SAND2022-5543C

Sandia National Laboratories

Exceptional service in the national interest

Sandia ATDM: Applications and Components

Session - NNSA/ATDM Application BoF

Curtis Ober for SNL ATDM Team

May 5, 2022 1:00pm – 2:30pm EDT

2022 ECP Annual Meeting (Virtual)

Sandia National Laboratories is a multimission 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.

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 ATDM Team Members

Program Management: Jim Stewart, Curt Ober, Paul Crozier, Richard Kramer, Rob Hoekstra

SPARC PI: Travis Fisher

SPARC Team: Matt Bopp, Sam Browne, Victor Brunini, Jared Crean, Derek Dinzl, Jeff Fike, Mike Hansen, Wyatt Horne, Michael Krygier, Brad Maeng, Tony Nguyen, James Overfelt, Heather Pacella, Cory Stack, and Jerry Watkins

SPARC V&V Team: Sarah Kieweg, Brian Carnes, Lincoln Collins, Derek Dinzl, Travis Fisher, Brian Freno, Cem Gormezano, Mike Hansen, Ryan Keedy, Jorge Moreno, Erin Mussoni, Jaideep Ray, and Greg Weirs

EMPIRE PO: Keith Cartwright

EMPIRE SCRUM Master: Wade Burgess

EMPIRE Team: David Collins, Eric Cyr, Daniel Jensen, Todd Kordenbrock, Caleb Logemann, Ed Love, Billy McDoniel, Duncan McGregor, Logan Meredith, Chris Moore, Roger Pawlowski, Tim Pointon, Troy Powell, Nick Roberds, Sidney Shields, David Sirajuddin, Shane Stafford, Tom Smith, Scot Swan

Components Project Managers: Ron Brightwell, Robert Clay, David Littlewood, Ron Oldfield, Andrew Salinger, Dan Turner, Michael Wolf

Components Principal Investigators: Ross Bartlett, Jonathan Hu, Si Hammond, Byron Hanks, Ken Moreland, Stephen Olivier, Roger Pawlowski, Eric Phipps, Siva Rajamanickam, Greg Sjaardema, Christian Trott

Components Team: Seher Acer, Andrew Bradley, Luc Berger-Vergiat, Lewis Cannada, Brian Carnes, Jan Ciesko, Sidafa Conde, Kevin Copps, Vinh Dang, Mohamed Ebeida, Nathan Ellingwood, James Elliott, Noah Evans, Kurt Ferreira, J. Forester, Joe Frye, Chris Glusa, Ryan Grant, Byron Hanks, G. Hansen, Evan Harvey, Daisy Hollman, Dan Ibanez, Brian Kelley, Kyungjoo Kim, Scott Levy, Jonathan Lifflander, Jeff Mauldin, Kathryn Maupin, D. Mavriplis (U. Wyoming), Duncan McGregor, Jeff Miles, Phil Miller, Sean Miller, Shymali Mukherjee, Curt Ober, Tom Otahal, Steve Owen, Kevin Pedtretti, Mauro Perego, Edward Phillips, David Poliakoff, Nathan Roberts, Roshan Quadros, W. Scott, Chris Siefert, Nicole Slattengren, Bill Spotz, Cory Stack, Paul Stallings, Dan Sunderland, Gary Templet, Irina Tezaur, A. Toth, Ray Tuminaro, Craig Ulmer, Lee Ward, Jerry Watkins, Jeremy Wilke, Ichi Yamazaki, Andrew Younge

Advanced Technology, Development and Mitigation (ATDM)

National security applications require development of advanced computational technologies for next-generation platforms (NGPs) and exascale computing systems, in order to achieve their full potential performance and enable future capabilities.

A primary challenge has been contending with the variety and rapidly evolving levels of parallelism, and heterogeneous architectures.

To mitigate the associated risks with this changing landscape and provide future capabilities, software components that are scalable, modular, and cross-cutting have been developed and utilized in ASC/ATDM.

Platforms

Applications

SPARC
Virtual flight-testing platform for hypersonic vehicles

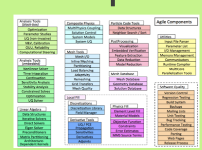
EMPIRE
Sandia's next-generation plasma simulation code

Software Components

TRILINOS

SEACAS, STK, Scalable Solvers, Tpetra, DARMA, UMR, Zoltan2

Agile Components



Performance Abstractions

kokkos kernels

Embedded Analysis

Sacado

ROL

ParaView

ParaView Catalyst

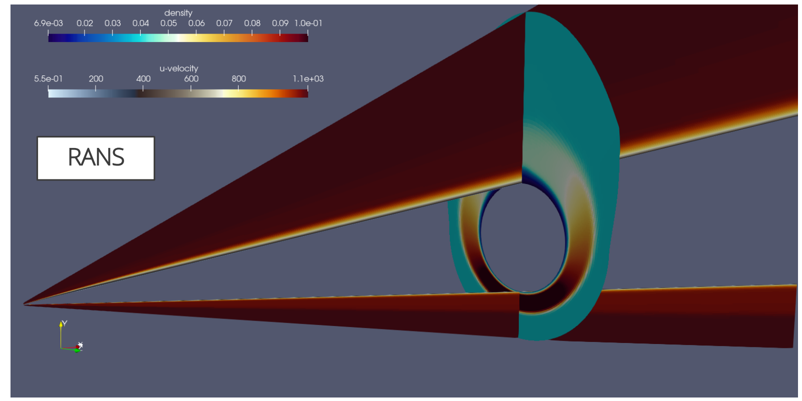
SPARC

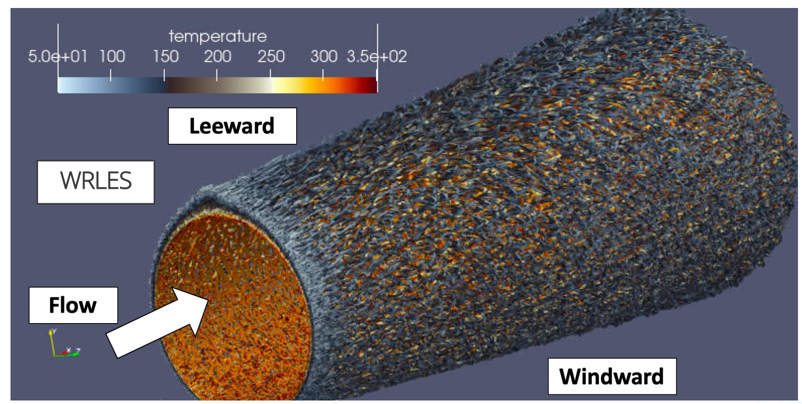
SPARC – Sandia Parallel Aerodynamics Reentry Code

SPARC Overview

Goal: Create a credible virtual flight-testing platform for hypersonic vehicles

- **Modeling**
  - Perfect and non-equilibrium thermal and chemical gas models
  - Euler, Laminar, RANS, Hybrid RANS/LES, LES, and DNS
  - Structured and Unstructured Finite Volume methods
  - R&D in structured and unstructured high-order methods
  - Simulate coupled ablation
  - Couples to SIERRA for full-system thermal and structural analyses
- **Performance and Portability**
  - Performance Portability through Kokkos
  - Good performance on x86, Arm, and GPU platforms
  - Uses performance portable/scalable linear solvers from Trilinos
  - Uses embedded geometry and inline mesh refinement
- **Credibility**
  - Validation with UQ against wind tunnel and flight test data
  - Visibility and peer review by external hypersonic community





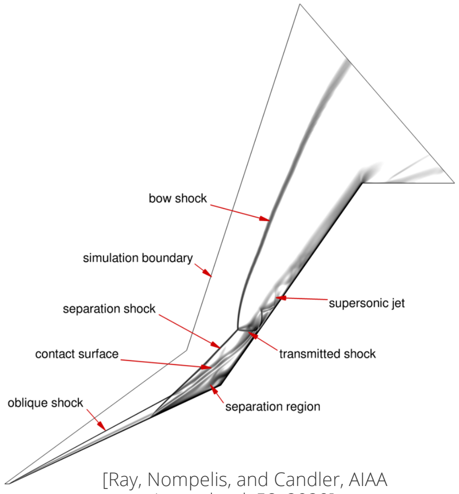
SPARC and Components

- Trilinos provides performance-portability capabilities
  - Kokkos/Tpetra for nonlinear residual and Jacobian assembly
    - Implement domain-specific data structures & mesh iteration abstractions
    - Can tune implementation for different platforms to maximize performance
  - Kokkos Kernels/Ifpack2/Belos/Teko for performance-portable linear solvers
  - Seacas/IOSS for performance-portable I/O and mesh decompose/recompose
  - All SPARC physics code remains platform agnostic
  - 96.1% of SPARC code base is platform agnostic
- Additional capabilities provided by other components, e.g.,
  - STK for mesh transfers and coupling
  - Sacado for low-level sensitivity computations
  - Paraview and Catalyst for in-situ visualization and quantitative analysis
- Components allow research into new algorithms and cutting-edge capabilities
  - MueLu for improved steady state solvers
  - NOX/LOCA for trajectory continuation methods
  - Tempus for forward and adjoint sensitivities

Adjoint Sensitivities and Calibration of Hypersonic Flow Problems
ATDM FY 2022 L2 Milestone for Embedded Components

- Goals
  - Assessment of the feasibility, accuracy and efficiency of computing steady-state parametric adjoint sensitivities along with their utility in solving calibration problems (e.g., double-cone)
    - Would allow scalability to much larger parameter spaces
    - Provide a foundation for future capabilities such as shape optimization for vehicle design
  - Leverage key component capabilities
    - Sacado – sensitivities via automatic differentiation
    - ROL – embedded optimization
    - Tempus – time integration, pseudo-transient adjoint sensitivities
    - Zoltan2 – graph coloring for calculation of SPARC adjoint
- Mid-Year Status
  - Forward sensitivity approach is robust but expensive
  - ROL is effective for solving these types of calibration problems
  - Adjoint sensitivities are very efficient compared to forward sensitivities (~3x)
  - Challenges remain with robustly solving adjoint systems

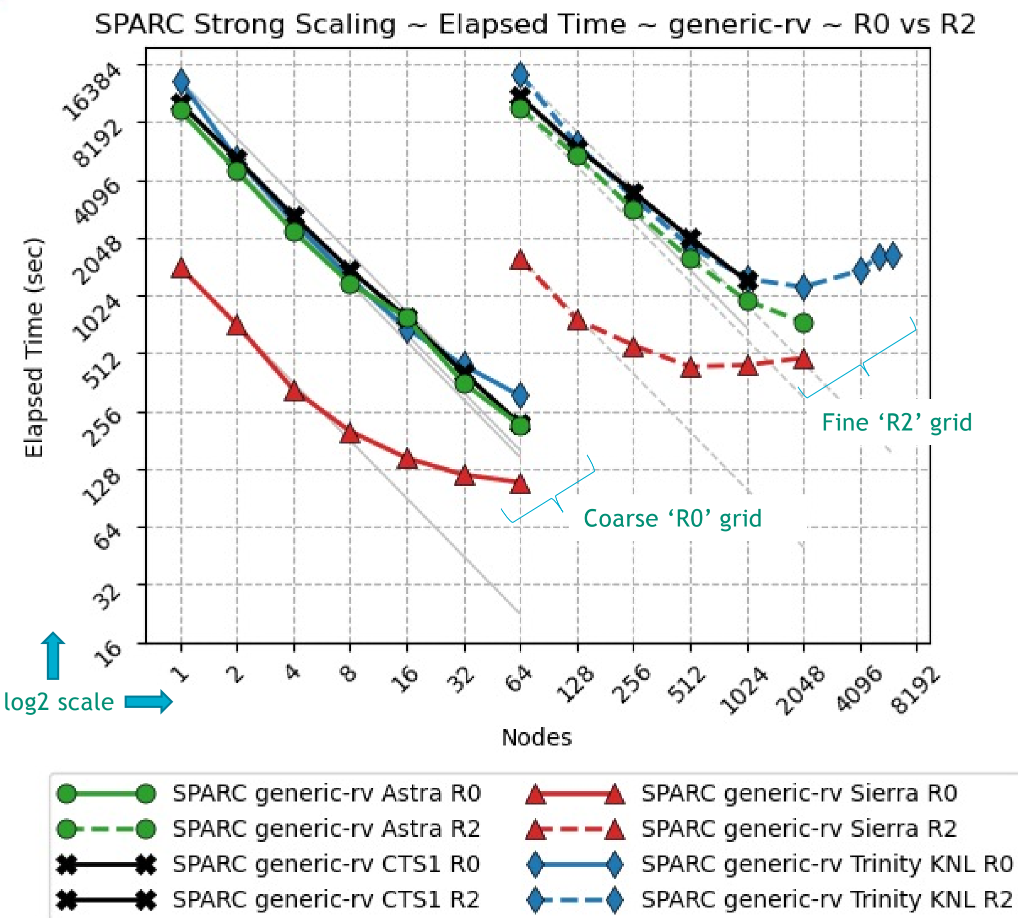
Double-cone geometry



[Ray, Nompelis, and Candler, AIAA Journal, vol. 58, 2020]

Team: Eric Phipps, Patrick Blonigan, Kathryn Maupin, Jaideep Ray

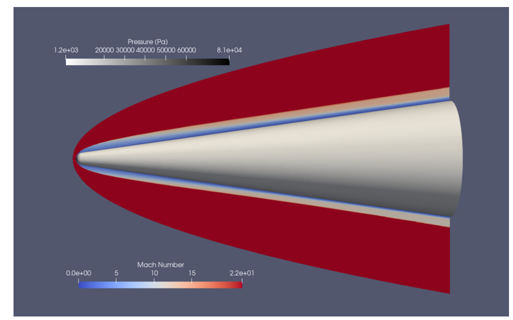
Hypersonic Performance – Strong Scaling, Overall Runtime



Performance Results from L1 Milestone

Generic RV

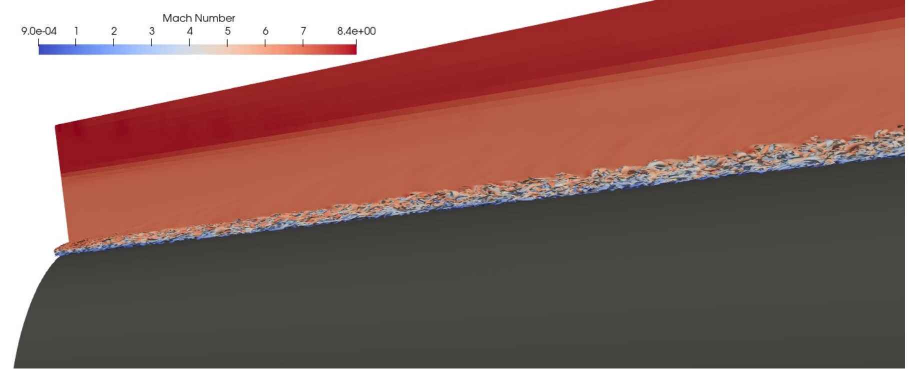
- Steady-state
- 5 species
- 2 Temperature
- RANS (11 dofs/cell)

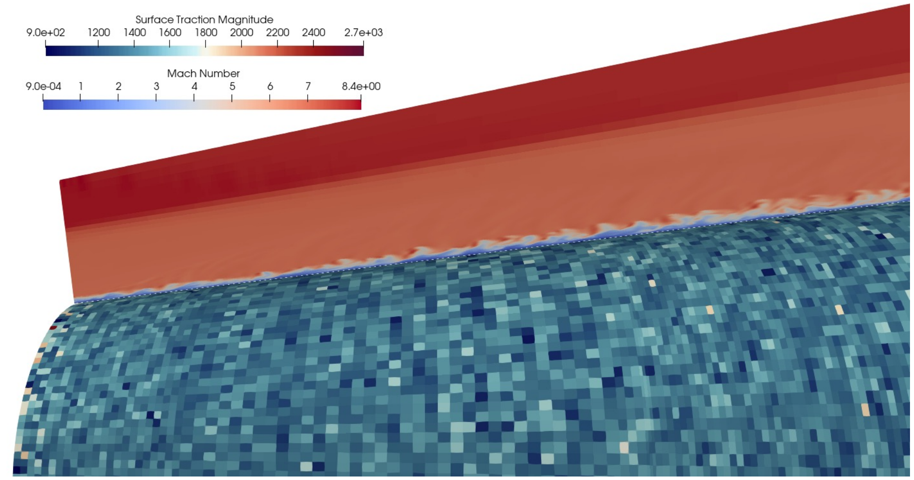


CTS1 as baseline:
Trinity: 0.8-1.2x
Astra: 1.2-1.4x
Sierra: 2-8x

“Production” LES Technologies with Components

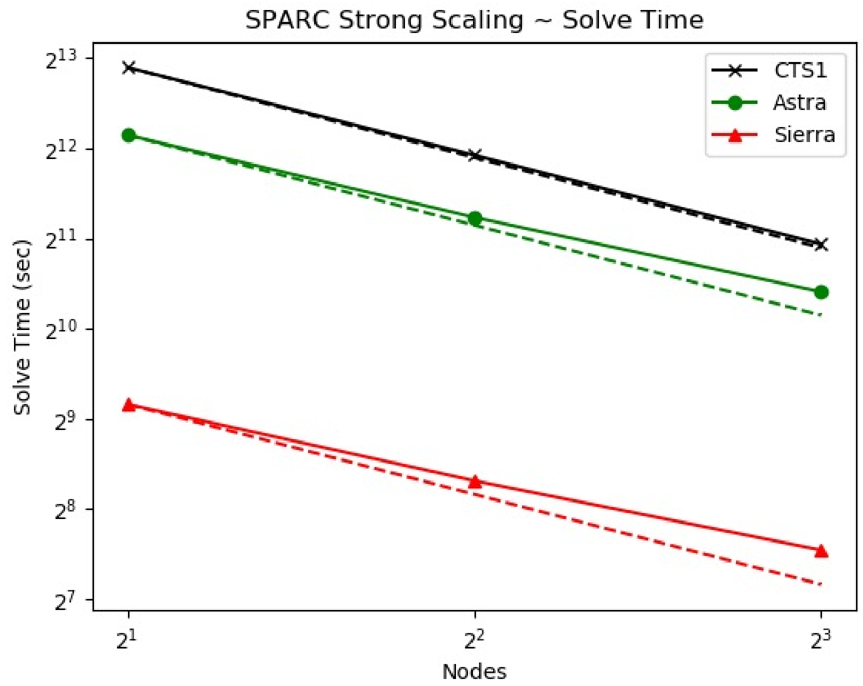
- Seacas/IOSS for mesh decompose/recompose
- Kokkos and Tpetra for assembly
- Ifpack2 block-Jacobi solver (performance portable)
- STK parallel transfer
  - Initial condition from RANS solution
  - Boundary data from RANS solution
  - Output extraction of subsets and transfer surface loads to structural dynamics cone mesh
- STK coupling (*new*)
  - MPMD coupling to Sierra/SD for passing loads when file coupling infeasible
  - Provides consistency checking facilities to reduce parallel hangs during development





Large Eddy Simulation Performance – Strong Scaling

- Initial FY22 focus on lower node counts
- CTS1 (x86) has good strong scaling
- Astra (ARM) nearly twice as fast at lower node counts
- ATS-2/Sierra (V100) is 12x faster
- Can be further improved
  - Expose more concurrency
  - Limit register spillage



EMPIRE – ElectroMagnetic Plasma In Realistic Environments

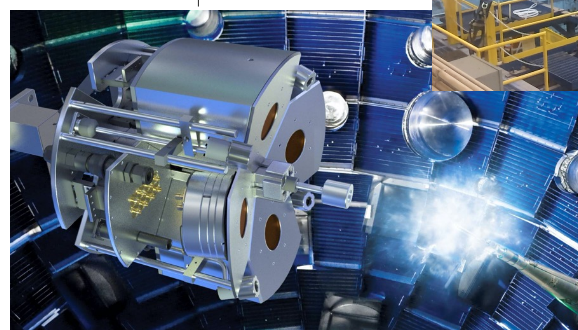
EMPIRE Overview

EMPIRE is Sandia's next-generation plasma simulation code

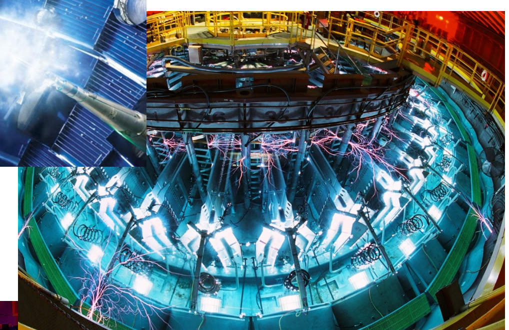
- Goals:
  - Simulate plasmas over a broad density range
    - Particle-In-Cell (PIC) dominating at low densities
    - fluid at high densities
    - hybrid approach in the middle.
  - Performance portability on next-generation architectures
- Code Design:
  - Three physics capabilities (stand-alone or coupled)
    - Electromagnetics, PIC, Fluids
  - Utilize software components (Trilinos, Kokkos, Darma, …)
- Significant progress made in supporting production use, especially for pulsed-power facility simulation
- Validation continues against NIF and Z facility experiments

HERMES-III Accelerator

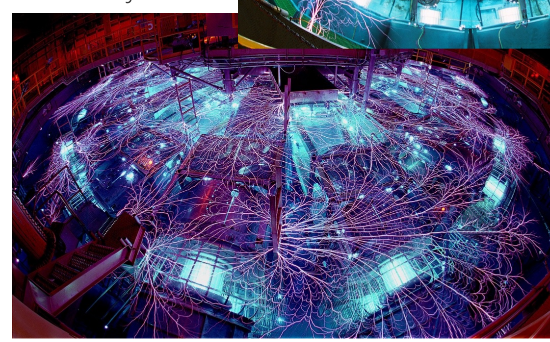
Validation experiment at NIF



Saturn



Z Facility



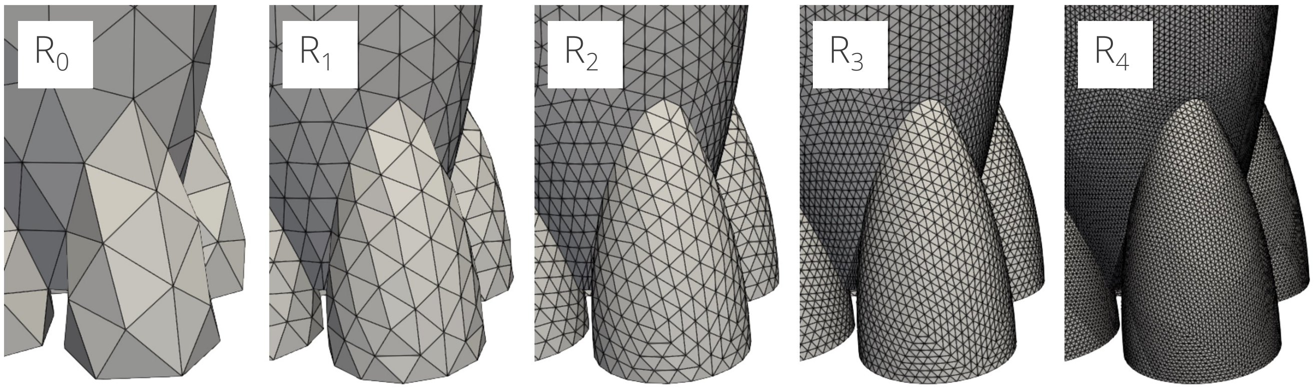
EMPIRE and Components

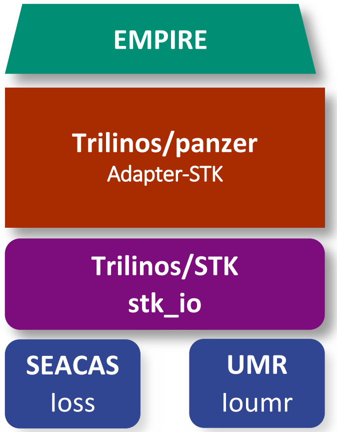
- Trilinos provides performance-portability capabilities
  - Kokkos/Kokkos Kernels/Tpetra for data structures
  - Kokkos Kernels/Ifpack2/Belos/Teko/MueLu/Amesos2/Zoltan2/Anasazi/Stratimikos/ for linear solvers
  - Shards/Intrepid2/Panzer/Thyra/Phalanx/Sacado for nonlinear residual and Jacobian assembly with Automatic Differentiation
  - Seacas/IOSS for performance-portable I/O and mesh decompose/recompose
  - NOX for Anderson acceleration for coupling the Fluid to the EM
- Additional capabilities provided by other components, e.g.,
  - STK/Percept/SEACAS/Panzer/Pamgen for meshing
  - Sacado for low-level sensitivity computations
  - UMR for in-situ uniform mesh refinement
  - Darma for checkpointing and load balancing

UMR In-situ Mesh Refinement within EMPIRE
ATDM FY 2022 L2 Milestone for Embedded Components

- Goal: Provide and demonstrate in-situ mesh refinement capability in EMPIRE at scale with a target size of ~120B elements.
- Refinement without geometry leads to inaccurate analysis
  - Only R_0 made with Cubit
  - R_1-R_4 made by UMR projection to geometry

- Mid-Year Status
  - ~120B elements on test geometry
  - 5040 MPI ranks with ~15B elements
    - Refine time dominated by read time
  - MPI memory usage
    - Refine once, read uses most memory
    - Refine twice, refine uses most memory
  - Geometry association now a pre-process
    - Reduces MPI comm., increases performance

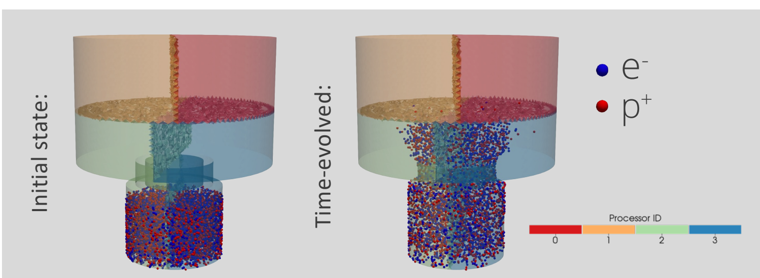




Team: Byron Hanks, Brain Carnes, Kevin Copps

Rebalancing in EMPIRE using Asynchronous Many Tasking (AMT)

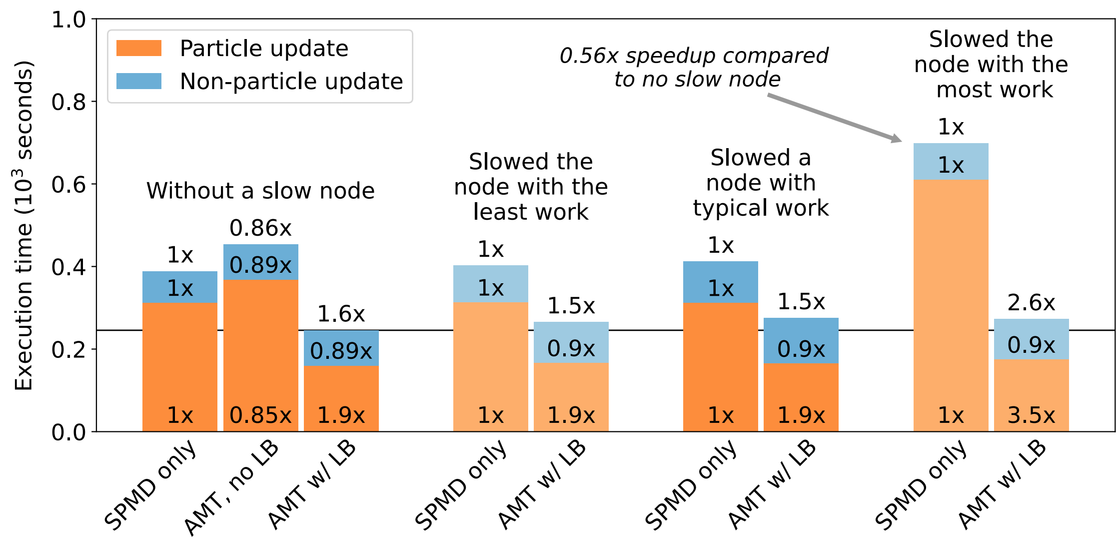
- **Conventional approach:** infrequently repartition the mesh to offset particle imbalance
  - Synchronous process: large volumes of data must be migrated to new processors or recomputed from the mesh
- **Our approach**: retain initial **static mesh decomposition** but further partition mesh **spatially into *k* colors**
- Utilizes **DARMA/vt** (Virtual Transport), a **highly MPI interoperable** tasking runtime to express dependencies and run code **asynchronously**
- Includes a suite of **highly scalable** load balancers (LB)



Team: Jonathan Lifflander, Nicole Slattengren, Roger Pawlowski, M. Scot Swan

- A node on Stria (ARM cluster) recently had a power supply **hardware failure** causing some nodes to run at *500 MHz*, instead of *2 GHz.*
- We simulated this by running a single node out of 16 at *1 GHz*, to demonstrate how load balancing could quickly observe and react.
  - The load balancer automatically moves work away from the slow node!

Experimental results with a highly particle imbalanced B-Dot problem*

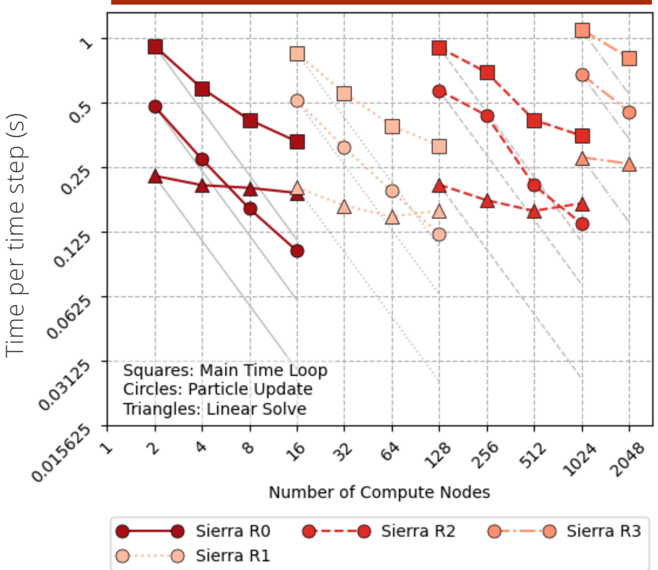


** Special thanks to Kevin Pedretti for HPC support to help us configure these experiments.*

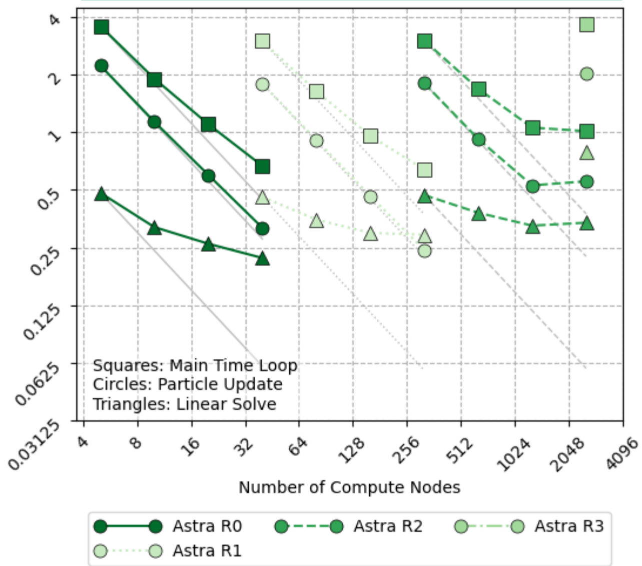
Performance results for EMPIRE

Single platform scaling results

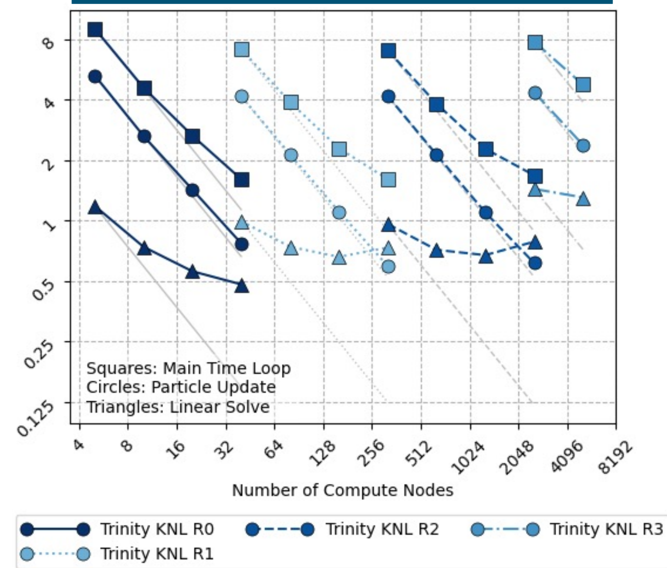
Sierra
4 GPU/socket; 1 MPI/GPU
Up to 2048 nodes (47% of total)



Astra
2 MPI/socket; 9 threads/MPI
Up to 2560 nodes (99% of total)



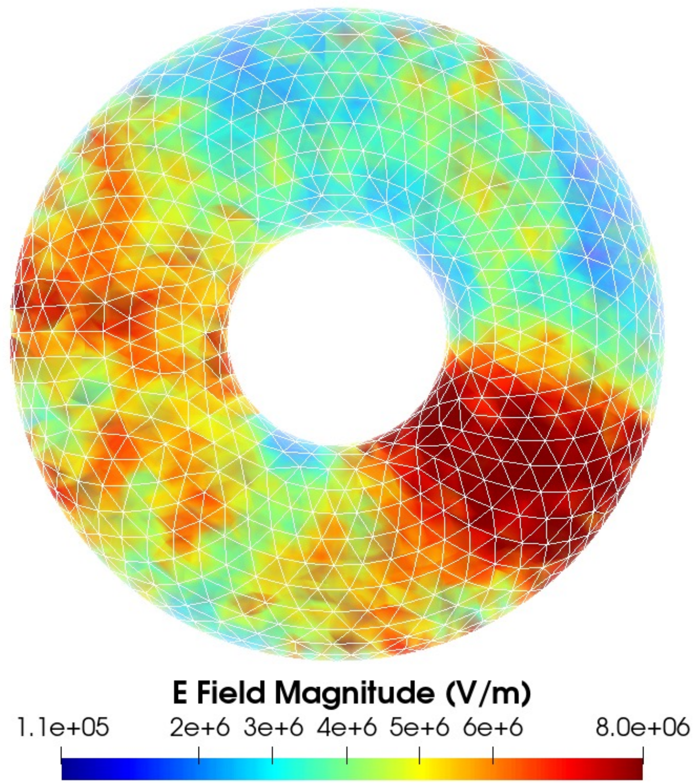
Trinity/KNL
4 MPI/socket; 16 threads/MPI
Up to 5120 nodes (52% of total)



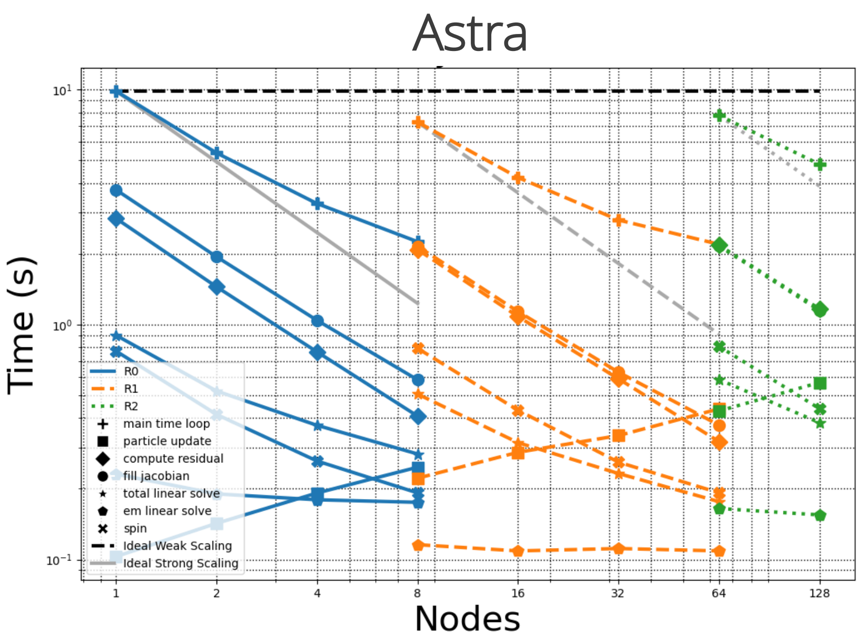
Performance Results from L1 Milestone

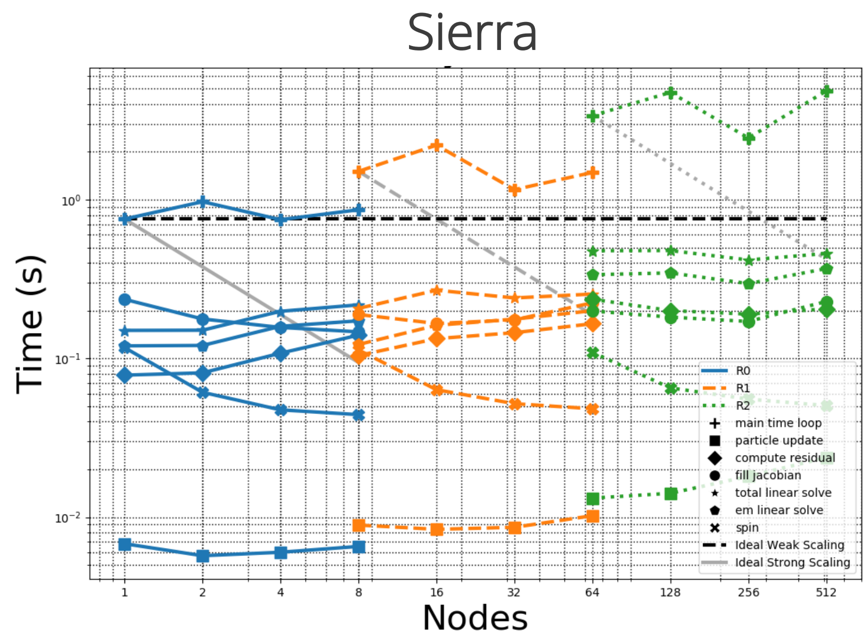
Toroidal reference problem for performance development

- The toroidal reference problem was designed to drive and test EMPIRE performance.
  - Designed to have simple geometry, but exercise all parts of the code.
  - Runs on all flavors of EMPIRE: fluid, PIC, and hybrid.
  - Runs with 3 different colliding gas species to test scaling in EMPIRE's collisions.
  - Regularly tested on Stria/Vortex, then ran on Astra/Sierra with scaling studies.
- The simple geometry of the problem allows the EMPIRE team to test the scaling of the code without the overhead of the complexity of the production workflow.



Toroidal reference problem Hybrid scaling





- Problem size was selected to run across PIC, Fluid and Hybrid.
  - R0 is ~100k elements, 750 particles per cell
- Astra scales well; investigated particle update scaling.
- Sierra ~10x faster, and not enough work to show scaling
- Plots from “near” automated process and will allow regular performance plots.
- Preliminary results and improvements are on-going.

Closing

ATDM applications (SPARC and EMPIRE) are using software components to

- Obtain performance portability
  - Kokkos, Kokkos Kernels, Tpetra
  - Trilinos Linear Solvers
    - Ifpack2, Belos, Teko, MueLu, Amesos2, Anasazi, Stratimikos
  - Seacas/IOSS

- Support and enable new and future capabilities
  - STK, Sacado, Paraview, Catalyst, MueLu, NOX, LOCA, Tempus, ROL, Zoltan2, Shards, Intrepid2, Panzer, Thyra, Phalanx, NOX, Percept, Panzer, Pamgen, UMR, Darma

Thanks!

Any questions?

Backup Slides

Enabling Research into Matrix-Free Solvers with Trilinos Solvers

Evaluating Matrix Free Solvers in SPARC

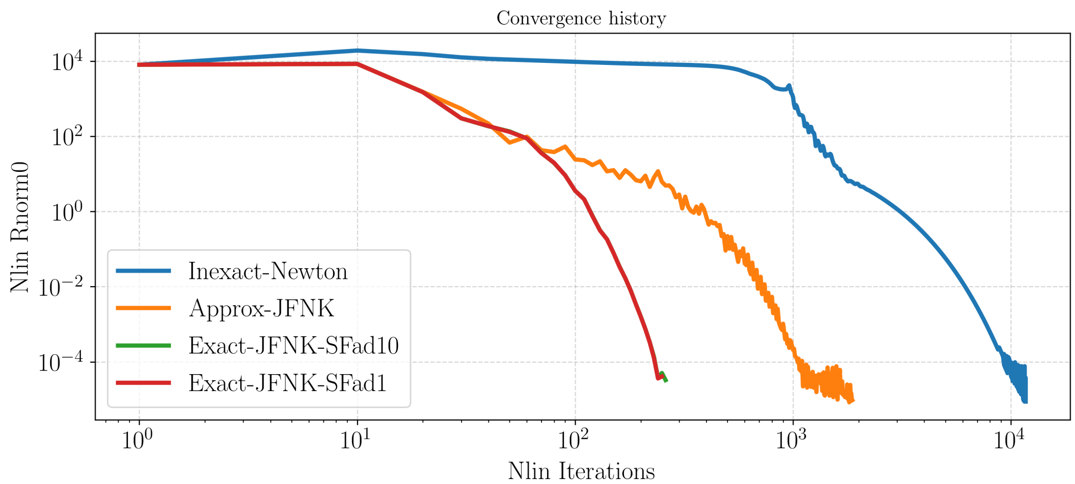
- **Inexact Newton**
  - 7-pt stencil
  - First-order inviscid Jacobian
  - Neglect viscous cross-terms
  - Linear convergence
- **Matrix Free Approximate**
  - Use inexact Newton matrix for preconditioning
  - Use Frechet derivative approximation
  - Quadratic convergence in ideal conditions
  - Nonlinear evaluations scale as number of linear iterations
- **Matrix Free Exact**
  - Use inexact Netwon matrix for preconditioning
  - Use Sacado/AD for matrix-vector product
  - More robust quadratic convergence
  - Nonlinear evaluation w/AD scales as number of linear iterations

No perturbation constant
Insensitive to equation scaling
Can neglect sensitivities easily (e.g. SST turbulence model terms)

Slower!
Requires templated code
Increases build time
May be harder to solve

Enabled and Performance Portable courtesy of Sacado and Belos

Success story: M = 0.2 Turbulent Flat Plate BL



	Nlin Iterations	Problem Solve Time (s)	Belos Solve Time (s)
Inexact-Newton	11684	109.216	58.7482
Approx-JFNK	1873	51.9301	43.1881
Exact-JFNK-SFad10	262	46.3098	44.2337
Exact-JFNK-SFad1	256	15.2841	13.4616

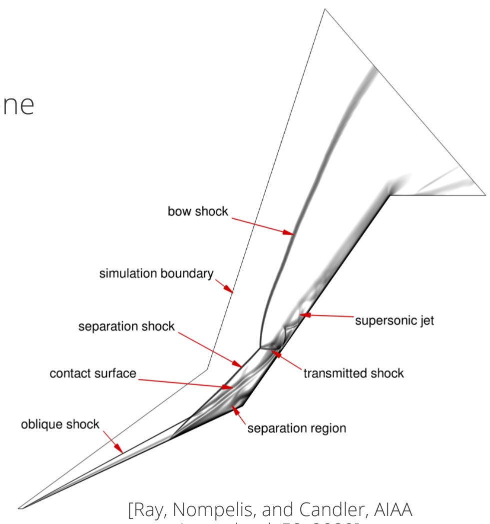
Spalart Allmaras turbulence model
- Second-order finite volume
- Aggressive CFL schedule
- Exact matrix-free leads to 7x speedup

SST turbulence model
- Approx matrix free doesn't work
- Exact matrix free can work if code is added to neglect terms
- Inexact Newton still preferred

Success of exact Jacobians for real problems is more challenging and still being wor

Data Propagation Components: Adjoint Sensitivities and Calibration of Hypersonic Flow Problems in SPARC

- Meets criterion: Assessment of the feasibility, accuracy and efficiency of computing steady-state parametric adjoint sensitivities along with their utility in solving calibration problems on the double-cone problem
- Exceeds criteria
  - Similar assessments for the turbulent HIFiRE-1 problem
  - Demonstration of the sensitivity and calibration calculations on one or more GPU architectures
- Mid-year update:
  - Good progress on double cone calibrations using forward and adjoint sensitivity approaches
  - Challenges remain with robustly solving adjoint systems
- Preliminary assessment:
  - Forward sensitivity approach is robust but expensive
  - ROL is effective for solving these types of calibration problems
  - Adjoint sensitivities are very efficient when a solution to the adjoint system can be obtained
  - Robustly solving adjoint system is challenging
- Remaining work to complete double-cone assessment
  - Complete case 1 and case 4 forward- and adjoint-based calibrations
  - Performance/accuracy comparisons between forward and adjoint sensitivities
  - Uncertainty estimate comparisons between ROL-based and prior surrogate-based calibrations



[Ray, Nompelis, and Candler, AIAA Journal, vol. 58, 2020]

Data Propagation Components: Adjoint Sensitivities and Calibration of Hypersonic Flow Problems in SPARC

- General agreement between ROL and prior surrogate-based calibrations:

	Experiment	Initial Guess	ROL Converged	Bayesian (90% CI)
Density [g/m^3]	0.5848 (0.5439, 0.6257)	0.59 [+0.9%]	0.5988 [+2.4%]	0.5737 (0.5471, 0.6209)
Velocity [m/s]	2545 (2469, 2621)	2500 [-1.8%]	2488 [-2.3%]	2490 (2441, 2653)

Run35 (easiest): perfect gas model, H_0=3.71 HJ/kg, no vibrational non-equilibrium, no reaction non-equilibrium

	Experiment	Initial Guess	ROL Converged	Bayesian (90% CI)
Density [g/m^3]	0.4990 (0.4641, 0.5339)	0.51 [+2%]	0.4340 [-13%]	0.4897 (0.4328, 0.5645)
Velocity [m/s]	3246 (3149, 3343)	3300 [+1.6%]	3536 [+9%]	3340 (3211, 3654)

Case 1 (more difficult): real gas, H_0=5.55 HJ/kg, vibrational non-equilibrium, no reaction non-equilibrium

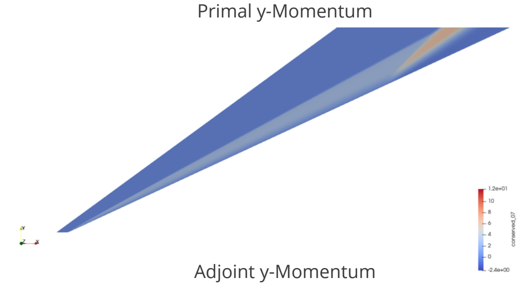
	Experiment	Initial Guess	ROL Converged	Bayesian (90% CI)
Density [g/m^3]	0.9840 (0.9151, 1.053)	0.86 [-13%]	0.8638 [-12%]	0.8608 (0.7996, 1.0396)
Velocity [m/s]	6479 (6285, 6673)	7060 [+9%]	6949 [+7%]	7060 (6380, 7089)

Case 4 (most difficult): real gas, H_0=21.77 HJ/kg, vibrational non-equilibrium, reaction non-equilibrium

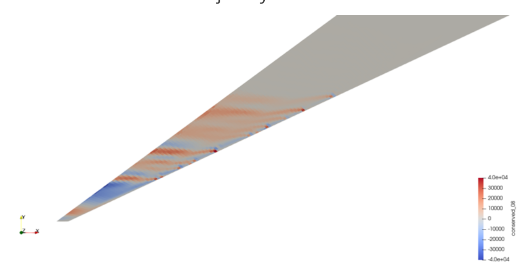
- Calibration progress:

	Run35			Case 1			Case 4		
	128x256	256x512	512x1024	128x256	256x512	512x1024	128x256	256x512	512x1024
Forward Sensitivity	✅	✅	✅	✅	✅	✅	✅	✅	✅
Forward Calibration	✅	✅	✅	✅	✅	✅	✅	❌ (bus error)	
Adjoint Sensitivity	✅	✅	✅	✅	✅	✅	✅ (SuperLU)	✅ (SuperLU)	
Adjoint Calibration	✅	✅	✅	✅	✅	✅	✅ (SuperLU)		

Primal y-Momentum



Adjoint y-Momentum

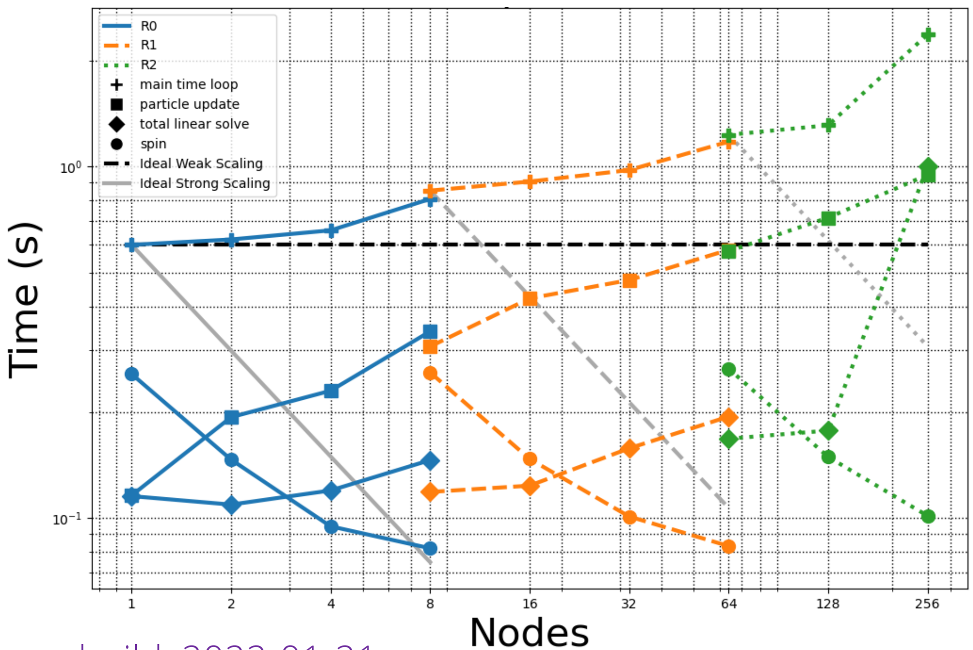


- Good adjoint performance:

Run35	State	Adjoint	ASA Total	FSA Total	x Speedup
128x256 (48 cores)	130.6 s	5.2 s	135.8 s	417.5 s	3.1
256x512 (96 cores)	302.9 s	13.4 s	316.3 s	923.9 s	2.9
512x1024 (192 cores)	1844.8 s	39.8 s	1884.6 s	5071 s	2.7

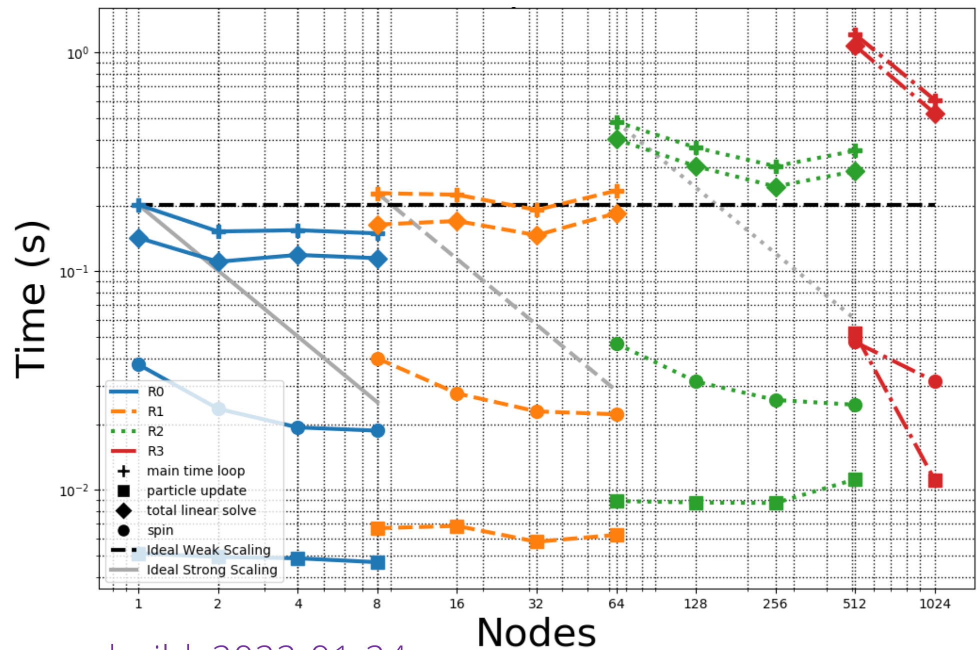
Toroidal reference problem PIC scaling

Astra



build: 2022-01-21

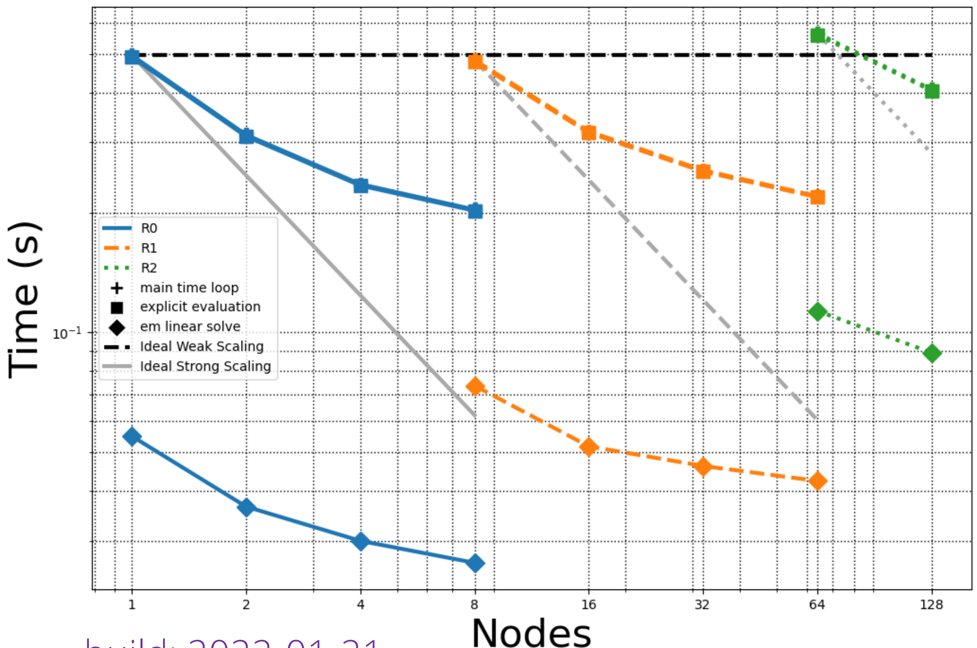
Sierra



build: 2022-01-24

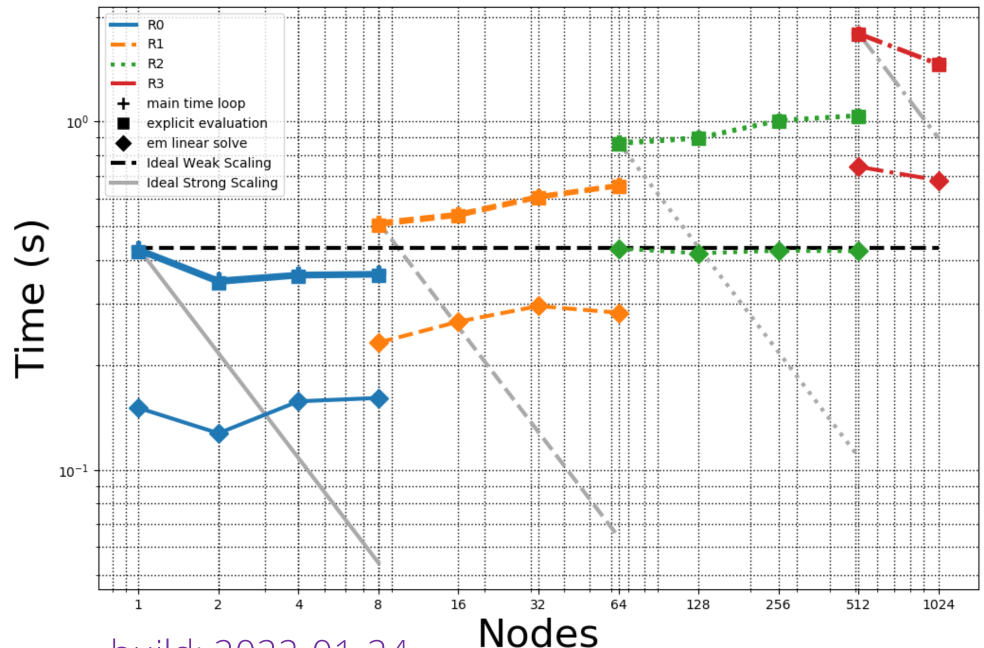
Toroidal reference problem Fluid scaling

Astra



build: 2022-01-21

Sierra



build: 2022-01-24