SAND2015-10294C

Component-Based Application Code Development, Part 2: Demonstration on a Land-Ice Model and Proposed Extension to Other Climate Components
November 23, 2015

Irina Tezaur
Sandia National Laboratories
Andrew Salinger, Mauro Perego, Ray Tuminaro (Sandia National Laboratories)

Abstract

This paper illustrates the success of the components-based code development strategy for developing a next-generation world-class climate code in the specific case of the *Albany/FELIX* land-ice solver created as a part of the PISCEES SciDAC3 project. The proposed idea is to find opportunities to apply this approach in other climate areas (e.g., atmosphere, sea-ice, ocean) to make these models more scalable, robust and portable to emerging architectures, to endow these models with improved analysis capabilities (e.g., adjoint-based optimization, embedded uncertainty quantification), all towards a more integrated climate modeling framework that shares software, data standards and tools, and model components. This submission is a companion paper to a submission by A. Salinger on the Agile Components Strategy and *Albany* code.

Background/Research to Date

According to a 2012 report by the National Research Council [1], there is a critical need for a next generation of advanced climate models. More specifically, the report calls for climate models to take a more integrated path and use a common software infrastructure while adding regional detail, new simulation capabilities, and new approaches for collaborating with their user community. Although climate models have improved in recent years, much work is needed to make these models reliable and efficient on continental scales, to quantify uncertainties in the models' outputs, and to port the models to a new generation of high performance computer (HPC) architectures. Many legacy climate models lack advanced analysis capabilities, like sensitivity and adjoint calculations. Moreover, legacy codes need to be rewritten substantially in order to run accurately and efficiently on new architecture machines (e.g., GPUs), as they are based on algorithms optimized for existing architectures (e.g., CPUs).

A promising approach for developing next-generation performance-portable solvers with advanced analysis capabilities is the so-called components-based code development strategy[1] to building application codes. In this approach, mature, modular libraries are combined using abstract interfaces and template-based generic programming, resulting in a final code that is verified, scalable, fast and robust, and has access to dozens of algorithmic and advanced analysis capabilities. One recent success story for the components-based code development approach in the area of climate modeling is *Albany/FELIX*[2] [4, 5], an unstructured grid (Fig. 1, left) finite element land-ice solver written using the *Trilinos* [3] libraries and the *Albany* [2] code base as a part of the SciDAC3 PISCEES[3] project for integration into earth system models (ESMs). The component-based code development approach has led to the rapid development (≈3 FTEs) of this world-class land-ice model with many sophisticated capabilities. The integration of automatic differentiation into the code has enabled robust nonlinear solves, sensitivity analysis,

[1] Detailed in a companion idea paper submitted by A. Salinger entitled "Component-Based Application Code Development, Part 1: The Agile Components Strategy and Albany Code".
[2] Finite Elements for Land Ice eXperiments.
[3] Predicting Ice Sheet and Climate Evolution at Extreme Scales.

adjoint-based optimization for ice sheet initialization, and uncertainty quantification (UQ). Ad hoc spin-ups and parameter tuning has been replaced with the use of formal, adjoint-based optimization techniques for generating realistic model initial conditions [6]. In collaboration with experts from the QUEST SciDAC institute, a framework for for forward and inverse UQ that incorporates formal UQ methods and computational tools that orchestrate these methods (e.g., *DAKOTA* [7]) has been developed. The *Albany/FELIX* code has demonstrated scalability up to 1 billion unknowns and tens of thousands of cores thanks parallel scalable iterative linear solvers and newly-developed preconditioning methods from *Trilinos* (Fig. 1, right, top). The development of new semi-implicit momentum balance and thickness coupling techniques has led to more stable and efficient time-stepping schemes, expected to reduce substantially run-times for transient land-ice simulations. Finally, using the *Kokkos Trilinos* library and programming model, we were able to create a single code that runs on different architectures (multi-core, many-core, GPU) by merely changing a template parameter (Fig. 1, right, bottom).

Proposed Direction of Work

The proposed direction is to look for ways to equip other climate components (e.g., atmosphere, sea-ice, ocean) and coupled ESMs with the advanced analysis and performance capabilities described above by integrating into these models software libraries and algorithms developed by domain experts. Success of this approach rests strongly on a collaboration model for the development of climate technologies which involves not only glaciologists and climate modelers, but also computational scientists, and is geared towards creating a unified modeling framework that shares software, data standards and tools, and model components. Based on our experience with the *Albany/FELIX* code, the following enhancements in other climate models are conceivable: (1) improved software quality through formal verification studies and regression testing, (2) improved scalability and robustness, (3) improved fidelity (e.g., through the use of unstructured, regionally refined meshes), (4) performance-portability to new and emerging architectures, (5) improved incorporation of data (e.g., through better, optimization-based model initiation techniques), (6) improved validation and UQ methods (e.g., embedded UQ), (7) improved time-evolution algorithms for more stable and faster transient simulations. Some specific ideas worth exploring are embedded UQ for atmosphere, non-linear solvers for sea-ice and implicit/semi-implicit solvers for ocean.

Connections to Math, Comp Sci & and Climate Science

For a successful integration of software libraries with advanced analysis and next-generation capabilities into climate components, close collaborations between climate scientists and mathematicians/omputational scientists (e.g, the current applied math and computer science SciDAC3 institutes: FastMath, SUPER, QUEST) are required. The proposed components-based approach would facilitate the development and numerical study of new mathematical approaches and computational algorithms. The use of similar software frameworks and libraries would make state-of-the-art methods readily available to a variety of climate applications.

Potential Impact on the Field

We believe the proposed approach would lead to better climate models, namely models that are more robust and efficient, equipped with advanced analysis capabilities, and capable of running on next generation platforms built towards exascale computing, as described in detail above.

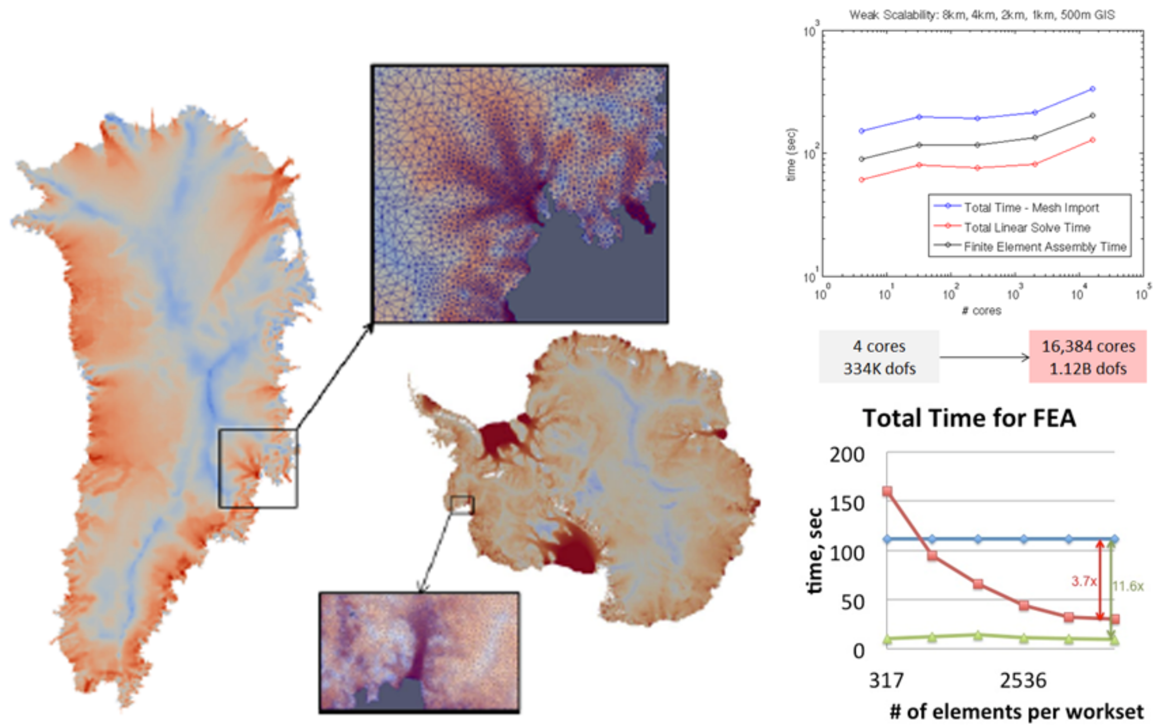

Figure 1: Greenland and Antarctica geometries, discretized by unstructured meshes ((left); weak scaling up to 1.12B unknowns and 16K cores (right, top); performance portability of finite element assembly using *Kokkos*: Serial - blue, OpenMP - green, CUDA - red (right, bottom)

References

[1] National Research Council of the National Academies, A national strategy for advancing climate modeling. The National Academies Press, Washington, DC, 2012.

[2] A. Salinger. Albany: A component-based partial differential equation code built on Trilinos. Technical report, Sandia National Labs, 2003.

[3] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, K. Long, R. Lehoucq, R. Pawlowski, E. Phipps, H. Thornquist, A. Salinger, R. Tuminaro, J. Willenbring, and A. Williams. An overview of trilinos. Technical Report SAND2003-2927, Sandia National Laboratories, 2003.

[4] I. Tezaur, M. Perego, A. Salinger, R. Tuminaro, and S. Price. Albany/FELIX: A parallel, scalable and robust finite element higher-order stokes ice sheet solver built for advanced analysis. *Geosci. Model. Develop.*, **8**:1–24, 2015.

[5] I. Tezaur, R. Tuminaro, M. Perego, A. Salinger, S. Price. On the scalability of the Albany/FELIX first-order Stokes approximation ice sheet solver for large-scale simulations of the Greenland and Antarctic ice sheets, *Procedia Computer Science*, **51**:2026–2035, 2015.

[6] M. Perego, S. Price, G. Stadler. Optimal initial conditions for coupling ice sheet models to Earth system models, *J. Geophys. Res.* **119**(9) 1894–1917.

[7] B. Adams, L. Bauman, W. Bohnhoff, K. Dalbey, M. Ebeida, J. Eddy, M. Eldred, P. Hough, K. Hu, J. Jakeman, L. Swiler, and D. Vigil. DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.4 User's Manual, Sandia Technical Report SAND2010-2183, 2009.