

satisfying the data monster with fewer resources

a quest to feed the GPU in deep learning training

Pinar Tözün

Associate Professor
IT University of Copenhagen



unsustainable growth of deep learning

2023

Gemini Ultra

GPT-4

PaLM (540B)

GPT-3 175B (davinci)

Megatron-Turing NLG 530B

Llama 2 70B

LaMDA

UNIVERSITY of WASHINGTON

RoBERTa Large

BERT-Large

Transformer

~5 orders of magnitude
increase in training cost.

~7 orders of magnitude growth
in computational footprint.

2017

Transformer

BERT-Large

RoBERTa Large

UNIVERSITY of WASHINGTON

GPT-3 175B (davinci)

Megatron-Turing NLG 530B

Llama 2 70B

LaMDA

RoBERTa Large

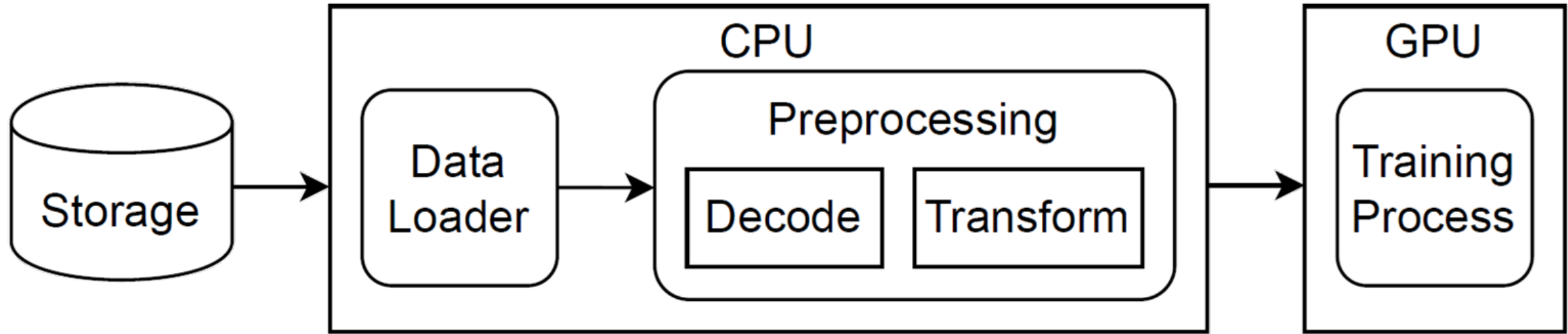
BERT-Large

Transformer

Training cost (in U.S. dollars - log scale)

Training compute (petaFLOP - log scale)

journey of data in deep learning training



CPU feeds the accelerators

- 16-64 cores per GPU (recommended)
- 96 cores per TPU*

➔ otherwise, accelerator may be underutilized

➔ can we do more with fewer CPUs & less of the CPU?

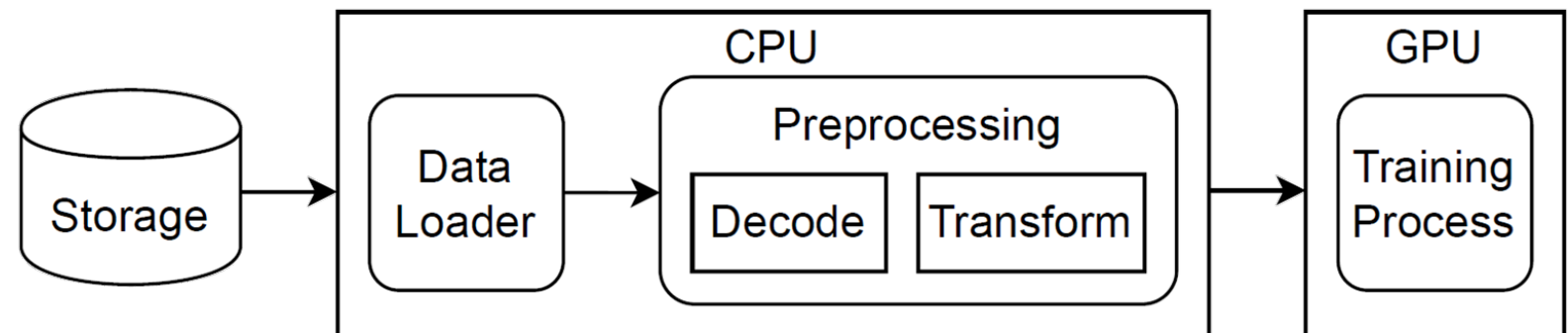
deep learning with less hardware

[Path to GPU-Initiated I/O for Data-Intensive Systems](#)

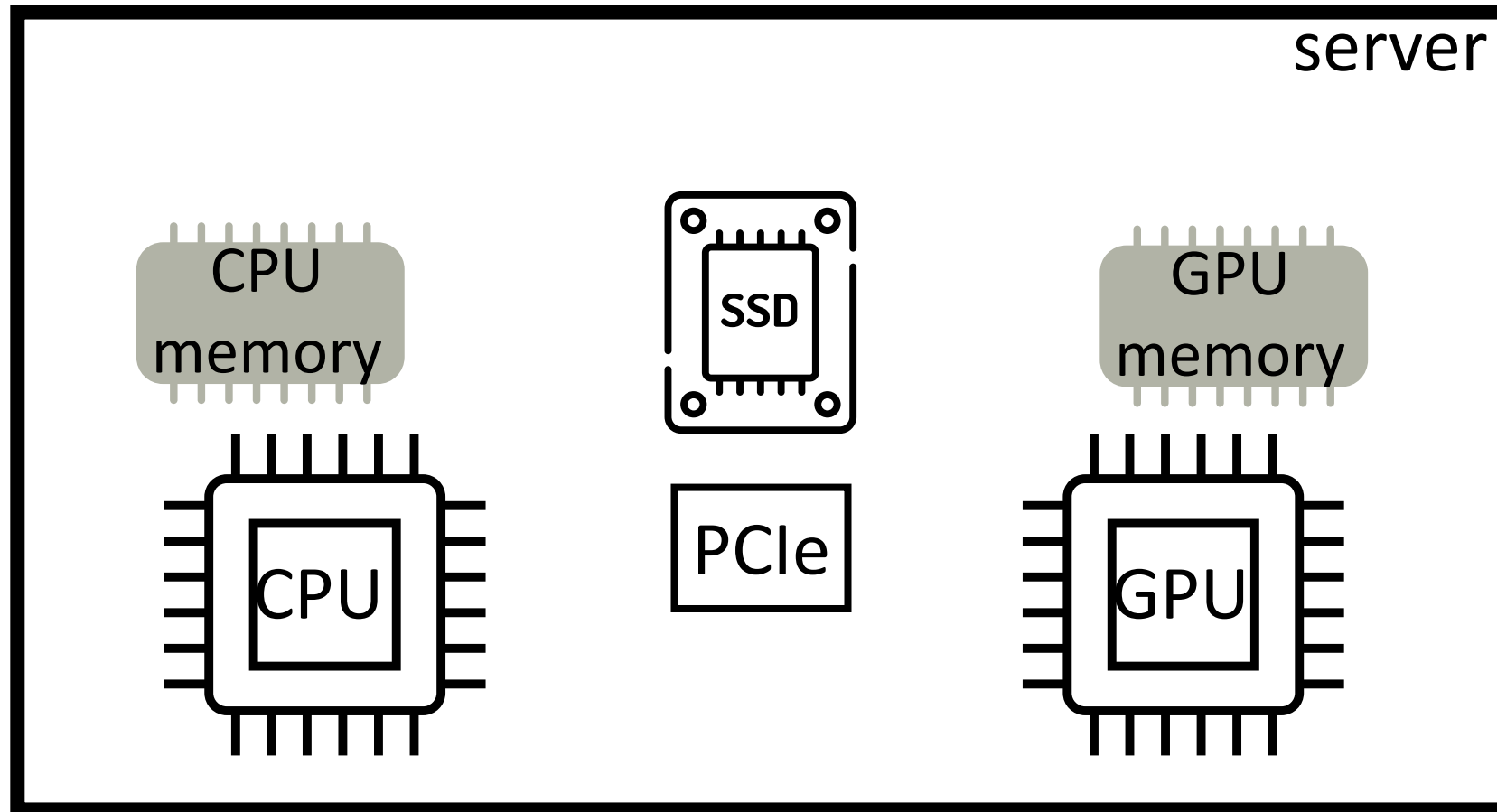
Karl B. Torp, Simon Lund, Pinar Tözün.

DaMoN 2025

- GPU-centric I/O path
- data & work sharing
- impact of data selection

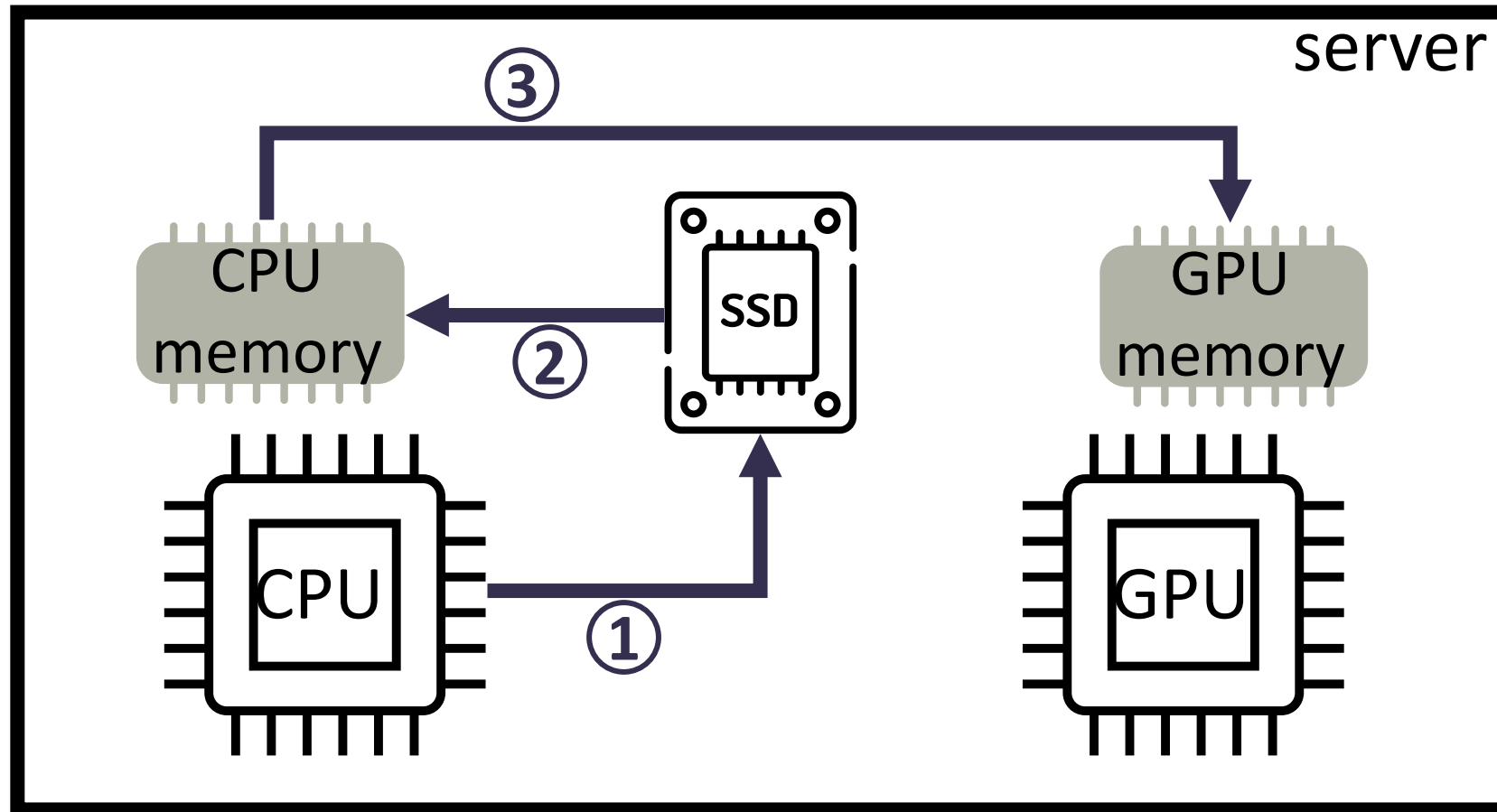


target hardware setup



* PCIe is dropped in the remaining figures for the sake of simplicity in illustrations.

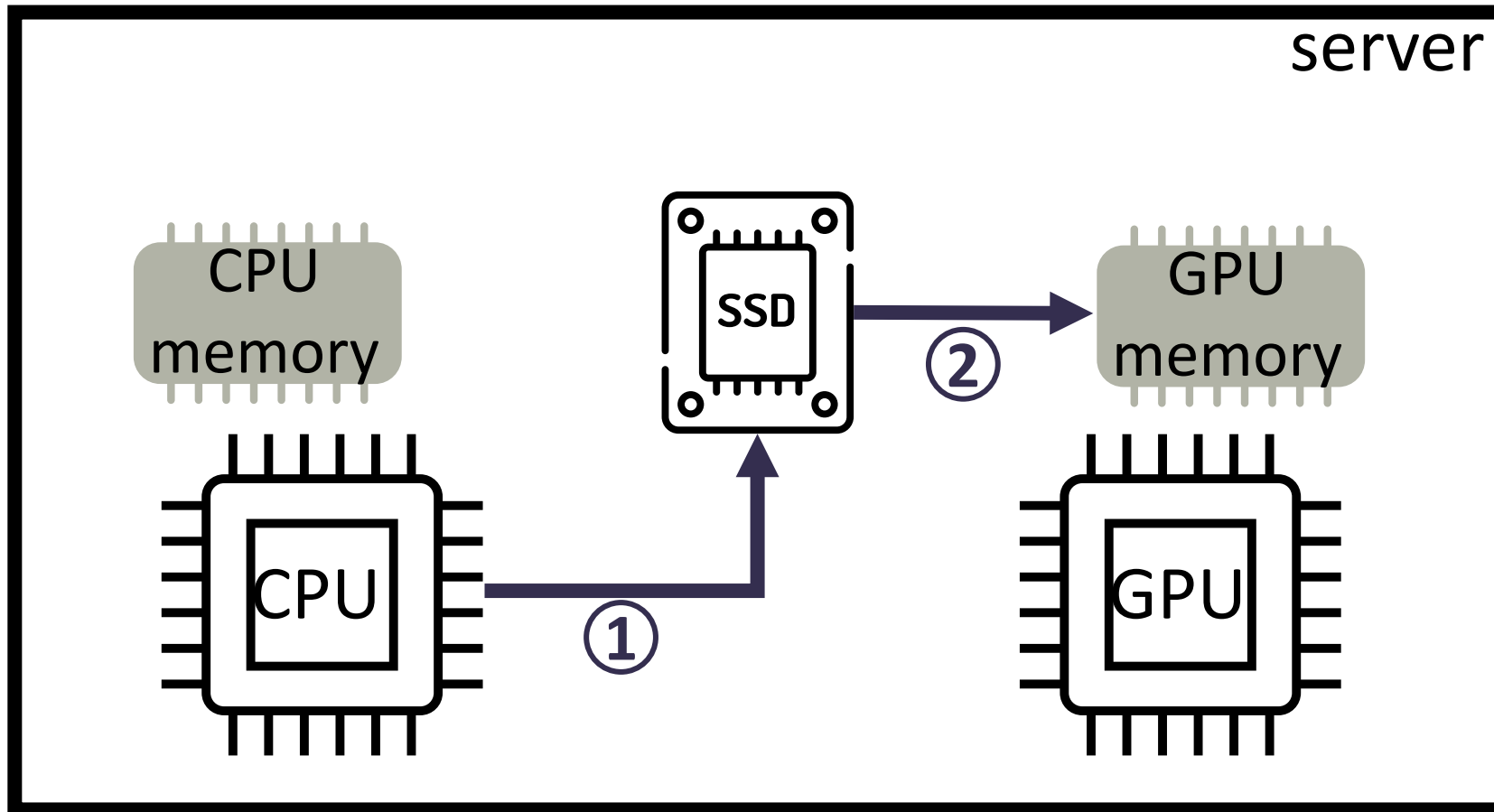
conventional: CPU-centric I/O



- ✓ **ecosystem support**
- × **CPU-bound & overhead from memory copy**

GDS: GPU-centric & CPU-initiated

GPUDirect
[NVIDIA'19]



- ✓ eliminates the extra memory copy
- × still CPU-bound

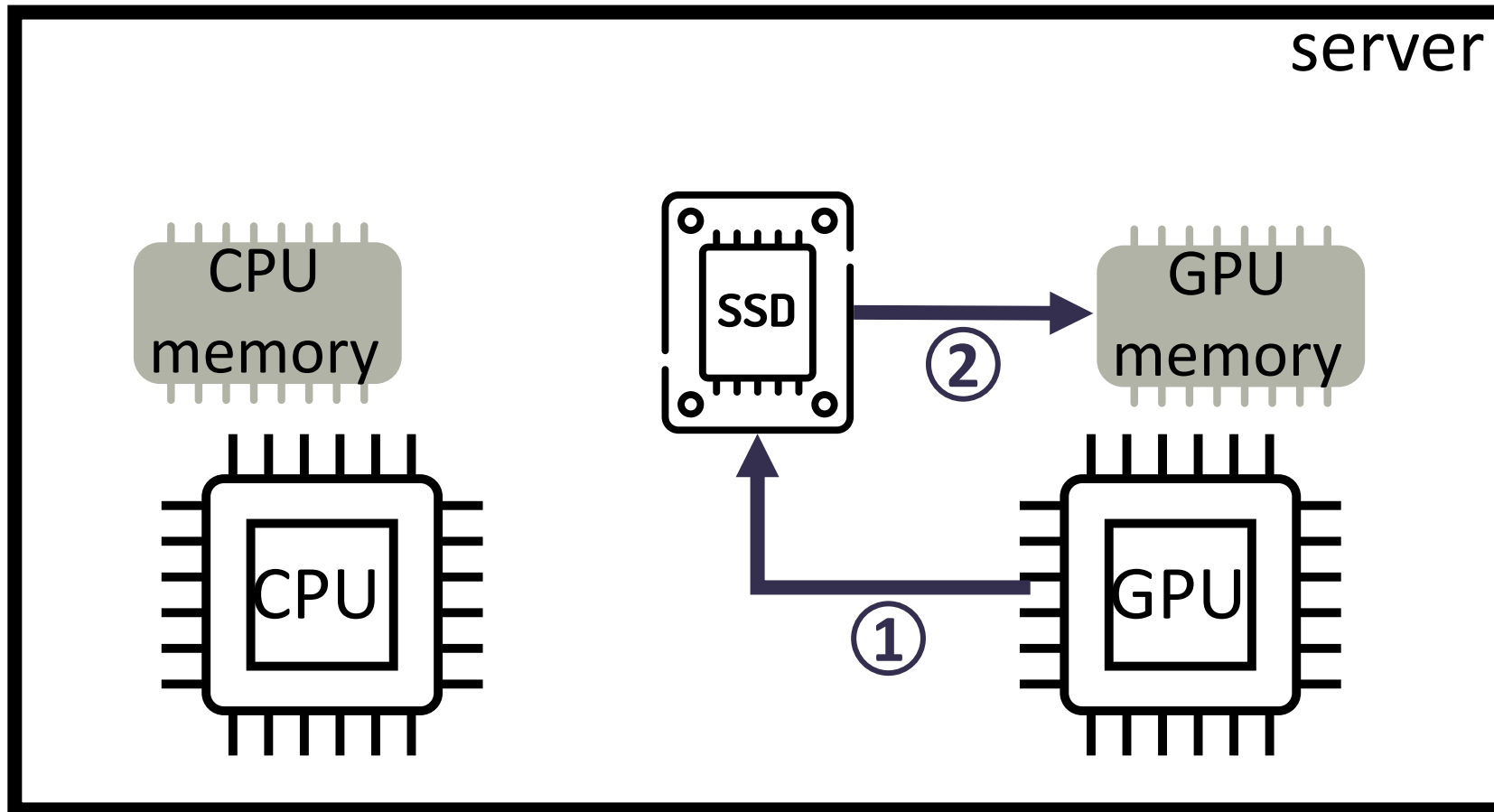
BaM: GPU-centric & GPU-initiated

Big

Accelerator

Memory

[ASPLOS'23]



- ✓ eliminates the CPU on the path
- × ecosystem missing & saturates GPU

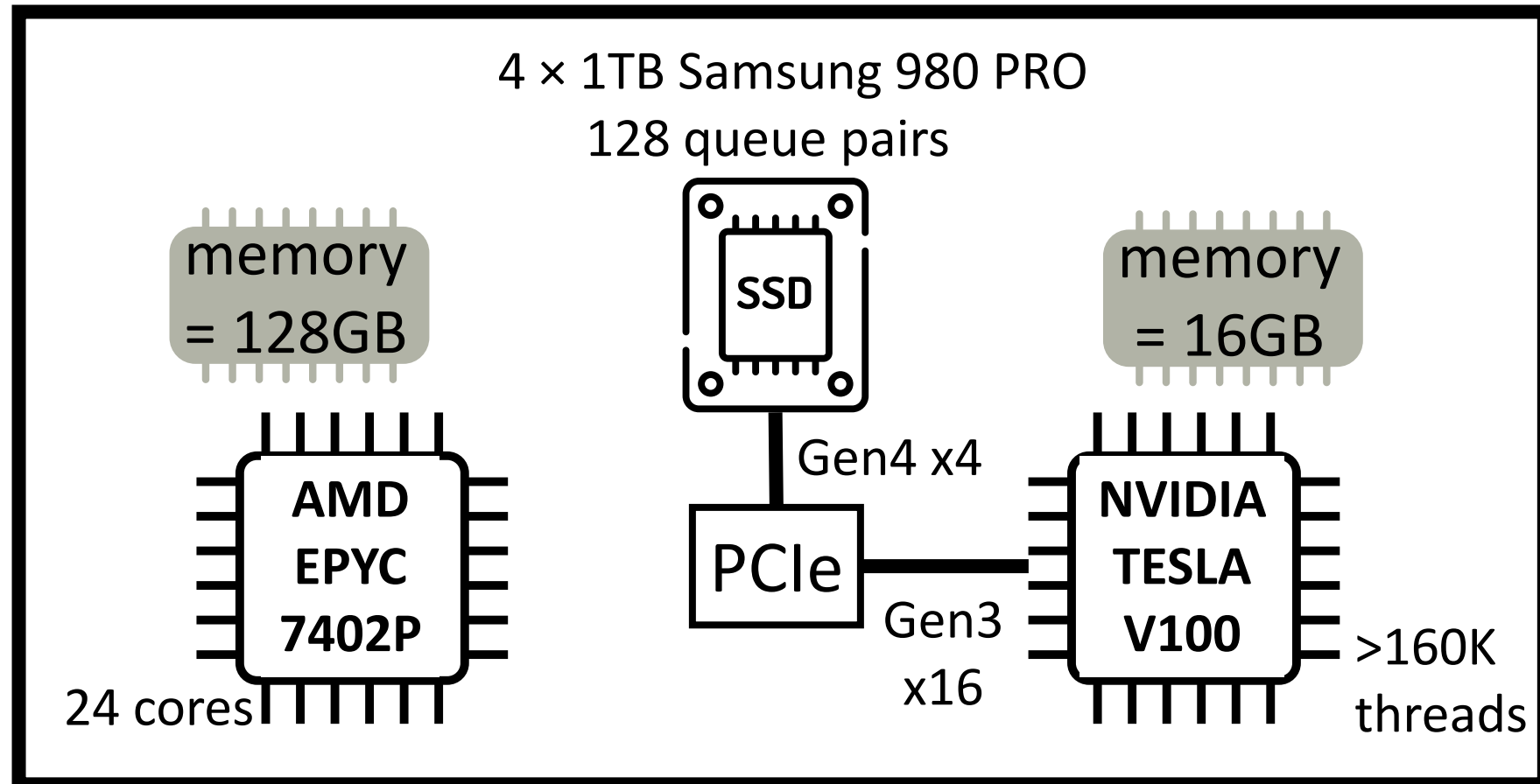
evaluation: CPU- vs GPU-centric I/O

mechanisms: CPU-centric: SPDK & GPU-centric: GDS, BaM

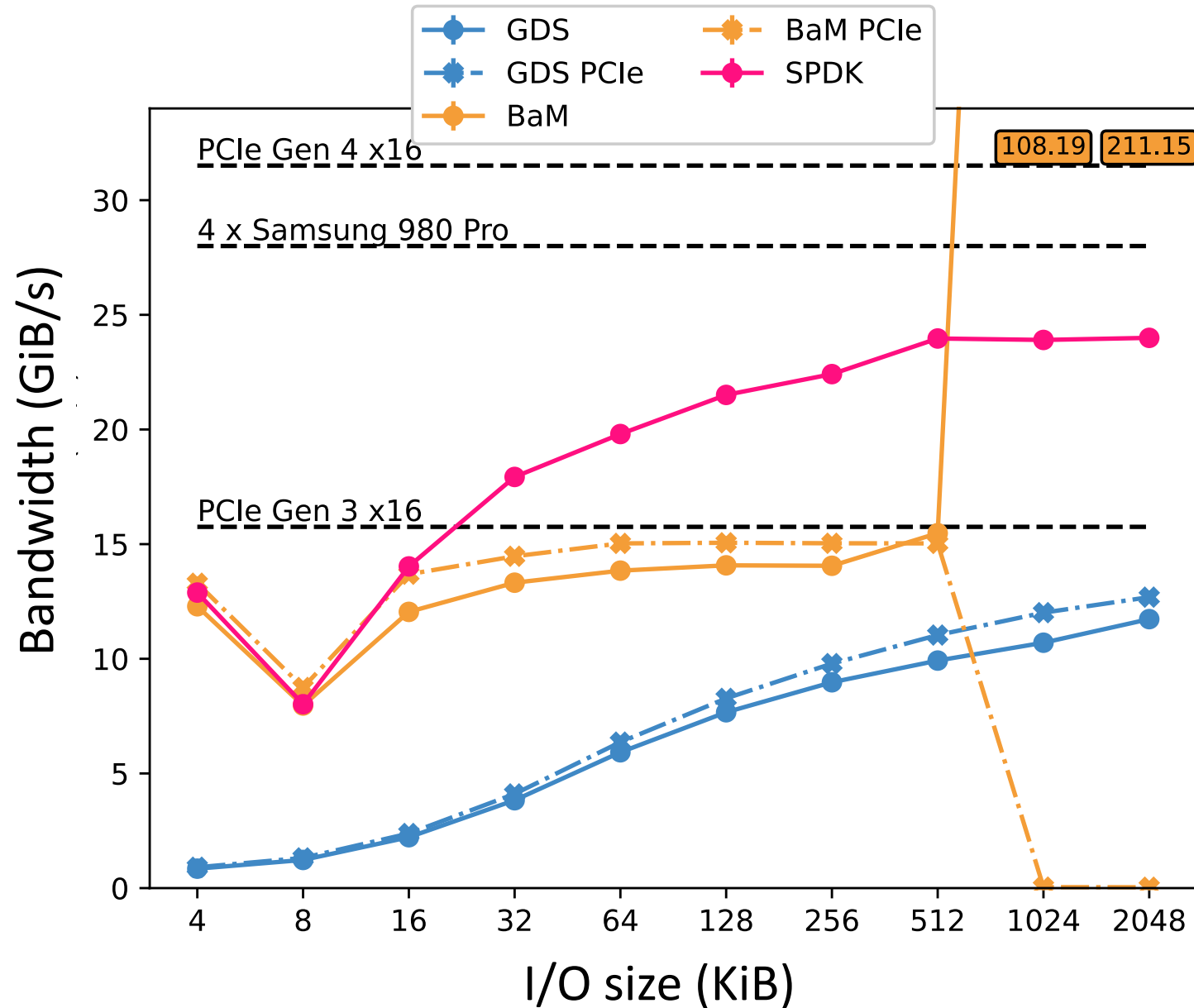
workload: random reads

→ each mechanism has their own tool for benchmarking

hardware



bandwidth utilization – 4 SSDs & PCIe



GDS is CPU-compute heavy.

➔ 16 logical cores utilized

BaM is limited by the PCIe Gen3 link & heavy on the GPU resources.

➔ whole GPU utilized

SPDK is the most resource-efficient but has a longer path to the GPU.

➔ 2 logical cores utilized

path to GPU-centric I/O

- need to reduce the dependency on CPUs for more efficient deep learning pipelines
- GPU-centric I/O is a way to do that & we have the mechanisms today (e.g., GDS, BaM)
 - GDS has dependency on CPUs still
 - BaM requires a lot of GPU resources

→ when to use which mechanism while being resource-aware?

→ how to best integrate them into popular deep learning frameworks (or GPU databases) for wider-scale use?

deep learning with less hardware

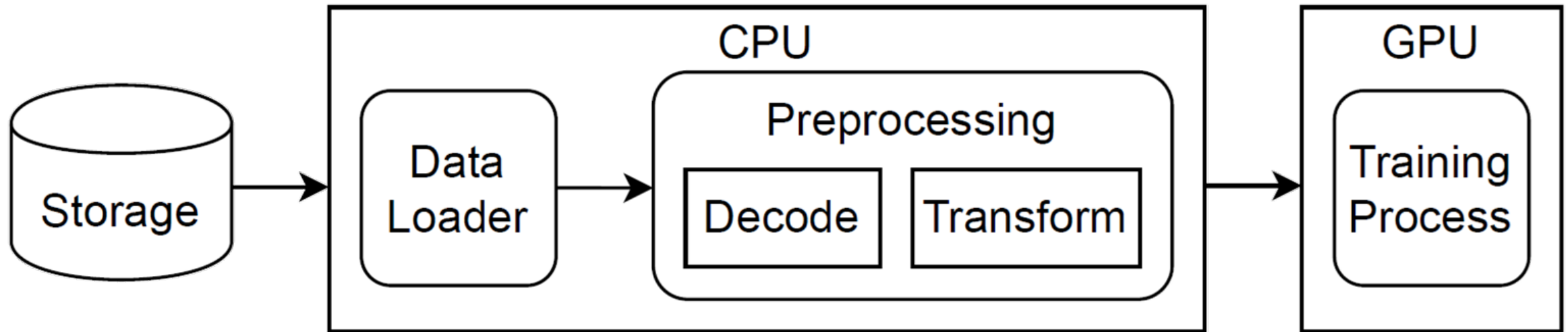
- GPU-centric I/O path
- data & work sharing
- impact of data selection

[TensorSocket: Shared Data Loading for Deep Learning Training](#)

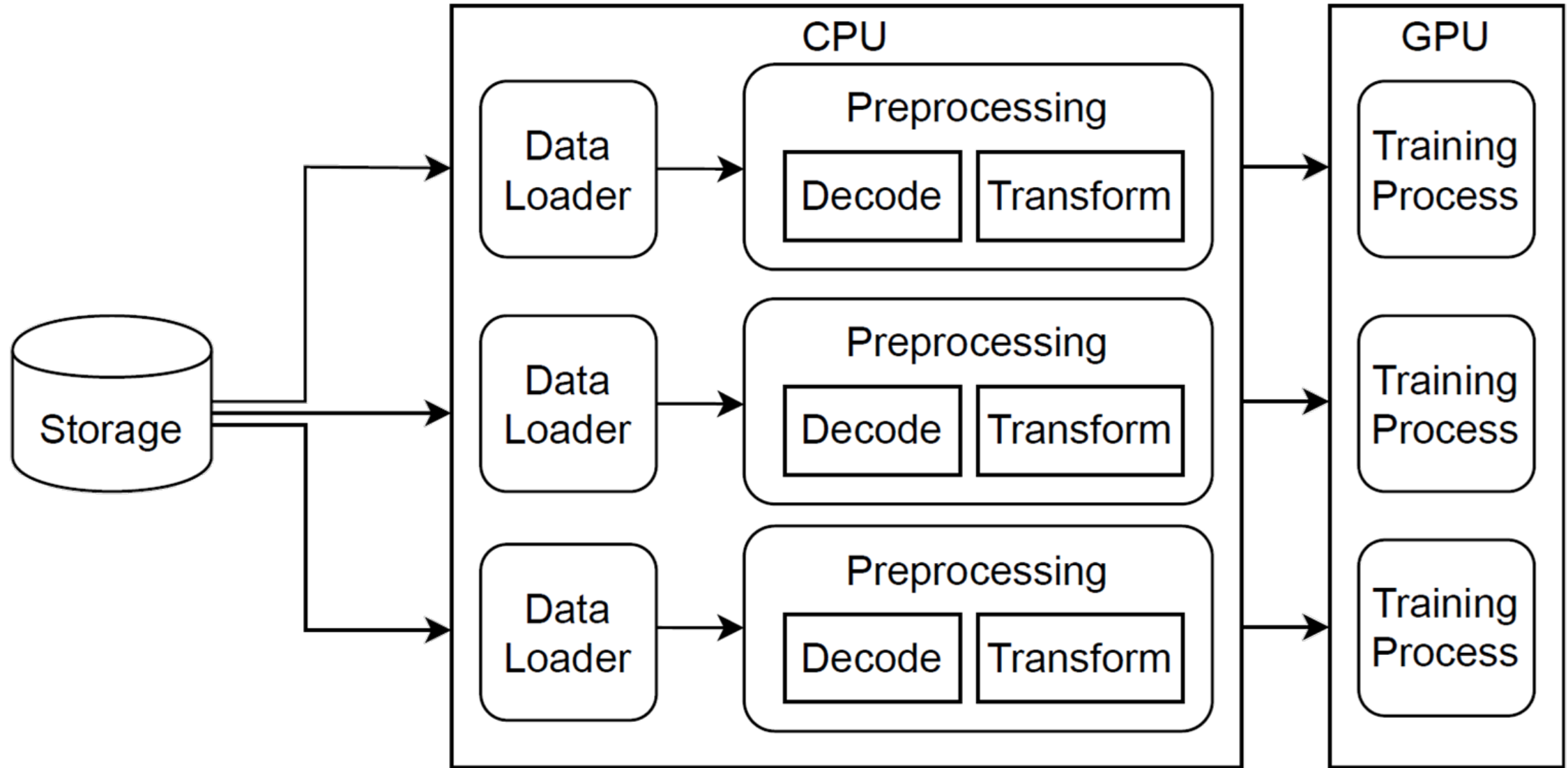
Ties Robroek, Neil Kim Nielsen, Pınar Tözün.

SIGMOD 2026

conventional journey of data while training

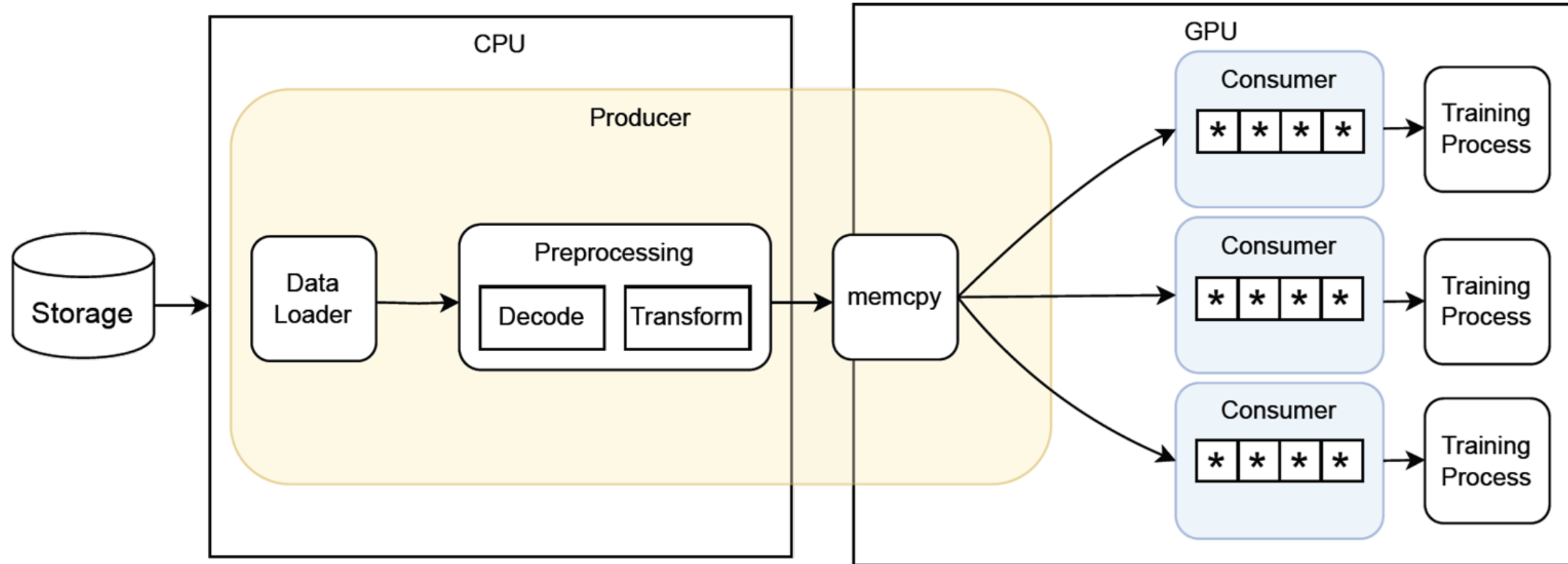


data journey in collocated training



redundant work & CPU use!

data sharing for collocated training

TensorSocket

minimize the redundancy!

TensorSocket requirements & limitations

→ consumers go through the data at the same rate

doesn't mean that consumers cannot ...

- join at different epochs of training
- train at differing speeds
- have different batch sizes

→ target is smaller scale

- collocation of model training on a single server
- models can fit into the memory of a single GPU

not everyone needs “big” models & scale!

for larger scales, check out *tf.data service*, *CoorDL* ...

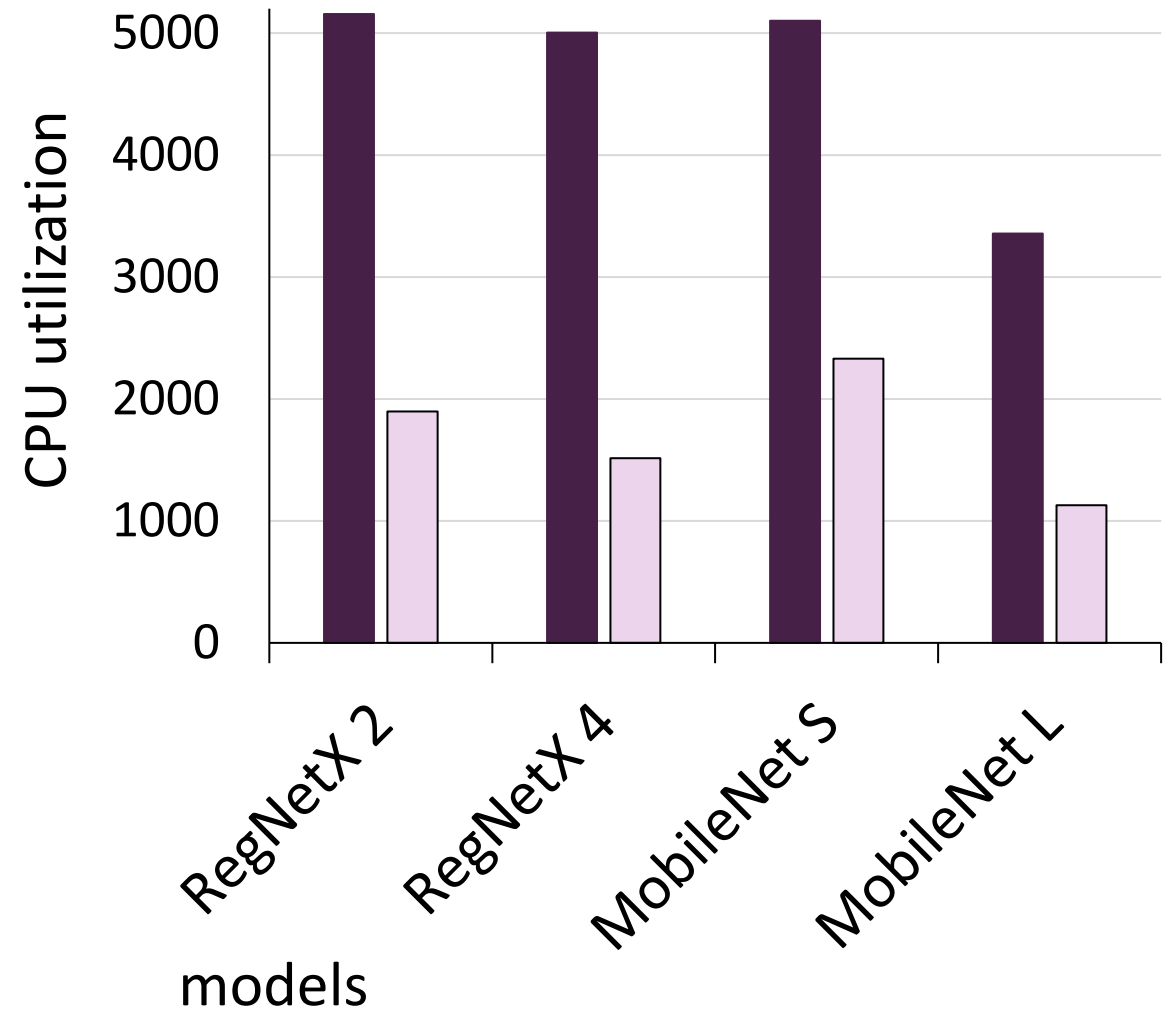
- Varoquaux et al. [Hype, Sustainability, and the Price of the Bigger-is-Better Paradigm in AI](#)
- Margot Seltzer, SIGMOD'25 keynote

[SoCC'23]

[PVLDB'21]

impact of data sharing

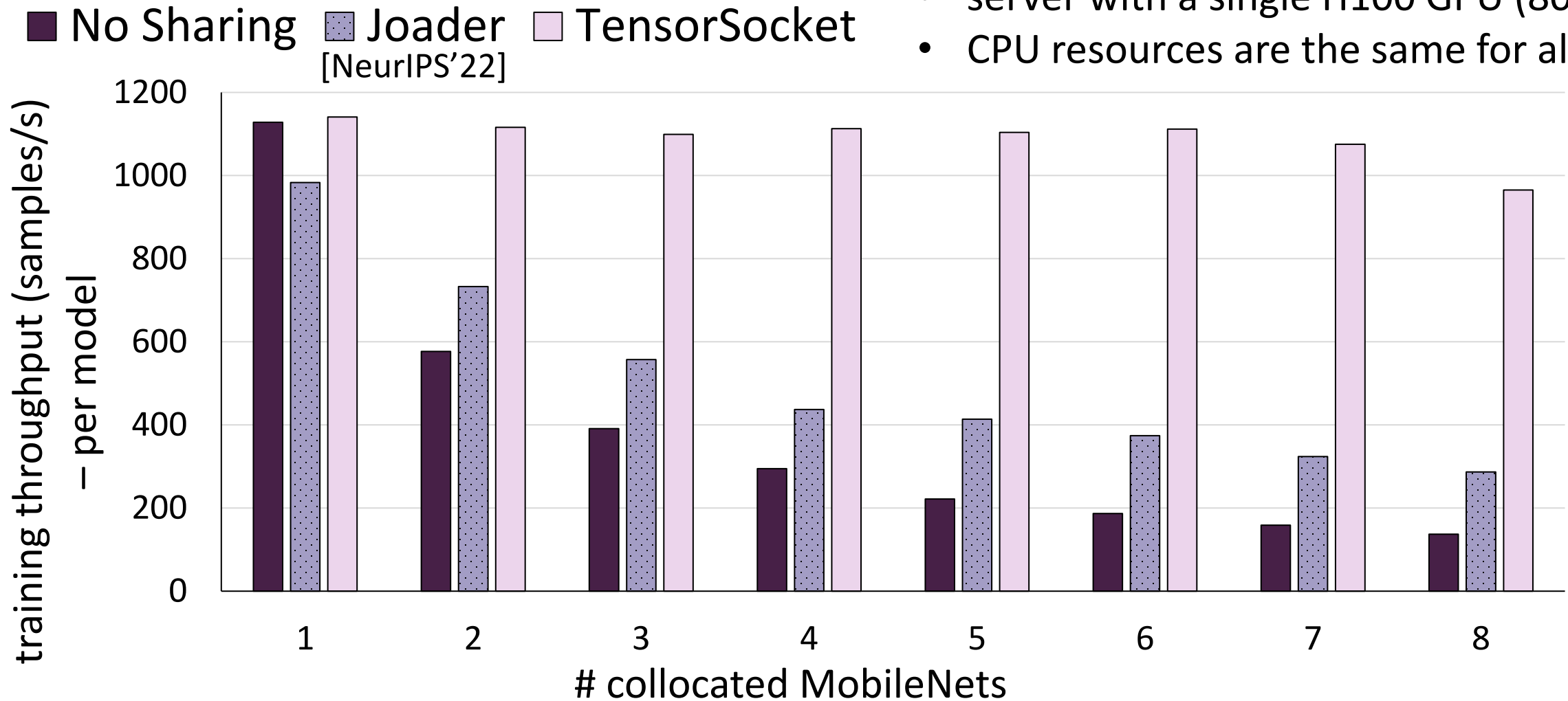
- on PyTorch
- a server with 4 A100 (40GB) GPUs
- one model training on each



higher overall throughput & reduced CPU need!

comparison to other techniques

- server with a single H100 GPU (80GB)
- CPU resources are the same for all



TensorSocket sustains throughput even with GPU collocation & reduces both CPU and GPU needs for the whole workload

sharing for deep learning training

- workload collocation allows data & work sharing
- ***TensorSocket*** enables data & work sharing for collocated training jobs on the same dataset



**can reduce both the CPU & GPU needs of training
while increasing training throughput**

deep learning with less hardware

- GPU-centric I/O path
- data & work sharing
- impact of data selection

Collaboration started at 2024 Dagstuhl seminar: [Resource-Efficient Machine Learning](#).
Ties Robroek, Maximilian Böther, **Niklas Christiansen**, **Daniel Sepehri**, Dagmar Kainmüller, Theo Rekatsinas, Stefanie Scherzinger, Ana Klimovic, Pınar Tözün.

why data selection?

- reduce the dataset size
- increase model accuracy
- fine-tuning

→ unclear impact on the end-to-end training time!

**→ trade-off: computational complexity
vs “better” data selection**

preliminary results

base model = Llama-3.2-1B-Instruct
dataset = TruthfulQA

server with a single H100 GPU (80GB)

	duration (mins)	GPU energy use (Wh)	accuracy gain over base model
LESS (25%) ^[ICML'24]			
Full			
Random (50%)			
Random (25%)			

preliminary results

base model = Llama-3.2-1B-Instruct
dataset = TruthfulQA

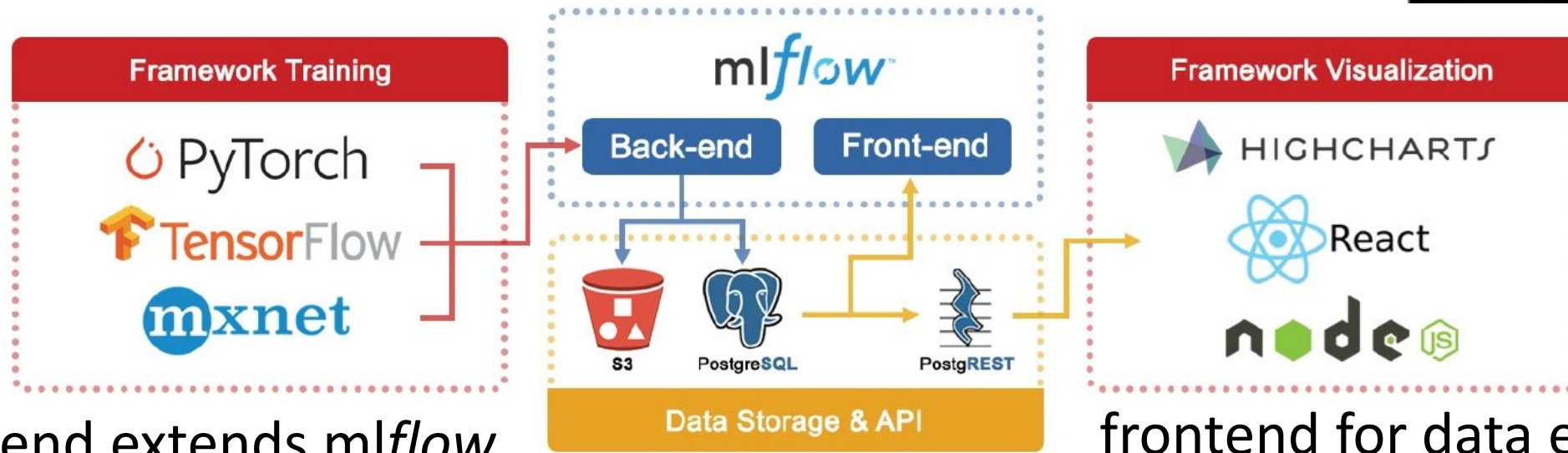
server with a single H100 GPU (80GB)

	duration (mins)	GPU energy use (Wh)	accuracy gain over base model
LESS (25%) ^[ICML'24]	318	1099	80%
Full	84	417	84%
Random (50%)	43	215	63%
Random (25%)	23	118	39%

if the data selection can be used across different fine-tuning processes, the costs may amortize

radT

[DEEM'23]



- backend extends mlflow
- incorporates collocation
- allows easy, extensible, and scalable hardware monitoring
(dcgm, nvidia-smi, top, iostat, carbontracker, nsight systems/compute, pytorch profiler ...)

frontend for data exploration



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

deep learning with less hardware

- GPU-centric I/O path

[Path to GPU-Initiated I/O for Data-Intensive Systems](#)

Karl B. Torp, Simon Lund, Pınar Tözün.

DaMoN 2025

- data & work sharing

[TensorSocket: Shared Data Loading for Deep Learning Training](#)

Ties Robroek, Neil Kim Nielsen, Pınar Tözün.

SIGMOD 2026

- impact of data selection

Collaboration started at 2024 Dagstuhl seminar: [Resource-Efficient Machine Learning](#).

Ties Robroek, Maximilian Böther, Niklas Christiansen, Daniel Sepehri, Dagmar Kainmüller, Theo Rekatsinas, Stefanie Scherzinger, Ana Klimovic, Pınar Tözün.



thank you!