SAND2019-7414C

June 27, 2019

# Trilinos Multigrid Solvers

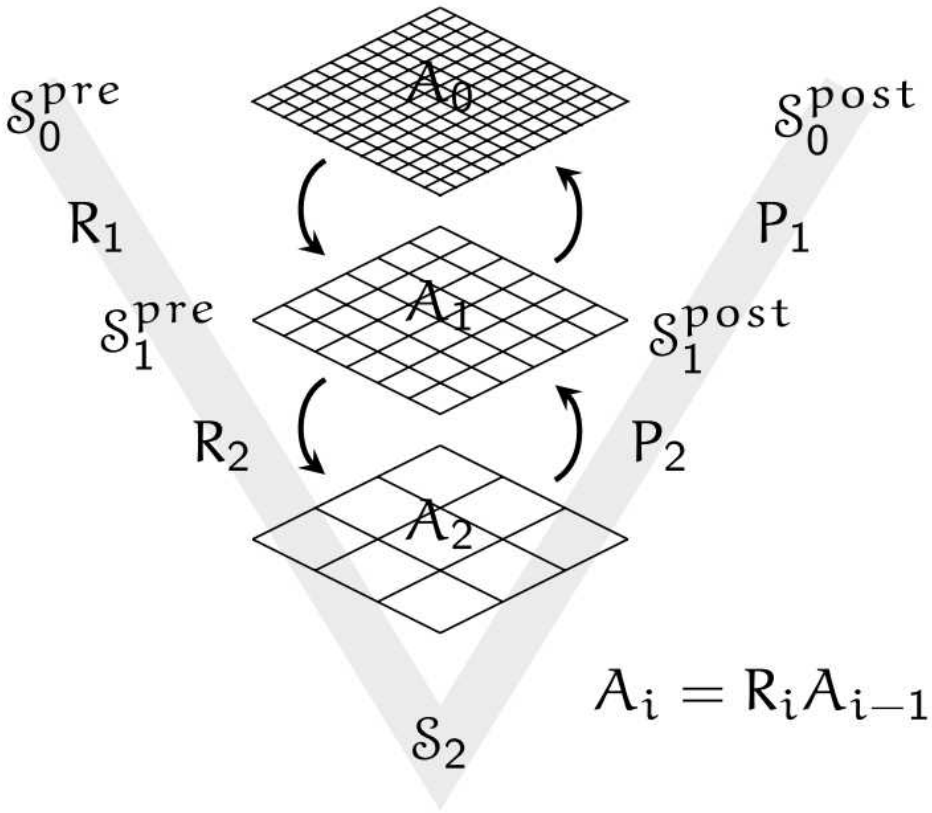
Luc Berger-Vergiat, Jonathan Hu
Center for Computing Research

CEA DAM/NNSA Annual Meeting
June 25-27 - Park City, Utah
SAND 2019-XXXXX

# Outline

Sandia National Laboratories

# MueLu framework




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

Goal: provide a framework to develop MG algorithms

In practive: use xml to specify each component used on each grid.

More info:

- MueLu user's guide [1]
- MueLu tutorial [2]

# Motivations

Sandia National Laboratories

Structured grid algorithms

- enable line smoothing on all levels for SPARC problems (ATDM)
- accelerate MG setup for high fidelity simulations in Drekar
- can provide preconditioner for ExaWind background solve (ECP)

Maxwell solver

- critical enabling technology for EMPIRE application (ATDM)
- key solver for ALEGRA shock physics code

# Outline

Sandia National Laboratories

# Background

Sandia National Laboratories

Team:

Ray Tuminaro, Matthias Mayr, Peter Ohm, and Luc Berger-Vergiat

Task:

introduce structured grid algo in MueLu to support HHG effort

Goal:

leverage grid structure to gain performance on manycore / GPU platforms

# Structured MG approach

Sandia National Laboratories

1. Uncoupled aggregation $\rightarrow$ Structured aggregation
2. Tentative prolongator $\rightarrow$ Geometric interpolation (const. or lin.)
3. Prolongator smoothing $\rightarrow$ use fused kernel for lower launch cost
4. Chebyshev smoother $\rightarrow$ structured SpMV
5. Use V-cycle with 10 levels max, coarse max size: 2000 $\rightarrow$ switch to default MG after repartitioning

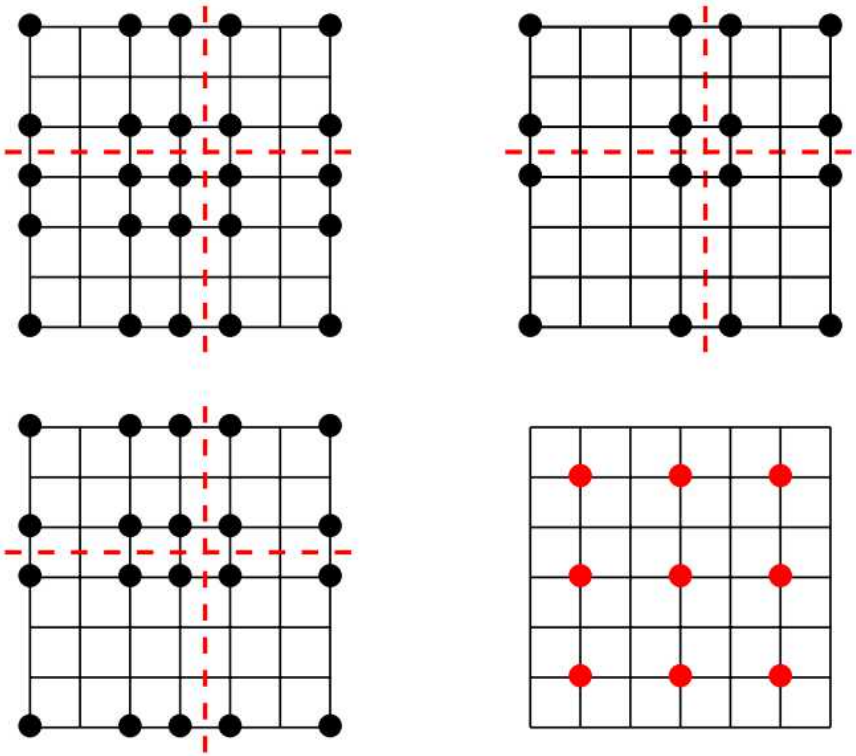
# Aggregation

Sandia National Laboratories

Goals:

- Construct the graph of P
  $\rightarrow$ symbolic phase of P construction
- Also construct aggregates for AMG compatibility
- Construct IndexManager for geometric interpolation
- Scale on many-core platform
- Run on GPU platform (hopefully faster than on many-core, how much?)

# Uncoupled-structured coarsening



Red dashed lines - -, represent processor boundaries.

1. Coarsening rates are variable and independent in each direction,
2. coarse points are chosen to include rank boundary points (better for BC),
3. coarsening is continuous within a rank
4. coarsening rate is automatically reduced at rank boundary.

# Geometric Interpolation Prolongator

Sandia National Laboratories

Goals:

- Perform piece-wise constant interpolation
  $\rightarrow$ can replace Tentative prolongator for prolongator smoothing
- Perform piece-wise linear interpolation
  $\rightarrow$ naturally provides smoother prolongation than piece-wise const.
- Assumes graph as an input
  $\rightarrow$ better reuse speed-up than aggregation based reuse

# RAP product

Sandia National Laboratories

High potential for optimization.

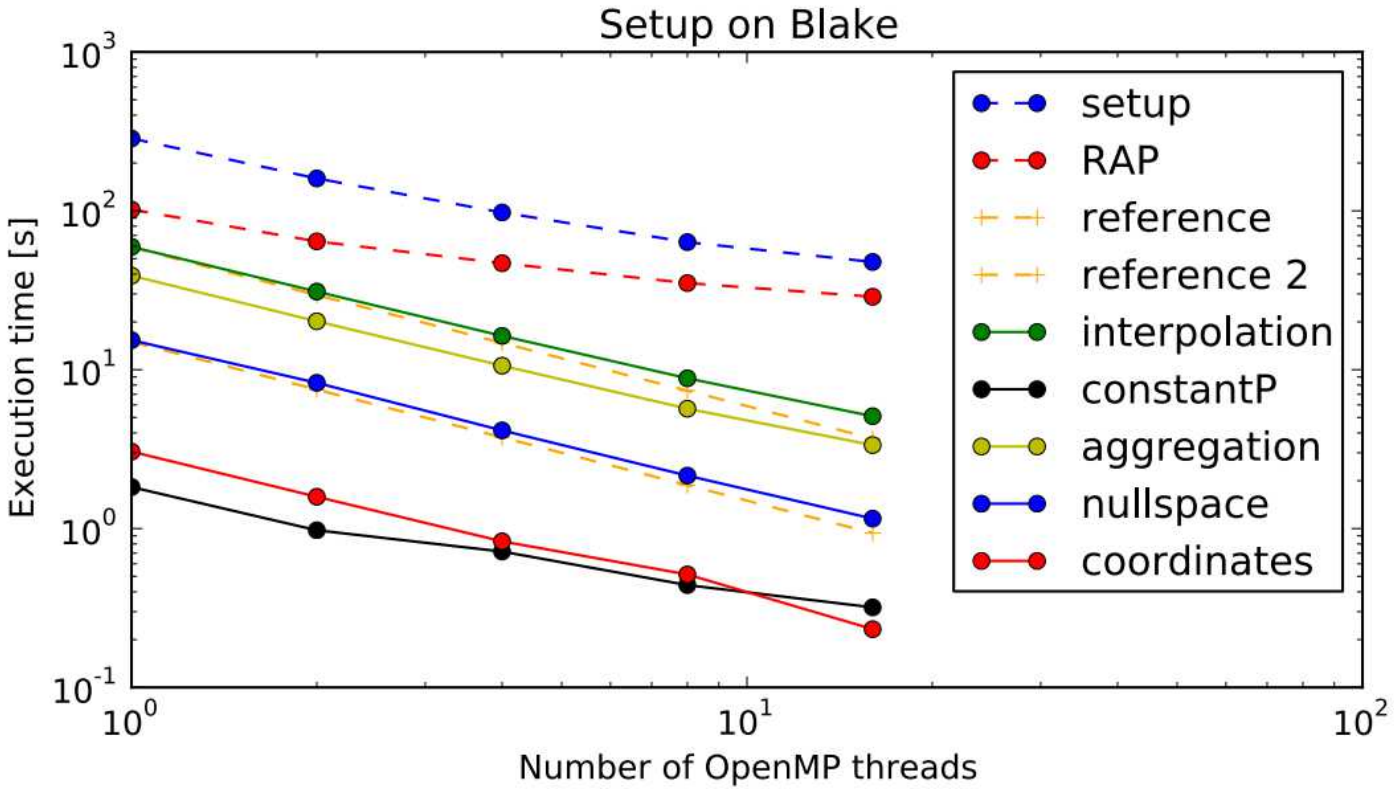
- Graph $A_c$ is known ahead of time
  1. easy to separate symbolic/numeric phases
  2. memory allocation for $A_c$ is cheap and accurate
- Stencil of $A_f$ is know
  1. fast to retrieve rows of P to compute $A_f \times P$
  2. less indirection for memory access
  3. potential for computation/communication overlap

But this kernel is complex to implement.

Plan:

1. Compute graph of $A_c$, perform "reuse" numerical phase
2. Quantify gain of specialized numerical phase

# Structured setup performance




Setup for 3D Laplacian ($200^3$ mesh) on SkyLake processor, using piece-wise constant interpolation

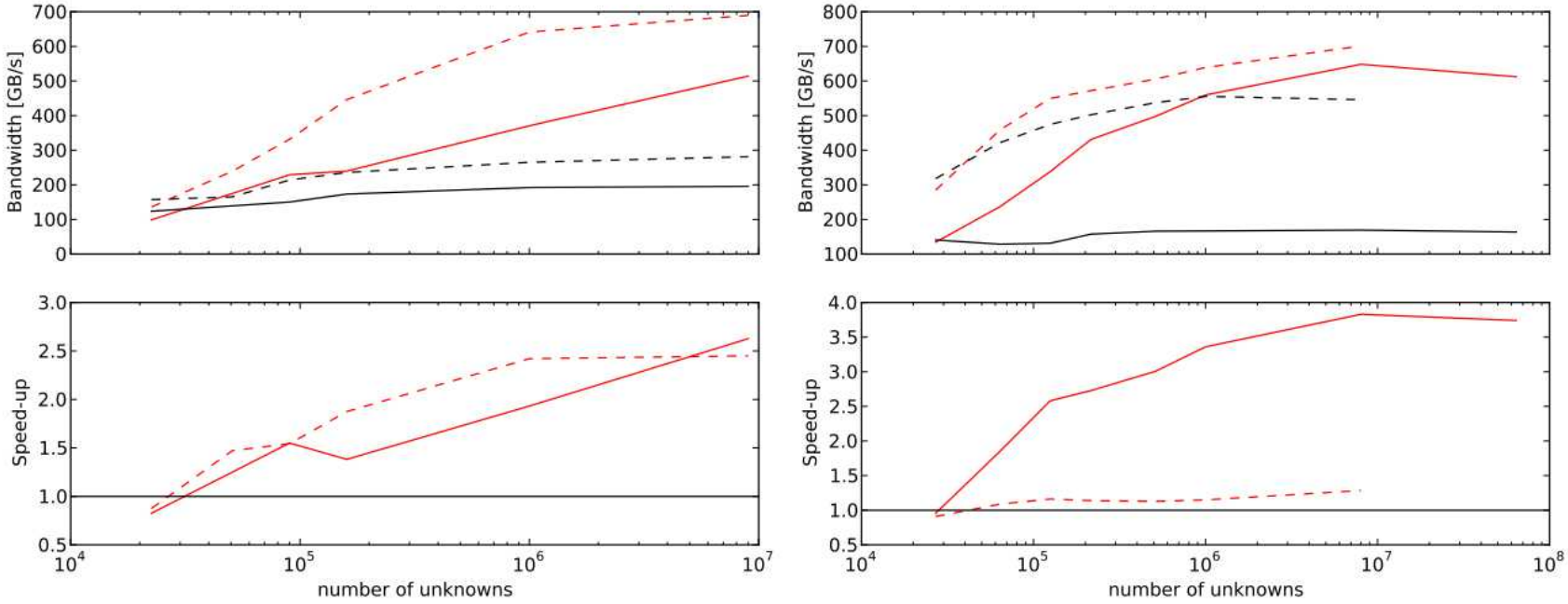
# Structured SpMV

Sandia National Laboratories

After setup, most of multigrid is SpMV + Smoothers:

- $r = b - Ax$, $x_c = R \times x_f$ and $x_f = P \times x_c$
- some smoothers are SpMV too
  - Chebyshev: $\mathrm{poly}(A) \times r$
  - Iterative ILU, krylov smoother, ...

# SpMV performance




Left: 2D stencils, solid line for FD laplacian and dashed line for FE laplacian

Right: 3D stencils, solid line for FD laplacian and dashed line for FE laplacian

# Outline

# Background

Sandia National Laboratories

Team:

Paul Lin, Christian Glusa, James Elliott,
Jonathan Hu and Chris Siefert

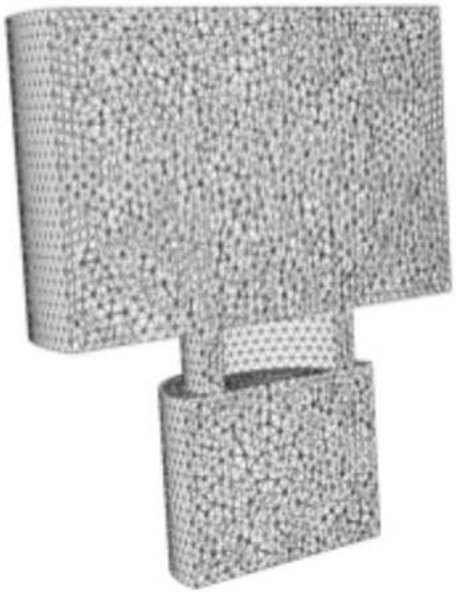
Task:

solve for the electric field in Empire (pic - plasma application)

Goal:

improve solver performance to allow 10 solves per second

# Blob test problem

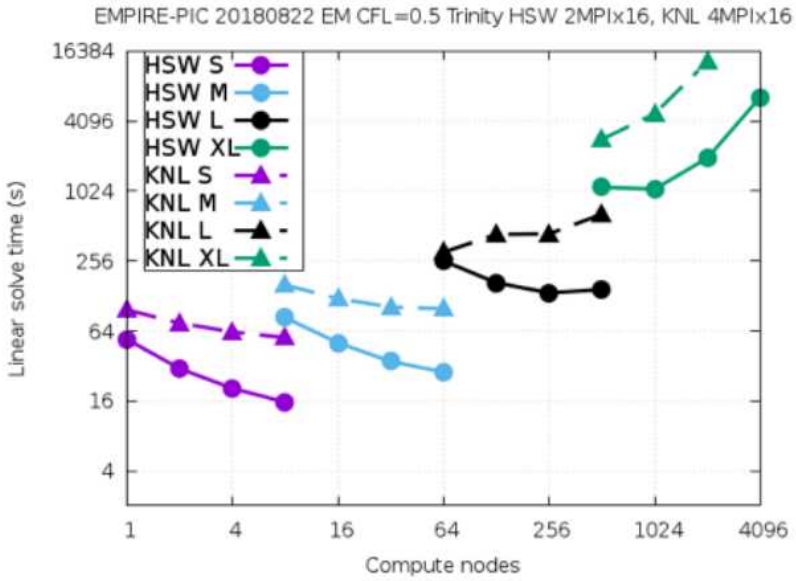


| Size | Elts | Nodes | Edges | Faces |
|---|---|---|---|---|
| S | 337k | 60k | 406k | 683k |
| M | 2.68M | 462k | 3.18M | 5.40M |
| L | 20.7M | 3.51M | 24.4M | 41.6M |
| XL | 166M | 27.9M | 195M | 333M |
| XXL | 1.33B | 223M | 1.56B | 2.67B |

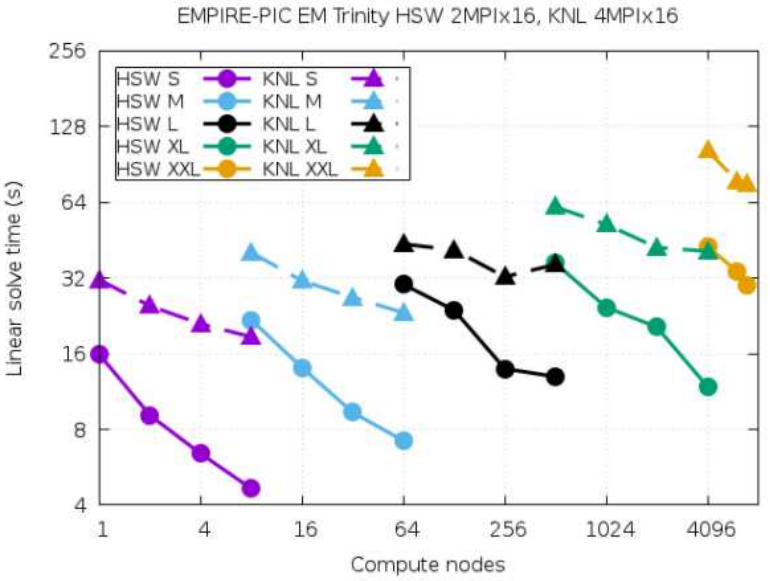
# Performance evolution

Sandia National Laboratories

## Solver performance, August 2018




## Solver performance, June 2019




Key highlights:

- Reformulate Maxwell solve as block LU solve
- MiniEM (Trilinos proxy app) for rapid prototyping and testing
- Careful analysis and optimization of compute intensive kernels
- Sustained large scale performance profiling  analysis on Trinity

**300x improvement, 6x higher CFL in ≈10 months**

# Outline

# Conclusion

Sandia National Laboratories

Structured algorithms:

- Implementation allows mix and match fo structured and unstructured algorithms
- Algorithms implemented using `Kokkos` work on many-core and GPUs
- Structured algorithm shows good scaling on many-core
- Structured SpMV improves performance up to 4x

Maxwell solver:

- running XXL mesh is now possible
- large improvements in linear solver time

# Future work

Sandia National Laboratories

Structured grid:

1. parallelize piece-wise linear interpolation
2. integrate structured SpMV with CrsMatrix
3. optimize code for GPU architecture

Maxwell solver:

1. focus on preconditioner setup cost
2. optimize algorithms for GPU architecture
3. scalable performance critical for FY20 tri-lab milestone

# References



L. Berger-Vergiat, *MueLu User's Guide*, Sandia Report SAND2019-0537


T. Wiesner, *The MueLu tutorial*, Sandia Report SAND2014-18624R