Algorithmic aspects and performance of AMG-based preconditioning for an implicit FE VMS resistive MHD model

Coupled Problems 2017

Rhodes, Greece

13 June, 2017

Paul Lin, John N. Shadid, Edward G. Phillips,
Jonathan J. Hu, Eric C. Cyr, Roger P. Pawlowski

Center for Computing Research

Sandia National Labs

Paul Tsuji

Lawrence Livermore National Laboratory

**David Sondak**

University of Texas

Need for large-scale simulations

- Goal: high-fidelity solutions of plasma physics models for large-scale problems with complex physics and geometry
  - magnetohydrodynamic (MHD), multifluid electromagnetic plasma simulations
  - US DOE interest: pulsed fusion reactors (e.g. z-pinch), magnetically confined fusion (e.g. ITER tokamak)
- Fast and robust solution methods for multiphysics problems
  - e.g. Newton-Krylov approaches
  - Block and fully-coupled preconditioners for multiphysics
  - Scalable solution methods: multilevel/multigrid
- "Next generation" platforms (manycore, accelerators)
- Talk focus: our AMG-based preconditioned approach for large-scale FEM simulations
  - Drekar CFD/MHD application code (resistive MHD, recent capability: multifluid plasmas)
  - Trilinos solvers
  - Brief discussion on matrix assembly for next gen platforms

U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration

Sandia National Laboratories

Drekar Implicit/IMEX FE Application

(J. Shadid, R. Pawlowski, E. Cyr, T. Smith, E. Phillips, T. Wildey, D. Sondak, M. Bettencourt, et al.)
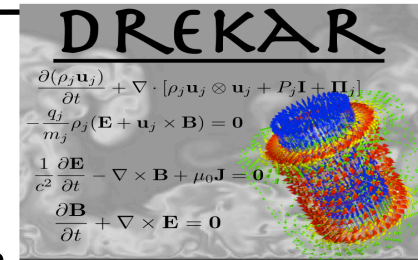
- Navier-Stokes, LES, RANS, MHD
- stabilized FEM, unstructured hexahedral meshes
- Newton-Krylov solvers with AMG-based preconditioners
- Drekar and Trilinos path forward for future architectures: Kokkos (C. Edwards, C. Trott, D. Sunderland, N. Ellingwood, S. Hammond; not part of this talk)
- Matrix assembly
  - Panzer and phalanx have been refactored to use Kokkos
  - Intrepid2 with Kokkos for discretization
- Solve
  - Most of Tpetra has been refactored to use Kokkos
  - Newton-Krylov: fully-coupled multigrid, physics-based block preconditioners
    - block preconditioners with Teko: E. Phillips talk
  - many solver packages in the process of being refactored to Kokkos
  - refactor not yet complete---results for this talk for MPI-only
    - good scaling with $O(10^5)$ MPI processes is still critical

U.S. DEPARTMENT OF ENERGY | NNSA National Nuclear Security Administration

Sandia National Laboratories

Brief Trilinos overview (solver library portion)

- Classic Trilinos (Epetra-based) (Heroux et al.):
  - Limited by 32-bit integer global objects
  - Most packages employ flat MPI-only; future architectures?
- "Next-gen" or "second generation" Trilinos solver stack (Tpetra-based):
  - No 32-bit limitation on globals (employs C++ templated data types)
  - Path forward for future architectures: Kokkos (Edwards, Trott, Sunderland, Ellingwood, Hammond; not part of this talk)

| Functionality	Classic stack	Newer solver stack
Distributed linear algebra	Epetra	Tpetra (Hoemmen,Trott, etc.)
Iterative linear solve	Aztec	Belos (Thornquist,Hoemmen,etc.)
Incomplete factor	Aztec, Ifpack	Ifpack2 (Hoemmen,Hu,Siefert, etc.)
Algebraic multigrid	ML	MueLu (Hu,Prokopenko,Wiesner,Siefert,Tuminaro,etc.)
Partition & load balance	Zoltan	Zoltan2 (Devine,Boman,Rajamanickam,Wolf,etc.)
Direct solve interface	Amesos	Amesos2 (Rajamanickam,etc.)

- PETSc is another well-known solvers library (ANL; Smith, Gropp, Knepley, Brown, McInnes, Balay, Zhang, et al.); 2015 SIAM/ACM CSE prize winner

# Trilinos MueLu Library: algebraic multigrid preconditioners

(J. Hu, A. Prokopenko, J. Gaidamour, T. Wiesner, C. Siefert, R. Tuminaro)

- "Next-gen" ML (R. Tuminaro, J. Hu, C. Tong, M. Gee, M. Sala, C. Siefert)
- Smoothed aggregation (Vanek,Mandel,Brezina 1996)
    - Create graph where vertices are block nonzeros in matrix A_k
    - Edge between vertices i and j added if block $B_k(i,j)$ contains nonzeros
- Uncoupled aggregation; prolongation/restriction; nonsmoothed; $A_{k-1} = R_k A_k P_k$
- Repartition coarser level matrices (MueLu+Zoltan2) to reduce communication
- Coarsest level: serial direct solve (KLU; T. Davis) on 1 MPI process

Other approaches: LLNL Hypre (R. Falgout, U. Yang, T. Kolev, A. Baker, E. Chow, C. Tong, et al.), MLBDDC (S. Badia, A. Martín, J. Principe, et al.), etc.

Weak Scaling: Linear Iterations (Ha=2.5)
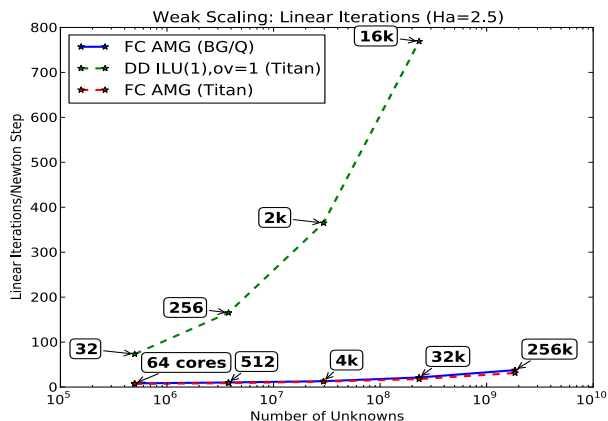
- Weak scaling: MHD generator
- Re = 500, Re_m = 1, Ha = 2.5
- Cray XK7, IBM Blue Gene/Q
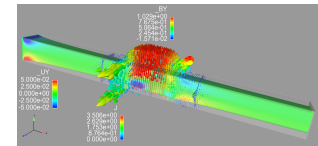
Additive Schwarz domain decomposition does not scale
Multigrid critical for performance and scaling

MueLu strong scaling: Poisson equation

(with J. Hu, J. Shadid, A. Prokopenko, E. Cyr, R. Pawlowski)

- 3D Poisson (1 DOF/mesh node); Drekar

- Simple cube geometry, near uniform mesh
- Fixed problem size (2.4B DOFs); 1 MPI task/core BG/Q

(Image courtesy LLNL)

- Optimal iteration count to 1.6 million cores (full-scale Sequoia BG/Q)

MPI	CG iterations	Solve t (s)	MG setup (s)	DOFs/MPI
131,072	6.3	1.17	7.67	~18,800
262,144	6.0	1.08	12.35	~9400
524,288	6.3	1	25.43	~4700
1,048,576	7.3	0.91	53.04	~2400
1,572,864	7.0	0.94	128.9	~1500

- MPI+X has potential to help (for Trilinos, X=Kokkos)
  - 100k MPI + 16 OpenMP would delay suboptimal scaling
  - 1.6M MPI processes ➡ MPI takes a lot of memory

Resistive (incompressible) MHD model

(J. Shadid, R. Pawlowski, E. Cyr, L. Chacon)

Navier-Stokes + Magnetic Induction

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) - \nabla \cdot (\mathbf{T} + \boxed{\mathbf{T_M}}) - \rho \mathbf{g} = 0$$

$$\mathbf{T} = -(P + \frac{2}{3}\mu(\nabla \cdot \mathbf{u}))\mathbf{I} + \mu[\nabla \mathbf{u} + \nabla \mathbf{u}^T]$$

$$\boxed{\mathbf{T_M} = \frac{1}{\mu_0}\mathbf{B} \otimes \mathbf{B} - \frac{1}{2\mu_0}\|\mathbf{B}\|^2 \mathbf{I}}$$
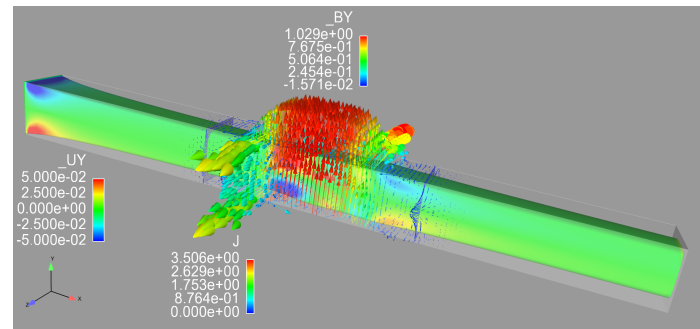
$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\rho C_p \left[\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right] + \nabla \cdot \mathbf{q} - \boxed{\eta \|\mathbf{J}\|^2} = 0$$
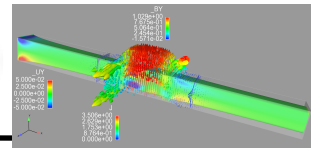
$$\boxed{\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{u} \times \mathbf{B}) + \nabla \times (\frac{\eta}{\mu_0} \nabla \times \mathbf{B}) = \mathbf{0}}$$

Steady-state MHD generator
• Flow with external cross-stream B field
• 8 DOFs/mesh node

Weak scaling: fully-coupled multigrid MHD

(with J. Shadid, J. Hu, A. Prokopenko, E. Cyr, R. Pawlowski)

| MPI	DOFs	GMRES iterations /Newton	Time/Newton step (s)		
			Multigrid setup		Solve
			Hier+smoo	Smoother	
128	845,000	14.0	12.4	11.0	4.7
1024	6,473,096	20.0	14.7	13.0	6.6
8192	50,658,056	30.8	16.9	14.2	10.1
65,536	400,799,240	53.4	20.3	16.1	17.9
524,288	3,188,616,200	98.7	45.3	19.1	40.1

- BG/Q: 1 MPI/core
- Multigrid prec setup time/Newton step
- Smoother: ILU(0) overlap=1

- Drekar 3D MHD generator on BG/Q (simple geometry)
- Algorithmic scaling challenging for nonsymmetric matrices
  - 4096x increase in size: 6.0x iterations, 7.3x time
  - Petrov-Galerkin or energy minimization approaches promising
  - Need better aggregation, better smoothers, etc.
- Another challenge: sparse matrix-matrix multiply ($A_c = R*A*P$)
- Employ reuse of construction of hierarchy + smoothers (Prokopenko, Hu)
  - Application dependent (e.g. cannot reuse for adaptive mesh)
  - Critical for transient simulations (10^4 or 10^5 time steps)

Smoothers for MHD systems

- Most MHD turbulence simulations employ spectral methods
- Stabilized FEM: variational multiscale (VMS) (Hughes 1995)
  - many authors employ VMS for CFD
- MHD/MHD turbulence with VMS (Shadid et al., Sondak+Oberai, Codina et al., Badia et al.)
- Fully-coupled Newton-Krylov (multigrid preconditioned)
- Effective smoothers critical for multigrid
  - Need to efficiently damp high frequency error
  - Relaxation not robust for our MHD problems; need ILU(0) overlap=1
- Krylov smoothers
  - Lots of previous work for SPD problems (Bank, Bornemann, Braess, Deuflhard, Douglas, Notay, Oosterlee, Shaidurov, Vassilevski, etc.)
  - Some previous work for Helmholtz (Elman, Ernst, O'Leary, etc.)
  - Far less previous work for nonsymmetric (recently: Birken, Bull, Jameson, etc.)
  - Drekar GMRESR solve preconditioned by multigrid with GMRES smoother (possibly preconditioned with e.g. block Jacobi)
  - Setup much cheaper than ILU, but solve can be expensive

Comparison of smoothers: transient Taylor-Green MHD vortex decay (VMS resistive MHD)

- Re=5100; cube domain, 20 time steps, CFL ~0.5
- linux cluster dual-socket SNB, IB fat-tree (TLCC2)

Induction
Time: 0.00
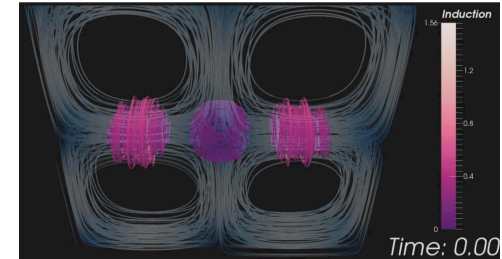
128^3 elem, 16.8M DOFs, 256 MPI

smoother		iters/dt	Prec setup(s)	Solve(s)	Prec+sol(s)	Mem(MB)
ILU(0)ov1		14.2	257	94	351	1436
GMRES	noprec	15.4	23	258	281	917
	ptGS	13.6	26	409	435	917
	bkJac	13.1	36	250	285	927
	bkGS	12.0	33	559	592	930

512^3 elem, 1.07b DOFs, 16384 MPI

| smoother		iters/dt	Prec setup(s)	Solve(s)	Prec+sol(s)	Mem(MB)
ILU(0)ov1	ov1	37.3	407	280	687	1519
GMRES	noprec	31.0	92	610	702	1002
	ptGS	21.5	94	728	822	1002
	bkJac	21.9	60	458	518	1017

- GMRES can be faster than ILU; requires less memory
- DD-GMRES smoother reduces global communication, but at a penalty of higher iteration count and higher solve times at this scale; need to go to large scales to see potential benefit

Note: have multiple Newton steps per time step

Initial weak scaling: island coalescence comparison of smoothers

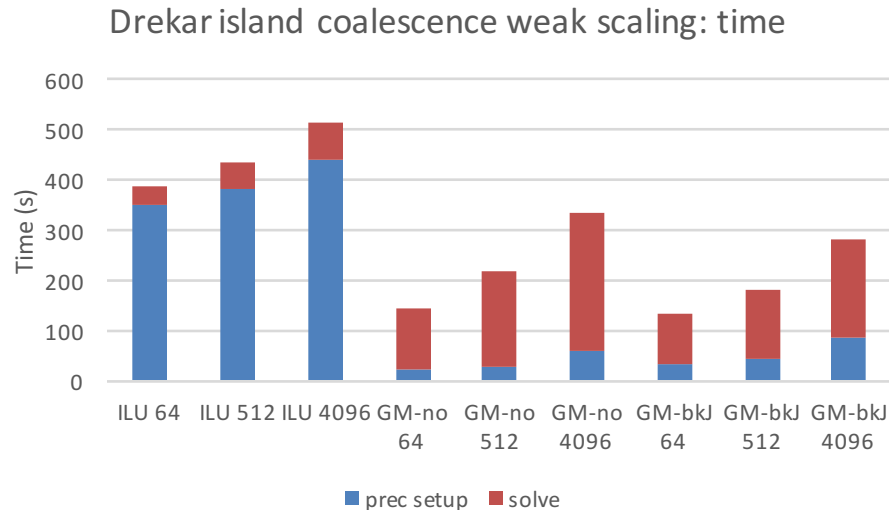## Transient resistive MHD (8 DOF/mesh node)
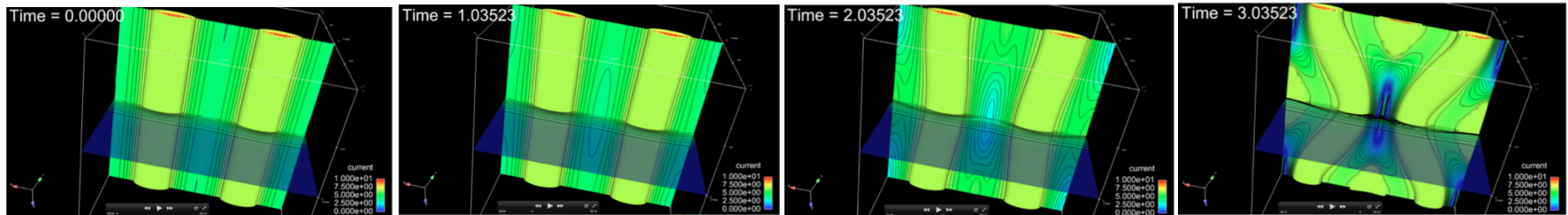



Drekar island coalescence weak scaling: time
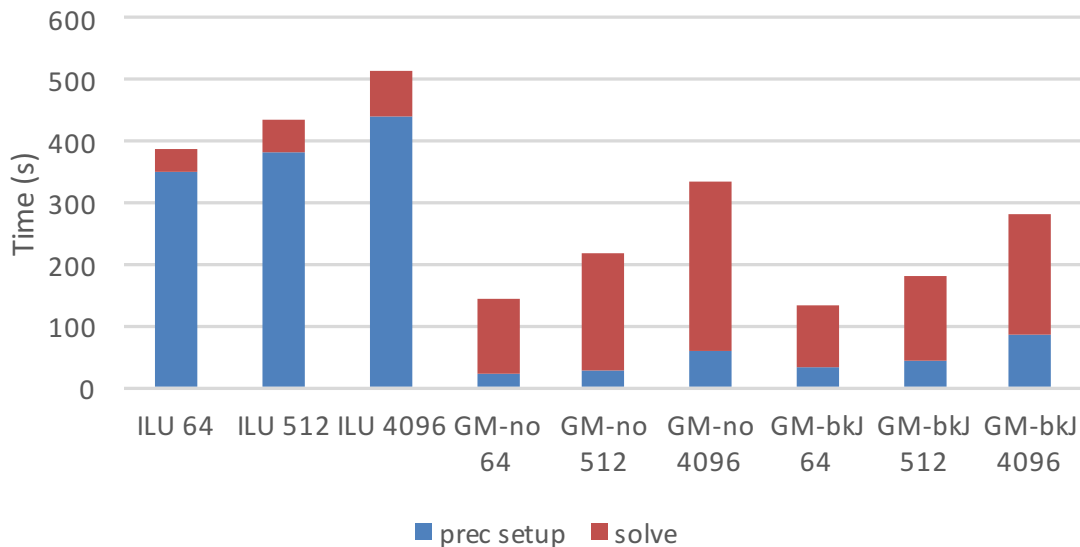
- Transient: fixed time step of 0.1 (20 steps); Smoothers: ILU, GMRES (std relaxation did not work)
- 64^3, 128^3, 256^3 elem: 2.1M, 16.9M, 135M DOF
- Dual-socket 2.6GHz SNB+IB fat-tree (TLCC2)

Initial weak scaling: island coalescence comparison of smoothers

Drekar island coalescence weak scaling: time

- Transient: fixed time step of 0.1 (20 steps); Smoothers: ILU, GMRES (std relaxation did not work)
- 64^3, 128^3, 256^3 elem: 2.1M, 16.9M, 135M DOFs
- Dual-socket 2.6GHz SNB+IB fat-tree (TLCC2)

Drekar island coalescence weak scaling: Iterations

Dreakar island coalescence weak scaling: Memory
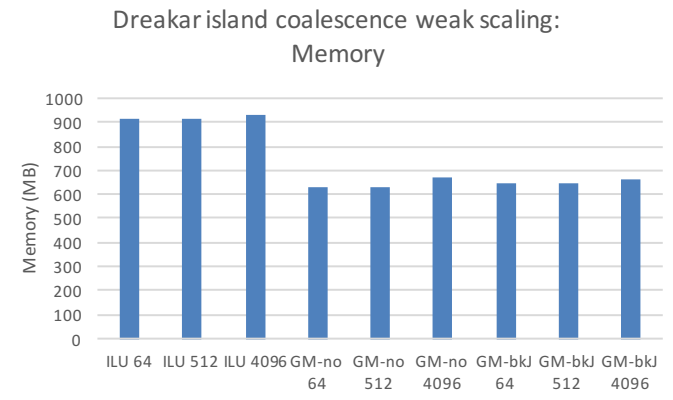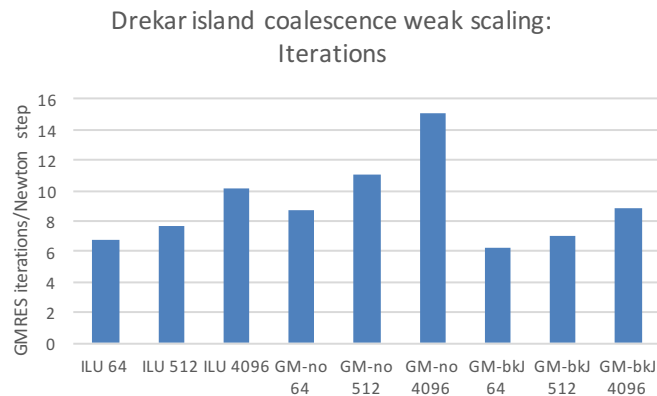
**GMRES smoother is faster than standard ILU smoother, requires less memory**

"Next generation" platforms (NGP)

- Many-core processors and accelerators
- Many large platforms around the world with many-core processors and accelerators
- We use US DOE platforms
  - NERSC8 Cori Cray XC40: ~9700 Intel Xeon Phi Knights Landing (KNL)
  - LANL/SNL Trinity ATS-1 Cray XC40
    - Phase 1 ~9400 dual-socket HSW compute nodes
    - Phase 2 ~9000 Intel Xeon Phi Knights Landing (KNL)
  - LLNL ATS-2 Sierra: IBM POWER+NVIDIA GPUs
  - ORNL Summit: IBM POWER+NVIDIA GPUs
  - ANL Aurora: Intel Xeon Phi
- DOE has many huge legacy codes
  - especially NNSA Tri-labs (LANL, LLNL, SNL)

First step towards NGP: matrix assembly
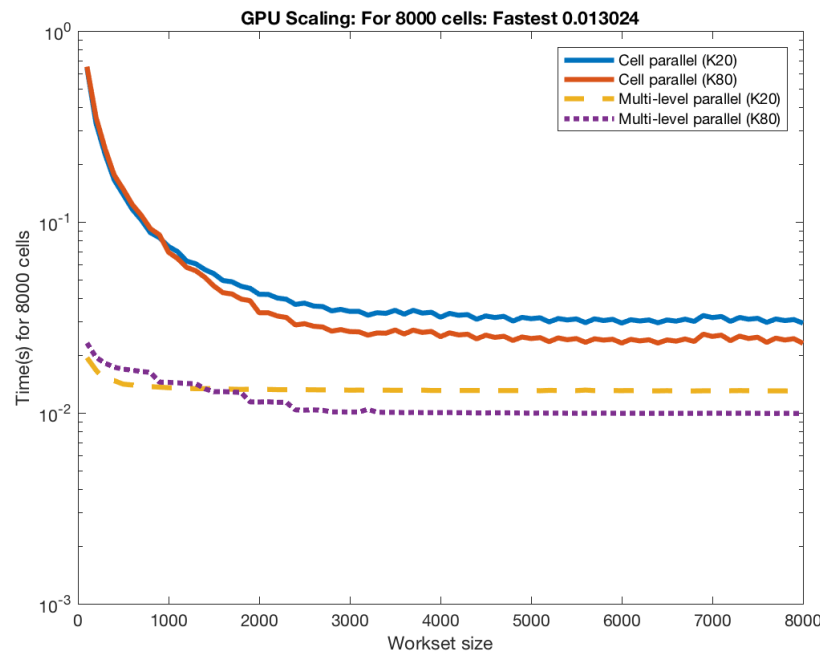## (M. Bettencourt, R. Pawlowski, E. Cyr)

- Drekar FE assembly with Kokkos

- Electron full-Maxwell plasma system simulation

- Mixed-basis assembly using nodal (fluid), edge (electric field), face (magnetic field)

- FE integration and gather/scatter kernels (local-dense to global-sparse data structures)

- Dual-socket 16-core HSW

- Single GPU (K80)

- Single KNC (224 threads)

- Single KNL (256 threads)
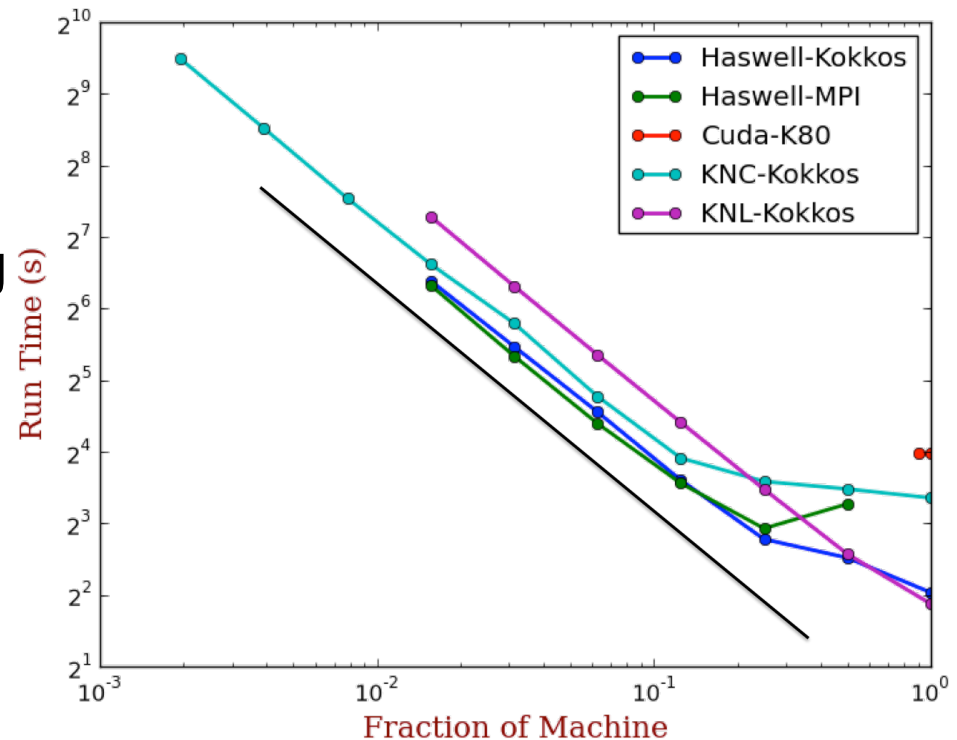
First step towards NGP: matrix assembly for GPU

(C. Trott, M. Bettencourt, R. Pawlowski, E. Cyr, E. Phipps)

- Original GPU results below expectations due to lack of parallelism in assembly kernels

- To expose more parallelism, need to extend physics kernels to hierarchical parallelism

- Impact of hierarchical parallelism of a core CFD computational kernel extracted to a standalone test (work with Christian Trott)

- Future work: hierarchical parallelism in Drekar matrix assembly

GPU Scaling: For 8000 cells: Fastest 0.013024

Legend:
- Cell parallel (K20)
- Cell parallel (K80)
- Multi-level parallel (K20)
- Multi-level parallel (K80)

Concluding remarks and future work

- Presented scaling studies for full MHD systems

- Performed a comparison of multigrid smoothers for our MHD problems

  - initial empirical study for two test cases

  - initial evaluation of GMRES as alternative smoother to current standard (ILU)

    - Shows promise: can solve an initial class of relevant problems (appears competitive; expensive, but so is ILU)

    - Memory usage benefits (ILU requires ~40% more)

  - Go to larger scale; more test cases

  - Drawback: more communication?

  - Need to go back and try to analyze method more carefully

- Kokkos for manycore and accelerators ("X" for MPI+X)

  - Preliminary threaded matrix assembly results promising

  - Lot of work remaining for threaded preconditioner setup and solve

- Many challenges for multigrid-preconditioned linear solve

  - algorithmic scaling for nonsymmetric problems

  - multigrid preconditioner setup (sparse mat-mat)

Thanks For Your Attention!

Paul Lin (ptlin@sandia.gov)

John Shadid, Edward Phillips, Jonathan Hu, Paul Tsuji,
Eric Cyr, Roger Pawlowski, David Sondak

Acknowledgments:

- Rest of MueLu team: Andrey Prokopenko, Jeremy Gaidamour, Tobias Wiesner, Chris Siefert, Ray Tuminaro

- Rest of Drekar team: Tom Smith, Tim Wildey, Matt Bettencourt, et al.

- Mark Hoemmen, Eric Phipps

- LLNL LC BG/Q team (especially John Gyllenhaal, Scott Futral, Tom Spelce, Roy Musselman)

U.S. DEPARTMENT OF **ENERGY** **NNSA**
National Nuclear Security Administration

Sandia National Laboratories

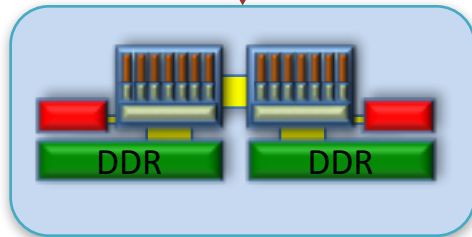# Extra slides

What is Kokkos?

LAMMPS

Albany

Drekar
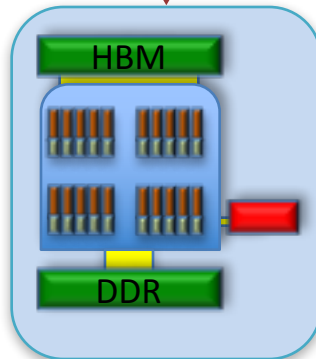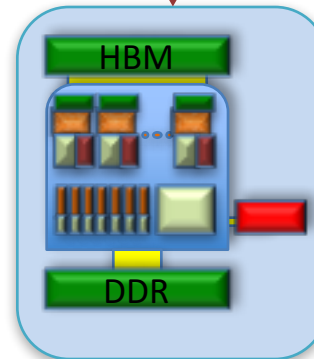
Applications & Libraries

Trilinos

Kokkos
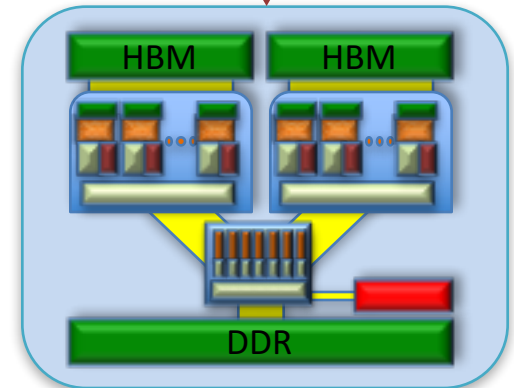performance portability for C++ applications

Multi-Core

Many-Core

APU

CPU+GPU

Cornerstone for performance portability across next generation HPC architectures at multiple DOE laboratories, and other organizations.

Slide stolen from M. Bettencourt, who probably stole it from the Kokkos team

Abstractions
Patterns, Policies, and Spaces

- **<u>Parallel Pattern</u> of user's computations**
  - parallel_for, parallel_reduce, parallel_scan, task-graph, ... *(extensible)*
- **<u>Execution Policy</u> tells *how* user computation will execute**
  - Static scheduling, dynamic scheduling, thread-teams, ... *(extensible)*
- **<u>Execution Space</u> tells *where* computations will execute**
  - Which cores, numa region, GPU, ... *(extensible)*
- **<u>Memory Space</u> tells *where* user data resides**
  - Host memory, GPU memory, high bandwidth memory, ... *(extensible)*
- **<u>Layout</u> (policy) tells *how* user array data is laid out**
  - Row-major, column-major, array-of-struct, struct-of-array … *(extensible)*
- **Differentiating: Layout and Memory Space**
  - Versus other programming models (OpenMP, OpenACC, …)
  - Critical for performance portability …

Slide stolen from M. Bettencourt, who probably stole it from the Kokkos team

Multi-fluid Plasma System Model

Density	$$\frac{\partial \rho_a}{\partial t} + \nabla \cdot (\rho_a \mathbf{u}_a) = \sum_{b \neq a} (n_a \rho_b \bar{\nu}_{ab}^+ - n_b \rho_a \bar{\nu}_{ab}^-)$$
Momentum	$$\frac{\partial (\rho_a \mathbf{u}_a)}{\partial t} + \nabla \cdot (\rho_a \mathbf{u}_a \otimes \mathbf{u}_a + p_a I + \Pi_a) = q_a n_a (\mathbf{E} + \mathbf{u}_a \times \mathbf{B})$$ $$- \sum_{b \neq a} \left[\rho_a (\mathbf{u}_a - \mathbf{u}_b) n_b \bar{\nu}_{ab}^M + \rho_b \mathbf{u}_b n_a \bar{\nu}_{ab}^+ - \rho_a \mathbf{u}_a n_b \bar{\nu}_{ab}^- \right]$$
Energy	$$\frac{\partial \varepsilon_a}{\partial t} + \nabla \cdot ((\varepsilon_a + p_a) \mathbf{u}_a + \Pi_a \cdot \mathbf{u}_a + \mathbf{h}_a) = q_a n_a \mathbf{u}_a \cdot \mathbf{E} + Q_a^{src}$$ $$- \sum_{b \neq a} \left[(T_a - T_b) k \bar{\nu}_{ab}^E - \rho_a \mathbf{u}_a \cdot (\mathbf{u}_a - \mathbf{u}_b) n_b \bar{\nu}_{ab}^M - n_a \bar{\nu}_{ab}^+ \varepsilon_b + n_b \bar{\nu}_{ab}^- \varepsilon_a \right]$$
Charge and Current Density	$$q = \sum_k q_k n_k \qquad \mathbf{J} = \sum_k q_k n_k \mathbf{u}_k$$
Maxwell's Equations	$$\frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{B} + \mu_0 \mathbf{J} = \mathbf{0} \qquad \nabla \cdot \mathbf{E} = \frac{q}{\epsilon_0}$$ $$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = \mathbf{0} \qquad \nabla \cdot \mathbf{B} = 0$$

Nodal discretization for fluids, compatible discretization for electromagnetics
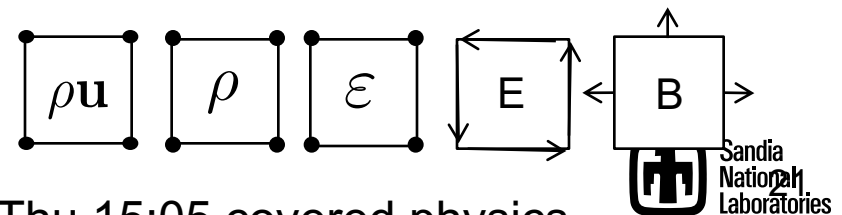
J. Shadid talk Thu 15:05 covered physics

Fully discrete 2-fluid system

$$\begin{bmatrix} \mathbf{D}_{\rho_i} & \mathbf{K}^{\rho_i}_{\rho_i u_i} & 0 & \mathbf{Q}^{\rho_i}_{\rho_e} & 0 & 0 & 0 & 0 \\ \mathbf{D}^{\rho_i u_i}_{\rho_i} & \mathbf{D}_{\rho_i u_i} & 0 & \mathbf{Q}^{\rho_i u_i}_{\rho_e} & \mathbf{Q}^{\rho_i u_i}_{\rho_e u_e} & 0 & \mathbf{Q}^{\rho_i u_i}_{E} & \mathbf{Q}^{\rho_i u_i}_{B} \\ \mathbf{D}^{\mathcal{E}_i}_{\rho_i} & \mathbf{D}^{\mathcal{E}_i}_{\rho_i u_i} & \mathbf{D}_{\mathcal{E}_i} & \mathbf{Q}^{\mathcal{E}_i}_{\rho_e} & \mathbf{Q}^{\mathcal{E}_i}_{\rho_e u_e} & \mathbf{Q}^{\mathcal{E}_i}_{\mathcal{E}_e} & \mathbf{Q}^{\mathcal{E}_i}_{E} & 0 \\ \mathbf{Q}^{\rho_e}_{\rho_i} & 0 & 0 & \mathbf{D}_{\rho_e} & \mathbf{K}^{\rho_e}_{\rho_e u_e} & 0 & 0 & 0 \\ \mathbf{Q}^{\rho_e u_e}_{\rho_i} & \mathbf{Q}^{\rho_e u_e}_{\rho_i u_i} & 0 & \mathbf{D}^{\rho_e u_e}_{\rho_e} & \mathbf{D}_{\rho_e u_e} & 0 & \mathbf{Q}^{\rho_e u_e}_{E} & \mathbf{Q}^{\rho_e u_e}_{B} \\ \mathbf{Q}^{\mathcal{E}_e}_{\rho_i} & \mathbf{Q}^{\mathcal{E}_e}_{\rho_i u_i} & \mathbf{Q}^{\mathcal{E}_e}_{\mathcal{E}_i} & \mathbf{D}^{\mathcal{E}_e}_{\rho_e} & \mathbf{D}^{\mathcal{E}_e}_{\rho_e u_e} & \mathbf{D}_{\mathcal{E}_e} & \mathbf{Q}^{\mathcal{E}_e}_{E} & 0 \\ 0 & \mathbf{Q}^{E}_{\rho_i u_i} & 0 & 0 & \mathbf{Q}^{E}_{\rho_e u_e} & 0 & \mathbf{Q}_E & \mathbf{K}^{E}_{B} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{K}^{B}_{E} & \mathbf{Q}_B \end{bmatrix} \begin{bmatrix} \rho_i \\ \rho_i \mathbf{u}_i \\ \mathcal{E}_i \\ \rho_e \\ \rho_e \mathbf{u}_e \\ \mathcal{E}_e \\ \mathbf{E} \\ \mathbf{B} \end{bmatrix}$$

16 Coupled Nonlinear PDEs (continuity, momentum, energy for ions+electrons; Maxwell equations)

Group the hydrodynamic variables together (similar discretization)

$$\mathbf{F} = (\rho_i, \rho_i \mathbf{u}_i, \mathcal{E}_i, \rho_e, \rho_e \mathbf{u}_e, \mathcal{E}_e)$$

Resulting 3x3 block system

$$\begin{bmatrix} \mathbf{D}_F & \mathbf{Q}^F_E & \mathbf{Q}^F_B \\ \mathbf{Q}^E_F & \mathbf{Q}_E & \mathbf{K}^E_B \\ 0 & \mathbf{K}^B_E & \mathbf{Q}_B \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{E} \\ \mathbf{B} \end{bmatrix}$$

Reordered 3x3

$$\begin{bmatrix} \mathbf{Q}_B & \mathbf{K}^B_E & 0 \\ \mathbf{K}^E_B & \mathbf{Q}_E & \mathbf{Q}^E_F \\ \mathbf{Q}^F_B & \mathbf{Q}^F_E & \mathbf{D}_F \end{bmatrix} \begin{bmatrix} \mathbf{B} \\ \mathbf{E} \\ \mathbf{F} \end{bmatrix}$$

Physics-based Approach Enables Optimal AMG Sub-block Solvers

Use upper triangular factor of block LU decomposition as preconditioner

$$P = \begin{bmatrix} \mathbf{Q}_B & \mathbf{K}_E^B & 0 \\ 0 & \hat{\mathcal{D}}_E & \mathbf{Q}_F^E \\ 0 & 0 & \hat{\mathbf{S}}_F \end{bmatrix}$$

2 Schur complements to solve

$$\hat{\mathbf{S}}_F = \mathbf{D}_F - \mathcal{K}_E^F \mathcal{D}_E^{-1} Q_F^E$$

Fluid sub-solve: Node-based coupled CFD type system.

SIMPLEC approximation for S_F

$$\hat{\mathcal{D}}_E = \mathbf{Q}_E - \mathbf{K}_B^E \bar{\mathbf{Q}}_B^{-1} \mathbf{K}_E^B$$

Maxwell subsystem: electric field Edge-based curl-curl type system.

$$\mathbf{B} = -\bar{\mathbf{Q}}_B^{-1} \mathbf{K}_E^B \, \mathbf{E}$$

Magnetic field: Face-based simple mass matrix Inversion.

Next: Results for Maxwell subsystem; Large-scale plasma fluid subsystem study is future work

Weak Scaling for Maxwell Subsystem

- 3D edge-based curl-curl E-B Maxwell subsystem: ML H(curl) AMG

- 3 sub-blocks (2 with rows = #edges); MueLu on sub-blocks

- cubic domain with cubic elements

- 10 time steps; total 20 linear solves (20 Teko/MueLu prec setup); Cray XC40

Drekar Tpetra/Teko/MueLu E-B Maxwell weak scaling

Teko 8.4B row matrices
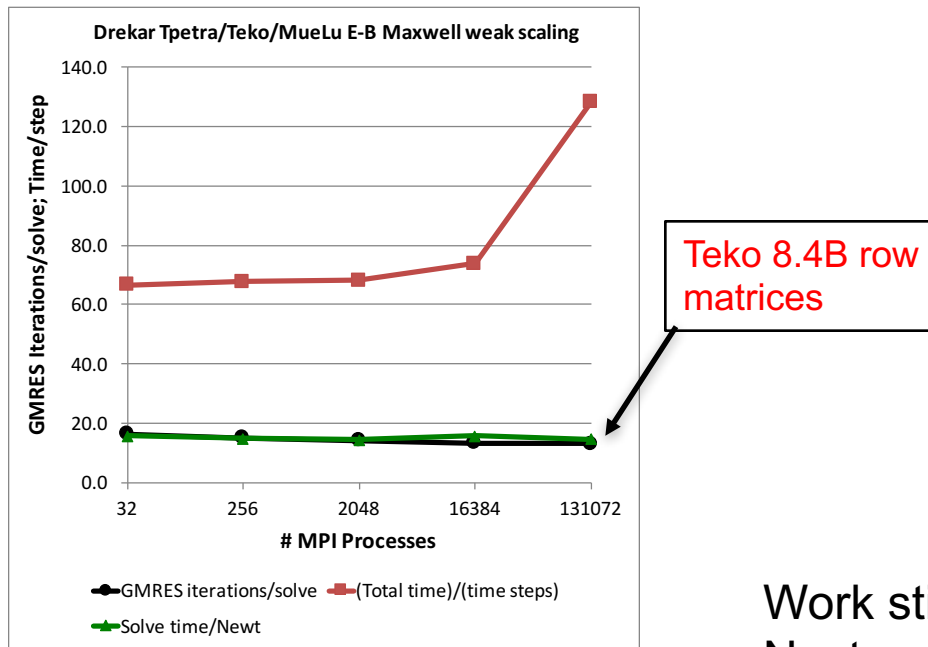
- Good scaling on block solves (at least for solve; setup needs improvement)
- Formulation presented Wed 13:55 E. Phillips

Work still in progress: multifluid results
Next we revisit single fluid full MHD systems

"Next generation" platforms (NGP)

- Many-core processors and accelerators
- ACES
  - NERSC8 Cori
    - ~9500 Intel Xeon Phi Knights Landing (KNL)
  - ACES Trinity ATS-1
    - Phase 1 ~9500 dual-socket HSW compute nodes
    - Phase 2 ~9500 Intel Xeon Phi Knights Landing (KNL)
      - ~68 cores
- CORAL
  - LLNL ATS-2 Sierra
    - IBM POWER+NVIDIA GPUs
  - ORNL Summit
    - IBM POWER+NVIDIA GPUs
  - ARNL Aurora
    - Intel Xeon Phi
- DOE has many huge legacy codes
  - especially NNSA Tri-labs (LANL, LLNL, SNL)