A Performance-Portable C++ Implementation of Atmospheric Dynamics

Andy Salinger, Luca Bertagna, Andrew Bradley, Michael Deakin, Oksana Guba, Daniel Sunderland, Irina Tezaur
Sandia National Laboratories
agsalin@sandia.gov, lbertag@sandia.gov, ambradl@sandia.gov, mdeakin@sandia.gov, onguba@sandia.gov, dsunder@sandia.gov, ikalash@sandia.gov

Abstract

In this work, we reimplement the atmospheric dynamical core (HOMME) of Energy Exascale Earth System Model (E3SM) so that a single implementation achieves performance on a variety of HPC architectures. The dynamical core combines an unstructured-grid spectral element method on the surface of the sphere with a finite difference discretization for vertical columns and an explicit time integration algorithm to simulate the flow and transport in the Earth's atmosphere. Atmospheric dynamical cores are among the most expensive parts of a climate model, so the new implementation must be of similar – or improved – performance to the original version on all production architectures for its adoption to be considered. They are also highly parallelizable, so the new implementation should also strong-scale well. Finally, next generation HPC architectures are heterogeneous, and the design of subsequent HPC architectures is still uncertain, so the code base must be designed with principles from computer architecture to achieve reasonable performance, and must be flexible enough to support tuning small, performance critical kernels to specific targets. To achieve these goals, the new implementation uses Kokkos, a C++ programming library for on-node parallelism designed to be a programming model for performance portability across heterogeneous architectures.

We will present our latest results for the performance and portability of our new version of the code on a variety of HPC architectures. With Kokkos, the new implementation achieves performance comparable to the original Fortran implementation, strong-scales better, and gives reasonable performance on next generation systems (such as Xeon Phi and CUDA) without specifically tuning for them. The new code base is better positioned for achieving performance on upcoming architectures as well. In addition, in refactoring of the code from Fortran into C++, the new implementation benefits from modern software development infrastructure, including a larger and safer library of data-structures and algorithms, intrinsically safe design patterns, and more features specifically for improving performance such as explicit vectorization through intrinsics and compile time computations.

References

1. John M. Dennis and Jim Edwards and Katherine J. Evans and Oksana Guba and Peter H. Lauritzen and Arthur A. Mirin and Amik St-Cyr and Mark A. Taylor and Patrick H. Worley. CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model . https://doi.org/10.1177/1094342011428142.