**Lab 3: Pipeline Parallelism**

**Radha Gulhane  
Date: 09 - Dec - 2022**

Q. 1 Evaluate and discuss the performance trend of pipeline parallelism for AlexNet on 256 \* 256 image size (Resized CIFAR-10 Dataset) on 1, 2, and 4 nodes (12 Experiments) (50 Points)

Test the following batch size 512  
For the above batch size, specify “train\_micro\_batch\_size\_per\_gpu” as BS/4 , BS/8  
Pipeline Size 1 Node: P = 0, 2 Nodes: P = 0 and 2, 4 nodes: P = 0, 2, and 4

=>  
**Experiment Results:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Pipeline size | | | Pipeline size | | |
|  | 0 | 2 | 4 | 0 | 2 | 4 |
| Micro Batch | 128 | | | 64 | | |
| No. Node |  |  |  |  |  |  |
| 1 | 1013.259 |  |  | 792.117 |  |  |
| 2 | 1574.769 | 797.796 |  | 1914.268 | 788.122 |  |
| 4 | 3138.038 | 1552.092 | 832.914 | 3238.214 | 1506.1 | 759.003 |

**Evaluation and Analysis:**  
Figure 1. Shows the Pipeline Parallelism using pipeline size is same as node size. Here, we can observe that as we increase the nodes, throughput is decreasing, in case of micro batch size 128. However, with 64 micro-batch size drop is not significant.  
Through pipeline parallelism provides advantage over model parallelism using pipeline mechanism, at some stages, GPU will be still ideal as GPU node need to be rely on previous GPU nodes due to distribution of layers. As we decrease micro batch size, each GPU will be able to process layers faster and it may improve the throughput.

Figure . Pipeline Prallelism AlexNet  
# nodes = pipeline size

The opposite trend can be observed in the case of pipeline size as 0. In this case, DeepSpeed will utilize all the nodes without distributing the model to different nodes. Thus, while increasing the number of nodes, throughput is also increasing, as complete model is on same GPU, it doesn’t not need to reply on next GPU for input/ output. Figure 2. Shows that throughout with micro batch size 128 and 64 with increase in number of nodes.

Figure Pipeline Parallelism AlexNet  
pipeline size = 0

A similar trend can be seen in the case of pipeline size as 2, i.e. increase in number of nodes (i.e. no. of GPUs), and keeping same pipeline size, throughput is increasing.

Q.2 Evaluate and discuss the performance trend of pipeline parallelism for VGG19 on 512 \* 512 image size (Resized CIFAR-10 Dataset) on 1, 4, and 8 nodes (16 Experiments) (50 points)

Test the following batch size 128  
For the above batch size, specify “train\_micro\_batch\_size\_per\_gpu” as BS/4, BS/16  
Pipeline Size: 1 Node: P=0, 4 nodes: P = 0, 2, and 4, 8 nodes: P = 0, 2, 4, and 8

=>

**Experiment Results:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Pipeline size | | | | Pipeline size | | |  |
|  | 0 | 2 | 4 | 8 | 0 | 2 | 4 | 8 |
| **Micro Batch** | **32** | | | | **8** | | | |
| No. Node |  |  |  |  |  |  |  |  |
| 1 | Memory Size Exceed | | | | 21.036 |  |  |  |
| 4 | 73.975 | 52.15 | 40.483 |  |
| 8 | 144.542 | 103.946 | 77.05 | 56.155 |

**Evaluation and Analysis:**Error observed while running the experiment for 32 micro batch size:   
Error: **torch.cuda.OutOfMemoryError: CUDA out of memory.**It is because of **memory requirement of input images and model size.** VGG model is itself large but can be fit in the memory with micro batch size as 8.  
As compared to AlexNet, where AlexNet able to run successfully using with 128 micro batch size and 0 pipeline size, VGG model is large, and thus it requires to run with greater number of nodes, thus layers can be further distributed and can be train with higher micro batch size.

Figure3 shows the Pipeline parallelism for VGG 19 in case of pipeline size same as number of nodes, we can observe the increase in throughput with increase in number of nodes. As observed in AlexNet with micro batch size as 64, where there is very less drop in throughput as compared to micro batch1 128, In VGG with micro batch size as 8, throughput performance is increasing. It can be said that GPU utilization will be better if we decrease the micro batch size.

Figure VGG19 pipeline Parallelism  
# nodes = pipeline size

Similar to Alex Net, as we increase the number of nodes, throughput is also increasing irrespective of pipeline size.

Figure VGG19 pipeline Parallelism  
pipeline size = 0