Sandia National Laboratories

Exceptional service in the national interest

Scalability and Performance of the Empire Plasma Physics Code on the El Capitan Platform

Roger Pawlowski, David Sirajuddin, Christopher Moore, Keith Cartwright, James Elliott, Christian Glusa, Jonathan Hu and Richard Kramer

HPSF Conference 2026, Chicago, IL

March 19th, 2026

EL CAPITAN

ADVANCED SIMULATION & COMPUTING

Unclassified Unlimited Release

SAND2026-17946C

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.

U.S. DEPARTMENT OF ENERGY

NNSA

Outline

1. The Empire Code
2. Results
3. Challenges
4. Summary

Acknowledgements

- HPE: Luke Roskop, Kostas Makrides, Srinath Vadlamani
- AMD: Austin Ellis, Ian Bogle, Kevin Huck
- COE/LLNL/SNL: Judy Hill, Scott Futral, Greg Tomaschke, Joel Stevenson

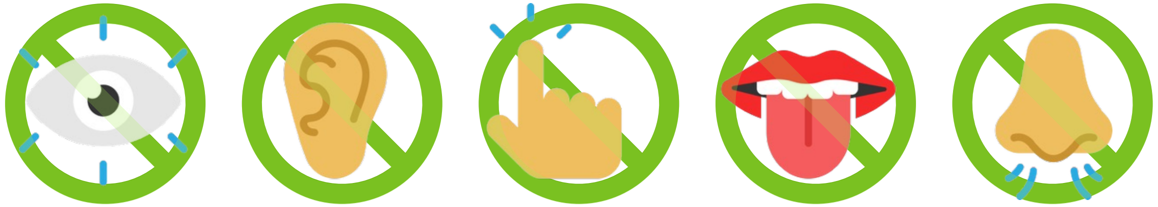
RADIATION & ELECTROMAGNETIC ENVIRONMENTS

INVISIBLE THREATS, REAL CONSEQUENCES

Mission success relies on systems being able to survive in these extreme environments.

INVISIBLE THREATS

Most forms of radiation and electromagnetic environments (e.g., x-rays, neutrons, gamma, electromagnetic pulses, etc.) are imperceptible to human senses.

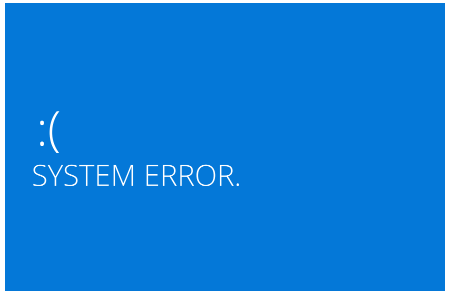


~~See~~ ~~Hear~~ ~~Touch~~ ~~Taste~~ ~~Smell~~

REAL CONSEQUENCES

These invisible threats can have *severe consequences* for electronic systems and materials.

Errors & Malfunctions



Physical Damage

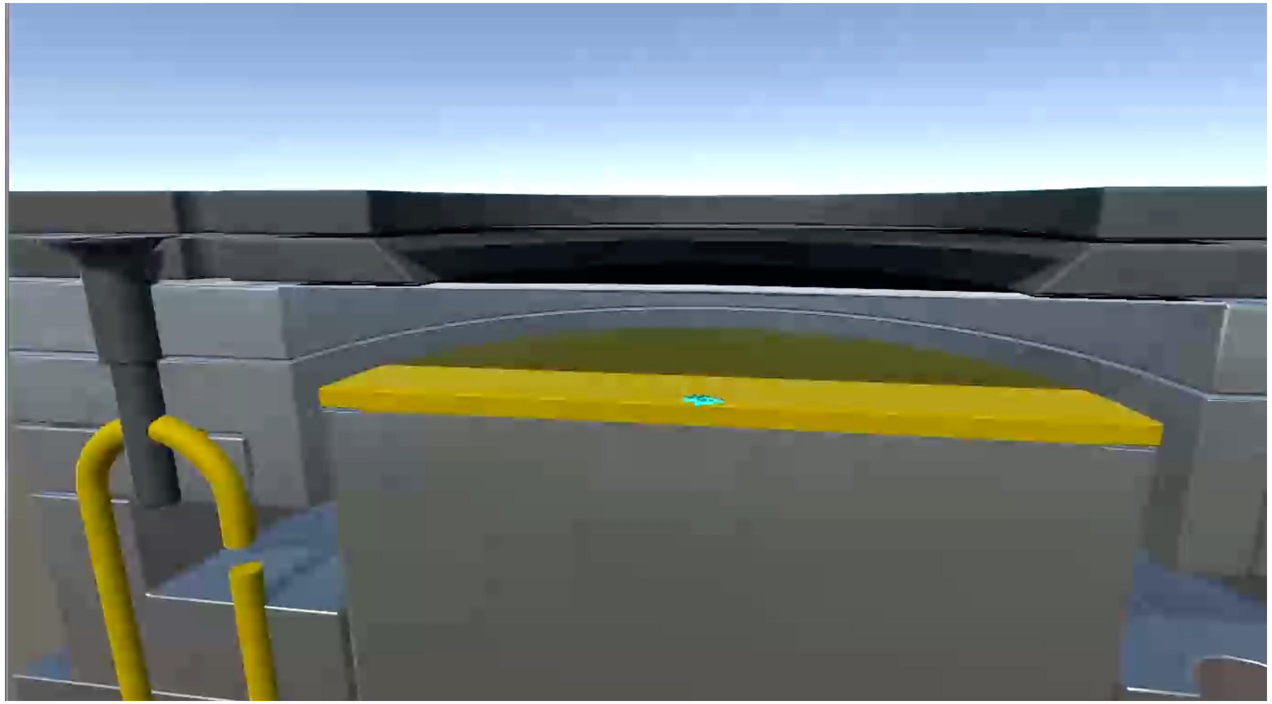
Disabled Systems

Sandia partners to qualify and design for radiation and electromagnetic effects on electronics and materials –
illuminating risks and ensuring mission success.

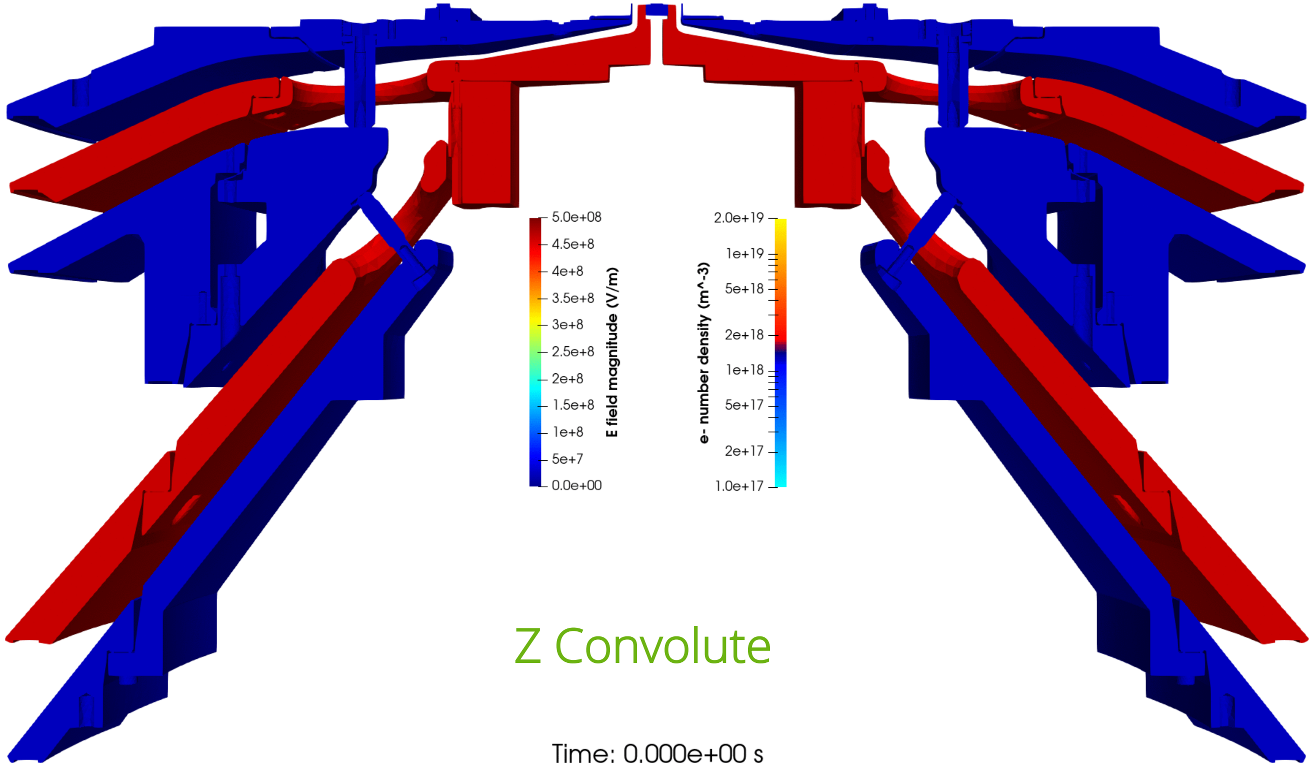
Example: BDot Experiment

- End irradiated cavity
- A standard diagnostic experiment fielded at X-ray facilities like Z and NIF
- X-rays strike the surface and generate a plasma
- Measures induced currents from the field induced by the plasma

Animation of a Bdot Experiment



The EMPIRE Plasma Simulation Code



- Empire is a next-gen plasma simulation code
- Self-consistently couples plasma dynamics with electromagnetic response
- Includes suites for additional physics
  - Collisions: ionization, excitation, elastic scattering
  - Circuit coupling, radiation drive (photo-electrons), & more

What is Empire?

Empire solves time-dependent plasma dynamics in the "void" or gas region (not inside solids) via the coupled Boltzmann Equation and Maxwell's Equations:

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

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

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

Empire uses a Finite Element Monte Carlo method to solve this system of equations

- Unstructured Finite Element (FEM): Field solve Maxwell's Equations
- Particle-In-Cell (PIC): Evolve a collection of macro-particles self-consistently w/EM field
- Direct Simulation Monte Carlo (DSMC): Evolve plasma density, momentum via collisions
- Tracks neutrals, electrons, ions, excited states, and photons as computational particles
- Solves for **E** & **B** fields and 1D surface temperature

What is Empire?

Empire solves time-dependent plasma dynamics in the “void” or gas region (not inside solids) via the coupled Boltzmann Equation and Maxwell’s Equations:

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

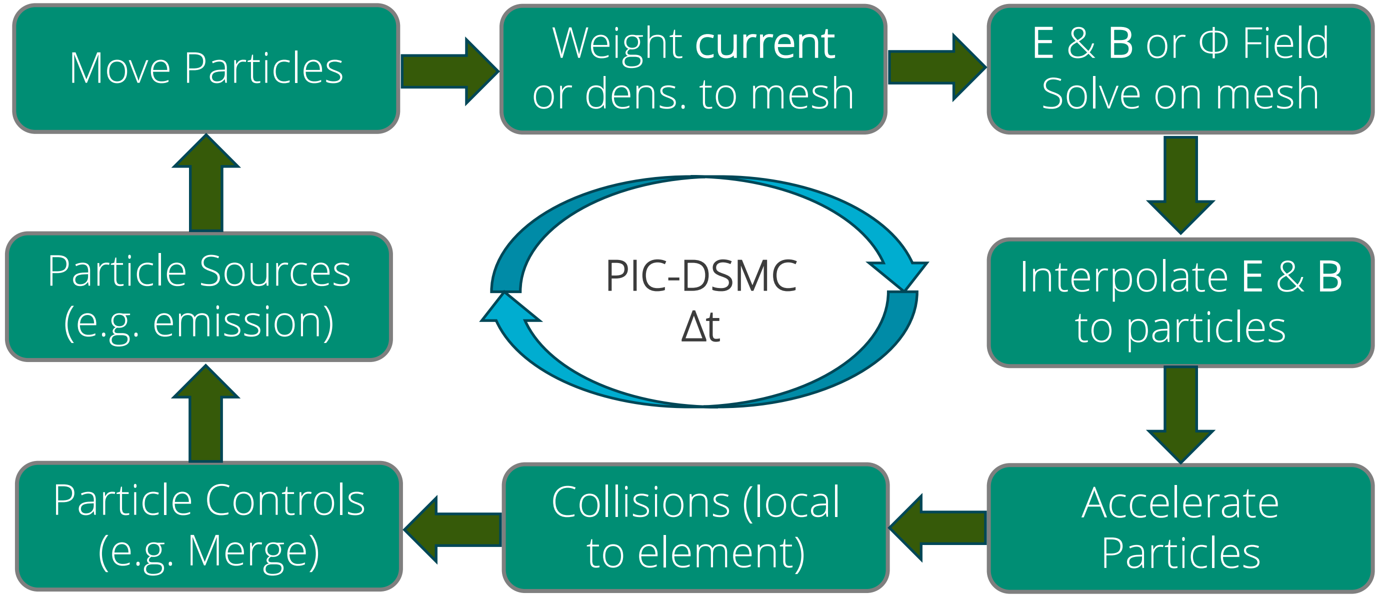
$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E}$$

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

Empire uses a Finite Element Monte Carlo method to solve this system of equations

- Unstructured Finite Element (FEM): Field solve Maxwell’s Equations
- Particle-In-Cell (PIC): Evolve a collection of macro-particles self-consistently w/EM field
- Direct Simulation Monte Carlo (DSMC): Evolve plasma density, momentum via collisions
- Tracks neutrals, electrons, ions, excited states, and photons as computational particles
- Solves for **E** & **B** fields and 1D surface temperature

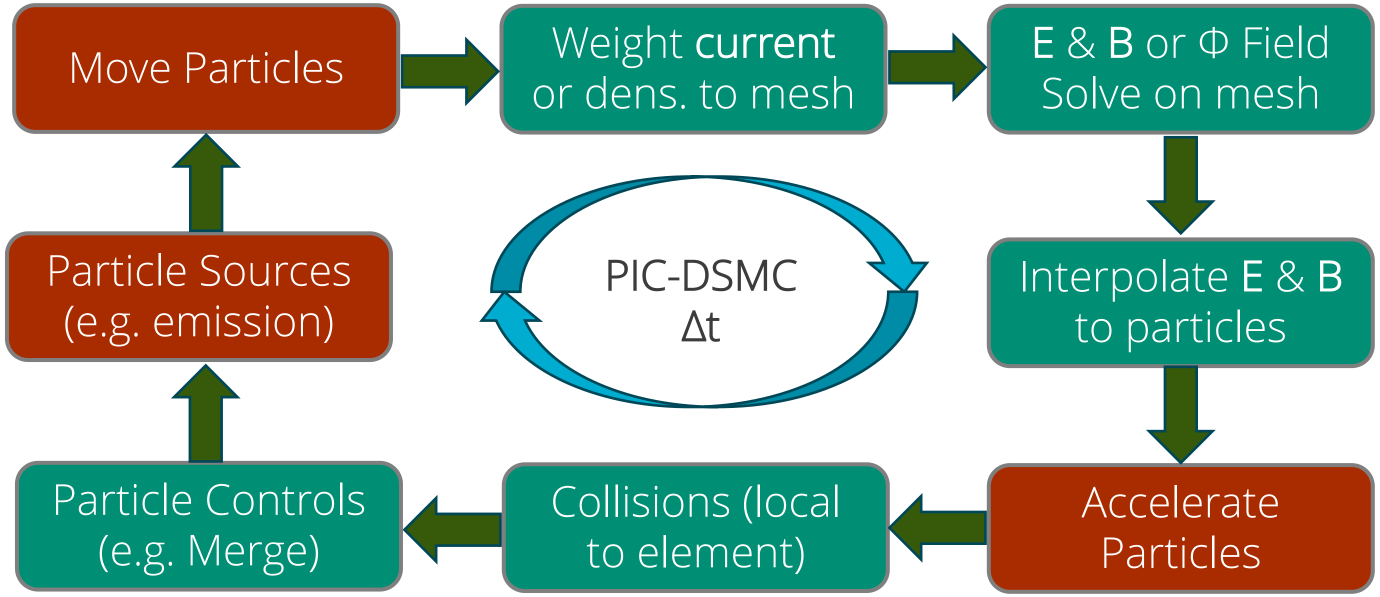
Empire numerical approach: FEM-PIC-DSMC



Empire numerical approach: FEM-PIC-DSMC

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f}{\partial \boldsymbol{x}} + \frac{q}{m}(\boldsymbol{E} + v \times \boldsymbol{B}) \cdot \frac{\partial f}{\partial \boldsymbol{v}} = \left(\frac{\partial f}{\partial t}\right)_{coll}$$

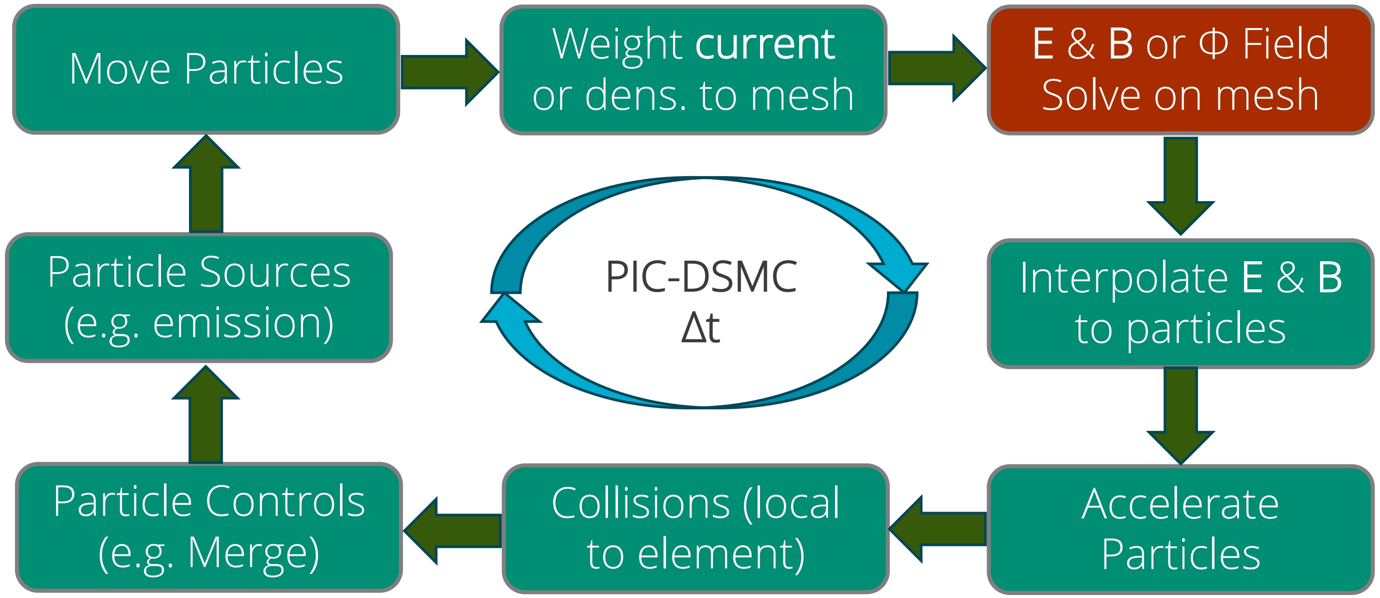
$$\frac{d\boldsymbol{x}_p(t)}{dt} = \boldsymbol{v}_p(t),$$



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

Empire uses Monte Carlo computational particles to evolve the density and momentum of the elements

Empire numerical approach: FEM-PIC-DSMC



$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla\times\boldsymbol{E} \qquad \mathbf{B} \in \mathbf{H}_{\nabla\cdot}(\Omega)$$

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{1}{\epsilon\mu}\nabla\times\boldsymbol{B} - \frac{1}{\epsilon}\boldsymbol{J} \qquad \mathbf{E} \in \mathbf{H}_{\nabla\times}(\Omega)$$

solved in strong form

solved in weak form

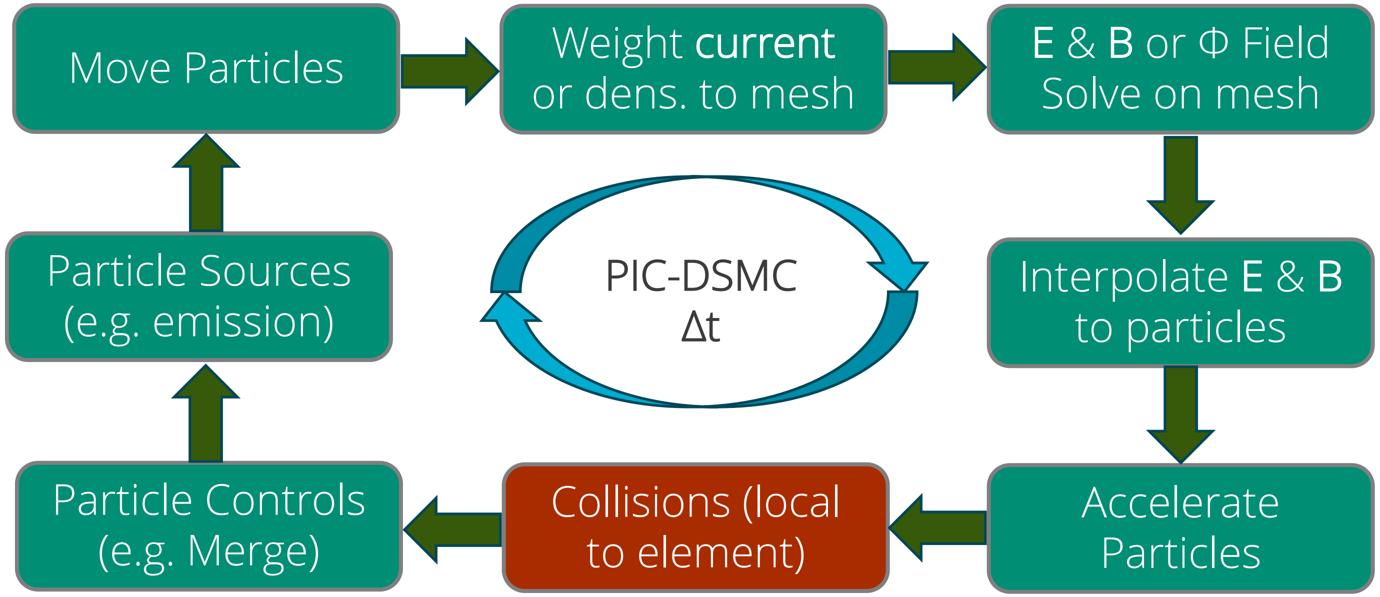
$$\begin{pmatrix} \Delta t^{-1}\mathbb{I}_{\mathcal{F}} & \mathbb{K}_h \\ -\mathbb{K}_h^T\mathbb{M}_{\mathcal{F}}(\mu^{-1}) & \Delta t^{-1}\mathbb{M}_{\mathcal{E}}(\epsilon) \end{pmatrix} \begin{pmatrix} \Delta\mathbf{B} \\ \Delta\mathbf{E} \end{pmatrix} = -\begin{pmatrix} \mathbf{r}_{\mathbf{B}} \\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

Using compatible discretization decisions ensures the Maxwell divergence constraints are satisfied for all time

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 size, $\boldsymbol{r}$ = residuals

Empire numerical approach: FEM-PIC-DSMC

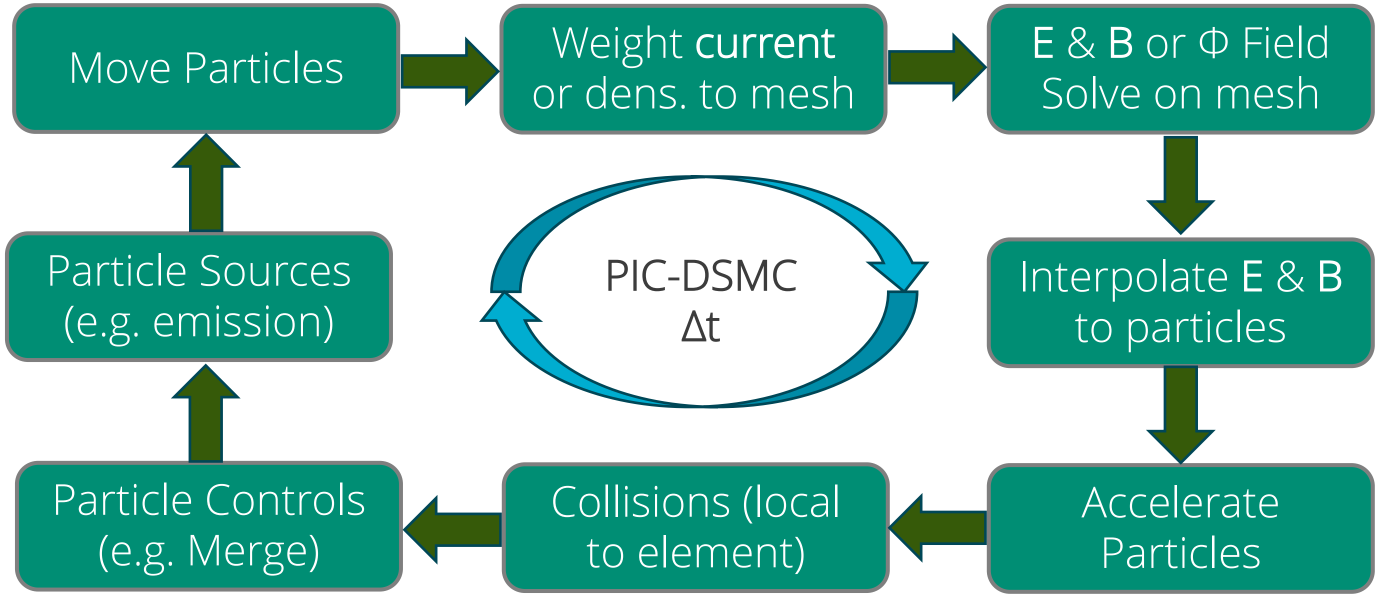
$$\frac{\partial f}{\partial t} + \boldsymbol{v}\cdot\frac{\partial f}{\partial \boldsymbol{x}} + \frac{q}{m}(\boldsymbol{E} + v\times\boldsymbol{B})\cdot\frac{\partial f}{\partial \boldsymbol{v}} = \left(\frac{\partial f}{\partial t}\right)_{coll}$$



$$\left(\frac{\partial f}{\partial t}\right)_{coll} = \iint gI(g,\Omega)[f(\boldsymbol{r},\boldsymbol{p}'_A,t)f(\boldsymbol{r},\boldsymbol{p}'_B,t) - f(\boldsymbol{r},\boldsymbol{p}_A,t)f(\boldsymbol{r},\boldsymbol{p}_B,t)]d\Omega d^3\boldsymbol{p}$$

Empire uses Monte Carlo collisions via computational particles to evaluate the (2-body) collision integral.

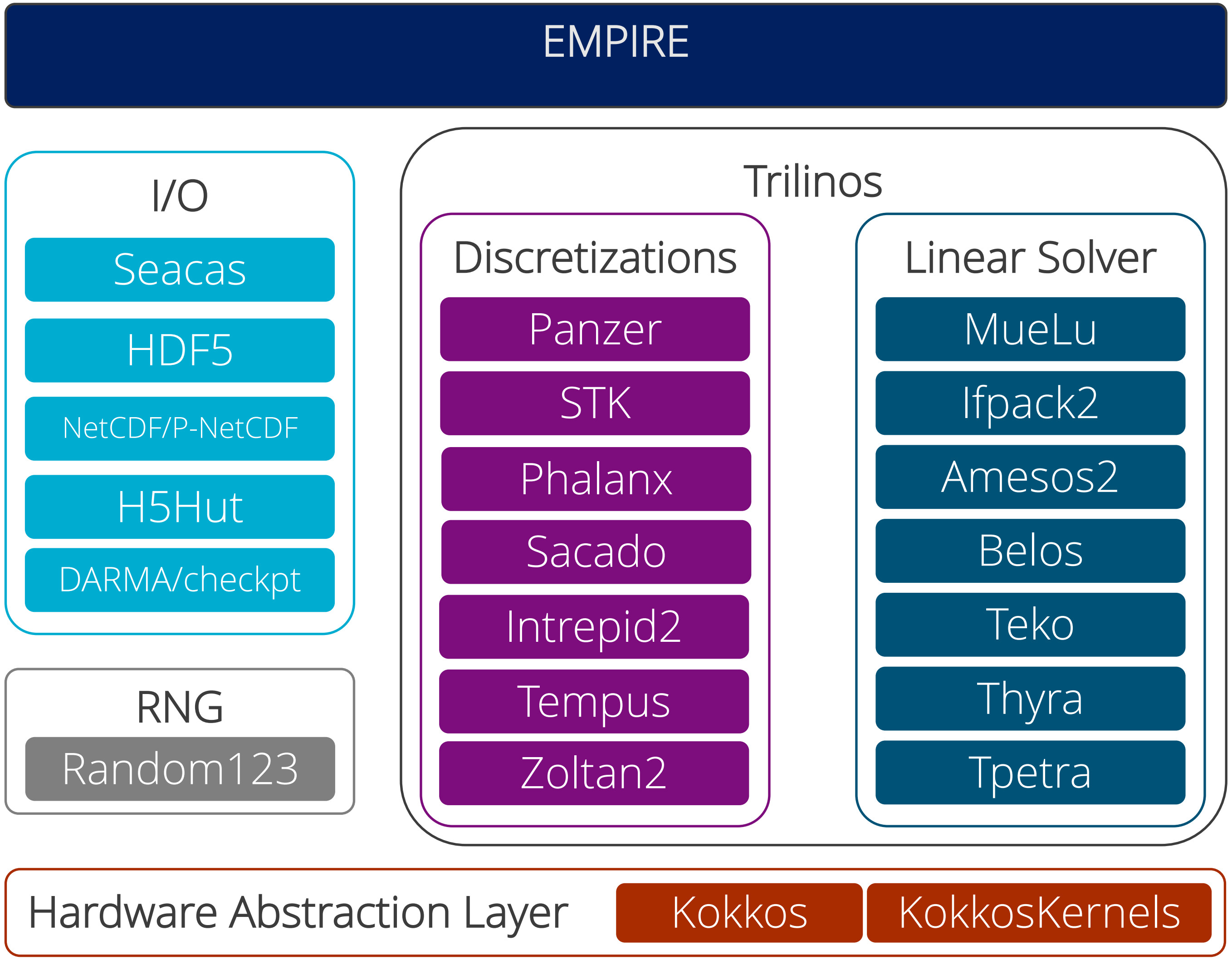
Empire numerical approach: FEM-PIC-DSMC



For GPUs, this involves a lot of kernel launches!

Empire relies heavily on external software libraries

- Chose a “components” heavy philosophy
- Leverages ASCR, ATDM and Exascale Computing Project investments
- Programmatically, it makes sense to share code, but a large number of dependencies means we are not always the most nimble with new architectures



Advanced Simulation & Computing™

Exascale Computing Project

The Linux Foundation

U.S. Department of Energy | Office of Science

SciDAC Scientific Discovery through Advanced Computing

EMPIRE's Electromagnetic Solver

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

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

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

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

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: $\Delta \mathbf{B} = -\Delta t \mathbb{K}_h \Delta \mathbf{E} - \Delta t \mathbf{r_B}$

Linear solve is latency bound for most problems

Meshing:
STK, Percept, SEACAS, Panzer

Data Structures:
Kokkos, KokkosKernels, Tpetra

Assembly:
Shards, Intrepid2, Panzer, Thyra

Linear Solve:
- Conjugate Gradient Solver
- RefMaxwell AMG Prec.
- 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.

Results

Hardware

- One MPI process per GPU (4 MI300A per node)
- On CPUs we are bandwidth bound
- On GPUs we launch a lot of kernels
- On GPUs we have a number of complexities
  - **Virtual functions** on device: boundary conditions
  - **Atomics**: Particle Push (BCs, integrated charge and diagnostics)
  - **Large/complex kernels**: Particle Push and Collisions
  - **Warp divergence**

	El Capitan
Nodes	11,136
MI300As per node	4
MI300As total	44,544
Node Peak (DP TFLOP/s)	250.8
System Peak (DP PFLOP/s)	2,792.9
System Peak (SP PFLOP/s)	3,688.2
System Peak (HP PFLOP/s)	17,639.4
Node Memory (GiB) (All HBM3)	512
System Memory (TiB)	5,568
Compute cabinets	87
Peak Power (MW)	34.8
Total Rabbit Modules	696
November 2024 Top500 position	1

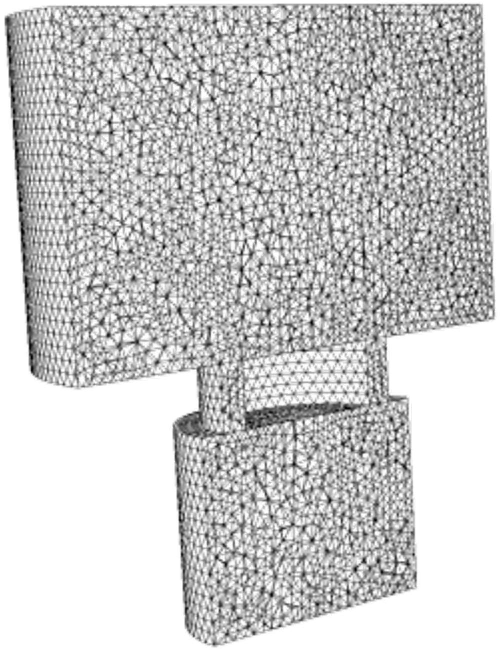
Machine	Node	STREAM Bandwidth (TB/s)	Relative Bandwidth
Amber	CTS-2 SPR x2	0.41	1.0
Sierra	ATS-2 V100 x4	3.4	8.3
El Capitan	ATS-4 MI300A x4	14	34.1

Machine	Peak FMA64 FLOPs (TF)	Relative FMA64
SPR	4.12	1.0
V100	7	1.7
MI300A	61.3	14.9

https://hpc.llnl.gov/documentation/user-guides/using-el-capitan-systems/hardware-overview

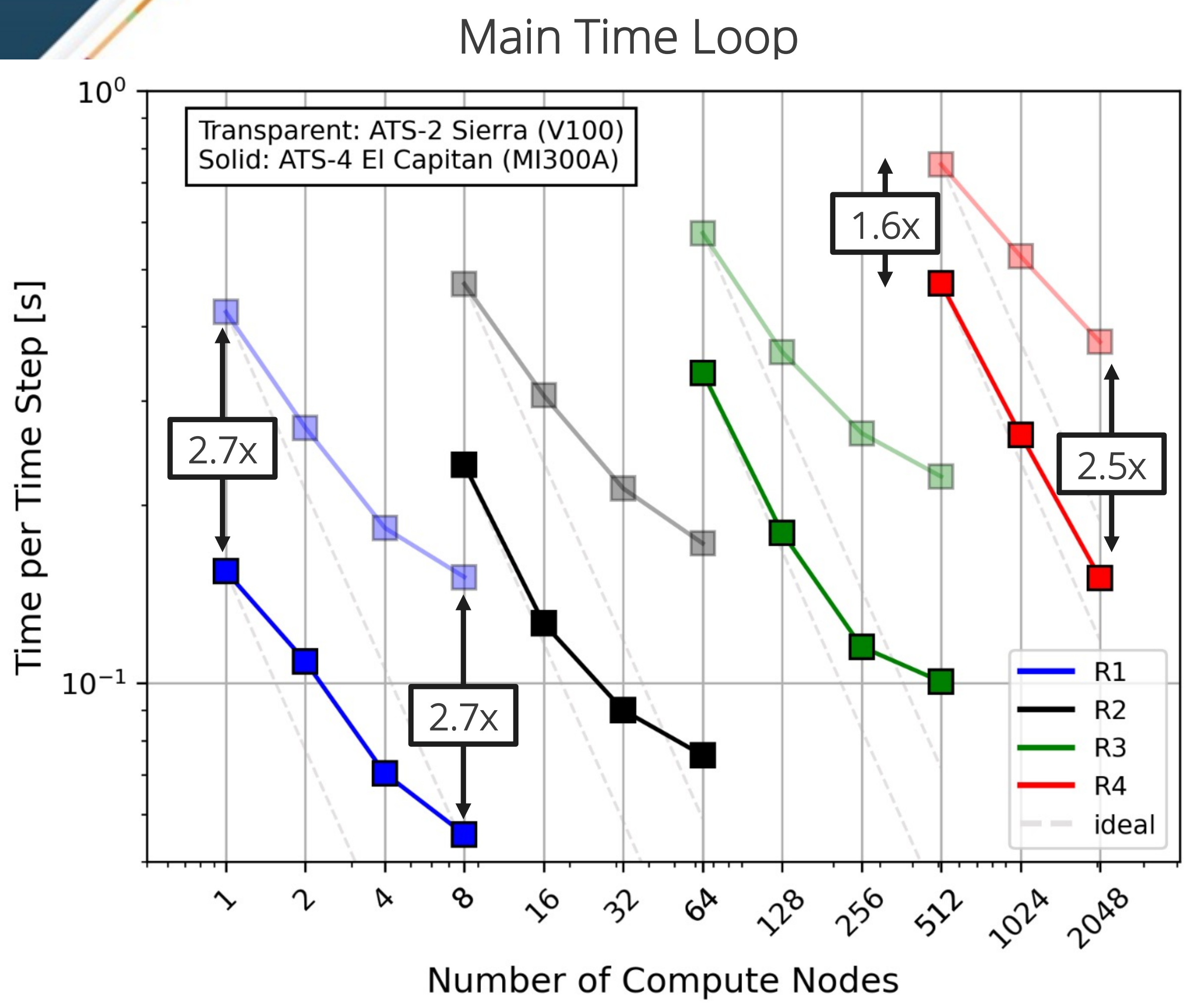
Comparison on Simple Cavity (ATS-2 Sierra vs ATS-4 El Capitan)

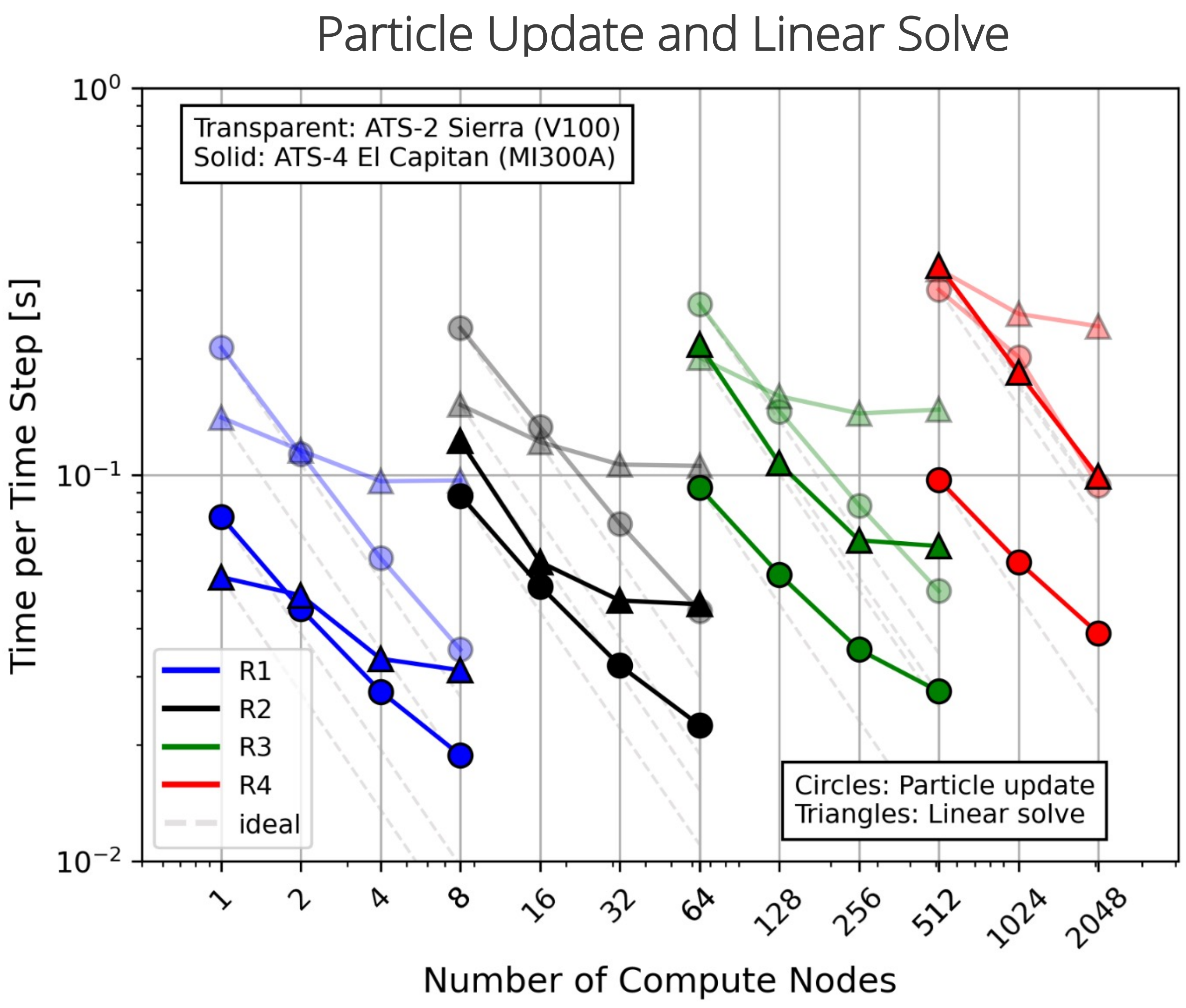
- "Simple Cavity" is for performance testing
  - Pre-loaded particles: **no startup effects**
  - Large amount of work in **short runtimes**: 100 time steps (5 min walltime)
- Last used in FY2020 L1 Milestone for Sierra
- Maybe not apples to apples – 5 years of Empire and Trilinos development
  - More complex kernels
  - More complex physics
  - More memory in device functors



Name	Num Elements	Max Particles	Scaling Node Counts	elems per MPI process (low)	elems per MPI process (high)
r1	2,678,390	128,562,720	1,2,4,8	669,598	83,700
r2	20,715,338	1,035,766,900	8,16,32,64	647,354	80,919
r3	165,722,704	8,286,135,200	64,128,256,512	647,354	80,919
r4	1,325,781,632	66,289,081,600	512,1024,2048	647,354	161,839

AST-2 Sierra vs ATS-4 El Capitan





- Node Comparison: 4x NVIDIA V100 vs 4x AMD MI300A
- Strong scaling has improved, particularly in the linear solver
- There are limitations as to what can be interpreted from this data

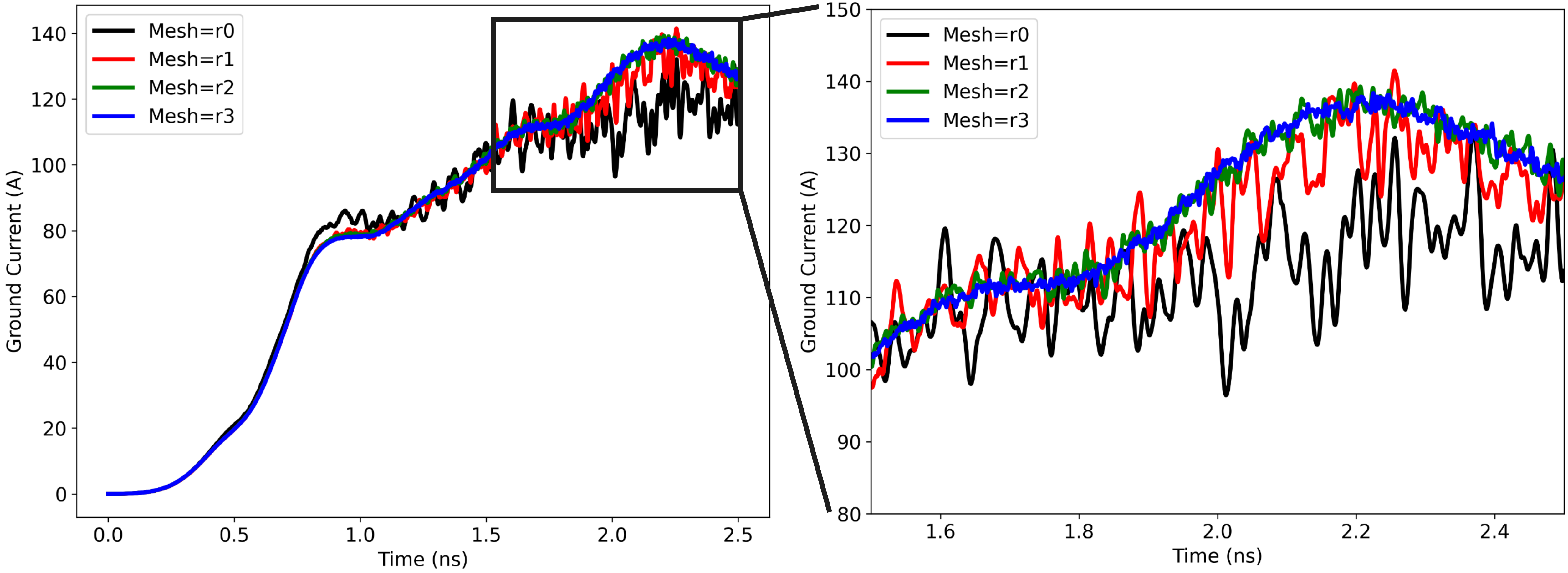
Machine	Node	STREAM Bandwidth (TB/s)	Relative Bandwidth
Sierra	ATS-2 V100 x4	3.4	1.0
El Capitan	ATS-4 MI300A x4	14	4.1

More Complex Radiation-Driven Cavity

Name	Number of Time steps (to 5ns)	Num Elements	Scaling Node Counts	elems per MPI process (low)	elems per MPI process (high)
r0	5k	611,675	testing only		
r1	10k	4,893,400	8,16,32,64	152,919	19,115
r2	20k	39,147,200	64,128,256,512	152,919	19,115
r3	40k	313,177,600	512,1024,2048,4096	152,919	19,115
r4	80k	2,505,420,800	4096, 8192	152,919	76,459

- Starts empty and particles enter through boundary
- Fewer elements per node than the “Simple Cavity” ~1/4
- Maximum Number of Particles (R3 @ 1 Torr): 537 billion

Mesh Convergence at P=0.1 Torr



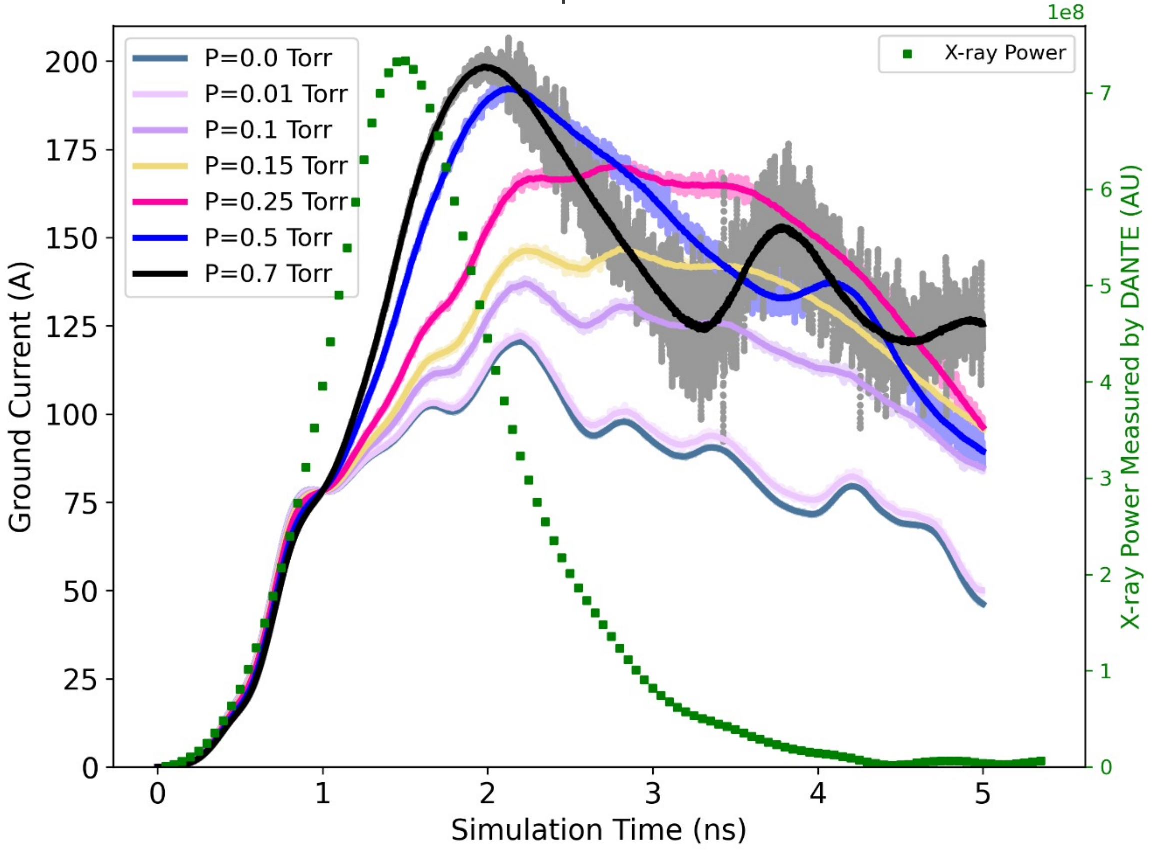
- Determine mesh resolution requirements at various pressures
- CTS-2 Amber ran out of memory on R3 at 512 nodes (Could run with 1024 if we took whole machine for a week)

Ensemble Calculations

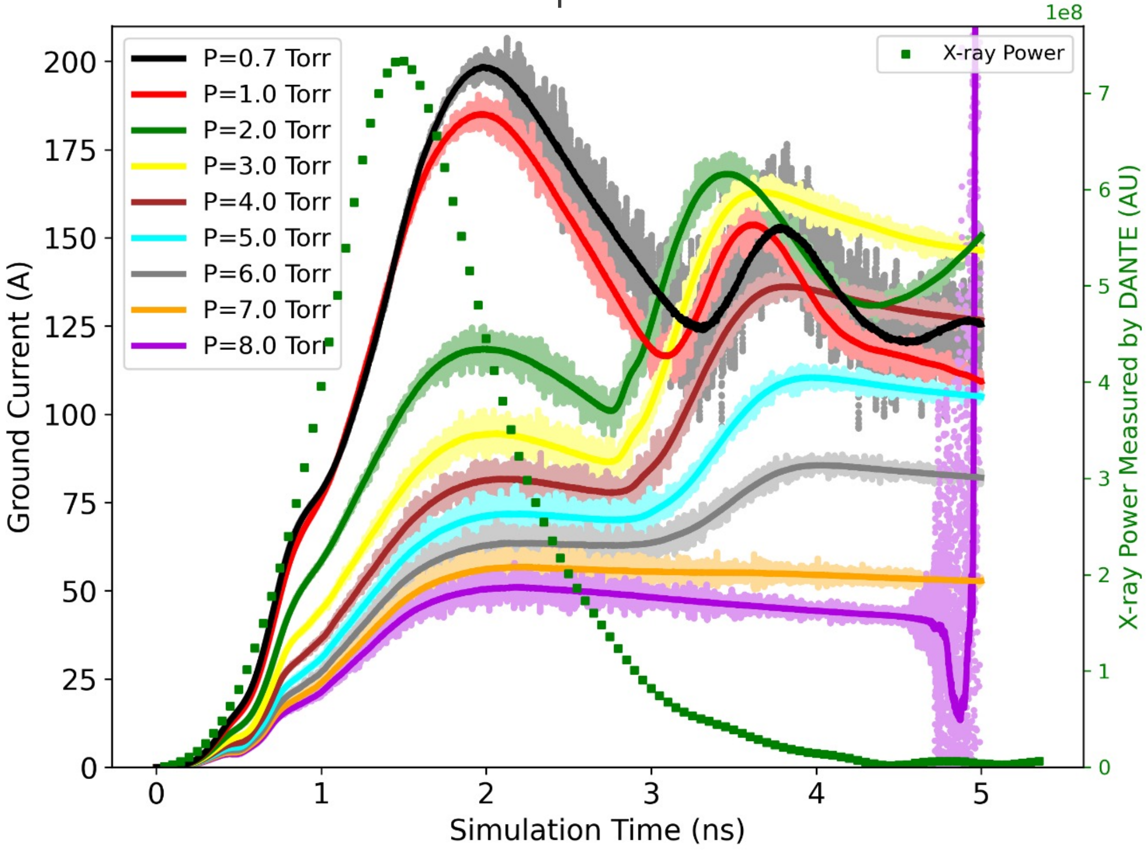
- Perform a pressure sweep (18 points) to identify max current in system
- **Requires R3 (512 nodes minimum due to memory)**
- Run for 5ns to get peak
- Took **~50 hours @512 nodes for 5ns**
  - 24 hr queue limit (requires 2 restarts)
- Analysts want to run a subset out to 50 to 100 ns!
  - After 10ns, particle counts rapidly drop, so the simulation speeds up
- Higher pressures will require more refined meshes

Radiation-Driven Cavity Ensemble Pressure Sweep on R3

Pressure Sweep 0.0 to 0.7 Torr



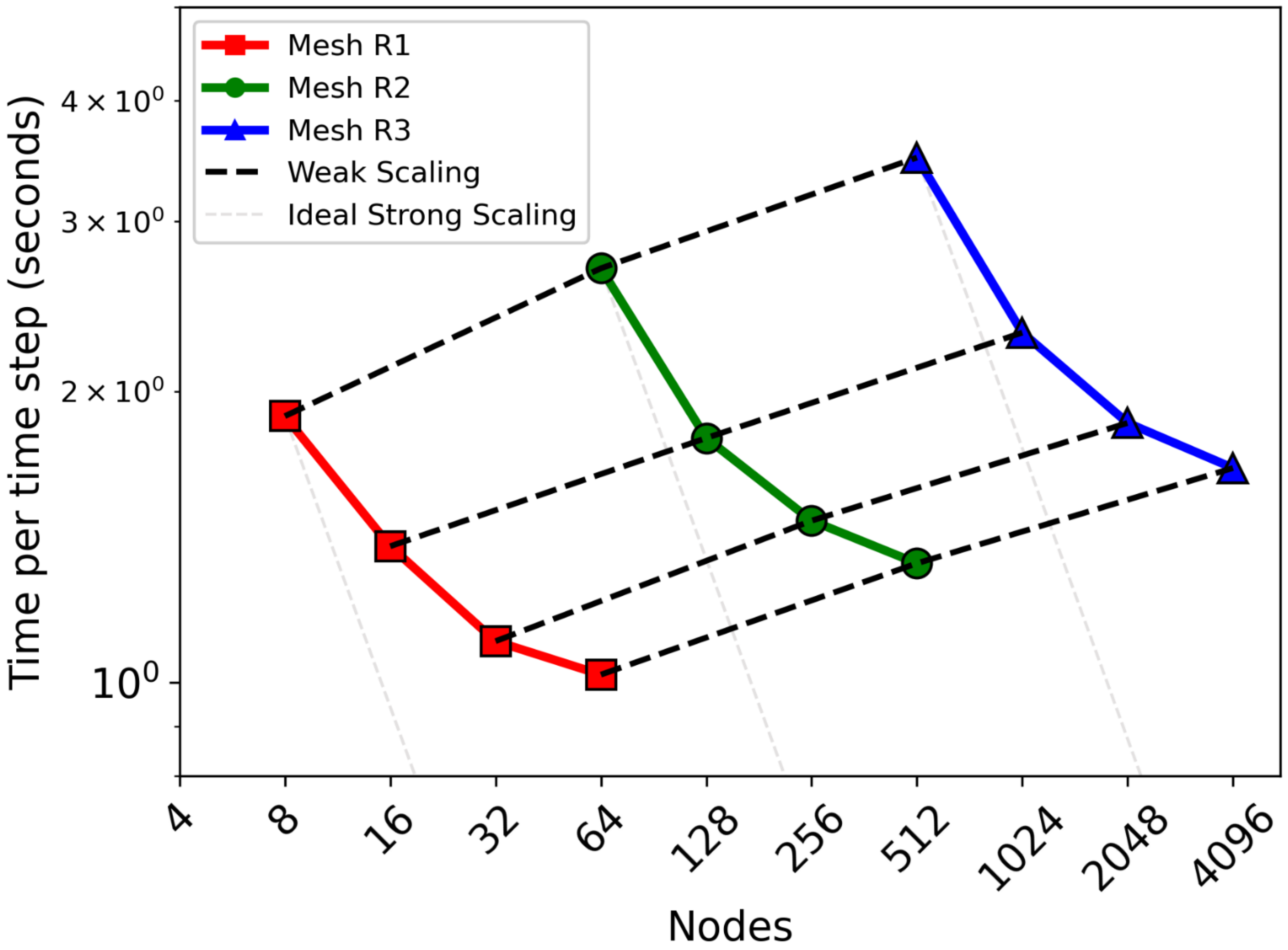
Pressure Sweep 0.7 to 8 Torr



- Successfully identified maximum at 0.7 Torr

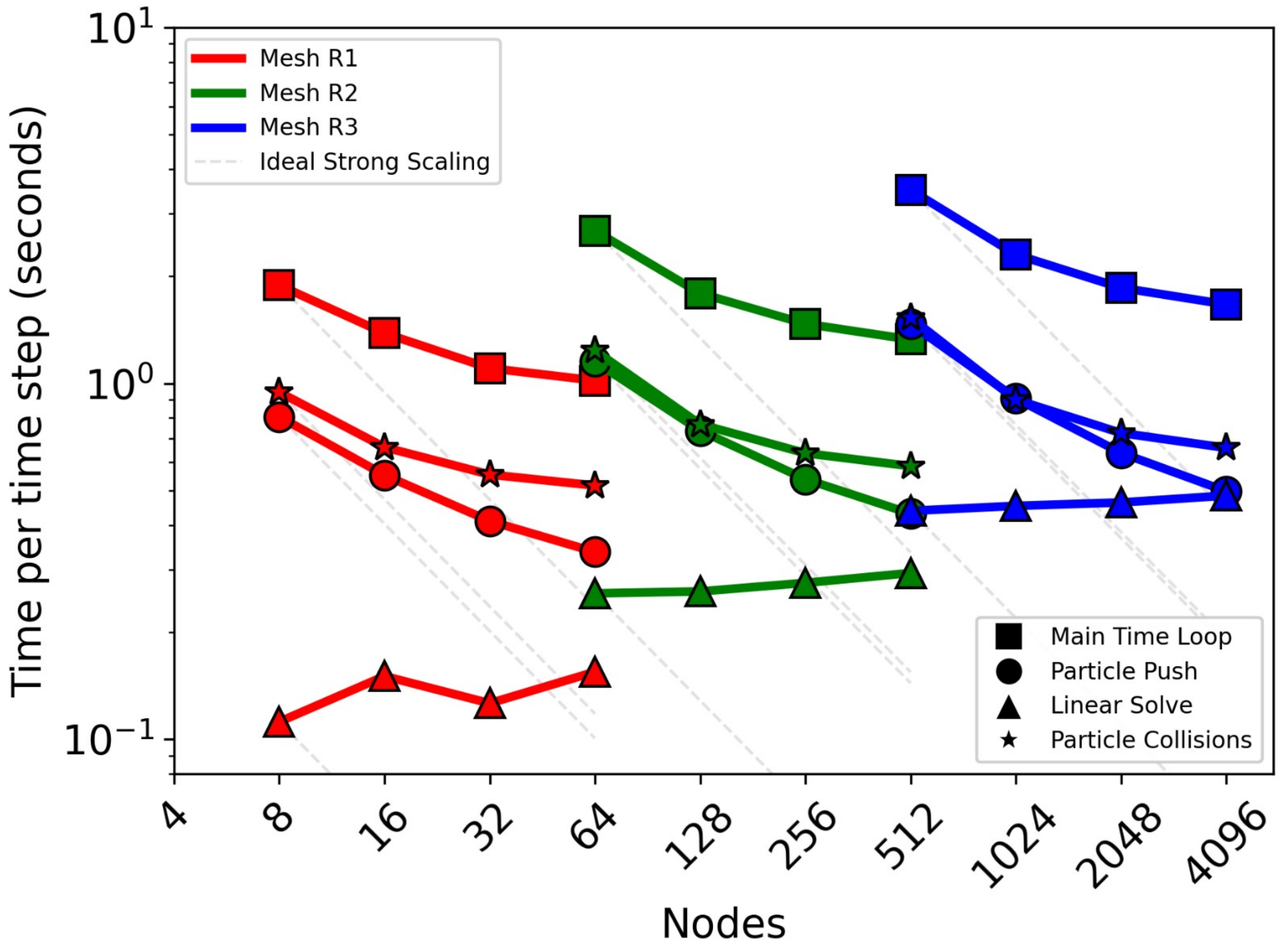
Scaling Study with Radiation-Driven Cavity (P=1 Torr)

- Demonstrate good strong and weak scaling
- In practice, we operate near the strong scaling limit
- Tightly constrained by memory/work per core



Strong Scaling Study: radiation-driven cavity (P=1 Torr)

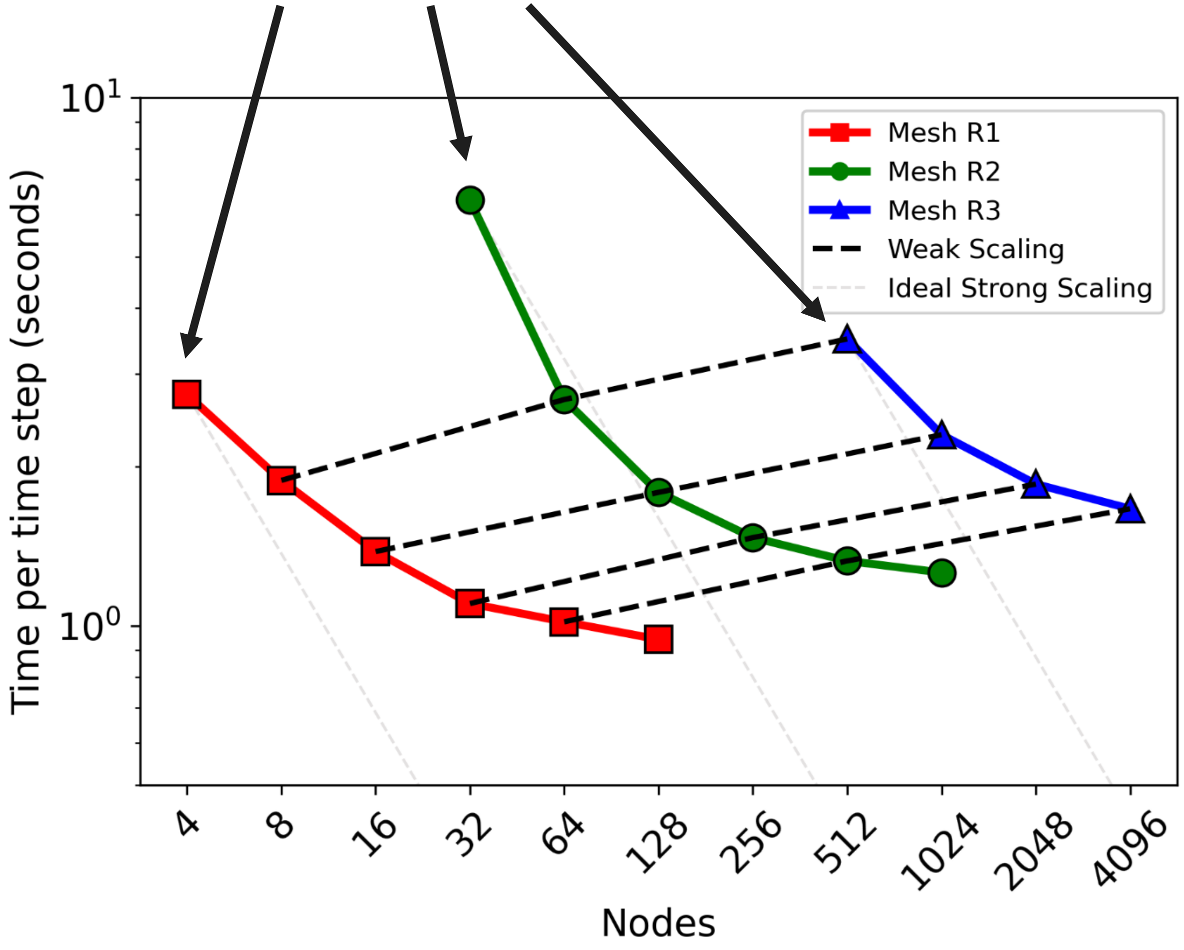
- Subtimers for the big three:
  - Particle Push
  - Linear Solve
  - Particle Collisions
- Accurate subtimer data requires extra MPI barriers
- Particle Update uses multiple kernel launches per time step due to MPI communication
- Linear solve is latency bound
- Collisions is surprising, being local work



Memory Requirements and Strong Scaling

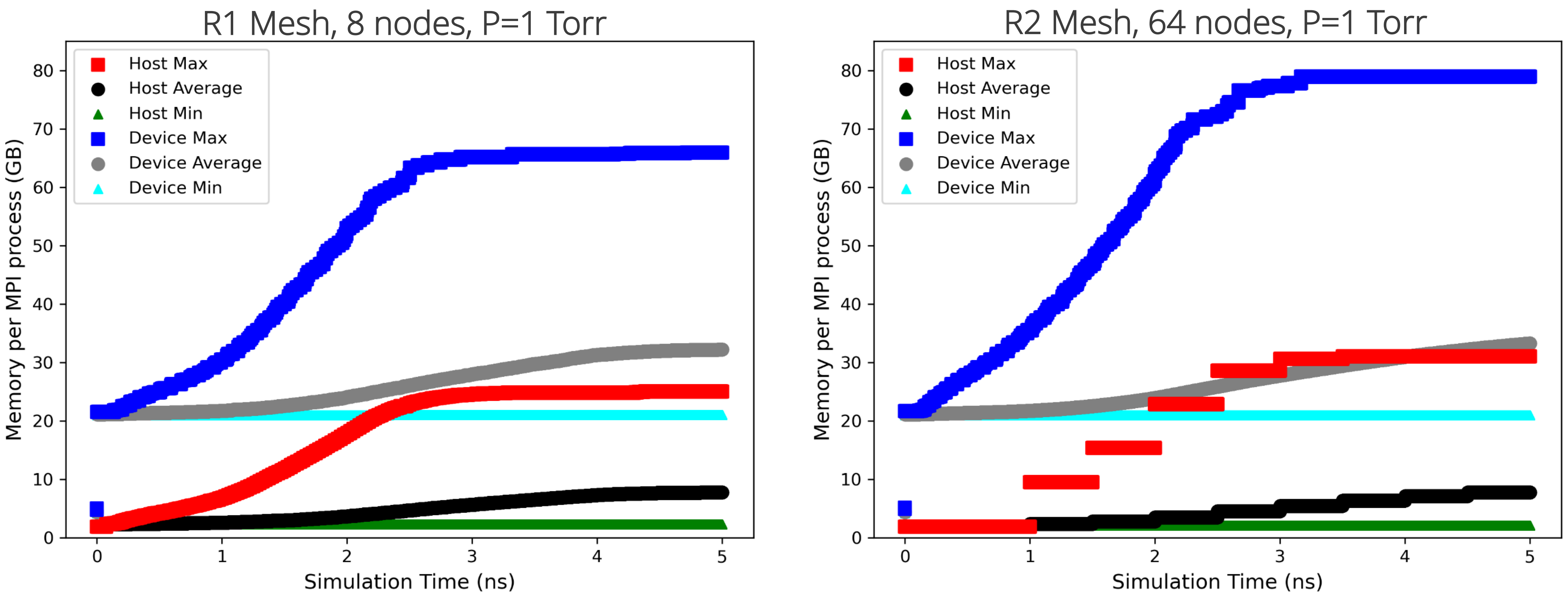
- Run with as few nodes as possible before running out of memory
- This problem requires R3: at least 512 nodes on El Capitan
- R4 requires 4096 nodes (maybe more?)
  - Not feasible on current CTS systems

Smallest node count possible due to memory requirements



Runtime Memory Usage

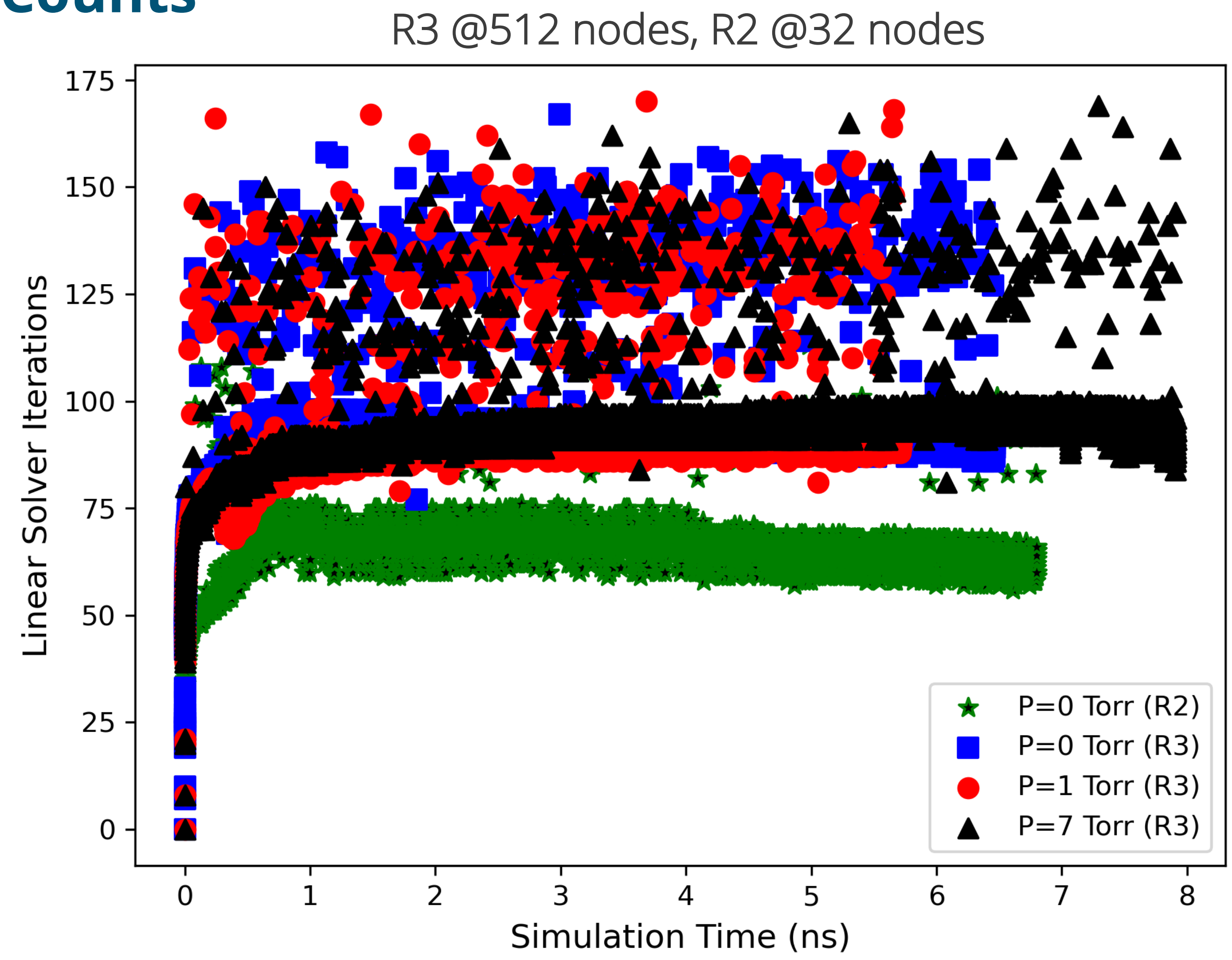
- Unified memory use depends on allocation mechanism: getrusage(), hipMemGetInfo()
- Particle numbers are limited on a per cell basis: max 1200 per cell
- 512 GB per node: 4x MI300a @ 128 GB each



Memory requirements are very high
Constrains scaling studies for hardware comparisons

Linear Solver Iteration Counts

- Preconditioned Algebraic multigrid:
  - Conjugate Gradient Iterative solver
  - RefMaxwell Preconditioner
  - Chebyshev smoother
- Iteration counts remain steady
- Pressure has a weak effect

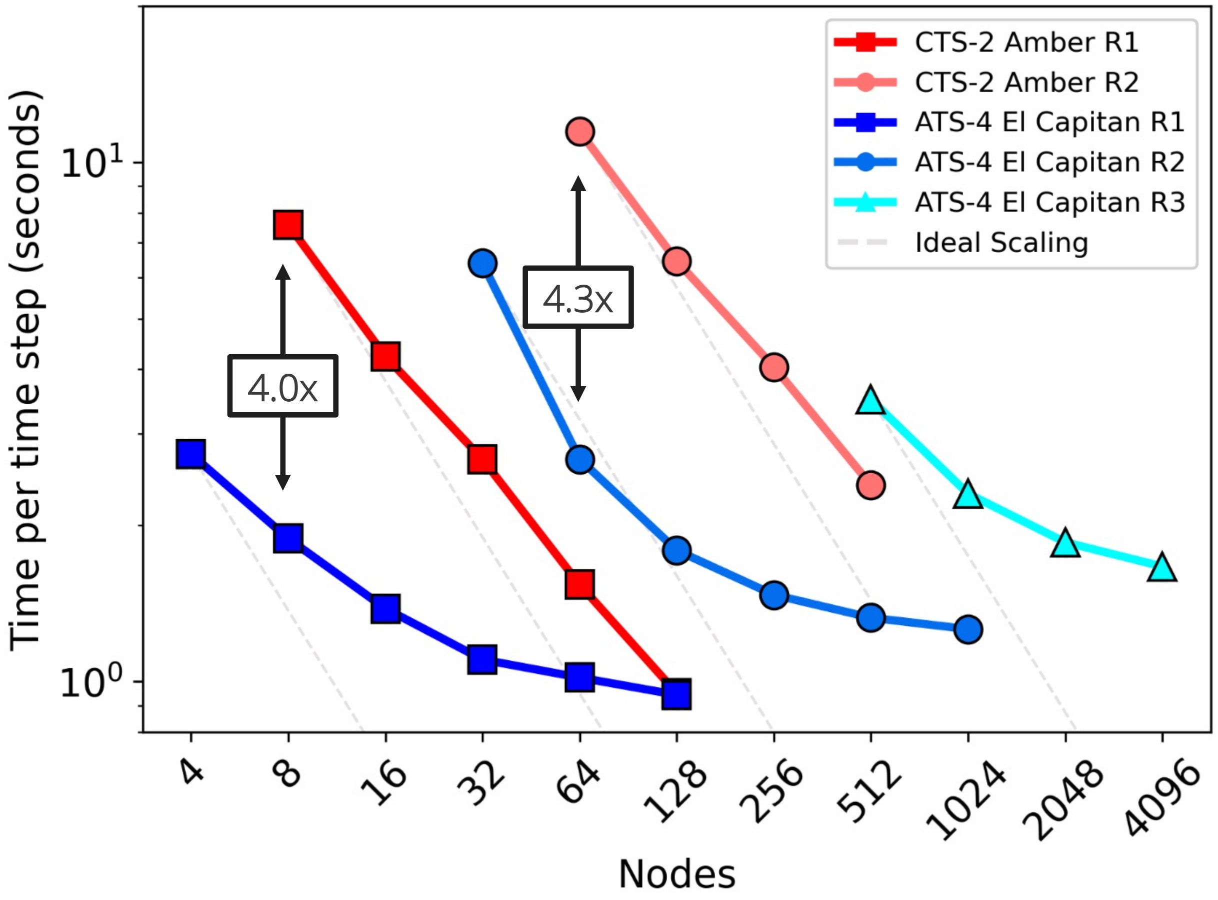


EM Linear solve is working well, even though latency bound

Comparison to CTS-2 Amber (P=1 Torr)

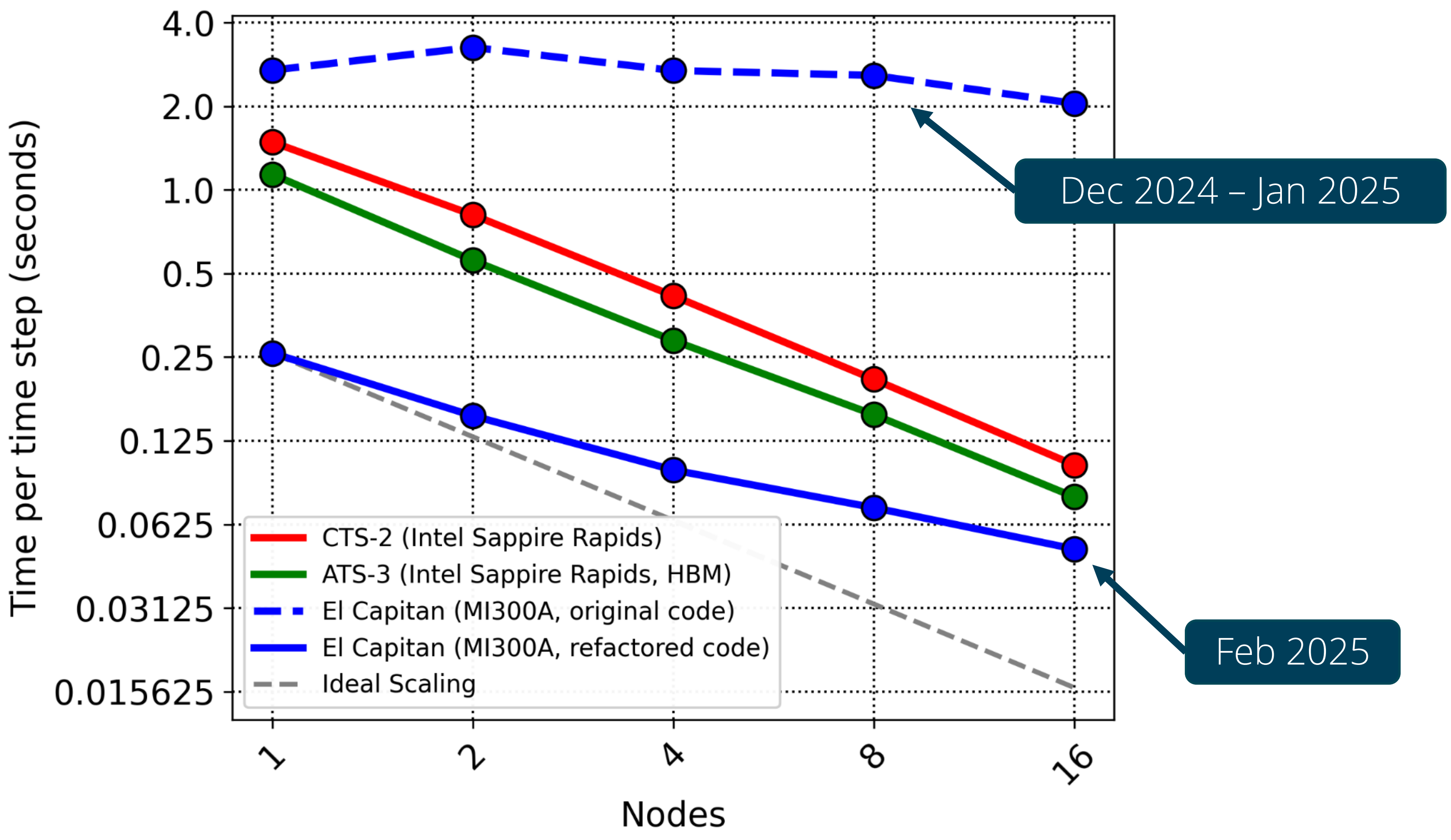
- Left end of each series is smallest node count that fits in memory
- Empire algorithms involve **many** more kernel launches than most PDE solvers
- More GPU performance improvements coming…
  - Collision kernels
  - Eliminate kernel launch overhead (combine kernels, remove fences, use graphs)
  - Evaluate effects of virtual functions on device and atomics
  - Overlap computations
  - Selective use of HostPinnedMemory

Machine	Node	STREAM Bandwidth (TB/s)	Relative Bandwidth
Amber	CTS-2 SPR x2	0.41	1.0
Sierra	ATS-2 V100 x4	3.4	8.3
El Capitan	ATS-4 MI300A x4	14	34.1



Challenge: Atomics are significantly worse on MI300A due to unified memory – had to change algorithms

- Unified memory enforces cache coherence across both GPU and CPU
- Did not observe this on MI250X (no unified memory)



Forced algorithmic change for a single architecture

Summary

- El Capitan allowed us to perform the largest plasma simulations to date of these problems
- Before this, there was no demonstration that we could do sampling (parameter scan) needed in a reasonable amount of time
- Good scaling of the algorithms (both strong and weak)
- Can’t do this analysis on current CTS systems