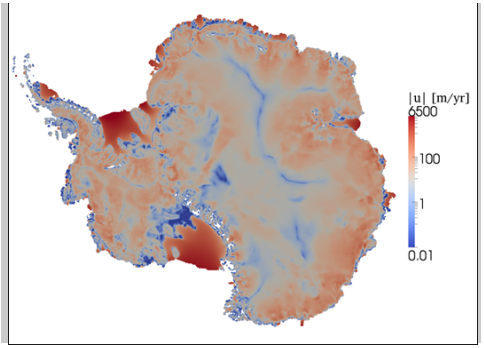
SAND2015-2186C

Exceptional service in the national interest

Sandia National Laboratories



A Kokkos Implementation of Albany: A Performance Portable Multiphysics Simulation Code

Irina Demeshko, , Andrew M. Bradley, Eric C. Cyr,H. Carter Edwards,
Michael A.Heroux, Eric T. Phipps, Andrew G. Salinger

SIAM CSE, 2015

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

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 NO. 2011-XXXXP

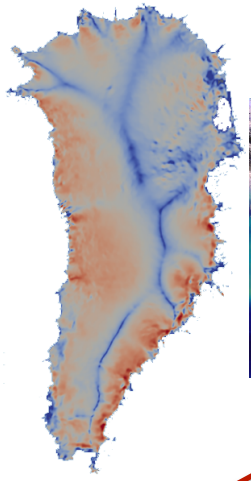
Outline

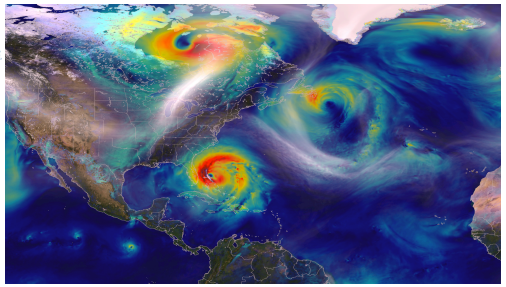
Sandia National Laboratories

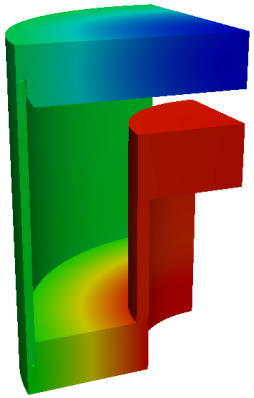
- Introduction
- Kokkos
- Albany
- Albany Greenland Ice Sheet Model (FELIX)
- Albany ThermoMechanics Application
- Performance results
- Conclusion

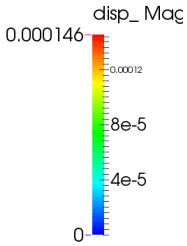
Introduction

Sandia National Laboratories









Tianhe-2 (National SuperComputer Center in Guangzhou)
Intel Xeon E5-2692+ Intel Xeon Phi

TITAN (ORNL)
Cray XK7 , Opteron 6274 NVIDIA K20x

Sequoia (DOE/NNSA/LLNL)
BlueGene/Q, Power BQC, Custom

Sandia National Laboratories

Performance portability has become a critical issue

parallel code needs to be executed correctly and performant despite variation in the architecture, operating system and software libraries.

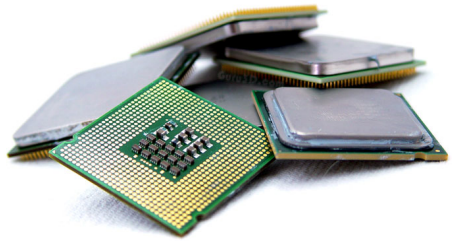
Approach: Kokkos programming model

C++ library, which provide performance portability across diverse devices with different memory models.


Kokkos

Kokkos* programming model

Sandia National Laboratories



- Kokkos - C++ library to provide scientific and engineering codes with an intuitive manycore performance portable **programming model**.

- ✓ **Standard C++, Not a language extension**
- ✓ **Uses C++ template meta-programming**
- ✓ **Provides portability across manycore devices (Multicore CPU, NVidia GPU, Intel Xeon Phi (potential: AMD Fusion))**
- ✓ **Abstract data layout for non-trivial data structures**

* Trilinos package

Kokkos* programming model

Sandia National Laboratories

- **Kokkos :**

✓ **Single code base**

✓ **Support most of the current (and future) hardware**

✓ **Flexible run configurations MPI-Only**

- ✧ **MPI + Threads**
- ✧ **MPI + GPU**
- ✧ **MPI + GPU + Threads**

✓ **Close to optimal performance (i.e. performance of a specialized code)**

✓ **Use vendor compilers**

✓ **Simple code**

* Trilinos package

Kokkos* programming model

Sandia National Laboratories

A programming model with two major components:

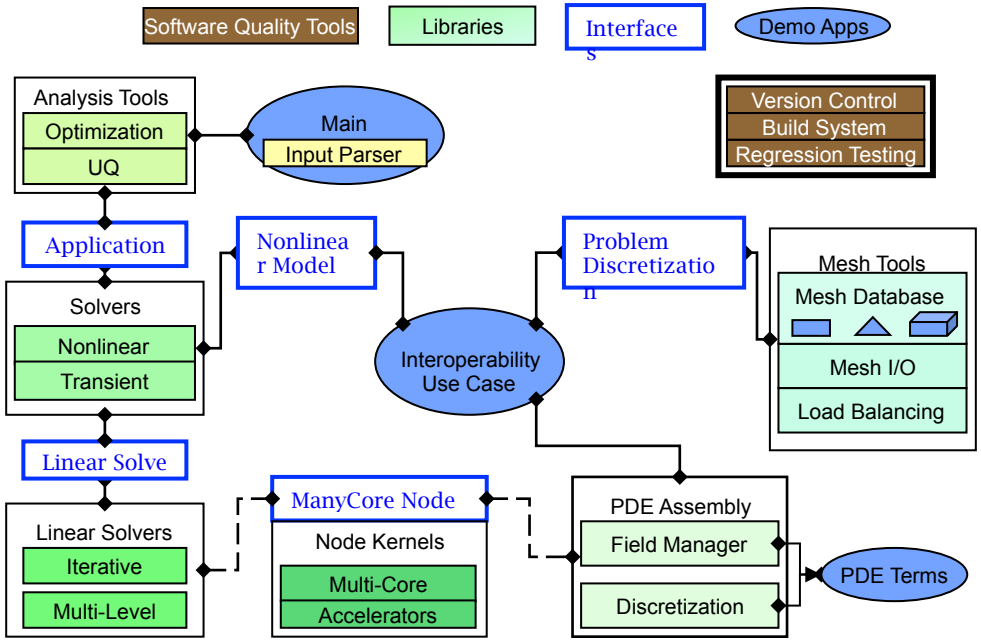
- **Data access abstraction**
  - ✓ Change data layout at compile time without changing access syntax => Optimal access pattern for each device
  - ✓ Data padding and alignment is transparent
    Access traits for portable support of hardware specific load/store units
- **Parallel dispatch**
  - ✓ Express algorithms with **parallel_for**, **parallel_reduce** etc.
    Using functor concept (functor: construct allowing an object to be invoked or called as if it were an ordinary function)
  - ✓ Transparently mapped onto back-end languages
    (e.g. OpenMP, CUDA, Pthreads …)

Sandia National Laboratories

Albany : agile component-based parallel unstructured mesh application

- A finite element based application development environment containing the "typical" building blocks needed for rapid deployment and prototyping of analysis capabilities
- A mechanism to drive and demonstrate the _AgileComponents_ rapid software development vision and the use of template-based generic programming (TBGP) for the construction of advanced analysis tools
- A Trilinos demonstration application, built almost exclusively from reusable libraries. Albany leverages 100+ packages/libraries.
- Open-source

Albany Structure:



Albany : agile component-based parallel unstructured mesh application

Sandia National Laboratories

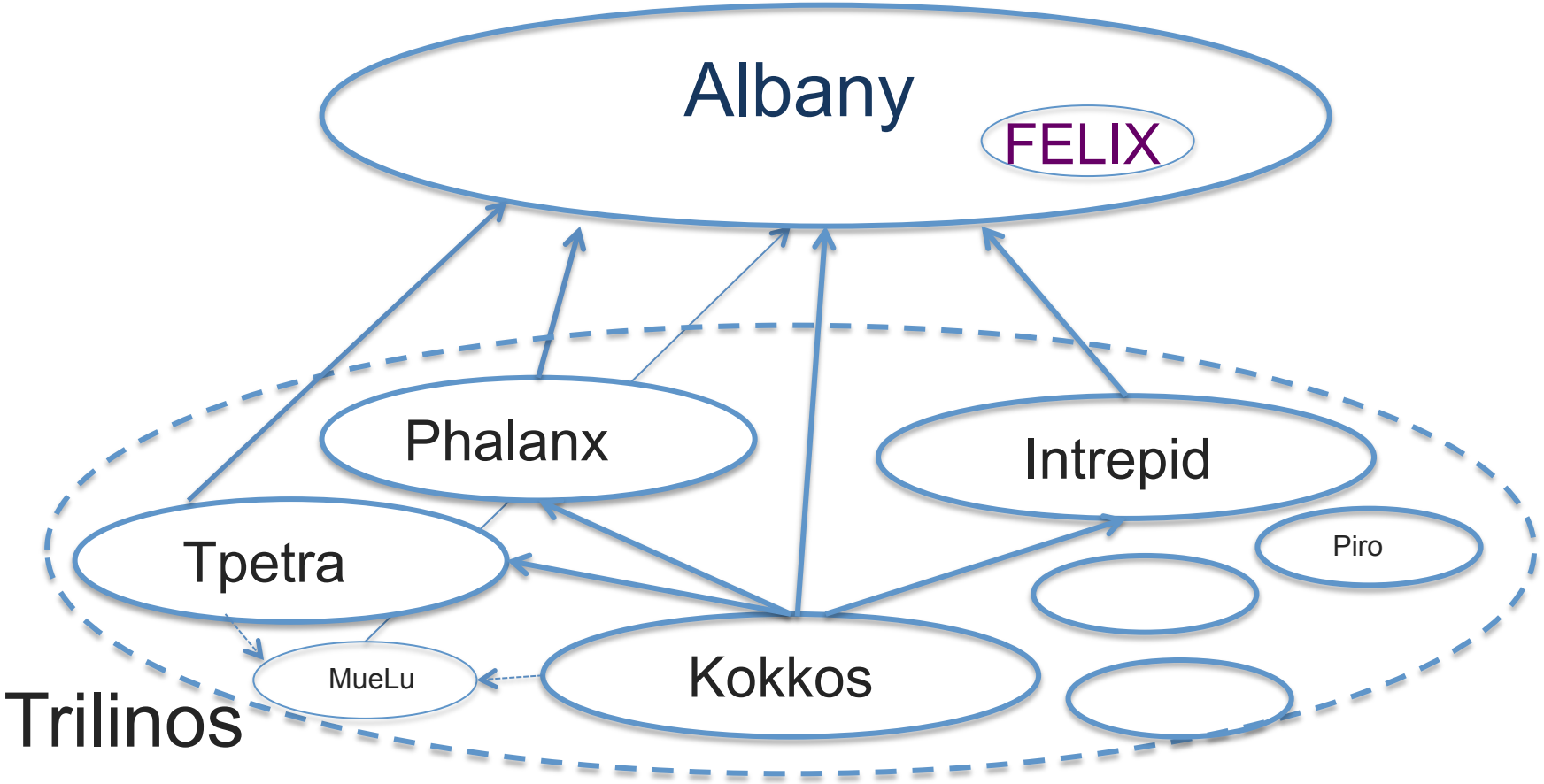
Agile component design provides

- Efficient in-memory integration of external mesh database
- Abstract interfaces to components (mesh, solvers, assembly, analysis tools)
- Requirements imposed on external tools through generic interfaces

Fully representative of a typical advanced implicit, unstructured finite element application

- Trilinos supplies components: modern linear & nonlinear solvers, preconditioning strategies, continuation tools, ...
- Genericism of physics evaluation, residual based, with Sacado Jacobian and matrix free options
- Parallel MPI(+X)
- Large problems, representative boundary conditions, test suite 230+ problems
- Embedded SA & UQ
- Large problems, representative boundary conditions
- Open source

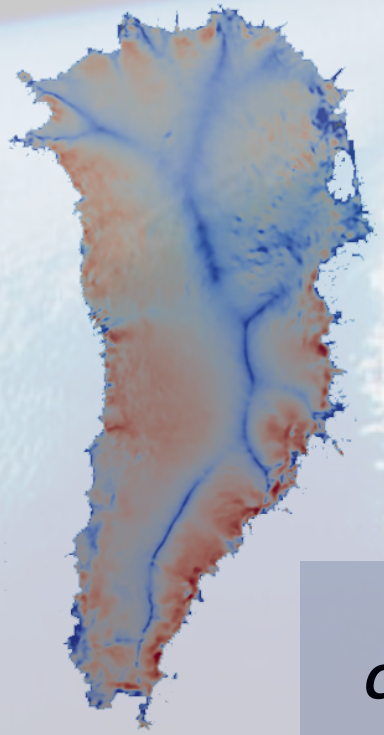
Albany to Kokkos refactoring



Albany Greenland Ice Sheet model

Albany Greenland Ice Sheet Model (FELIX project)

Sandia National Laboratories



- An unstructured-grid finite element ice sheet code for land-ice modeling.

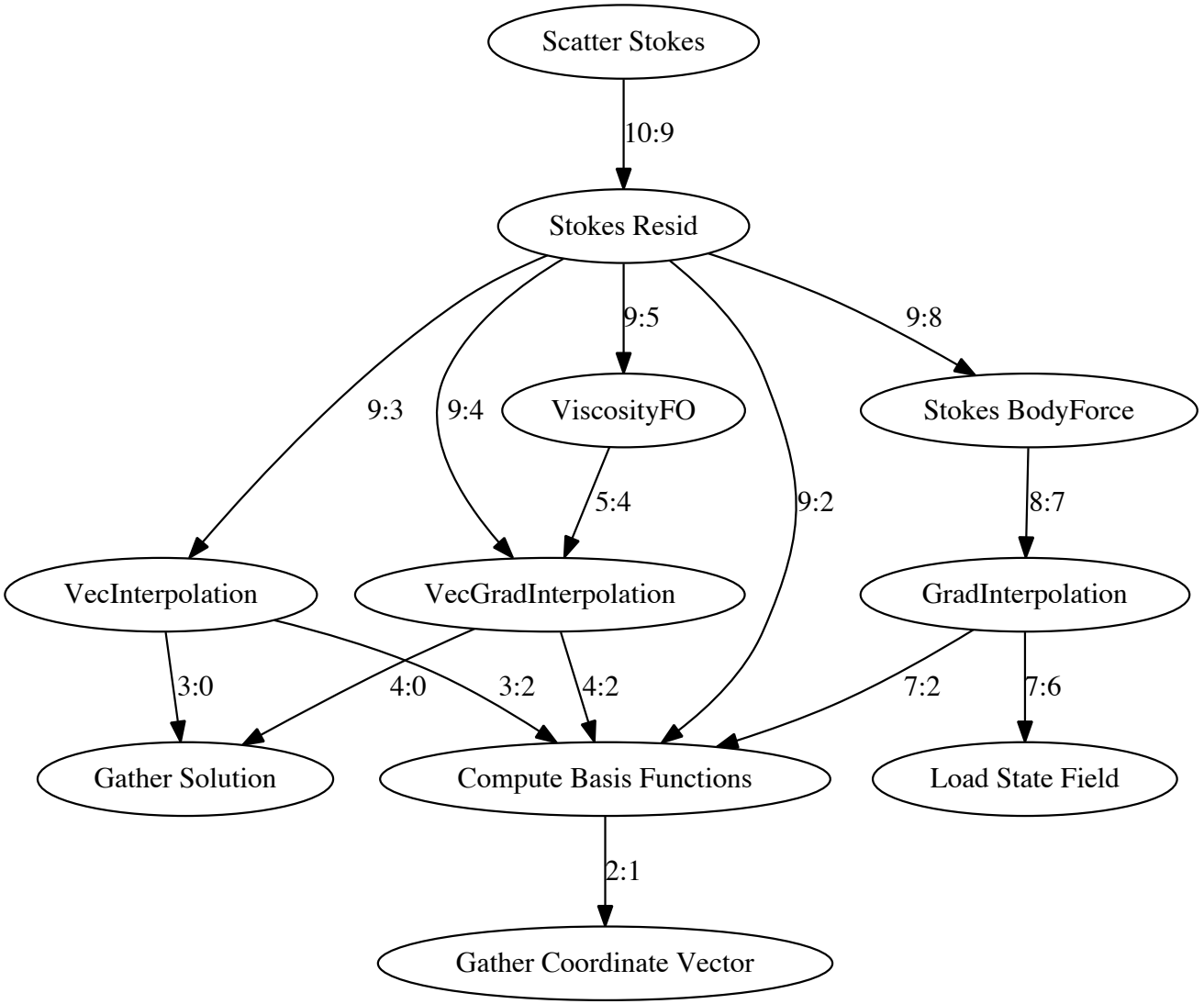
- **Project objective:**
- Provide sea level rise prediction
- Run on new architecture machines (hybrid systems).
  - **50% time spent in FE Assembly**
  - 50% time spent in Linear Solves

Funding Source: SciDAC

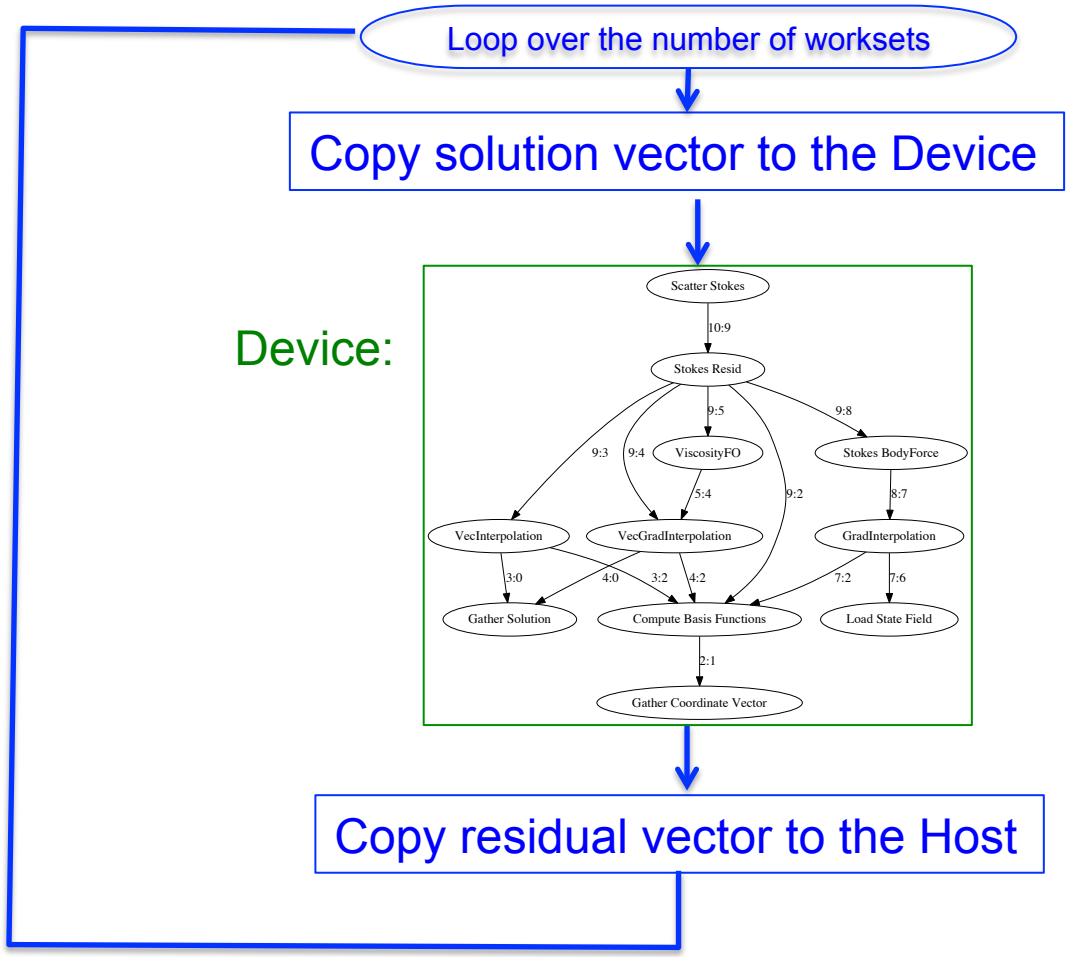
Collaborators: SNL, ORNL, LANL, LBNL, UT, FSU, SC, MIT, NCAR

Sandia Staff: A. Salinger, I. Kalashnikova, M. Perego, R. Tuminaro, J. Jakeman, M. Eldred

Greenland Ice-Sheet model



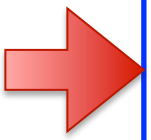
Greenland Ice-Sheet model (Kokkos implementation)



Kokkos_functor example: Jacobian

Sandia National Laboratories

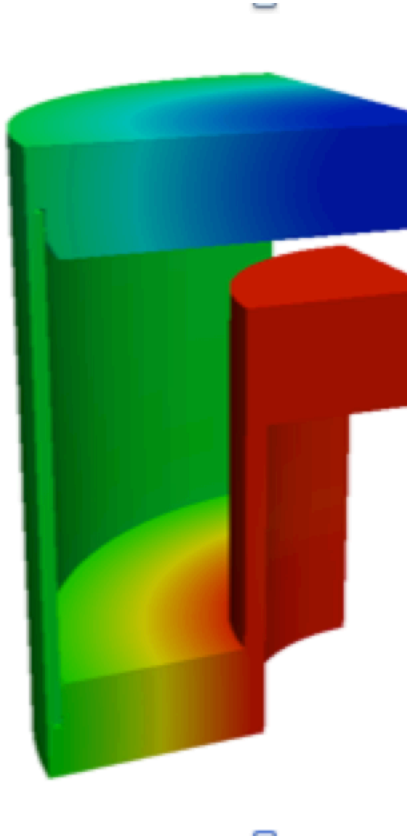
```
template<typename EvalT, typename Traits>
Void Jacobian<EvalT, Traits>::evaluateFields(typename
Traits::EvalData d)
{
for(int cell = 0; cell < worksetNumCells; cell++) {
    for(int qp = 0; qp < numQPs; qp++) {
      for(int row = 0; row < numDims; row++){
        for(int col = 0; col < numDims; col++){
          for(int node = 0; node < numNodes; node++){
            jacobian(cell, qp, row, col) +=
                      coordVec(cell, node, row)
                      *basisGrads(node, qp, col);
          } // node
        } // col
      } // row
    } // qp
  } // cell
}
```



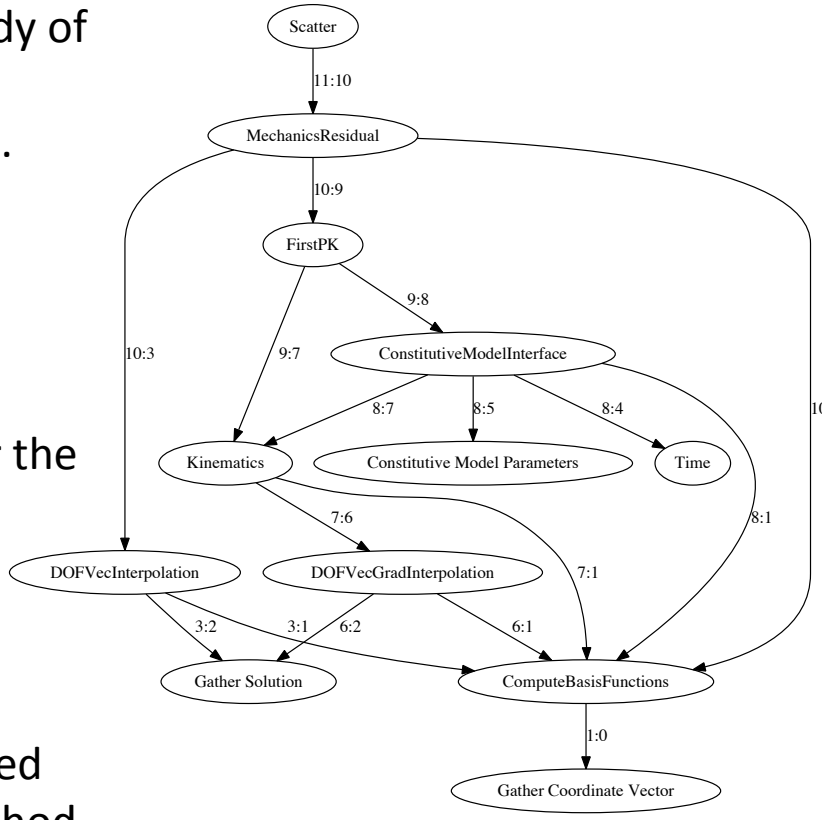
```
**********************************************
template<typename EvalT, typename Traits>
KOKKOS_INLINE_FUNCTION
void Jacobian<EvalT, Traits>:: operator () (const int i) const
 {
   for(int qp = 0; qp < numQPs; qp++) {
     for(int row = 0; row < numDims; row++){
       for(int col = 0; col < numDims; col++){
         for(int node = 0; node < numNodes; node++){
           jacobian(cell, qp, row, col) +=
                      coordVec(cell, node, row)
                      *basisGrads(node, qp, col);
         } // node
       } // col
     } // row
   } // qp

}
//**********************************************
template<typename EvalT, typename Traits>
Void Jacobian<EvalT, Traits>::evaluateFields(typename
Traits::EvalData d)
{
  Kokkos::parallel_for (worksetNumCells, *this);
}
```

Albany ThermoMechanics problem

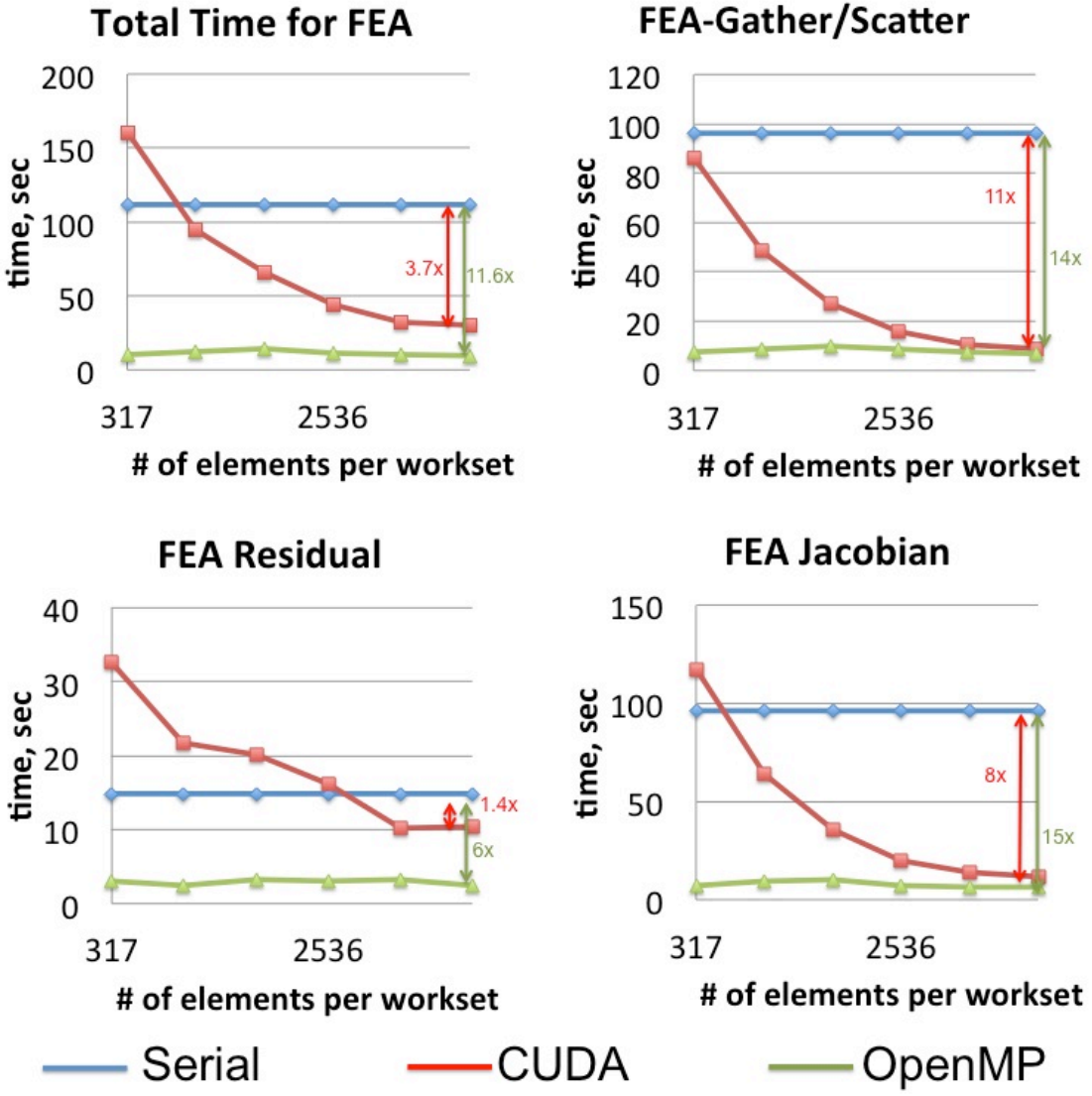


- The ThermoMechanics application problem is a study of how a container will deform under certain external loads.
- The model includes tight coupling of a J2-plasticity mechanics model with heat transfer, where we solve for the displacement vector and temperature as the loading increases.
- A fully-implicit preconditioned Newton-Krylov solution method is used to solve the unstructured-grid finite element discretization.



Performance results

FELIX Performance results



Evaluation environment:

Compton:

42 nodes:

Two 8-core Sandy Bridge Xeon E5-2670 @ 2.6GHz (HT activated) per node,

*24GB (3*8Gb) memory per node,*

Two Pre-production KNC (Intel MIC) 2 per node (57 cores per each)

Shannon:

32 nodes:

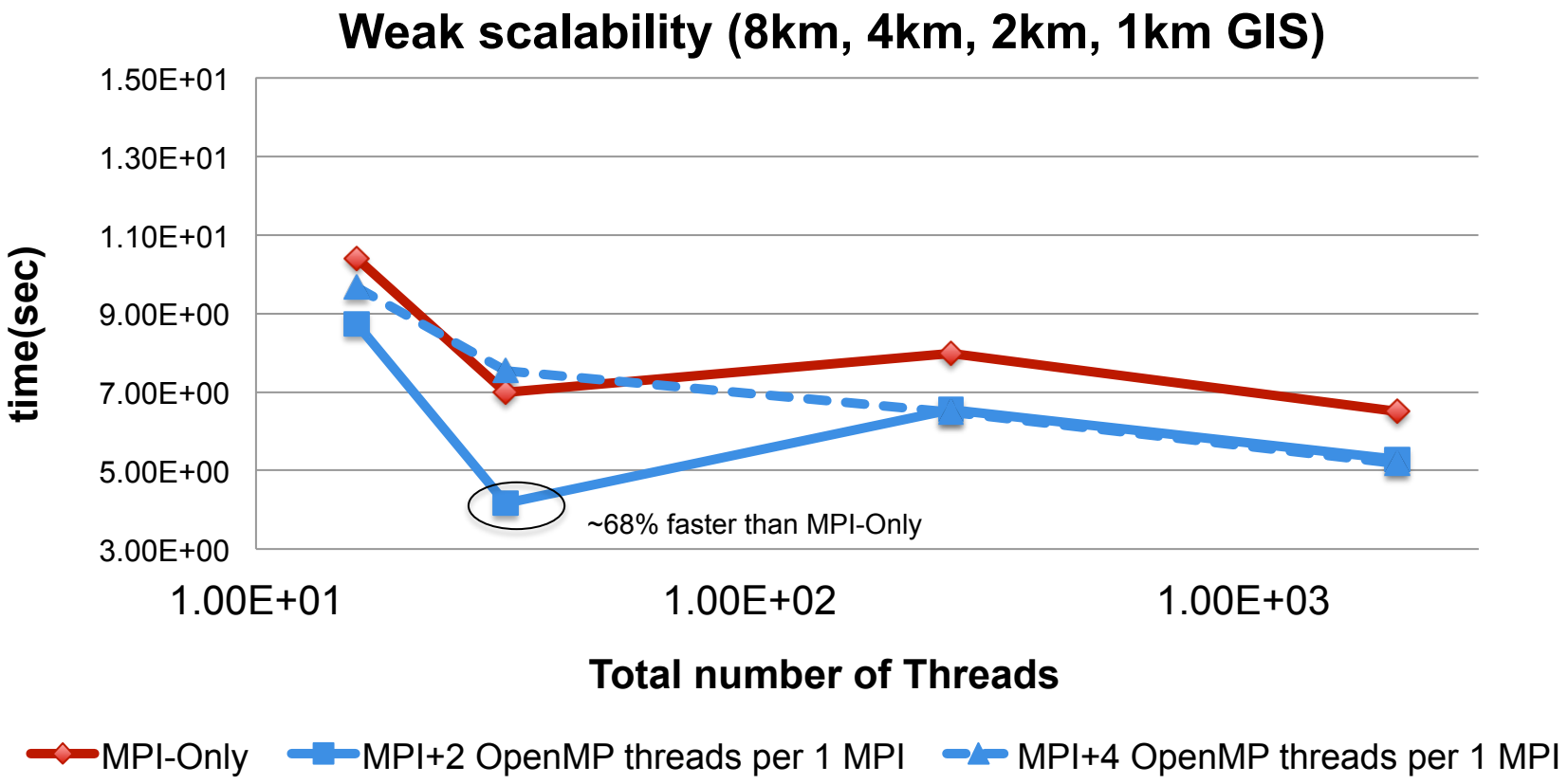
Two 8-core Sandy Bridge Xeon E5-2670 @ 2.6GHz (HT deactivated) per node,

128GB DDR3 memory per node,

2x NVIDIA K20x per node

FELIX code: 1 element block, up to 10144 elements per workset on GPU

FELIX Performance results



Evaluation environment:

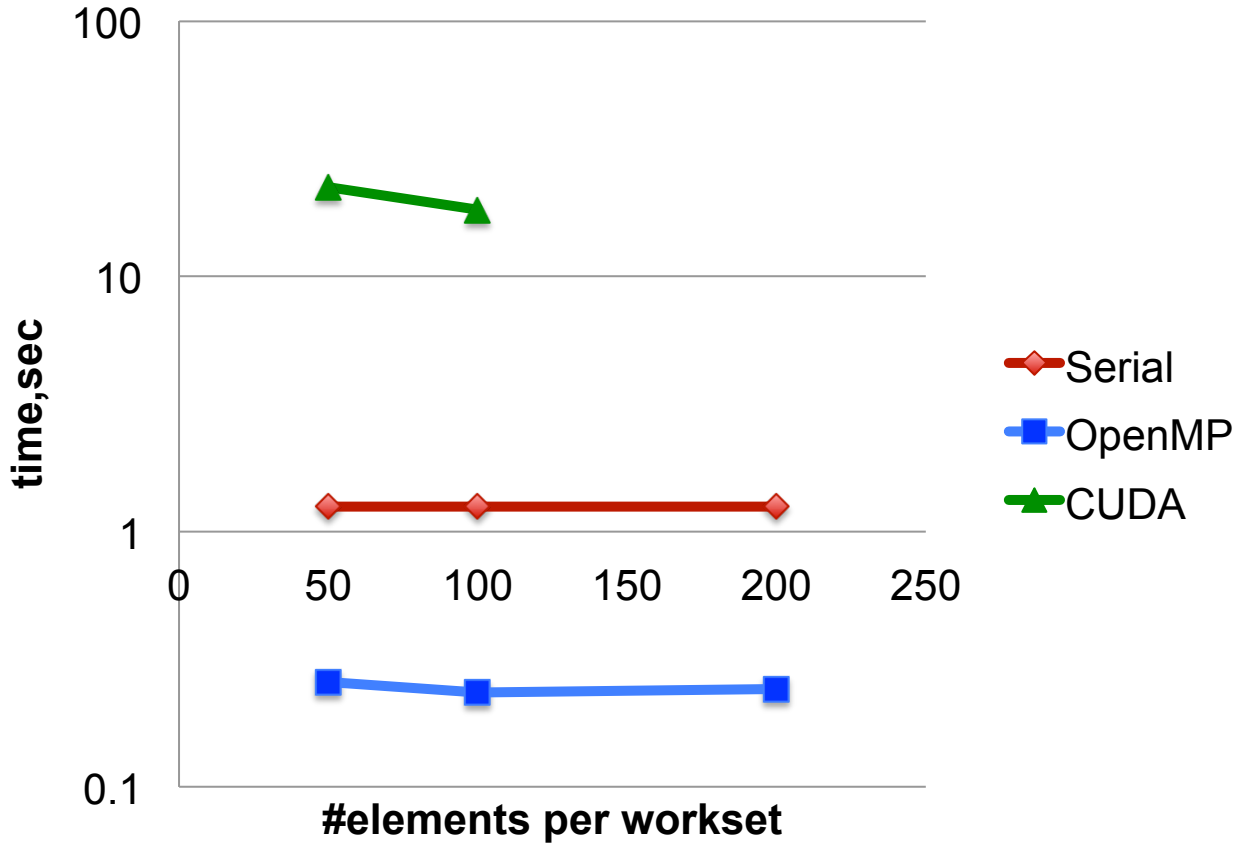
TITAN:

18,688 AMD Opteron nodes:

- 16 cores per node,
- 1 K20X Kepler GPUS per node,
- 32GB + 6GB memory per node

ThermoMechanics problem Performance results

Sandia National Laboratories



Evaluation environment:

Compton:

42 nodes:

Two 8-core Sandy Bridge Xeon E5-2670 @ 2.6GHz (HT activated) per node,
*24GB (3*8Gb) memory per node,*
Two Pre-production KNC (Intel MIC) 2 per node (57 cores per each)

Shannon:

32 nodes:

Two 8-core Sandy Bridge Xeon E5-2670 @ 2.6GHz (HT deactivated) per node,
128GB DDR3 memory per node,
2x NVIDIA K20x per node

TermoMecanics code: several element blocks, up to 100 elements per workset on GPU

Not enough parallelism for GPU

Conclusion

Sandia National Laboratories

Conclusions:

- Kokkos provides a portable implementation environment for emerging node architectures;
- New version of Albany provides an interface for rapid development of architecture-portable Finite Element code;
- MPI-only is not sufficient.

Ongoing work:

- Kokkos kernels optimization
- Porting Trilinos Linear Algebra Libraries to Kokkos