutilization Challenge

Industry reports show ~50% GPU utilization✷

Hardware Limitations

1. Compute/memory requirements of models don't always match with the GPUs
   - e.g., transfer learning, small models
2. No fine-grained sharing mechanism/ No CPU-like virtual memory on GPU

Mem: 141GB

Software Limitations

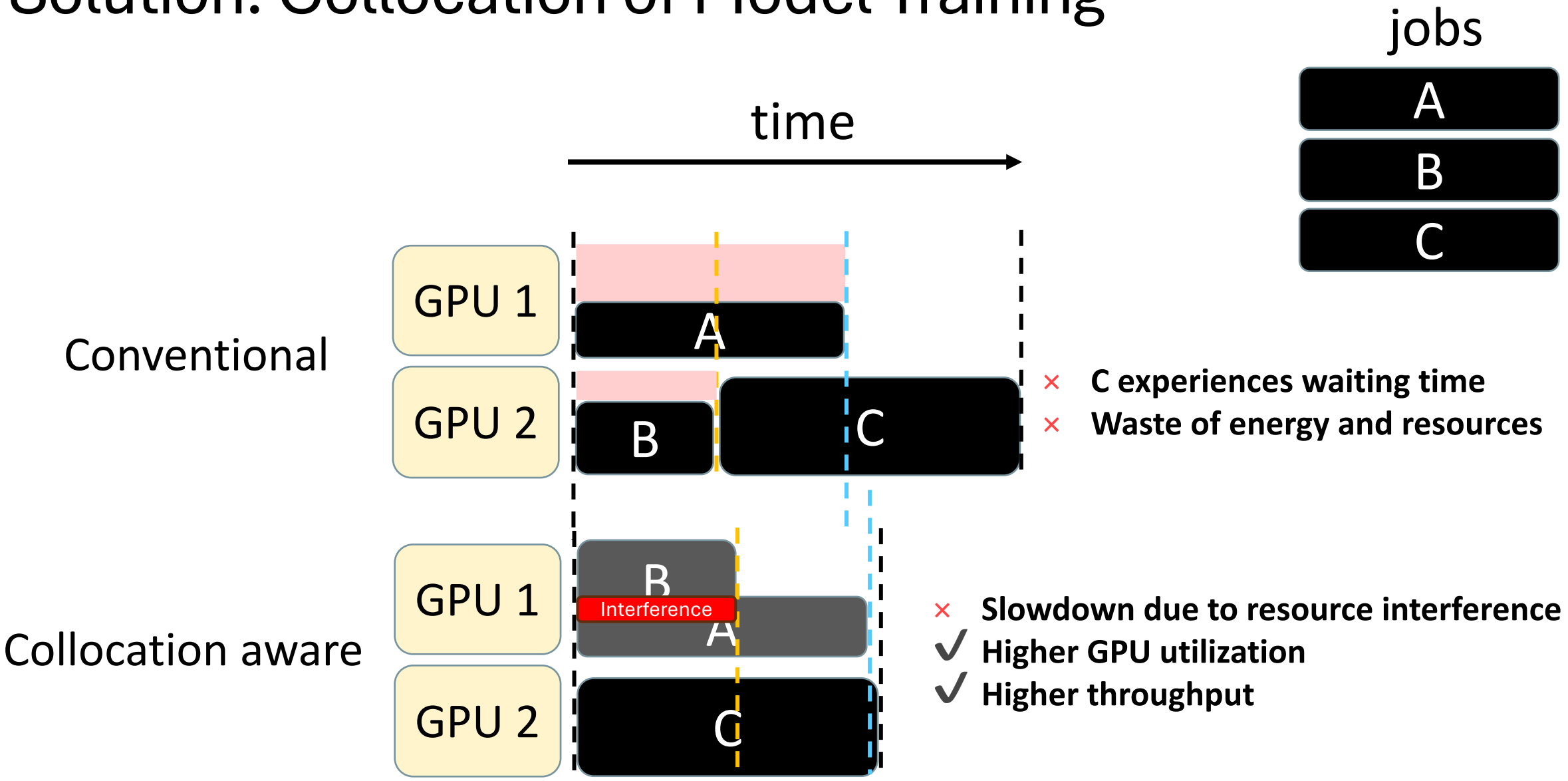
1. Exclusive GPU assignments
2. Black-box view of tasks

slurm workload manager

kubernetes

✷ Jeon, Myeongjae, et al. "**Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads.**" USENIX Annual Technical Conference. 2019.
✷ Yanjie Gao et al. **"An Empirical Study on Low GPU Utilization of Deep Learning Jobs."** In: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. 2024.

Solution: Collocation of Model Training



Overview

Profiling & Monitoring (EuroMLSys'23) 1 → **Collocation Analysis (EuroMLSys'24)** 2 → **GPUMemNet: GPU Memory Estimation Framework** 3 → **CARMA: Collocation-aware Resource Manager** 4

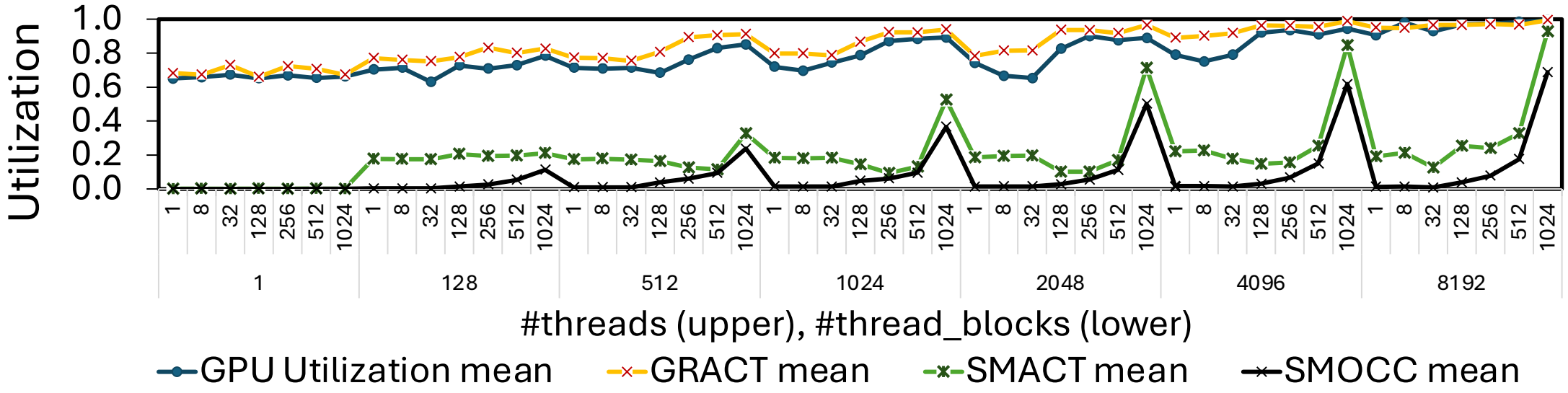
```
sh-4.2$ nvidia-smi
Wed Dec 11 09:52:10 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.87.01    Driver Version: 418.87.01    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           On   | 00000000:00:1E.0 Off |                    0 |
| N/A   44C    P0    60W / 149W |  11292MiB / 11441MiB |     13%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0      9389      C   ...naconda3/envs/tensorflow_p36/bin/python 11012MiB |
|    0     21268      C   ...naconda3/envs/tensorflow_p36/bin/python   267MiB |
+-----------------------------------------------------------------------------+
```

Read the paper here!

Overview

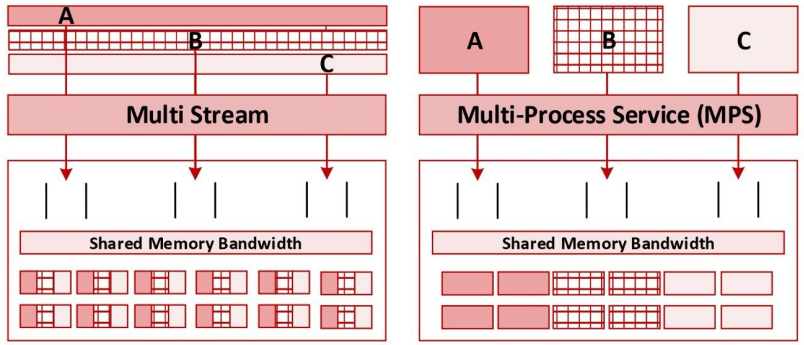
Profiling & Monitoring (EuroMLSys'23) 1 → **Collocation Analysis** (EuroMLSys'24) 2 → **GPUMemNet: GPU Memory Estimation Framework** 3 → **CARMA: Collocation-aware Resource Manager** 4

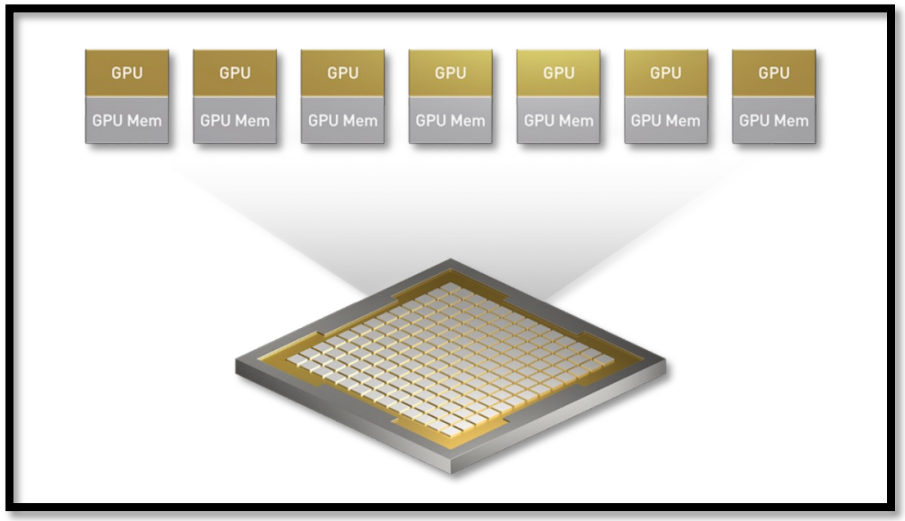


Coarser-grained utilization metrics can be misleading.

Overview

Profiling & Monitoring (EuroMLSys'23) 1 → **Collocation Analysis (EuroMLSys'24) 2** → GPUMemNet: GPU Memory Estimation Framework 3 → CARMA: Collocation-aware Resource Manager 4



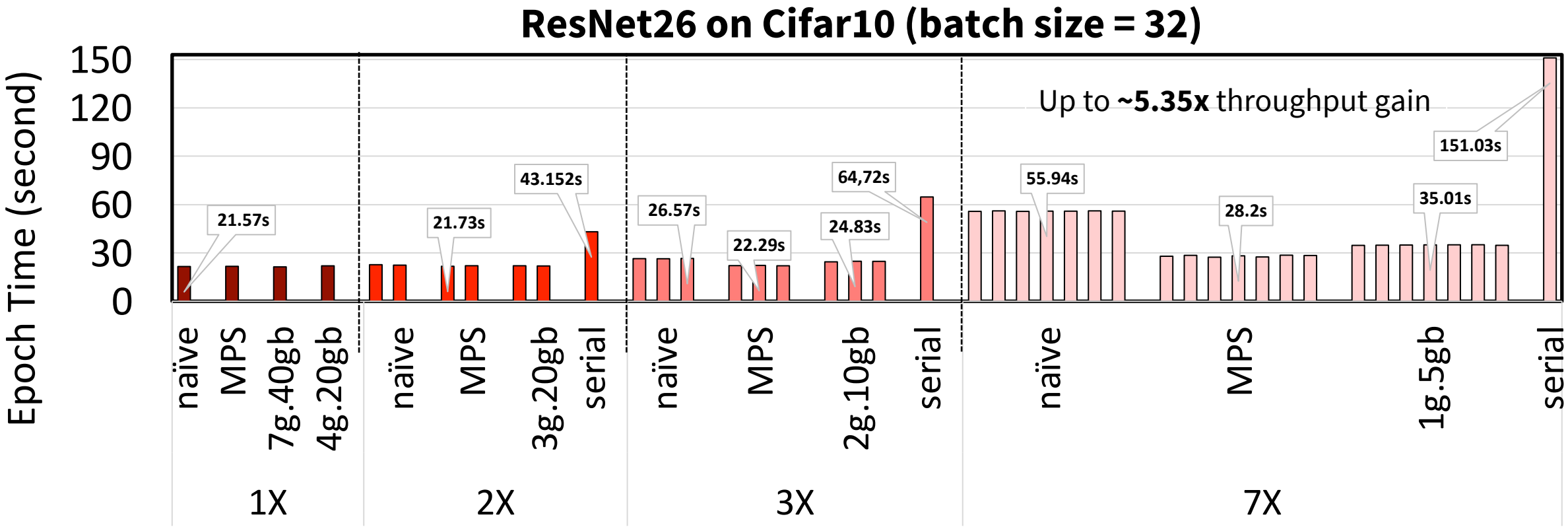


https://www.nvidia.com/en-us/technologies/multi-instance-gpu/

Read the paper here!

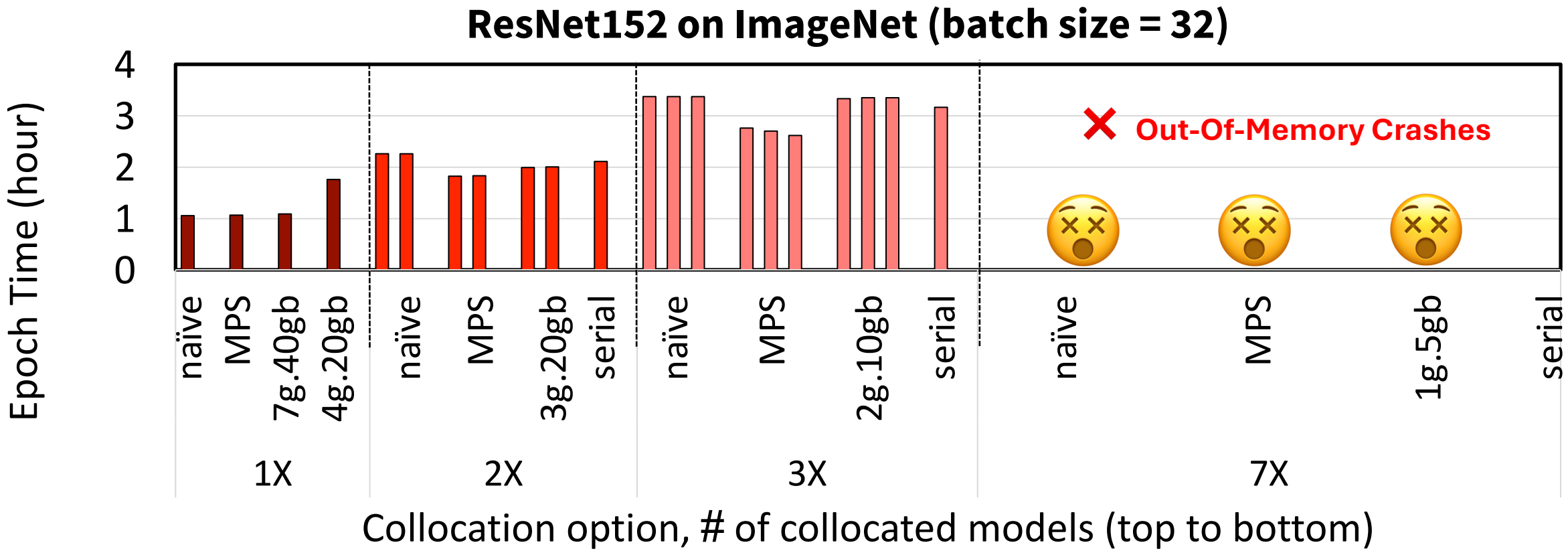
Overview

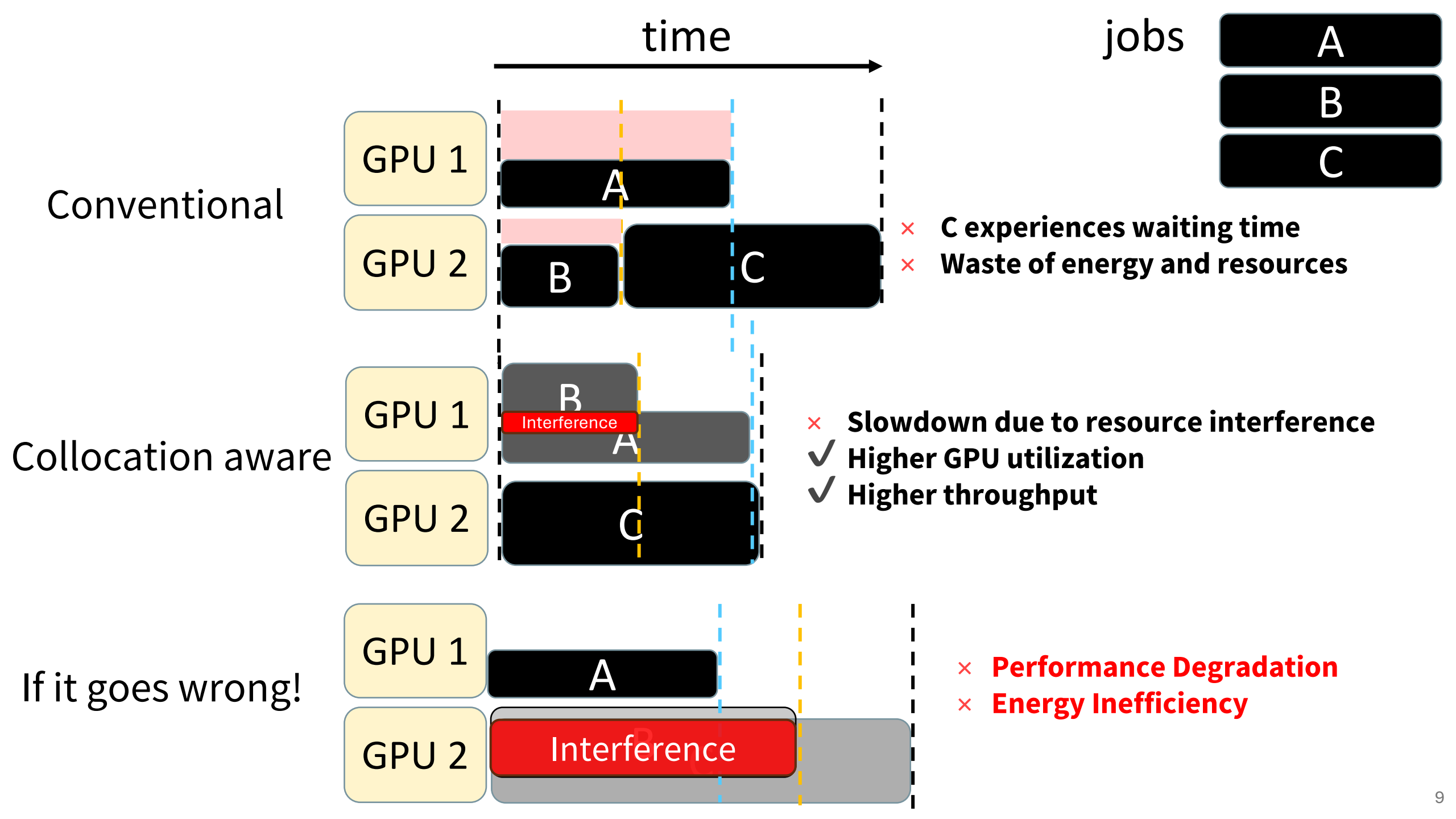
Profiling & Monitoring (EuroMLSys'23) 1 → Collocation Analysis (EuroMLSys'24) 2 → GPUMemNet: GPU Memory Estimation Framework 3 → CARMA: Collocation-aware Resource Manager 4



Overview

Profiling & Monitoring (EuroMLSys'23) 1 → Collocation Analysis (EuroMLSys'24) 2 → GPUMemNet: GPU Memory Estimation Framework 3 → CARMA: Collocation-aware Resource Manager 4

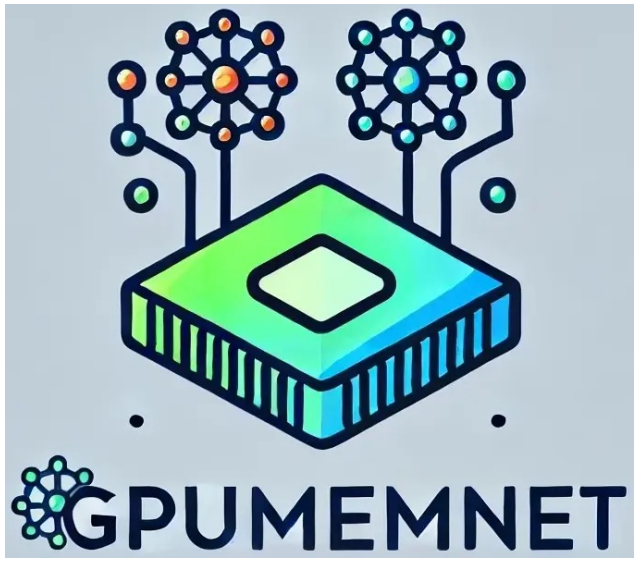



time
jobs
A
B
C
Conventional
GPU 1
GPU 2
A
B
C
× C experiences waiting time
× Waste of energy and resources
Collocation aware
GPU 1
GPU 2
B
Interference
A
C
× Slowdown due to resource interference
✓ Higher GPU utilization
✓ Higher throughput
If it goes wrong!
GPU 1
GPU 2
A
Interference
× Performance Degradation
× Energy Inefficiency

Overview

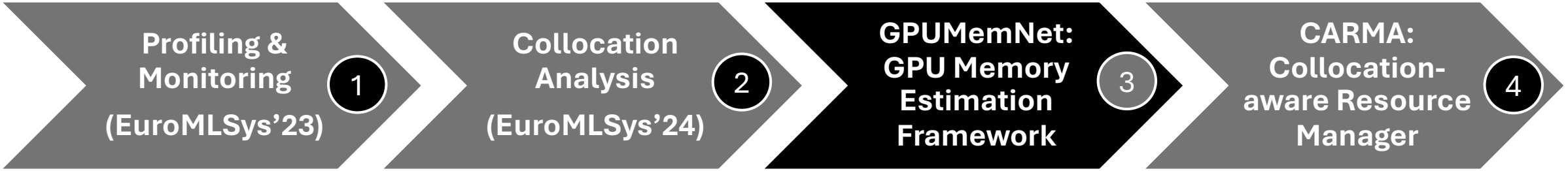
1. Profiling & Monitoring (EuroMLSys'23)
2. Collocation Analysis (EuroMLSys'24)
3. **GPUMemNet: GPU Memory Estimation Framework**
4. CARMA: Collocation-aware Resource Manager

Check the code here!



Read the paper here!

Overview



- **Memory estimators** (analytical, CPU-side, and ML-based tools) share key limitations in resource management:
  - Rely on model specifications or the computation graph
  - Require intrusive design changes, or depend on the user to provide them
  - Impose overhead on the critical decision-making path
  - Do not generalize well

- **Applying ML to this problem** requires costly dataset curation and problem formulation.

- **GPU utilization is a poor proxy for resource interference** — utilization is not additive.

Overview

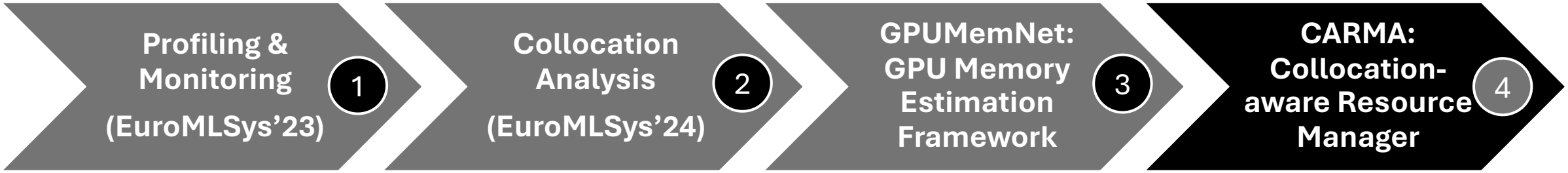
1. Profiling & Monitoring (EuroMLSys'23)
2. Collocation Analysis (EuroMLSys'24)
3. GPUMemNet: GPU Memory Estimation Framework
4. CARMA: Collocation-aware Resource Manager

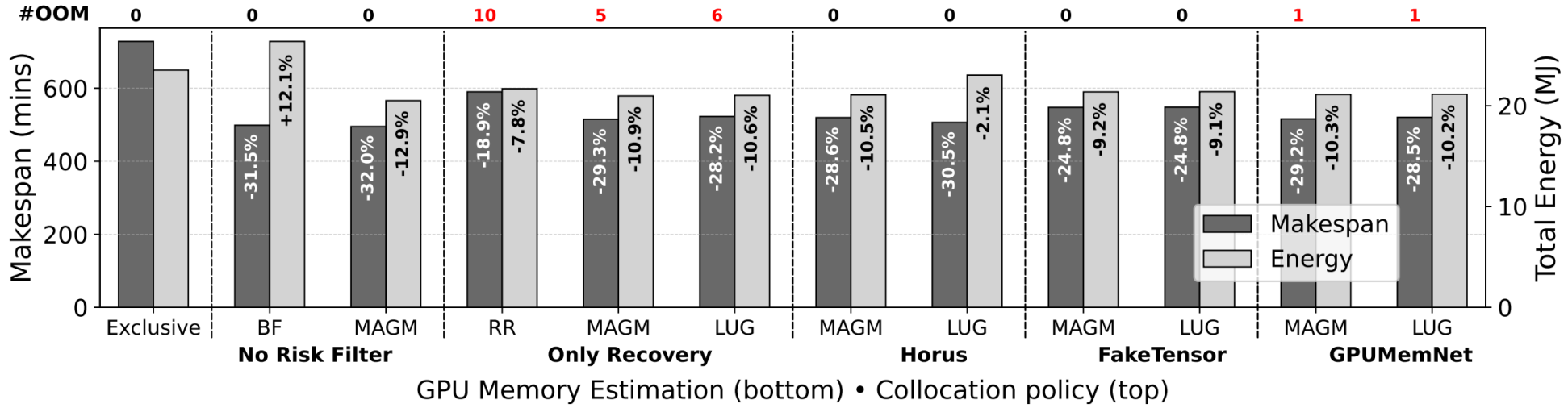
Check the code here!

CARMA

Read the paper here!

Overview





- **Collocation Policies + active monitoring of GPUs + recovery method (for OOMs) improves GPU utilization and energy efficiency.**
- **High resource interference can negate collocation energy efficiency.**

Ongoing & Future Work

- Deeper look into taming GPU interference with more general approach
  - Ditching estimations
- Colocation raises fairness concerns ?
  - Is GPU-minutes sufficient as a fairness metric?
- Extend support to AMD GPUs
- Characterize direct GPU memory and storage access methods

Thanks ☺

ehyo@itu.dk