SAND2014-18194PE

Second-generation Trilinos for large-scale low Mach CFD simulations

Sequoia Users Meeting (SUM)

LLNL

September 18, 2014

Paul Lin, Matthew Bettencourt, Stefan Domino, Travis Fisher, Mark Hoemmen, Jonathan Hu, Eric Phipps, Andrey Prokopenko, Sivasankaran Rajamanickam, Christopher Siefert, Stephen Kennon

Sandia National Labs

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.

Sandia National Laboratories

Outline

- Background
  - Brief Trilinos overview
  - Example SNL motivation
- Application code overview
- Tracking performance over time
- Results
- Conclusions
  - Future work

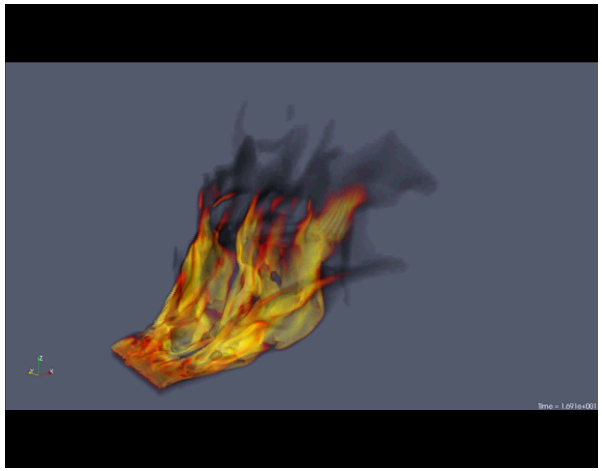
Brief Trilinos Overview

Object-oriented software framework for the solution of large-scale, complex multi-physics engineering and scientific problems

- Current Trilinos (Epetra based):
  - Limited by 32-bit integer global objects
    - need for increasing fidelity for challenging problems at SNL
  - Most packages employ flat MPI-only; future architectures?
- Second-generation Trilinos solver stack
  - No 32-bit limitation on global objects (employs C++ templated data types)
  - Path forward for future architectures (Kokkos; not part of this talk)
- Want Trilinos to impact production engineering applications
  - Significant effort to mature second-generation Trilinos
- PETSc (ANL; Smith, et al.) is another popular solvers framework

Example SNL Motivation: Fire Environment

- Many challenging scientific engineering problems at SNL require increasing fidelity for computational simulation
- Example: accurate simulation of fire environment
- Large-scale hydrocarbon pool fires that occur in accident scenarios
  - Hydrocarbon JP-8 10m fire
  - Hydrocarbon JP-8 pool fire



- Multi-physics pool fire simulation that supported a SNL qualification effort

Sierra Low Mach Application Code

- Sierra/Fuego app used at SNL to simulate fire environment
  - Low Mach number, turbulent reacting flow code
  - Sierra/Framework
  - Trilinos solvers through Finite Element Interface (FEI)
  - 32-bit ordinals in Framework, FEI and Trilinos/Epetra limited simulations to fewer than ~2 billion entities
- Need for new low Mach application: Sierra/Nalu
  - new app code not constrained by 32-bit global entities
  - Employs second-generation Trilinos which provides path forward for future architectures

Sierra/Nalu Low Mach Application Code

- Generalized unstructured, massively parallel, low Mach number variable density turbulent flow code
- Large eddy simulation (LES); spatially filtered equations
- Pressure projection method (equal order interpolation using residual-based pressure stabilization)

Discretization

Control Volume Finite Element (CVFEM)

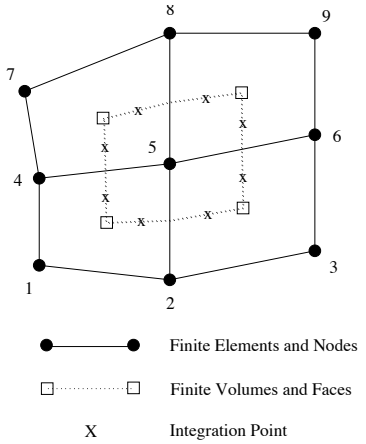
Edge-based Vertex Centered (EBVC)

$$\int \frac{\partial \bar{\rho}}{\partial t}\, dV + \int \bar{\rho}\tilde{u}_i n_i\, dS = 0$$

$$\int \frac{\partial \bar{\rho}\tilde{u}_i}{\partial t}\mathrm{d}V + \int \bar{\rho}\tilde{u}_i\tilde{u}_j n_j \mathrm{d}S + \int \bar{p} n_i \mathrm{d}S = \int \bar{\tau}_{ij} n_j \mathrm{d}S + \int \tau_{u_i u_j} n_j \mathrm{d}S + \int \left(\bar{\rho} - \rho_\circ\right) g_i \mathrm{d}V$$

$$\int \frac{\partial \bar{\rho}\tilde{Z}}{\partial t}\mathrm{d}V + \int \bar{\rho}\tilde{u}_j\tilde{Z} n_j \mathrm{d}S = -\int \tau_{Z u_j} n_j \mathrm{d}S + \int \bar{\rho} D \frac{\partial \tilde{Z}}{\partial x_j} n_j \mathrm{d}S$$

$$\int \frac{\partial \bar{\rho} k^{\mathrm{sgs}}}{\partial t}\mathrm{d}V + \int \bar{\rho} k^{\mathrm{sgs}}\tilde{u}_j n_j \mathrm{d}S = \int \frac{\mu_t}{\sigma_k}\frac{\partial k^{\mathrm{sgs}}}{\partial x_j} n_j \mathrm{d}S + \int \left(P_k^{\mathrm{sgs}} - D_k^{\mathrm{sgs}}\right)\mathrm{d}V$$



Integrating Trilinos in Nalu

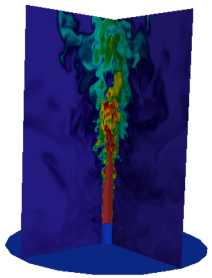
- Trilinos packages used by Nalu

Functionality	Current solver stack	New solver stack
Distributed linear algebra	Epetra	Tpetra
Iterative linear solvers	AztecOO	Belos
Incomplete factorizations	AztecOO, Ifpack	Ifpack2
Algebraic multigrid	ML	MueLu
Partition & load balance	Zoltan	Zoltan2
Direct solvers interface	Amesos	Amesos2

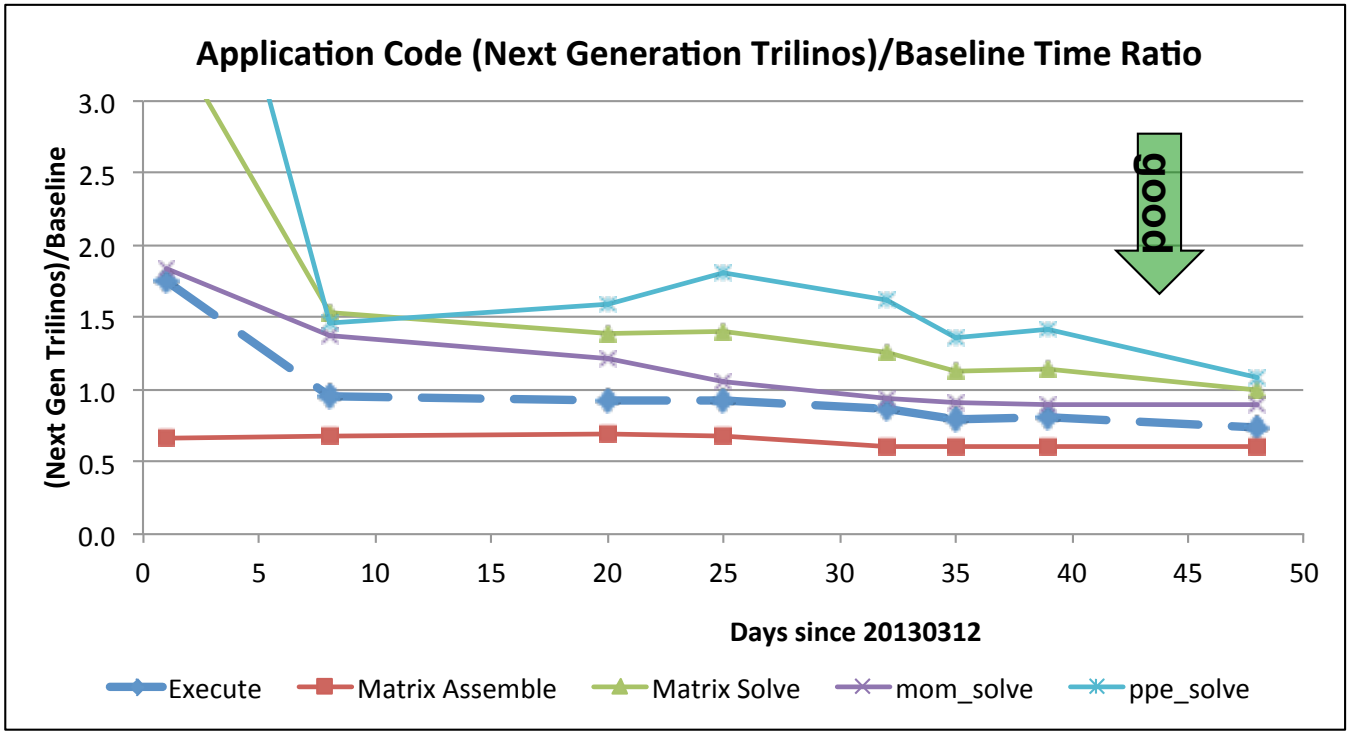
- Matrix assembly and data structures
  - Direct matrix assembly into Tpetra sparse matrices and vectors (in contrast with FEI)
  - Two graphs/matrices: one for owned rows, other for shared
  - Local indices for both owned and shared nodes so do not have to perform local->global lookup

Test case: Mixture fraction-based turbulent jet (Re=6600)

- Large eddy simulation (LES)
- PPE, momentum, subgrid-scale KE, mixture fraction



Tracking Performance Improvements Over Time

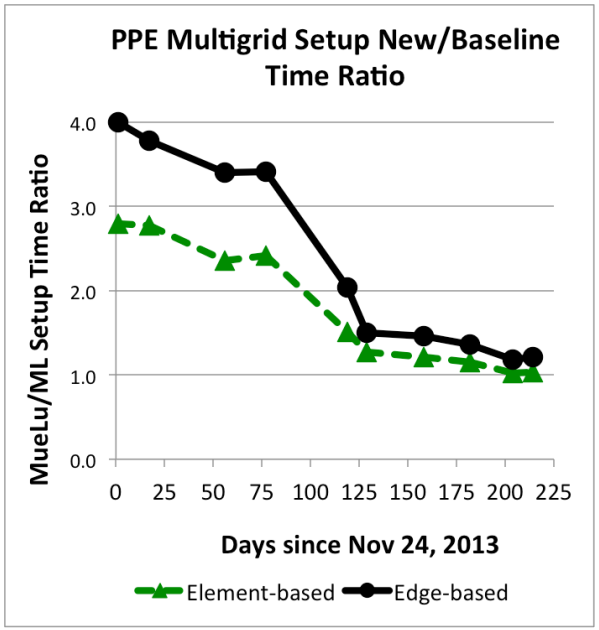


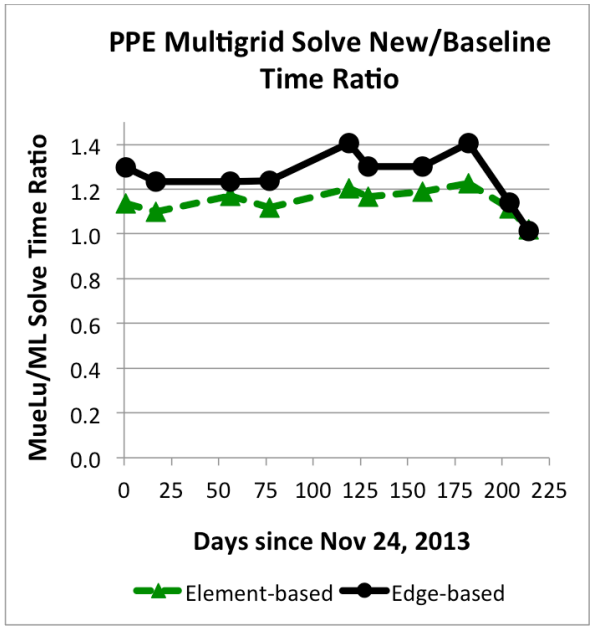
- New app enables high fidelity; don't want to sacrifice perf for smaller problems
- 140M elem, 2048 cores
- Ratio of time for Tpetra/Epetra codes; < 1 means new code is faster
- By end of time period, total and solve times are 26% and 1% faster
- Performance sometimes regressed; need to constantly track performance

Multigrid Preconditioner Setup Performance

- Adaptive mesh: regenerate hierarchy; static mesh: save hierarchy
- Trilinos ML a mature AMG package; used the last ~15 years
- MueLu adds new capability; want MueLu to be competitive with ML

Tracking Nov 2013-July 2014

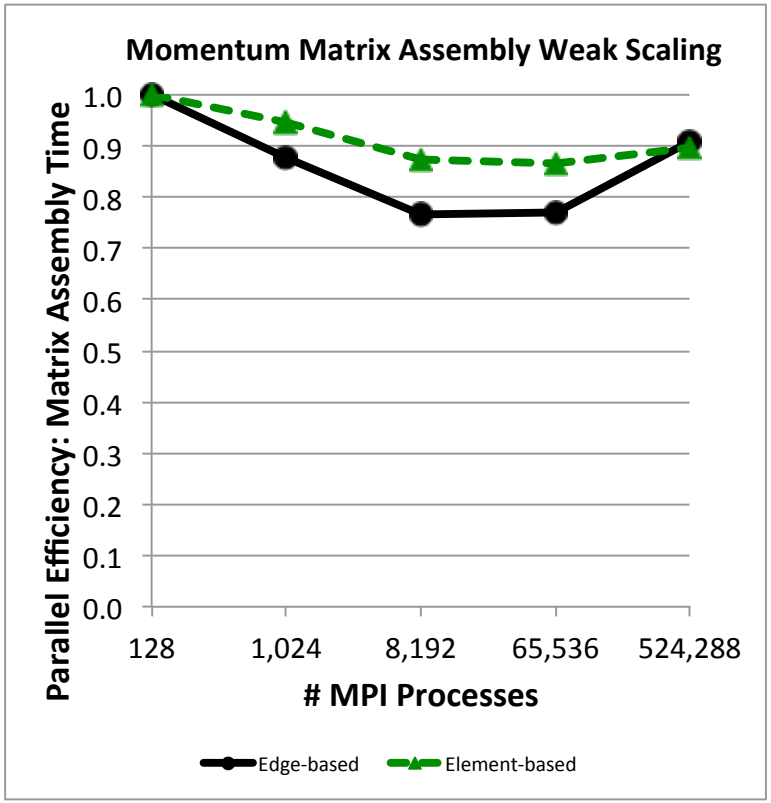


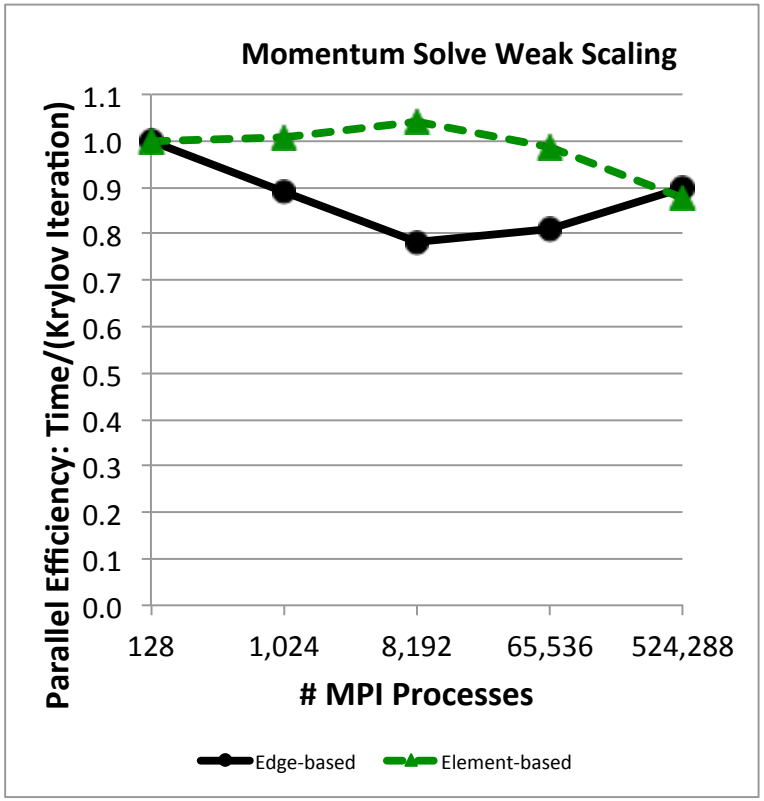


- MueLu/ML setup May 2013: ~6; Nov 2013: ~4; Jul 2014: <=1.2, solve ~1
- However, ratio is worse at scale

- Hypre (Falgout, Yang, Kolev, Baker, et al.) is also a popular AMG library

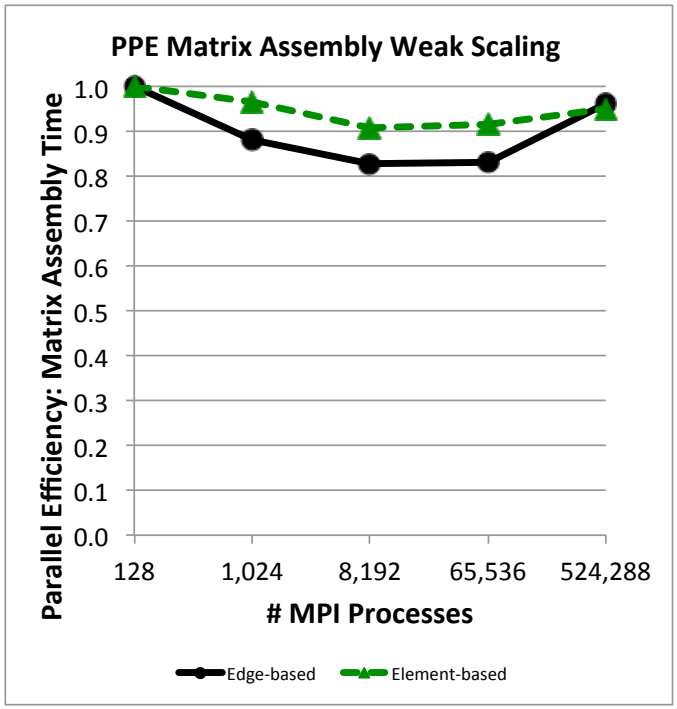
Nalu Weak Scaling Sequoia BG/Q: Coupled Momentum

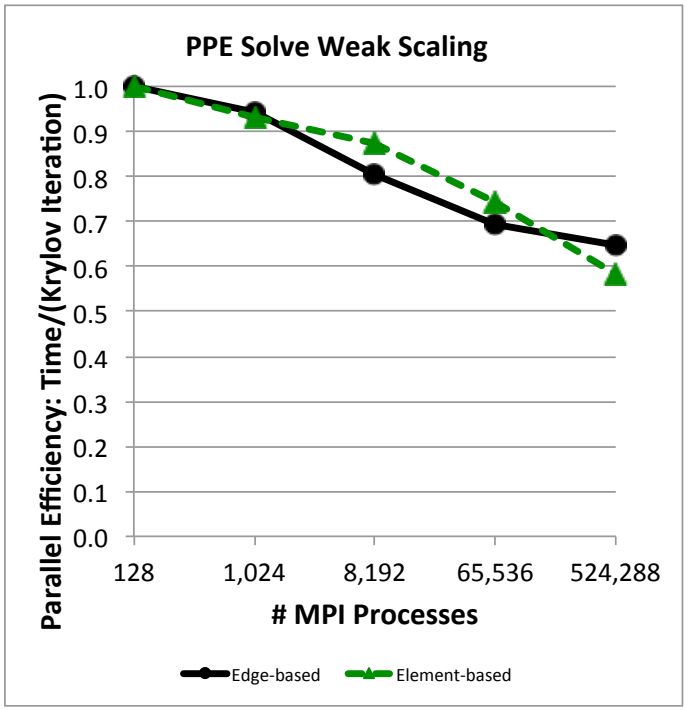




- Largest problem: 8.97 billion elements (26.9 billion row matrix) on 524,288 cores (1 MPI process per core)

Nalu Weak Scaling Sequoia BG/Q: PPE





- Largest problem: 8.97 billion elements (8.97 billion row matrix) on 524,288 cores (1 MPI process per core)
- Large jump in memory usage at 524,000 cores; still investigating

Conclusions and Future Work

- Significant revision of Trilinos' new solver stack
- Integration of new Trilinos into an engineering app
  - 9 billion element simulation on 524,288 cores
  - Nalu used in LANL/SNL Trinity procurement
- New Trilinos solver stack provides capability
  - Arbitrarily large global entities
  - Path forward for future architectures: Tpetra being modified to employ Kokkos (handle manycore nodes and accelerators)
- Sierra/Trilinos integration still needs further work
- Optimize new solver stack for other use cases
  - Focused on 6 core packages; need to mature others
  - Multigrid preconditioner setup scaling
  - Other preconditioners and smoothers, e.g. ILU, etc.
- New Trilinos will enable algorithms research
  - CA-algorithms, advanced multigrid methods, etc.

Thanks For Your Attention!

Paul Lin (ptlin@sandia.gov)

Acknowledgments:

- SNL colleagues: Ryan Bond, Kevin Copps, Eric Cyr, Mehmet Deveci, Karen Devine, Jeremie Gaidamour, Mike Glass, Mike Heroux , Rob Hoekstra, Kyran Mish, Patrick Notz, Brent Perschbacher, Kendall Pierson, Jim Willenbring, Alan Williams
- ACES (LANL/SNL) Cielo team
- LLNL LC and Sequoia BG/Q team (especially John Gyllenhaal, Scott Futral, Roy Musselman)

Funding Acknowledgment:
The authors gratefully acknowledge funding from the DOE NNSA Advanced Simulation & Computing (ASC) program