WeiPipe: Weight Pipeline Parallelism for Communication-Effective Long-Context Large Model Training

Junfeng Lin Tsinghua University, Ziming Liu National University of Singapore, Yang You National University of Singapore, Jun Wang CETHIK Group Co. Ltd., Weihao Zhang Lynxi Technologies Co. Ltd., Rong Zhao Tsinghua University

Presenter
Dipak Acharya
University of North Texas
Denton TX

Long Context Training

Transformer models are Growing based on 2 major dimensions

1. **Model size**
   - The sizes of the model parameters. (eg. LLama 3.1 with 405B parameters)
   - Distributed training with several GPUs necessary to accommodate the parameters

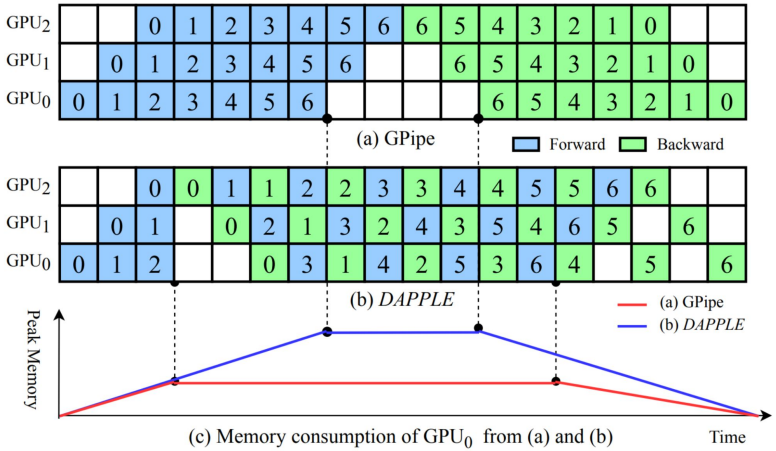
2. **Context Length**
   - With large context, the model needs to compute attention across longer sequence of tokens
   - This results in larger activation size during training
   - Even more relevant in scientific computation tasks such as Alpha Fold

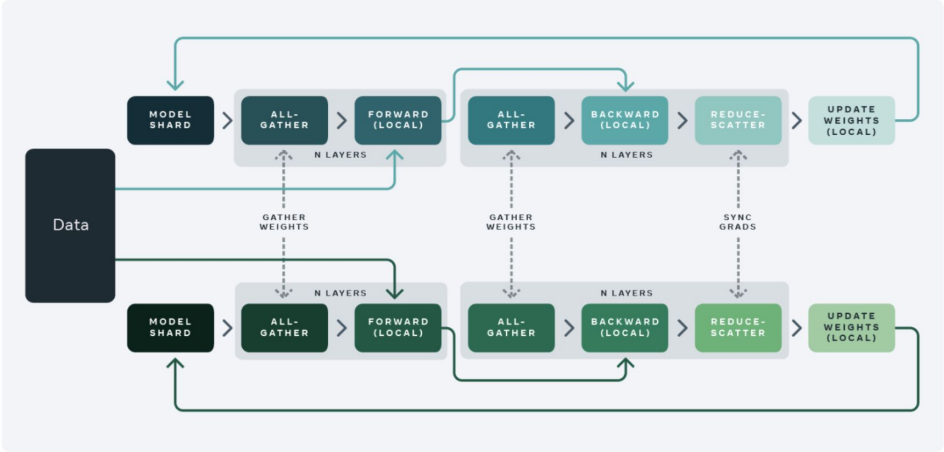
Distributed training necessitates several parallelism techniques:

1. **Data Parallelism (DP):** Fully Sharded Data Parallelism (FSDP), Zero Redundancy Optimizer (ZeRO)
2. **Tensor Parallelism (TP):** Megatron-LM
3. **Pipeline Parallelism (PP):** GPipe, Dapple, Chimera, Hanayo

DP is favoured for its ease of use and PP for low communication overhead as it doesn't require collective communication

Parallel DL Training





Pipeline Parallelism

- Divide the model into sequential segments and assign to the devices
- Feed the data through the pipeline in microbatches

Recent efforts such as *Chimera, Hanayo* and *Zero-bubble pipeline* use more intricate scheduling to optimize pipeline execution

Fully-Sharded-Data-Parallelism

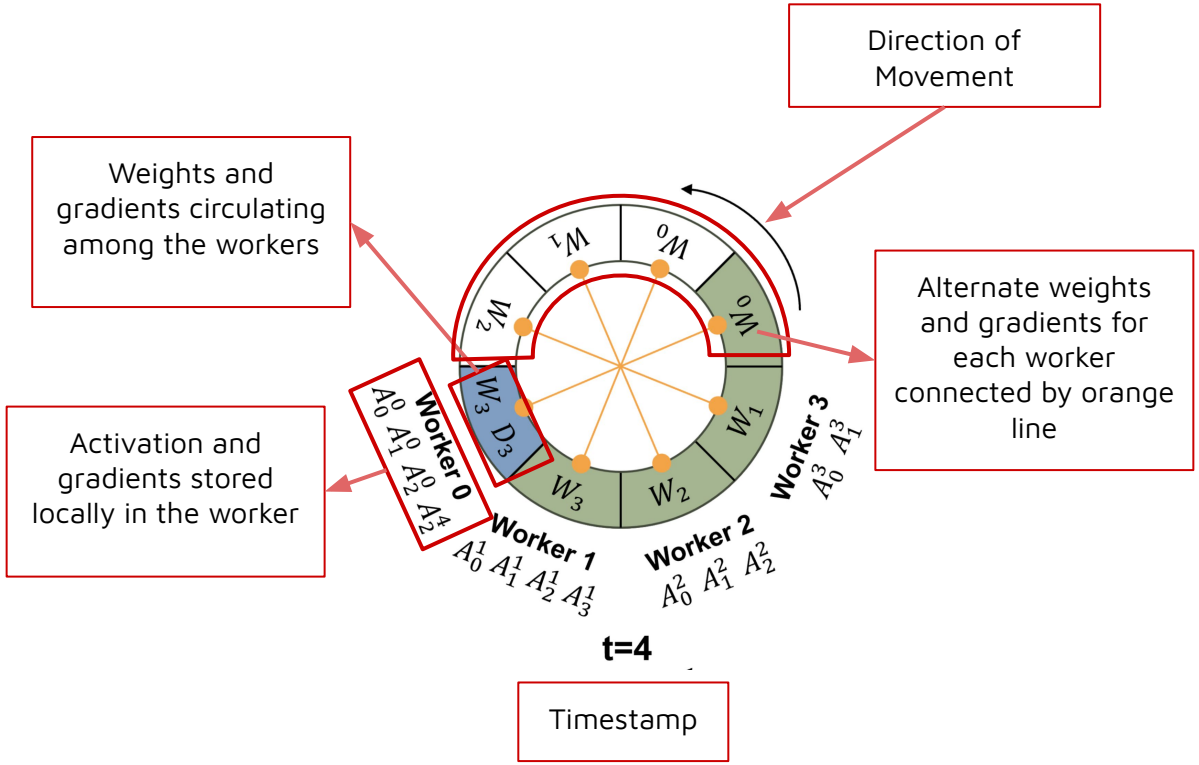
- Shards the data, weights and optimizer states among the GPUs
- Sharding and Unsharding is facilitated by all-gather and reduce-scatter operation

This requires very large amount of collective communication so low-bandwidth systems struggle

Understanding the Figure

N	The number of micro-batches in an iteration
$Iter$	The number of iteration
G	Micro-batch size
P	The number of workers
L	The number of layers in neural network
A_j^i	Activation values of jth layer in ith micro-batch
B_j^i	Gradients of A_j^i
W_j	Weights of jth layer
D_j	Gradients of W_j
$M_{A/B/W/D}$	Memory consumption of A, B, W or D
$T_{F/B/W}$	Time cost for complete forward pass, B pass or W pass.
T_{BW}	$T_B + T_W$
α	The percentage of left memory consumption of activations after B pass.
H	The size of hidden dim in Transformer
S	The sequence length in Transformer

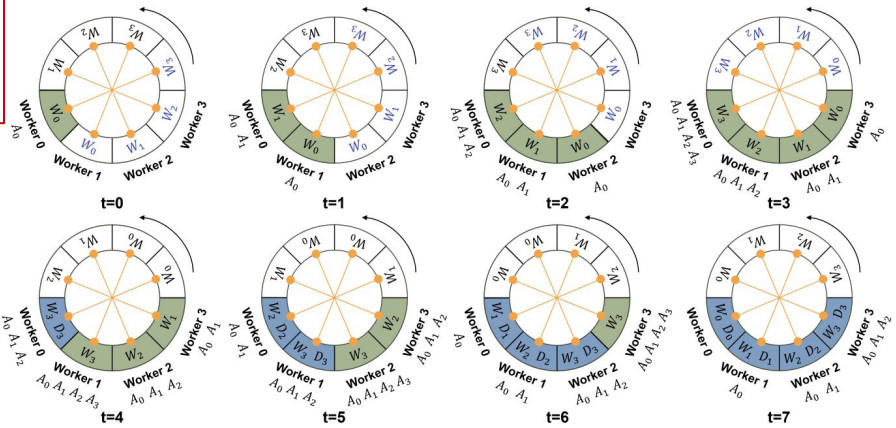
Fig: Notations



WeiPipe Naive

Each worker holds weight of each layers
Worker 0
- initiates forward comp
- Send W_0 to Worker 1
- Receive W_1 from Worker 3

Worker 0 discards W_0, keeps A_0



Worker 0 Completes forward computation on all Layers
It also has activations for all the layers

Each worker holds weight for two layers (Worker 0 has W_3 and W_0)

Worker 0 initiates backward computation starting from last layer. Weight gradients D_x is also transferred similar to the weights.

On Each step it receives weight for next layer for backward computation

D_X is normalized with each worker that computes same layer

Worker 0 completes the backward pass for all the layers

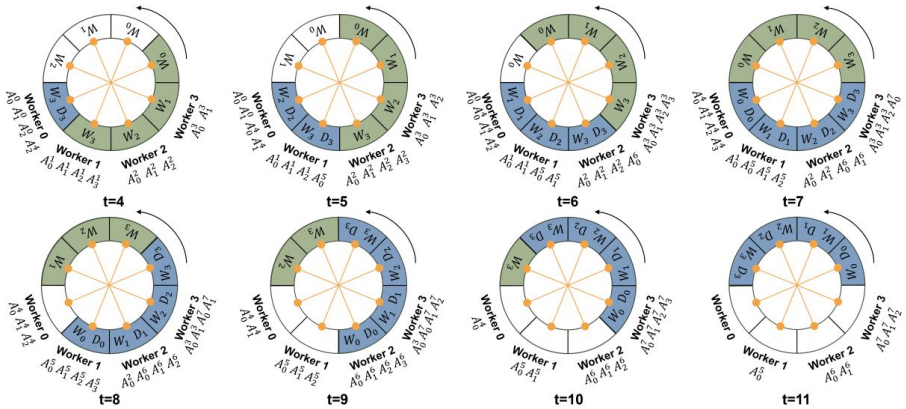
It now starts the same cycle for the another microbatch

After all microbatches processed, each worker applies their D_s

WeiPipe Interleave:Lower Bubble Ratio

As worker 0 performs backward utilizing W_3, it simultaneously performs forward computation with W_0 on a new microbatch.

After all the workers enter the forward-backward interleave stage, the computation is balanced among all workers



Until all workers enter the forward-backward interleave stage, they wait for the execution of the previous stage to complete

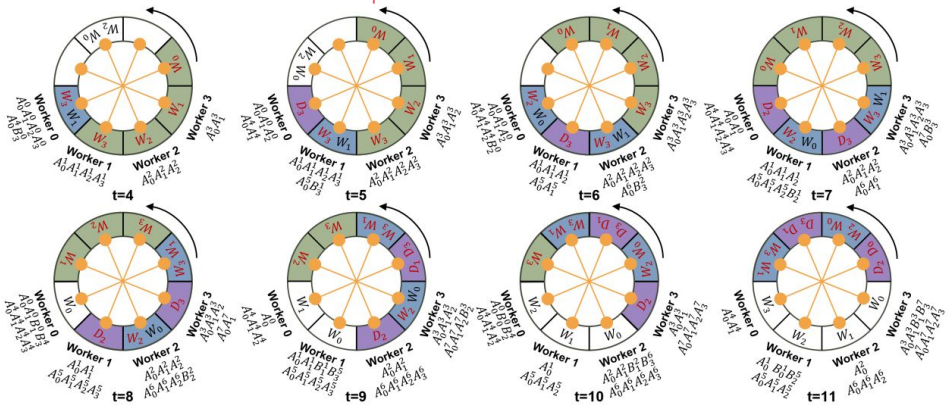
Even if the forward and backward computations are unbalanced, the pipeline will have no bubble during the forward-backward interleave stage. The communication will be efficient leading to better computation to communication ratio

WeiPipe-Zero-Bubble-1

Zero-Bubble Pipeline divides the backward pass into B-Pass (gradient of activations) and W-pass (gradient of Weights)

Worker 0 performs forward on layer 0 (new microbatch) and B pass for layer 3 (original microbatch)
W_1 is stored together with W_3

Worker 0 performs W pass to generate D_3
Worker 0 also performs the forward pass on layer 1 (new microbatch)



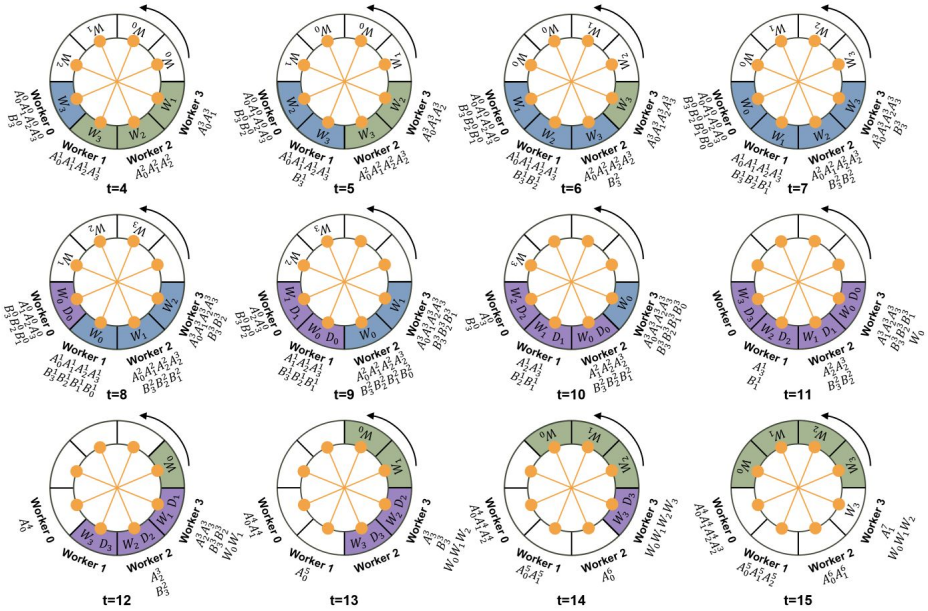
Continue alternating one-forward-one-B and one-forward-one-W until forward on new microbatch is complete.

Until t=11, alternate between two B passes and two W passes when all the passes for the original microbatch are complete.
Then repeat for a new microbatch

WeiPipe-Zero-Bubble-2

After the forward pass is completed in all layers, then each worker performs B pass and then W pass sequentially

The W pass progresses from layer 0 to layer 3, matching the forward order.

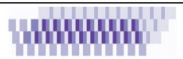


The B passes are performed in the reverse order from layer 3 to layer 0

The last worker aggregates the D_s and updates the weights with specific optimizer.

It then sends the updated weights to the next worker which initiates the new forward pass.

Theoretical Analysis

Pipeline Schemes	Bubble Ratio	Communication	Communication Distribution	Peak Memory	Memory Distribution
1F1B	$\frac{P-1}{N+P-1}$	$\frac{2PGSH}{T_F+T_{BW}}$		$\frac{GM_B+M_W+M_D}{P}+GM_A$	
Zero-Bubble 1	$\frac{P-1}{N\left(1+\frac{T_{BW}}{T_F}\right)+P-1}$	$\frac{2PGSH}{T_F+T_{BW}}$		$\frac{M_W+M_D}{P}+G(\alpha M_A+M_B)$	
Zero-Bubble 2	$\frac{P-1}{2NIter(1+\frac{T_B}{T_F+T_W})+P-1}$	$\frac{2PGSH}{T_F+T_{BW}}$		$\frac{M_W+M_D}{P}+\left(2-\frac{1}{P}\right)G(\alpha M_A+M_B)$	
WeiPipe-Interleave	$\frac{P-1}{N+P-1}$	$\frac{36PH^2}{T_F+T_{BW}}$		$\frac{GM_B+2M_W+M_D}{P}+GM_A$	
WeiPipe-Zero-Bubble 1	$\frac{P-1}{N(1+\frac{T_B}{T_F+T_W})+P-1}$	$\frac{54PH^2}{T_F+T_{BW}}$		$\frac{2G(\alpha M_A+M_B)+3\max(M_W,M_D)}{P}+1.5GM_A$	
WeiPipe-Zero-Bubble 2	$\frac{P-1}{2NIter(1+\frac{T_B}{T_F+T_W})+P-1}$	$\frac{63PH^2}{T_F+T_{BW}}$		$\frac{M_W+\max(M_W,M_D)}{P}+G(\alpha M_A+M_B)$	

Weipipe Interleave has same bubble ratio as 1F1B whereas ZB2 and WZB2 have almost no bubbles along the iteration

Communication for Activation passing pipelines increases with G and S, whereas for Weipipe, the communication is independent of G and S

WeiPipe-interleave has similar memory consumption as 1F1B but is more balanced. ZB1 and WZB2 both need to store $G(\alpha M_A + M_B)$ data. ZB2 nearly doubles this requirement.

WZB1 can achieve maximum memory consumption of just 1.5 GM_A

Experimental Setup

Model Setup
LLama-2 models
Number of head and layers set to 32

Variable H(hidden-dimension), S(sequence length), G(micro-batch size) and N(number of micro-batches)

Baselines
1F1B (Megatron-LM)
Zero-bubble 1 (Megatron-LM)
Zero-bubble 2 (Megatron-LM)
FSDP (DeepSpeed)

Hardware
Colossal Cloud, A800 GPUs
- 80GB HBM
- 312 TFlops fp16/bf16
- 400 GB/s bandwidth size

Communication Infrastructures
- 16 A800 with NVLink
- 32 A800 across 4 cluster; NVLink within the clusters, 10GbE across clusters
- NCCL as communication library with ring-based implementation

Evaluation: Throughput

Model Config			Throughput(Tokens/second/GPU)					Memory(GB)				
H	S	G	1F1B	ZB1	ZB2	FSDP	WeiPipe	1F1B	ZB1	ZB2	FSDP	WeiPipe
1024	4096	16	8581.7	7547.0	7638.5	11525.9	**15138.8**	13.0	20.4	39.3	8.6	9.4
	8192	8	7403.8	6739.6	6768.1	9424.4	**12122.3**	9.9	10.7	20.5	8.6	9.4
	16384	4	5641.2	5651.6	5651.9	6973.6	**8188.3**	9.1	21.6	42.2	8.6	9.4
2048	4096	16	4163.2	3823.3	OOM	4104.8	**6499.7**	18.7	44.3	OOM	17.9	19.9
	8192	8	3791.3	3517.8	OOM	3706.8	**6033.2**	19.6	22.3	OOM	17.9	19.9
	16384	4	3146.3	3050.1	OOM	3087.2	**4607.8**	22.9	42.9	OOM	17.9	19.9
4096	4096	16	1662.7	OOM	OOM	1110.5	**2023.1**	40.5	OOM	OOM	39	44.5
	8192	8	1556.2	OOM	OOM	1063.2	**2059.4**	41.6	OOM	OOM	39	44.5
	16384	4	1331.6	OOM	OOM	944.2	**1684.9**	45.1	OOM	OOM	39	44.5

Fig: 16 GPU, Nvlink and ethernet connection,

When H = 4096 and S = 16384, WeiPipe achieves **22.3%** and **78.4%** improvement of throughput compared to 1F1B and FSDP respectively.

ZB strategies usually run out of memory, 1F1B maintains low memory due to Flash Attention and recomputation

Weipipe has slightly higher memory than FSDP due to buffers required for data transmission

Evaluation:Throughput

Model Config			Throughput(Tokens/second/GPU)				
H	S	G	1F1B	ZB1	ZB2	FSDP	WeiPipe
1k	4k	16	8193	7708	7952	11545	**13847**
	16k	4	5394	4583	4630	6764	**7551**
2k	4k	16	4030	3701	OOM	4205	**5587**
	16k	4	2907	2638	OOM	3150	**4151**
4k	4k	16	**1530**	OOM	OOM	1186	1402
	16k	4	1232	OOM	OOM	966	**1505**

Fig: 16 GPU, PCIe and ethernet connection

This setup lacks NVLink so it increases the effect of communication

WeiPipe-Interleave achieves **31.7%** and **22.2%** improvement of throughput compared to the best performance of other strategies when S = 16384 and H = 2048, 4096

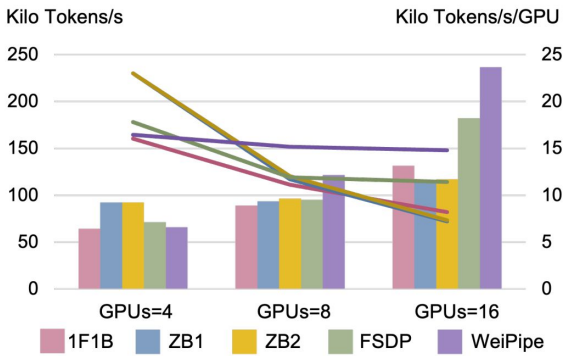
Model Config			Throughput(Kilo Tokens/second/GPU)				
H	S	G	1F1B	ZB1	ZB2	FSDP	WeiPipe
1k	4k	16	32.0	45.8	**46.5**	37.9	31.3
	16k	4	15.9	22.0	**22.1**	17.8	16.9
2k	4k	16	15.0	**22.4**	OOM	17.0	14.2
	16k	4	9.4	12.8	OOM	10.1	9.7
4k	4k	16	5.2	OOM	OOM	**6.0**	4.9
	16k	4	3.7	OOM	OOM	**3.8**	3.6

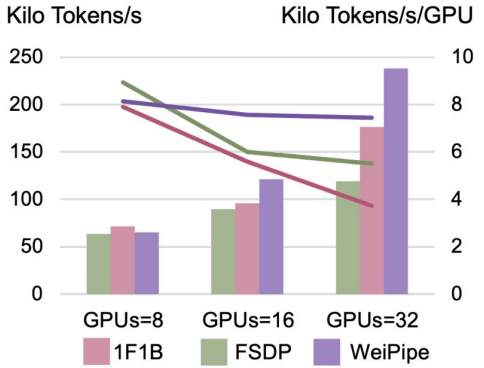
Fig: 8 GPU, NvLink connection

On environment using NVLink only, the effects of communication is less significant.

In such cases, conventional methods outperform WeiPipe

Evaluation:Weak Scaling





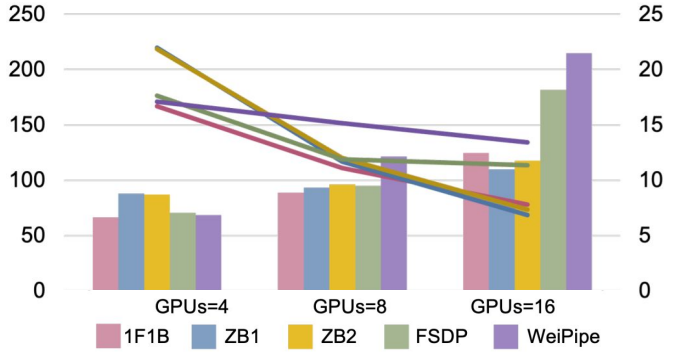
Small Scale Weak Scaling

- 4-16 GPUs
- 16 layers
- 64-256 Batch Size
- NVLink and Ethernet
- Sequence Length =16384
- Micro Batch Size = 2

Large Scale Weak Scaling

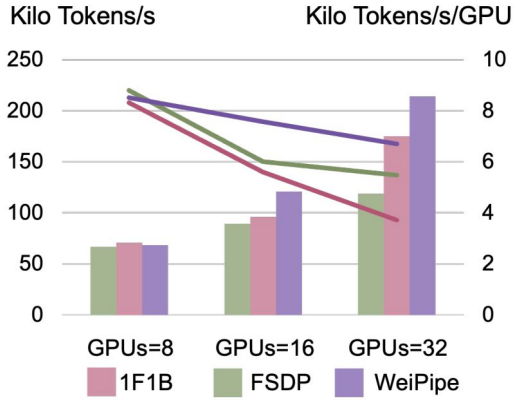
- 8-32 GPUs
- 32 layers
- 128-512 Batch Size
- NVLink and Ethernet
- Sequence Length =16384
- Micro Batch Size = 2
- ZB strategies missing because of OOM

Evaluation:Strong Scaling



Small Scale Weak Scaling

- 4-16 GPUs
- 16 layers
- 128 Batch Size



Large Scale Weak Scaling

- 8-32 GPUs
- 32 layers
- 256 Batch Size

Weipipe demonstrates higher scalability compared to the baselines in both cases

Thank You!