UNCLASSIFIED UNLIMITED RELEASE

Sandia National Laboratories

Exceptional service in the national interest

AN OVERVIEW OF THE TRILINOS CORE PRODUCTS

Roger Pawlowski

HPSF Conference 2025

May 5-8, 2025, Chicago, IL

U.S. DEPARTMENT of ENERGY NNSA

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.

SAND2025-05583C

CORE AREA PROVIDES FOUNDATIONAL CAPABILITIES FOR PACKAGES

- Performant data structures
- Abstraction layers (for linear algebra, linear solvers and nonlinear analysis)
- Utility libraries
- Common look and feel
- Includes “snapshotted” packages: Kokkos, Kokkos-Kernels and SEACAS

CORE AREA – PROVIDES BASIC CAPABILITIES

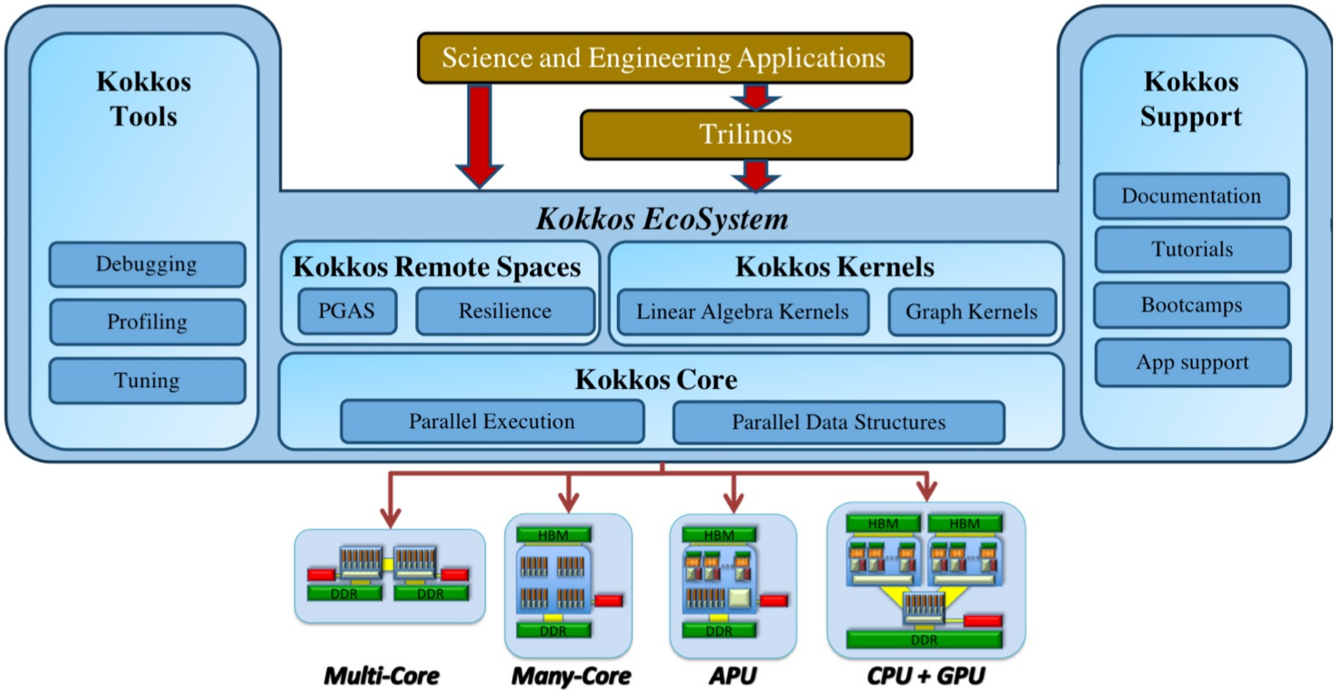

Roger Pawlowski

Alegbra/Numerics		
Kokkos (@kokkos)	**Performance portability library**	**Christian Trott** (@crtrott)
Kokkos Kernels (@kokkos-kernels)	**Optimized math kernels for Kokkos library**	**Luc Berger-Vergiat** (@lucbv)
RTOp (@rtop)	**Vector Reduction/Transformation OPerators**	**Ross Bartlett** (@bartlettroscoe)
Thyra (@thyra)	**Abstract numerical algorithms and interfaces**	**Ross Bartlett** (@bartlettroscoe)
Tpetra (@tpetra)	**Distributed linear algebra objects**	**Chris Siefert** (@csiefert2)
Utilities		
Galeri (@galeri)	**Matrix generation tools**	**Jonathan Hu** (@jhux2)
PyTrilinos2 (@pytrilinos2)	**Python bindings for Trilinos packages**	**Christian Glusa** (@cgcgcg)
SEACAS (@SEACAS)	**Utilities supporting finite element analysis**	**Greg Sjaardema** (@gsjaardema)
Teuchos (@teuchos)	**Common tools *(smart pointers, parameter lists, …)***	**Roger Pawlowski** (@rppawlo)
Meshing / Domain Decomposition		
Pamgen (@pamgen)	**Mesh generation tools for simple shapes**	**Roger Pawlowski** (@rppawlo)
Zoltan/Zoltan2 (@zoltan, @zoltan2)	**Load balancing and partitioning**	**Erik Boman** (@egboman)

PyTrilinos: Christian Glusa's talk @4:40pm

PERFORMANCE PORTABILITY: KOKKOS™

- Shared memory programming model in C++
- Write once, run on many architectures
- Thread scalable parallel patterns: for, reduce, scan, task
- Execution and Memory spaces
- Multidimensional array with compile-time polymorphic layouts
- Kokkos devs are on the c++ committee

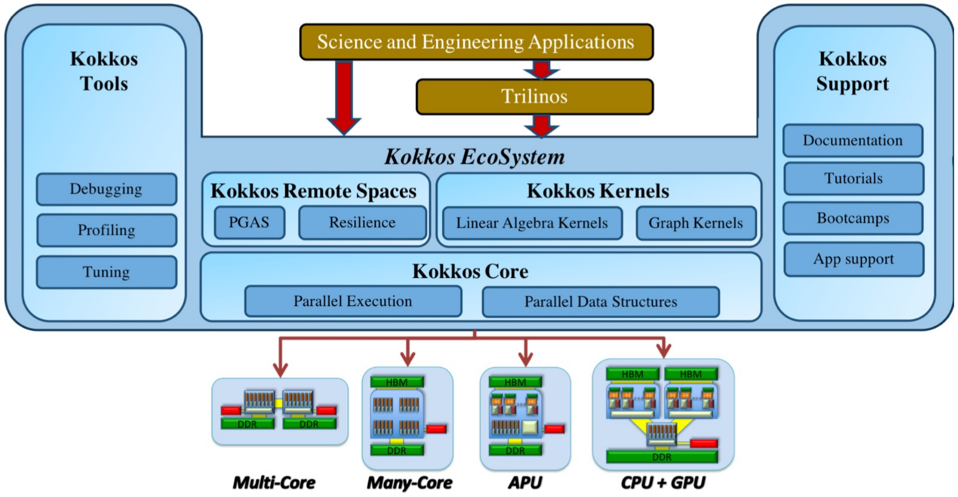


https://github.com/kokkos/kokkos

PERFORMANCE PORTABLE LINEAR ALGEBRA: KOKKOS-KERNELS

KokkosKernels provides math kernels for dense and sparse linear algebra as well as graph computations.

- Portable BLAS, Sparse and Graph kernels
- Generic implementations for various scalar types and data layouts
- Access to major vendor optimized math libraries.
- Expand the scope of BLAS to hierarchical implementations.
- Batched BLAS



PARALLEL SPARSE LINEAR ALGEBRA: TPETRA

Basic building blocks for Trilinos solvers and preconditioners

- **Provides**
  - Maps (distributed index array)
  - Sparse graphs, (block) sparse matrices, & dense vectors/multivectors
  - Norms, SpMV, SpGEMM, …
  - MPI communication & data (re)distribution
  - Fill: {Create,modify} {graph,matrix,vector}
- **Key features**
  - Can pick type of values (real, complex, automatic differentiation, ensemble, …)
  - Detailed performance monitoring (**see Chris Siefert's talk @4:20pm**)

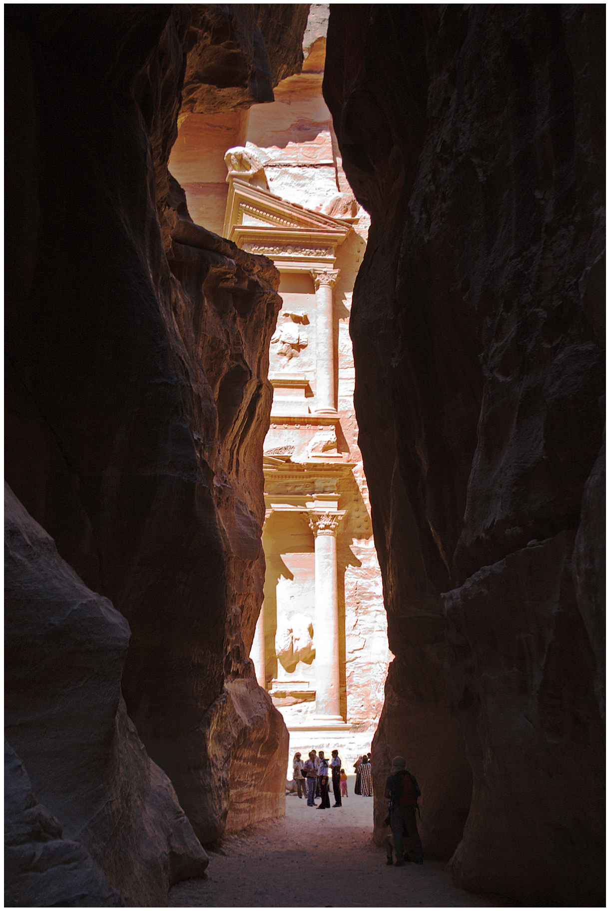
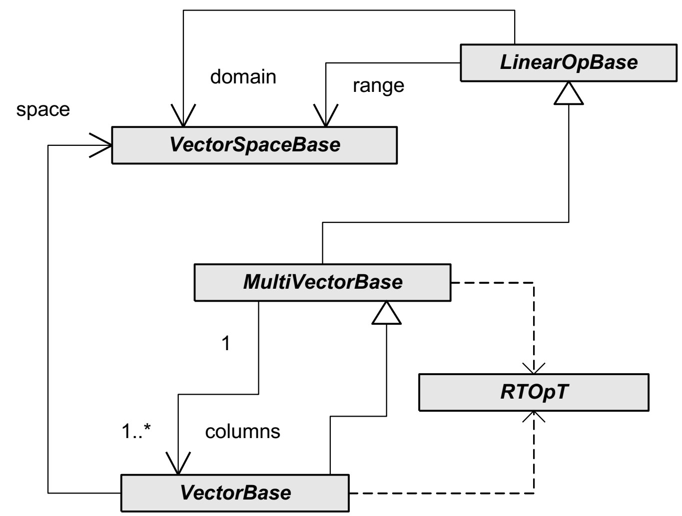


Image source:
https://en.wikipedia.org/wiki/Siq#/media/File:Petra_First_Glimpse.jpg

SOLVER ABSTRACTION LAYER: THYRA

- Abstractions for:
  - Vector/Operator (+Blocked wrappers)
  - Linear solvers/Preconditioners
  - Nonlinear solvers
  - Physics model interface (solver callback API)
- Matrix/Vector operations are handled in native data structures
  - Tpetra, Epetra, PETSC, …
- Used to write higher level packages/algorithms
  - Wrapper to all linear solver/preconditioners (**Stratimikos**)
  - Physics-based block preconditioners (**TEKO**)
  - Nonlinear solver (**NOX**)
  - Time integration (**Tempus**)
  - Optimization (**ROL**)
  - Bifurcation/Stability Analysis (**LOCA**)

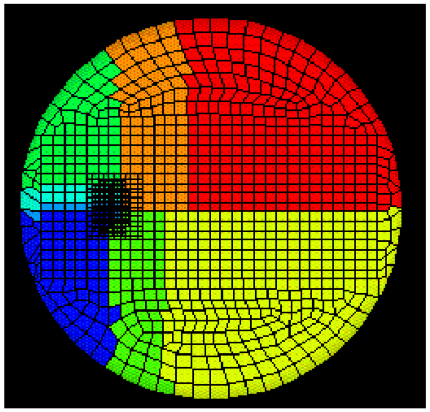


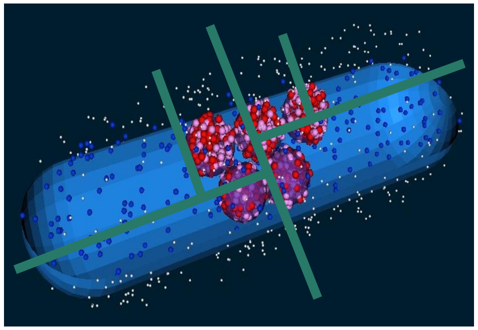
$$\begin{bmatrix} \mathbf{Q}_B & \mathbf{K}_E^B & 0 \\ \mathbf{K}_B^E & \mathbf{Q}_E & \mathbf{Q}_F^E \\ \mathbf{Q}_B^F & \mathbf{Q}_E^F & \mathbf{D}_F \end{bmatrix} \begin{bmatrix} \mathbf{B} \\ \mathbf{E} \\ \mathbf{F} \end{bmatrix}$$

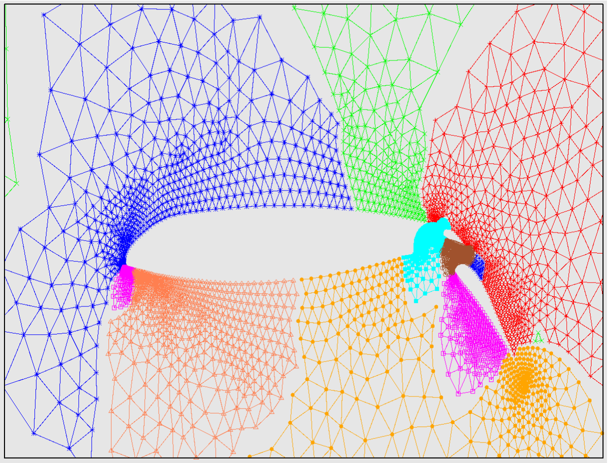
See Malachi Phillips's Talk @4pm

PARALLEL PARTITIONING AND LOAD BALANCING: ZOLTAN/ZOLTAN2

- **Zoltan** and **Zoltan2** are packages for partitioning, load balancing, and combinatorial scientific computing
  - Both have a collection of algorithms (geometric partitioning, graph, hypergraph, etc.)
- Zoltan is quite old (~2000) and can be used stand-alone on CPUs
- Zoltan2 is fairly new and a tightly integrated Trilinos package
  - Supports Kokkos and GPUs
  - Works with Tpetra matrices and vectors
  - Currently depends on Zoltan (though some algorithms have been reimplemented)
  - Has similar features/functionality to Zoltan but not exactly the same







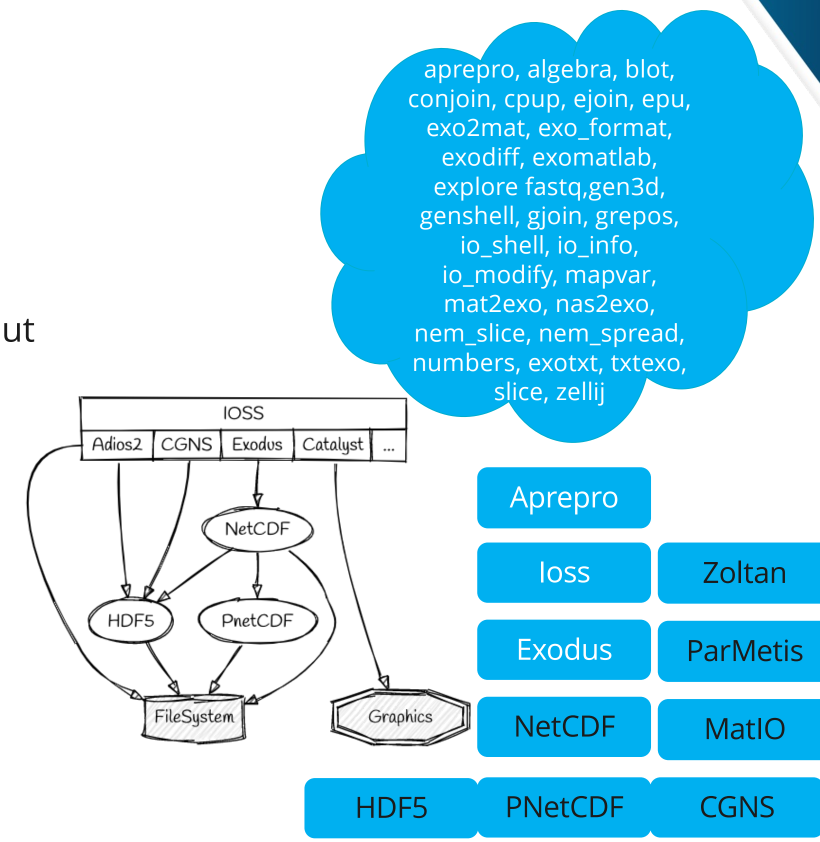
FINITE ELEMENT I/O TOOLS: SEACAS

- Provide IO for Sandia finite-element applications
- Provide applications and libraries for analyst workflow
- Support Production Analysts and Path-Forward Developers

SEACAS contains the **Exodus** and **IOSS** finite element input/output libraries; applications supporting meshing, preprocessing, postprocessing, graphics, database translation; and support libraries..

Exodus is a model developed to store and retrieve data for finite element analyses in a random access, machine independent, binary file that is written and read via C, C++, or Fortran APIs.

The **IO Subsystem** (IOSS) is a high-level database access API designed to give a format-agnostic interface to I/O capabilities with multiple backends outputting data to multiple formats. Supported input and output formats are **Exodus**, **CGNS**, HeartBeat, History, Generated, Pamgen, TextMesh, ADIOS2, FAODEL, Catalyst2, Null.

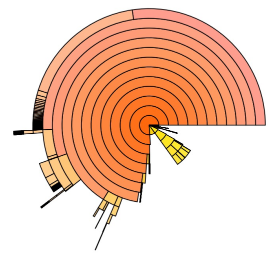


A supercomputer is a device for turning compute-bound problems into I/O-bound problems

TEUCHOS: COMMON PACKAGE UTILITIES (PRONOUNCED "TEFOS")

- Provides a common look and feel
- Capabilities:
  - Reference counted smart pointer (**RCP**)
  - **ParameterList**: runtime database (with yaml and xml I/O)
  - C++ Abstraction for an **MPI Communicator**
  - **Timers**
  - CPU BLAS/LAPACK wrappers
  - Scalar traits
  - Unit test harness
  - I/O Interface

```
NOX:
  Direction:
    Method: Newton
    Newton:
      Forcing Term Method: Constant
      Rescue Bad Newton Solve: true
      Stratimikos Linear Solver:
        NOX Stratimikos Options: { }
        Stratimikos:
          Linear Solver Type: Belos
```

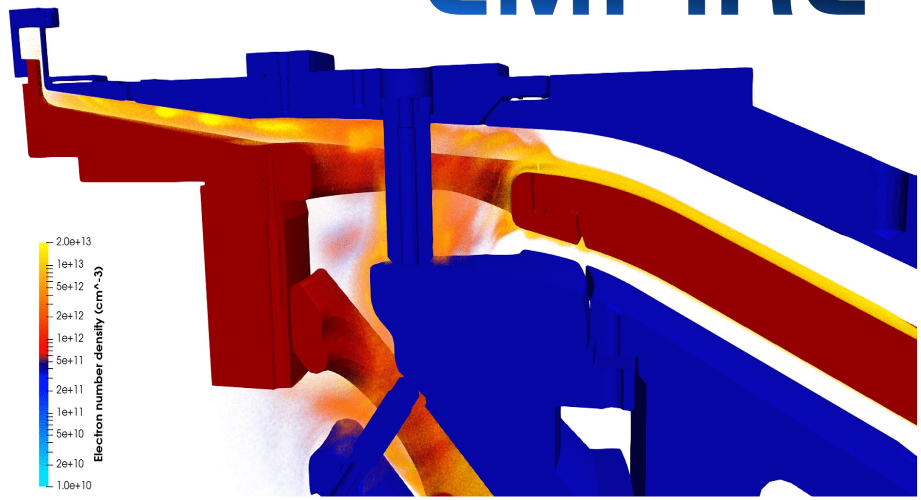


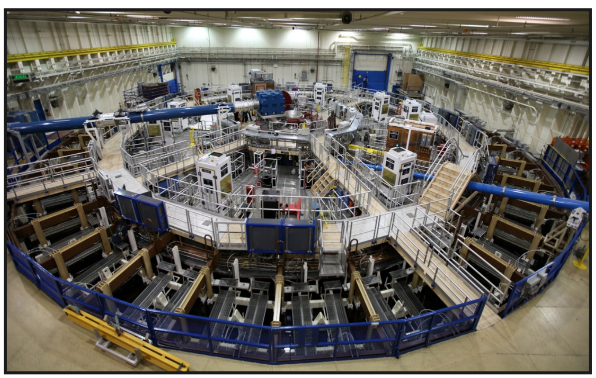
```
Panzer MixedPoisson Test: 4.39393 [1]
|   Mixed Poisson: 4.39382 - 99.9976% [1]
|   |   panzer::CubeHexMeshFactory::buildUncomittedMesh(): 0.000528984 - 0.0120393% [1]
|   |   panzer::CubeHexMeshFactory::completeMeshConstruction(): 0.765863 - 17.4305% [1]
|   |   panzer::DOFManagerFactory::buildUnqueGlobalIndexer: 0.0451156 - 1.0268% [1]
…
|   |   Ifpack2::Relaxation::initialize: 4.403e-06 - 0.000100209% [1]
|   |   Ifpack2::Relaxation::compute: 0.00249703 - 0.0568305% [1]
|   |   Belos: Operation Op*x: 0.00338377 - 0.0770121% [1]
|   |   Belos: PseudoBlockGmresSolMgr total solve time: 0.432027 - 9.83259% [1]
|   |   |   Belos: ICGS[2]: Orthogonalization: 0.0212883 - 4.92755% [31]
|   |   |   |   Belos: ICGS[2]: Ortho (Norm): 0.000489003 - 2.29705% [31]
|   |   |   |   Belos: ICGS[2]: Ortho (Inner Product): 0.00904903 - 42.507% [60]
|   |   |   |   Belos: ICGS[2]: Ortho (Update): 0.00662776 - 31.1333% [60]
|   |   |   |   Remainder: 0.00512255 - 24.0627%
|   |   |   Belos: Operation Prec*x: 0.0420421 - 9.73136% [31]
|   |   |   |   Ifpack2::Relaxation::apply: 0.0418983 - 99.658% [31]
|   |   |   |   Remainder: 0.000143774 - 0.341977%
|   |   |   Belos: Operation Op*x: 0.104452 - 24.1772% [31]
|   |   |   Remainder: 0.264244 - 61.1639%
…
```

EXAMPLE APPLICATION

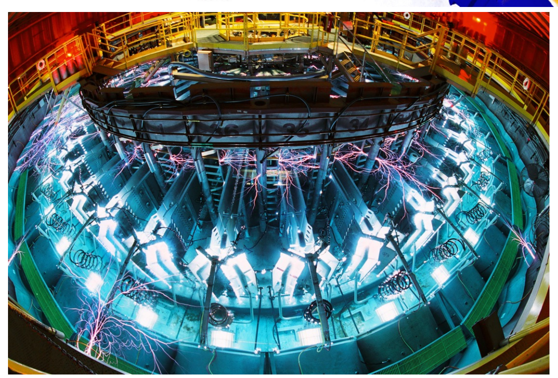
EMPIRE MODELS PLASMA EFFECTS IN HOSTILE NUCLEAR ENVIRONMENTS

- Empire is a plasma physics code used in weapon modernization work for hostile nuclear environments
- Simulation is necessary - test facilities can only access parts of parameter space
- Used extensively in conjunction with experimental facilities





Z: World's most powerful pulsed-power facility and x-ray generator



Saturn: World's brightest x-ray generator, tests electronics

HERMES-III: World's most powerful gamma simulator, for testing electronics and military systems

EMPIRE'S NUMERICAL APPROACH: FEM-PIC

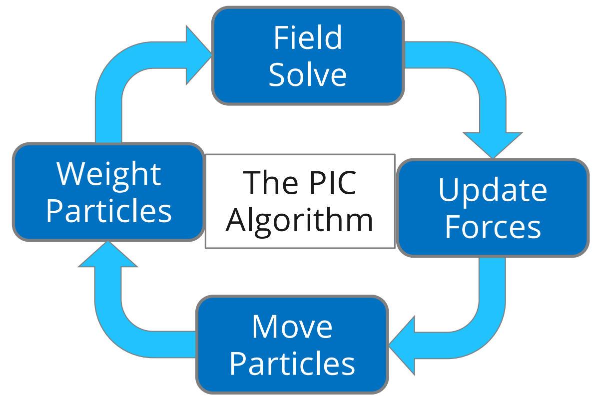
Empire's Particle-in-Cell (PIC) scheme[1] solves the equations of motion

$$\frac{d\boldsymbol{x}_p(t)}{dt} = \boldsymbol{v}_p(t),$$

$$\frac{d\boldsymbol{v}_p(t)}{dt} \equiv \boldsymbol{a}_p(t) = \frac{q_p}{m_p}\left[\boldsymbol{E}(t, \boldsymbol{x_p}(t)) + \boldsymbol{v}_p(t)\times\boldsymbol{B}(t, \boldsymbol{x_p}(t))\right]$$

encoded by the *plasma kinetic equation* for each particle (p):

$$\frac{df(t, \boldsymbol{x}, \boldsymbol{v})}{dt} \equiv \frac{\partial f}{\partial t} + \boldsymbol{v}\cdot\frac{\partial f}{\partial \boldsymbol{x}} + a\cdot\frac{\partial f}{\partial \boldsymbol{v}} = \left(\frac{\partial f}{\partial t}\right)_{coll}$$



Flow diagram showing a pure PIC timestep

with fields evolved according to the Maxwell curl equations:

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla\times\boldsymbol{E}$$

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{1}{\epsilon\mu}\nabla\times\boldsymbol{B} - \frac{1}{\epsilon}\boldsymbol{J}$$

$\xrightarrow[\text{solved in weak form}]{\text{solved in strong form}}$

$$\begin{pmatrix} \Delta t^{-1}\mathbb{I}_{\mathcal{F}} & \mathbb{K}_h \\ -\mathbb{K}_h^T\mathbb{M}_{\mathcal{F}}(\mu^{-1}) & \Delta t^{-1}\mathbb{M}_{\mathcal{E}}(\epsilon) \end{pmatrix} \begin{pmatrix} \Delta\mathbf{B} \\ \Delta\mathbf{E} \end{pmatrix} = -\begin{pmatrix} \mathbf{r}_{\mathbf{B}} \\ \mathbf{r}_{\mathbf{E}} \end{pmatrix}$$

where $\mathbb{K}$ = discrete curl operator, $\mathbb{M}$ = mass matrix, $\mathbb{I}$ = identity matrix, $\mathcal{F}$ and $\mathcal{E}$ label operators acting on face and edge spaces, h is the characteristic mesh dimension which labels quantities as spatially discretized versions of their continuum counterparts, Δt = timestep width, $\boldsymbol{r}$ = residuals

Using compatible discretization decisions[2,3] ensures the Maxwell divergence constraints are satisfied for all time

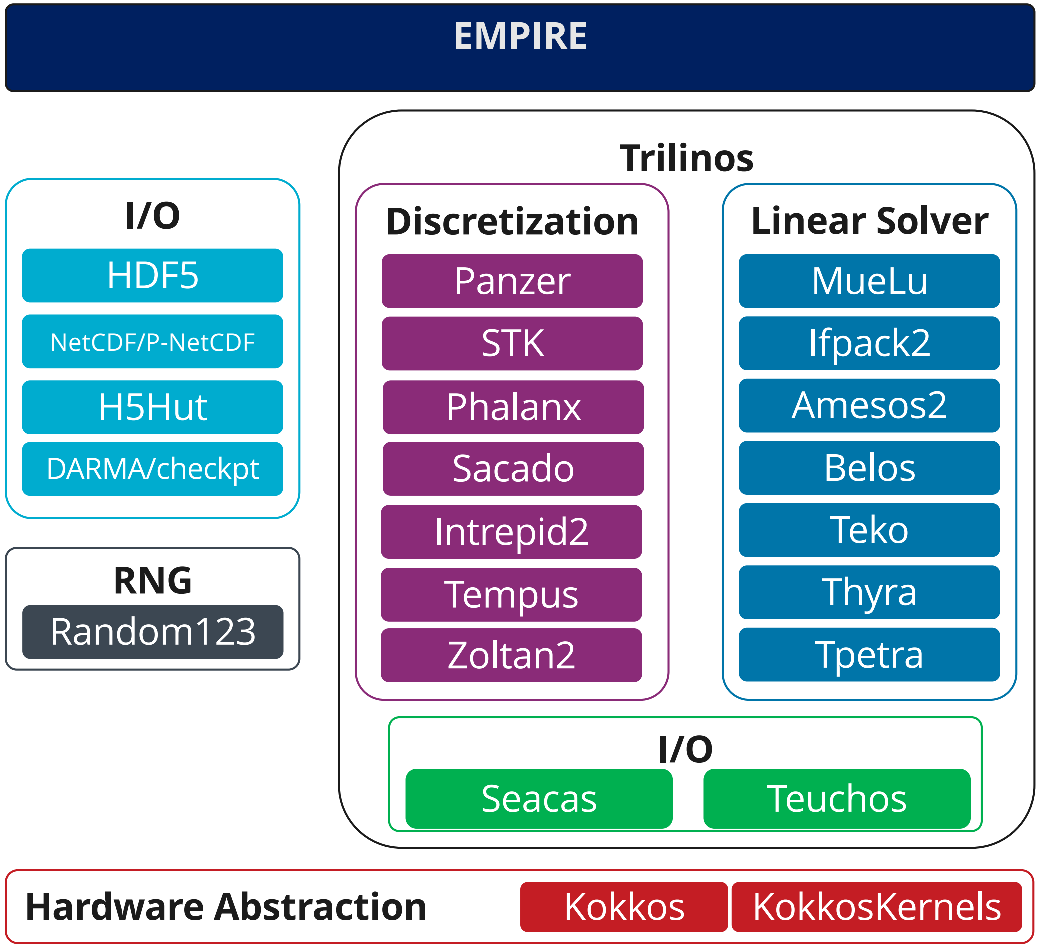
[1]Bettencourt, et. al., *EMPIRE-PIC: A Performance Portable Unstructured Particle-in-Cell Code*, 2021
[2]Nédélec J-C. *Mixed finite elements in* $\mathbb{R}^3$. Numerische Mathematik, 35(3):315–341, 1980.
[3]Raviart P-A. and Thomas, J. T. *A mixed finite element method for 2nd order elliptic problems*. Mathematical aspects of finite element methods, volume 606, pages 292–315. 1977

EMPIRE WAS DESIGNED ON A "COMPONENT" STRATEGY

- Directly depends on 21 Trilinos packages
- 38 total Trilinos packages with indirect dependencies
- Empire is built from the ground up for performance portability
- Requires large-scale iterative linear solve for electromagnetics

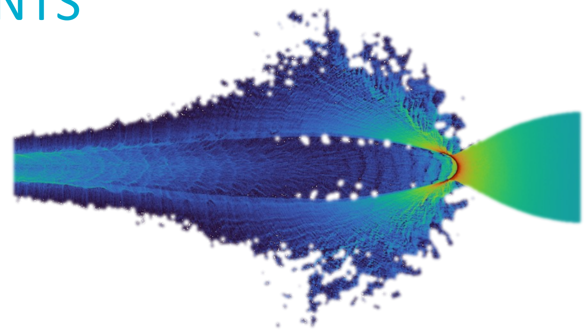
Linear Solvers: See Jonathan Hu's Talk @2:15pm

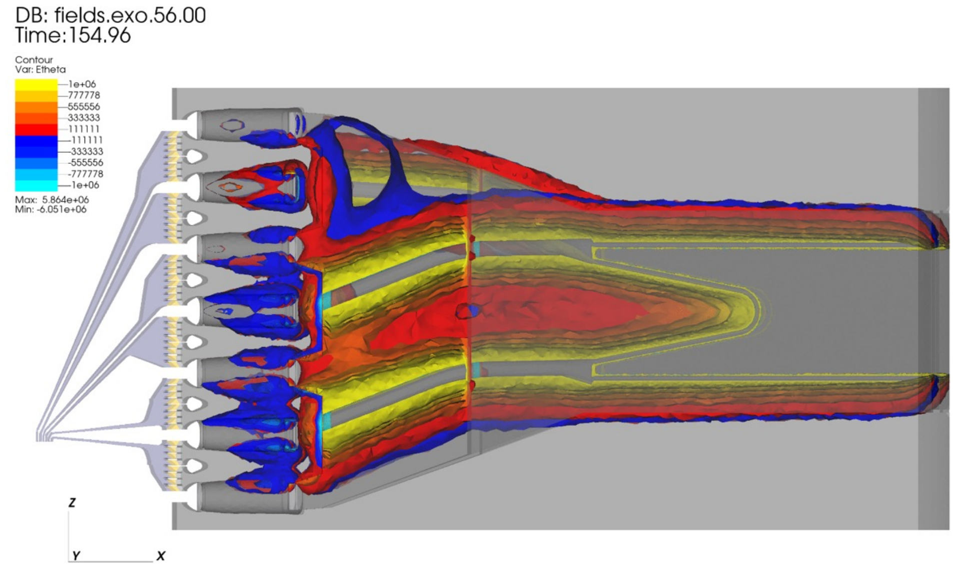
Discretization: See Mauro Perego's Talk @2:35pm



USE OF CORE AND DISCRETIZATION COMPONENTS

- **Kokkos** for underlying data structures of the entire code
- **Tpetra** for parallel distributed linear algebra
- Empire's (yaml) input file uses **Teuchos::ParameterList** for parsing and validation
- **SEACAS** is used to read and write mesh data
- **Zoltan** used to load balance the finite element mesh for parallel simulations
- **Thyra** used for blocked linear solves and coupling to circuits
- **Shards** for cell topology
- **Intrepid2** for finite element (FE) bases
- **Phalanx** for performance portability tools (view-of-views, virtual-function-on-device, …)
- **Panzer** for mesh tools, periodic bcs and FE assembly
- **STK** for runtime mesh database
- **Percept** for uniform mesh refinement





ELECTROMAGNETIC SOLVER USING COMPATIBLE FINITE ELEMENTS

$$\begin{pmatrix} \Delta t^{-1}\mathbb{I}_{\mathcal{F}} & \mathbb{K}_h \\ -\mathbb{K}_h^T \mathbb{M}_{\mathcal{F}}(\mu^{-1}) & \Delta t^{-1}\mathbb{M}_{\mathcal{E}}(\epsilon) \end{pmatrix} \begin{pmatrix} \Delta \mathbf{B} \\ \Delta \mathbf{E} \end{pmatrix} = - \begin{pmatrix} \mathbf{r_B} \\ \mathbf{r_E} \end{pmatrix}$$

$$\mathbf{B} \in \mathbf{H}_{\nabla .}(\Omega)$$

$$\mathbf{E} \in \mathbf{H}_{\nabla \times}(\Omega)$$

Block LU Decomposition

$$\begin{pmatrix} \Delta t^{-1}\mathbb{I}_{\mathcal{F}} & \mathbb{K}_h \\ -\mathbb{K}_h^T \mathbb{M}_{\mathcal{F}}(\mu^{-1}) & \Delta t^{-1}\mathbb{M}_{\mathcal{E}}(\epsilon) \end{pmatrix} = \begin{pmatrix} \mathbb{I}_{\mathcal{F}} & 0 \\ -\Delta t \mathbb{K}^T \mathbb{M}(\mu^{-1}) & \mathbb{I}_{\mathcal{E}} \end{pmatrix} \begin{pmatrix} \Delta t^{-1}\mathbb{I}_{\mathcal{E}} & \mathbb{K}_h \\ 0 & \mathbb{S}_{\mathcal{E}} \end{pmatrix}$$

Assemble Schur Compliment $\mathbb{S}_{\mathcal{E}} = \Delta t^{-1}\mathbb{M}_{\mathcal{E}}(\epsilon) + \Delta t \mathbb{K}_h^T \mathbb{M}(\mu^{-1})\mathbb{K}_h$
as monolithic matrix

Solve for dE with PCG: $\mathbb{S}_{\mathcal{E}} \Delta \mathbf{E} = -\mathbf{r_E} + \Delta t \mathbb{K}_h^T \mathbb{M}(\mu^{-1}) \mathbf{r_B}$

Explicit back solve for dB: $\Delta \mathbf{B} = -\Delta t \mathbb{K}_h \Delta \mathbf{E} - \Delta t \mathbf{r_B}$

Meshing:
STK, Percept, SEACAS, Panzer

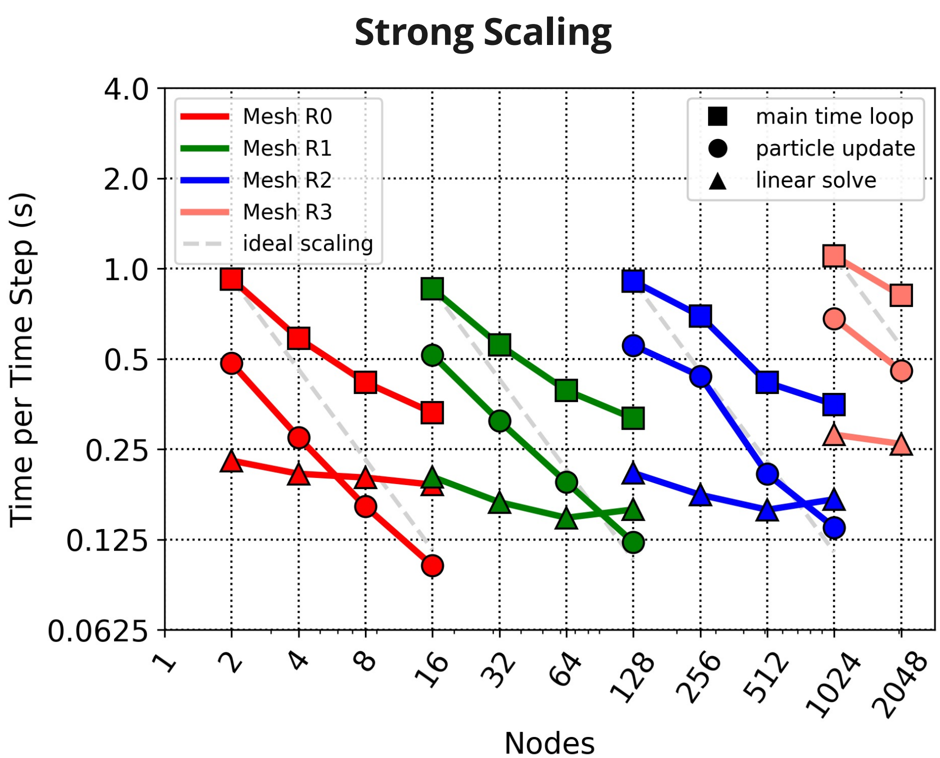
Data Structures:
Kokkos, KokkosKernels, Tpetra

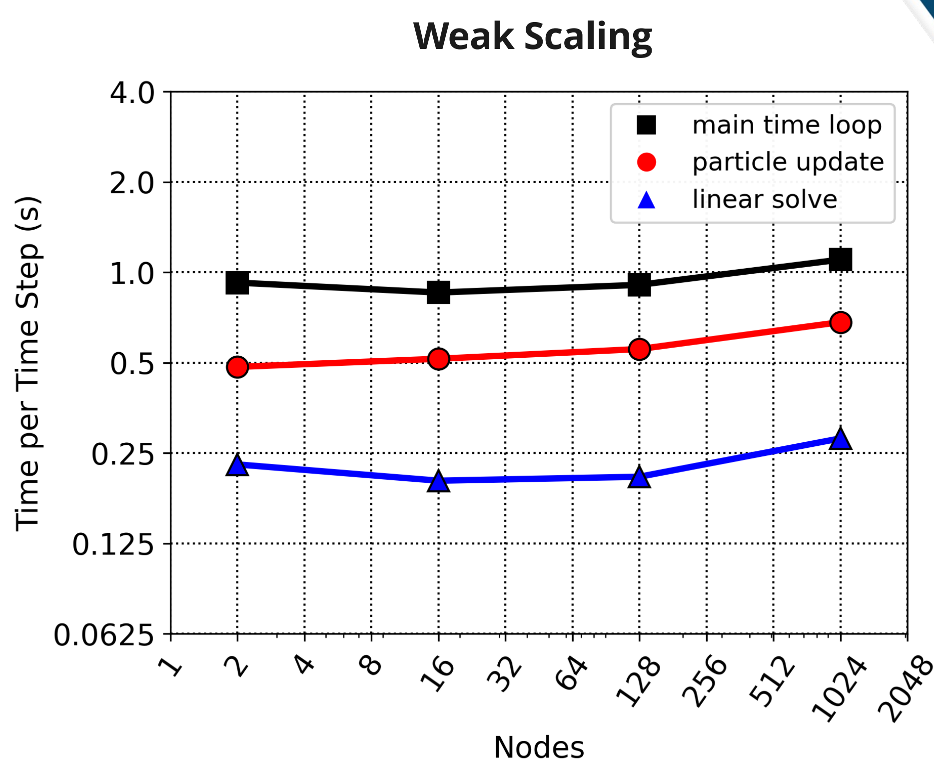
FE Assembly:
Shards, Intrepid2, Panzer, Thyra

Linear Solve:
- Uses RefMaxwell AMG with Conjugate Gradient
- Chebyshev smoother
- Prec setup done once
- Belos, Teko, MueLu, Ifpack2, Amesos2, KokkosKernels, Zoltan2

Bettencourt, et. al., *EMPIRE-PIC: A Performance Portable Unstructured Particle-in-Cell Code*, 2021
Lourenco Beirao de Veiga, Konstantin Lipnikov, and Marco Manzini, *Mimetic Finite Difference Method for Elliptic Problems*.
Bochev et al., *An algebraic multigrid approach based on a compatible gauge reformulation of Maxwell's equations*, 2008.

SCALABILITY ON CAVITY PROBLEM ON ATS-2 SIERRA (4 NVIDIA V100 GPUS/NODE)



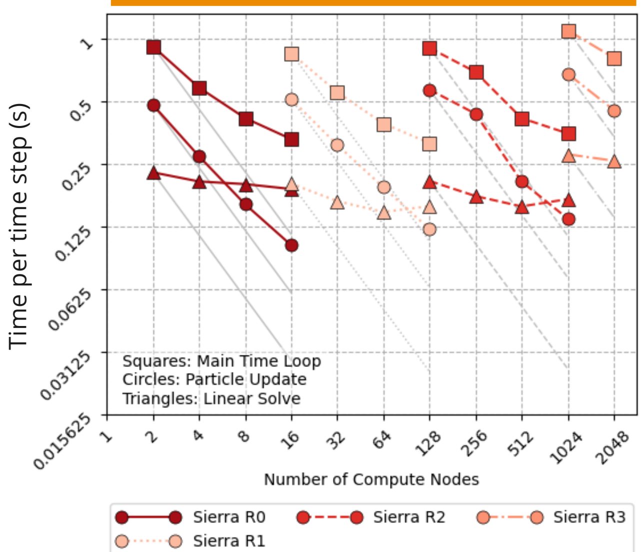


- Largest Simulations: 1.6B Elements and 160B particles!

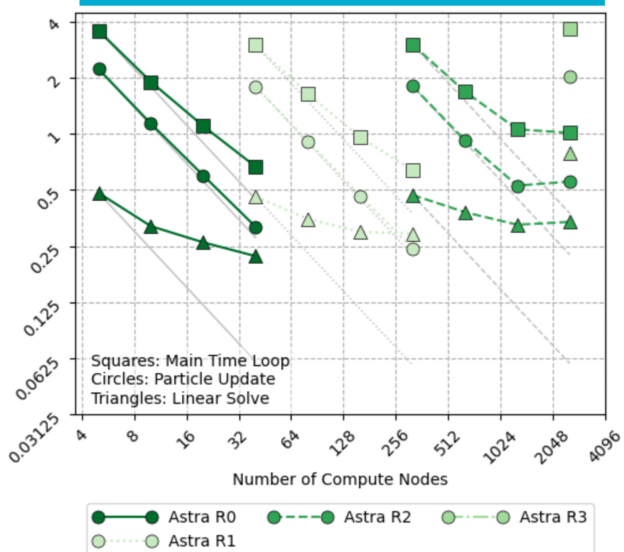
PERFORMANCE RESULTS FOR GENERIC CAVITY

Single platform scaling results

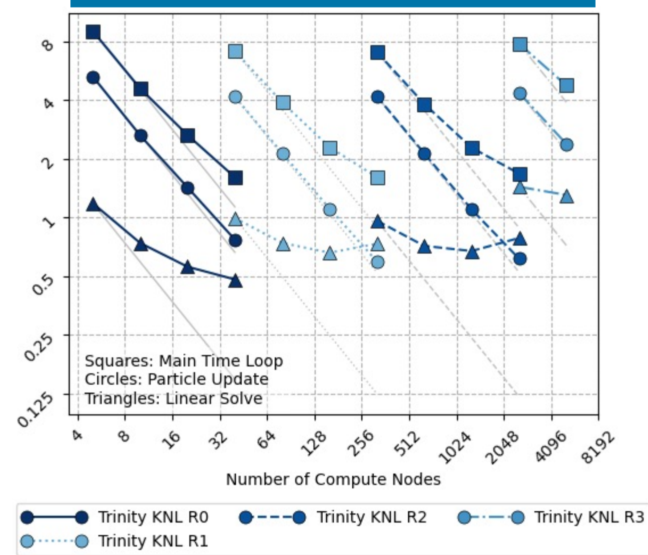
Sierra
4 GPU/socket; 1 MPI/GPU
Up to 2048 nodes (47% of total)



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 ON AMD GPUS (MI250X)

EAS3: 36 nodes available, will show results up to 32 nodes

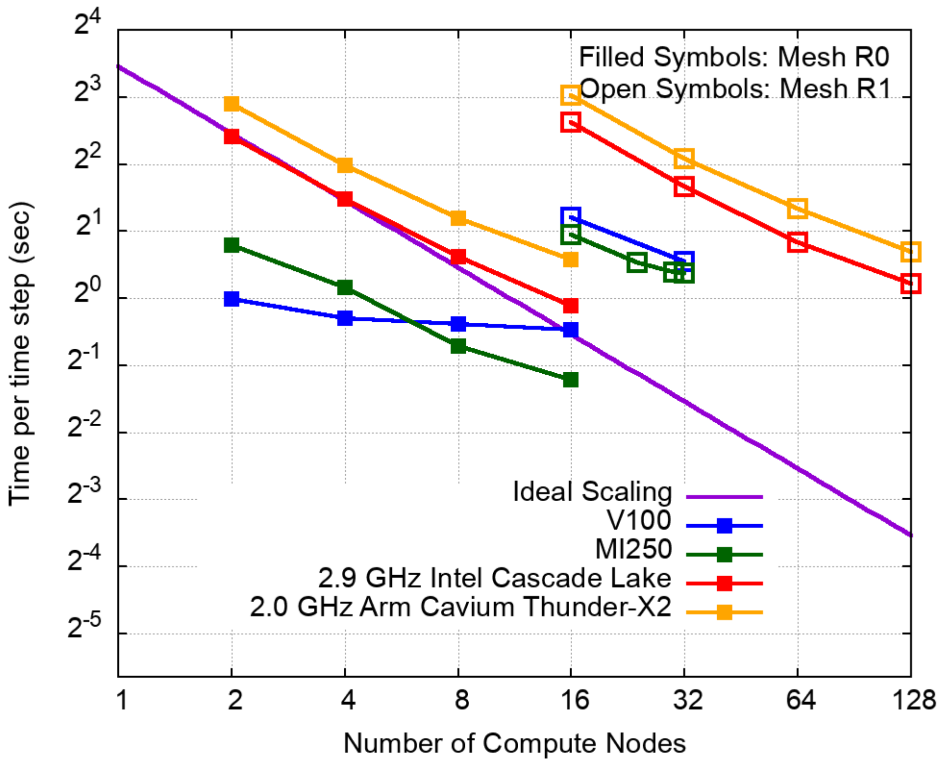
Machines:

- **EAS3**: 4 AMD MI250x GPUs per node
- **CTS-1**: Intel Cascade Lake 48 cores per node
- **ATS-2**: 4 NVIDIA V100 GPUs per node
- **Vanguard**: 2.0 GHz ARM Cavium Thunder-X2, 56 cores per node

System	Peak FLOPs	STREAM Bandwidth
Single MI250x	48 TF	2.68 TB/s
Single V100	7 TF	855 GB/s
Intel Cascade Lake node	4.45 TF	208 GB/s
Arm Cavium Thunder X2 node	896 GF	250 GB/s

Two meshes, uniformly refined:

Mesh	Elements	Nodes	Edges	Particles
R0	4.6M	791K	220K	170M
R1	36.8M	6.2M	883K	1.35B

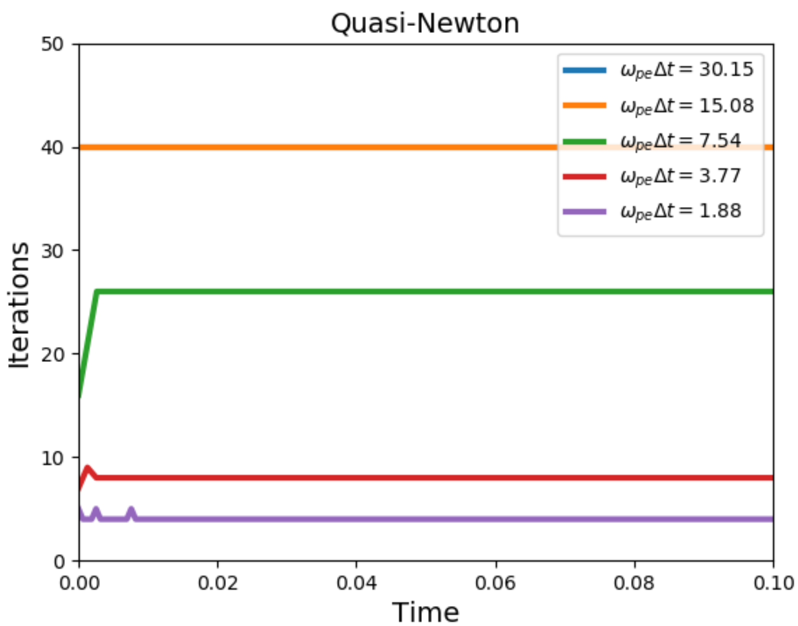


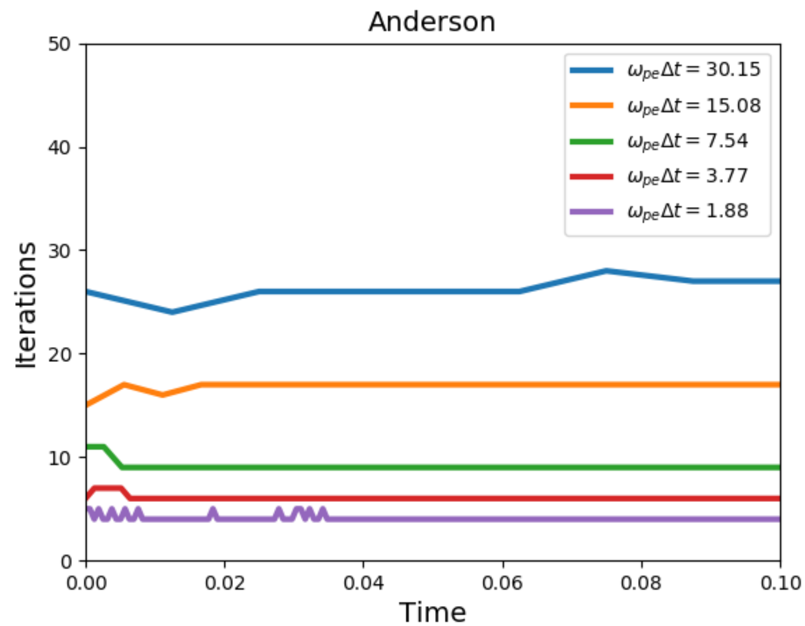
- **Default linear solve used on EAS3. No work to date on optimizing the linear solve on EAS3!**
- Other platforms all have linear solves that were hand tuned/optimized for the hardware.

ANDERSON ACCELERATION IMPROVES PERFORMANCE

- Hybrid Two fluid plasma vortex in MHD limit
- IMEX time discretization, DG fluid discretization, coupled to CG Maxwell discretization
- Using Schur-Complement in all simulations

- Fluid block diagonal solver: **KokkosKernels**: KokkosBatched_LU and KokkosBatched_Trsm
- Coupled outer solve: **NOX** for Anderson Acceleration





AN INCOMPLETE LIST OF CONTRIBUTORS

Trilinos Core Area

- Chris Baker
- Ross Bartlett
- Erik Boman
- Karen Devine
- Christian Glusa
- Roger Pawlowski
- Chris Siefert
- Greg Sjaardema
- Heidi Thornquist
- Jonathan Hu
- Michael Heroux
- Daniel Ibanez
- Mark Hoemmen
- Kevin Long
- Christian Trott
- Damien Lebrun-Grandie
- Siva Rajamanikan
- Luc Berger-Vergiat

Empire

- Matt Bettencourt
- Keith Cartwright
- Eric Cyr
- Adam Darr
- James Elliott
- Daniel Jensen
- Todd Kordenbrock
- Richard Kramer
- Caleb Logeman
- Ed Love
- William McDoniel
- Duncan McGregor
- Logan Meredith
- Chris Moore
- Roger Pawlowski
- Edward Phillips
- Bryan Reuter
- Sid Shields
- David Sirajuddin
- Shane Stafford
- Scot Swan

And many more…

SUMMARY

- Trilinos provides powerful tools for modeling complex physical systems
- The "Core" product area supplies foundational building blocks for other packages
- Trilinos enables scalable large-scale simulations on multiple architectures

Advanced Simulation & Computing™

ECP Exascale Computing Project

U.S. Department of Energy | Office of Science

SciDAC Scientific Discovery through Advanced Computing

The Linux Foundation

HPSF High Performance Software Foundation