Mitigating the Impact of Tail Latency of Storage Systems on Scalable Deep Learning Applications

Hiroki Ohtsuji, Erika Hayashi, Naoto Fukumoto, Eiji Yoshida
Fujitsu Laboratories Ltd.
Kawasaki, Japan

Takuya Okamoto
Fujitsu Ltd.
Kawasaki, Japan

Takeru Kuramoto, Osamu Tatebe
University of Tsukuba
Tsukuba, Japan

I. Introduction

Massive scale deep learning enables HPC systems to finish the training of the large-scale data sets (e.g. ImageNet) in several tens of seconds [1]. Therefore, even a small tail latency of a storage system relatively impacts the total training time. Most of the distributed deep learning applications adopt the data parallel approach, which runs multiple training processes for the different data set on different compute nodes. For each iteration of a training process, all processes exchange values with other processes with all-reduce operations in a synchronous manner. This type of data parallel distributed deep learning application is sensitive to delayed I/O operations. In most cases, training data sets are stored in shared parallel file systems, which usually have tail latency problems. Replacing the only a few fractions of training data set does not have major impacts so that we are developing a method to eliminate the effect of the I/O tail latency by discarding the delayed I/O requests.

II. I/O latency sensitiveness

Fig. 1 depicts the architecture of the target system. Each computing node obtains training data sets from a shared storage system and synchronizes the training model with all-reduce operations in each iteration. If even a single training process is delayed, it blocks the whole progress of the training. Due to this characteristic, synchronous data parallel deep learning applications are sensitive to the I/O tail latency.

III. Avoiding the effect of the tail latency

If the replacement of the training data sets has a minor effect to the precision of the trained model, we can replace the training data sets when the training process is waiting for the I/O request. We are working on the new I/O implementation that monitors I/O requests of deep learning processes and skips the delayed I/O requests. Then, the system gives the alternative training data sets to the training process. We can avoid the effect of the tail latency of the storage system by using this procedure.

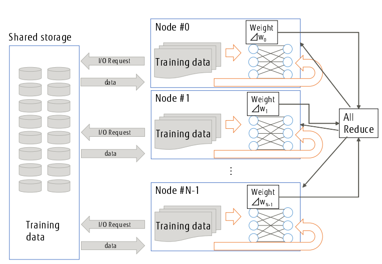

Fig. 1: A functional diagram of a distributed deep learning system

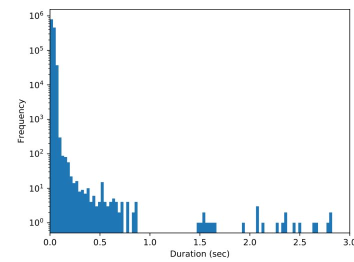

Fig. 2: The histogram depicts the read latency distribution of the shared storage system. X-axis corresponds to the measured latency.

IV. Preliminary Evaluation

A. Distribution of I/O latency of a shared storage system

Fig. 2 depicts the latency distribution of a shared storage system when the application loads the ImageNet data set. The result was measured on Cygnus supercomputer at Univ. of Tsukuba. We observed several delayed (more than 1 sec.) read operations, which occupy less than 0.01 percentage. If we can discard those I/O requests, the longest latency will be shortened by 65 %.

B. Effect of replacing the training data sets

As described in the previous section, if 0.01 % of the training data set can be discarded, we can mitigate the impact of the tail latency of the storage system. [2] proposes a shuffling method for deep learning applications. According to their result, replacing less than 0.01 % of the training data sets does not cause significant precision degradation.

V. Conclusion and future work

Our preliminary evaluation showed that there is an inevitable long tail latency issue and our strategy to replace training data sets when I/O requests stuck on delayed response can cover the effect of the tail latency issue. Future work is implementing the proposed method as a complete set of a deep learning framework and evaluating the actual precision of the trained model and measuring the entire benefit of the method.

References

[1] Masafumi Yamazaki et al. Yet Another Accelerated SGD: ResNet-50 Training on ImageNet in 74.7 seconds. *CoRR*, abs/1903.12650, 2019.

[2] Y. Zhu et al. Entropy-Aware I/O Pipelining for Large-Scale Deep Learning on HPC Systems. In *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 145–156, Sep. 2018.