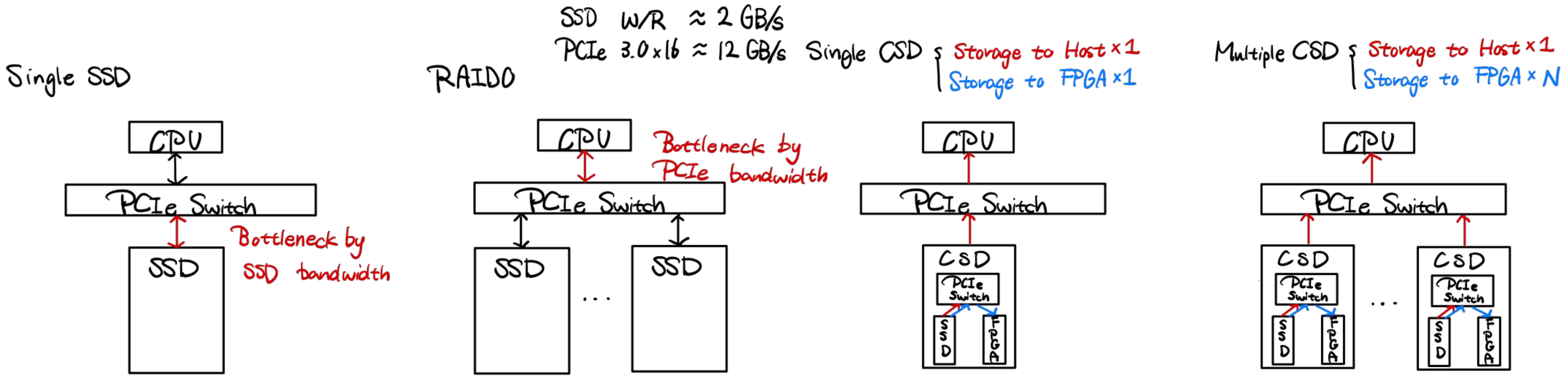
Final Project

蕭泓佐 林胤辰 方淑玲 王俊閔

Computational Storage Device (CSD) - SmartSSD

Scale internal bandwidth with # of CSDs

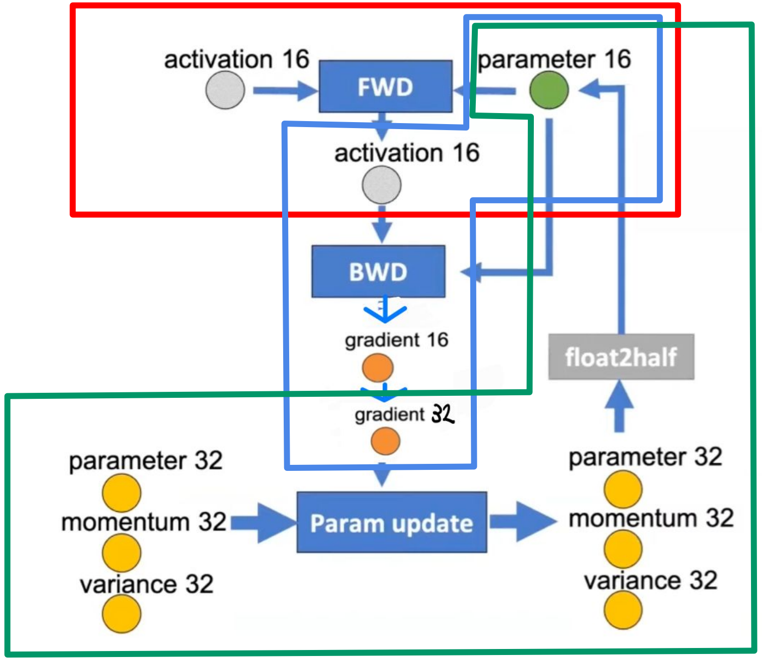


Smart-Infinity

Problem Statement

- Workload: Heterogeneous (GPU+CPU+SSD) mixed precision DL training
- Existing method (storage-offload training) comes at the cost of storage bandwidth bottleneck

Existing Method (ZeRO-Infinity)



Mixed precision training iteration for a layer.

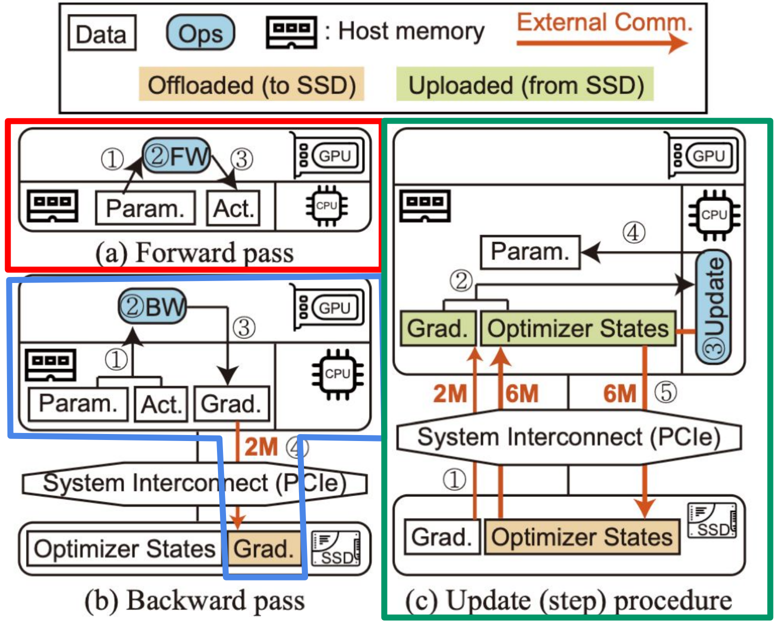
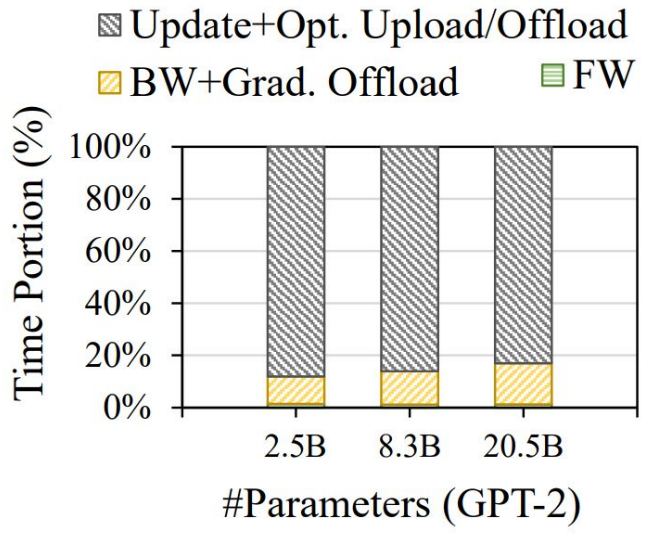
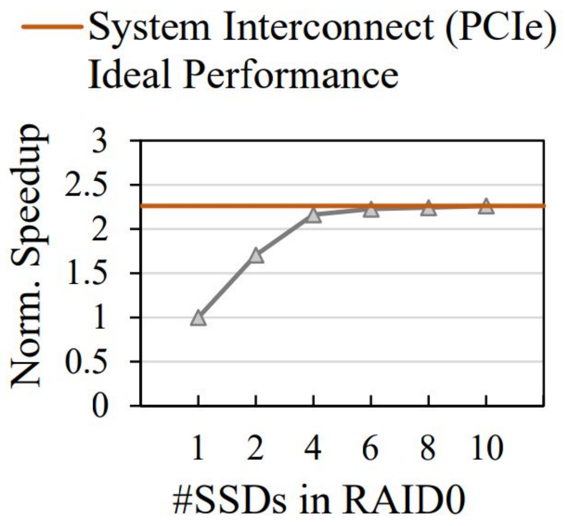


Fig. 1: A conceptual diagram of the storage-offloaded LLM training. Overview of (a) the forward pass, (b) the backward pass, and (c) the update (step) procedure.

Bottleneck Analysis



(a) Breakdown of baseline

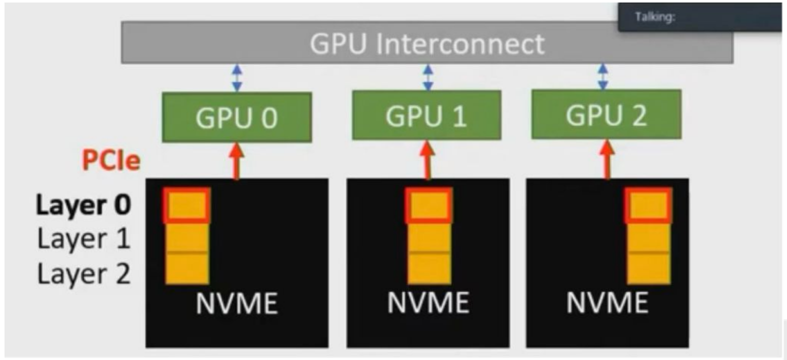


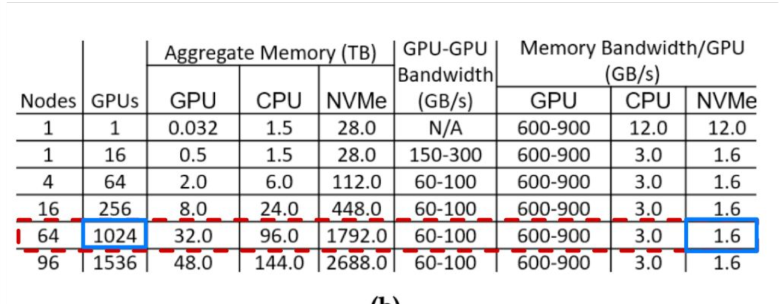
(b) RAID0 performance

Existing Method (ZeRO-Infinity)

Scaling node (i.e. PCIe lane) is proposed as the solution in ZeRO-Infinity

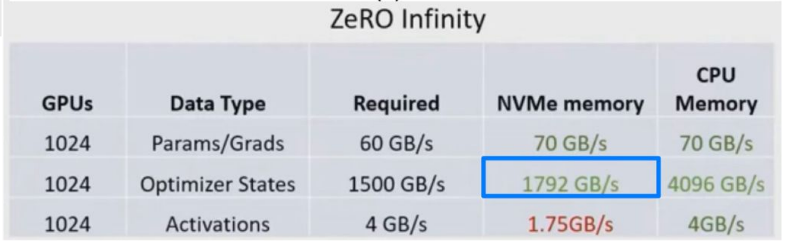
→ Smart-Infinity: scale CSDs instead of compute nodes





Nodes	GPUs	Aggregate Memory (TB)			GPU-GPU Bandwidth (GB/s)	Memory Bandwidth/GPU (GB/s)		
		GPU	CPU	NVMe		GPU	CPU	NVMe
1	1	0.032	1.5	28.0	N/A	600-900	12.0	12.0
1	16	0.5	1.5	28.0	150-300	600-900	3.0	1.6
4	64	2.0	6.0	112.0	60-100	600-900	3.0	1.6
16	256	8.0	24.0	448.0	60-100	600-900	3.0	1.6
64	1024	32.0	96.0	1792.0	60-100	600-900	3.0	1.6
96	1536	48.0	144.0	2688.0	60-100	600-900	3.0	1.6

(b)



ZeRO Infinity

GPUs	Data Type	Required	NVMe memory	CPU Memory
1024	Params/Grads	60 GB/s	70 GB/s	70 GB/s
1024	Optimizer States	1500 GB/s	1792 GB/s	4096 GB/s
1024	Activations	4 GB/s	1.75GB/s	4GB/s

Smart Update

1. By offloading parameter update to CSDs, we can scale bandwidth with # of CSDs.
2. Gradient & Param transfer still goes through CPU-PCIe → The CSD scaling upper bound occur when gradient and param transfer become the new bottleneck.

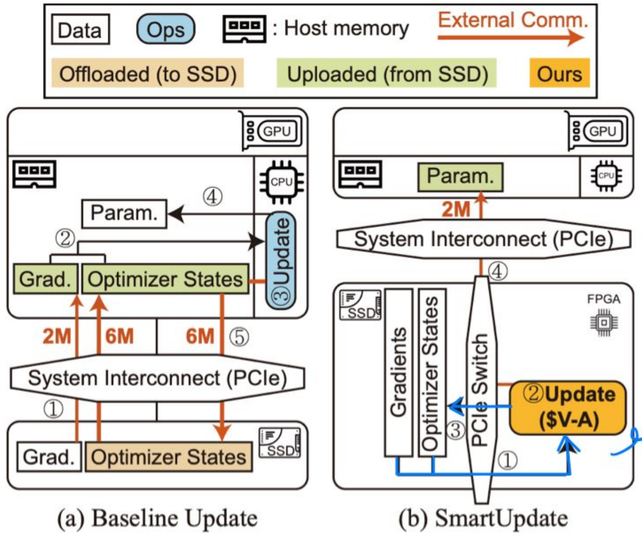


Fig. 4: Update procedure of the storage-offloaded training with (a) baseline [96] and (b) SmartUpdate.

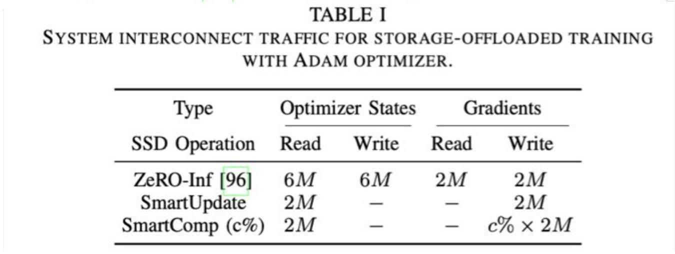


TABLE I
SYSTEM INTERCONNECT TRAFFIC FOR STORAGE-OFFLOADED TRAINING WITH ADAM OPTIMIZER.

Type	Optimizer States		Gradients	
SSD Operation	Read	Write	Read	Write
ZeRO-Inf [96]	$6M$	$6M$	$2M$	$2M$
SmartUpdate	$2M$	–	–	$2M$
SmartComp (c%)	$2M$	–	–	$c\% \times 2M$

Internal bandwidth scales linearly with # of CSDs (Scale CSDs instead of compute nodes)

* Gradient and param transfer still goes through CPU-PCIe
⇒ The CSD scaling upper bound occur when gradient and Param transfer become new bottleneck.

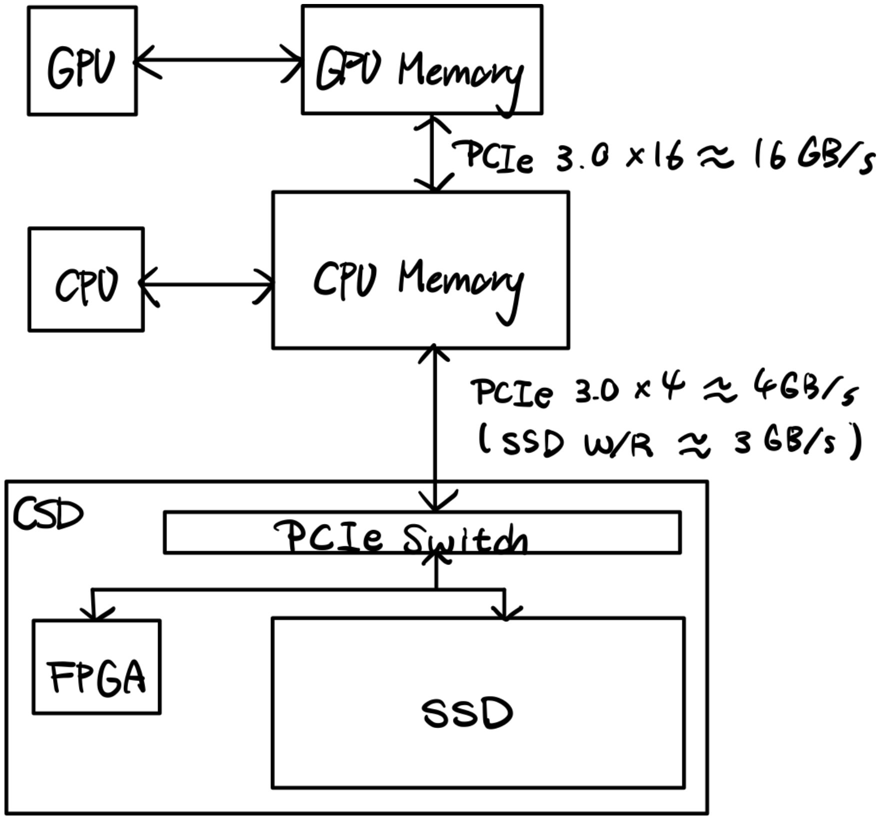
Environment - System Configuration

Hardware

1. NVIDIA RTX 3090-GPU (24GB)
2. 16-core Intel I7 12700 CPU
3. 128GB RAM
4. Samsung SmartSSD

Software

1. Ubuntu 20.04 with Linux kernel 5.4.0-164
2. CUDA 11.6
3. PyTorch 1.13
4. DeepSpeed v0.9.3



Environment - Challenges

System Support

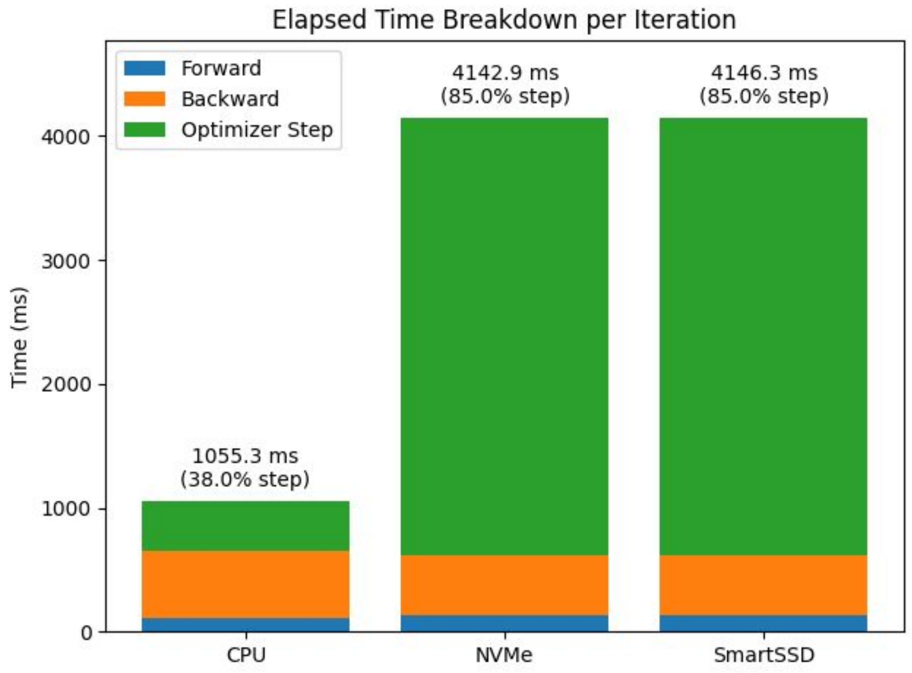
1. The SmartSSD product line has been discontinued, making it difficult to find systems with compatible configurations or support.
   → We find the compatible system configuration by trial and error.

Cooling

1. The SmartSSD is designed for installation in servers with controlled airflow for effective cooling, which makes local installation in a standard PC challenging due to potential overheating.
   →We use a external fan to cooldown the SmartSSD.

Experiment Result

1. As we only have one SmartSSD device we conduct a scalability analysis to verify the result from the paper.
2. From configuration 2, we can observe that most time spent during an iteration is in optimizer step.
3. From configuration 1 and 2, we can derived that most of the time spent during optimizer step is used in IO not computation.



Scalability Analysis

There are no data-dependencies between each parameter in optimizer step

→ Optimizer step scales linearly with # of CSDs.

$$For\ each\ Parameter\ w^j$$

$$(j\ subscript\ dropped\ for\ clarity)$$

$$\nu_t = \beta_1 * \nu_{t-1} - (1 - \beta_1) * g_t$$
$$s_t = \beta_2 * s_{t-1} - (1 - \beta_2) * g_t^2$$

$$\Delta\omega_t = -\eta \frac{\nu_t}{\sqrt{s_t + \epsilon}} * g_t$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t$$

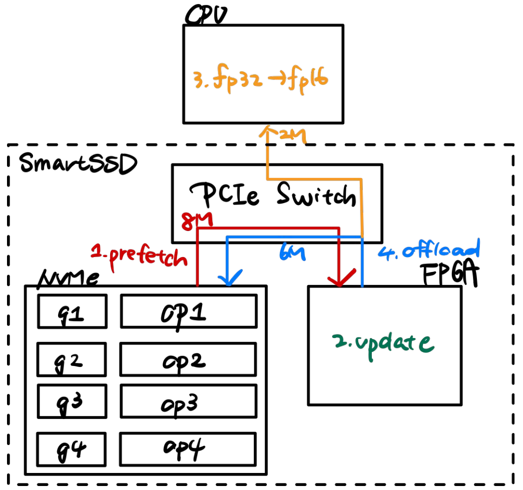
Scalability Analysis

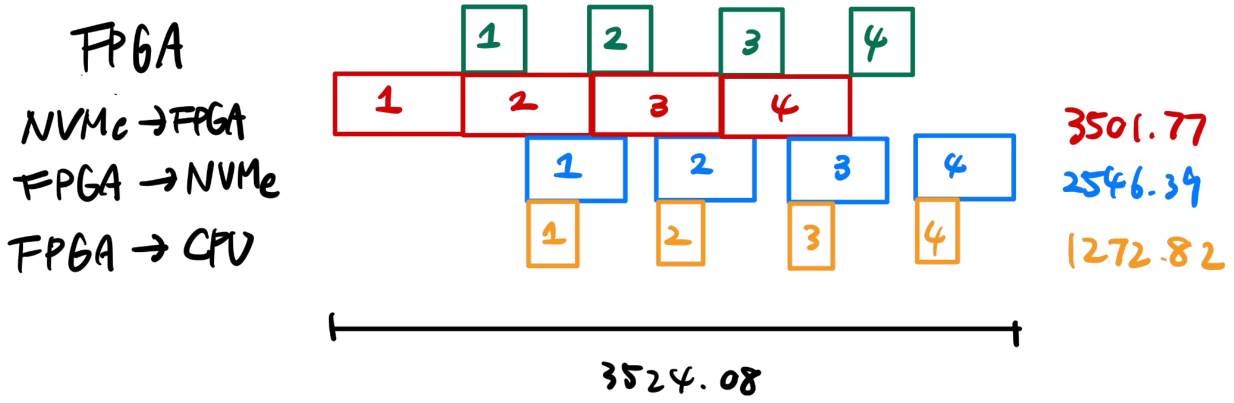
Weakly Parallelizable : Scaling upper bounded by PCIe bandwidth

1. FPGA → CPU Communication

Parallelizable : Scale linearly with # of CSDs

1. FPGA computation
2. NVME→FPGA communication
3. FPGA→NVMe communication





Scalability Analysis

$$T_{opt,n} = max\left(\frac{T_{opt}}{n}, \frac{T_{param\_swap}}{min\left(n, \frac{B_{PCIe}}{B_{SSD}}\right)}\right)$$

$$T_{total,n} = T_{opt,n} + T_{fwd} + T_{bwd}$$

n: number of CSDs

T_{opt}: time of optimizer step using 1 CSD

T_{fwd}: time of forward pass

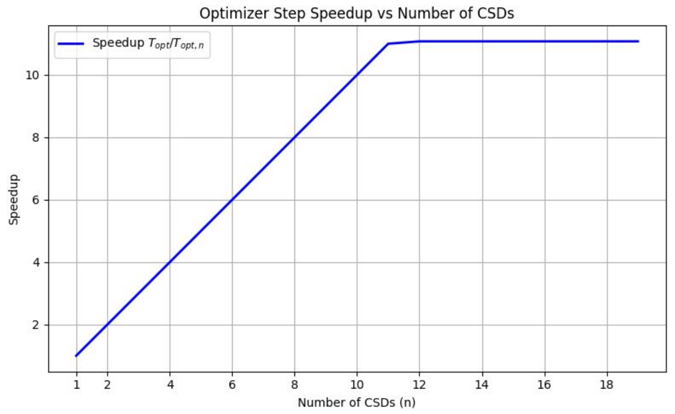
T_{bwd}: time of backward pass

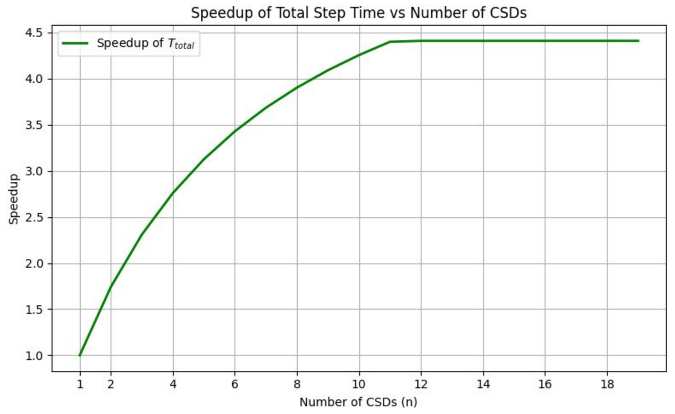
$T_{opt,n}$: time of optimizer step using n CSDs

T_{param_swap}: time of swapping in parameters to CPU

B_{PCIe}: PCIe bandwidth

B_{SSD}: SSD bandwidth


Optimizer Step Speedup vs Number of CSDs
Speedup $T_{opt}/T_{opt,n}$
Speedup
Number of CSDs (n)


Speedup of Total Step Time vs Number of CSDs
Speedup of T_{total}
Speedup
Number of CSDs (n)