End-to-End Computing using Functional Partitioning: A Community Earth System Model (CESM) Case Study

Karan Sapra, Saurabh Gupta, Ross Miller, Valentine Anantharaj, Scott Atchley, Sudharshan S. Vazhkudai,
Devesh Tiwari, Melissa C. Smith

With the advent of accelerator based supercomputing nodes, the CPU resources on these compute nodes can often be underutilized due to the mismatch in the processing abilities of the CPUs. With the advent of accelerator based supercomputing nodes, the CPU resources on these compute nodes can often be underutilized due to the mismatch in the processing abilities of the CPUs and the GPUs. For sustained Exascale computing, however, efficient utilization of resources is the key. Many scientific applications utilize the accelerators (e.g. GPUs) while CPUs only act as a master leaving a large fraction of free compute cycles on the CPUs. Furthermore, many jobs cannot utilize all of the available cores on a node due to limitations such as memory capacity and I/O bandwidth. For example, the Titan machine at Oak Ridge National Lab consists of 18,688 compute nodes, each of which is equipped with 16 CPU cores and 1 GPU, with the GPUs providing the bulk of the FLOPs (27 petaflop peak). Thus, there is an opportunity to exploit the underutilized CPU cores towards an application's own end-to-end activities.

We have developed the Functional Partitioning (FP) runtime framework, where applications can exploit underutilized computing resources for post-processing tasks such as data analytics, data validation, visualization preprocessing, and also towards running the computational simulation itself reliably and at scale. These tasks are an essential part of the scientific workflow and typically performed on smaller-scale clusters and in an out-of-band fashion, incurring costs such redundant data movement and delayed end-to-end processing. With only a few FP library calls added to the application, our framework allows scientists to overlap the post-processing tasks with the main application in a pipelined fashion. Further, the aforementioned tasks are performed on the same compute node where the data is generated. We demonstrate the usability of our approach by integrating the CESM climate application with an essential post-processing task. We believe that this approach of making the compute node itself a self-contained entity brings a fresh perspective to in-situ processing and end-to-end workflow processing. Due to improved data locality and higher scientific productivity in this approach, this is an essential step towards the path to Summit and Exascale computing.