SAND2023-05197C

Sandia National Laboratories

Exceptional service in the national interest

Linear Solvers In Trilinos

euroTUG 2023

Luc Berger-Vergiat

European Trilinos User Group Meeting 2023
Delft University of Technology, June 28-30, 2023

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly artment of Energy's National Nuclear Security Administration under contract DE-NA0003525.

PACKAGES PRESENTED

Focus on Tpetra

- Kokkos Core/Kokkos Kernels → on node (no MPI) portable algorithms
- Tpetra → Distributed linear algebra
- Belos → Iterative solvers
- Amesos2 →Direct solvers interfaces
- Ifpack2 → Preconditioners
- MueLu → Multigrid methods
- Zoltan2 → Repartitioning algorithms

TLDR:

- Recent work mostly on GPU algorithms
- Epetra stack is not recommended,
  - switch old codes to Tpetra
  - Start new codes with Tpetra
- Slowly deprecating/removing old packages from framework

LINEAR ALGEBRA

KOKKOS ECOSYSTEM INTRO

Kokkos ecosystem: node level performance portability (https://kokkos.github.io, kokkosteam.slack.com).

Two major libraries

- Kokkos Core (https://github.com/kokkos/kokkos, https://kokkos.github.io/kokkos-core-wiki/)
  - programming model (policies, executions patterns, memory management…)
  - std algorithm (sorting, math functions, random numbers…)
  - containers (views, array, StaticCrsGraph, unorderedMap…)
- Kokkos Kernels (https://github.com/kokkos/kokkos-kernels)
  - BLAS
  - Sparse linear algebra + solvers
  - Graph algorithms

Additional libraries

- Kokkos Tools: debugging and profiling tools
- Kokkos remote spaces:
- pyKokkos: using Kokkos from python

KOKKOS CORE UPDATE

Recent releases

- Kokkos 4.0
  - Moving HIP out of experimental
  - Require c++17 → see requirements (https://kokkos.github.io/kokkos-core-wiki/requirements.html)
  - SharedSpace → defines memory accessible by all execution spaces enabled
  - Parallel Scan with View return type
  - Extend MD to hierarchical parallelism: TeamThreadMDRange, ThreadVectorMDRange and TeamVectorMDRange
- Kokkos 4.1
  - MDSpan as implementation of Kokkos::View
  - Lots of SIMD improvement
  - Remove Trilinos subpackages (does not remove targets!)

With Kokkos 4.0 deprecated code in 3.X is no longer supported!

KOKKOS KERNELS UPDATE

Recent releases

- Kokkos Kernels 4.0
  - Directories reorganization by component
  - BLAS Level 1 complete
  - Two new ILU: MDF and parILU_t
  - SpGEMM TPL refactor and upgrades (cuSPARSE, rocSPARSE)
- Kokkos Kernels 4.1
  - BLAS: execution space argument added for stream/queue support
  - ILUK on stream
  - ODE solvers
  - BLAS level 2: mostly complete for GE matrices

After 4.0, some support for deprecated Kokkos 3.X features is removed.

TPETRA INTRO

Tpetra: distributed linear algebra capabilities

- Communication Layer
  - Map describe object distribution
  - Import/Export use two Maps to represent data movement
- Sparse linear algebra support
  - Vector/MultiVector, dot, norm
  - CrsGraph, CrsMatrix, Matrix-Vector and Matrix-Matrix product
  - BlockCrsMatrix, useful for Multiphysics representation
  - Leverage Kokkos Core/Kernels for portability

TPETRA UPDATES

Recent updates

- No major changes to algorithms
- New debug hook to track memory transfers (HtoD, DtoH)
- Improved BlockCrs support
- Fully working with HIP backend
- Partially working with SYCL backend
- Moving more code to GPU in Transfer and FillComplete (TAFC)
- Emphasis on performance testing and improvements

ZOLTAN2 INTRO

Zoltan2 repartitioner

- Feature
  - Coloring
  - Reordering / Permutations
  - Partitioning / Load balancing
    - Geometric: Nested Dissection (ND), multijagged
    - Graph Based: PT-Scotch, ParMETIS
    - Hypergraph: PHG and PaToH
- Updates
  - Sphynx: new spectral partitioning methods
  - GPU implementation of select algorithms

LINEAR SOLVERS

BELOS

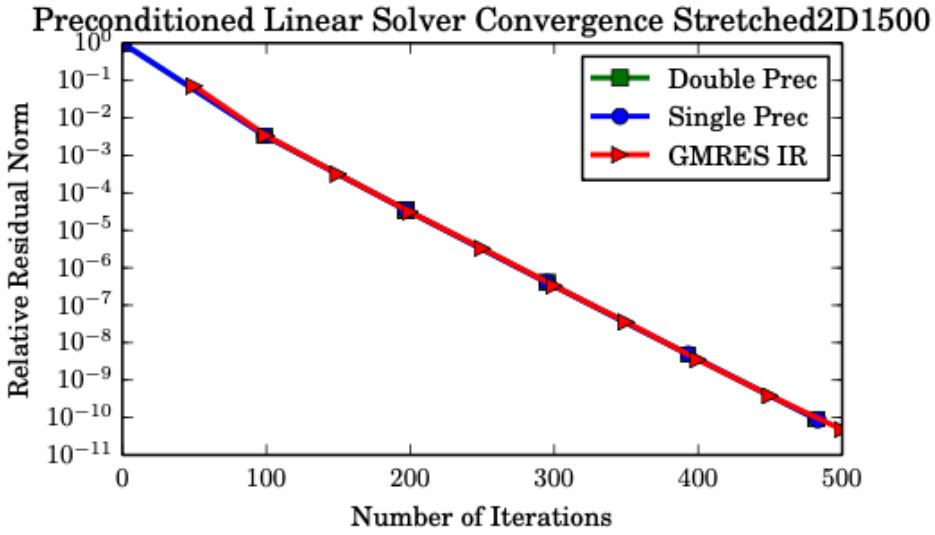
Iterative linear solvers

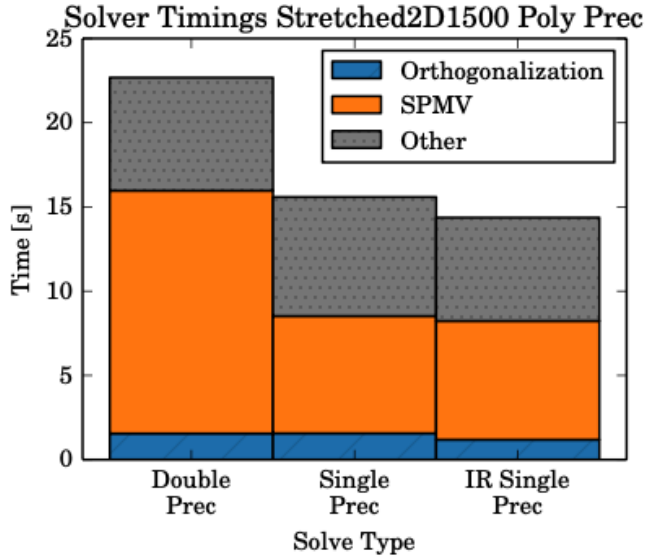
- Krylov solvers
  - CG and variants: PCPG, BlockCG, CGPipeline, CGSingleReduce
  - BiCGStab
  - GMRES and variants: BlockGMRES, GMRESSingleReduce, GMRESSstep
  - MINRES
  - TFQMR
- Other solvers
  - FixedPoint
  - LSQR

Update

- Support for Kokkos backend
- Mixed precision solver GMRES-IR

BELOS MIX PRECISION





AMESOS2

Direct Solvers

- Implements
  - KLU/KLU2
  - Basker
- Interfaces
  - Superlu/Superlu_dist
  - MUMPS
  - UMFPACK

Updates

- ECP work with SuperLU and UMFPACK teams
  - Updated interfaces to leverage GPU versions of TPLs
  - Currently testing and evaluating the new interfaces

IFPACK2 INTRO

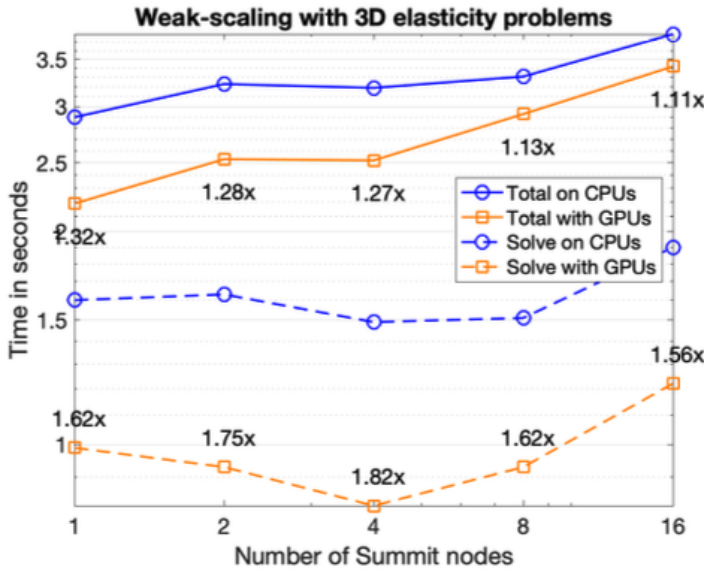
Preconditioner algorithms

- Three concepts
  - Domain Decomposition (solution diffusion across MPI ranks)
  - Local Container (local matrix is banded, sparse, triangular, dense, line ordered…)
  - Local Solver, any preconditioner Gauss-Seidel, ILU, Chebyshev, Jacobi…
- Domain decomposition
  - Additive Schwarz, FROSCH
  - Control overlap length, overlap contribution, coarse problem solve
- Local Solver
  - Solver type, Number of iterations, damping factor

IFPACK2 UPDATE

Focus on ILU and FROSCH

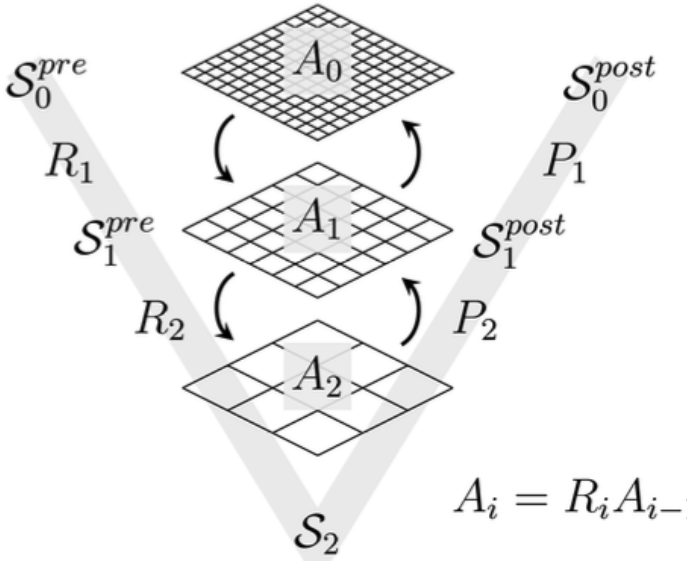
- More ILU implementations
  - ILUK: serial or parallel
  - FastILU: approximate iterative factorization
  - ParILU_t: approximate iterative method (skips some synchronization)
  - Similar variants for triangular solver (TRSV, fastTRSV…)
- FROSCH
  - Two level Schwarz
  - CPU and GPU versions
  - Good scalability (see left)
  - GPU accelerates solves phase the most (2x factor)



MUELU INTRO

Multigrid methods package

- Part of the second-generation of Trilinos
  - Templated on scalars, ordinals and nodes
- Multigrid is an optimal complexity O(n) solver for linear systems.
  1. Start with a "fine grid”
  2. Smooth error, transfer to coarser grid
  3. Repeat 2…
  4. Perform a directsolve on the "coarsest grid”
  5. Transfer to finer grid, smooth error
  6. Repeat 5…
  7. Transfer to original "fine grid", smooth error
- "Is multigrid right for me?”
  - When backslash doesn't cut it

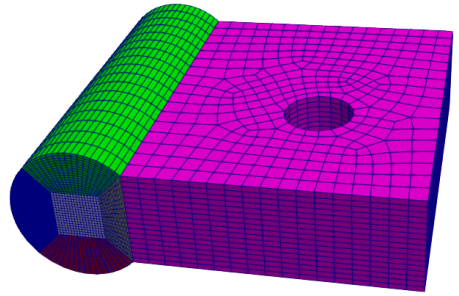


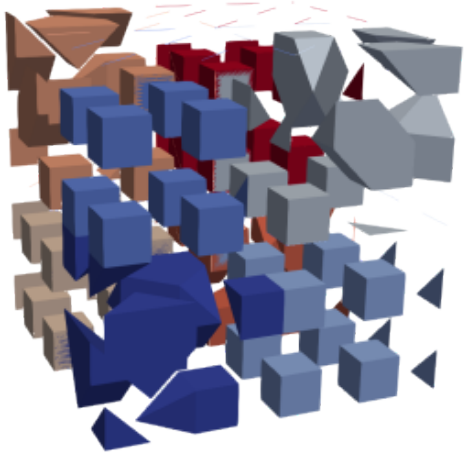
$$A_i = R_i A_{i-1} P_i$$

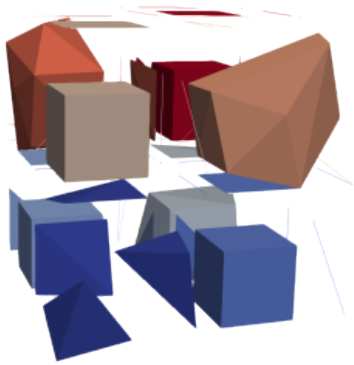
MUELU CAPABILITIES

Main features of MueLu

- Use as precondioner or solver
- Supports many coarsening/smoothers
  - Geometric coarsening
  - Smoothed aggregation
  - Pairwise aggregation
- Matrix Free variant
- Patch-based smoothers







CONCLUSION

SUMMARY

- Kokkos Ecosystem as TPL
- Lots of performance portability work
- More performance tuning upcoming
- Old Epetra stack on the way out
- Looking at production/large scale runs
- More upcoming algorithmic opportunities on future architectures
  - Grace-Hopper
  - MI300
  - Data flow accelerators?

Any Questions?