DASYA

www.dasya.dk

RAD

rad.itu.dk

IT UNIVERSITY OF COPENHAGEN

www.itu.dk

deep learning with fewer resources: *collocation-aware GPU orchestration*

Pınar Tözün

Associate Professor, IT University of Copenhagen

pito@itu.dk, pinartozun.com, @pinartozun

UNIL Feb 10, 2026
HEIG-VD Feb 12, 2026

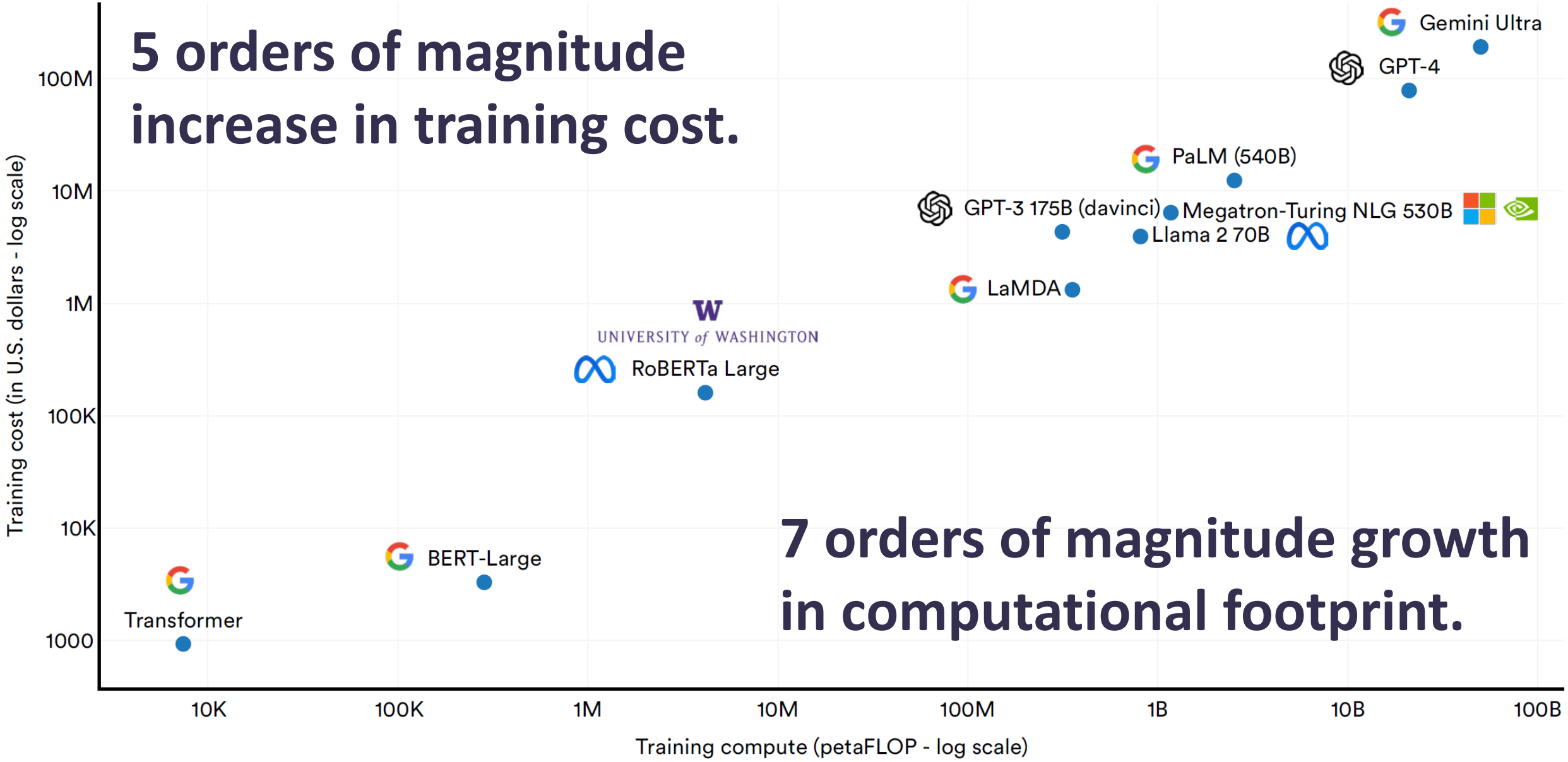
Innovationsfonden

INDEPENDENT RESEARCH FUND DENMARK

Swiss National Science Foundation

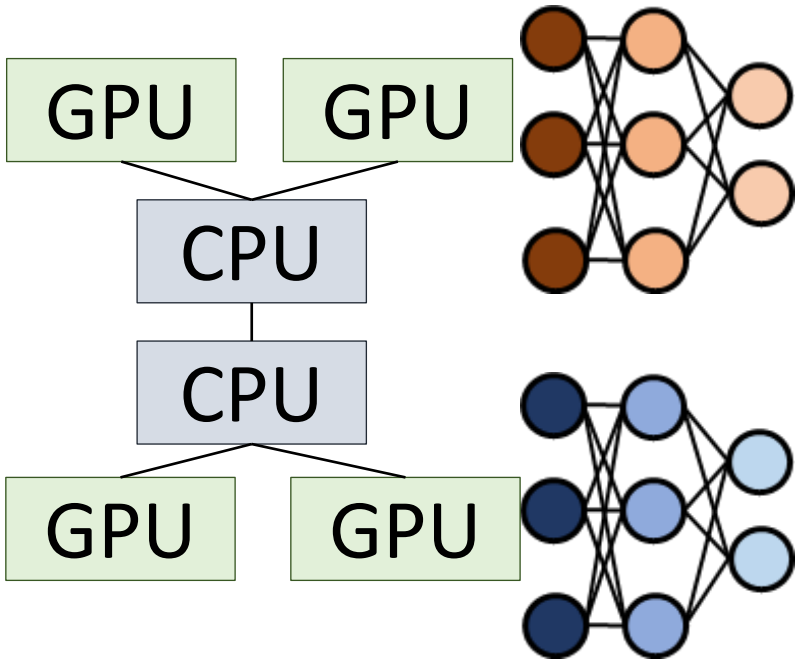
novo nordisk foundation

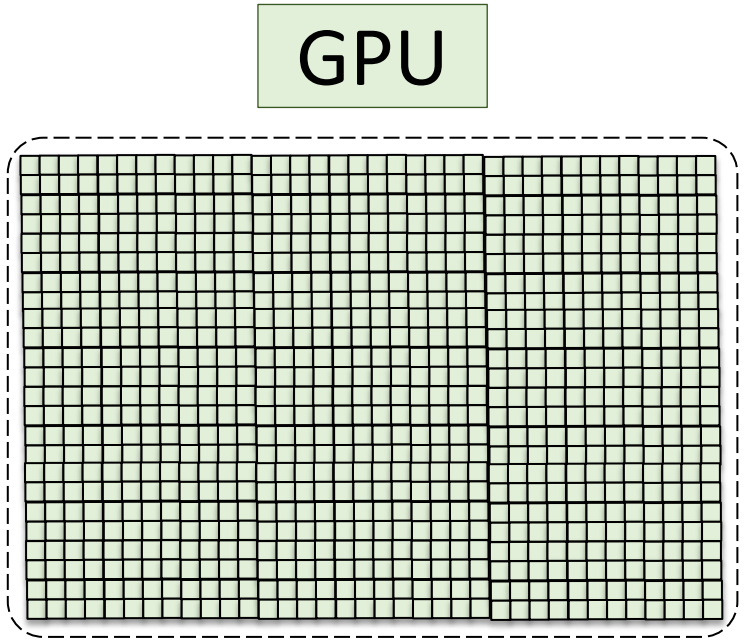
language model training (2017 – today)



source: Stanford AI Index Report 2024

commodity hardware for deep learning





graphics processing unit

➔ many (simple) cores

➔ good for throughput-oriented & embarrassingly parallel tasks

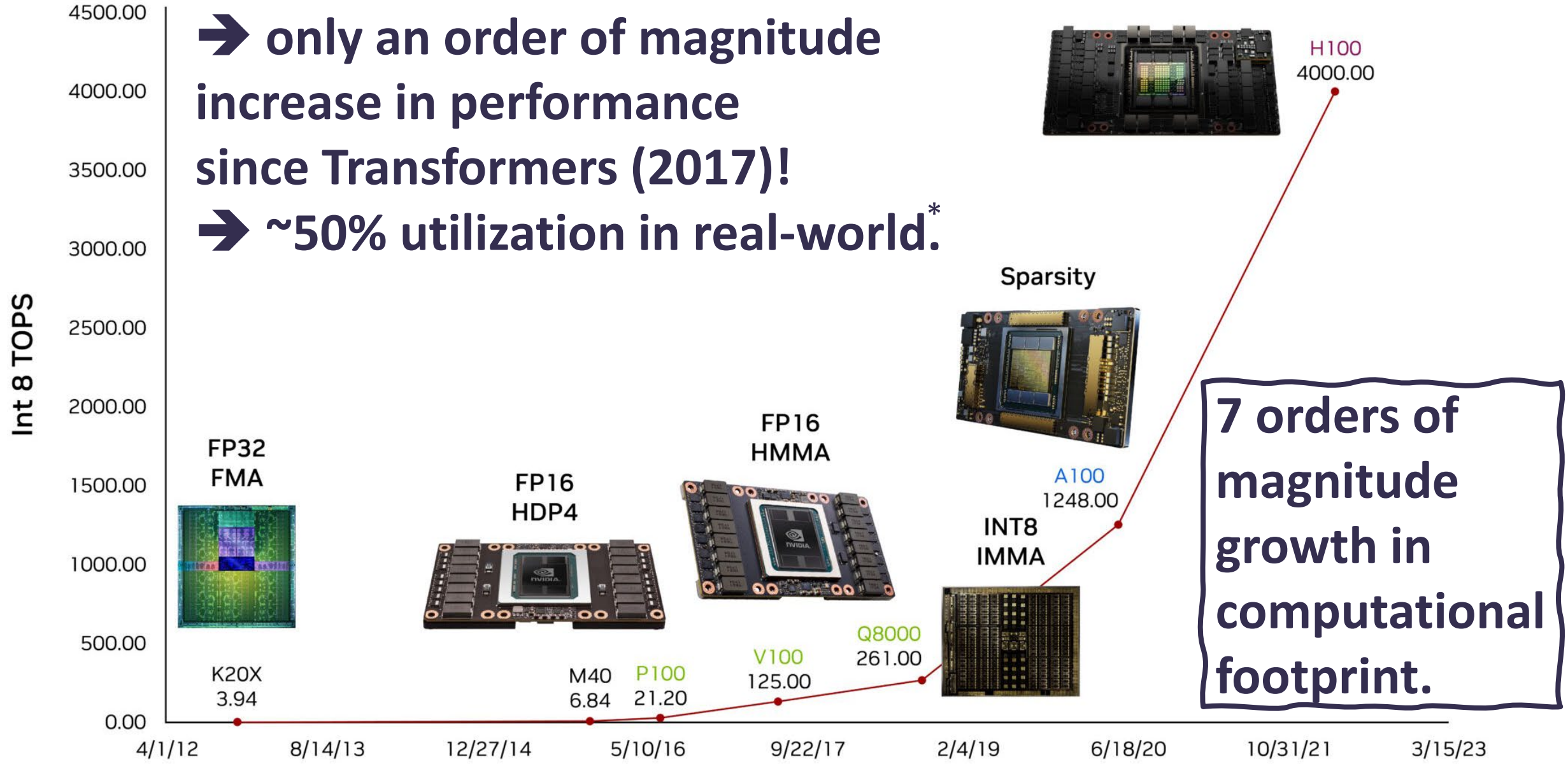
➔good for deep learning

- e.g., large matrix operations

NVIDIA GPUs (2012 – 2023)

➔ only an order of magnitude increase in performance since Transformers (2017)!

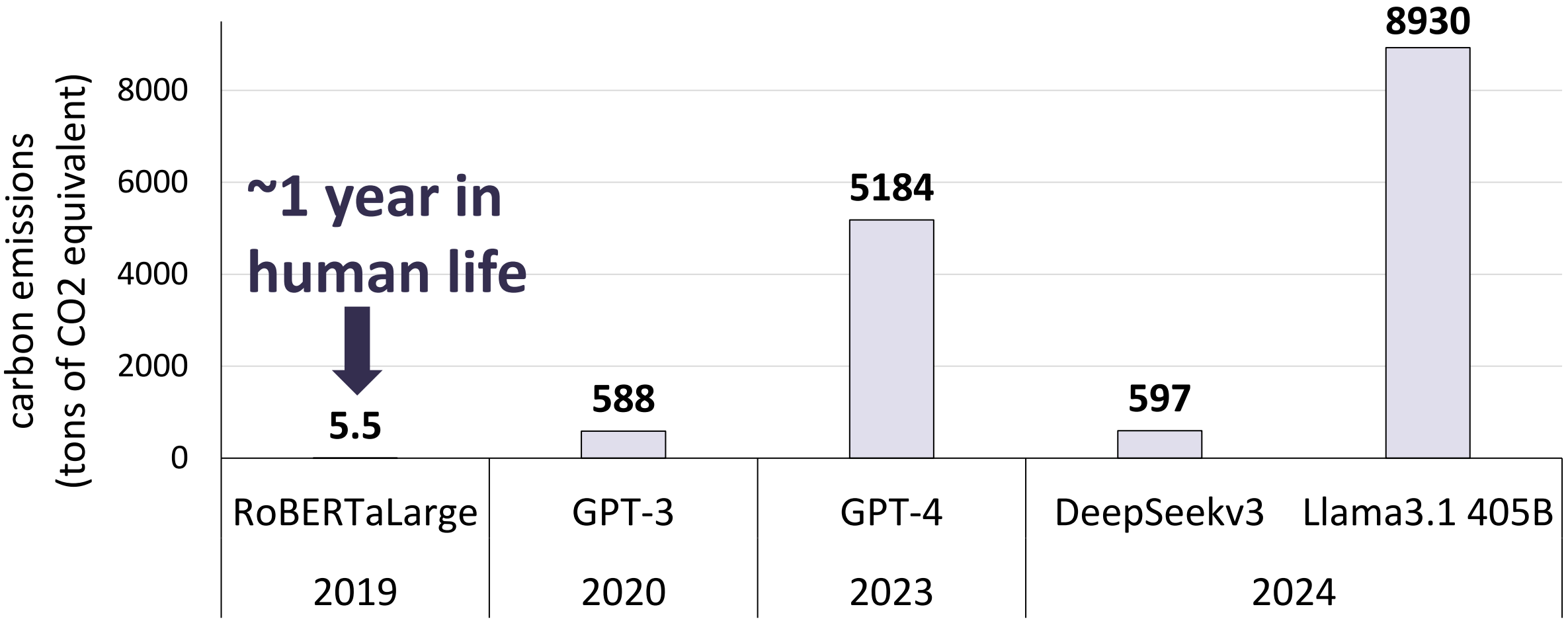
➔ ~50% utilization in real-world.*



7 orders of magnitude growth in computational footprint.

sources: figure by Bill Dally, NVIDIA & *Gao et al. "An Empirical Study on Low GPU Utilization of Deep Learning Jobs." ICSE 24

carbon footprint of language model training



can we do better while using fewer resources?

model accuracy cannot be the only metric to aim for!

source: Stanford AI Index Report 2025

workload collocation for model training

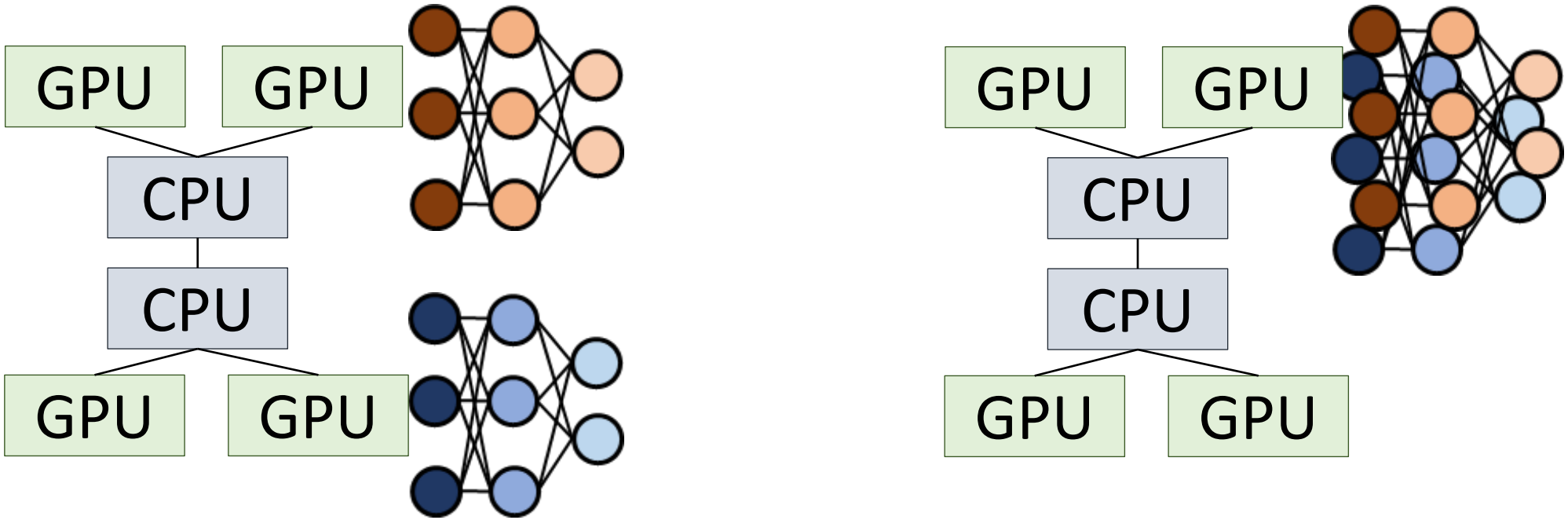
- analysis of GPU resource sharing primitives

[An Analysis of Collocation on GPUs for Deep Learning Training]
Ties Robroek, Ehsan Yousefzadeh-Asl-Miandoab, Pınar Tözün.
EuroMLSys 2024

- collocation-aware resource management

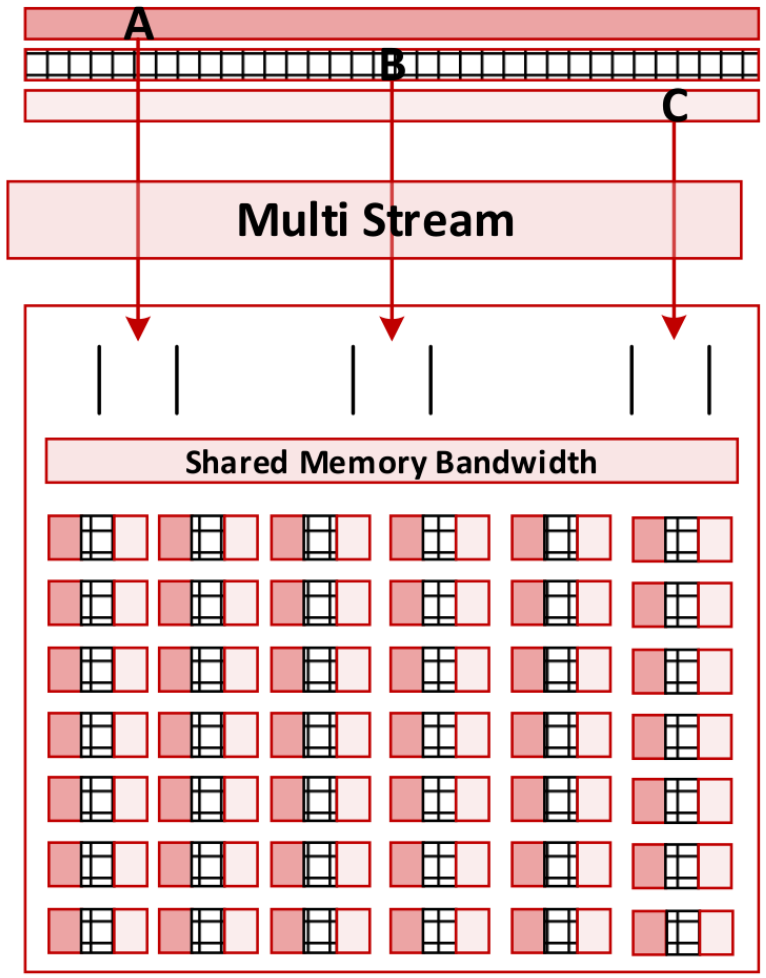
[CARMA: Collocation-Aware Resource Manager]
Ehsan Yousefzadeh-Asl-Miandoab, Reza Karimzadeh, Bulat Ibragimov, Florina M Ciorba, Pınar Tözün.

collocated training

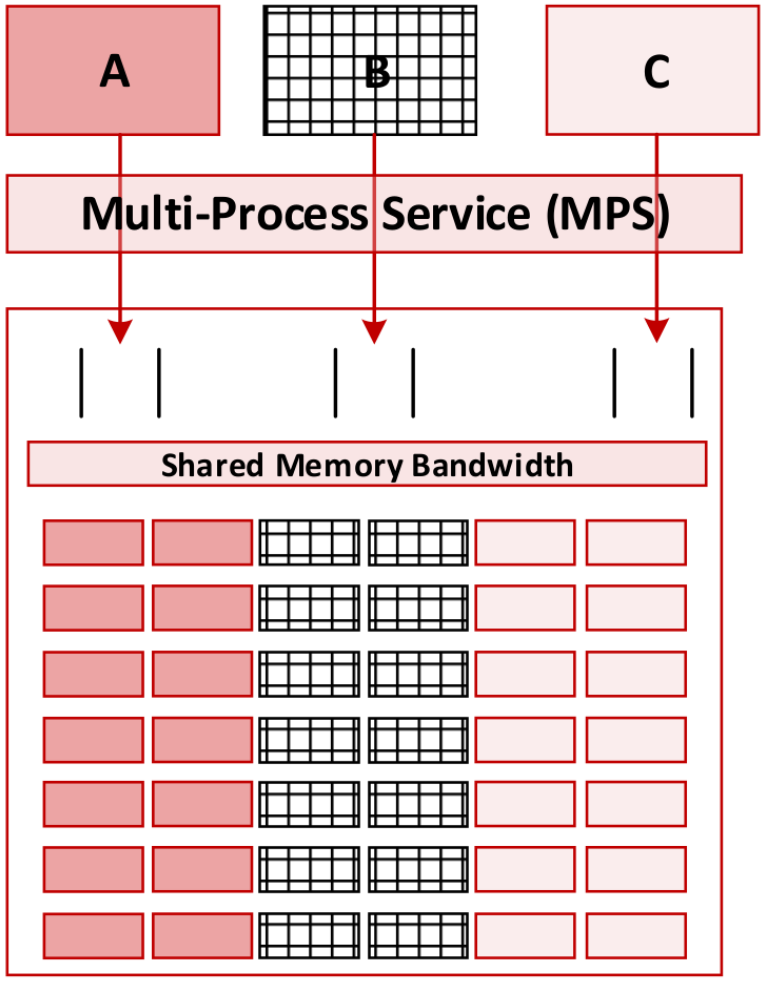


- **training more models with fewer hardware resources**
- **if done well ➔ better hardware utilization & reduces costs**
- **if not done well ➔ interference & performance degradation**

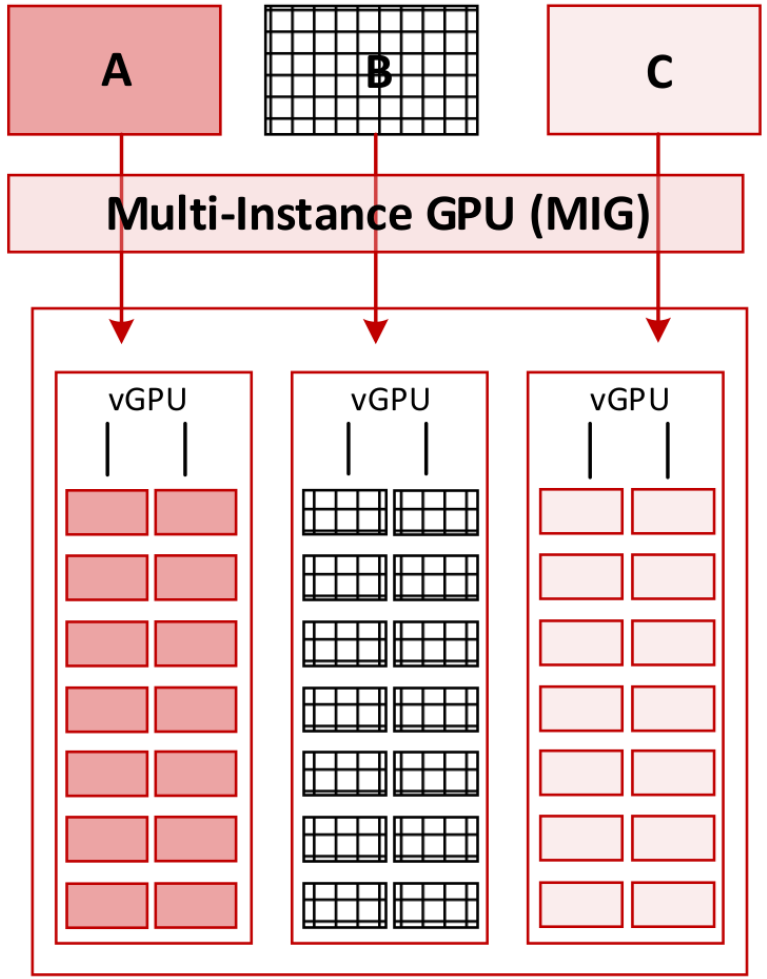
sharing resources on (NVIDIA) GPUs



- most straightforward
- time-multiplexing
- × limited parallelism

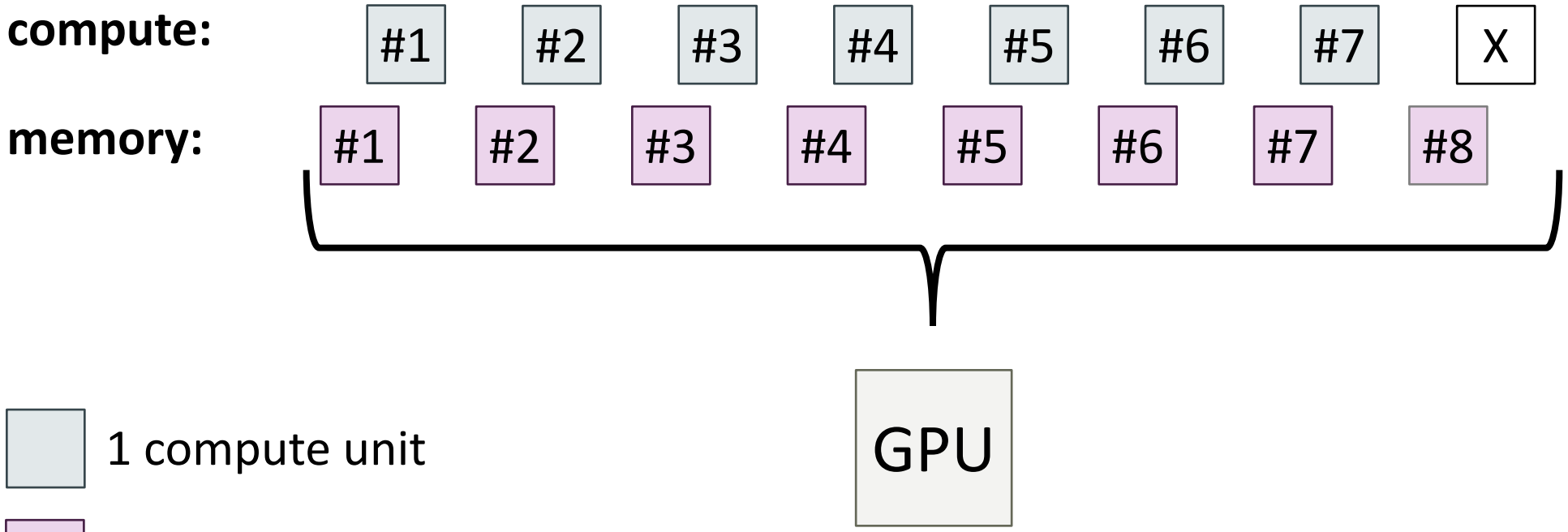


- finer-grained sharing
- × higher chances of interference



- hardware-support for resource split
- × rigid partitioning

multi-instance GPU



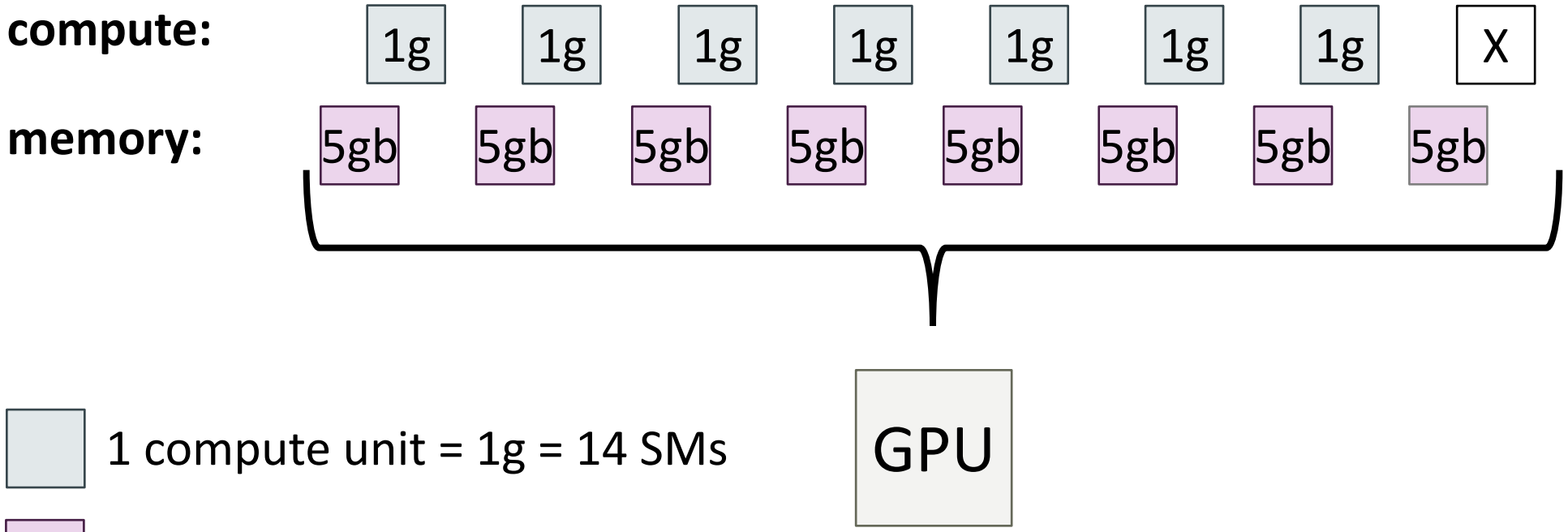
1 compute unit

1 memory unit

unused available (memory/compute) unit

X unavailable compute unit

multi-instance GPU on A100 (40GB)



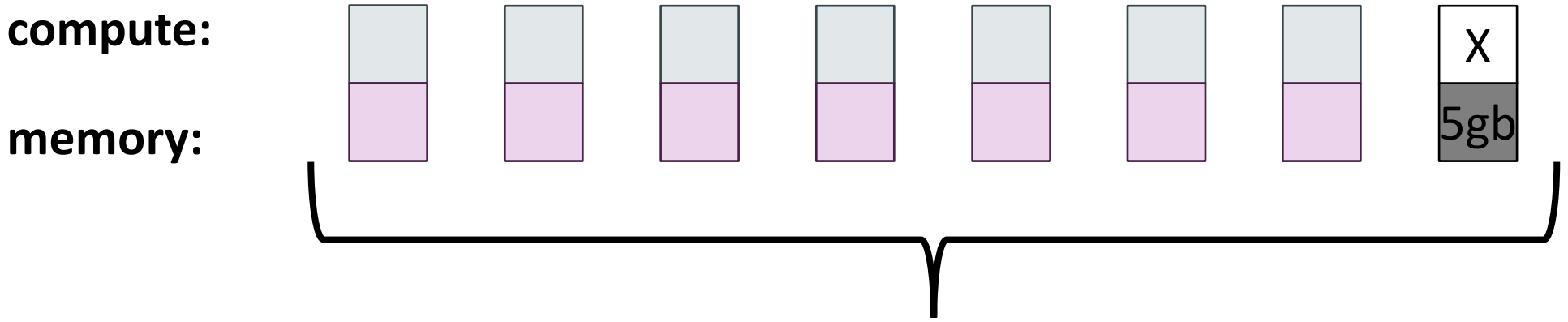
1 compute unit = 1g = 14 SMs

1 memory unit = 5GB

unused available (memory/compute) unit

X unavailable compute unit = 10 SMs (streaming multiprocessor)

multi-instance GPU on A100 (40GB)



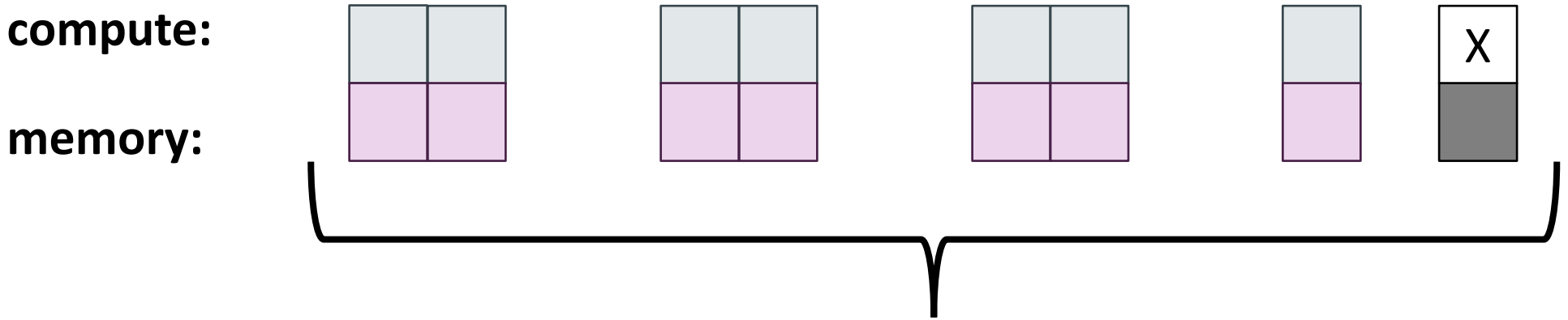
1 compute unit = 1g = 14 SMs

1 memory unit = 5GB

unused available (memory/compute) unit

X unavailable compute unit = 10 SMs (streaming multiprocessor)

multi-instance GPU on A100 (40GB)



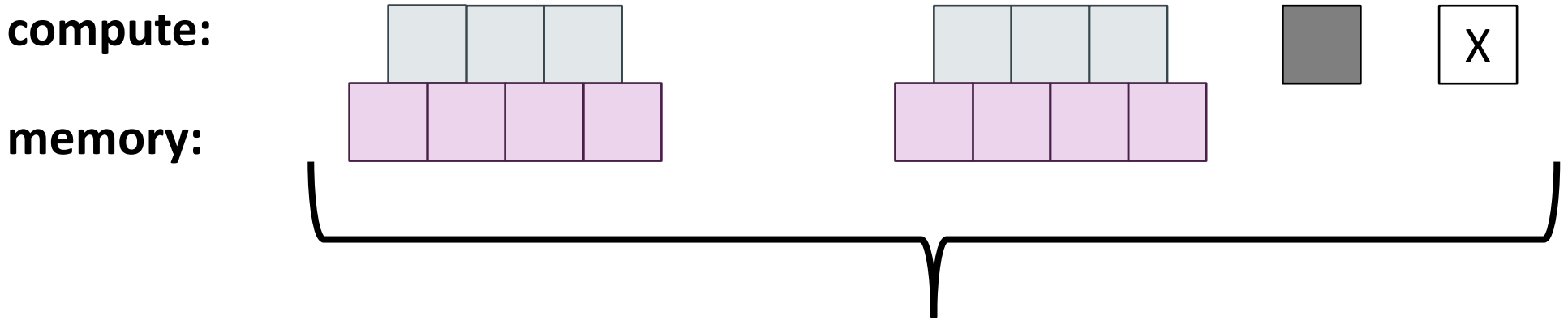
1 compute unit = 1g = 14 SMs

1 memory unit = 5GB

unused available (memory/compute) unit

X unavailable compute unit = 10 SMs (streaming multiprocessor)

multi-instance GPU on A100 (40GB)



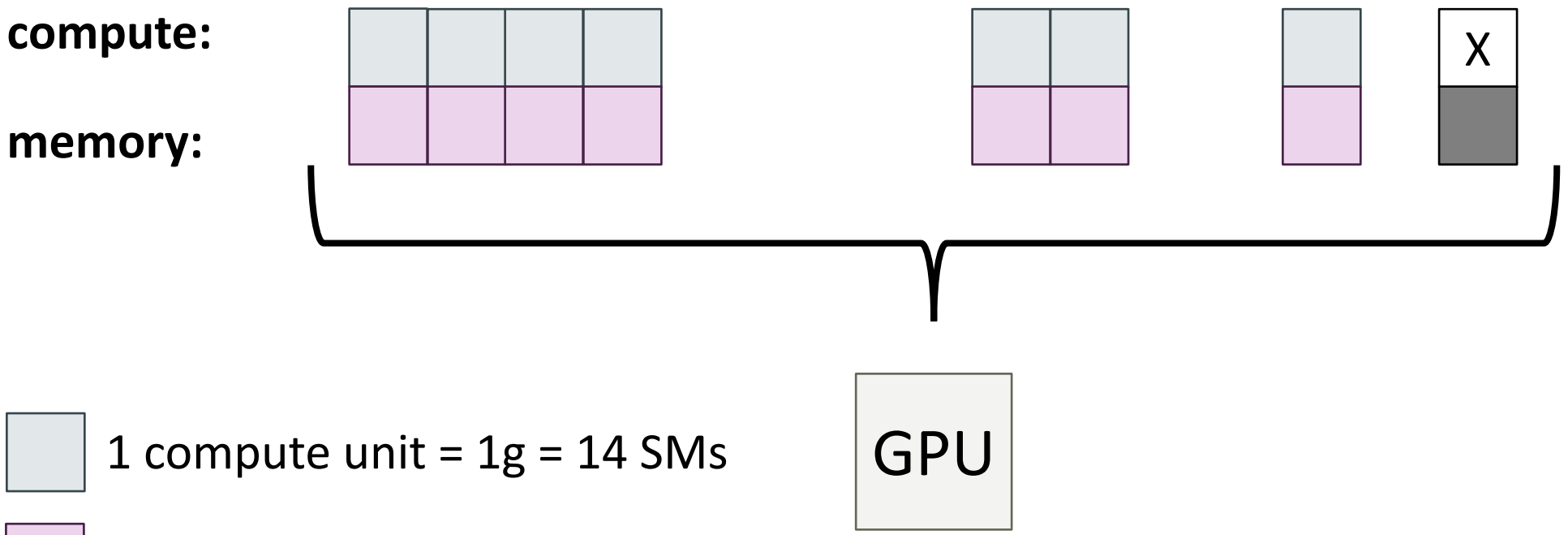
1 compute unit = 1g = 14 SMs

1 memory unit = 5GB

unused available (memory/compute) unit

X unavailable compute unit = 10 SMs (streaming multiprocessor)

multi-instance GPU on A100 (40GB)



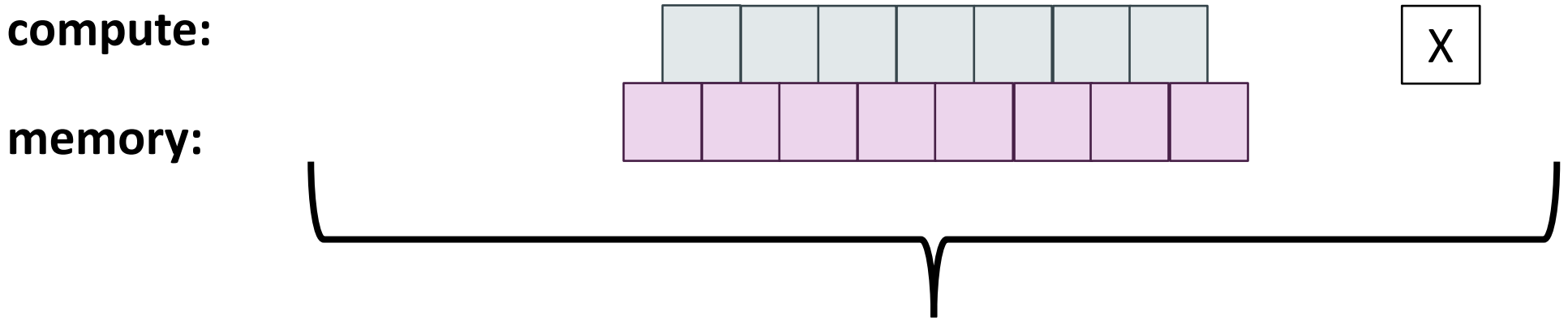
1 compute unit = 1g = 14 SMs

1 memory unit = 5GB

unused available (memory/compute) unit

X unavailable compute unit = 10 SMs (streaming multiprocessor)

multi-instance GPU on A100 (40GB)



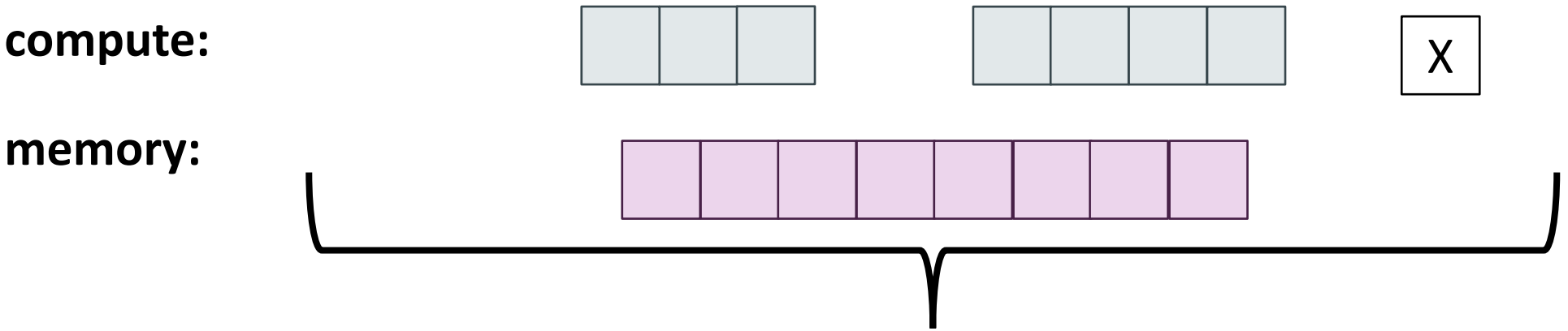
1 compute unit = 1g = 14 SMs

1 memory unit = 5GB

unused available (memory/compute) unit

X unavailable compute unit = 10 SMs (streaming multiprocessor)

multi-instance GPU on A100 (40GB)



1 compute unit = 1g = 14 SMs

1 memory unit = 5GB

unused available (memory/compute) unit

X unavailable compute unit = 10 SMs (streaming multiprocessor)

performance impact of collocation

NVIDIA DGX Station A100

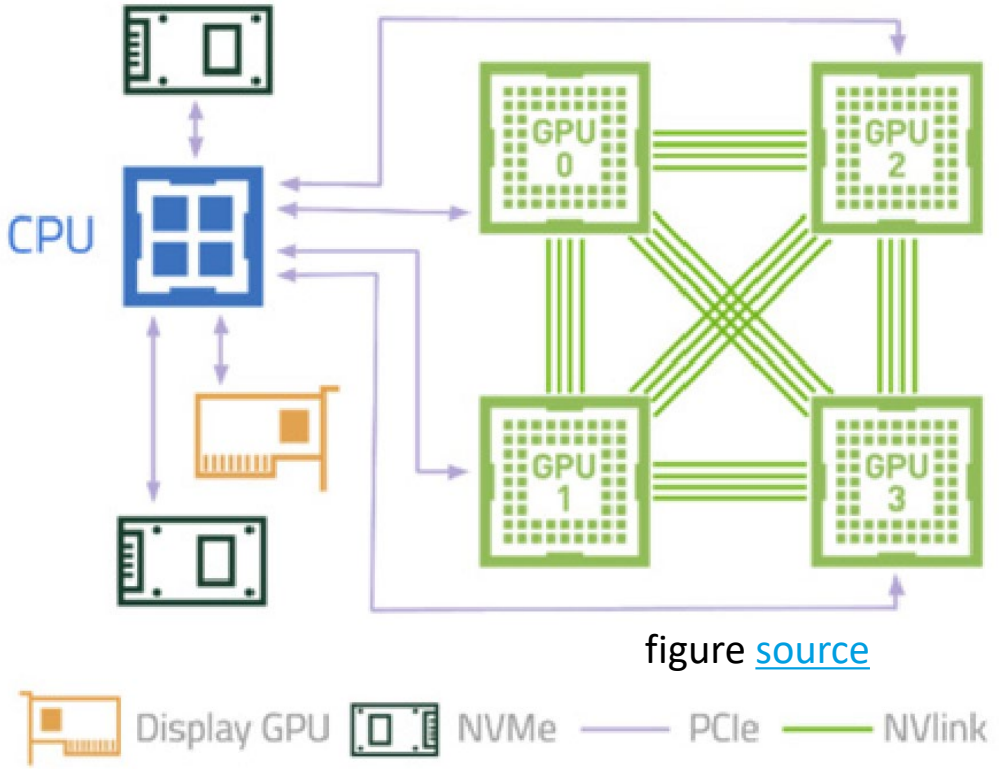


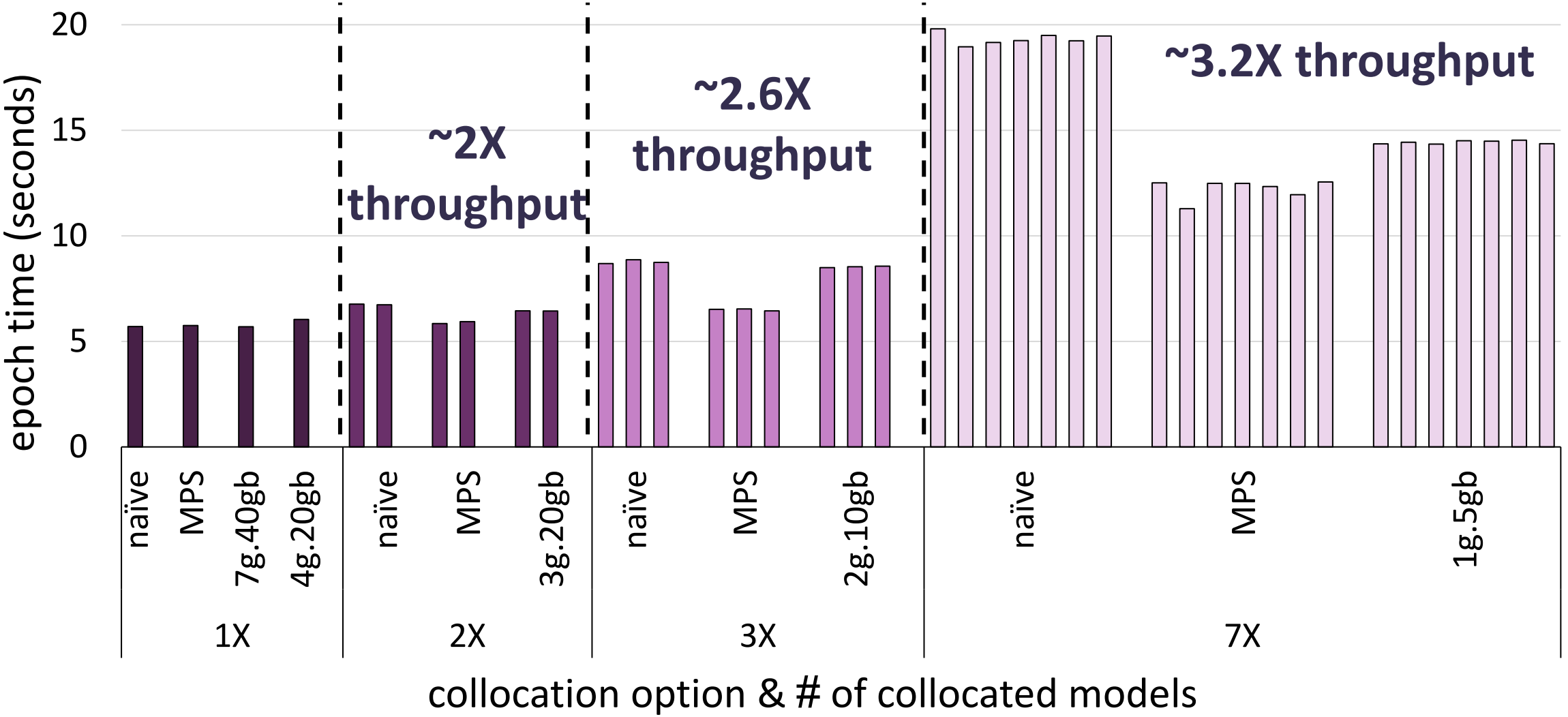
figure source

CPU = AMD 7742 – 512 GB RAM
64 physical cores
GPU = NVIDIA A100 – 40 GB RAM

workloads	model	batch size	dataset
small	**ResNet26** EfficientNet	128	CIFAR-10
medium	**ResNet50** EfficientNet	128	downsampled ImageNet*
large	**ResNet152** CaiT	32 128	ImageNet (2012)
xlarge	**DLRM**	1	Criteo Terabyte

- image models: CNN & transformers recommender model
- on single GPU with PyTorch v2.0
- results reported from 2nd epoch of training

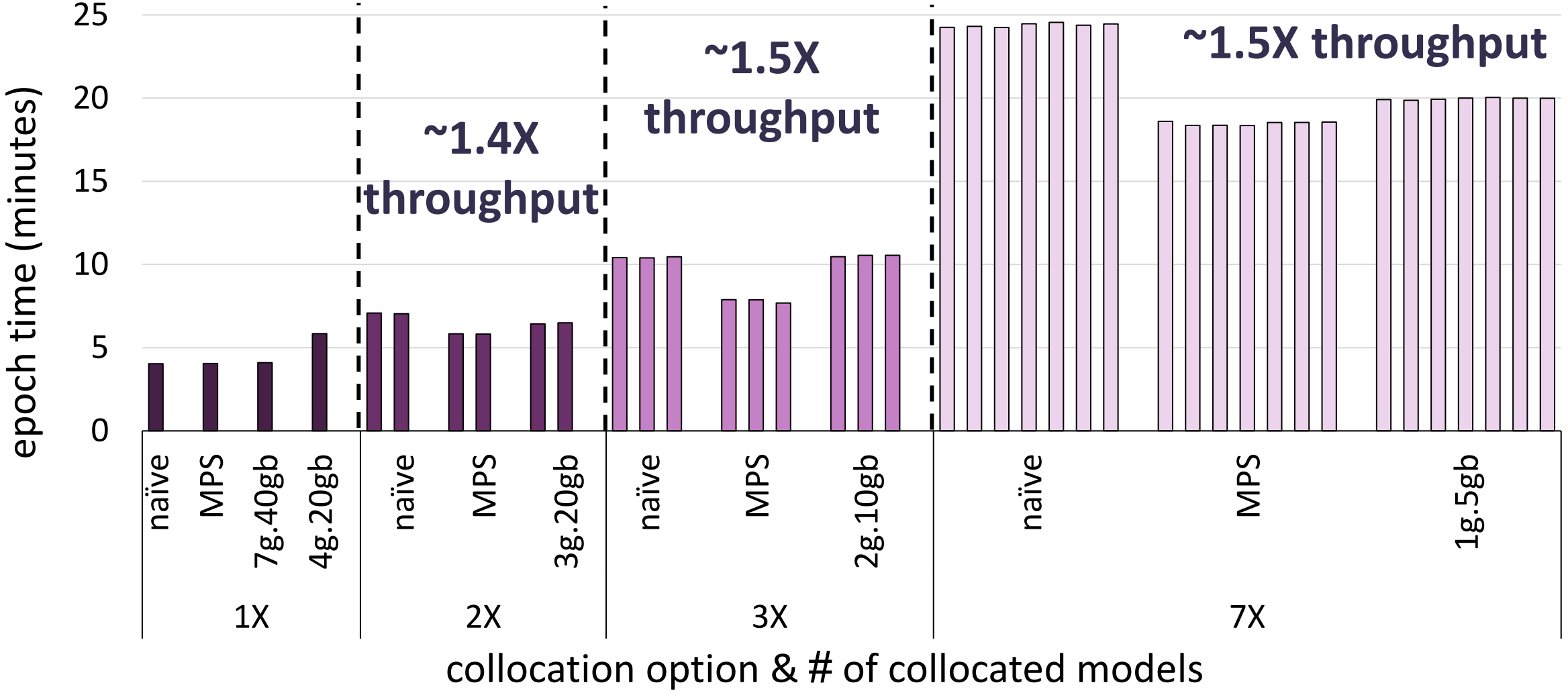
small case – ResNet26



collocation benefits despite increased epoch time

MPS > MIG > naïve

medium case – ResNet50

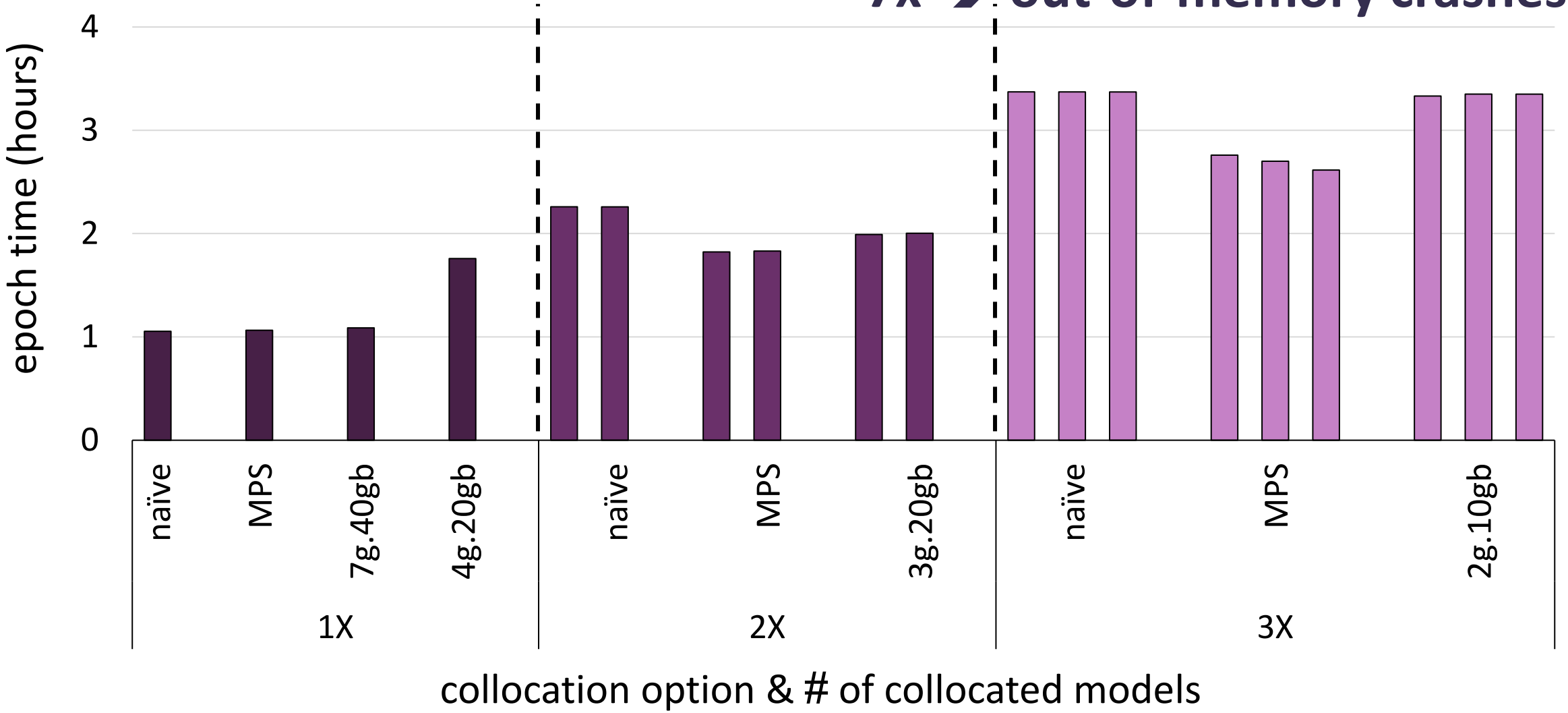


still some throughput benefits

but diminishing returns for increased collocation

large case – ResNet152

7x ➔ out-of-memory crashes



no more throughput benefits – 80% utilization when training alone

better to collocate with smaller or less compute heavy tasks

mixed workloads: compute- & memory-heavy

	DLRM time per training block	ResNet152 time per epoch	sm activity	memory footprint
DLRM alone			5%	29.14 GB
ResNet152 alone			82%	8.47 GB

mixed workloads: compute- & memory-heavy

	DLRM time per training block	ResNet152 time per epoch	sm activity	memory footprint
DLRM alone			5%	29.14 GB
ResNet152 alone			82%	8.47 GB
MPS			81%	37.62 GB

mixed workloads: compute- & memory-heavy

	DLRM time per training block	ResNet152 time per epoch	sm activity	memory footprint
DLRM alone	5.36 h	-	5%	29.14 GB
ResNet152 alone	-	1.05 h	82%	8.47 GB
MPS			81%	37.62 GB

mixed workloads: compute- & memory-heavy

	DLRM time per training block	ResNet152 time per epoch	sm activity	memory footprint
DLRM alone	5.36 h	-	5%	29.14 GB
ResNet152 alone	-	1.05 h	82%	8.47 GB
MPS	5.57 h (+5%)	1.10 h (+4%)	81%	37.62 GB

collocation can lead to (almost) free lunch
when large models stress different hardware resources

workload collocation for model training

- analysis of GPU resource sharing primitives

[An Analysis of Collocation on GPUs for Deep Learning Training]
Ties Robroek, Ehsan Yousefzadeh-Asl-Miandoab, Pınar Tözün.
EuroMLSys 2024

- collocation-aware resource management

[CARMA: Collocation-Aware Resource Manager]
Ehsan Yousefzadeh-Asl-Miandoab, Reza Karimzadeh, Bulat Ibragimov, Florina M Ciorba, Pınar Tözün.

collocation-aware resource management

requirements:

(1) don’t overload the GPU compute
➔ degrades performance

(2) minimize & recover from out-of-memory crashes

CARMA: collocation-aware resource manager

submit training task

plug & play

monitoring

nvidia-smi

dcgmi

recovery queue

tasks' queue

parser

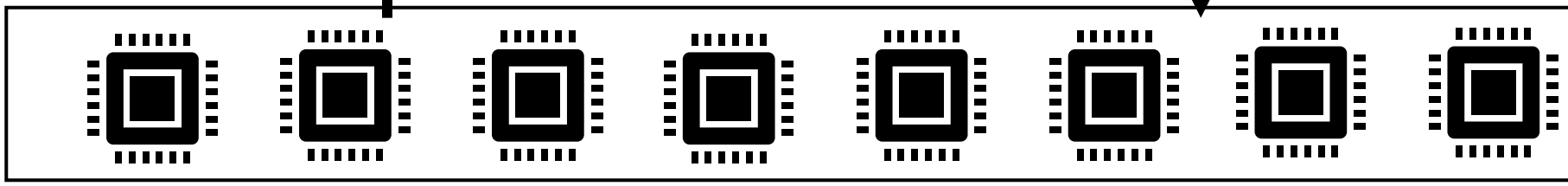
memory estimator

Horus,
Fake Tensors,
GPUMemNet …

most available memory,
least utilized GPU,
random …

utilization information

mapping decisions



CARMA evaluation

trace mix

- ***heavy:*** 1-2GPUs, epoch time: 7.5mins to >1hour, memory: 9GB to 30GB
  XLNet (base & large), BERT (base & large), GPT2 (large)
- ***medium:*** 1 GPU, epoch time: 1min to >1hour, memory: 1GB to 30GB
  EfficientNet, ResNet (50), MobileNet, VGG …
- ***light:*** 1 GPU, epoch time: up to 1min, memory: up to 1GB
  ResNet (18, 34), MobileNet (small) …

2 traces of 60 training tasks:

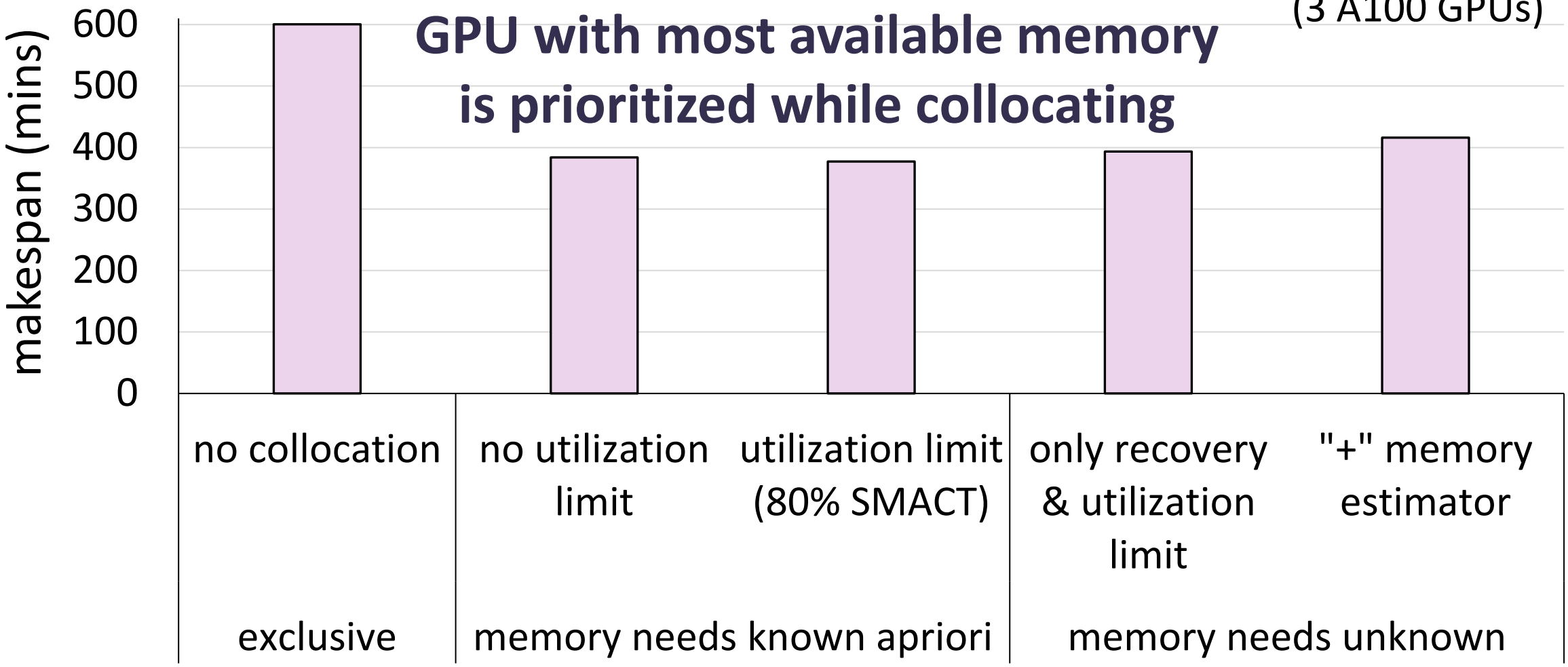
- 30% light, 60% medium/heavy, and 10% heavy 2 GPU models

based on real-world workload trace analysis

- task submission times: "Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads." ATC 2019
- task distribution: "An Empirical Study on Low GPU Utilization of Deep Learning Jobs." TPDS 2022

CARMA on a training workload trace

on NVIDIA DGX Station
(3 A100 GPUs)



30-37% reduction in end-to-end trace completion time leads to ~15% reduction in total energy need

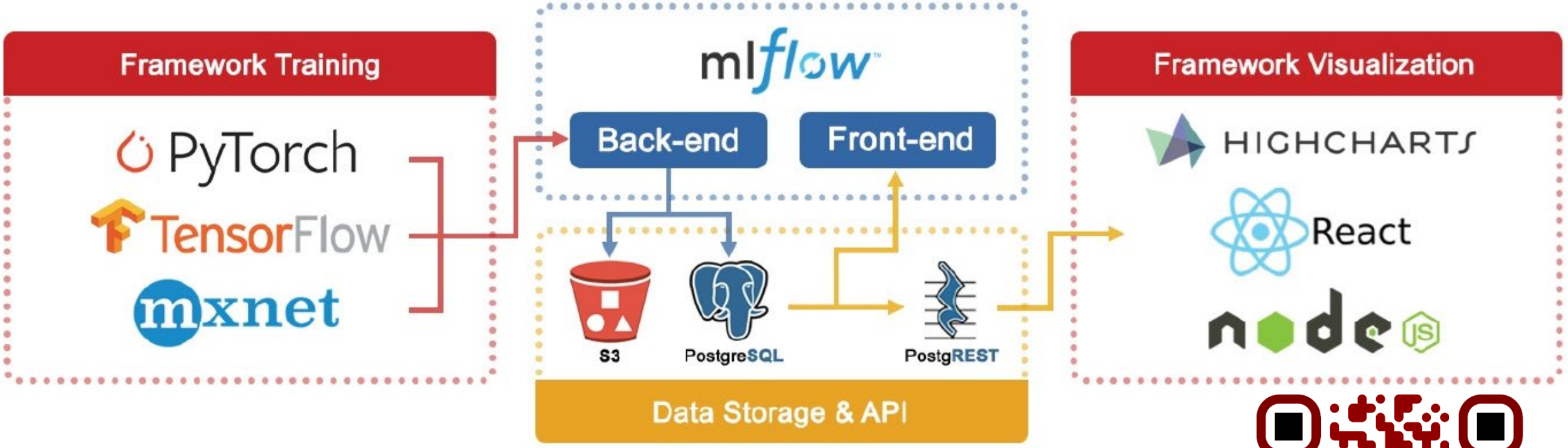
workload collocation for model training

- not all training needs all the resources of a single GPU
- collocation on GPUs benefits when the aggregate compute & memory needs of the collocated training runs fit in the GPU
- a collocation-aware resource manager help reduce time & energy required for a training workload with various models

need to build collocation-aware resource managers for deep learning targeting both small & large scales!

how to monitor hardware? – radT

[DEEM'23]



➔ easy, extensible, and scalable tracking of hardware metrics (GPU utilization, storage access, carbon footprint …)

➔ frontend for data exploration

RAD

used by our group & data scientists @ITU for systematic benchmarking of deep learning training

RAD - resource-aware data systems

IT UNIVERSITY OF COPENHAGEN

postdocs

Ties Robroek

Ehsan Yousefzadeh-Asl-Miandoab

phd students

Robert Bayer

Jens Birk Andersen

collaborators

Pamela Delgado HEIG-VD

Tilmann Rabl HPI

Ana Klimovic ETH

Julian Priest ITU

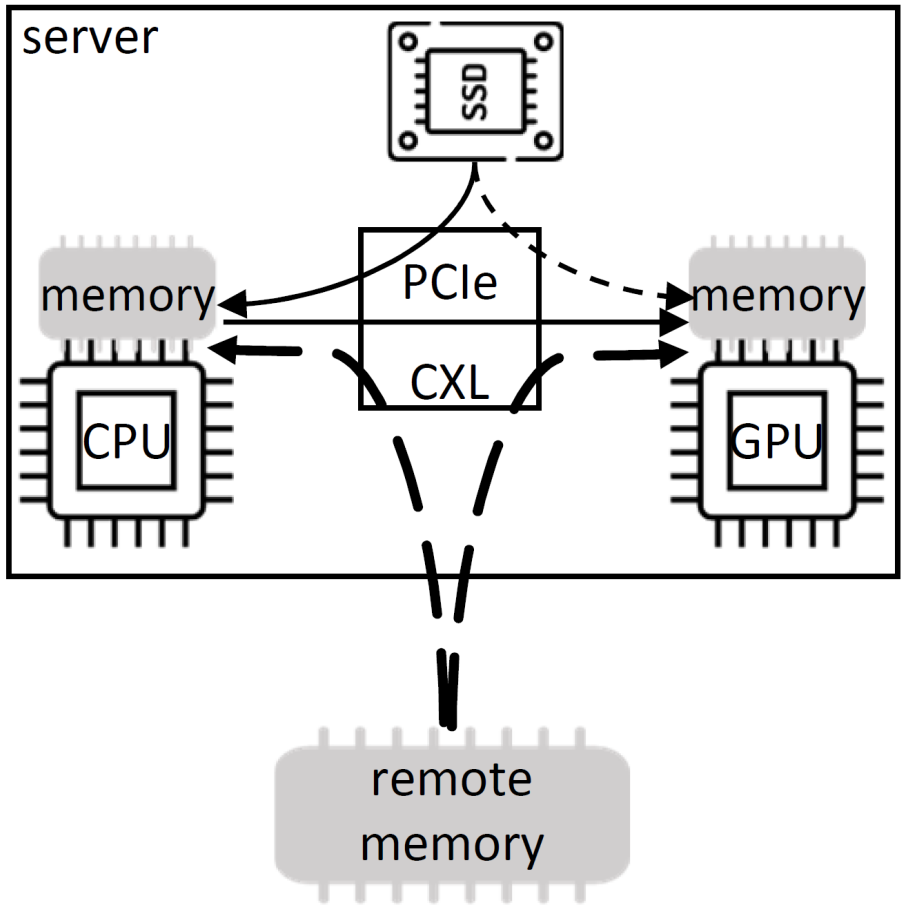
SAMSUNG

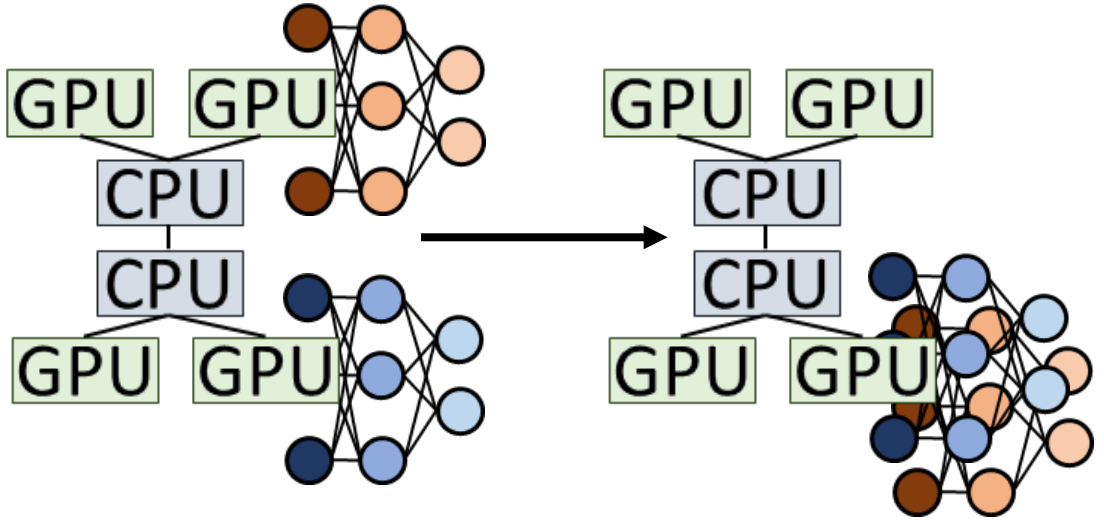
RAD

rad.itu.dk

- can we utilize modern hardware well?
- can we do more with less?

impact of storage hierarchy on data-intensive systems





workload collocation on GPUs

data management & processing on tiny hardware @ the edge

workload collocation for model training

thank you!

- not all training needs all the resources of a single GPU
- collocation on GPUs benefits when the aggregate compute & memory needs of the collocated training runs fit in the GPU
- a collocation-aware resource manager help reduce time & energy required for a training workload with various models

need to build collocation-aware resource managers for deep learning targeting both small & large scales!