

HPSF CONFERENCE 2025, CHICAGO, IL
MARCH 5-8, 2025



Exceptional service in the national interest

LINEAR SOLVER CAPABILITIES IN THE TRILINOS PROJECT

Sandia National Laboratories

Jonathan Hu

HPSF Conference 2026

March 16-20, Chicago, IL



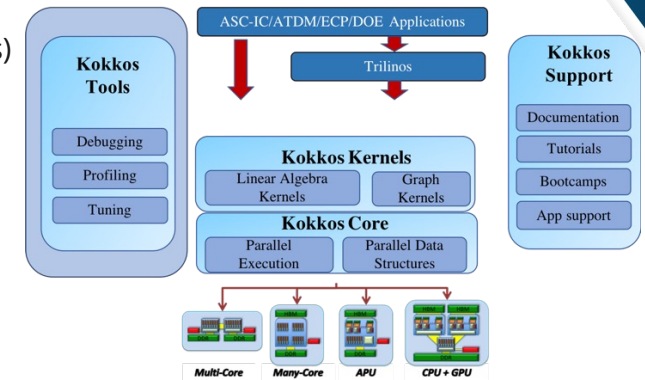
Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525

SAND2026-18533C

TRILINOS PROJECT



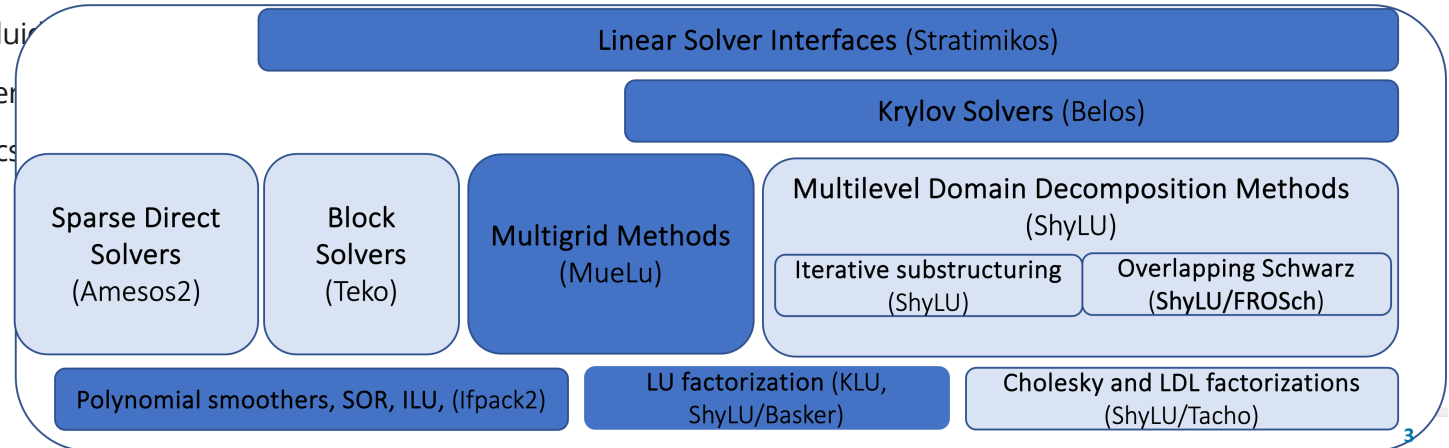
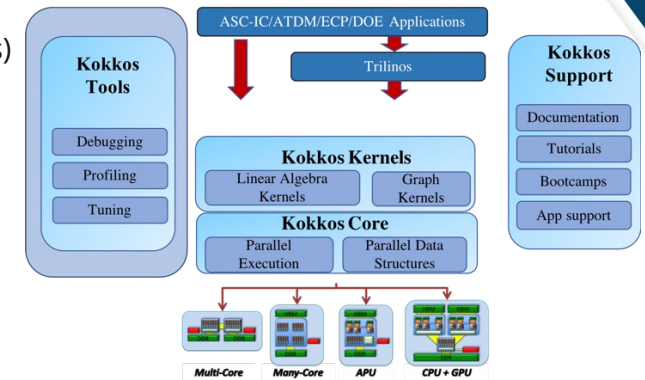
- Numerical libraries project
 - Relies on Kokkos ecosystem for on-node performance portability (github.com/kokkos)
 - Open source (github.com/trilinos/Trilinos)
 - Part of the High Performance Software Foundation (<https://hpsf.io>)
- Supports many architectures
 - CPUs (AMD, Intel)
 - Accelerators (Nvidia, AMD, Intel)
- Enabling technology for variety of simulations codes
 - Sandia internal (thermal fluid, plasma, multimaterial)
 - DOE Office of Science (energy, complex systems modelling)
 - University-led (multiphysics)



TRILINOS PROJECT



- Numerical libraries project
 - Relies on Kokkos ecosystem for on-node performance portability (github.com/kokkos)
 - Open source (github.com/trilinos/Trilinos)
 - Part of the High Performance Software Foundation (<https://hpsf.io>)
- Supports many architectures
 - CPUs (AMD, Intel)
 - Accelerators (Nvidia, AMD, Intel)
- Enabling technology for variety of simulations codes
 - Sandia internal (thermal fluid)
 - DOE Office of Science (ener)
 - University-led (multiphysics)





CURRENT TRILINOS SOLVER DEVELOPERS

- Luc Berger-Vergiat
- Erik Boman
- Filipe Cumaru*
- Vinh Dang
- Max Firmbach*
- Jim Foucar
- Christian Glusa
- Graham Harper
- Andrew Higgins
- Alexander Heinlein*
- Kyrill Ho*
- Jonathan Hu
- Brian Kelley
- Sebastian Kinnewig*
- Jascha Knepper*
- Stephan Koehler*
- Dane Lacey*
- Matthias Mayr*
- Roger Pawlowski
- Malachi Phillips
- Siva Rajamanickam
- Lea Sassmannshausen*
- Chris Siefert
- Heidi Thornquist
- Ray Tuminaro
- Ichitaro Yamazaki
- many other contributors!

*university collaborator

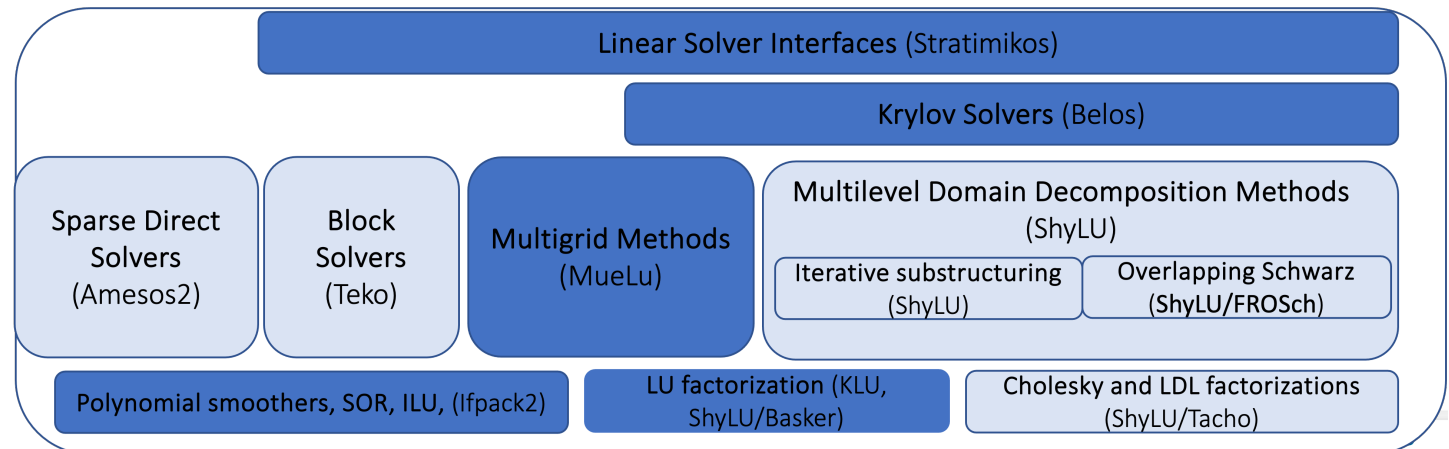


SOLVER PACKAGE TOUR



SOLVER PACKAGE TOUR

- With Trilinos release 17, the Epetra-based solvers were removed
 - Amesos (direct solvers)
 - AztecOO (Krylov solvers)
 - Ifpack (iterative algebraic preconditioners)
 - ML (multigrid)



ITERATIVE KRYLOV METHODS (BELOS)



C++ framework for delivering iterative algorithms for large-scale linear problems.

- Classic Krylov methods
 - GMRES, CG, BiCG, BiCGStab, MinRes
- Single/multiple systems and right-hand sides
 - Single systems
 - Simultaneously solved systems w/ multiple RHS:
 - Sequentially solved systems w/ multiple RHS:
 - Sequences of multiple RHS systems:
- Advanced methods
 - Block methods: block GMRES, block CG/BICG
 - "Seed" solvers: hybrid GMRES
 - Recycling solvers: CGRODR (block recycling GMRES)
 - Restarting techniques, orthogonalization techniques, ...

$$Ax=b$$

$$AX = B$$

$$AX_i = B_i, i=1,\dots,t$$

$$A_i X_i = B_i, i=1,\dots,t$$

ALGEBRAIC SOLVERS (IFPACK2)

- Provides:
 - Iterative methods based on matrix splitting: Jacobi, Gauss-Seidel, SOR
 - Chebyshev polynomial
 - Incomplete factorizations (threshold & fill-based)
 - CPU-based, traditional & iterative
 - Interfaces to GPU-based incomplete factorizations in Kokkos-Kernels
 - Overlapping additive Schwarz with various subdomain solution methods
 - Block Jacobi
 - Block tri-diagonal
- Can be used standalone, as preconditioners to a Krylov method, or within a multigrid algorithm

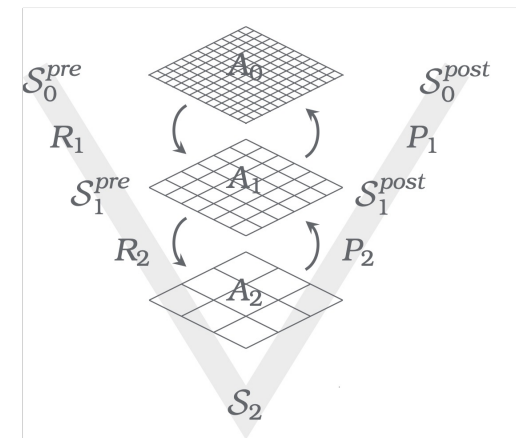




MULTIGRID (MUELU)

- Smoothers: Jacobi, GS, /1 GS, polynomial, ILU (traditional & iterative), block
- Unstructured multigrid algorithms
 - classic smoothed aggregation (SA) AMG
 - non-symmetric AMG
 - AMG for Maxwell's equations (see Christian Glusa's talk)
- Structured Algorithms
 - semi-coarsening AMG, geometric MG
 - structured-grid aggregation-based MG
- Leverages many other Trilinos libraries for setup and solve
 - Sparse distributed linear algebra: **Tpetra**
 - Shared memory parallelism
 - **Kokkos** → architecture portability
 - SpGEMM, distance-2 coloring: **Kokkos-Kernels**
 - Smoothers
 - Distributed smoothers: **lfpack2**
 - Shared memory smoothers: **Kokkos-Kernels**
 - Matrix load balancing: **Zoltan2**
 - Direct Solvers: **Amesos2**

$$A_0 u = f$$

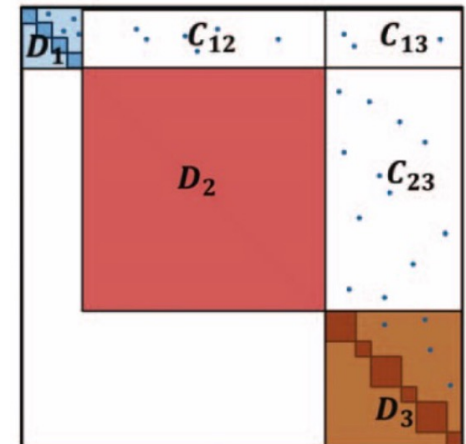


See Christian Glusa's talk @ 3:40pm Thursday, *Multigrid Solvers for Maxwell's Equations in Trilinos*

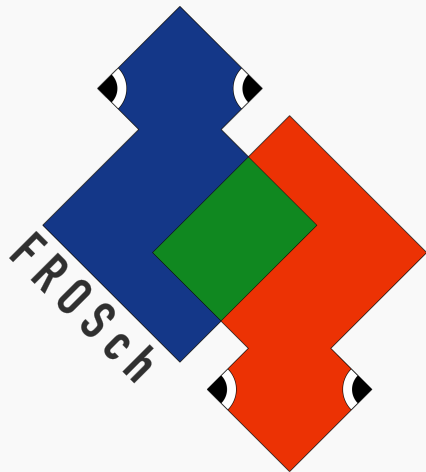


SHARED MEMORY DIRECT SOLVERS AND INCOMPLETE FACTORIZATIONS (SHYLU)

- Basker [Booth, Rajamanickam, Thornquist, '16]
 - Threaded sparse direct solver
 - Targets non-symmetric systems
 - Variant of KLU [Davis, Natarajan, '10]
- Tacho [Kim, Edwards, Rajamanickam, '18]
 - Sparse symmetric factorizations (Cholesky, LDLt, Symmetric LU)
 - GPU-enabled
- FastILU
 - Iterative variants of ILU(k) & sparse-triangular solve
 - Similar to [Chow, Patel, '15]



FROSch (Fast and Robust Overlapping Schwarz) Framework in Trilinos



Software

- Object-oriented C++ domain decomposition solver framework with MPI-based distributed memory parallelization
- Part of TRILINOS with the parallel linear algebra based on TPETRA
- Node-level parallelization and performance portability on CPU and GPU architectures through KOKKOS and KOKKOSKERNELS
- Accessible through unified TRILINOS solver interface STRATIMIKOS

Methodology

- **Parallel scalable multi-level Schwarz domain decomposition preconditioners**
- **Algebraic construction** based on the parallel distributed system matrix
- **Extension-based coarse spaces**

Team (active)

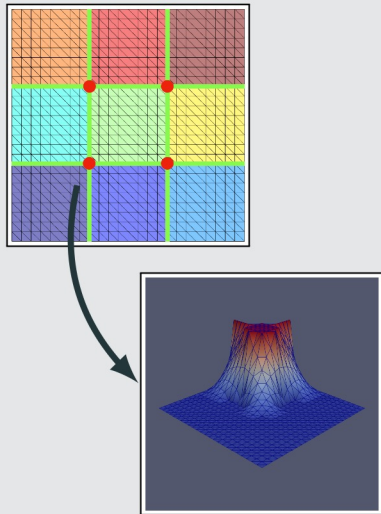
- | | |
|---------------------------------|---------------------------------|
| ▪ Filipe Cumaru (TU Delft) | ▪ Stephan Köhler (TUBAF) |
| ▪ Alexander Heinlein (TU Delft) | ▪ Friederike Röver (TUBAF) |
| ▪ Kyrill Ho (UCologne) | ▪ Siva Rajamanickam (SNL) |
| ▪ Sebastian Kinnewig (LUH) | ▪ Oliver Rheinbach (TUBAF) |
| ▪ Axel Klawonn (UCologne) | ▪ Lea Saßmannshausen (UCologne) |
| ▪ Jascha Knepper (UCologne) | ▪ Ichitaro Yamazaki (SNL) |

Algebraic Multiscale Finite Element Coarse Space in FROSch

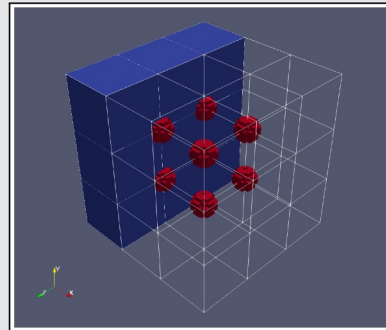


Coarse space definition

The multiscale finite element method (MsFEM) coarse basis functions are computed as energy-minimizing extensions of solutions to reduced boundary condition problems on the interface.

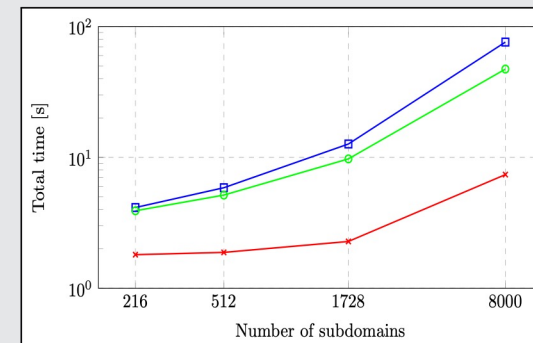
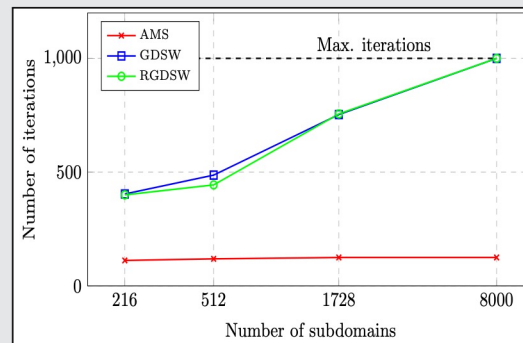


Weak scalability – Heterogeneous 3D poisson problem



- The **MsFEM coarse space** is **robust** for **high coefficient contrast** and fully **algebraic** (uses only global system matrix).
- Compared with **GDSW/RGDSW coarse spaces**, the extra setup cost is **negligible**, yielding **strong parallel efficiency**.

Cf. **Cumaru, Heinlein, and Hajibeygi (acc. 2025)**.

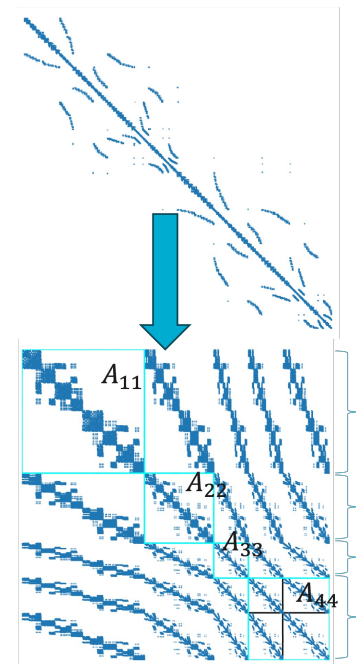


High coefficient inclusions on the coarse vertices with a coefficient $\alpha = 10^8$ in the red regions and $\alpha = 1$ elsewhere. The problems were solved using the **preconditioned conjugate gradient** method in at most 1000 iterations. Computations were run on the **Dutch National Supercomputer Snellius**.

Slide courtesy A. Heinlein, A.Heinlein@tudelft.nl

MONOLITHIC BLOCK MULTI-PHYSICS PRECONDITIONING (TEKO)

- Provides block inverse schemes:
 - Generic: Block Jacobi, (Hierarchical) Block Gauss-Seidel
 - Navier-Stokes: SIMPLEC, PCD, LSC
 - Extensible: Easy-to-develop user-defined block splitting
- Sub-blocks can be inverted via various Trilinos algorithms:
 - AMG, Krylov, incomplete fact., SOR techniques
 - *Adaptive* sub-block solvers based on target residual reduction
- Interfaces with Tpetra types
 - GPU sub-block assembly for Tpetra
- Able to convert standard CRS matrix to blocked system
- See Malachi Phillips' talk @ 11:40am on Thursday
 - *Block-based Algebraic Multigrid Preconditioners in Trilinos/Teko*



(Teko means “fuse”)

GETTING STARTED

TRILINOS RESOURCES FOR NEW USERS



- ATPESC solver tutorials
 - <https://tinyurl.com/atpesc-2025-trilinos-solvers> or
- MueLu tutorial
 - <https://muelu.github.io>
- User guides
 - MueLu (<https://trilinos.github.io/pdfs/mueluguide.pdf> or in the Trilinos github repo)
 - Ifpack2 (in Trilinos github repo)





STANDALONE MUELU MATRIX SOLVER

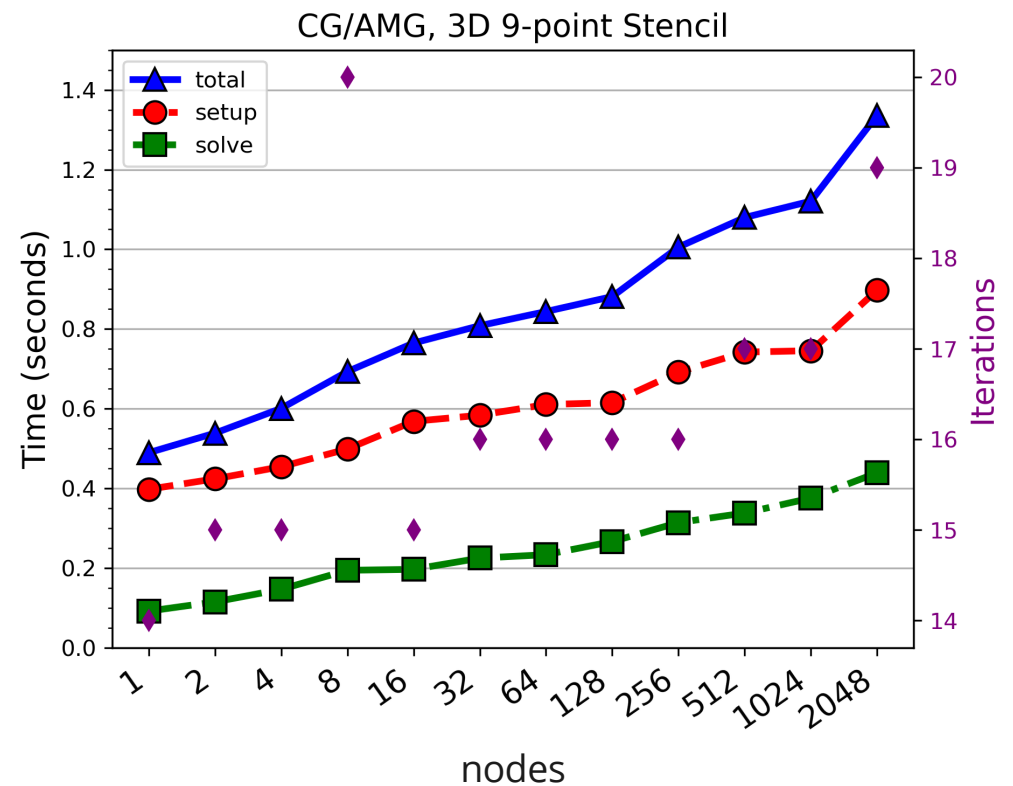
- MueLu provides a stand-alone example for experimentation: "MueLu_Driver.exe"
- Can generate canonical matrix types (Poisson, convection-diffusion, elasticity)
- Can read user-supplied matrix, right-hand sides from file
- Straight-forward to test variety of solver/preconditioner combinations
 - Krylov methods (Belos)
 - Algebraic precondition. (Ifpack2)
 - Direct solvers (Amesos2)
 - AMG (MueLu)

```
1 <ParameterList name="MueLu">
2
3 <!--
4 For a generic symmetric scalar problem, these are the recommended settings for MueLu.
5 -->
6
7 <!-- ===== GENERAL ===== -->
8 <Parameter name="verbosity" type="string" value="high"/>
9
10 <Parameter name="coarse: max size" type="int" value="1000"/>
11
12 <Parameter name="multigrid algorithm" type="string" value="sa"/>
13
14 <!-- reduces setup cost for symmetric problems -->
15 <Parameter name="transpose: use implicit" type="bool" value="true"/>
16 <Parameter name="transfers: half precision" type="bool" value="false"/>
17
18 <!-- start of default values for general options (can be omitted) -->
19 <Parameter name="max levels" type="int" value="10"/>
20 <Parameter name="number of equations" type="int" value="1"/>
21 <Parameter name="sa: use filtered matrix" type="bool" value="true"/>
22 <!-- end of default values -->
23
24 <!-- ===== AGGREGATION ===== -->
25 <Parameter name="aggregation: type" type="string" value="uncoupled"/>
26 <Parameter name="aggregation: drop scheme" type="string" value="classical"/>
27 <!-- Uncomment the next line to enable dropping of weak connections, which can help AMG convergence
28 for anisotropic problems. The exact value is problem dependent. -->
29 <!-- <Parameter name="aggregation: drop tol" type="double" value="0.02"/> -->
30
31 <!-- ===== SMOOTHING ===== -->
32 <Parameter name="smoother: type" type="string" value="CHEBYSHEV"/>
33 <ParameterList name="smoother: params">
34 <Parameter name="chebyshev: degree" type="int" value="2"/>
35 <Parameter name="chebyshev: ratio eigenvalue" type="double" value="7"/>
36 <Parameter name="chebyshev: min eigenvalue" type="double" value="1.0"/>
37 <Parameter name="chebyshev: zero starting solution" type="bool" value="true"/>
38 </ParameterList>
39
40 <!-- ===== REPARTITIONING ===== -->
41 <Parameter name="repartition: enable" type="bool" value="true"/>
42 <Parameter name="repartition: partitioner" type="string" value="zoltan2"/>
43 <Parameter name="repartition: start level" type="int" value="2"/>
44 <Parameter name="repartition: min rows per proc" type="int" value="800"/>
45 <Parameter name="repartition: max imbalance" type="double" value="1.1"/>
46 <Parameter name="repartition: remap parts" type="bool" value="false"/>
47 <!-- start of default values for repartitioning (can be omitted) -->
48 <Parameter name="repartition: remap parts" type="bool" value="true"/>
49 <Parameter name="repartition: rebalance P and R" type="bool" value="false"/>
50 <ParameterList name="repartition: params">
51 <Parameter name="algorithm" type="string" value="multijagged"/>
52 </ParameterList>
53 <!-- end of default values -->
54
55 </ParameterList>
```



MUELU DRIVER, 3D MODEL PROBLEM

- 3D 9-point stencil
- Weak scaling, 300k rows/rank
- Preconditioned conjugate gradient
 - aggregation-based algebraic multigrid
 - Polynomial smoother on all levels
- Frontier
 - AMD MI-250x
 - 8 to 16384 MPI ranks



RELATED TALKS



- Thursday:
 - Jonathan Hu @ [9:40am](#), *Recent Developments in Trilinos Linear Solvers*
 - Malachi Philips @ [11:40am](#), *Block-based Algebraic Multigrid Preconditioners in Trilinos/Teko*
 - Christian Glusa @ [3:40pm](#), *Multigrid Solvers for Maxwell's Equations in Trilinos*