SAND2015-1637C

Towards Exascale Implementation of the Finite Element Based Application Development Environment

Sandia National Laboratories
Irina Demeshko, H. Carter Edwards, Michael A. Heroux, Eric T. Phipps, Andrew G. Salinger , Roger P. Pawowski
Sandia National Laboratories, New Mexico, 87185

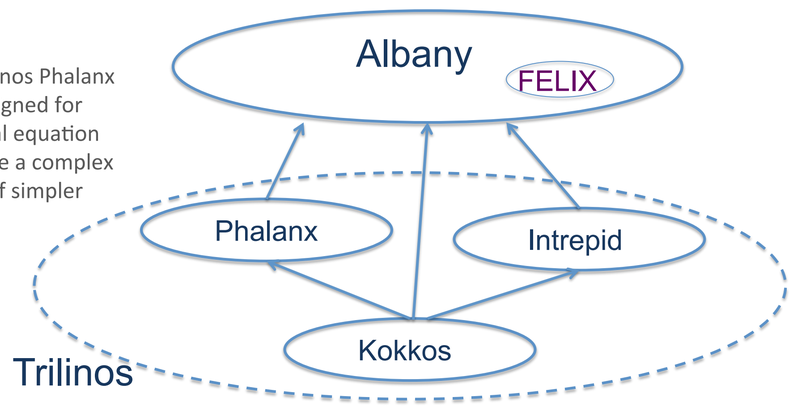
Introduction

We believe that multicore-CPUs and manycore-accelerator devices, that promise improvements in both runtime performance and energy consumption, are major candidates to be used in the future HPC systems.
Nowadays, the Finite Element Method is actively used in diverse variety of scientific and industry simulations, that require High Performance Computing (HPC) technology. This poster presents our strategy for adapting Albany (Finite Element Based architecture development environment) code for future architecture, based on using a new Phalanx-Kokkos local field evaluation kernel from Trilinos[1].

Albany

Albany[2] – a finite element based application development environment containing the "typical" building blocks, called "Albany evaluators", needed for rapid deployment and prototyping of analysis capabilities.

The code is based on Trilinos Phalanx package, specifically designed for general partial differential equation solvers, which decompose a complex problem into a number of simpler problems with managed dependencies.



Kokkos programming model

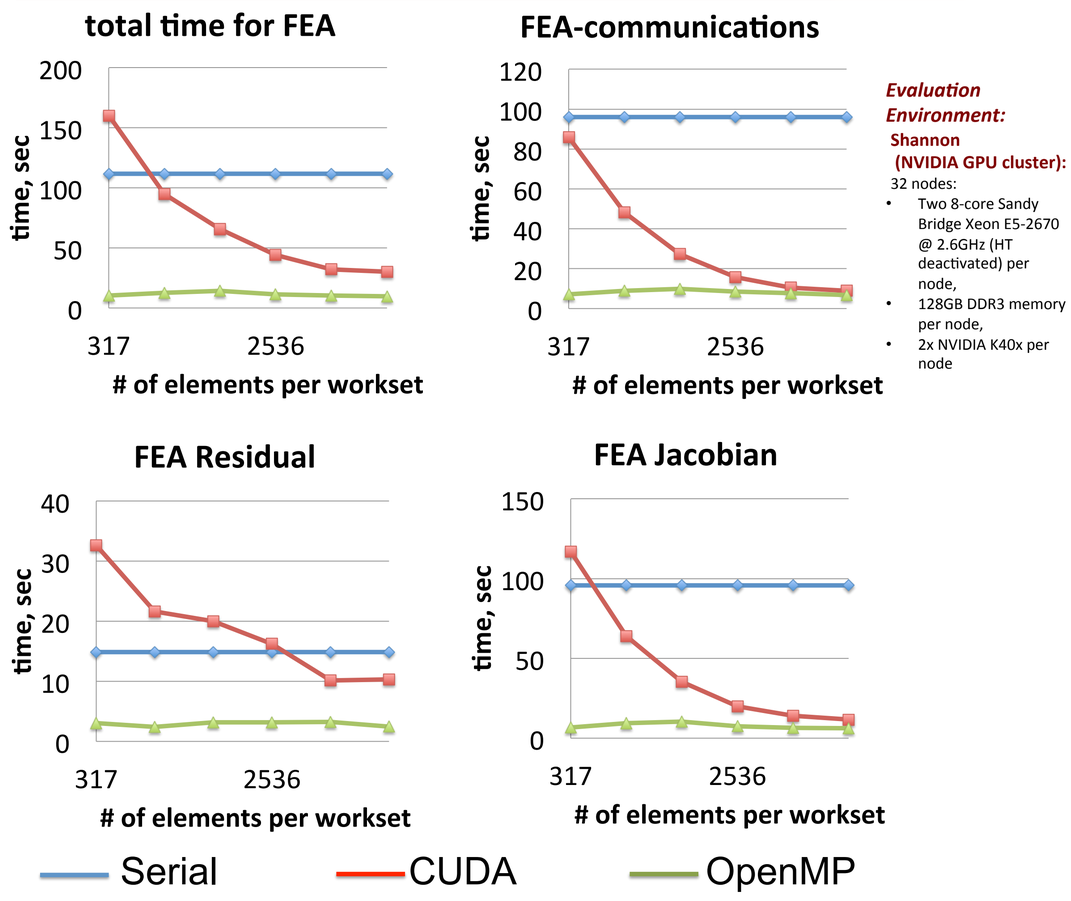
Kokkos [3] - C++ library from Trilinos [2] to provide scientific and engineering codes with an intuitive manycore performance portable **programming model**.

- Provides portability across manycore devices (Multicore CPU, NVidia GPU, Intel Xeon Phi (potential: AMD Fusion))
- Abstract data layout for non-trivial data structures
- Uses library approach:
  - *Maximize amount of user (application/library) code that can be compiled without modification and run on these architectures*
  - *Minimize amount of architecture-specific knowledge that a user is required to have*
- Performant: Portable user code performs as well as architecture-specific code

Main abstractions:

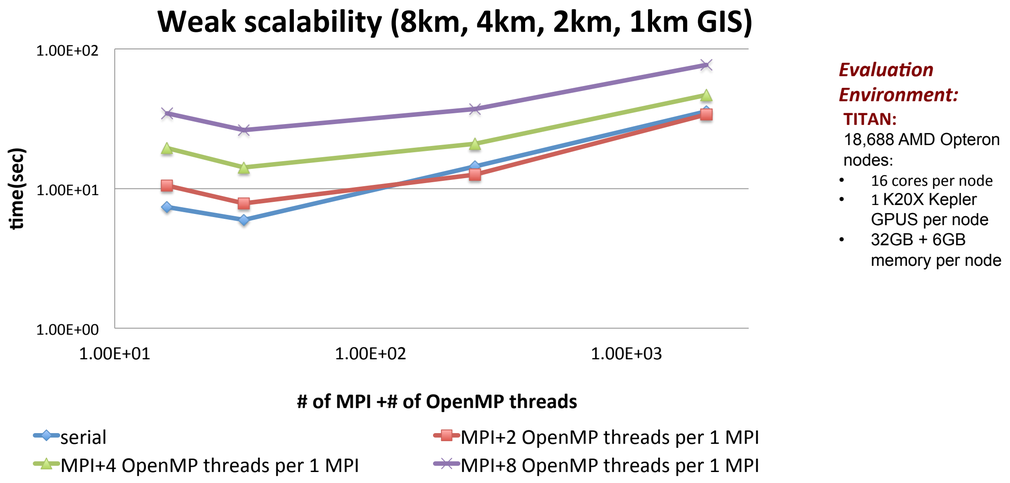
- Kokkos executes computational kernels in fine-grain data parallel within an *Execution space.*
- Computational kernels operate on multidimensional arrays (*Kokkos::View*) residing in *Memory spaces.*
- Kokkos provides these multidimensional arrays with polymorphic data layout, similar to the Boost. MultiArray flexible storage ordering.

Performance Results



Evaluation Environment:
Shannon (NVIDIA GPU cluster):
32 nodes:
- Two 8-core Sandy Bridge Xeon E5-2670 @ 2.6GHz (HT deactivated) per node,
- 128GB DDR3 memory per node,
- 2x NVIDIA K40x per node

Figure 1. On-node performance: weak scalability graph, 20km



Evaluation Environment:
TITAN:
18,688 AMD Opteron nodes:
- 16 cores per node
- 1 K20X Kepler GPUS per node
- 32GB + 6GB memory per node

Figure 2. Weak scalability graph for MPI+OpenMP

Ice Sheet Simulation Code (FELIX)

We are developing a performance portable implementation of an Ice Sheet simulation code*, build with the Albany application development environment

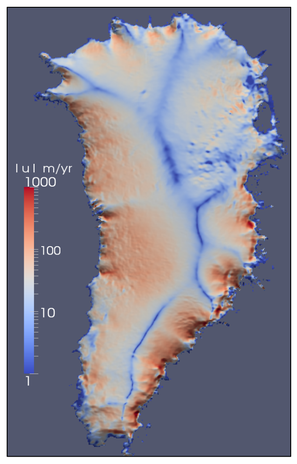
Project objectives:

- Develop a robust, scalable, unstructured-grid finite element code for land Ice dynamics.
- Provide sea level rise prediction
- Run on new architecture machines (hybrid systems).

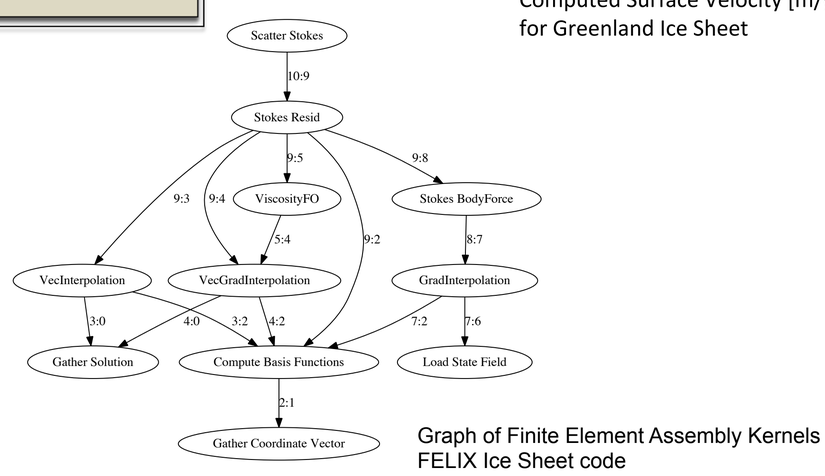
Nonlinear Stokes' Model for Ice Sheet Stresses

$$-\nabla\cdot(2\mu\dot{\epsilon}_1) = -\rho g\frac{\partial s}{\partial x}$$
$$-\nabla\cdot(2\mu\dot{\epsilon}_2) = -\rho g\frac{\partial s}{\partial y}$$

Fully Implicit Solution Algorithm:
- Finite Element Assembly
  - ~50% CPU time
- Linear Solve ILU + CG
  - ~50% CPU Time
- Problem Size, up to:
  - 1.12B Unknowns
  - 16384 Cores on Hopper



Computed Surface Velocity [m/yr] for Greenland Ice Sheet



Graph of Finite Element Assembly Kernels FELIX Ice Sheet code

***Funding Source**: PISCEES* SciDAC Application (BER & ASCR)

Authors: A. Salinger, I. Kalashnikova, M. Perego, R. Tuminaro [SNL], S. Price[LANL]

References:
1. Willenbring, James M.,and Michael Allen Heroux. *Trilinos Users Guide.*, August 01, 2003.
2. Salinger, Andrew G. *Component-based Scientific application Development.*, December 01, 2012
3. Edwards, H. Carter, Trott, Christian R., Sunderland, Daniel *Kokkos: Enabling manycore performance portability through polymorphic memory access patterns.* Journal of Parallel and Distributed Computing, 2014.

National Nuclear Security Administration
U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

SAND

Sandia National Laboratories