An Evaluation of the TensorFlow Programming Model for Solving Traditional HPC Problems

Steven Wei Der Chien, Chaitanya Prasad Sishtla, Stefano Markidis, Jun Zhang, Ivy Bo Peng and Erwin Laure

KTH Royal Institute of Technology, Sweden

Computational intensive applications such as pattern recognition, and natural language processing, are increasingly popular on HPC systems. Many of these applications use deep-learning, a branch of machine learning, to determine the weights of artificial neural network nodes by minimizing a loss function. Such applications depend heavily on dense matrix multiplications, also called *tensorial* operations. The use of Graphics Processing Unit (GPU) has considerably speeded up deep-learning computations, leading to a *Renaissance* of the artificial neural network. Recently, the NVIDIA Volta GPU [1] and the Google Tensor Processing Unit (TPU) have been specially designed to support deep-learning workloads. New programming models have also emerged for convenient expression of tensorial operations and deep-learning computational paradigms. An example of such new programming frameworks is TensorFlow, an open-source deep-learning library released by Google in 2015.

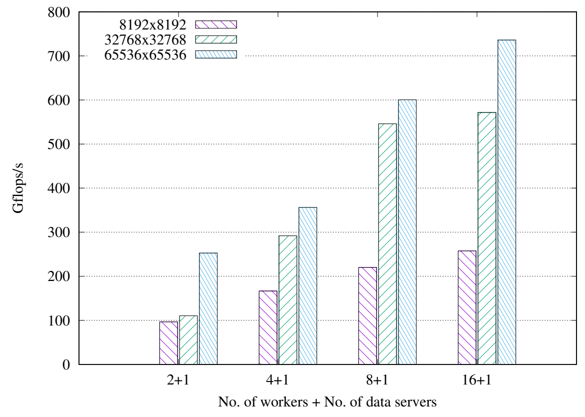

Figure 1. Performance of distributed matrix multiplication.

TensorFlow expresses algorithms as a *computational graph* where nodes represent operations and edges between nodes represent data flow. Multi-dimensional data such as vectors and matrices which flows between operations are called *Tensors.* For this reason, computation problems need to be expressed as a computational graph. In particular, TensorFlow supports distributed computation with flexible assignment of operation and data to devices such as GPU and CPU on different computing nodes. Computation on devices are based on optimized kernels such as MKL, Eigen and cuBLAS. Inter-node communication can be through TCP and RDMA.

This work attempts to evaluate the usability and expressiveness of the TensorFlow programming model for traditional HPC problems. As an illustration, we prototyped a distributed block matrix multiplication for large dense matrices which cannot be co-located on a single device and a Conjugate Gradient (CG) solver. We evaluate the difficulty of expressing traditional HPC algorithms using computational graphs and study the scalability of distributed TensorFlow on accelerated systems. Our preliminary result with distributed matrix multiplication shows that distributed computation on TensorFlow is extremely scalable. This study provides an initial investigation of new emerging programming models for HPC.

The work is funded by the European Commission through the SAGE project (Grant agreement no. 671500).

References

[1] S. Markidis, S.W.D Chien, E. Laure, I.B. Peng, and J.S. Vetter. NVIDIA Tensor Core Programmability, Performance & Precision. *arXiv preprint arXiv:1803.04014*, 2018.