HPCA 2024

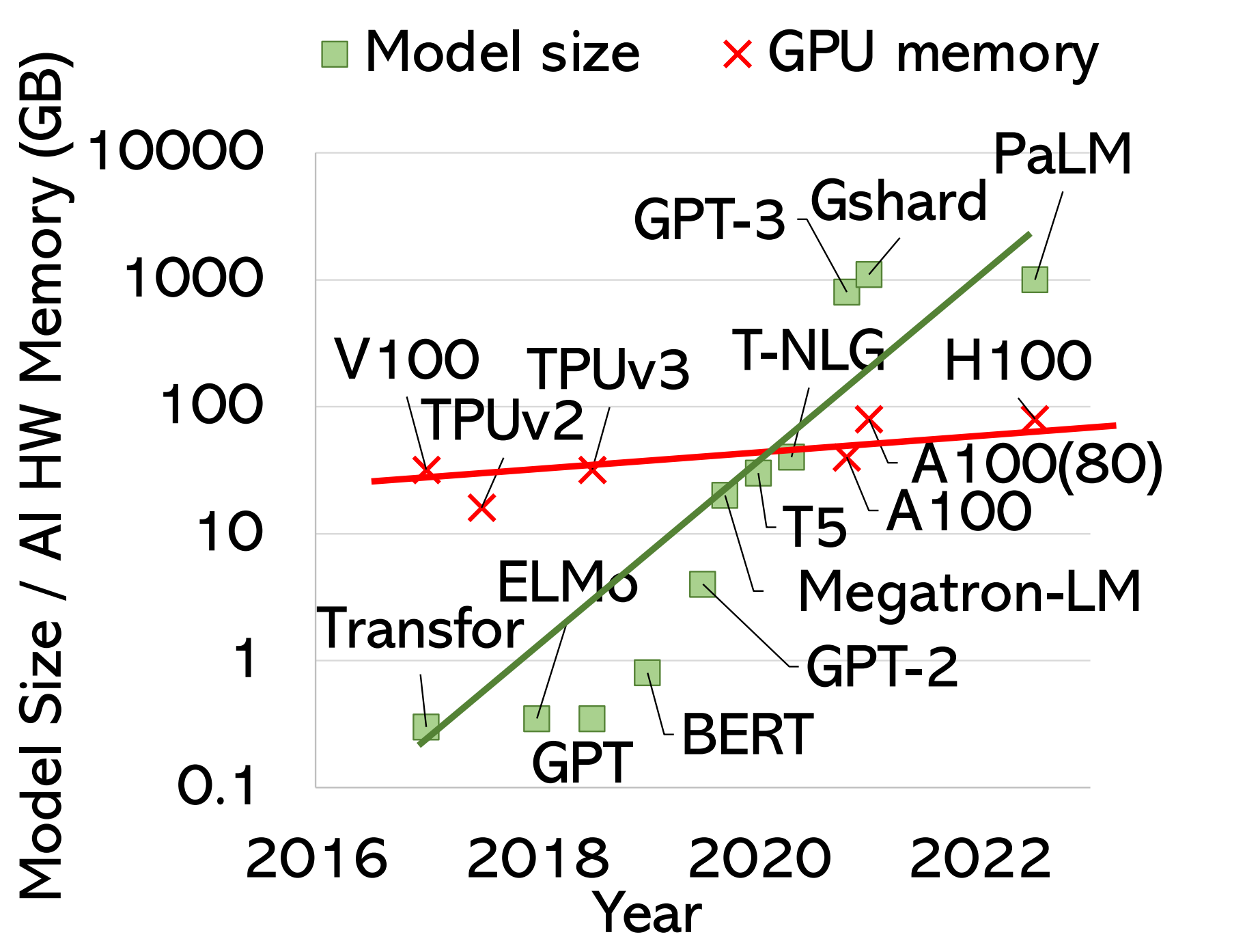
Smart-Infinity: Fast Large Language Model Training using Near-Storage Processing on a Real System

Hongsun Jang, Jaeyong Song, Jaewon Jung, Jaeyoung Park,
Youngsok Kim, and Jinho Lee

Accelerated Intelligent Systems Lab.

Introduction

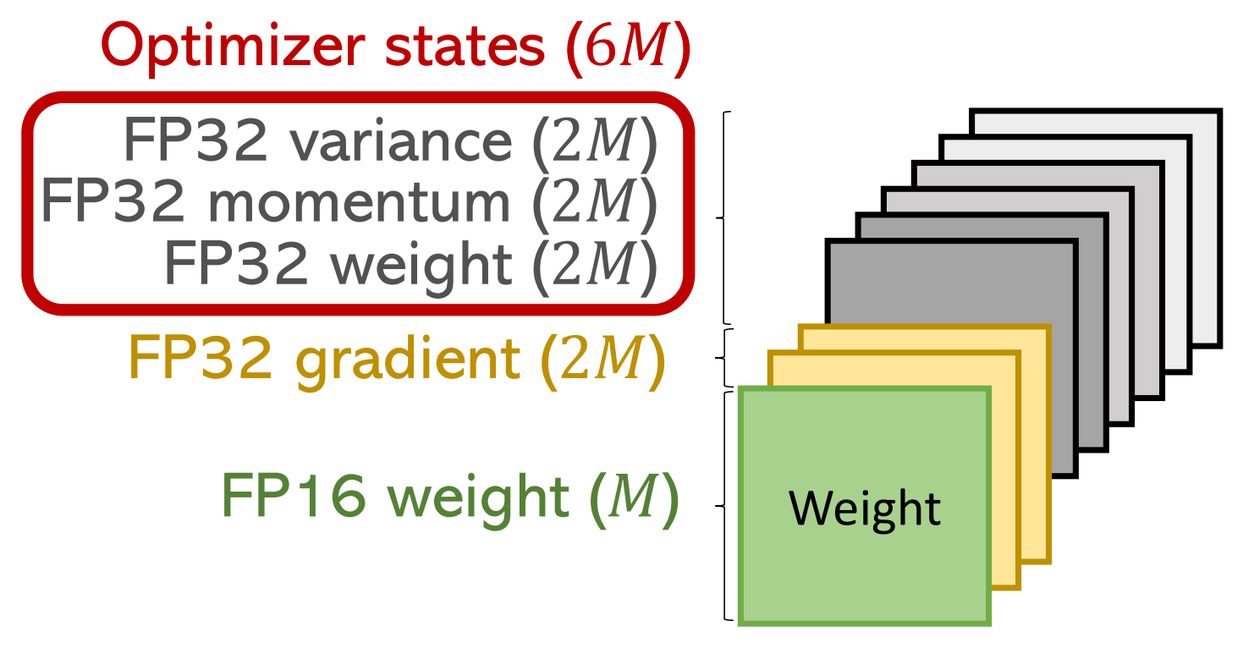
- **Large Language Models (LLMs)**



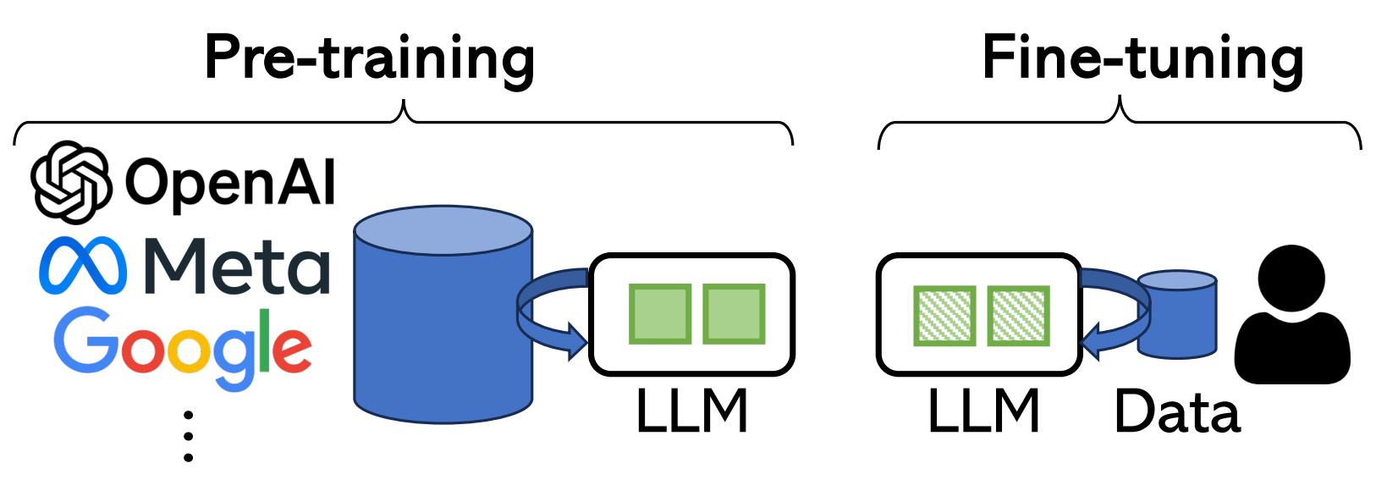
Model size: 5000×	GPU memory: 2.5×

(1) Memory Capacity Wall

* P Micikevicius et al., "Mixed Precision Training", ICLR'18
* DP Kingma et al., "Adam: A method for stochastic optimization", arXiv'14



(2) Large "*Optimizer states*"



(3) Fine-tuning

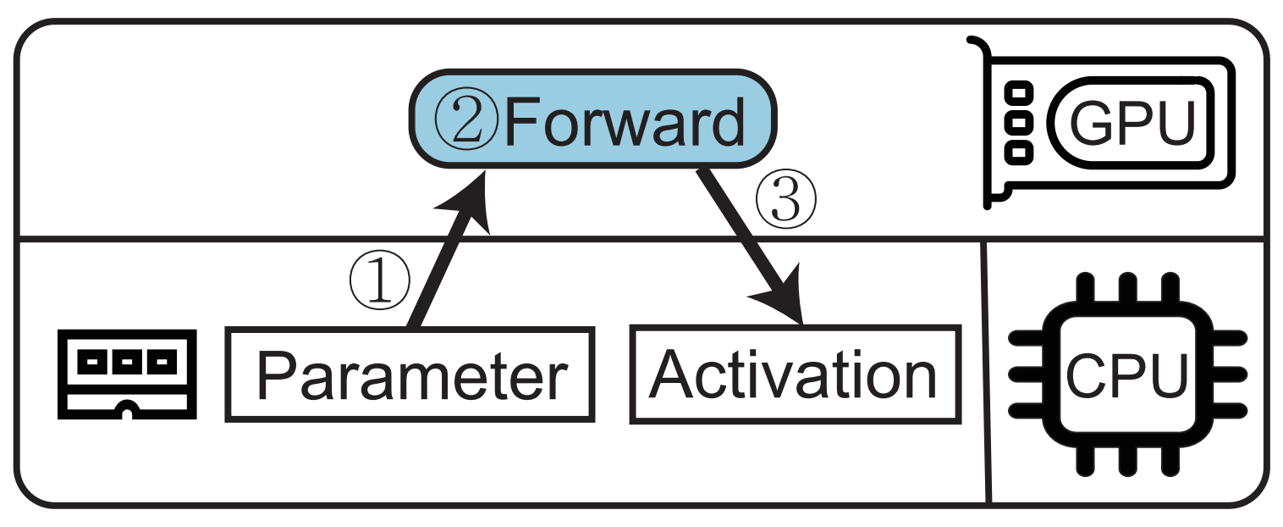
Background

* Microsoft, "ZeRO-Infinity: Breaking the GPU memory wall for extreme scale deep learning", SC'21

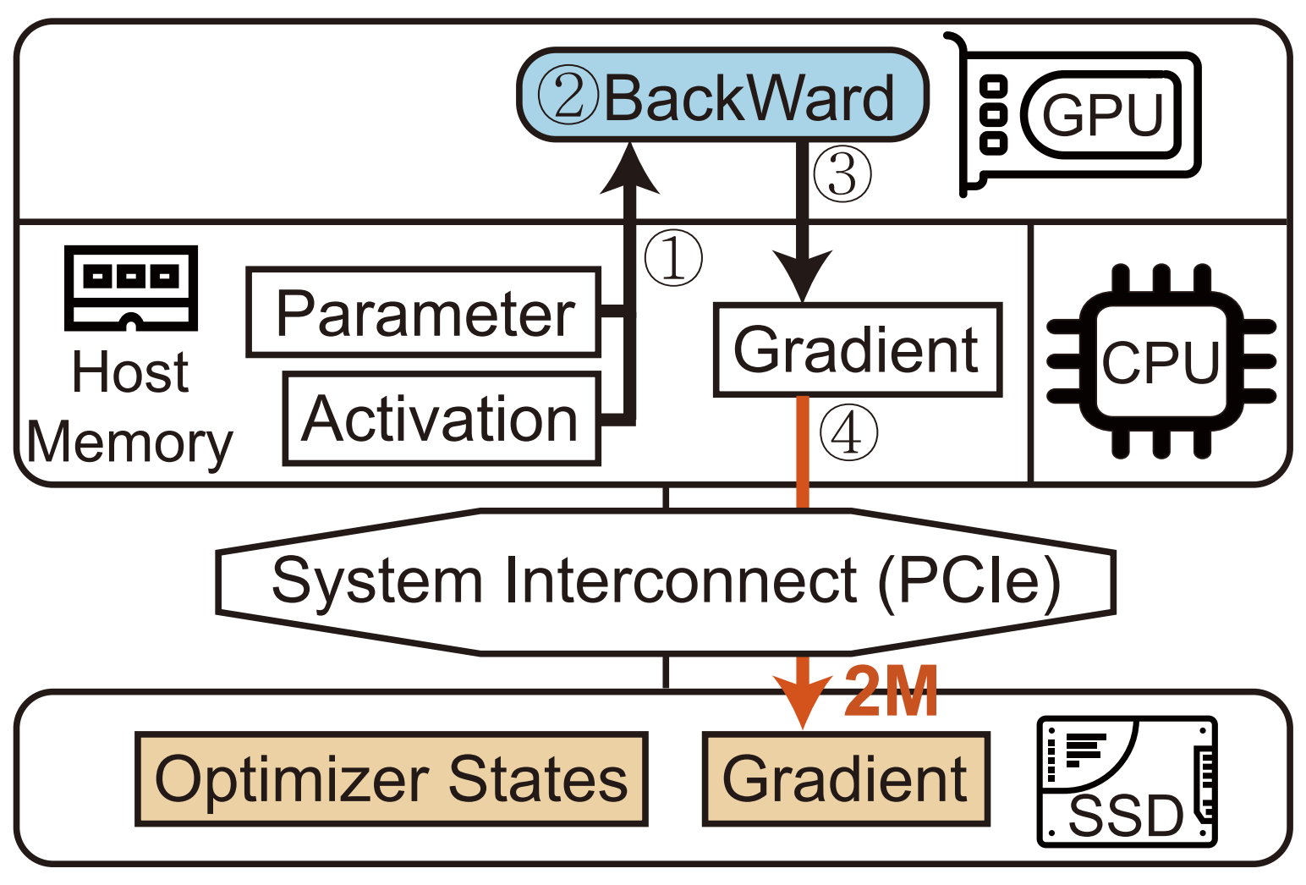
- ***Storage-offloaded training***

[Forward → Backward → Update]

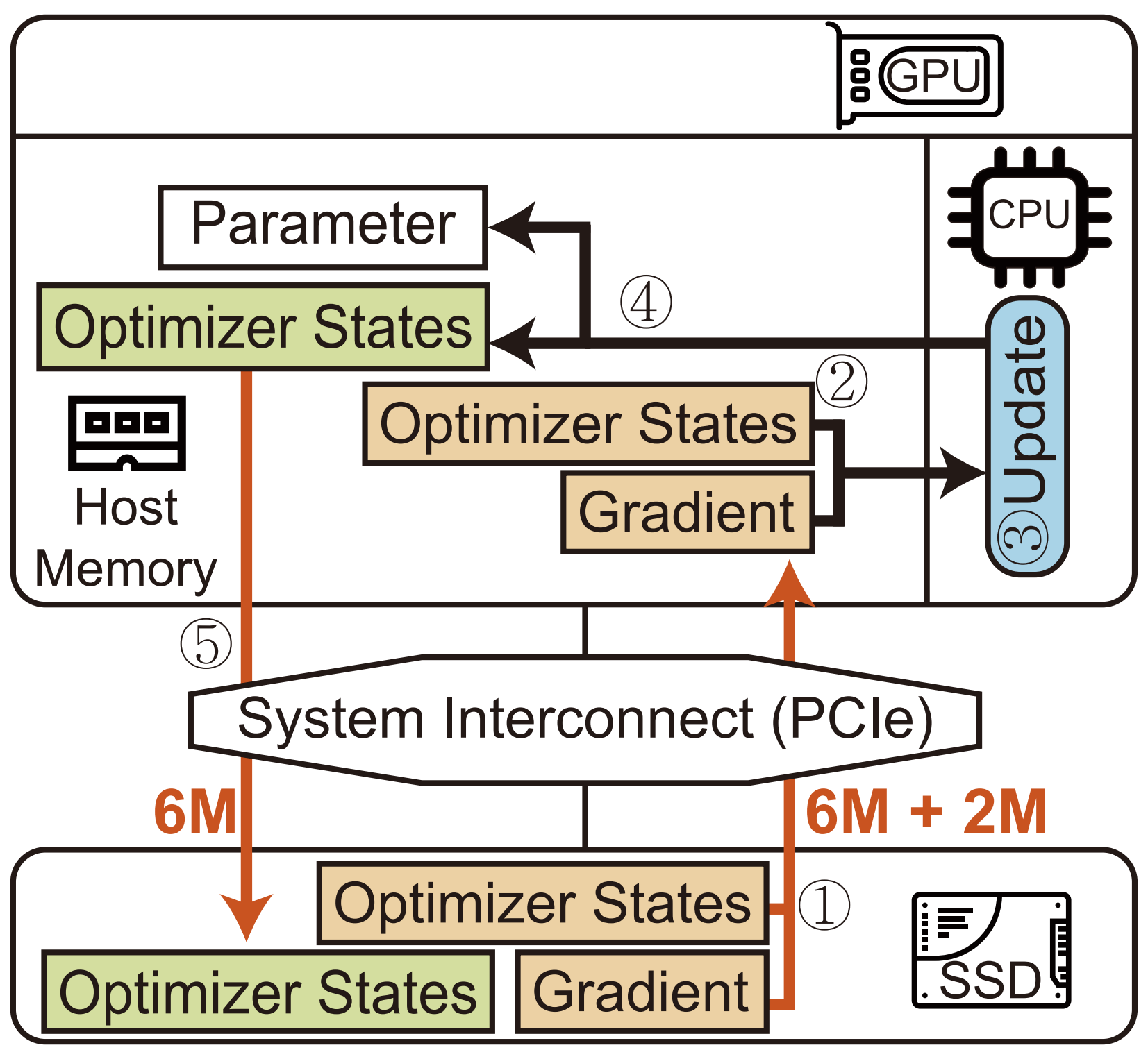
(1) Forward



(2) Backward (Gradient)



(3) Update (Optimizer states + Gradient)



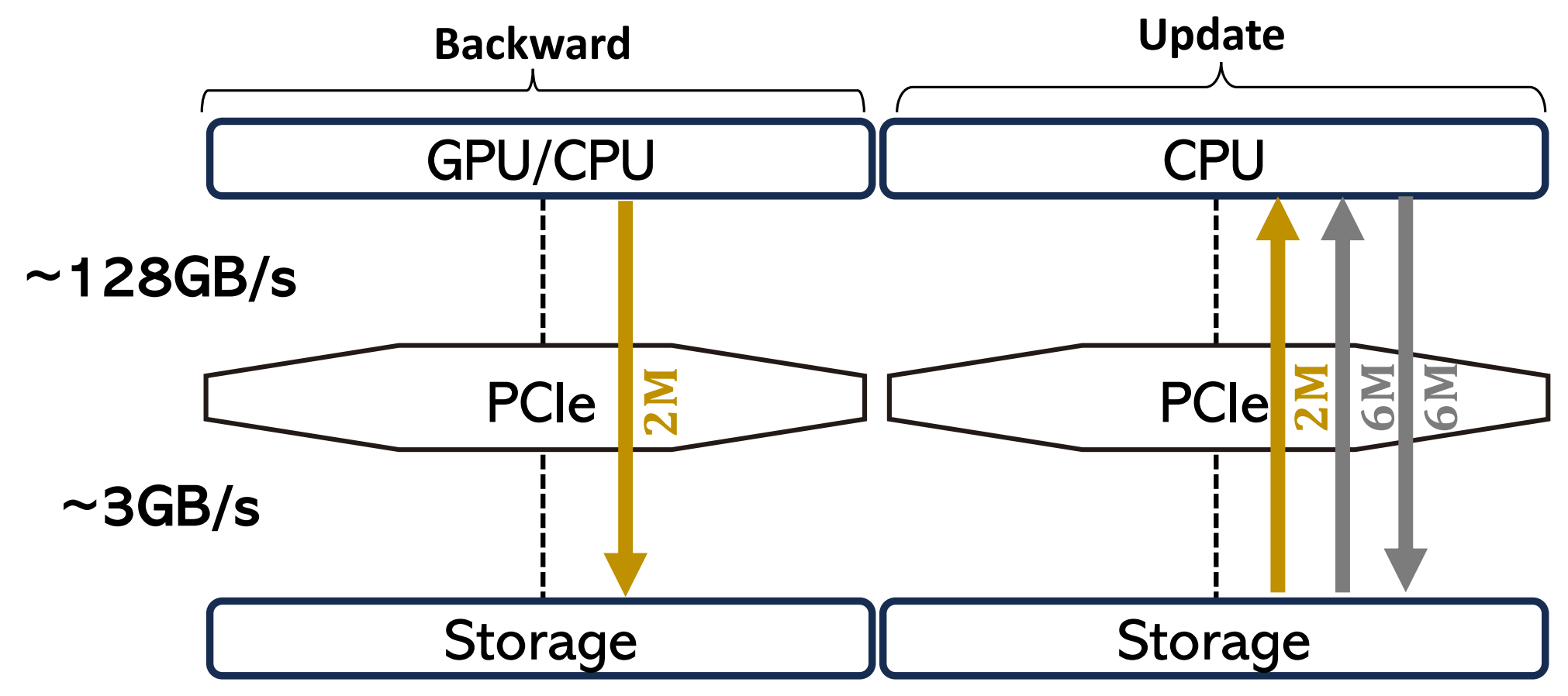
- Total storage data traffic

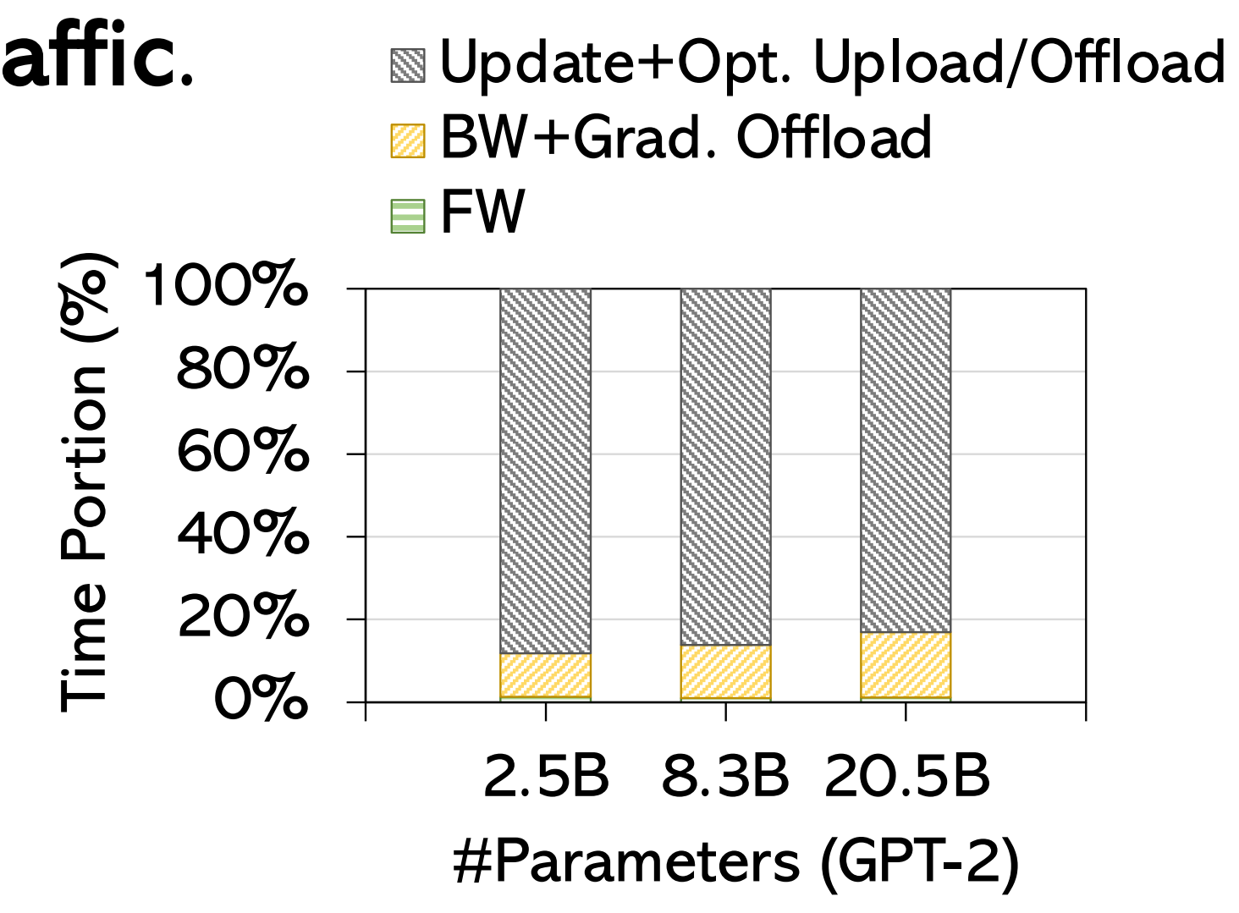
Read + Write (Gradient) = 2M + 2M

Read + Write (Optimizer states) = 6M + 6M

Motivation

(1) The main bottleneck is **huge storage traffic.**



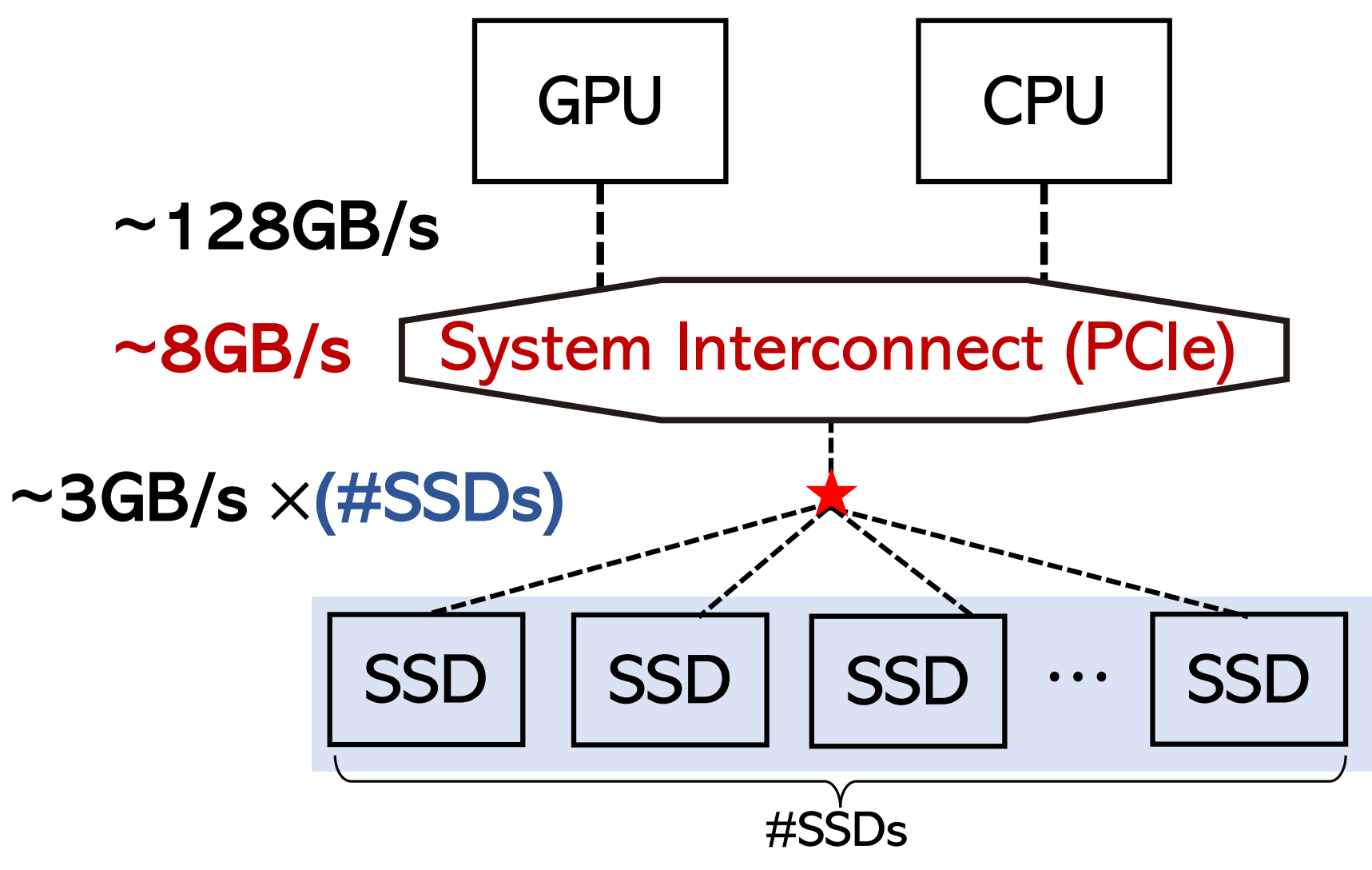


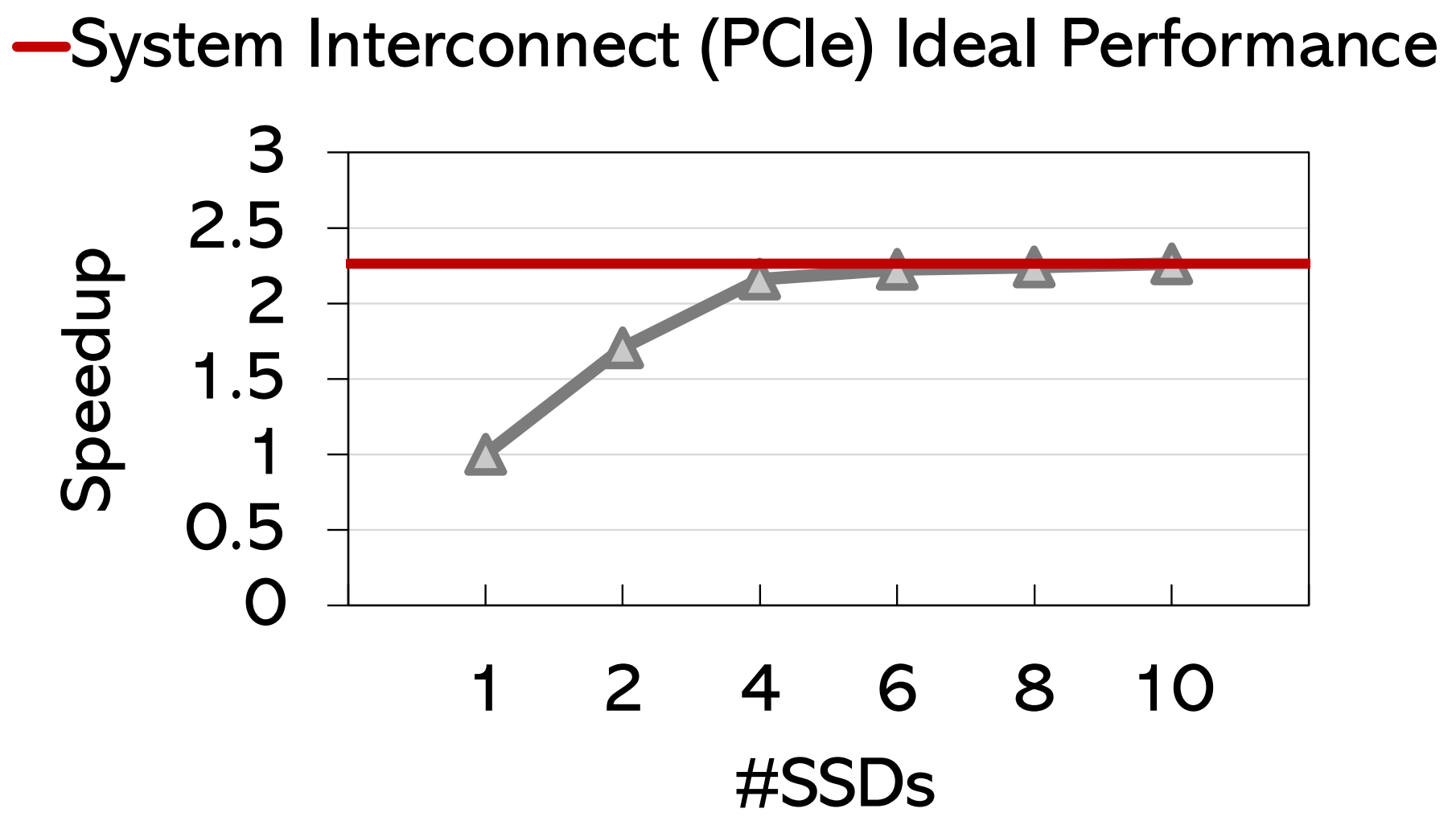
Breakdown of baseline

Data traffic through **a few GB/s** takes up **> 88%** of the training time.

→ Sol. #1: **Utilizing more SSDs** to accelerate storage-offloaded training

(2) Increasing #SSD is limited by the system interconnect (PCIe).





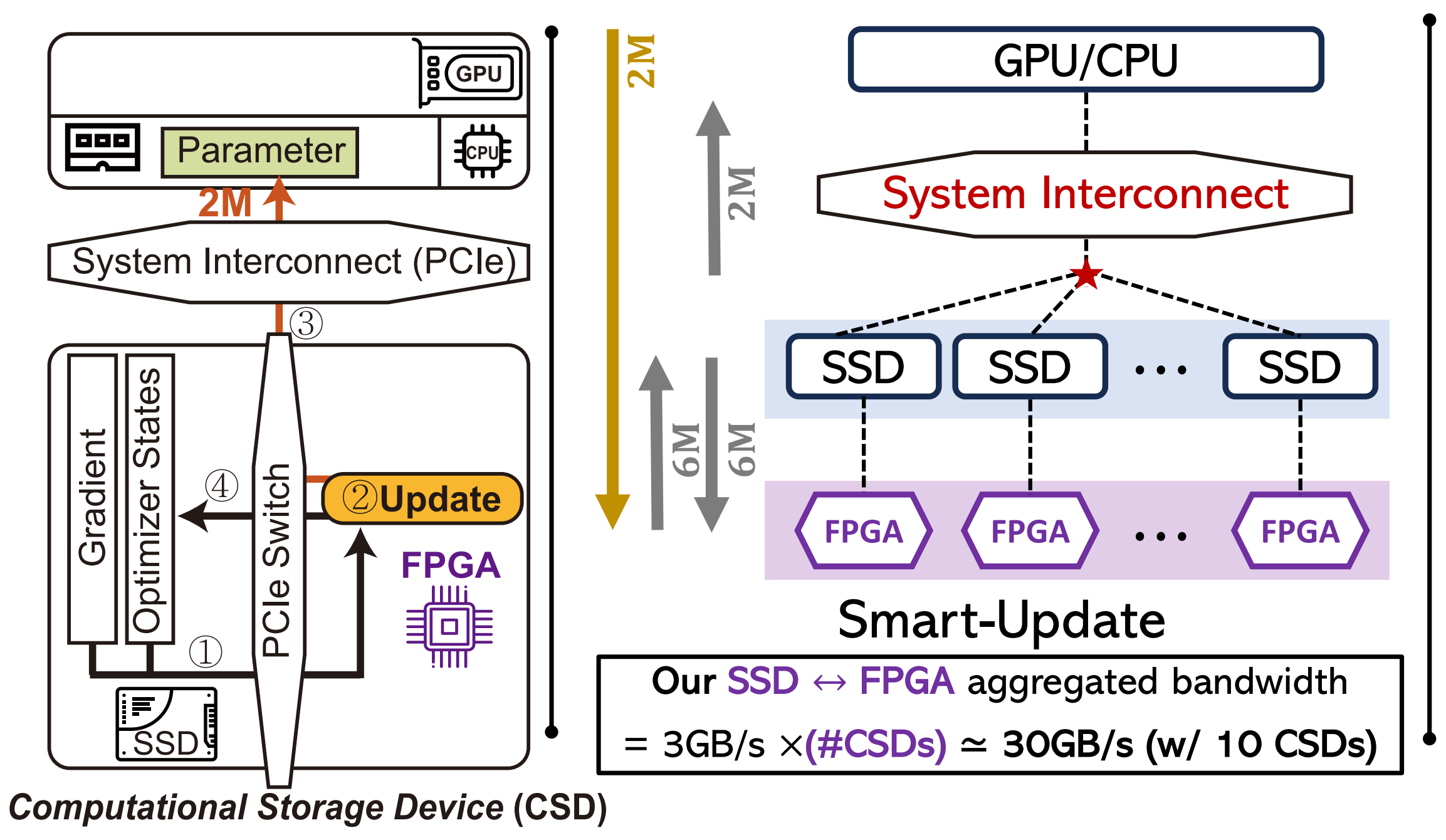
Expected aggregated bandwidth = ~3GB/s ×(#SSDs)	Measured maximum aggregated bandwidth = ~8GB/s ≃ 4×SSDs

→ Sol. #2: **Minimizing the traffic** through **shared system interconnect**

Smart-Infinity

1. Smart-Update (SU)

Utilizing aggregated internal bandwidth of CSDs for update



Our SSD ↔ FPGA aggregated bandwidth = 3GB/s ×(#CSDs) ≃ 30GB/s (w/ 10 CSDs)

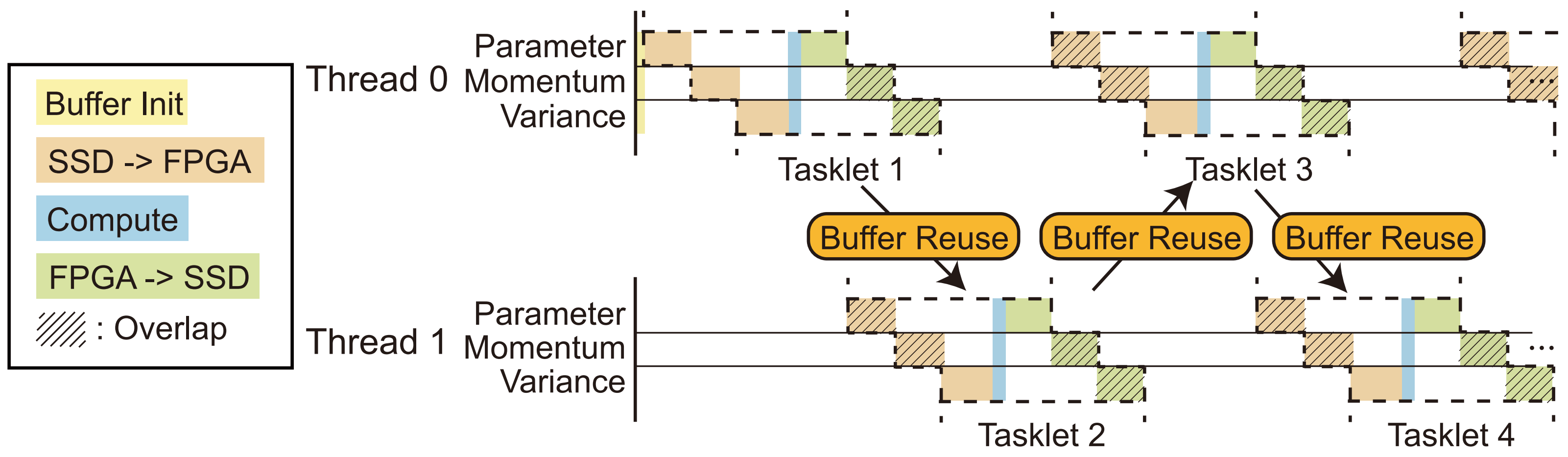
Baseline traffic via system interconnect
Read + Write (Gradient) = 2M + 2M
Read + Write (Optimizer states) = 6M + 6M

4× reduced

Our traffic via system interconnect
Read (Parameter) = 2M
Write (Gradient) = 2M

2. Internal Data Transfer Handler (+O)

Efficient and optimized structure for CSD application

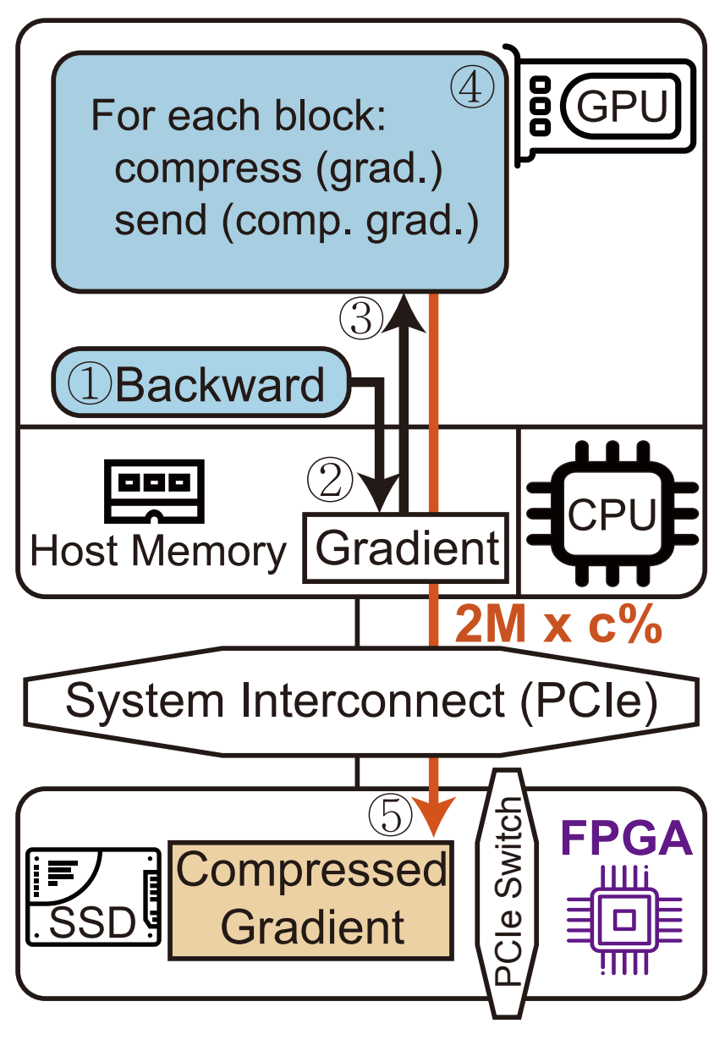


3. Smart-Comp (+C)

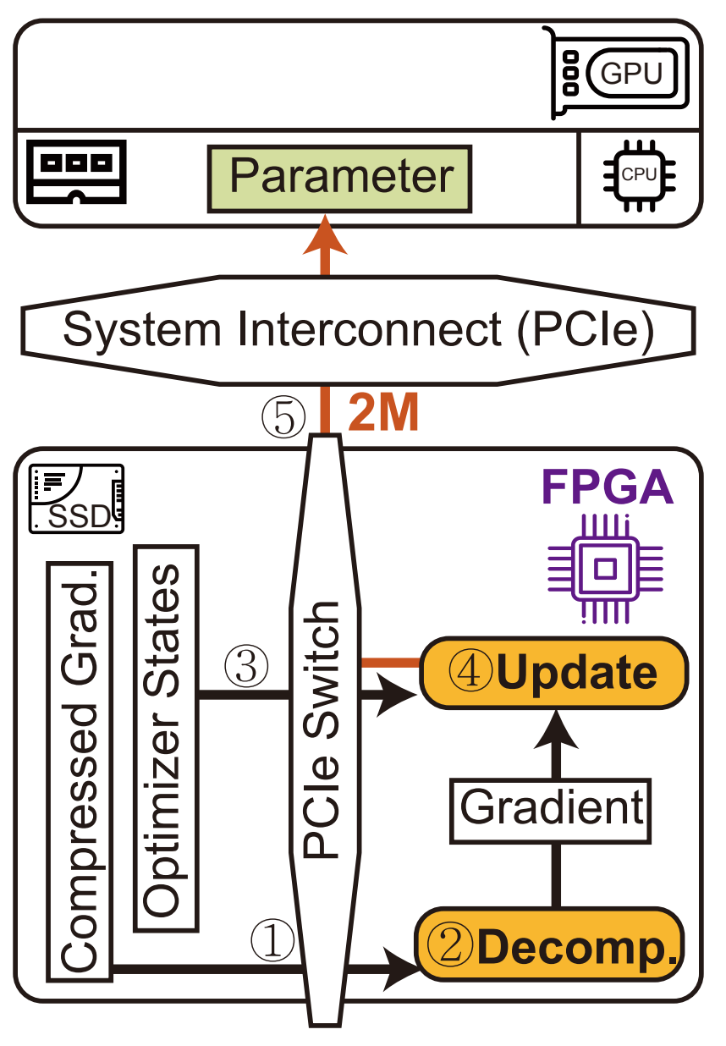
CSD-aided gradient compression (e.g. Top-K)

≃ 8× reduced

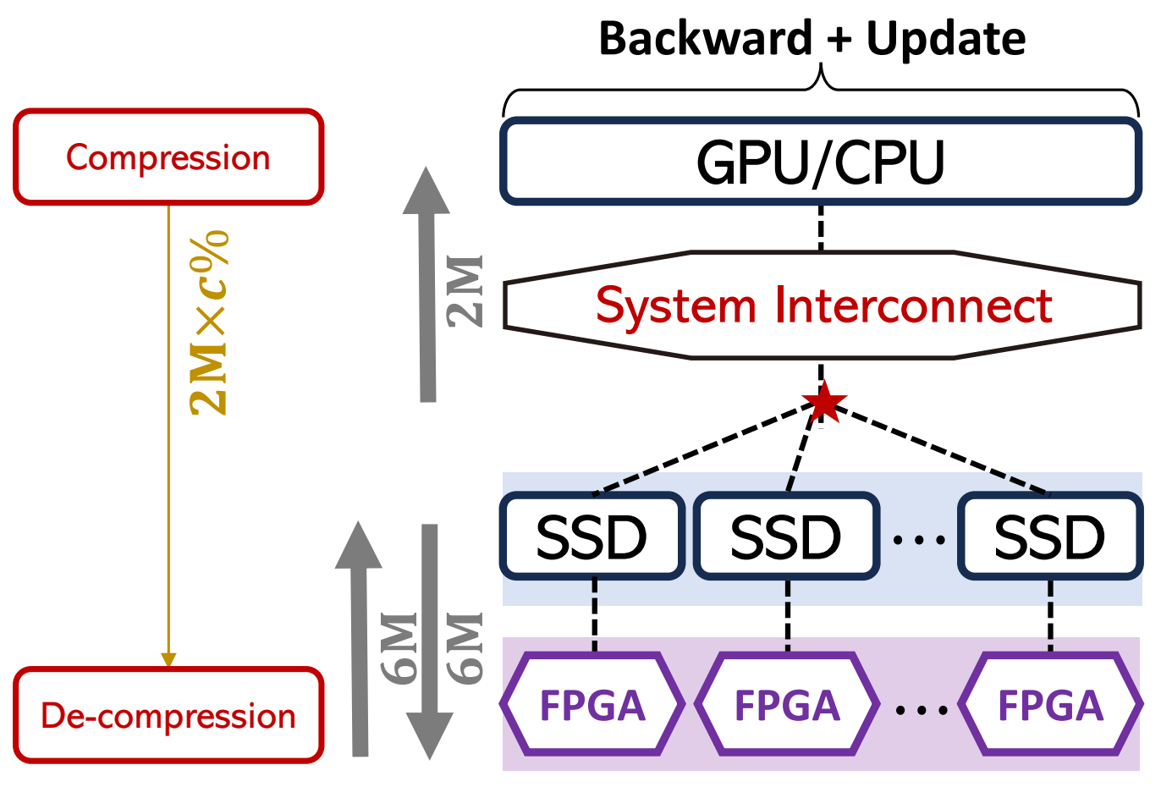
Our data traffic via system interconnect
(Parameter) + ~~(Gradient)~~ ≃ 2M



Backward w/ Smart-Comp



Update w/ Smart-Comp

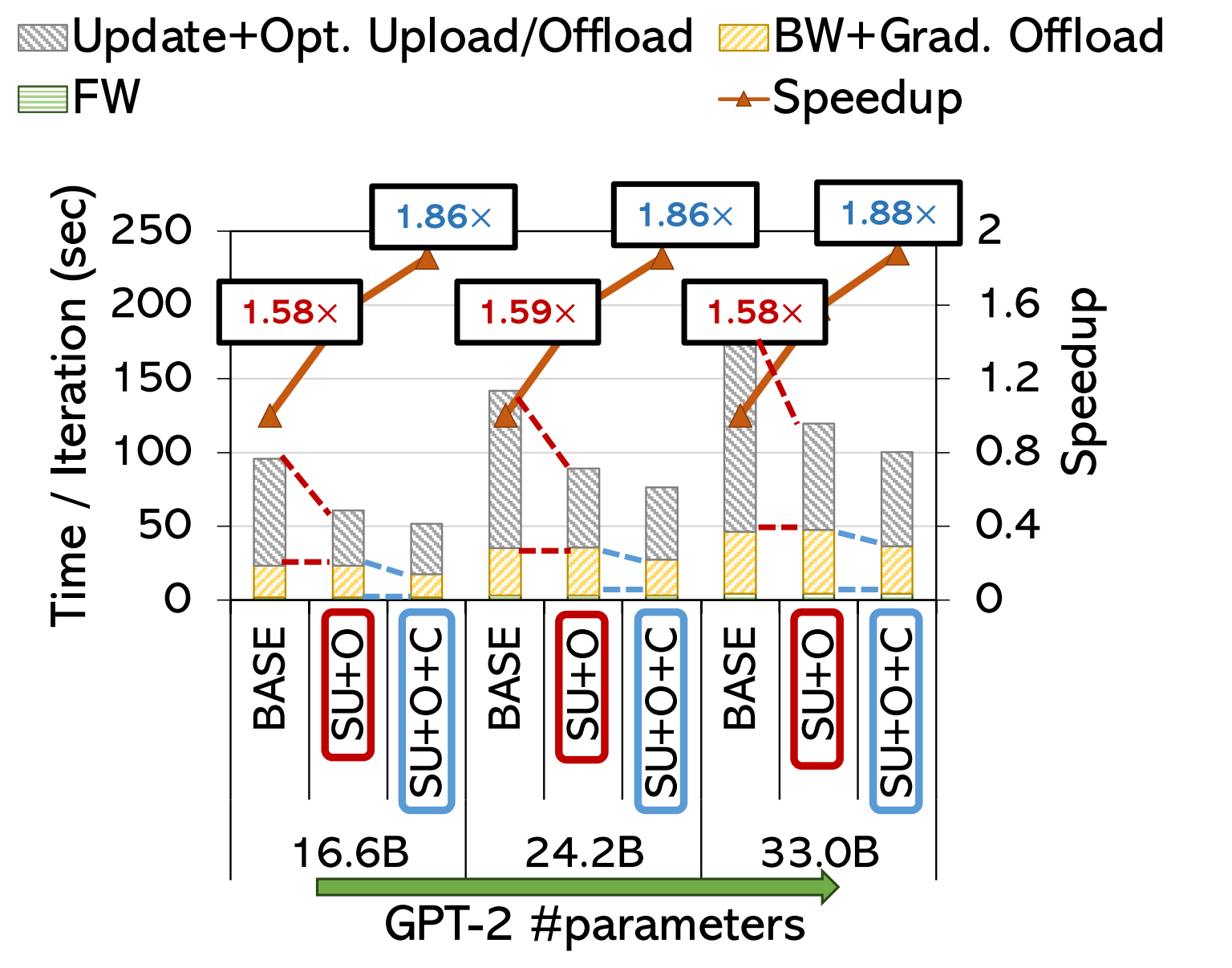


Smart-Update w/ Smart-Comp

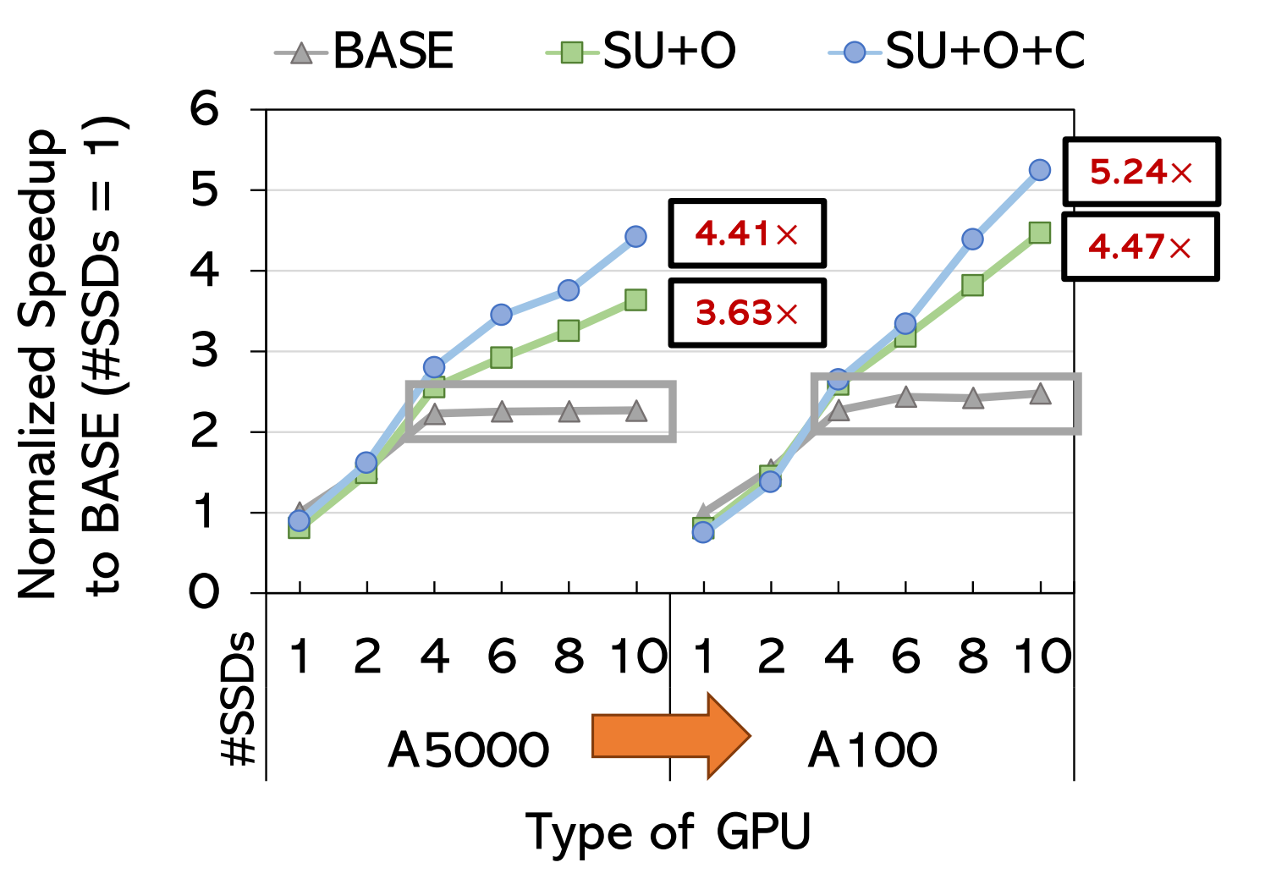
Evaluation

- **PCIe Expansion**: H3 Falcon 4109
- CSD: **SAMSUNG SmartSSD×10**
- Magnitude-based Top-K, 2% (default)

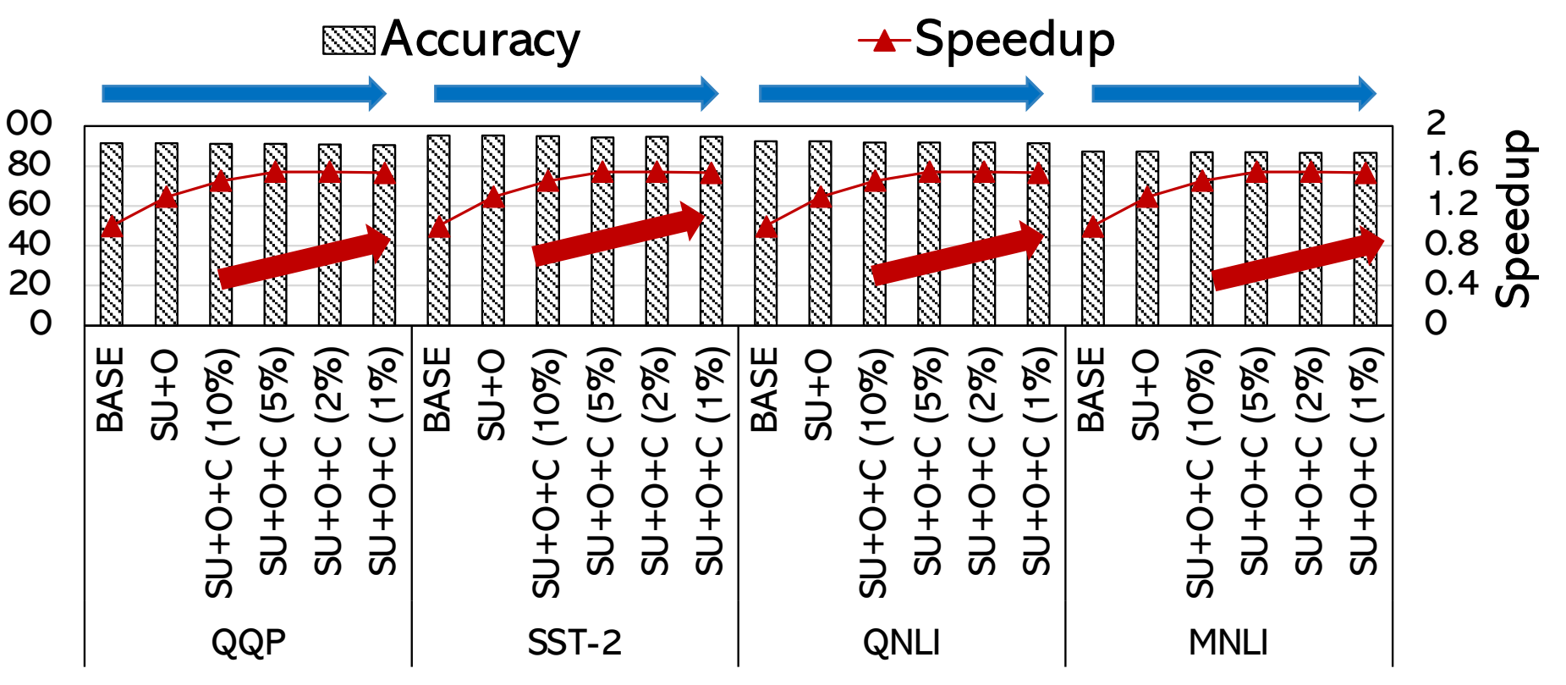
Notations	
BASE	Baseline with RAID0
SU+O	Smart-Update w/ Optimizations
SU+O+C	SU+O+Smart-Comp



- Significant 1.5~1.9x speedup w/ 10 CSDs
- Scalable on larger models (➡)



- The baseline has limited scalability.
- Smart-Infinity is scalable.
- Scalable on more compute power (➡)



- NLP fine-tuning task (GLUE benchmark)
- Smart-Update does not change the algorithm.
- Smart-Comp shows comparable accuracies (➡) with steady speedups (➡).

Conclusion

- Smart-Infinity greatly reduces traffic through shared system interconnect (PCIe lanes).
  1) Utilizing aggregated internal bandwidth of CSDs
  2) CSD-aided traffic compression
- Smart-Infinity is a ready-to-use and open-source framework for storage-offloaded training. (https://github.com/AIS-SNU/smart-infinity)