Third Biweekly Update: Traffic Pattern Analysis and Comparison of Distributed Deep Learning Models

Li He V00972954

Progress Summary

This biweekly update focuses on resolving technical challenges related to module loading and environment setup on the Alliance Canada clusters, as well as initial steps toward capturing network performance during distributed deep learning experiments. While the setup now supports scalable multi-GPU training, capturing network performance metrics remains a challenge. To address this, I explored ASTRA-sim, a distributed machine learning simulator, and successfully deployed it locally using Docker.

Problem Solved

- **Module Loading Issues:** For the issue mentioned in the mid-term update, after troubleshooting, the Python version conflicts were resolved by ensuring compatibility between the installed PyTorch version and the Python version in the virtual environment. The myenv environment was successfully configured with the required dependencies, allowing the execution of distributed training scripts.
- **Slurm Job Submission:** I have successfully configured and executed distributed training jobs using Slurm job submission scripts on the Alliance Canada clusters. This allows me to efficiently run large-scale experiments in a controlled environment.
- I successfully ran a distributed deep learning training script using PyTorch Distributed Data Parallel (DDP). It trains a Wide ResNet model on the FashionMNIST dataset using multiple GPUs across multiple nodes.
  - Implements PyTorch DDP for scalable multi-GPU training.
  - Uses distributed samplers for efficient data parallelism.
  - Synchronizes gradients, accuracy, and loss across GPUs.

```
Epoch =  1: Cumulative Time = 81.000, Epoch Time = 81.000, Images/sec =
    740.742431640625, Validation Loss = 0.354, Validation Accuracy = 0.878
Communication Time (all_reduce): 0.006016 seconds
Epoch =  2: Cumulative Time = 159.930, Epoch Time = 78.931, Images/sec =
    760.1616821289062, Validation Loss = 0.249, Validation Accuracy = 0.910
Communication Time (all_reduce): 0.007944 seconds
Early stopping after epoch 2
```

Epoch 1 Summary:

Time Taken: 81.000 seconds.
Throughput: 740.74 images/second.
Validation Loss: 0.354.
Validation Accuracy: 87.8%.
Communication Time for all_reduce (gradient synchronization): 0.006016 seconds.

Epoch 2 Summary:

Time Taken: 78.931 seconds .
Throughput: 760.16 images/second .
Validation Loss: 0.249.
Validation Accuracy: 91.0% (exceeds the target accuracy of 0.85).
Communication Time for all_reduce: 0.007944 seconds.

Ongoing Challenges

- **Capturing Network Performance Metrics:** While I can now run my distributed training script on the Alliance Canada clusters, I am facing challenges in developing a method to capture and analyze network performance data. This step is crucial for understanding the traffic patterns and communication overhead in distributed deep learning training.

  To cope with the above challenge, I looked into alternative tools and I found the simulator ASTRA-sim (https://github.com/astra-sim/astra-sim) which is a distributed machine learning system simulator developed by Intel, Meta, and Georgia Tech. The integration of Network Simulator in ASTRA-sim enables us to easily analyze network performance. More importantly, it allows us to customize the topology for the distributed training such as parameter servers, all-connected, ring, etc. This salient feature perfectly matches with my initial goal of traffic pattern study with varying topologies. Hence, I put efforts into gaining insights into how to use this simulator and try to run it locally.

  After exploration, I realized there are two methods to deploy the simulator locally: one is the common way, i.e., installing necessary dependencies and compiling the source code. Another one is to deploy the system in a docker way. After trying, the docker way is easier to deploy. As of now, I have successfully installed the docker manager and the ASTRA-sim container on my local machine. With this environment, the output of running a distributed training exercise is given below:

```
Success in opening system file
Var is: scheduling-policy: ,val is: LIFO
Var is: endpoint-delay: ,val is: 10
Var is: active-chunks-per-dimension: ,val is: 1
Var is: preferred-dataset-splits: ,val is: 4
Var is: boost-mode: ,val is: 1
Var is: all-reduce-implementation: ,val is: ring
Var is: all-gather-implementation: ,val is: ring
Var is: reduce-scatter-implementation: ,val is: ring
Var is: all-to-all-implementation: ,val is: direct
Var is: collective-optimization: ,val is: localBWAware
Var is:  ,val is: localBWAware
The final active chunks per dimension after allocating to queues is: 100000000
ring of node 0, id: 0 dimension: local total nodes in ring: 8 index in ring: 0
    offset: 1total nodes in ring: 8
ring of node 0, id: 0 dimension: local total nodes in ring: 8 index in ring: 0
    offset: 1total nodes in ring: 8
ring of node 0, id: 0 dimension: local total nodes in ring: 8 index in ring: 0
    offset: 1total nodes in ring: 8
ring of node 0, id: 0 dimension: local total nodes in ring: 8 index in ring: 0
    offset: 1total nodes in ring: 8
total nodes: 8
LogGP model, the local reduction delay is: 1
LogGP model, the local reduction delay is: 1
LogGP model, the local reduction delay is: 1
LogGP model, the local reduction delay is: 1
Shared bus modeling enabled? false
LogGP model, the L is:0 ,o is: 0 ,g is: 0 ,G is: 0.0038
communication delay (in the case of disabled shared bus): 10
Shared bus modeling enabled? false
LogGP model, the L is:0 ,o is: 0 ,g is: 0 ,G is: 0
communication delay (in the case of disabled shared bus): 10
Success in opening workload file
id: allreduce , depen: -1 , wg_comp_time: 1
type: MICRO ,num passes: 1 ,lines: 1 compute scale: 1 ,comm scale: 1
stat path: /app/tutorials/mlsys2022/exercise_1/result/ ,total rows: 1 ,stat row: 0
CSV path and filename: /app/tutorials/mlsys2022/exercise_1/result/detailed.csv
Success in opening CSV file for writing the report.
CSV path and filename: /app/tutorials/mlsys2022/exercise_1/result/EndToEnd.csv
Success in opening CSV file for writing the report.
CSV path and filename: /app/tutorials/mlsys2022/exercise_1/result/
    backend_end_to_end.csv
Success in opening CSV file for writing the report.
CSV path and filename: /app/tutorials/mlsys2022/exercise_1/result/backend_dim_info.
    csv
Success in opening CSV file for writing the report.
CSV path and filename: /app/tutorials/mlsys2022/exercise_1/result/tutorial_result.
```

```
    csv
Success in opening CSV file for writing the report.
info: all-reduce weight grad collective issued for layer: allreduce with size:
    1048576, involved dimensions:  1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
***** info: weight gradient collective for layer: allreduce is finished***********
******************
Layer id: allreduce
Total collectives issued for this layer: 1
************************  Workload stats  ************************ allreduce
id: allreduce ,Total cycles spent on fwd pass compute: 0
id: allreduce ,Total cycles spent on weight grad compute: 0
id: allreduce ,Total cycles spent on input grad compute: 0
id: allreduce ,Total cycles spent idle waiting for fwd finish: 0
id: allreduce ,Total cycles spent idle waiting for weight grad finish: 45681
id: allreduce ,Total cycles spent idle waiting for input grad finish: 0
id: allreduce ,Total cycles spent on fwd pass comm: 0
id: allreduce ,Total cycles spent on weight grad comm: 45681
id: allreduce ,Total cycles spent on input grad comm: 0
************************  Queuing stats  ************************ allreduce
id: allreduce ,Average cycles spent on queuing for phase 0 of algorithm (per chunk)
    : 0
id: allreduce ,Average cycles spent on queuing for phase 1 of algorithm (per chunk)
    : 17130
************************  Network stats  ************************ allreduce
id: allreduce ,Average cycles spent on network for phase 1 of algorithm (per
    message): 805
************************  Chunk Stats Per Logical Dimension (for all layers)
    ************************ allreduce
 ,Average chunk latency for logical dimension  1 of topology: 11420
************************
all passes finished at time: 45681, id of first layer: allreduce
path to create csvs is: /app/tutorials/mlsys2022/exercise_1/result/
success in openning file
*****
Time to exit: Sat Mar 22 04:27:23 2025
all-reduce Collective implementation: ring
reduce-scatter Collective implementation: ring
all-gather Collective implementation: ring
all-to-all Collective implementation: direct
Collective optimization: localBWAware
Total sim duration: 0:0 hours
Total streams injected: 4
Total streams finished: 4
Percentage of finished streams: 100 %
```

Next Step

- Put more efforts into understanding some unique features of ASTRA-sim, which might be useful for my future project.
- Go deeper into the code structure of ASTRA-sim and fully understand the performance evaluation part.
- Last but not least, correlate this simulator with Alliance Canada platform, thinking the limitations and advantages of each platform and thus we can better understand how to fully utilize these resources.