Benchmarking Distributed training Framework for Large Scale Models: FSDP vs DeepSpeed

Dev Mehrotra
Sai Krishna Ponnam
Rachael Lang

Problem Statement

- Training large models like Vision Transformers requires distributed training to address memory and compute bottlenecks. Different distributed frameworks vary in performance. There is a lack of detailed work comparing their efficiency
- Our study aims to fill this gap by benchmarking **FSDP** and **DeepSpeed** comparing key metrics, This work provides insights into the strengths and weaknesses of both frameworks, helping researchers and practitioners optimize their large-scale training pipelines.

Related work

- FSDP uses full and hybrid sharding to distribute model parameters, minimizing memory usage by only storing necessary parts of the model on each device.
- DeepSpeed leverages ZeRO (Zero Redundancy Optimizer) stages to partition optimizer states, gradients, and activations.

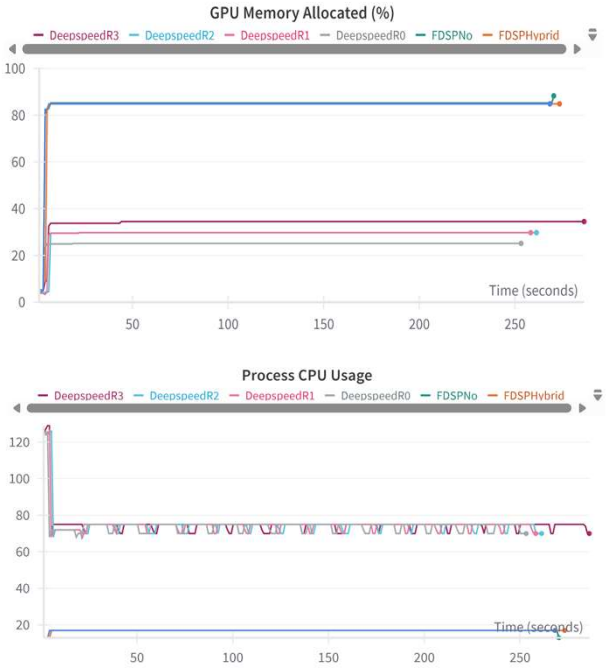
Approach

- We benchmark two leading distributed frameworks:
  - PyTorch FSDP: Full and hybrid sharding, mixed precision (FP16/BF16)
  - DeepSpeed: ZeRO-0/1/2/3 optimization stages, activation checkpointing, and precision tuning.
- The goal is to compare scalability, memory usage, throughput, and convergence behavior.

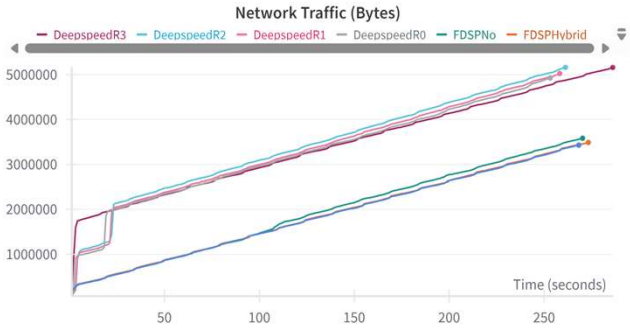
Design

We trained for 15 epochs on the CIFAR-10 dataset with a learning rate of 0.01, using two CloudLab nodes, each with 4 GPUs (NVIDIA A100-SXM4-40GB). Configurations included FSDP and DeepSpeed with various batch sizes. We tracked 25 metrics, including CPU/GPU memory usage, network bandwidth, throughput, training time, error rates, and thread/process counts.

Early results



•Single Node: FSDP excelled for smaller models on a single node speeding up training by 13% across all bath sizes.
•Multi-Node: DeepSpeed outperformed FSDP using upto 30% less memory at cost of more network bandwidth.



Config	Best For	When to Use
FSDP	Smaller models	When memory efficiency and single-node performance are key.
DeepSpeed ZeRO-2	Large models, multi-node	For large models with distributed memory optimization.
DeepSpeed ZeRO-3	Very large models	For massive models needing full memory and compute scaling.
DeepSpeed ZeRO-1	Medium to large models	For models needing optimized memory without full checkpointing.
DeepSpeed ZeRO-0	Basic memory optimization	For smaller models or less complex optimization needs.

Next Steps / WIP

- **DeepSpeed** is better for large-scale models across multiple nodes, FSDP remains highly effective for single-node training, especially when memory constraints are less critical.
- **Explore data loading and communication overhead** to assess their impact on scalability.
- **Monitor convergence behavior** during extended training.
- **Expand experiments to larger models** to fully evaluate performance differences across both frameworks.

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON