EML Lab 3

Name: Nandan Bharatkumar Parikh
GT ID: 903487850

Task 1

Nvidia-smi

```
[nparikh44@atl1-1-03-012-8-0 ~]$ nvidia-smi
Thu Oct 30 15:06:27 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 575.57.08              Driver Version: 575.57.08      CUDA Version: 12.9     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA H100 80GB HBM3          On  |   00000000:0A:00.0 Off |                    0 |
| N/A   38C    P0             95W /  700W |       0MiB /  81559MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   1  NVIDIA H100 80GB HBM3          On  |   00000000:90:00.0 Off |                    0 |
| N/A   32C    P0             81W /  700W |       0MiB /  81559MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   2  NVIDIA H100 80GB HBM3          On  |   00000000:BE:00.0 Off |                    0 |
| N/A   33C    P0             79W /  700W |       0MiB /  81559MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   3  NVIDIA H100 80GB HBM3          On  |   00000000:C7:00.0 Off |                    0 |
| N/A   39C    P0             81W /  700W |       0MiB /  81559MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

The output indicates a node with four NVIDIA H100 GPUs (each with 80 GB of HBM3). All GPUs are detected, currently idle, and have ECC turned on, with MIG turned off. They are connected through PCIe Gen 5 along with NVLink or NVSwitch, giving high-bandwidth GPU-to-GPU communication.

`nvidia-smi topo -m`

```
[nparikh44@atl1-1-03-012-8-0 ~]$ nvidia-smi topo -m
        GPU0    GPU1    GPU2    GPU3    NIC0    CPU Affinity    NUMA Affinity   GPU NUMA ID
GPU0     X      NV18    NV18    NV18    SYS     0-19    0               N/A
GPU1    NV18     X      NV18    NV18    NODE    32-51   1               N/A
GPU2    NV18    NV18     X      NV18    PXB     32-51   1               N/A
GPU3    NV18    NV18    NV18     X      PIX     32-51   1               N/A
NIC0    SYS     NODE    PXB     PIX      X

Legend:

  X    = Self
  SYS  = Connection traversing PCIe as well as the SMP interconnect between NUMA nodes (e.g., QPI/UPI)
  NODE = Connection traversing PCIe as well as the interconnect between PCIe Host Bridges within a NUMA node
  PHB  = Connection traversing PCIe as well as a PCIe Host Bridge (typically the CPU)
  PXB  = Connection traversing multiple PCIe bridges (without traversing the PCIe Host Bridge)
  PIX  = Connection traversing at most a single PCIe bridge
  NV#  = Connection traversing a bonded set of # NVLinks

NIC Legend:

  NIC0: mlx5_0
```

- All four H100 GPUs are connected through NVLink (NV18), giving direct high-bandwidth GPU-to-GPU communication without PCIe routing.
- The GPUs are split across two CPU sockets: GPUs 0–1 on one socket and GPUs 2–3 on the other, forming two NUMA domains.
- In a NUMA setup, each CPU has its own local memory, and cross-socket memory access is slower than local access.
- This topology provides fast intra-socket GPU communication and a balanced layout for distributed or pipeline parallel training.

Task 2

Broadcast	Running Time (in s)
broadcast1	1.504
broadcast2	0.404
broadcast3	0.103

In **broadcast1**, every GPU pulls the data from the CPU on its own. That means four separate host-to-device transfers that run one after another and depend on the CPU, so it ends up being the slowest (about 1.5 s).

In **broadcast2**, the data moves from the CPU to GPU 0 only once. GPU 0 then forwards it to the other GPUs through peer-to-peer transfers over NVLink. This avoids duplicated host transfers and speeds things up to around 0.4 s, though the steps still run one after another.

In **broadcast3**, the data is split into large pieces and moved using asynchronous streams and events. While GPU 0 is receiving the next chunk from the CPU, earlier chunks are already being

pushed to the other GPUs over NVLink. This overlap keeps both host-to-device and device-to-device links busy and brings the total time down to roughly 0.1 s.

Task 3

Step 1

Bert Config

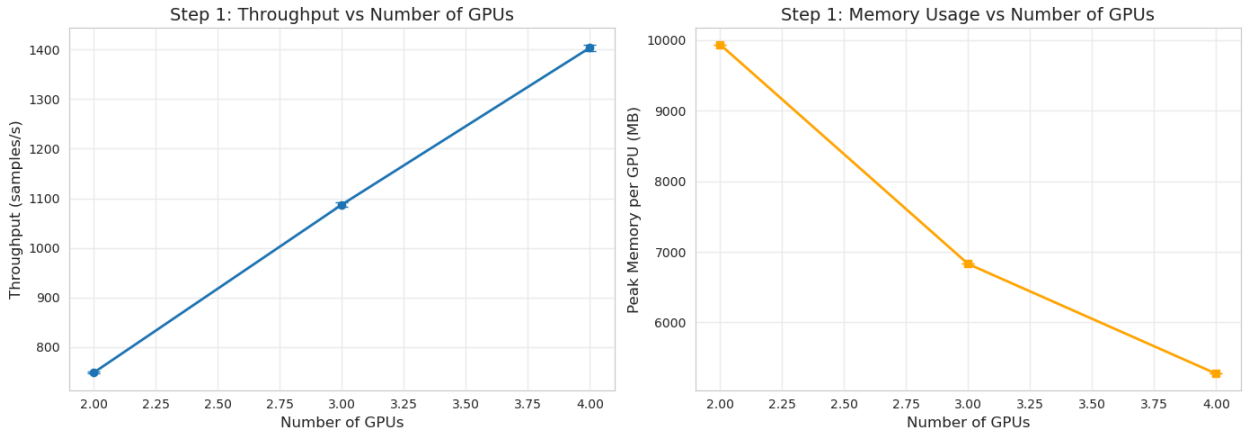

```
BertConfig {
  "attention_probs_dropout_prob": 0.1,
  "classifier_dropout": null,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "position_embedding_type": "absolute",
  "return_dict": false,
  "transformers_version": "4.56.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 30522
}
```


Step 2

Performance v/s Number of GPUs

Description	Throughput (samples/s)	Avg Memory (MB)
2 GPUs	748.5 ± 2.3	9936.4
3 GPUs	1087.3 ± 5.0	6828.2

4 GPUs	1403.8 ± 6.5	5274.1

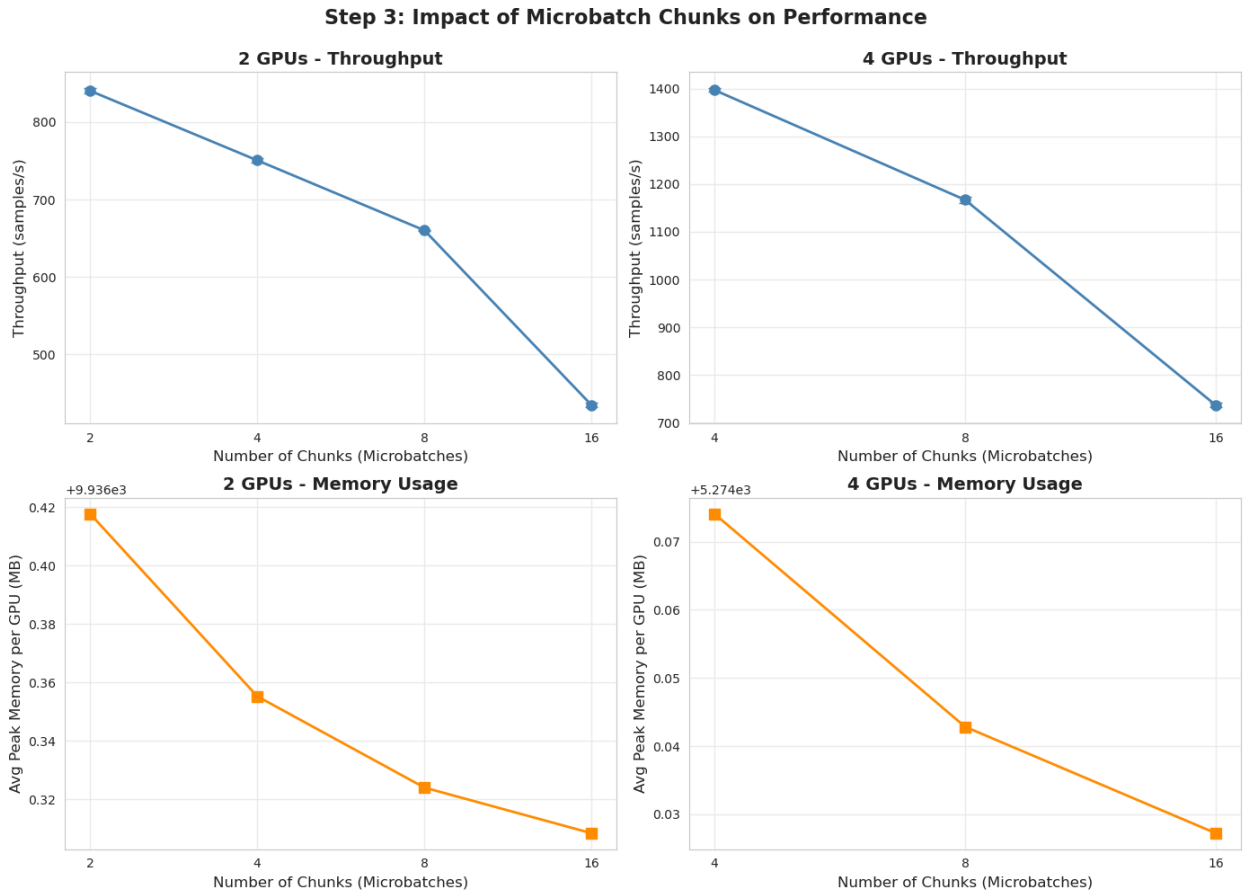


Observations:
Scaling from 3 to 4 GPUs yielded a **1.29x speedup**, achieving 96.8% of the ideal 1.33x scaling factor. The 4-GPU configuration delivered **1403.8 ± 6.5 samples/s compared to 1087.3 ± 5.0** samples/s for 3 GPUs which is a 29.1% throughput improvement. This near-linear scaling demonstrates effective pipeline parallelism with minimal overhead from inter-GPU communication and pipeline bubbles on high-bandwidth H100 interconnects.

Memory distribution improved with additional parallelism: average peak memory per GPU decreased by 22.8% (from 6828 MB to 5274 MB), as the model was partitioned into smaller stages. However, the per-rank memory profile shows slight imbalance, likely due to the uneven distribution of BERT encoder layers (12 layers split across ranks). The first and last stages include additional embedding and pooling layers, resulting in marginally higher memory footprints on those ranks.

The **low standard deviation** across seeds (< 6.50 samples/s) indicates stable performance, and the high efficiency suggests that the 4-microbatch configuration effectively masks pipeline latency. On H100 GPUs with NVLink/NVSwitch, communication overhead remains negligible, enabling strong scaling for this model size and batch configuration.

Step 3



On H100s the pipeline bubble is not the main bottleneck at low chunk counts. The compute on each microbatch is very fast, so the idle time created by the bubble is a small fraction of the total step time. When we increase the number of chunks the microbatches become smaller, which increases the number of kernel launches, synchronization points, and communication steps. These overheads grow faster than the small benefit we get from reducing the bubble. As a result the throughput does not rise at first and instead drops immediately.

Step 4

Split	Throughput (sample / s)	Per-Rank Memory (MB)
2-2-2-2	1403.811 ± 6.497	**Rank 0:** 5322.4 **Rank 1:** 5226.20 **Rank 2:** 5226.20 **Rank 3:** 5322.40
6-2-2-2	741.699 ± 3.051	**Rank 0:** 9938.87 **Rank 1:** 3687.01 **Rank 2:** 3687.01 **Rank 3:** 3784.52

Observations
Baseline even partition throughput: 1403.81 ± 6.50 samples/s.
Uneven partition throughput: 741.70 ± 3.05 samples/s (-662.11 samples/s, -47.17% change).

Stage time balance (CV) shifts from 0.0485 (even) to 0.6399 (uneven), indicating worse load balancing across ranks.
Memory balance (CV) changes from 0.0092 to 0.5105, showing greater variance in per-rank peak usage.

Per-rank timings and memory suggest that front/back heavy stages in the uneven layout do not improve end-to-end throughput. Any gain/loss reflects trade-offs between alleviating a straggler stage vs introducing imbalance elsewhere.

Overall, the uneven split does not improve throughput relative to the even baseline while degrading stage time balance.

Task 4

Aggregate Metrics

Method	Throughput (samples / s)	Time/Step (s)
DDP	415.166 ± 0.104	0.154155 ± 0.000039
FSDP (Full)	414.292 ± 0.127	0.154480 ± 0.000047

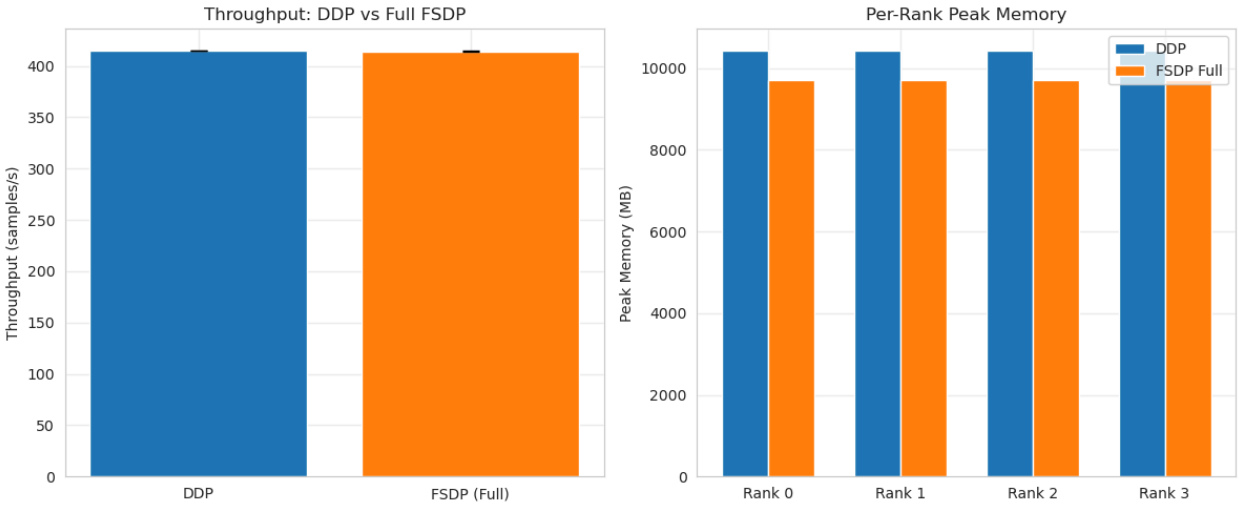
Per Rank Metrics

Method	Rank	Peak
DDP	0	10438.31 ± 0.96
	1	10437.81 ± 0.31
	2	10437.81 ± 0.31
	3	10437.81 ± 0.31
FSDP (Full)	0	9705.47 ± 170.56
	1	9705.47 ± 170.55

	2	9705.47 ± 170.55
	3	9705.47 ± 170.55

Observation

Setup: 4 GPUs, global batch size 64, BERT-base, 3 seeds, no AMP.



Throughput: DDP achieves 415.17 ± 0.10 samples/s; Full FSDP reaches 414.29 ± 0.13 samples/s (Δ -0.87 samples/s, -0.210%). This indicates near parity: full parameter sharding does not materially degrade step throughput at this scale. Slightly lower FSDP throughput likely reflects added all-gather / reduce-scatter synchronization overhead, yet high-bandwidth interconnect keeps the penalty minimal (<0.1%).

Latency: Mean time per step shifts from 0.154155s (DDP) to 0.154480s (FSDP), a +0.211% change which is consistent with the small throughput delta.

Memory: Average per-rank peak memory drops from 10437.9 MB (DDP) to 9705.5 MB (FSDP), a reduction of 7.02%. This saving derives from sharding model parameters (and optimizer states, if present) instead of fully replicating them across ranks.

Interpretation: Full FSDP provides a meaningful ~7.0% memory headroom without sacrificing throughput. This headroom enables either larger batch sizes, deeper models, or activation checkpointing trade-offs. For this model scale, communication overhead is effectively masked by compute; scaling to more GPUs or larger hidden dimensions would further differentiate memory profiles while potentially amplifying communication cost.

Recommendation: Prefer Full FSDP when pushing model or batch size limits; retain DDP when simplicity or slightly lower synchronization complexity is desired. Given near-identical

performance, migration cost should be minor compared to benefits for memory-constrained scenarios.

Bonus Section - FSDP v/s DeepSpeed

Aspect	FSDP	DeepSpeed
Memory behavior	Full sharding of params/grad/optimizer. Hence, good baseline savings	ZeRO stages with stronger memory reduction; supports NVMe/CPU offload
Offload	Limited and newer	Mature CPU/NVMe offload (ZeRO-Offload, ZeRO-Infinity)
Pipeline / model parallel	Works but more manual	Built-in pipeline + tensor parallel support
Config complexity	Simpler, PyTorch-native	More knobs, JSON configs, more tuning
Ecosystem	First-party PyTorch	Large ecosystem; used for very large models

Scenarios

- **FSDP:** Want PyTorch-native training, simpler configuration, and strong memory savings without external tooling.
- **DeepSpeed:** Need aggressive offload, pipeline/tensor parallel integration, or support for very large models that exceed GPU memory.