InfiniTensor: A Tensor-Friendly, Efficient Parallel Programming Library for Accelerator-Centric Clusters

March 12, 2024

Abstract

Rising AI-centric workloads, such as AI4Science, large language models, and large multimodal models, are increasingly deployed on accelerator-centric clusters. These clusters utilize parallel accelerators (e.g., GPUs and TPUs) and feature heterogeneous memory devices (e.g., SRAM, HBM, and DRAM) connected by fast network links (e.g., NVLink and InfiniBand) to deliver substantial computing, memory, and networking resources for tensor operations.

For optimal performance, it's essential to exploit accelerators for both AI model training/inference and data processing tasks, such as preprocessing, clustering, and cleaning. While AI models benefit from distributed training libraries like Megatron-LM and AI compilers like XLA when utilizing parallel accelerators, data processing tasks often rely on CPUs solely, suffering from bottlenecks when processing and communicating data.

AI programmers seek a parallel programming library that offers a tensor-friendly interface for managing complex data workflows and automatically utilizes the compute, memory, and network resources on parallel accelerators. Current solutions are inadequate; high-level libraries like Ray require significant code rewrites to comply with the actor-centric message-passing interface, and users must manually distribute tensors when they go beyond a single accelerator. Meanwhile, low-level libraries like NCCL and NVSHMEM demand learning complicated programming paradigms (e.g., collective communication and asynchronous programming), making the porting of existing data tasks to accelerators prohibitively expensive.

In this talk, we will explore the concept of Partitioned Global Address Space (PGAS), a tensor-friendly programming abstraction originally proposed for scientific computing, but not yet applied to accelerator clusters. A primary reason is its lack of mechanisms for automatically inferring computational dependencies among tensors. This gap hinders efficient data prefetching, caching, and parallelism mechanisms from being effectively realized, critical for utilizing accelerator clusters.

To bridge this gap, we've designed InfiniTensor, a tensor-friendly and efficient parallel programming library. InfiniTensor can automatically analyze tensor dependency in PGAS programs, and it leverages analysis results to facilitate effective data prefetching and caching. Our early experiments shows that InfiniTensor can facilitate AI programers to transition complex data processing tasks (i.e., all clustering operations supported in the widely used scikit-learn library) from CPUs to accelerators, achieving high performance with minimal programming effort.