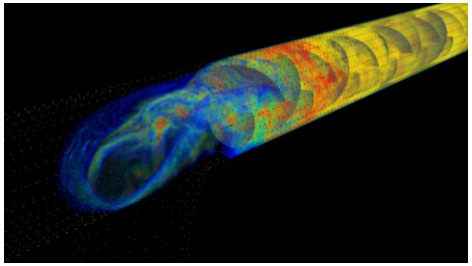
Novel Use of Multigrid/Level/AMR type Methods in Uintah Approaches to Exascale



Open source at www.uintah.utah.edu

Alan Humphrey, Todd Harman, Derek Harris, John Schmidt , Jeremy Thornock, Brad Peterson, John Holmen, Damodar Sahasbarude, **Martin Berzins**

- Introduction,
- Cleaner coal boilers
- Scalable Radiation using AMR
- Towards Performance Portability

Thanks DOE ASC, NETL PSAAP 1997-2019, INCITE Argonne, Pervious Uintah developers and Dave Pershing Phil Smith, PSAAP predictive physics group

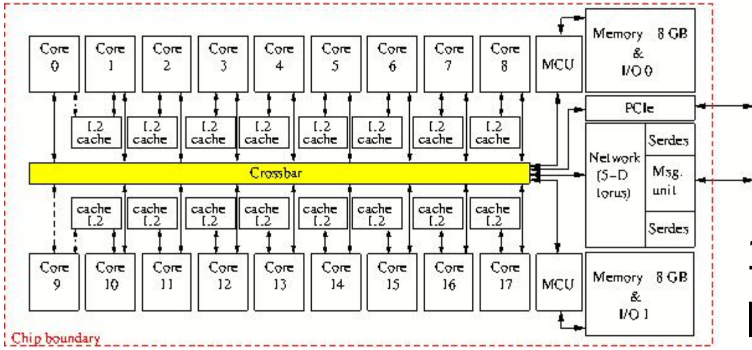
Coal Use in Electric Power Generation

- Still widely used and unlikely to vanish
- Use increasing in Japan
- Important as part of the portfolio
- Works when it is raining and there is no wind so is needed at least as a backup

What if we could reduce the coal burned in US by 50% through better coal boiler designs and design using Exascale computing?

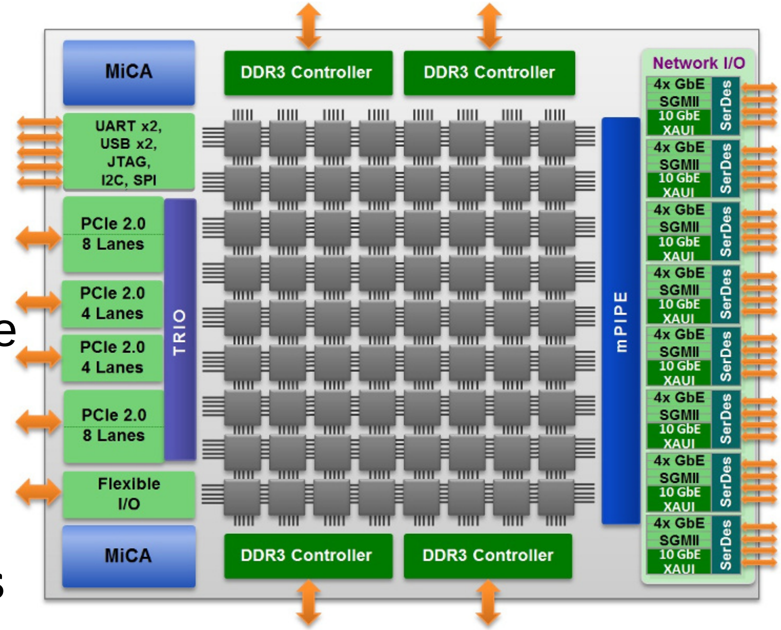
Present nodes used here and Future Nodes Summit Aurora

IBM BGQ 16 core Mira



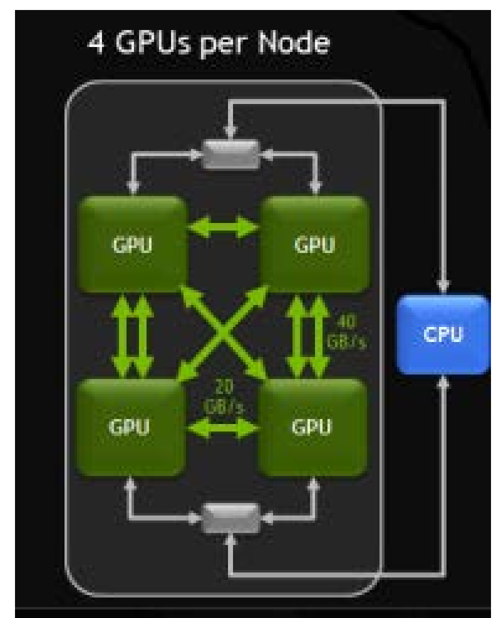
10x performance

Intel Xeon Phi KNL



TITAN NODE

FUTURE GPU NODEs
Pascal Volta Summit
nodes have 2 Power 9
with 3 GPUs



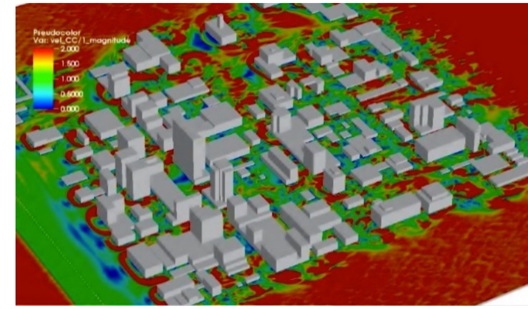
- **Five very different swim lanes**
  (i) Evolving standard Intel x86 (skylake)
  (ii) Intel Xeon PHI KN*
  (iii) NVIDIA GPUs
  (iv) Novel designs e.g. Epiphany 1024 cores,
  Sunway light 160 cores
  (v) Fujitsu Post K Arm chips 7nm process

Uintah Applications and Computer Portability

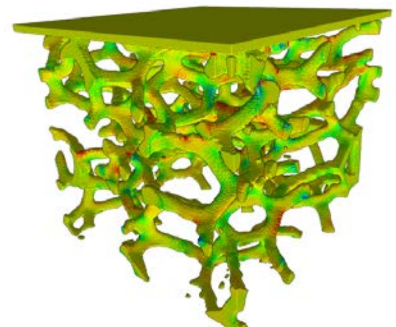
Wide range of cpu + infiniband + gpu + Xeon Phi machines

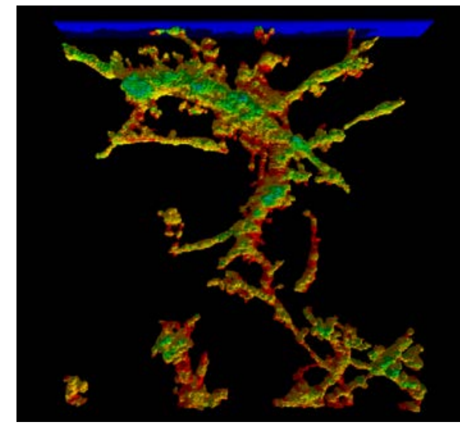
Vulcan and Mira IBM BG/Q
Cray Blue Waters, Titan
Variety of Cray XC 30s
IBM Dataplex DOD

TACC Stampede CPU+Xeon Phi

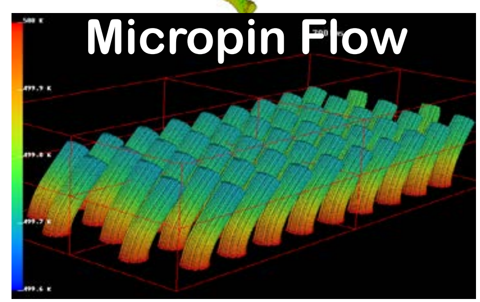


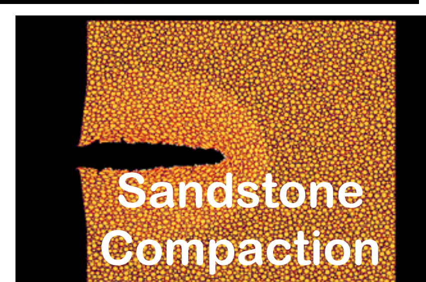
Foam Compaction

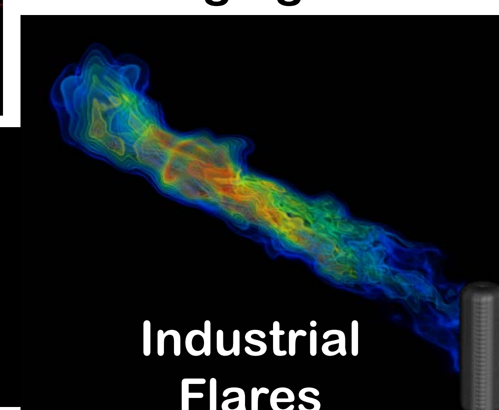




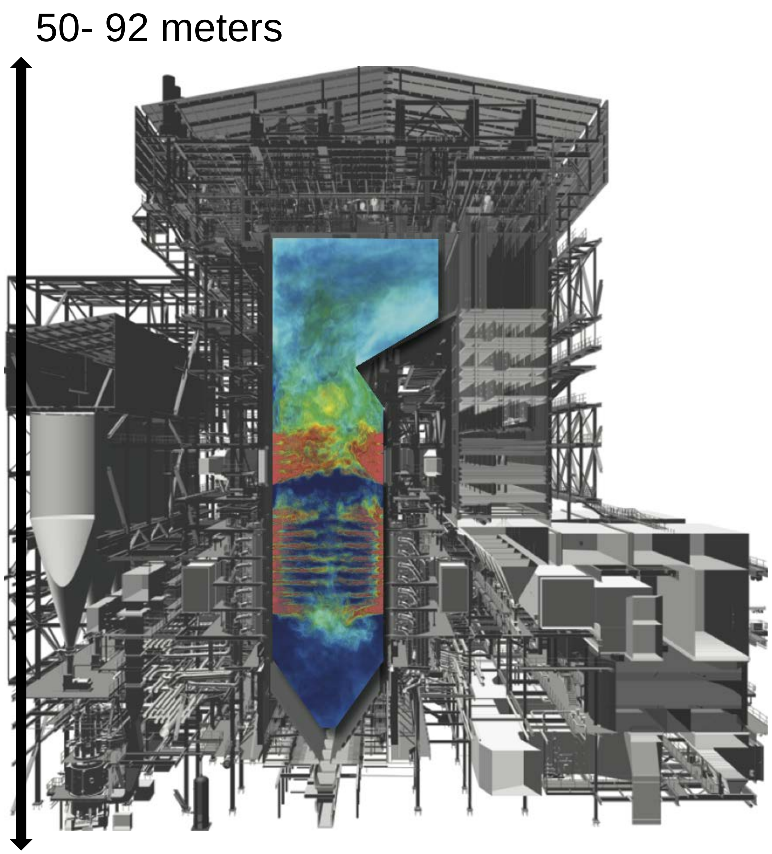
Angiogenesis

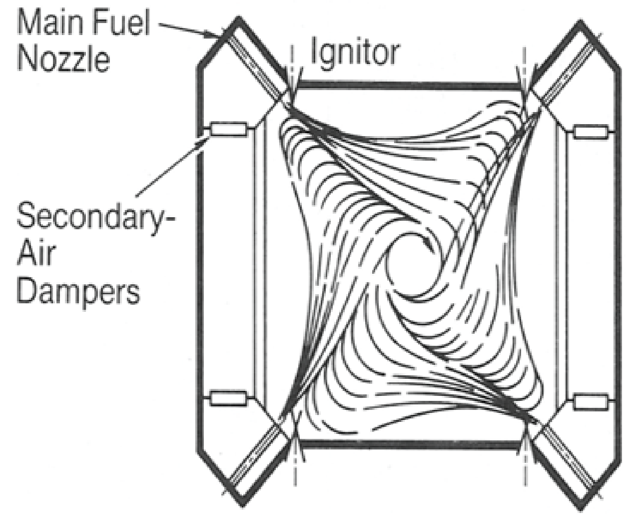






Exascale Target Problem DOE NNSA PSAAP II Center –

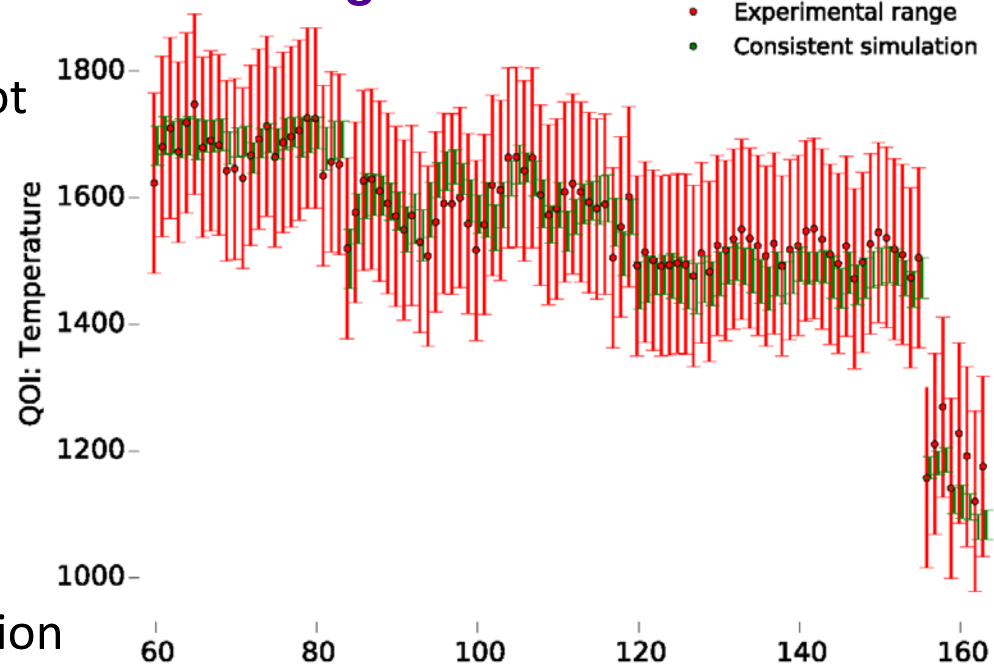




- GE 1000MWe “Twin Fireball” boiler
- Supply power for 1M people
- 1mm grid resolution = 9×10^{12} cells
- 1000 times larger than largest problems solved today
- Thermal radiation key for heat transfer

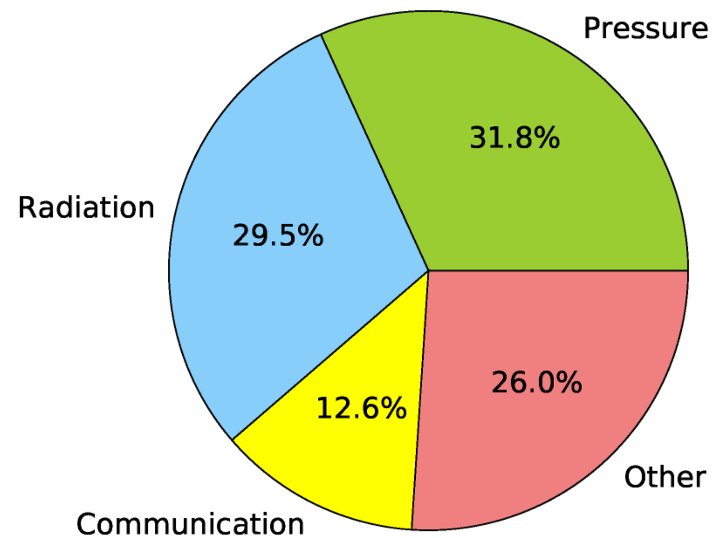
Existing Simulations of Clean coal Boilers using ARCHES in Uintah

(i) Lagrangian/RANS approaches do not address particle effects
(ii) LES has potential to predict oxy--coal flames and to be an important design tool
(iii) LES is “like DNS” for coal



- Structured, high order finite-volume
- Mass, momentum, energy conservation
- LES closure
- Tabulated chemistry
- PDF mixing models
- DQMOM (many small linear solves)
- Uncertainty quantification

- **Low Mach number approx. (pressure Poisson solve up to 10^{12} variables**
- **Radiation via Discrete Ordinates – massive solves or Ray tracing.**

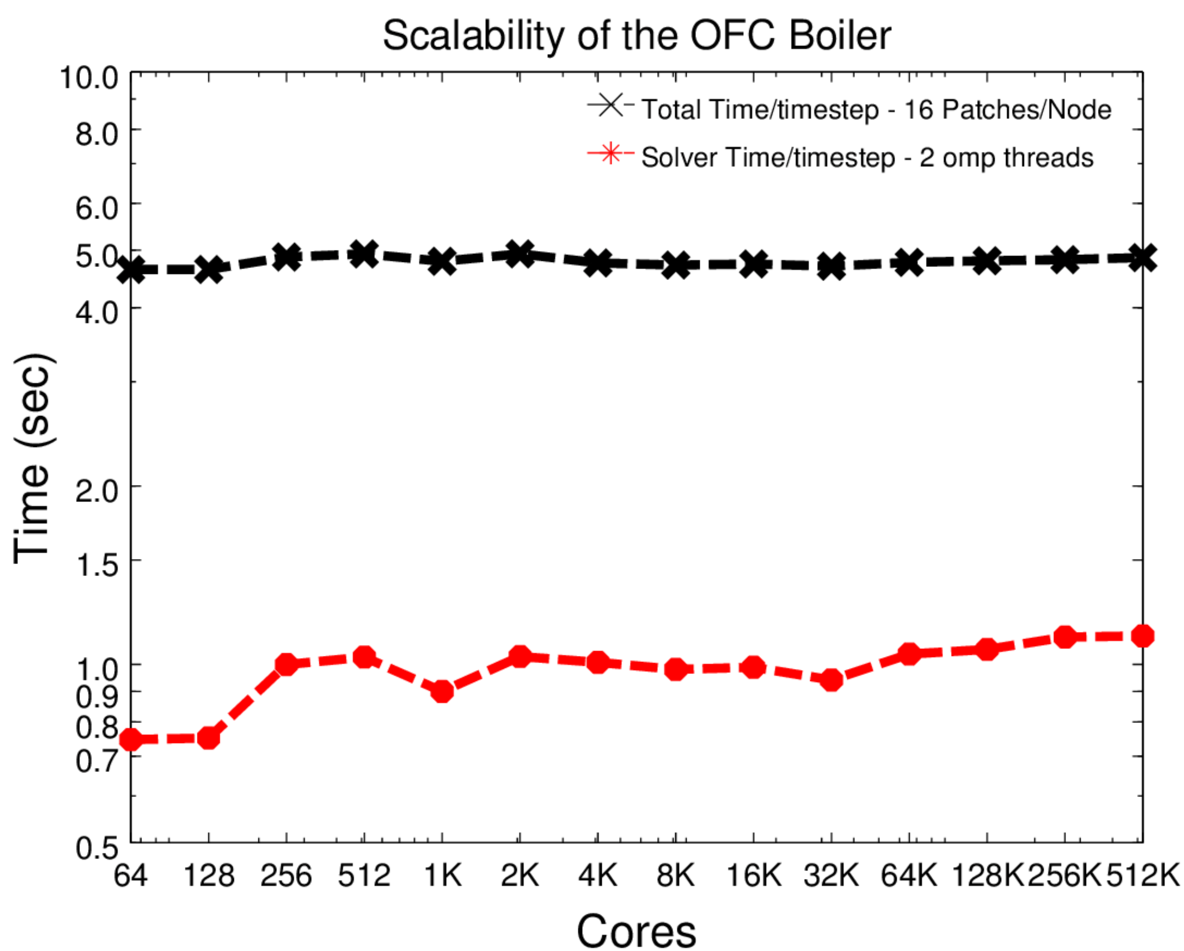


Linear Solves arise from Low Mach Number Approximation in Navier–Stokes Equations

pressure Poisson equation $\nabla^2 p = R, \quad \text{where } R = \nabla . F + \frac{\partial^2 p}{\partial t^2}$

Use hypre Solver distributed by LLNL Preconditioned Conjugate Gradients on regular mesh patches used Multi-grid pre-conditioner used. 20^3 patch with one MPI process per core no AMR or radiation.

[John Schmidt]



Uintah software Architecture

Applications code
Programing model

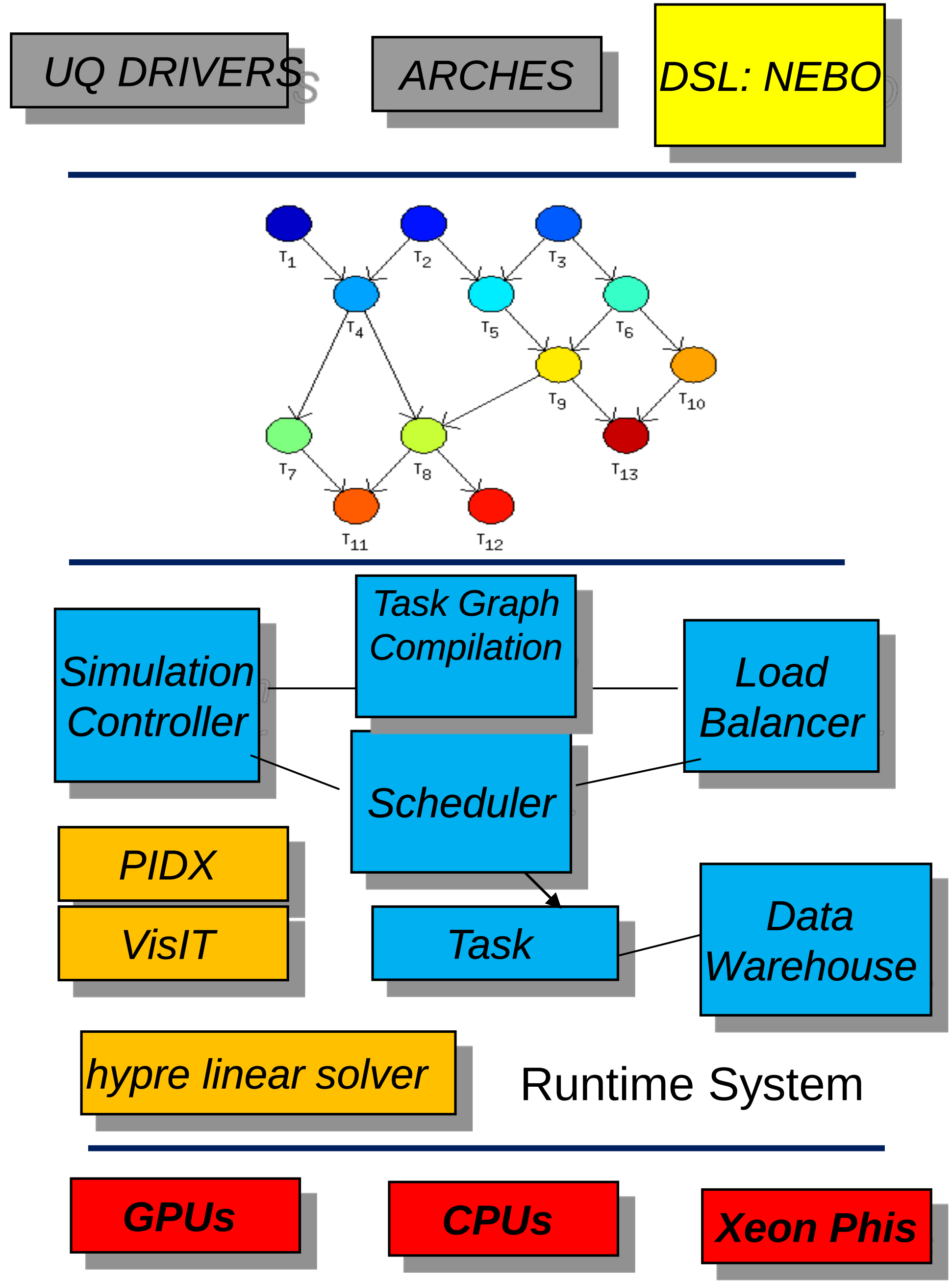
Components NOT architecture specific and do not change

Automatically generated abstract C++ task graph form

Adaptive execution of tasks

asynchronous out-of-order execution, work stealing, overlap communication & computation.

Open source software
Worldwide distribution
Broad user base



Radiation Overview

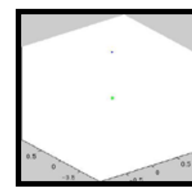
<u>Solving **energy** and **radiative heat transfer** equations simultaneously</u>

$$\frac{\partial T}{\partial t} = \text{Diffusion} - \text{Convection} + \text{Source/Sinks}$$

$$\nabla \cdot q$$

- *Energy equation solved by ARCHES (finite volume)*
- *Temperature field, **T** used to compute **net radiative source term,** requires integration of incoming intensity about a sphere*

$$\nabla . q = \kappa \left(4\pi I - \int_{4\pi} I d\Omega\right) \rightarrow \sum_{rays} \alpha_r I_r$$



- ***Net radiative source term** goes back into ongoing CFD calculation*

- *Monte Carlo, Discrete Ordinates or Spherical harmonics are the options*

Example of Derivation [Source Cailot]

$$\frac{\partial \rho}{\partial t} + \overrightarrow{\nabla} \cdot (\rho \vec{v}) = 0$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \overrightarrow{\nabla} \cdot (\rho \vec{v} \otimes \vec{v}) = -\overrightarrow{\nabla} p + \overrightarrow{\nabla} \cdot \overrightarrow{\overrightarrow{\tau}} + \rho \vec{f}$$

$$\frac{\partial (\rho e)}{\partial t} + \overrightarrow{\nabla} \cdot [\, (\rho e + p) \vec{v}\,] = \overrightarrow{\nabla} \cdot \left(\overrightarrow{\overrightarrow{\tau}} \cdot \vec{v} \right) + \rho \vec{f} \cdot \vec{v} - \underbrace{\overrightarrow{\nabla} \cdot \vec{q}}_{\text{Flux Divergence}} + R$$

$$\overrightarrow{\nabla} \cdot \vec{q} = \overrightarrow{\nabla} \cdot (\underbrace{\vec{q}_c}_{\text{Conductive}} + \underbrace{\vec{q}_r}_{\text{Radiative}})$$

$$\overrightarrow{\nabla} \cdot \vec{q}_c = \overrightarrow{\nabla} \cdot (-\lambda \overrightarrow{\nabla} T)$$

$$\begin{aligned} \overrightarrow{\nabla} \cdot \vec{q}_r &= -\int_0^\infty \kappa_{a\nu} \left(\int_{4\pi} d\Omega\, I_\nu - 4\pi I_{b\nu} \right) d\nu \\ &= -\int_0^\infty \kappa_{a\nu} (G_\nu - 4\pi I_{b\nu})\, d\nu \end{aligned}$$

Discrete Ordinates Approach [Source Brunner]

Discretize Intensity of radiation *I(r, ...)* in *M* distinct directions Ω_n

$$I(\mathbf{r}, \mathbf{\Omega}, \varepsilon, t) = \sum_{n=1}^{M} I_n(\mathbf{r}, \varepsilon, t)\delta(\mathbf{\Omega} - \mathbf{\Omega}_n).$$

Putting this in equations that come from original Boltzman equation

$$\frac{1}{c}\frac{\partial I_n}{\partial t} + \mathbf{\Omega}_n \cdot \nabla I_n = -\sigma_t I_n + \frac{\sigma_s}{4\pi}\sum_{m=1}^{M} w_m I_m + \sigma_a B(T_m) + S_n,$$

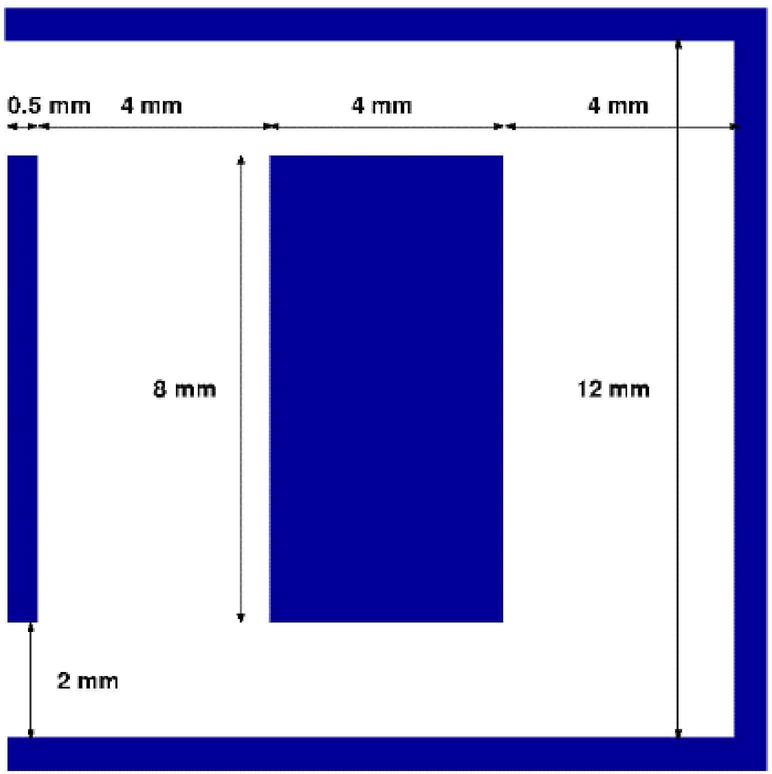
Where σ_* are opacity/absorb coeffs

B is a black body term

	Quadrature Order				
Dimensions	2	4	8	16	N
One	2	4	8	16	N
Two	4	12	40	144	$\frac{1}{2}N^2 + N$
Three	8	24	80	288	$N^2 + 2N$

With more dimensions and quadrature order the number of back solves increases

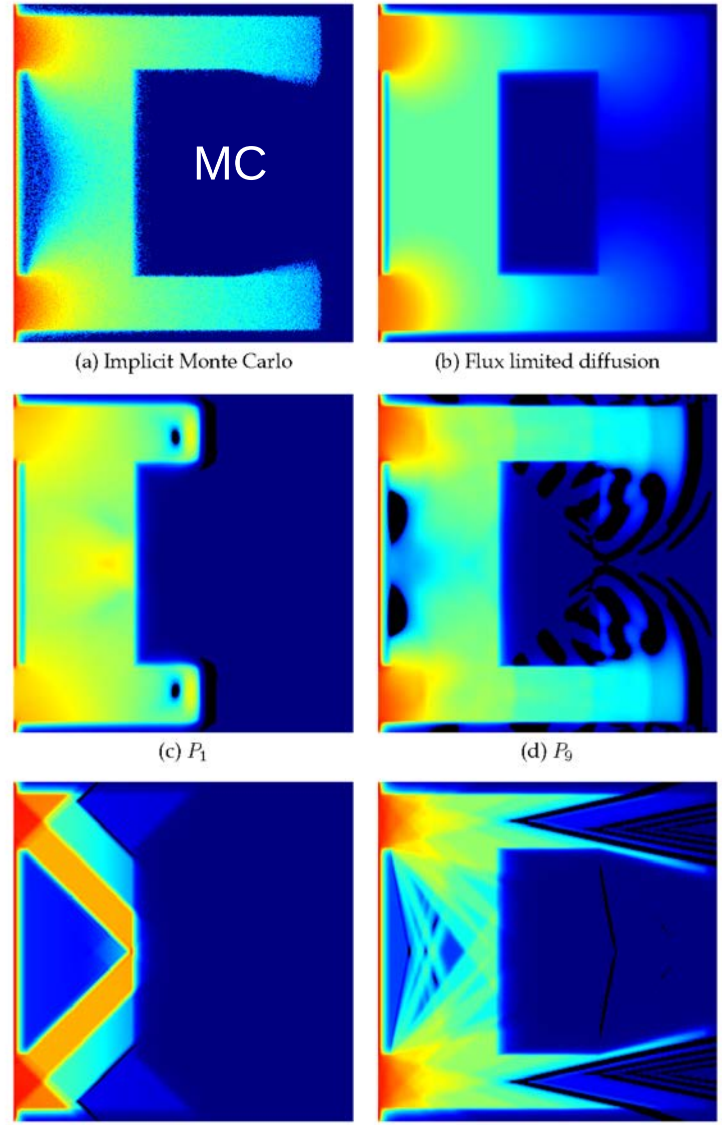
Radiation Comparison
Brunner
[sand2002-1776]
line source Z pinch (or NIF) holraum problem



Monte Carlo Raytracing

Discrete Ordinates

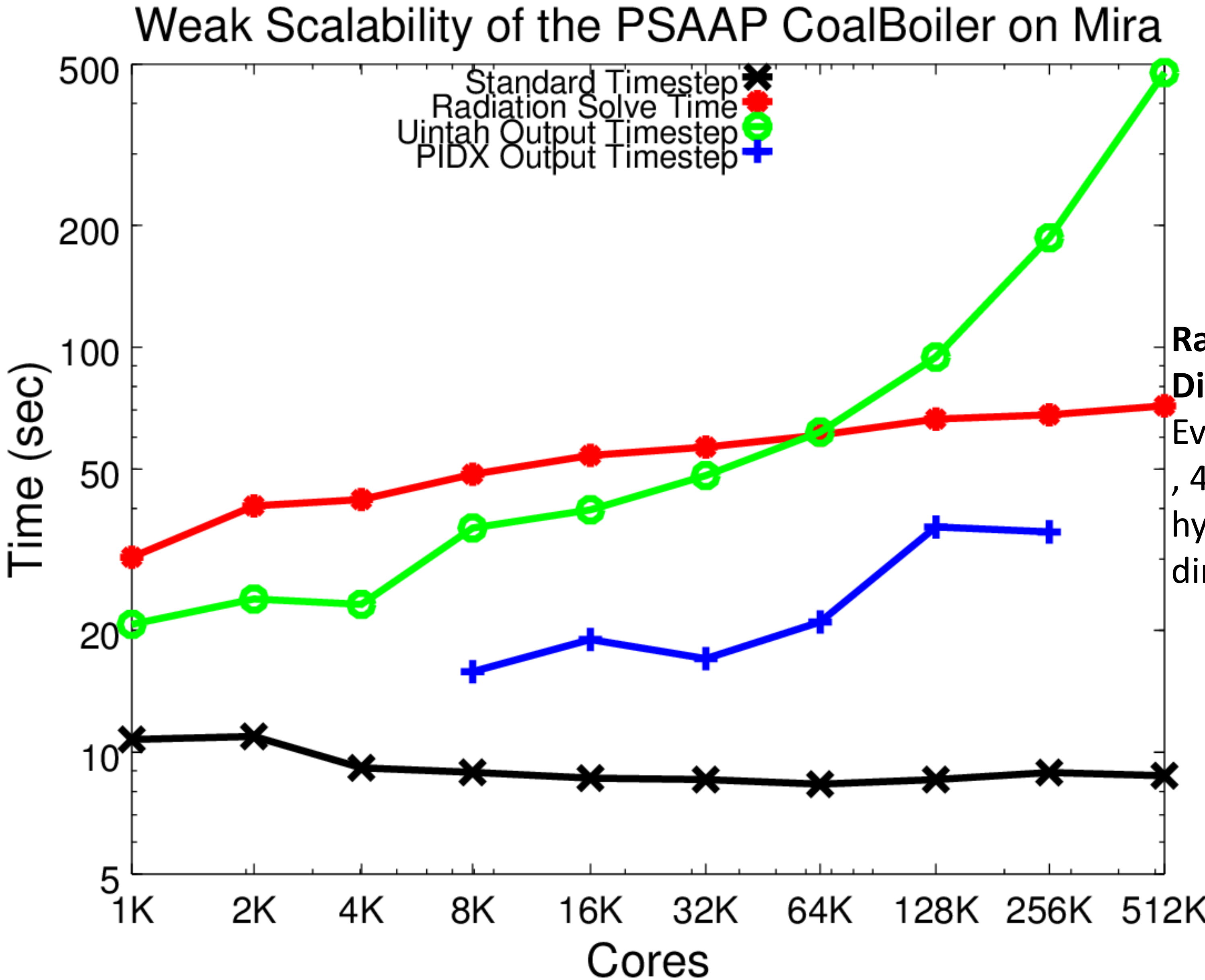
Spherical Harmonics



(a) Implicit Monte Carlo (b) Flux limited diffusion

(c) P_1 (d) P_9

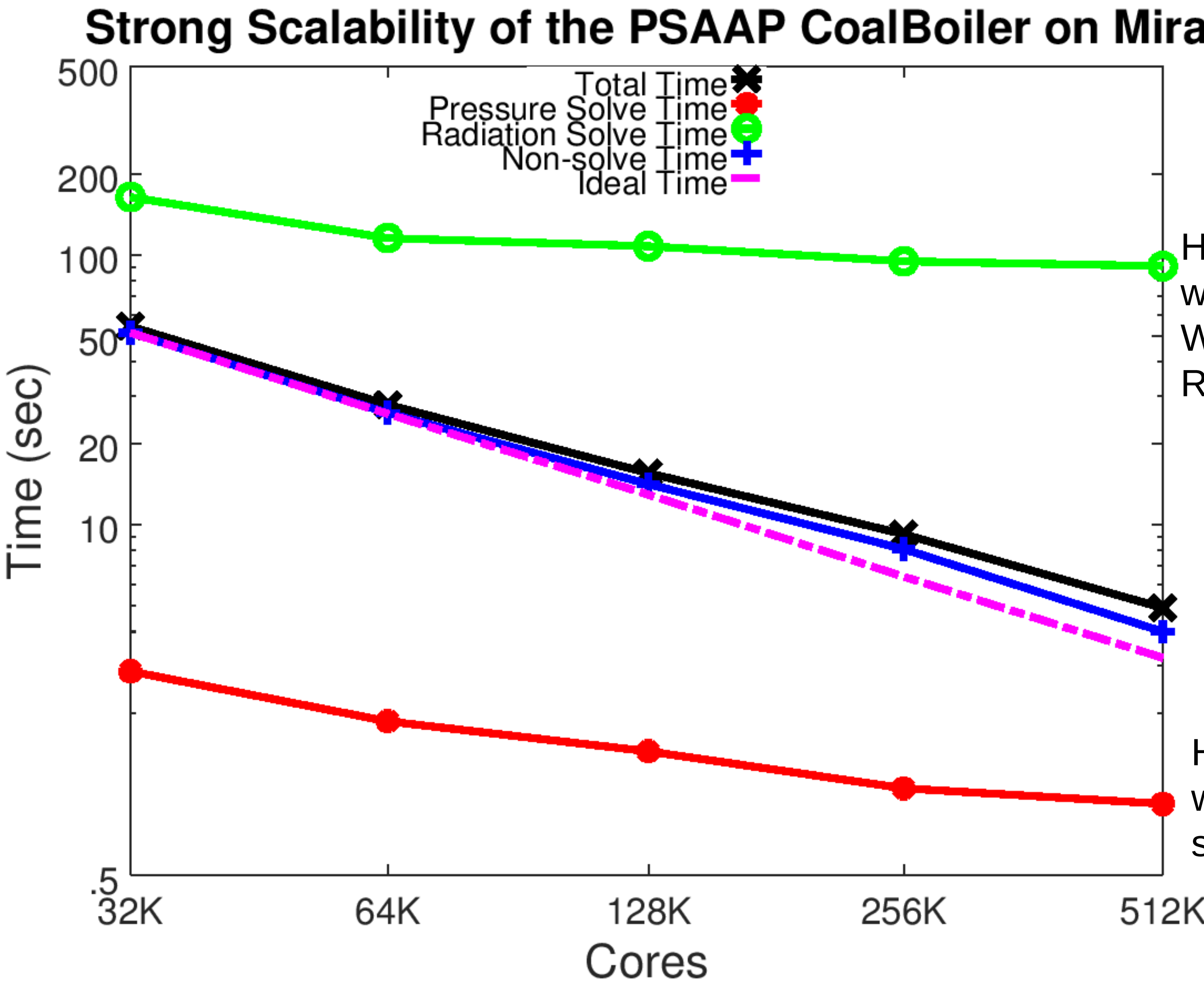
(e) S_2 (f) S_8



Radiation solve
Discrete Ordinates
Every 7 steps S_N 6 , 48 directions hypre for each direction

One 12x12x12 patch per core, 10K variables per core, 31 timesteps. Largest case 5 Bn unknowns. Production runs use 250K cores For I/O PIDX scales better and is being linked to Uintah For radiation use Raytracing . PIDX gives 30x speed up for IO

In response to the scaling CS question



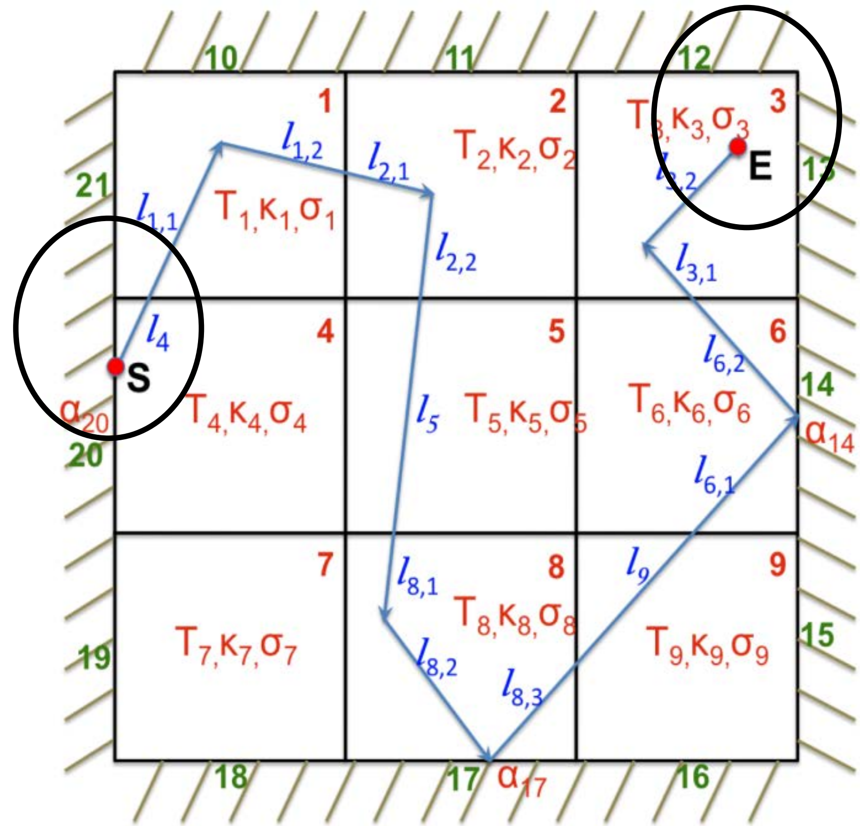
Hypre only weak scales

We need RMCRT

Hypre only weak scales

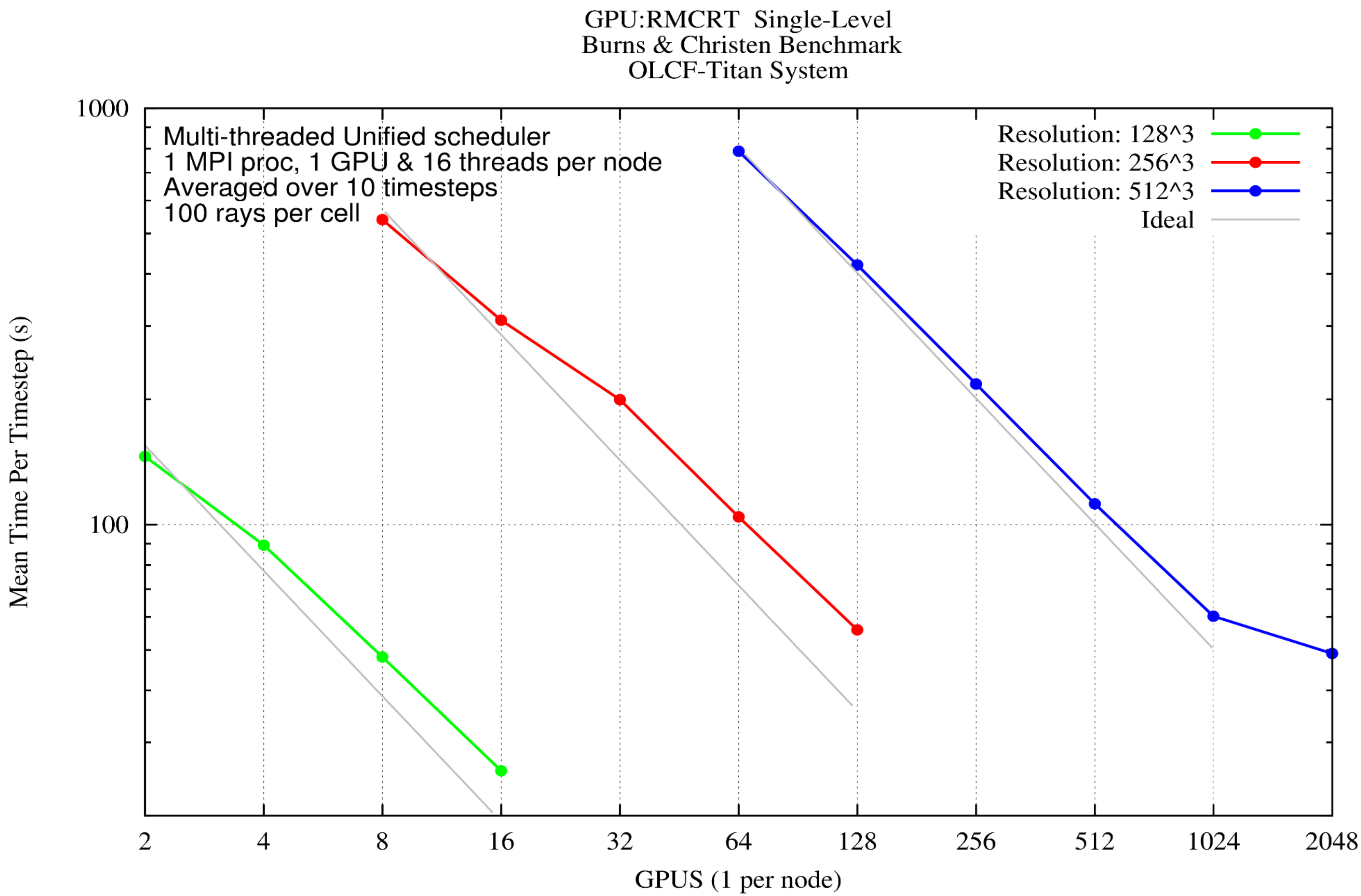
Ray Tracing Radiation Overview

- *Radiation-energy coupling incorporated by radiative source term*
- *Temperature field, **T** used to compute **net radiative source term*** requires integration of incoming intensity about a solid angle with reverse Monte Carlo ray tracing (RMCRT)
- Radiative properties and radiative fluxes calculated on **each node's replicated mesh** and their values transmitted to minimize communication.
- Need to global all to all of heat fluxes, temperature *T,* adsorption coeffs κ scattering coeff σ



Global approach involves too much communication

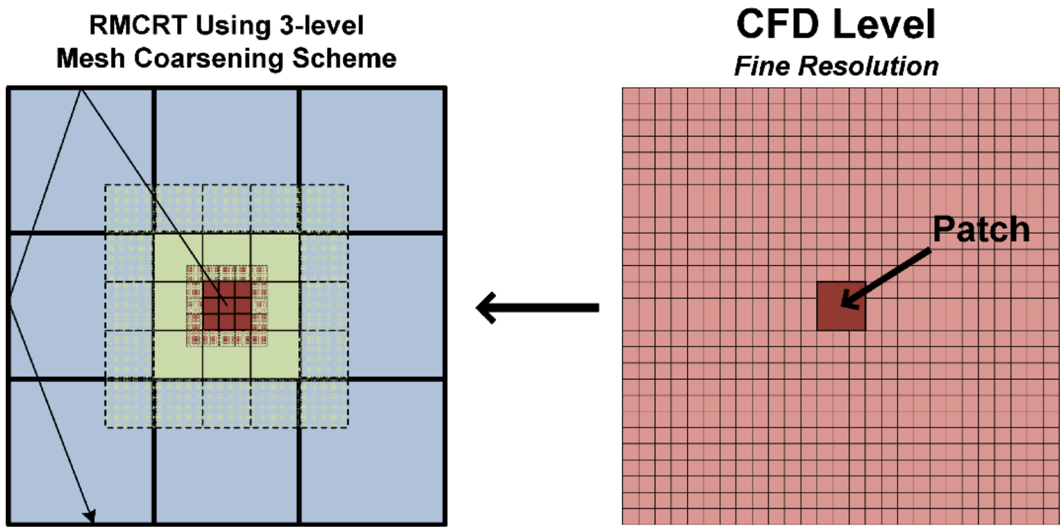
GPU Single Mesh Scaling



Multi-Level AMR GPU RMCRT

- **Use multilevel representation of computational domain**
- Define Region of Interest (**ROI**)
- Surround **ROI** with successively coarser grid
- As rays travel away from **ROI**, th stride taken between cells becomes larger
- **Transmit new information relating to heat fluxes adsorption and scattering coeffs using same adaptive ideas**

- This reduces computational cost, memory usage and MPI message volume.



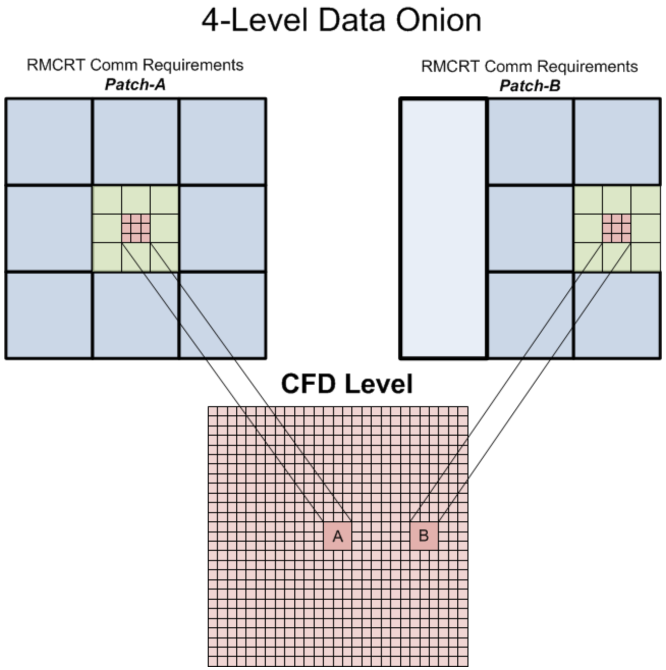
RMCRT - 2D diagram of three-level mesh refinement scheme, illustrating how a ray from a fine-level patch (right) might be traced across a coarsened domain (left).

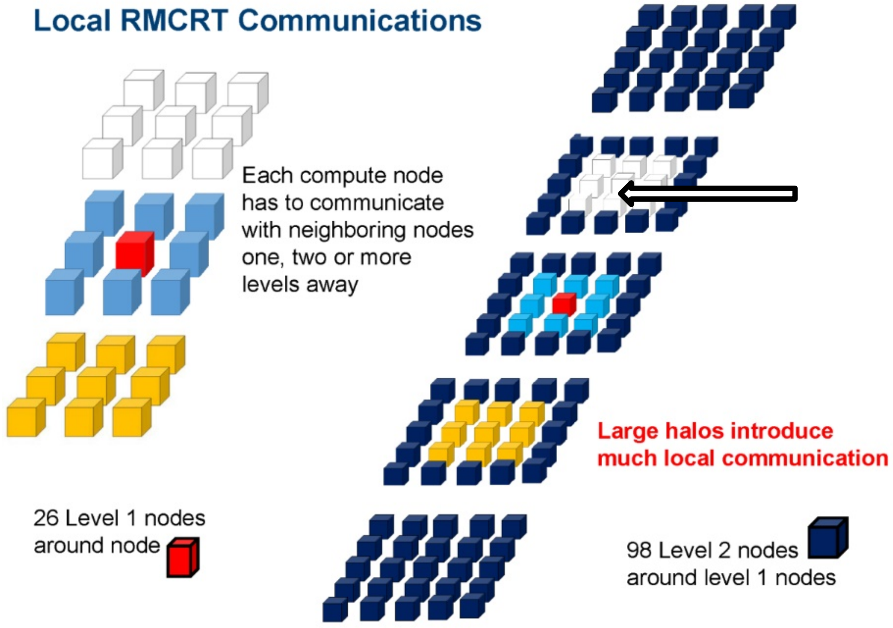
We have production-grade single and multi-level GPU-RMCRT for Titan

Reducing Communication with AMR

Single Fine Mesh – Limits Scalability

- Each cell communicated to every other cell in domain.
- <u>**N**</u> fine mesh cells == <u>**N²**</u> total communication.
- Global comm overwhelms system at large core counts





<u>Need AMR mesh at large core counts</u>

- Fine mesh used locally
- Coarse mesh used further away
- Shared coarse mesh per node
- Refinement ratio of 4 reduces global comm by factor of 64 over fine mesh approach

Partitioned Ray Tracing Algorithm

Original

(i) Distribute copy of **fine mesh** to each node

(ii) Ray-trace in parallel on each node all the rays from the "**home**" mesh blocks on that node across **fine mesh**

(iii) Update the heat fluxes Temperature absorption coeffs and scattering coefficients by global sends of **fine mesh** values

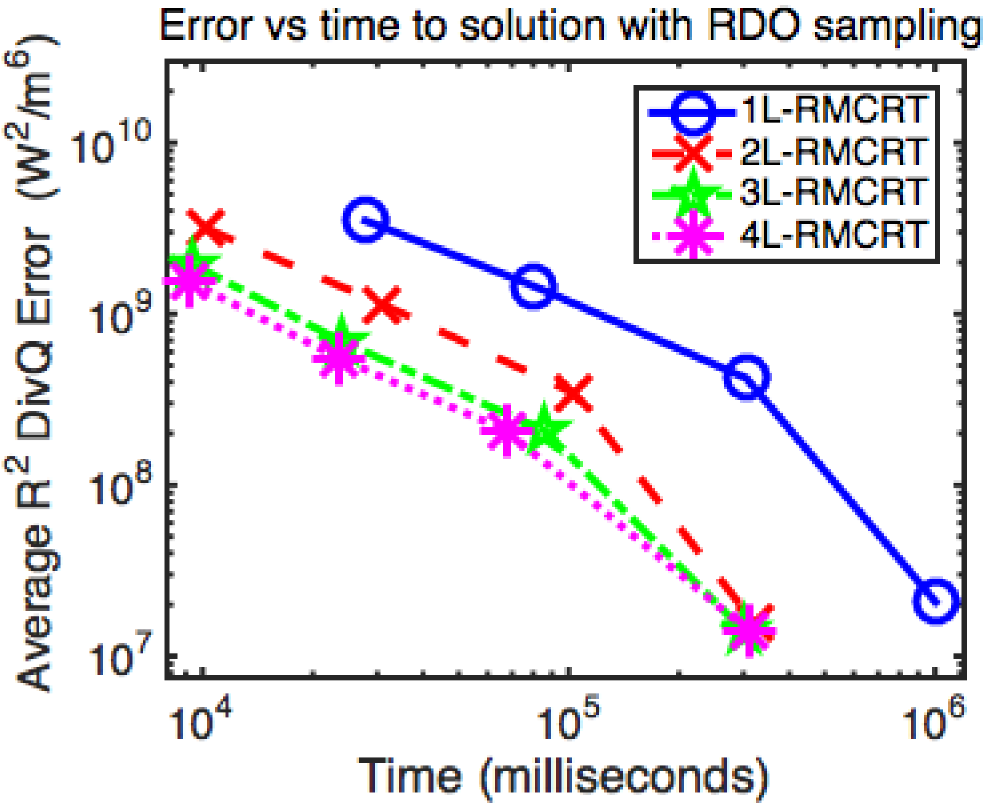
AMR Version

(i) Distribute copy of **coarse mesh** to each node

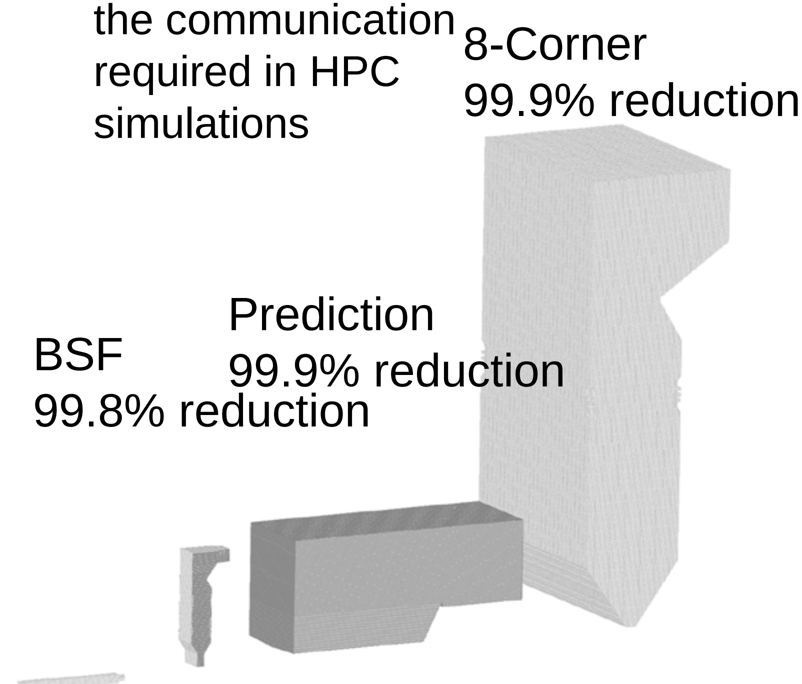
(ii) Ray-trace in parallel on each node all the rays from the "home" fine mesh blocks on that node to the close-by fine patches and the **coarse mesh**

(iii) Update the heat fluxes Temperature absorption coeffs and scattering coefficients by global sends of **coarse mesh values**

Computational Benefits of ML-RMCRT over 1L-RMCRT



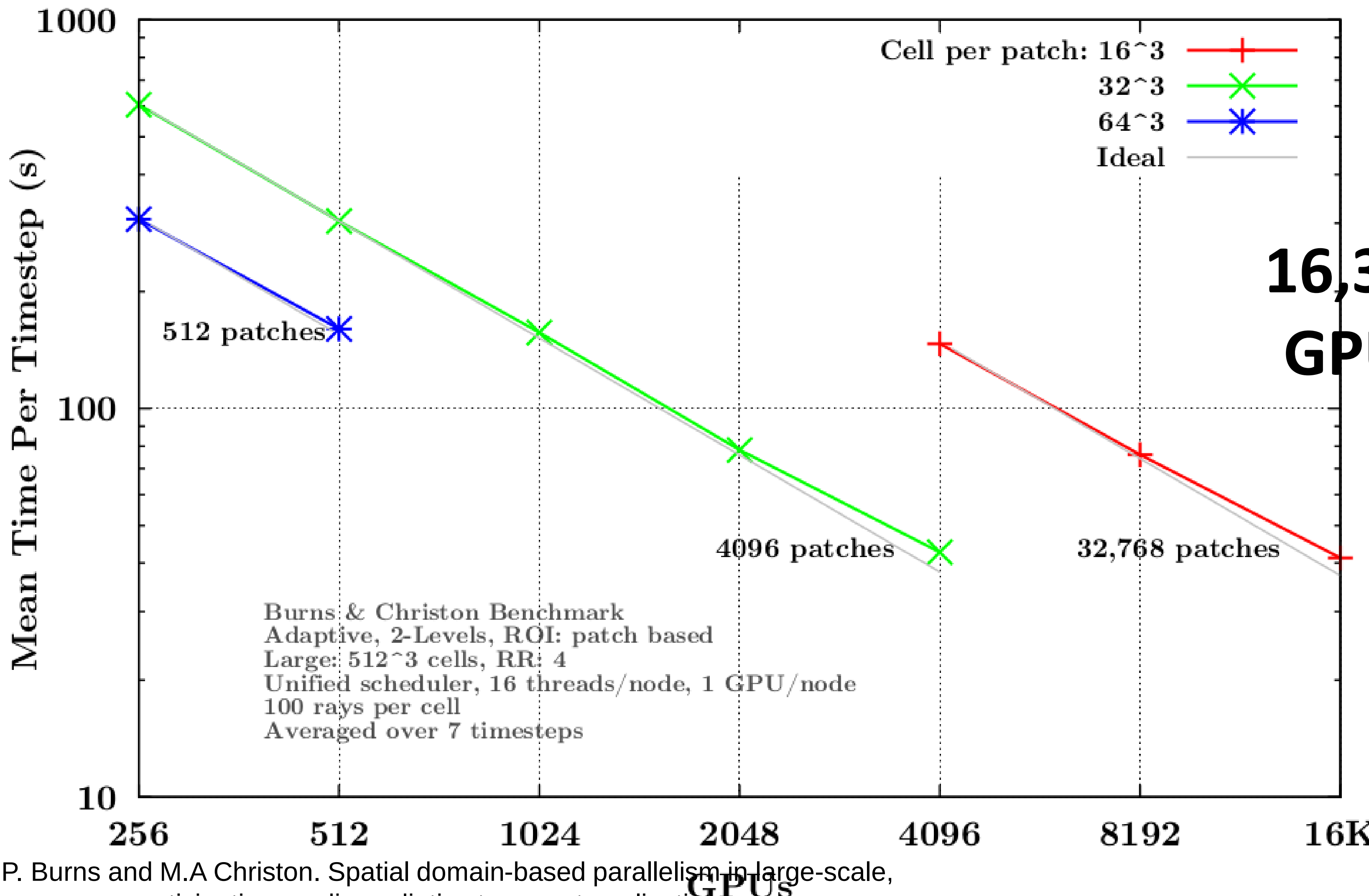
The use of 4 Levels in ML-RMCRT greatly reduces the communication required in HPC simulations



500-1000x faster communication,
lower memory footprint

~5x faster ray tracing
Communications volume
Reduced by a factor of 8x
With 2 level RMCRT 64x with 3 level RMCRT

GPU Strong Scaling on DOE Titan

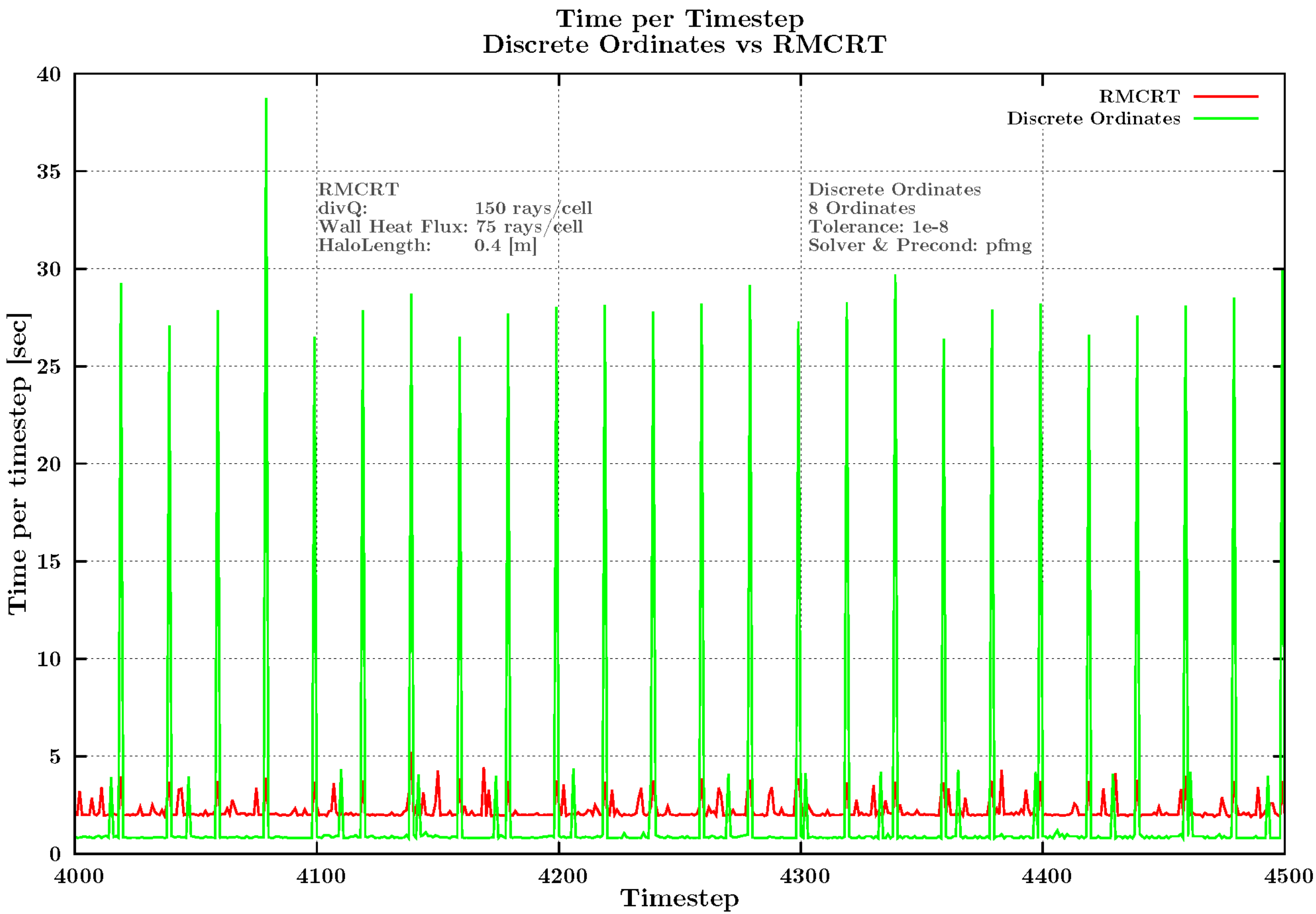


16,384 GPUs

S. P. Burns and M.A Christon. Spatial domain-based parallelism in large-scale, participating-media, radiative transport applications. Numerical Heat Transfer, Part B, 31(4):401-421, 1997.

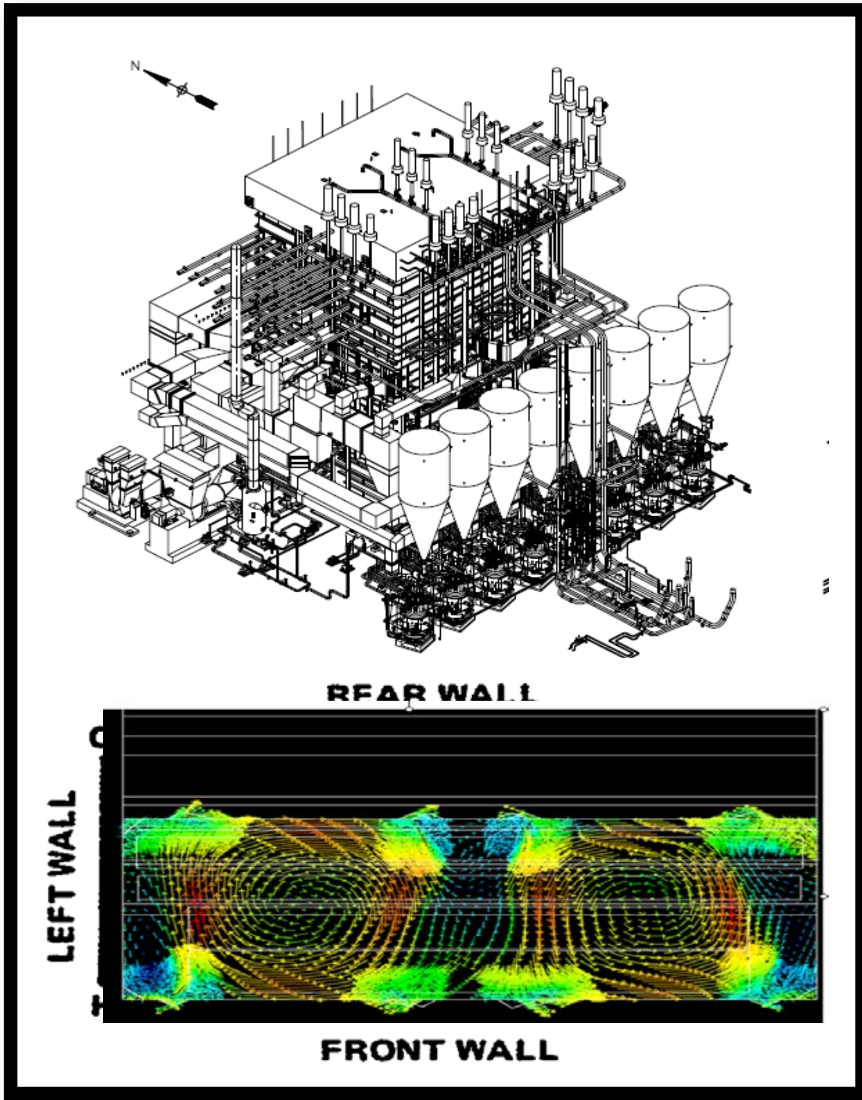
Alan Humphrey, Dan Sunderland, Todd Harman

RMCRT vs Discrete Ordinates Todd Harman



2016 INCITE RUNS

- GE/Alstom power USC "flagship" boiler ~430 inlets, 65m x35m x 15m 450M grid cells
- feed: 130 kg/s of coal (100 train cars of coal per day),O_2 from 1000 kg/s of air
- division panels, platens, super-heaters and re-heater tubing ~210 miles of piping
- walls, and tubing made of 11 different varied thickness metals
- 4 runs 450M hours
- Optimize design with LES runs
- Assess combustion efficiency, heat flux distribution and optimal position of inlets



Summary of Infrastructure Improvements

- **Modernization of Uintah codebase**
  - Adoption of C++11 standard – removal and update of old threads library etc 10K lines of code
  - Better performance, portability and developer productivity
  - Using standard library for concurrency APIs, algorithms, etc.
- **Significant infrastructure modifications required for INCITE runs**
  - Addressing taskgraph scalability issues and added support for multiple primary taskgraphs – temporal scheduling
- **Advancements to Uintah GPU capabilities for INCITE runs**
  - Improved GPU concurrency allows multiple concurrent kernel executions
  - Added support for patch memory combining to reduce GPU memory footprint

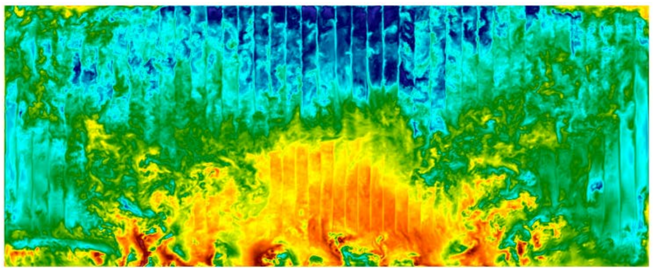
Infrastucture Improvement Results

Operation	Before	After
Nodal Memory footprint (GB)	21	2.5
Timestep (sec)	10-11	~3
Pressure Solve (sec)	2.5 - 5	1 – 2.3
Task wait comm time (sec)	3.25	0.065
Taskgraph compile	4.5 hours	15 minutes*
Time to solution (sec)	2590	1012

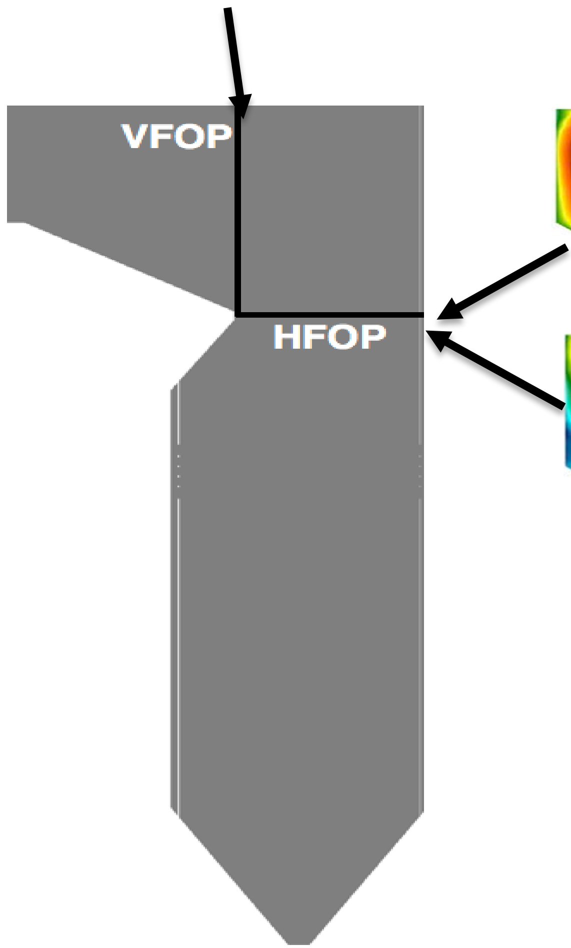
Results obtained from 128k core Titan run using 2-level RMCRT (CPU baseline)

** Currently working to reduce this to ~2-3 minutes total for both taskgraphs*

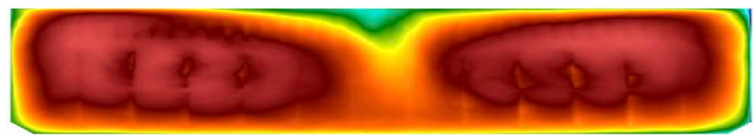
Instantaneous enthalpy flux ($J/m^2/s$) profile at the VFOP for Case 1.



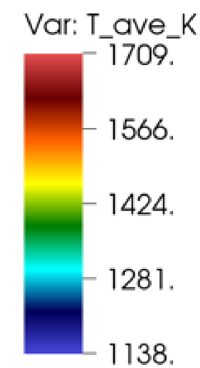
INCITE Results
Runs 1 and 2 100Mcpu hrs
455M cells



Case 1 temperature



Case 2 temperature



Science accomplishments:
validated our design process
now driving field expts
achieve results that conventional RANS cannot

Kokkos Abstraction for Portability and Node Performance

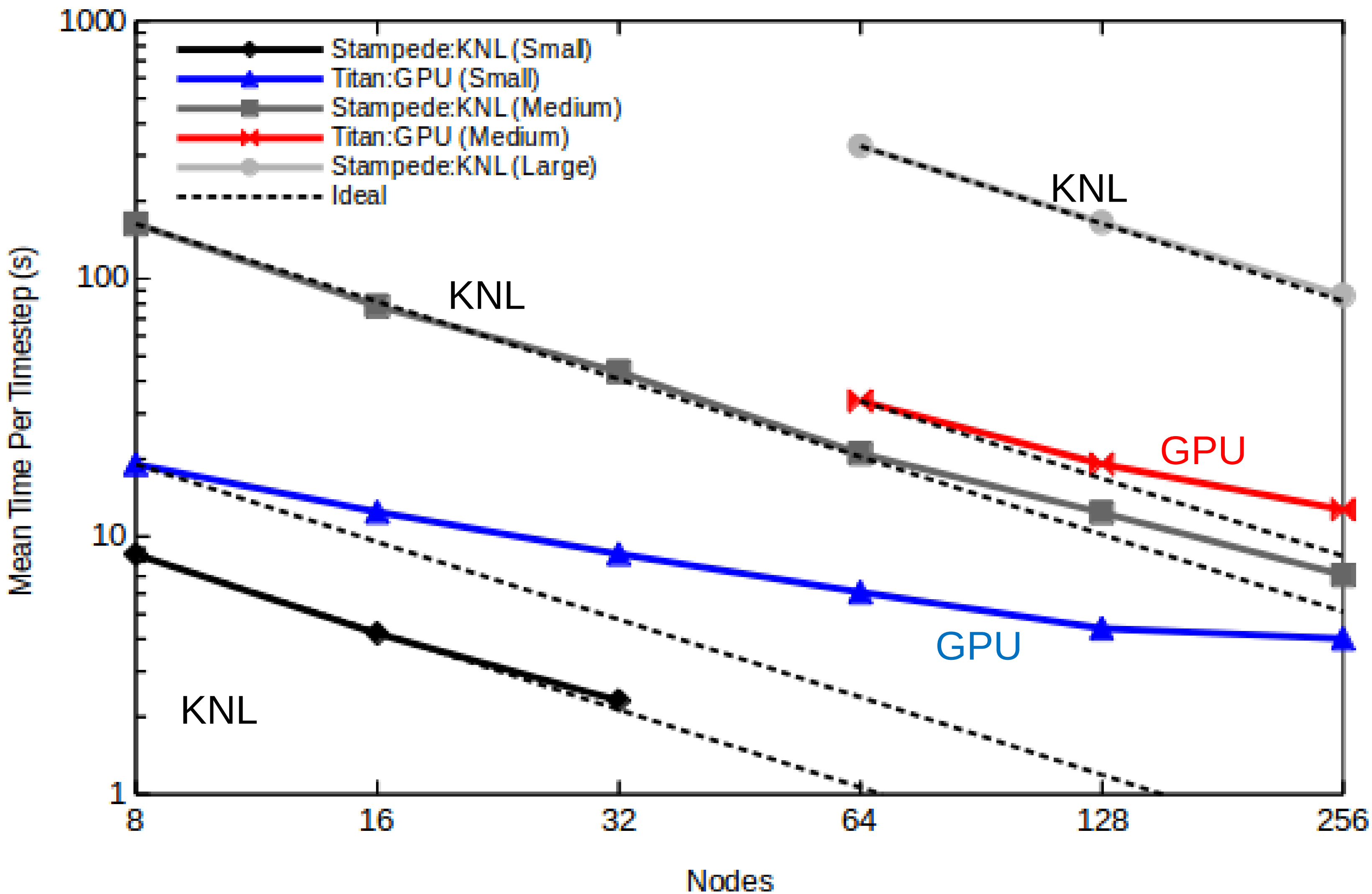
- **Use Kokkos (or Raja) abstraction layer that maps loops onto machine (CPU GPU MIC) efficiently using cache aware memory models and vectorization / Openmp**
- Use **C++ template metaprogramming** for compile time data structures and functions and allow vectorization
- **Incremental** refactor to Kokkos parallel patterns/views
- **Replace patch grid iterator loops**
- **for** (auto itr = patch.begin(); itr != patch.end(); ++itr) {
  IntVector iv = *itr;
  A[iv] = B[iv] + C[iv];}

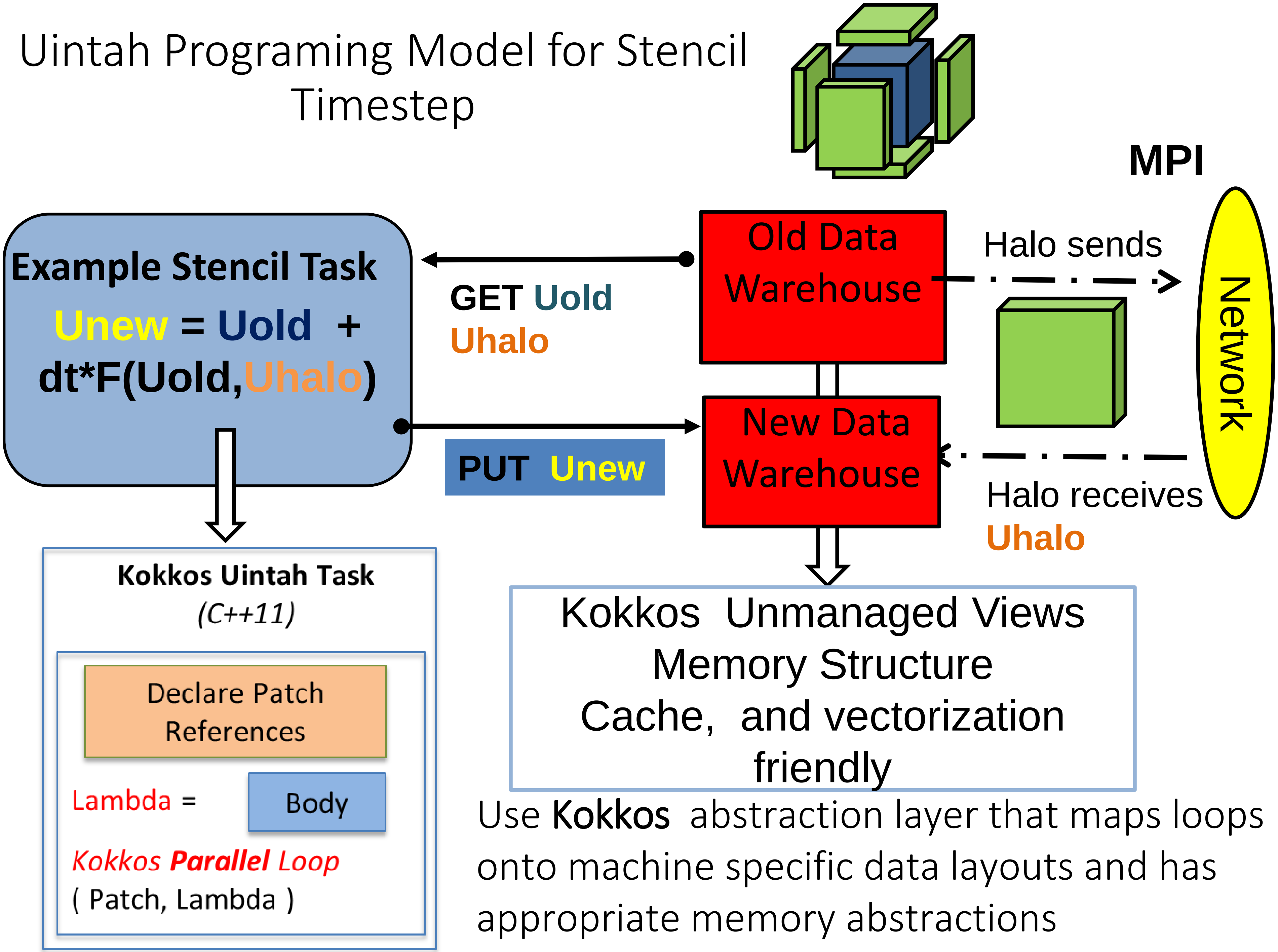
BECOMES

- **parallel_for**(patch.range(), LAMBDA(int i, int j, int k) {
  A(i,j,k) = B(i,j,k) + C(i,j,k)});
- Users prefer this to DSL

Early KNL Results Multi-Level RMCRT Comparisons with Titan GPUs

1 MPI Rank per node 16 threads per GPU 64 threads per KNL




Uintah Programing Model for Stencil Timestep
MPI
Example Stencil Task
Unew = Uold + dt*F(Uold,Uhalo)
GET Uold
Uhalo
Old Data Warehouse
Halo sends
Network
PUT Unew
New Data Warehouse
Halo receives
Uhalo
Kokkos Uintah Task
(C++11)
Declare Patch References
Lambda =
Body
Kokkos Parallel Loop
(Patch, Lambda)
Kokkos Unmanaged Views
Memory Structure
Cache, and vectorization friendly
Use Kokkos abstraction layer that maps loops onto machine specific data layouts and has appropriate memory abstractions

Summary

(i) Radiation AMR approach used both in raytracing and communications makes scalability of ray tracing possible to 250K CPUs and 16K GPUs
(ii) Results on industrial scale simulations for GE
(iii) Required improvements in core infrastructure
(iv) Need to think about performance portable approaches using Kokkos
(v) Work underway right now.

Thanks DOE NNSA INCITE ORNL ANL GE/ALSTOM