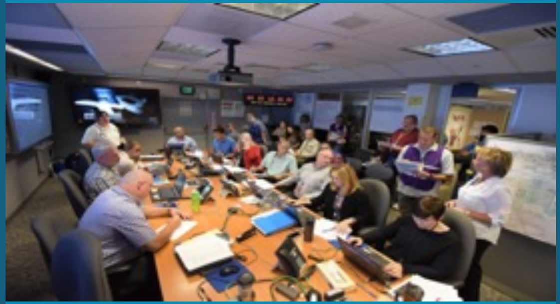
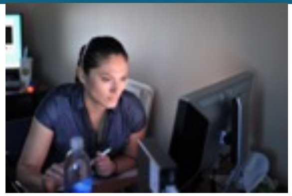
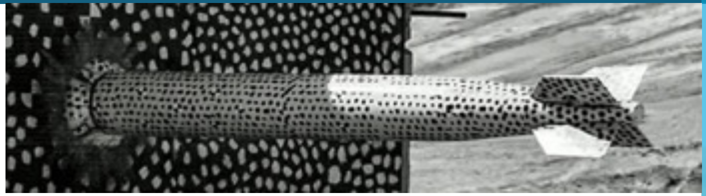
Performance of Aria running on ATS-2







PRESENTED BY

Jonathan Clausen

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.

SAND2022-14973 PE

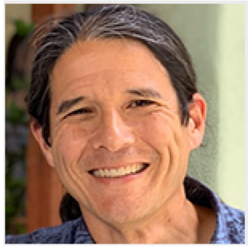
Thermal Fluids Development Team

COMPSIM
THERMAL FLUIDS

Jon Clausen (Product Owner)
Zu Tejan-Kella (Scrum Master)

Victor Brunini
Lincoln Collins
Mike Hansen
Robert Knaus
Alec Kucala
Stephen Lin
Neil Matula
Dan Moser
Malachi Phillips
Tom Ransegnola
Phil Sakievich
Sam Subia
Yaro Vasyliv
Tyler Voskuilen

Trilinos and Kokkos Kernels Milestone Team

Ichi Yamazaki and Jonathan Hu: Trilinos linear solvers and Ifpack2

Evan Harvey: Kokkos Kernels support

Brian Kelley: Trilinos Tpetra and Kokkos Kernels support

Jennifer Loe: Trilinos linear solvers and Kokkos Kernels ODEs

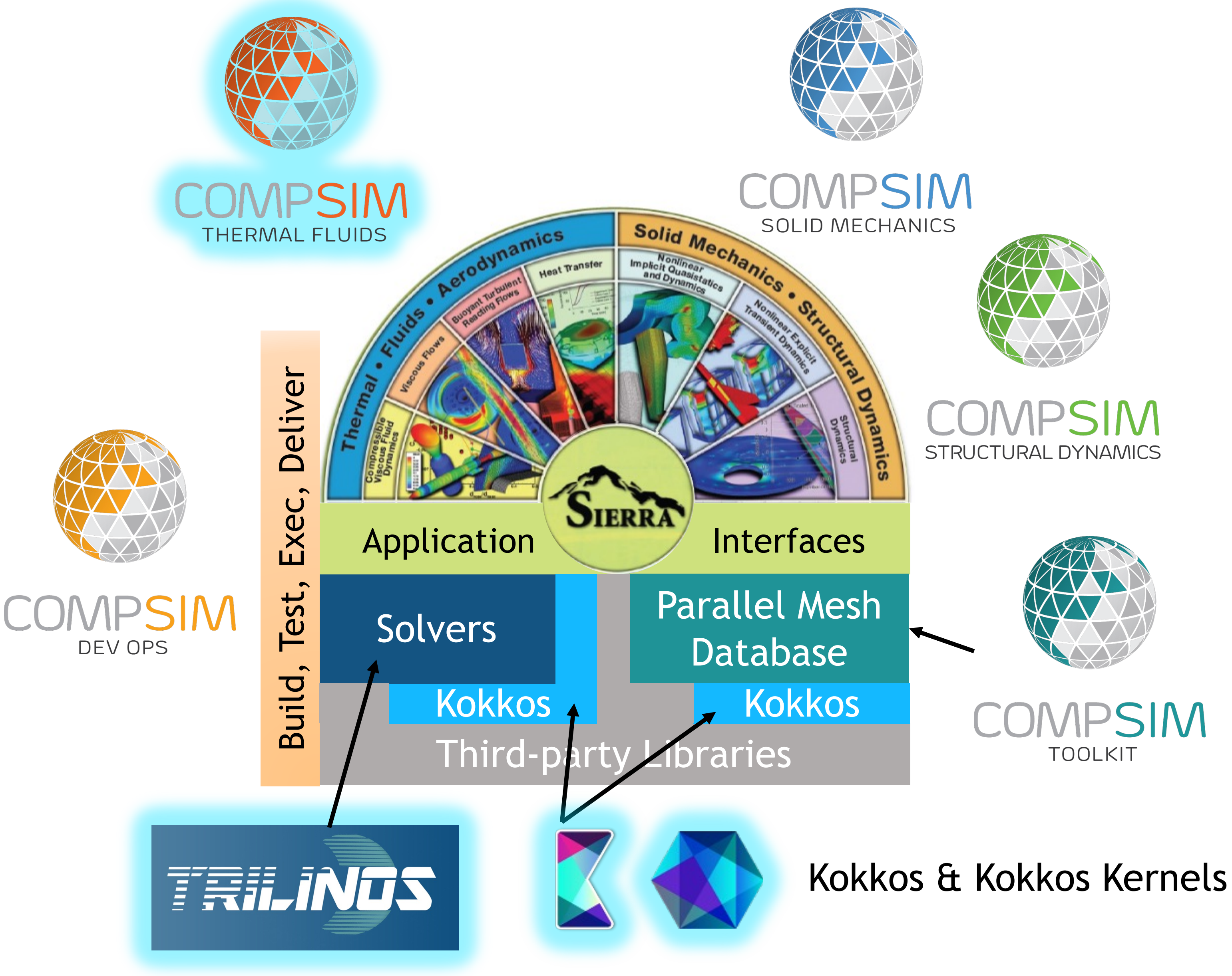
Jim Foucar: Kokkos Kernels ParILUT

Curt Ober: Trilinos and ODE support


Nathan Ellingwood and Vinh Dang: Kokkos Kernels ILU(k) and SpTRSV


Luc Berger-Vergiat and Siva Rajamanickam: Kokkos Kernels leads

SIERRA Mechanics Overview



Background

Thermal Simulations: key physics

- conduction
- convection/prescribed external BCs
- enclosure radiation
- contact
- chemistry (ODE)
- pressurization regions
- level set burn front models

ATS-2

- 3.1 GHz Power9 (44 cores/node)
- 4 NVIDIA V100 GPU/node
- 256 GB CPU per node, 64 GB GPU per node
- 4320 GPU nodes

CTS-1

- 2.1 GHz Intel Broadwell E5-2695 (2 socket, 18 core)
- 128 GB per node
- 1488 nodes (eclipse)

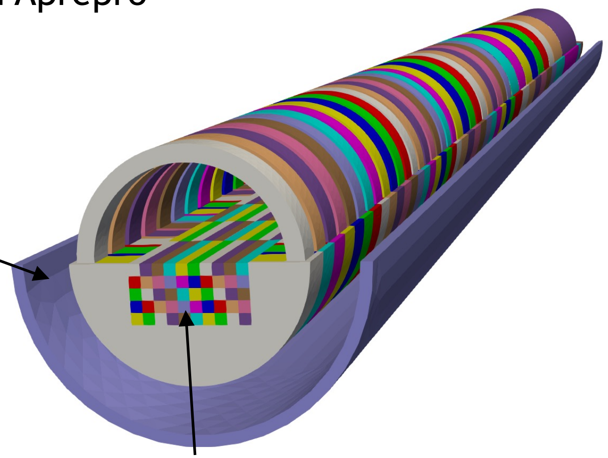
CTS-1

ATS-2

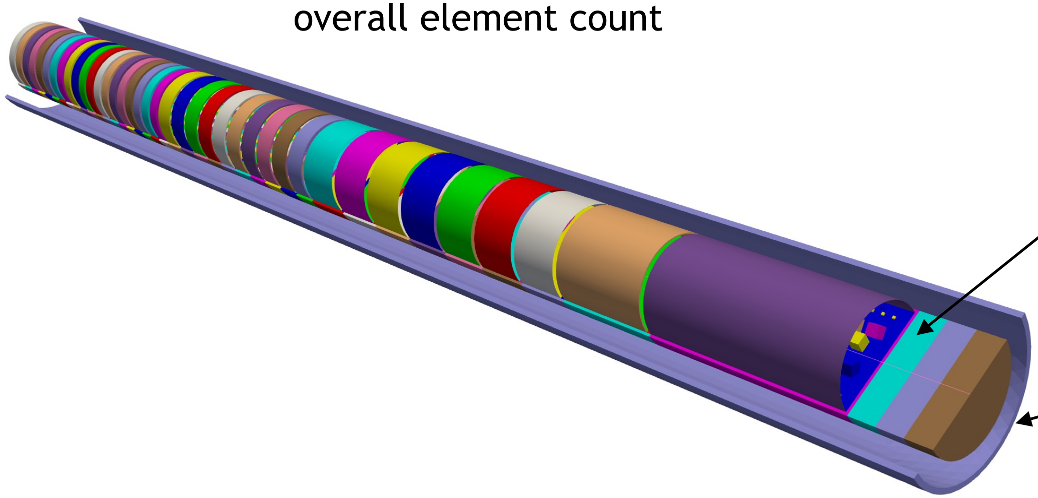
Thermal surrogate geometry

Fully parameterized geometry via Cubit journal file and Aprepro

Annular partial enclosure



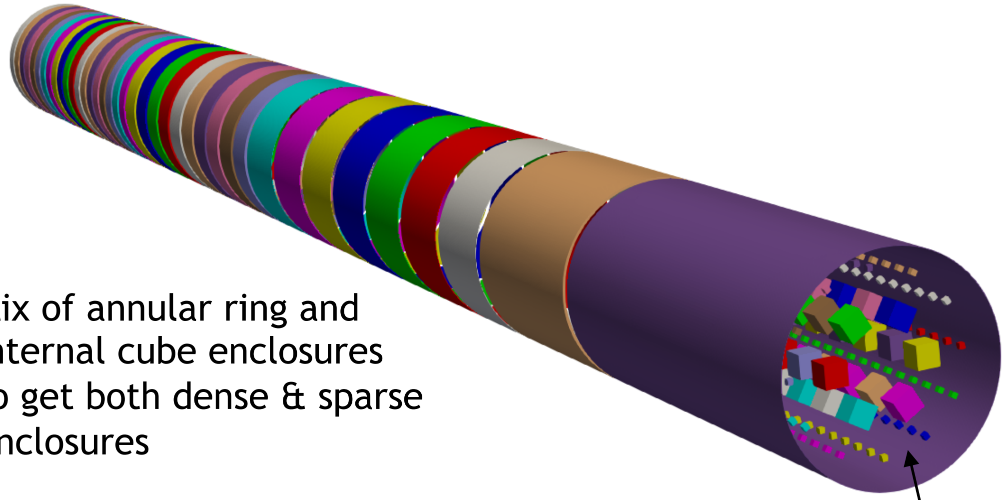
Cut out rectangle and partitioned into 1200 blocks (10x10x12), mesh size on these blocks adjusted to set overall element count



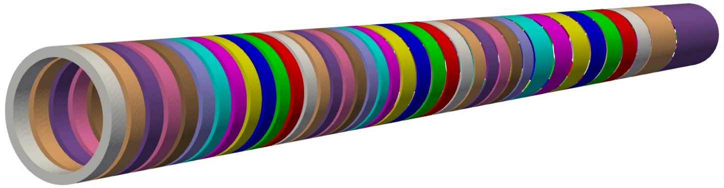
~200k element ODE solve (chemistry)

Extra discs on the end tied with contact

Enclosures



Mix of annular ring and internal cube enclosures to get both dense & sparse enclosures



contains composite material decomposition

Abnormal surrogate test characteristics

Acceptance Test	Surrogate
7M tet elements	7.6M tet elements
1250 blocks	1438 blocks
131 side sets	307 side sets
80 bulk nodes & 7 pressurization zones	39 bulk nodes and pressurization zones
80 enclosures 105k, 1.4% dense 8 between 10k-26k, ~5-15% dense other smaller ones	74 enclosures 133k, 2.4% dense 106k, 1.6% dense 7x 21k, 3.9% dense other smaller ones 1 partial
14 foam blocks, 1.75M elements	50 foam blocks, 1.8M elements
18 EM blocks, 580k elements, some burn front models	10 EM blocks, 600k elements, 300k elements with burn front
3 contact surfaces (7k, 8k, 1.7k faces in contact, one search has 800k faces but most without matches)	Yes, (7.7k, 8k, 3.5k faces in contact, one has 150k faces in search)

Milestone results

Physics porting

- chemistry expressions
- tftk::chemistry library
- tftk::ode (explicit RK for now)
- user-defined string functions
- burn front models
- composite burn models
- level set expressions
- phase support within expressions

Surrogate and acceptance test

- equivalent execution verified
- acceptance test is full length (all physics)

Performance Results

- Rebalance (on or off)
- ODE solver (explicit, implicit in FY23)
- Solver and Preconditioner combo
  - GMRES + RILUK
  - GMRES + FastILU
  - BICGSTAB + Jacobi

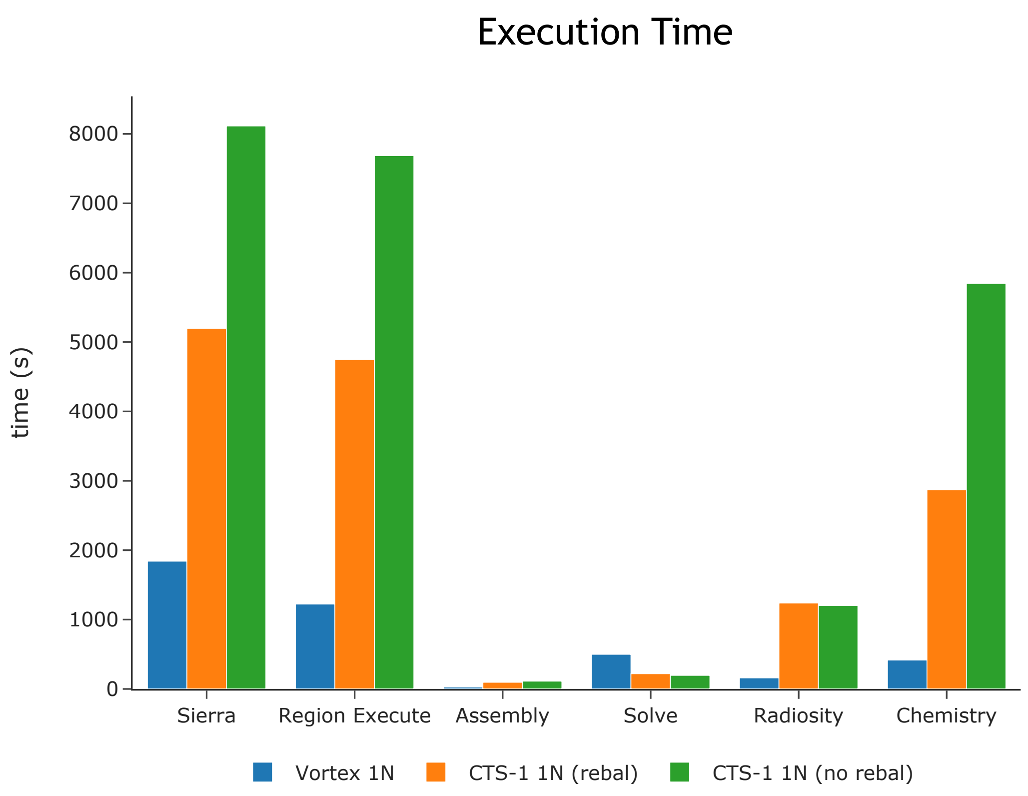
Timers

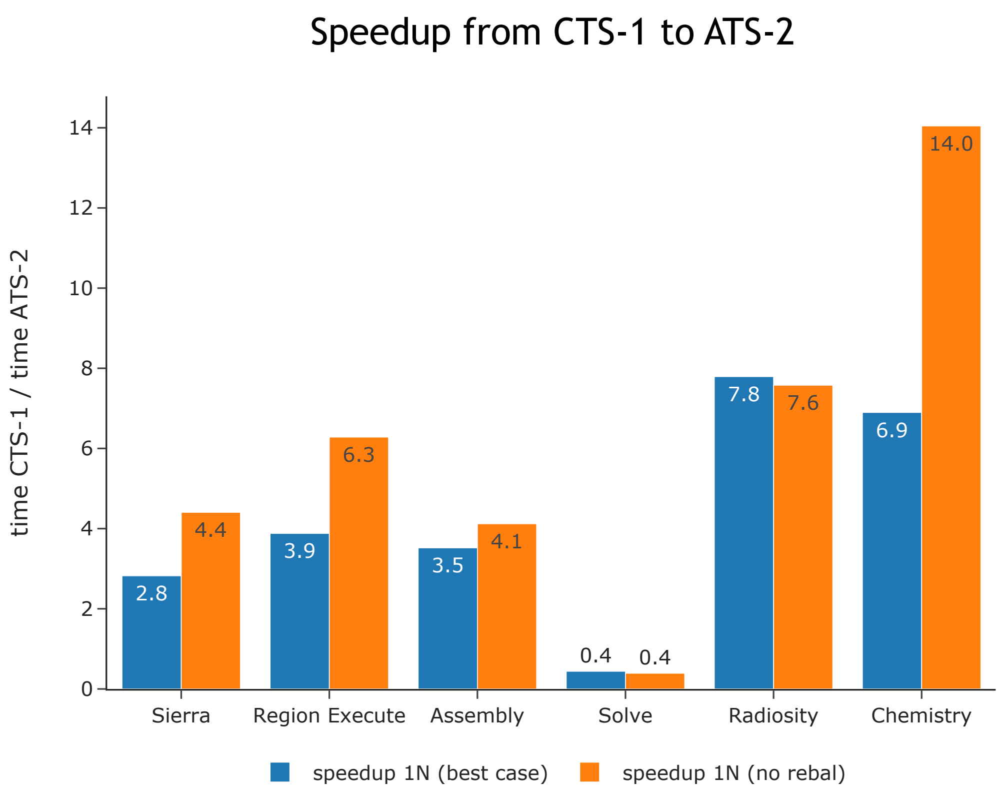
- Sierra
- Region Execute
  - Assembly
  - Solve
  - Radiosity
  - Chemistry

Summary of Solvers Team Accomplishments for ATS2

- BiCGStab + Jacobi
  - **3x speedup** over corresponding CTS1 preconditioner
- Standard RILU(3) + KKSpTRSV
  - **12.5x speedup** on ATS2 from beginning of milestone.
  - **1.4x speedup** over best ILU-type CTS1 preconditioner.
- FastILU + FastSpTRSV
  - Fixed GPU build problems to make this option available to ARIA.
  - **16x speedup** over prior best ILU-type preconditioner.
  - **1.8x speedup** over best ILU-type CTS1 preconditioner.
- GPU-capable **ODE solvers** near completion (explicit and implicit options).
- Improved Trilinos **performance monitoring**- including Sierra test problems.
- Future: New ILU preconditioner to more closely emulate CTS1 preconditioner properties.

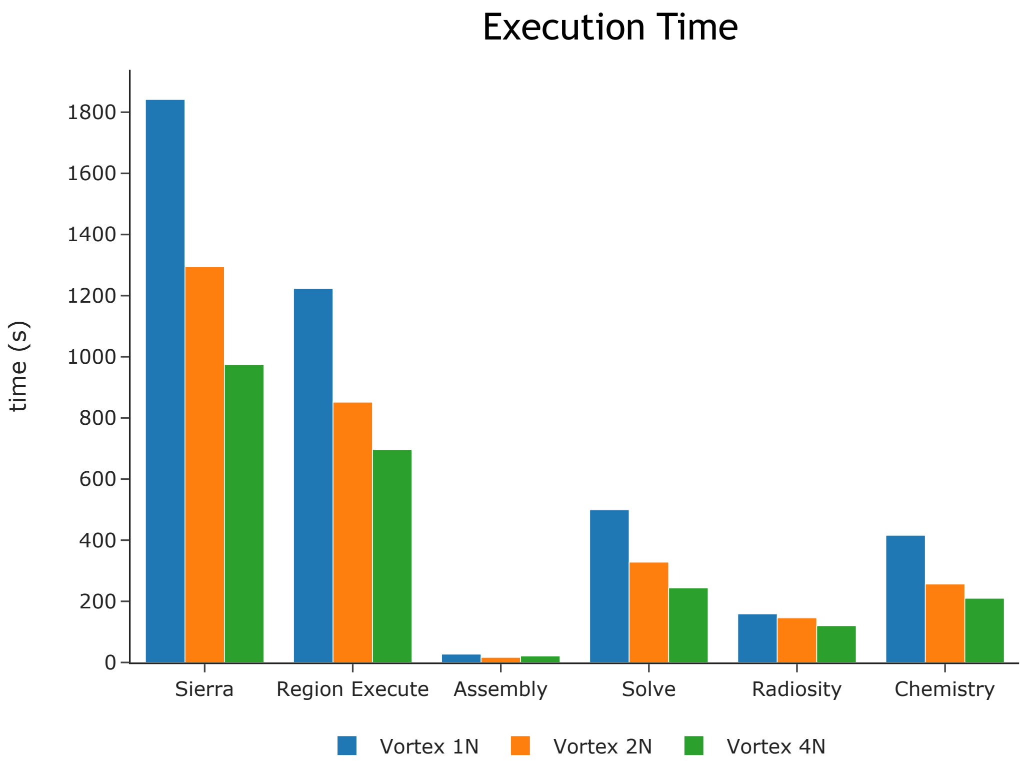
Abnormal surrogate performance: Single node, ILU0+GMRES

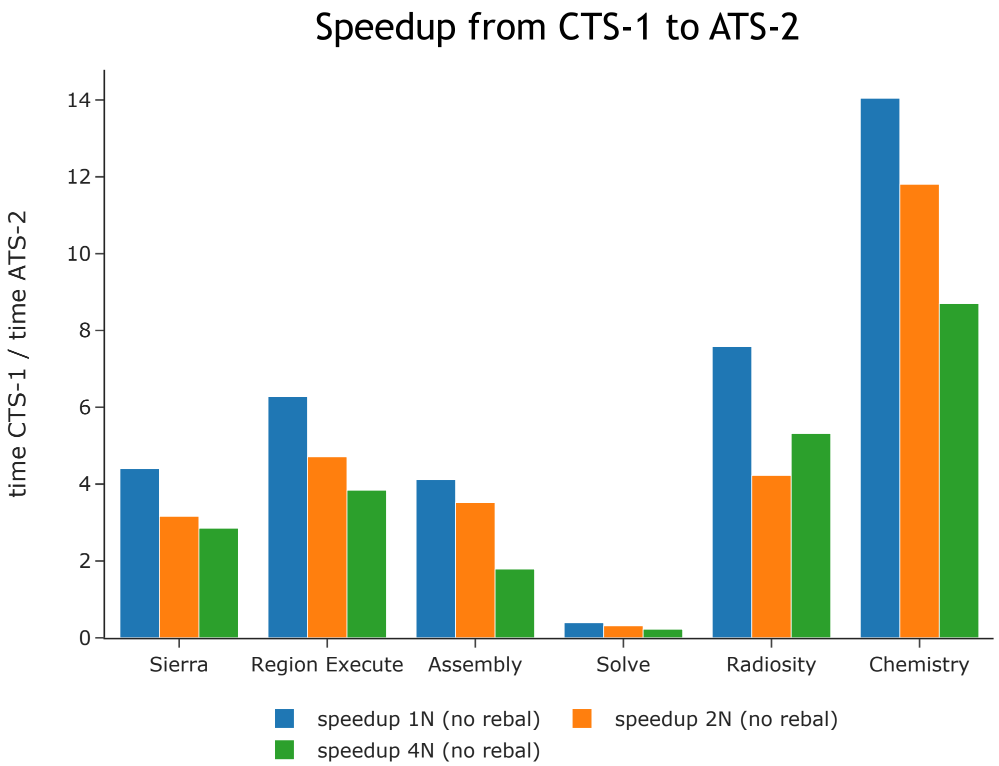




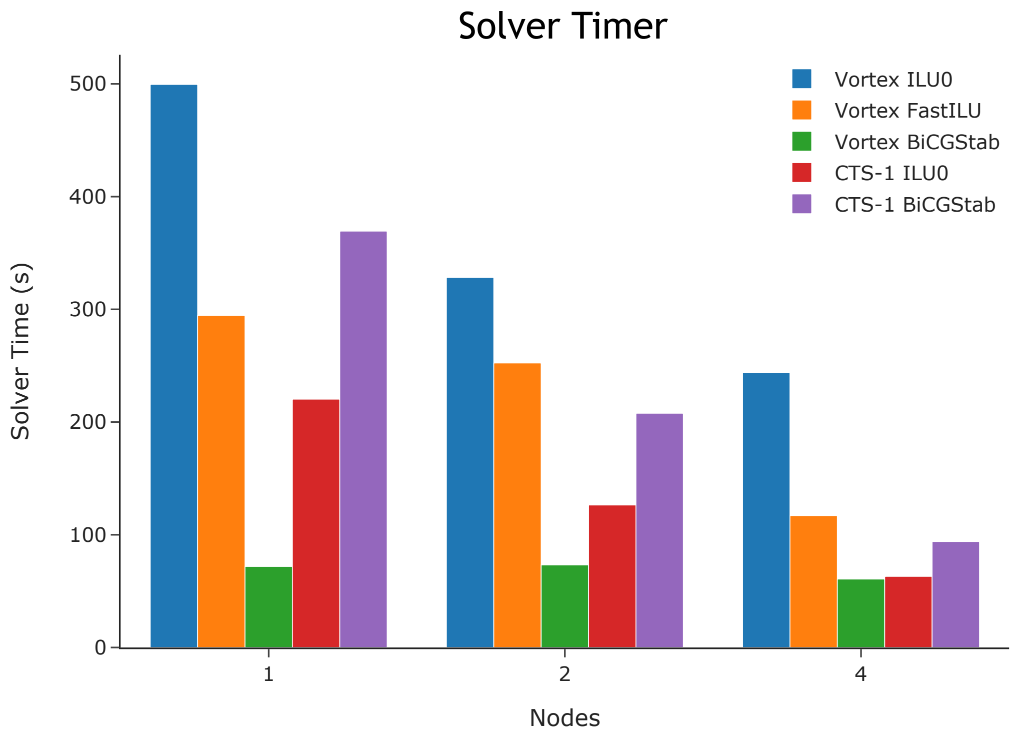
All results use explicit chemistry

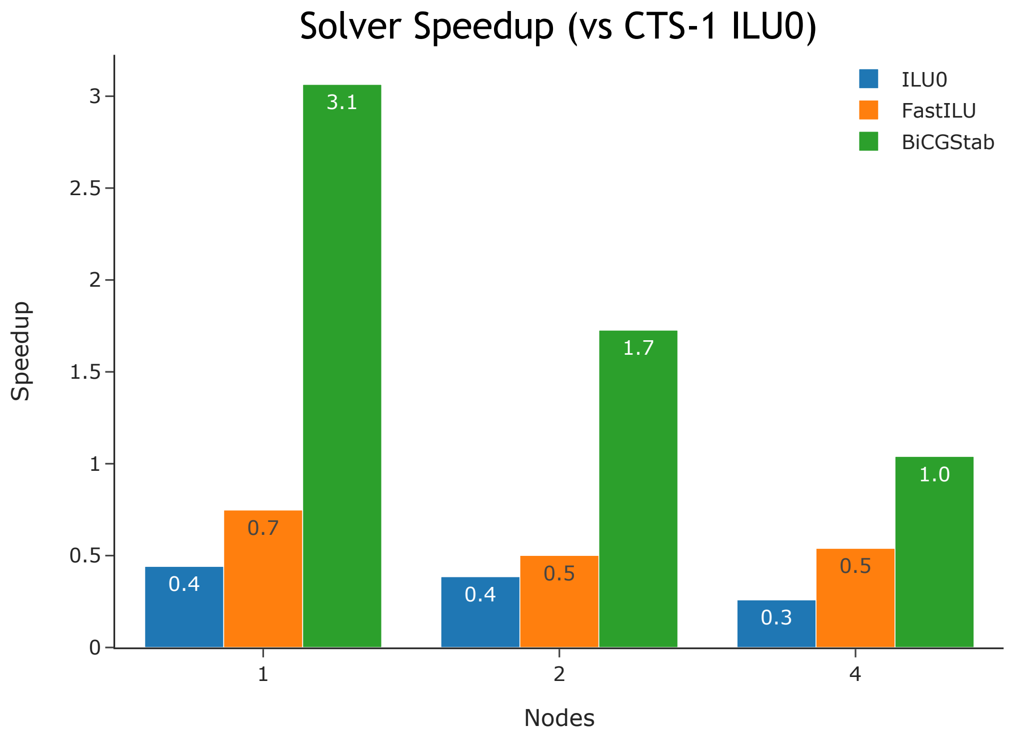
Abnormal surrogate performance: Node scaling, ILU0+GMRES





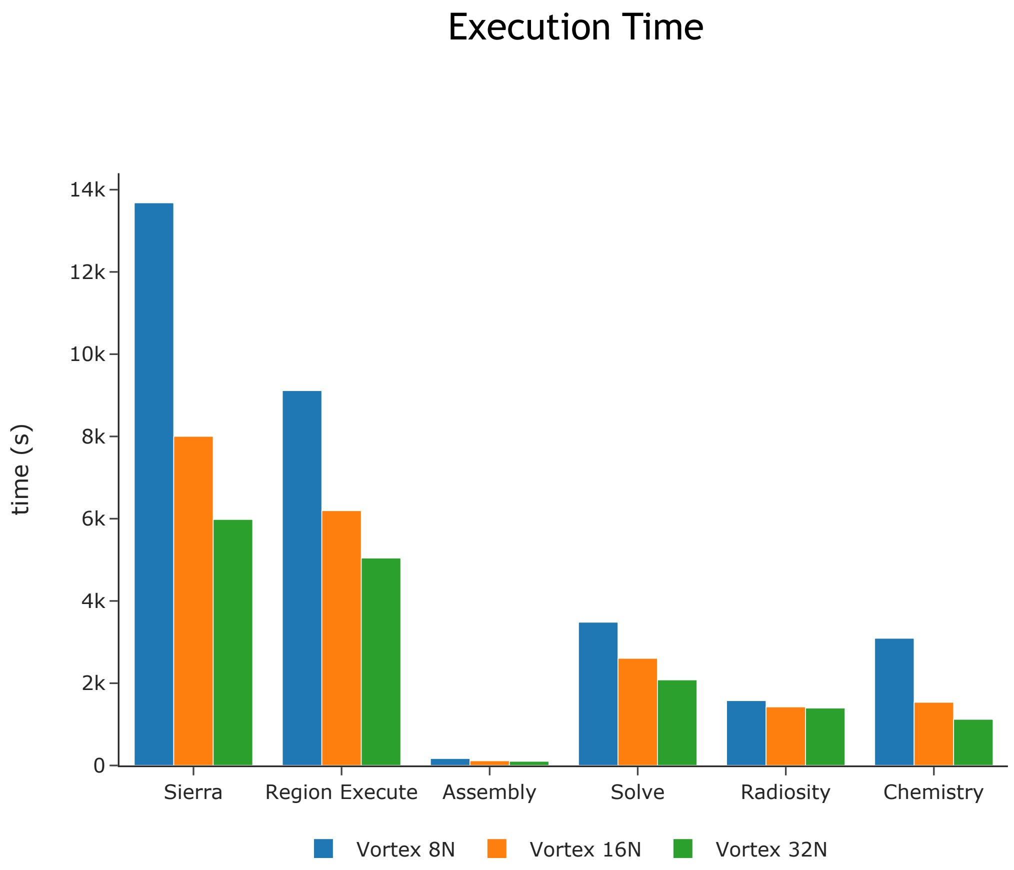
Abnormal surrogate performance: ATS-2 solver comparisons

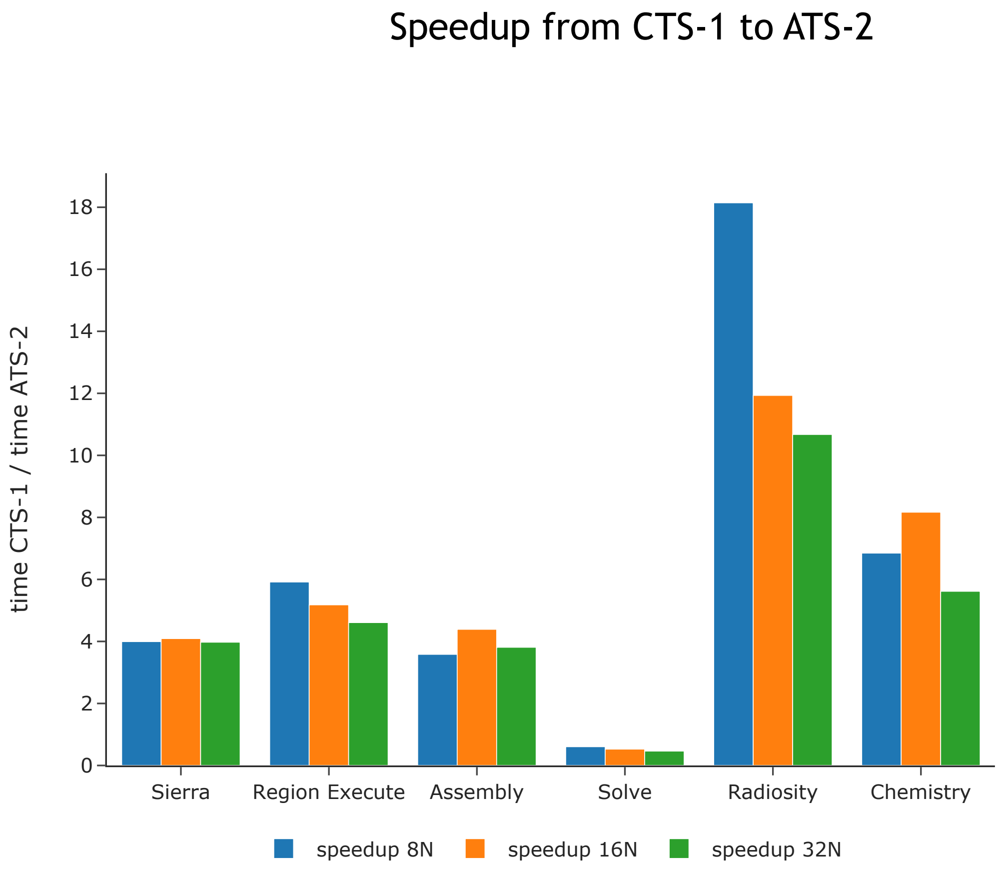




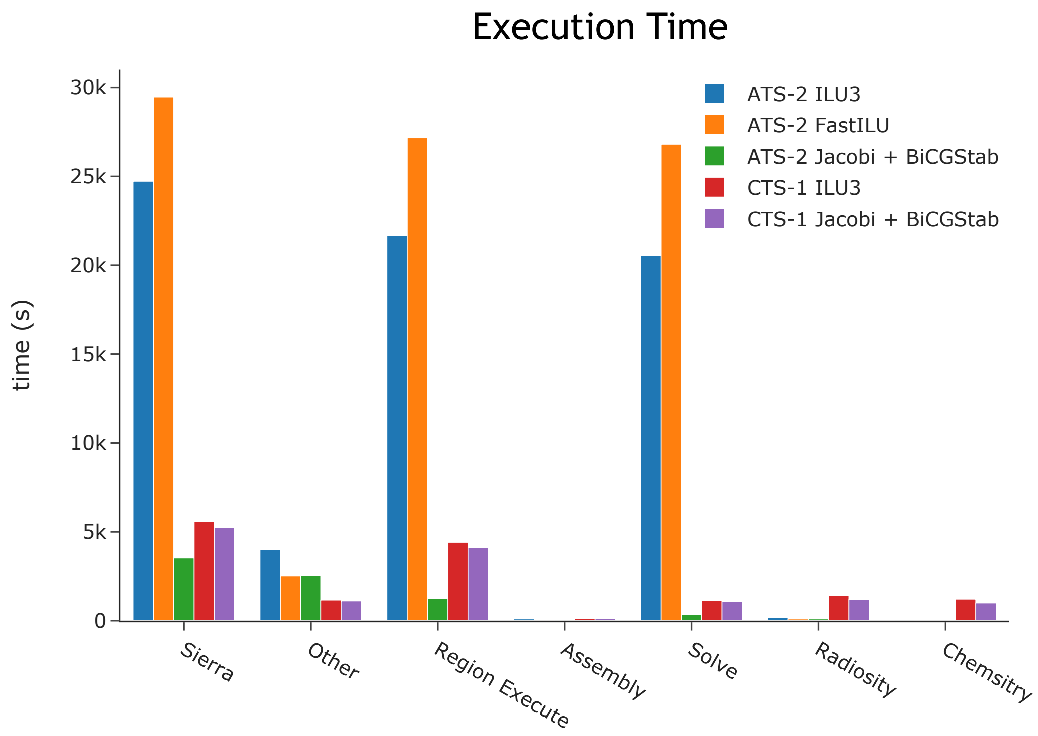
*Note: ATS2 preconditioners are quite different internally from those on CTS1.

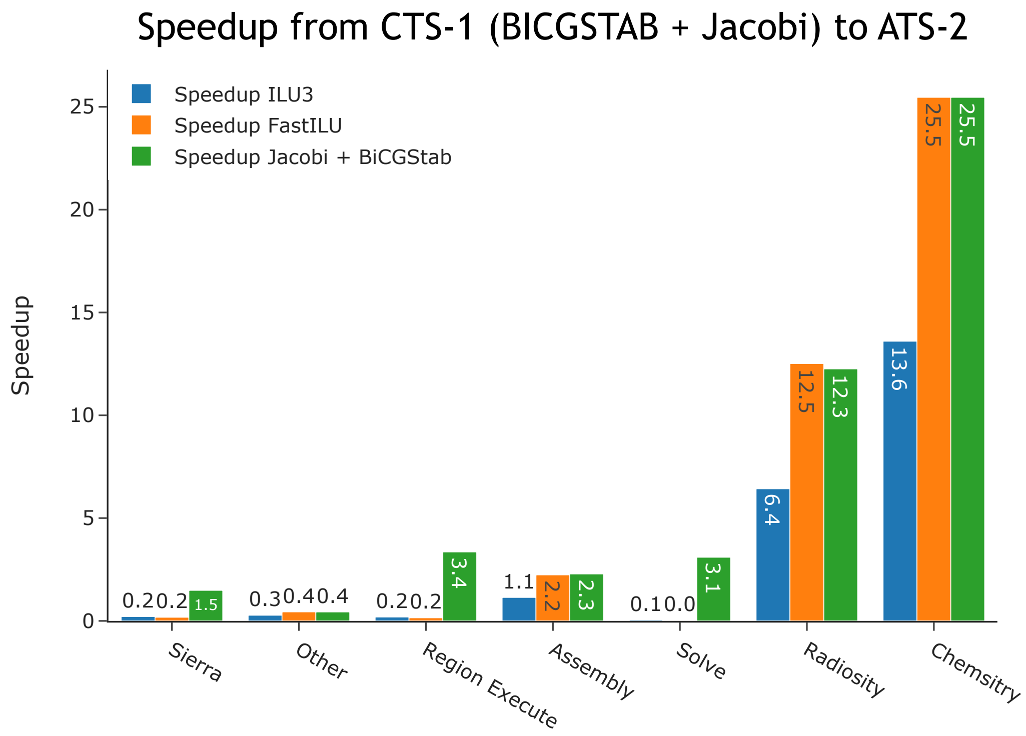
Uniform mesh refinement study: ILU0





Acceptance Test





Why ILU is challenging to run well on GPUs for Aria:

- ILU is based upon algorithms that are **inherently sequential**!
  (LU factorization and triangular solves)
- ARIA acceptance problem needs **high level 3 fill,** which limits available parallelism.
- Options:

Slow, Strong, and Sturdy

Fast, but Reckless!

Which preconditioner wins?

Incomplete LU factorization (ILU) Preconditioning Choices:

Least Robust

Most Robust

Robustness

Factorization (ILU)

Triangular Solve (TRSV)

Fast ILU	**RILU(k)**	**Serial RILU(k)**
Approximate iterative factorization	Extracts some parallelism with level scheduling	Non-parallel
Fast TRSV	**KKSpTRSV**	**Serial TRSV**
Polynomial approximation to triangular solve	Extracts some parallelism with level scheduling	Non-parallel

Parallelism

Faster on GPU

Slower on GPU

Solvers Team Approach

Why are solver team results different from Sierra results?

- Focus on **ILU(3)** test case.
  - Only ILU fill option that worked for the acceptance test
  - Sierra team shows results for ILU(0).
- All tests on **4 nodes**.
  - 36 MPI ranks/node CTS1
  - 4 MPI ranks (1 per GPU) ATS2
  - Sierra tests also show scaling for 1 and 2 nodes. Detected anomalies in some preconditioner options for smaller node count.
- Testing and development on **one test matrix.**
  - Actual matrix from surrogate problem.
  - Sierra tests run entire suite of matrices resulting from several nonlinear solves.
- Solvers tests included **most recent updates**.
  - Sierra timings do not include latest Trilinos PRs.

Conclusions

Although didn’t strictly hit 4x speedup, demonstrated speedup is useful for analysts to leverage ATS-2 hardware

Fixing robustness issues in parallel solve on acceptance test remain an issue that will be addressed in FY23

“Other” time (initialization, IO, etc.) is problematic on full acceptance test

Milestone spurred productive collaboration between app and solver teams, and we foresee this collaboration continuing indefinitely

Lessons learned

Collaborating across centers poses challenges (1400 and 1500)
- scheduled collaboration meetings
- shared milestone deliverables (including on official milestone)

Surrogates are useful
- access issues, faster development iterations…
- …but aren't the same problem
- solver implementation -> testing on surrogate -> application to acceptance test takes time

Milestone timeline could have been accelerated
- identify surrogate and acceptance problems from analyst team
- Trilinos team had to develop a viable ILU implementation
- optimization on surrogate and acceptance test
- application to full acceptance test
- delayed realization of issues with GPU implementation on acceptance test late in the problem runtime led to less than optimal outcome
- Vortex DST at milestone conclusion caused further heartburn

Specific use cases can focus efforts

Future Aria Work

Looking at performance speedup outside of core physics ("Other" timer)

- IO issues in STK or IOSS layers

Responsive porting of any new physics identified by ND analysts as their models change

Working with analyst community to include user subroutines in Aria proper

Preparing for ATS-3 & ATS-4

Future Trilinos Work

Complete Kokkos Kernels ODE Solvers. Expand user options and features.

ParILUT preconditioner

- ILUT traditionally works for ARIA problems but is intractable on GPUs (essentially serial).
- New ParILUT implementation coming into Kokkos Kernels. (Parallel variation of traditional ILUT.)

Troubleshoot numerical issues (RILU(3) + KKSpTRSV) causing solve failures in acceptance test

Expand preconditioner development and testing and further enhance Trilinos performance monitoring dashboard

Add new polynomial preconditioning option for subdomain solves. (Very parallel.)

Questions?