Contribution ID: **18** Type: **not specified**

Exploiting Emerging Multi-core Processors for HPC and Deep Learning using MVAPICH2 MPI Library

Tuesday, September 24, 2019 2:30 PM (30 minutes)

Emerging multi-core architectures such as Intel Xeon are seeing widespread adoption in current and next-generation HPC systems due to their power/performance ratio. Similarly, the recent surge of Deep Learning (DL) models and applications can be attributed to the rise in computational resources, availability of large-scale datasets, and easy to use DL frameworks like Tensorflow, Caffe and PyTorch. However, this increased density of the compute nodes and the performance characteristics of the new architecture bring in a new set of challenges that must be tackled to extract the best performance. In this work, we present some of the advanced designs to tackle such challenges in the MVAPICH2 MPI library on the latest generation HPC systems using Intel multi-core processors.

From the HPC angle, we will focus on the following aspects — a) how can we achieve fast and scalable startup on large HPC clusters with Omni-Path and InfiniBand, b) contention-aware, kernel-assisted designs for large-message intra-node collectives, c) designs for scalable reduction operations on different message sizes, and d) shared-address space-based scalable communication primitives. We also compare the proposed designs against other state-of-the-art MPI libraries such as Intel MPI and OpenMPI. Experimental evaluations show that the proposed designs offer significant improvements in terms of time to launch large-scale jobs, the performance of intra-node and inter-node collectives, and performance of applications.

From the DL angle, we will focus on efficient and scalable CPU-based DNN training. We will provide an in-depth performance characterization of state-of-the-art DNNs like ResNet(s) and Inception-v3/v4 on three different variants of the Intel Xeon Scalable (Skylake) processor. We provide three key insights based on our study: 1) Message Passing Interface (MPI) should be used for both single-node and multi-node training as it offers better performance, 2) TensorFlow's single-process training is under-optimized to fully utilize all CPU cores even with advanced Intel MKL primitives and the Intel-optimized TensorFlow runtime, and 3) Overall performance depends on various features like the number of cores, the process per node (PPN) configuration, hyper-threading and DNN specifications like inherent parallelism between layers (inter-op parallelism) and the type of DNN (ResNet vs. Inception). We also provide an in-depth performance evaluation. The results show that using four MPI processes using Horovod for training the same DNN and same effective batch size is up to 1.47x faster than a single process (SP) approach. Using this 4ppn configuration, we achieve up to 125x speedup (compared to a single node) for training ResNet-50 on 128 Skylake nodes using MVAPICH2 2.3 MPI library.

Presenter: Prof. PANDA, Dhabalaeswar (The Ohio University)

Session Classification: Session 2: Libraries and Tools