Poster: A Novel Shared Memory Framework for Distributed Deep Learning in High-Performance Computing Architecture

Shinyoung Ahn
KAIST & ETRI, Korea
syahn@etri.re.kr

Joongheon Kim
Chung-Ang Univ., Korea
joongheon@cau.ac.kr

Sungwon Kang
KAIST, Korea
sungwon.kang@kaist.ac.kr

ABSTRACT

This paper proposes a novel virtual shared memory framework, Soft Memory Box (SMB), which *directly* shares the memory of remote nodes among distributed processes to improve communication performance/speed via deep learning parameter sharing.

CCS CONCEPTS

• **Software and its engineering → Software system structures**; **Distributed systems organizing principles**;

KEYWORDS

Distributed deep learning, remote shared memory, parameter sharing, HPC

ACM Reference Format:
Shinyoung Ahn, Joongheon Kim, and Sungwon Kang. 2018. Poster: A Novel Shared Memory Framework for Distributed Deep Learning in High-Performance Computing Architecture. In *ICSE '18 Companion: 40th International Conference on Software Engineering Companion, May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3183440.3195091

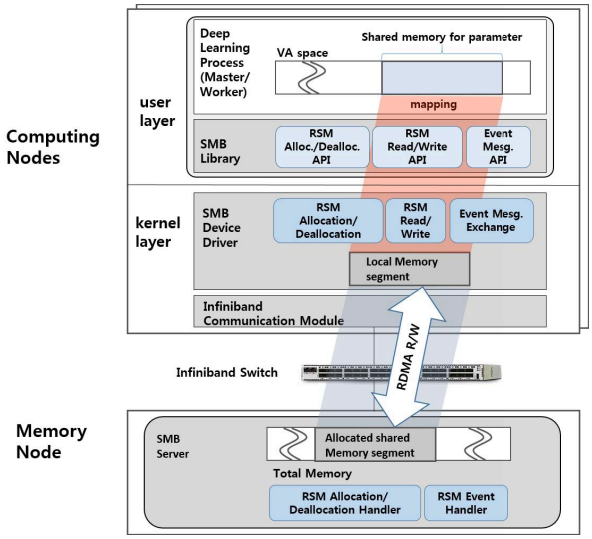

Figure 1: The Architecture of Shared Memory Framemwork

1 INTRODUCTION

In modern artificial intelligence research, deep learning shows excellent performance in various applications such as image recognition, voice recognition, and text mining [2, 3]. Large-scale deep learning requires high-performance computing resources and distributed deep learning platforms. In distributed deep learning platforms, the *workers* which train deep learning model have to share massive parameters, and the sharing introduces communication overheads. With the overheads, the waiting time of the processing units such as CPU/GPU becomes longer, and this eventually introduces the degradation of computation resource usage ratios. Therefore, high speed interconnect networking and distributed parallel programming model is desired for large-scale distributed deep learning.

In this paper, we proposed a distributed deep learning architecture based on the parallel processing and remote shared memory. The proposed architecture utilizes *Remote Direct Memory Access (RDMA)* in Infiniband network. It eliminates the copy operations of communication data between application-level buffers and kernel-level buffers. Eventually, RDMA reduces access time as well as improves bandwidth. Therefore, the proposed architecture enables sharing of deep learning parameters among distributed workers using RDMA for reading/writing data in the memory of remote nodes. In addition, the proposed method enables the distributed workers to share learning parameters among them. This sharing mechanism can be realized by designing a new memory sharing architecture and allowing the access to shared memory to each worker.

The main contributions of our study are as follows: First, we propose a novel shared memory framework, which enables distributed processes share data via remote shared memory buffer. Second, we implement the framework, *Soft Memory Box (SMB)*, which consists of the SMB server, the SMB device driver, the Infiniband communication module, and the SMB library. Third, we evaluate the framework by emulating the asynchronous deep learning parameters sharing with massage-passing interface (MPI) and SMB in high performance computing servers.

2 VIRTUAL SHARED MEMORY FRAMEWORK

The Virtual Shared Memory Framework consists of the SMB server providing shared memory, the SMB library providing application programming interface, the SMB device driver and the kernel-level Infiniband Communication module, as shown in Fig. 1.

The SMB server can provide the physical memory of the memory node by pooling it in advance, and also can provide it on-demand to client processes. In order to enable RDMA from remote processes,

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg,Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3195091

the SMB server should register the virtual-to-physical address mapping information of the provided memory pages into the host channel adapter (HCA). Since this registration time is relatively large, memory pooling is generally preferred. The SMB server includes a remote shared memory (RSM) allocation/deallocation handler to functionally allocate/deallocate shared memory to remote processes, and an RSM event handler function for accumulating data between shared memory segments in the memory node.

The SMB device is a novel virtual device which serves RSM allocation/deallocation functions for handling allocation/deallocation of the remote shared memory, managing allocation information, RDMA read/write functions from/to the allocated RSM, and an event message exchange function which sends/receives event request messages. The SMB device drivers maps the local physical memory segments, which serve as the caches of the remote shared memory segments, to the virtual address of the deep learning processes (see Fig. 1). The SMB device driver communicates with the SMB server through the Infiniband communication module that provides the kernel-level interface for communication with the remote SMB Server by wrapping kernel-level Infiniband verbs. Lastly, SMB library provides API to application processes that use the remote shared memory.

3 IMPLEMENTATION AND EVALUATION

To evaluate SMB, we compare our proposed shared memory framework with MPI as a way of exchanging parameters among deep learning workers. The reason for comparing with MPI is because MPI is the most well-known programming models in parallel processing and used in deep learning platforms [4]. The programs emulating asynchronous stochastic gradient descent (SGD) method consist of the parameter server and several workers which are configured with the star topology.

In order to emulate distributed Deep Neural Network (DNN) training, first of all, we need to measure the real computation time when training actual DNN models in a single GPU. For this purpose, BVLC Caffe (v1.0.0) [8] is used to measure the parameter's memory size and the computation time of an iteration during training of the four convolutional neural network (CNN) models. The Inception_v1 model [7] requires about *51MB* and *189ms* when the mini-batch size is 50. The VGG16 model [5] requires *528MB*, *190ms* when the mini-batch is 64, which is smaller than that of Resnet_50 model [1] with *121MB*, *233ms* when mini-batch is 50. The model with the largest computation time is Inception_Resent_v2 [6] among the 4 models, which requires *214MB* and takes *296ms* when the mini-batch is 6.

The evaluation environment consists of 6 SuperMicro 4028GR-TRT2 servers with 2 socket CPU (Intel Xeon E5-2690 v4, 14cores, 2.3GHz), 128GB DDR4-2400MHz, 4 Nvidia Titan X GPUs and one HP DL380p server with 2 socket CPU (Intel Xeon E5-2690 v4, 14cores, 2.3GHz) and 256GB DDR3-1866MHz memory. Servers are connected to FDR Infiniband switch (56Gbps).

Fig. 2 shows the ratios of communication time in an iteration of DNN training (computation+communication). Fig. 2 also shows that SMB achieves better scalability and efficiency than MPI in terms of the ratios of communication time with the two methods of asynchronous update. SMB is much better than MPI for all models and all the numbers of workers. The communication time of SMB, which exceeds its computation time, was *0ms* for up to 24 workers

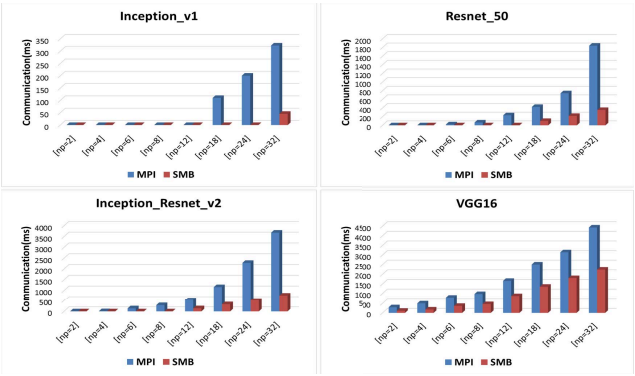

Figure 2: Comparison of communication time ratios of MPI and SMB.

in Inception_v1, which has relatively a small model size, and *45ms* even when extended to 32 workers, which is 7 times better than that of MPI (*323ms*). As the model size increases, the difference in communication time between MPI and SMB decreases. SMB is 5.2 times better than MPI in the case of Resnet_50, 4.8 times better with the Inception_Resnet_v2 model and 1.8 times better with the VGG16 model.

4 CONCLUDING REMARKS

This paper proposes a new software framework, Soft Memory Box (SMB), that accelerates distributed large-scale deep learning processing and its parameter sharing. SMB uses RDMA to transfer the parameters stored in the local machine's memory directly into the memory of the remote shared memory server and exchange the DNN parameters by reading/writing the remote shared memory buffer. As shown in Section 3, it greatly reduces communication overhead due to memory copy and network protocol processing.

ACKNOWLEDGEMENT

This work was supported by ICT R&D program of Ministry of Science and ICT/IITP (No. 2016-0-00087 Development of HPC System for Accelerating Large-scale Deep Learning). J. Kim is a corresponding author of this paper.

REFERENCES

[1] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR'16*.
[2] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. Deep neural networks for acoustic modeling in speech recognition. In *IEEE Signal Process. Mag., 2012.*
[3] Krizhevsky, A., Sutskever, I., and Hinton, G. Imagenet classification with deep convolutional neural networks. In *NIPS'12.*
[4] Shi, S., and Chu, X. Performance modeling and evaluation of distributed deep learning frameworks on GPUs. In *arXiv:1711.05979v2 [cs.DC].*
[5] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR'15.*
[6] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI'17.*
[7] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *CVPR'15.*
[8] Yangqing, J., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *MM'14.*