SCALABLE GRAPH NEURAL NETWORK (GNN) TRAINING FOR TRACK FINDING

Ivan Ladutska Brenden Reeves, Caroline Manjerovic, Alina Lazar, Tuan Minh Pham, Chen-Hsun Chan, Daniel Murnane, Xiangyang Ju, Paolo Calafiura

Abstract

Graph Neural Networks (GNNs) have demonstrated significant performance in addressing the particle **track-finding problem** in High-Energy Physics (HEP). Traditional algorithms exhibit high computational complexity in this domain as the number of particles increases. We address the challenges of training GNN models on large, rapidly evolving datasets, a common scenario given the advancements in data generation, collection, and increase in storage capabilities. The computational and GPU memory requirements present significant roadblocks in efficiently training GNNs on large graph structures. One effective strategy to reduce training time is **distributed data parallelism** (DDP) on multi-GPUs, which involves averaging gradients across the devices used for training.

TrackML Dataset

- The **TrackML** dataset consists of independent events generated through a Monte Carlo simulation of proton-proton collisions.
- The full dataset contains **10,000** events. During preprocessing, we convert these coordinates to spherical coordinates.
- This dataset is notable because the simulations give us information about the **ground truth**: we know the "actual" tracks of each particle. We can compute the accuracy in terms of efficiency and purity.

Hardware Platform

Fig. 1: Perlmutter Suppercomputer

Perlmutter supercomputer hosted at the the National Energy Research Scientific Computing Center (NERSC) has 6,159 NVIDIA A100 Tensor Core GPUs with 80GB memory per GPU and 4 GPUs per node.

Distributed Data Parallelism (DDP)

The goal is to investigate the scalability of DDP in relationship with the number of GPUs used.
What is the impact of varying GPU numbers on training runtime acceleration and model performance?

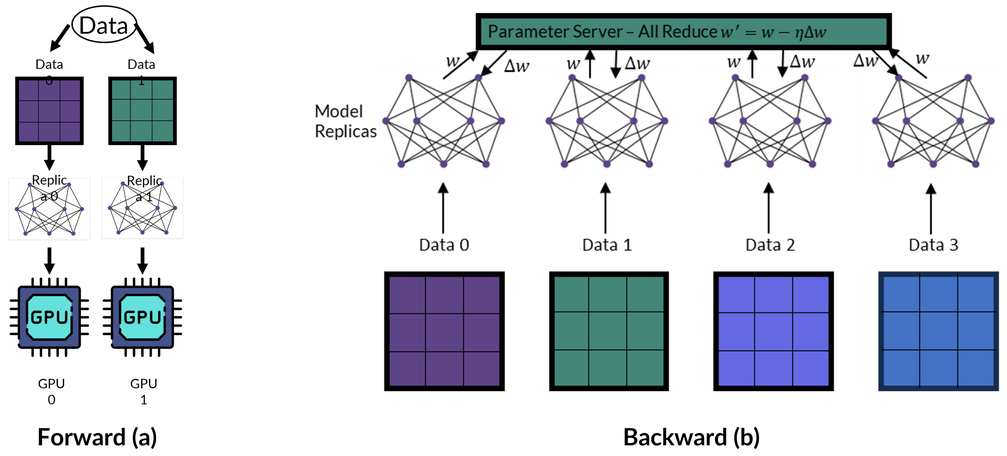

Fig. 2: Each process performs a full forward (a) and backward (b) pass in parallel

DDP Speed up and Performance

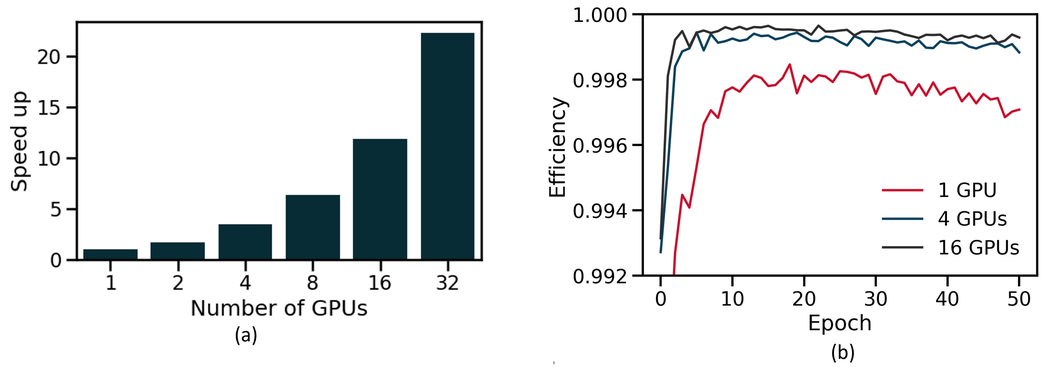

Fig. 3: Speed up of running DDP (a) and efficiency on the validation dataset (b)

Table 1: Validation Performance Metrics during Training and Runtime

# of GPUs	Efficiency (%)	Target Purity v	Total Purity (%)	Purity (%)	AUC	Validation Loss	Epoch (s)	Batch (s)
1	99.6	87.831	99.249	99.152	98.577	0.00406	34.36	0.24
2	99.778	88.431	99.513	99.452	98.842	0.00226	39.63	0.29
4	99.838	88.508	99.587	99.536	98.846	0.00176	38.73	0.28
8	99.871	88.524	99.658	99.616	98.841	0.00141	42.12	0.31
16	99.903	88.499	99.674	99.633	98.884	0.00114	45.48	0.34
32	99.945	87.931	99.443	99.371	98.79	0.00124	47.6	0.37

Distributed Data Parallel

1. Each GPU across each node gets its own process.
2. Each GPU gets visibility into a subset of the overall dataset. It will only ever see and train on that subset.
3. Each process initializes a copy of the model.
4. Each process performs a full forward and backward pass in parallel as shown in figure 2.
5. The gradients are synchronized and averaged across all processes when triggered by a signal.

GPU and Memory Utilization

Table 2: Max GPU Utilization and Memory on the Main Node

# of GPUs	GPU Util (%)	Memory (GB)	Speed up
1	59.93	20.52	1
2	65.8	22.23	1.67
4	65.88	23.58	3.52
8	68.02	26.51	6.31
16	73.91	26.18	11.86
32	80.91	23.37	22.26

Discussion and Conclusions

- **Data Parallelism:** DDP allows you to split your dataset across multiple GPUs (figure 2).
- **Reduced Training Time:** By distributing the workload across multiple devices, DDP can significantly reduce the training time. However, as shown in figure 3 (a) the speed up is not linear with the number of GPUs.
- **Increase Performance:** By utilizing DDP effectively, you can process more data and increase performance as shown in figure 3 (b) and in table 1.

Acknowledgements

This research was supported by the U.S. Department of Energy's Office of Science, Office of High Energy Physics, of the US Department of Energy under Contracts No. DE-SC0024364 (FAIR).
This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 and the Ohio Supercomputer Center (OSC).

BERKELEY LAB