# Reseach on Task Sharding



## Target

The goal is to split the training task of a model (i.e. GPT3 and T5) among several computing nodes using "teamwork", in order to reduce the computational requirements of each node and involve more users in the training task.

## Proposed Solution

When training large models, in order to improve efficiency and solve memory limits, the training is divided among several GPUs. In one iteration, each GPU trains the current data/model separately, and then the training processes are synchronized through communication.

Two common parallel techniques can be used as reference:

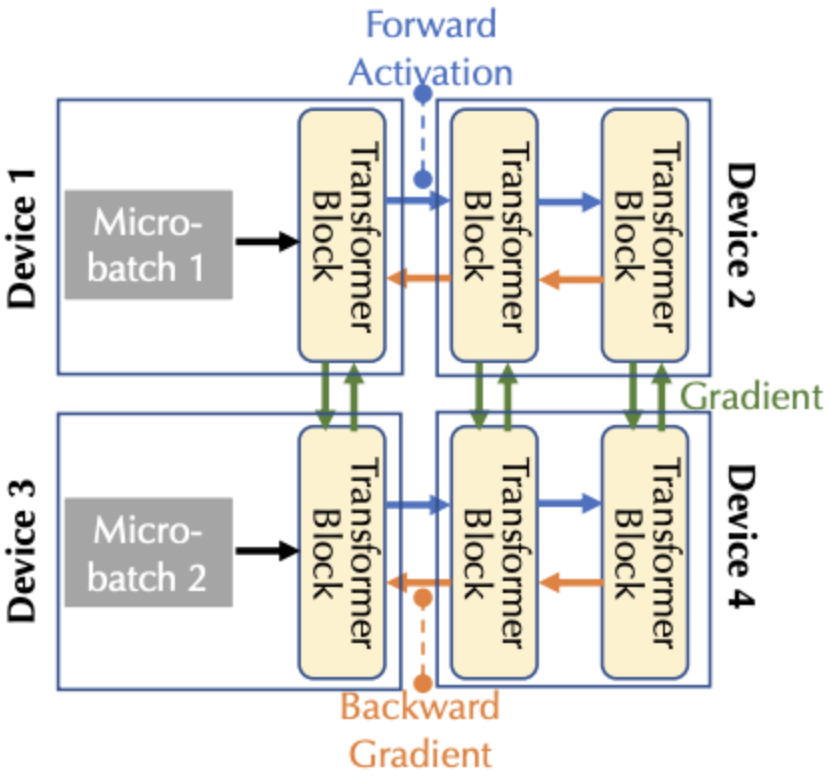
1. Distributed Data Parallel (DDP) is a parallel training strategy that accelerates the training process of deep learning models. In distributed training, the dataset is split (or "partitioned") into multiple computing devices, such as GPUs. Each device performs forward pass and backpropagation separately, but only processes a subset of the entire dataset. Then, all devices synchronize their model parameter gradient updates at the end of each training step.
2. Pipeline parallelism is another parallel computing strategy for deep learning models, mainly used to handle limited computing resources but huge model sizes. In this strategy, different parts (or stages) of the model are computed on multiple devices. This method is similar to the assembly line in a factory, so it is also called "pipeline parallelism". Specifically, in PP, the input data first enters the first device and is computed through the first stage of the model, and then

the computed result is sent to the next device for computation of the second stage, and so on. The advantage of this is that while the first device is processing the first stage of the second input data, the second device can simultaneously process the second stage of the first input data, thereby effectively improving the computational efficiency.

In addition, there are some other more advanced parallel strategies, such as Megatron and ZeRO-S3, but these strategies require more frequent communication to synchronize training progress. In the current setup, the training nodes that perform subtasks are not in the same physical location, so training synchronization cannot be achieved through physical connections such as NVLINK. Lower bandwidth interconnection is a major bottleneck here. If the data transmission time is larger than the proportion of GPU training time for the model, then our distributed training will not be cost-effective.

In 2022, AI top conference NeurIPS, there is a paper that demonstrates that distributed training across regions via network connection is also feasible.

Distributed training can be achieved through relatively simple DDP+PP parallel strategies (as shown below), where DDP is used between Device 1+2 and 3+4, and PP is used between Device 1 and 2, and 3 and 4.




The author tested the strategy with 8 training nodes spread across the globe (as shown below), and the data transmission speed between training nodes was 100 times slower than that in the data center. After optimizing the scheduling of training nodes, the results were very promising.

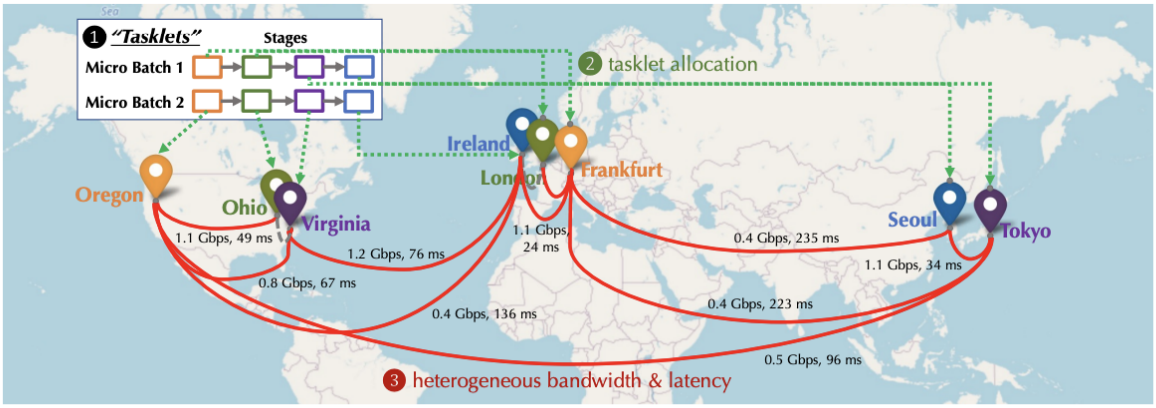



Figure 1: Given ① a set of computation tasklets involved in training foundation models (corresponding to different micro-batches and layers), and ② a heterogeneous network between devices, the goal is to find the optimal ③ allocation of tasklets to devices.

The author trained a smaller version of GPT3 and compared the parallel training optimization in one data center with that of world-wide geo-distributed computing nodes. The result showed that the total time of method 2 was only about 1.8 times longer than that of method 1.

## Conclusion

In our project, we can refer to the following:

1. Avoiding world-wide distributed training can allow computing nodes that are physically close to each other to form teams to perform tasks, thereby reducing communication costs and optimizing computing time.
2. In the paper, the parallel training of method 1 includes more advanced Megatron and Deepspeed, but these parallel strategies have not been explored in world-wide distributed training. If we can reduce physical distance and communication costs, more advanced distributed training strategies can greatly improve training efficiency.
3. Building and maintaining a data center is very expensive, requiring a large amount of start-up capital and high operating costs (such as GPU electricity bills). Therefore, although our strategy still has some gap compared with data centers, it still has great potential.

## Reference

[1] https://huggingface.co/docs/transformers/perf_train_gpu_many

[2] https://arxiv.org/pdf/2206.01288.pdf