Towards Performance-Portability of the Albany/FELIX Land-Ice Solver to New and Emerging Architectures Using Kokkos

Irina Tezaur, Jerry Watkins
Sandia National Laboratories
ikalash@sandia.gov, jwatkin@sandia.gov

Irina Demeshko
Los Alamos National Laboratory
irina@lanl.gov

Abstract

As high performance computing (HPC) architectures become more heterogeneous, climate codes must adapt to take advantage of potential performance capabilities. This talk will focus on performance-portability of the Sandia Albany/FELIX finite element land-ice solver to new and emerging architecture machines. The computational time for an ice sheet simulation in FELIX is divided into two pieces, each comprising approximately 50 percent of the total run time: finite element assembly (FEA), and linear solves. In the context of two climate applications implemented within Albany, the FELIX land-ice solver and also the Aeras global atmosphere dycore, we will discuss our efforts (Demeshko et al, 2017) in transitioning the FEA in Albany from an MPI-only to an MPI+X programming model via the Kokkos library (Edwards et al, 2014) and programming model. In this model, MPI is used for internode parallelism and X denotes a shared-memory programming model for intranode parallelism (e.g., X=OpenMP, CUDA). With Kokkos data layout abstractions, the same code can run correctly and efficiently on current and future HPC hardware with different memory models. We will describe some key performance developments in the finite element assembly process within the FELIX land-ice model, as well as future performance goals in strong and weak scalability on a variety of different architectures including NVIDIA GPUs and Intel Xeon Phis. A perspective towards performance portability of the Trilinos-based linear solvers utilized by FELIX will also be provided.

References

1. I. Demeshko and J. Watkins and I. Tezaur and O. Guba and W. Spotz and A. Salinger and R. Pawlowski and M. Heroux.. Towards performance-portability of the Albany finite element analysis code using the Kokkos library. J. HPC Appl. (2017).
2. H. Edwards and C. Trott and D. Sunderland. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. J. Parallel and Distributed Computing, 74(12) 3202-3216, 2014.