Bringing (Clean) Coal Combustion to Drax via Computational Modeling and Software Abstractions for Exascale

Martin Berzins

with slides from James Sutherland, Chuck Hansen, Valerio Pascucci, Phil Smith and Jeremy Thornock

(i) The Changing Nature of Computational Science
(ii) Clean Coal Boiler Design
(iii) The Uintah framework
(iv) Uintah Scalability
(v) Solvers EDSLs Visualization
(vi) Designing for Exascale
(vii) Conclusions

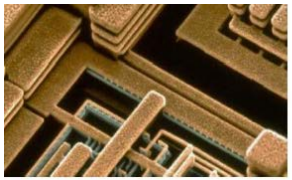
Funding thank DOE ASCI (97-10), NSF , DOE NETL+NNSA ARL , NSF , INCITE, XSEDE

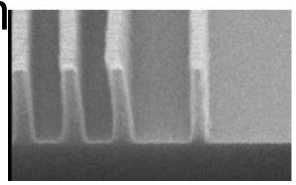
The Changing nature of Computational Science

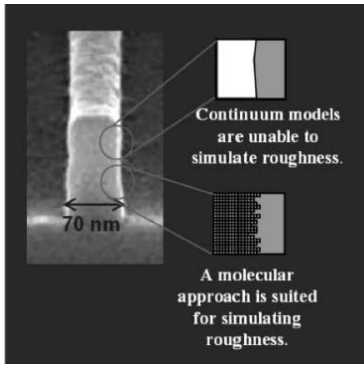
- The need for predictive simulations
- The move towards Exascale Computing

Predictive Computational Science [Oden Karniadakis]

Predictive Computational (Materials) Science is changing e.g. nano-maufacturing





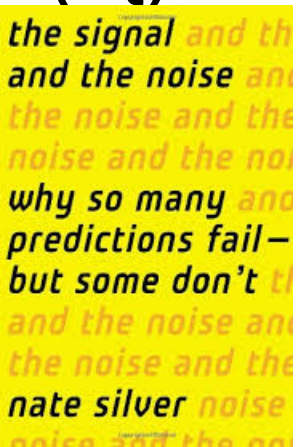


Science is based on subjective probability in which predictions must account for uncertainties in parameters, models, and experimental data . This involves many "experts" who are often wrong

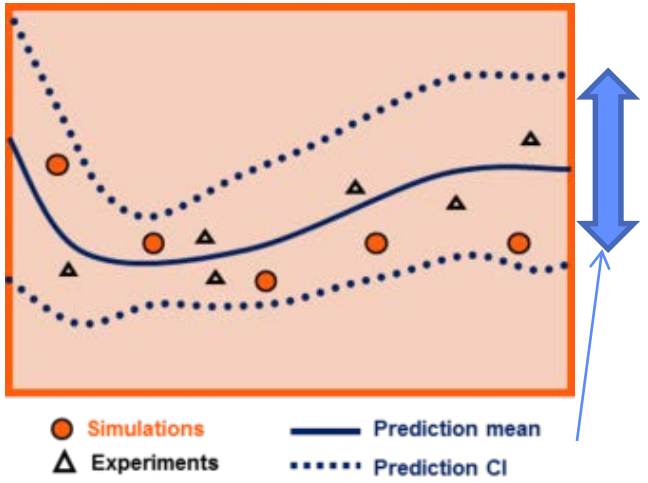
We cannot deliver predictive materials by design over the next decade without quantifying uncertainty

Predictive Computational Science:

Successful models are verified (codes) and validated (experiments) **(V&V)**. The uncertainty in computer predictions (the QoI's) must be quantified if the predictions are used in important decisions. **(UQ)**



"Uncertainty is an essential and non-negotiable part of a forecast. Quantifying uncertainty carefully and explicitly is essential to scientific progress." Nate Silver



Confidence interval

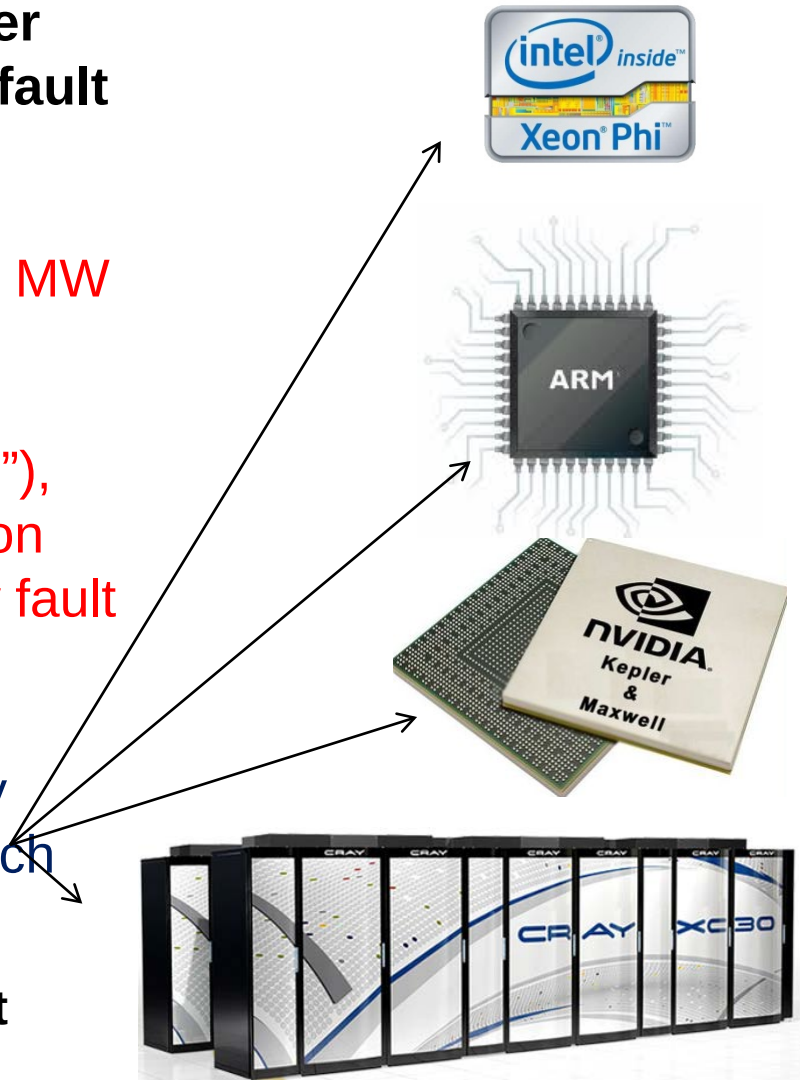
The Challenge for Future Software?

2013 Titan, Blue Gene Q - 2 Petaflops per MegaWatt 300K cpus 5M gpu cores h/w fault every 12 hours

202X Exascale "goal" $\Longrightarrow$ 50 Petaflops per MW
Or **20pJ per op.**

Many more cores (majority on "accelerators"), variable Power consumption. Communication delays. Many more component failures. h/w fault every 14 mins?

Great uncertainty in architectures probably accelerator-based machines that will be much more energy efficient.

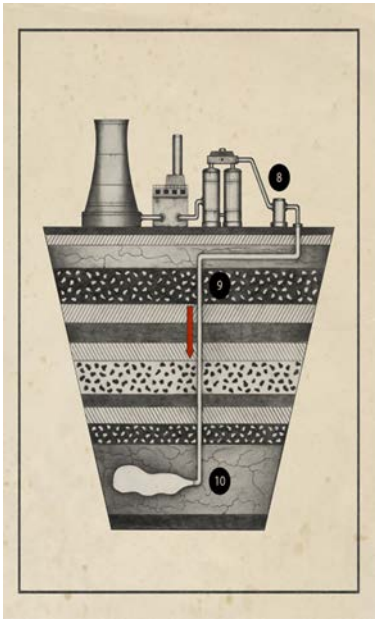


Exascale also means Petascale in a cabinet

Can we move from petascale (10^{15} flops) to exascale (10^{18} flops) computing for real engineering problems?

Clean Coal Boiler Design using Predictive Computational Science

- Can we help design the next generation of clean coal boilers?
- The CCMSC team
- The Application
- Current Simulations

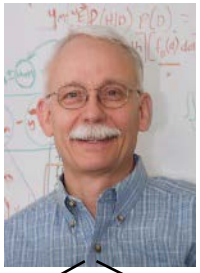


Wired Magazine
BY CHARLES C. MANN 03.25.14 |
Renewables Aren't Enough.
Clean Coal Is the Future

CARBON CAPTURE
MULTIDISCIPLINARY
SIMULATION CENTER

The team

The Exec


Computer Science Predictive Modeling Uncertainty Quantification












ALSTOM
Institute for CLEAN AND SECURE ENERGY THE UNIVERSITY OF UTAH
SCI www.sci.utah.edu
Berkeley UNIVERSITY OF CALIFORNIA
BYU BRIGHAM YOUNG UNIVERSITY
NNSA National Nuclear Security Administration

Taking UINTAH-X beyond petascale?

(i) UintahX Runtime System
(ii) Wasatch Nebo Domain Specific Approach (James and Matt)
(iii) Visus PIDX and Visit Visualization (Valerio and Chuck)

Todd Harman

Allen Sanderson

Dav de St Germain

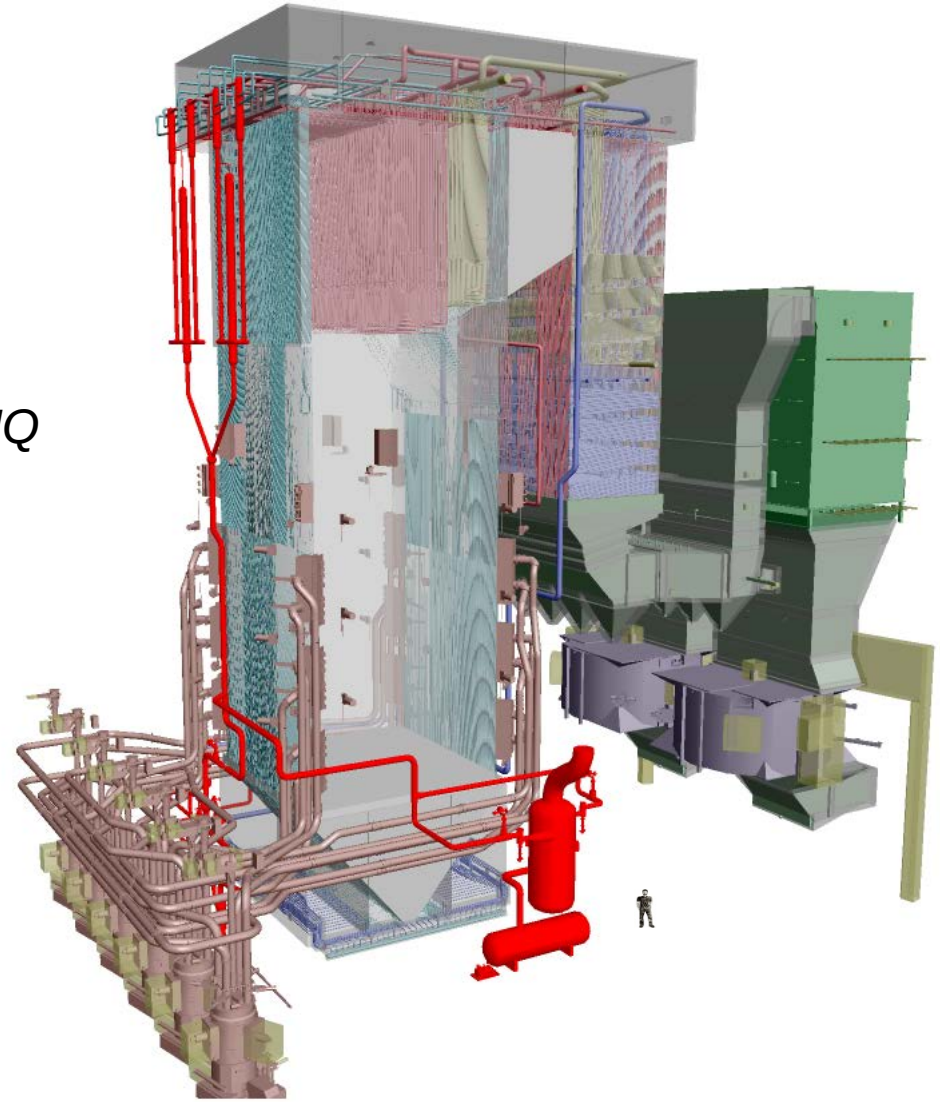
John Schmidt

Alan Humphrey

Thanks to Qingyu Meng (Google) and Justin Luitjens (NVIDIA)

Overarching Application

- high efficiency advanced ultra-supercritical (AUSC) oxy-coal tangentially-fired power boiler



extreme computing
- *predictive science w hybrid validation/UQ*
- expensive function evaluation
- expensive data
- *rapid design and deployment w Alstom*

ALSTOM

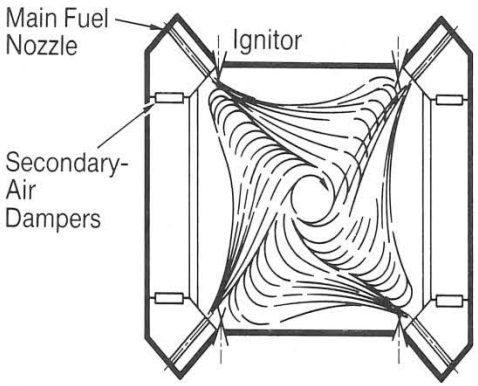
- *global reach:*

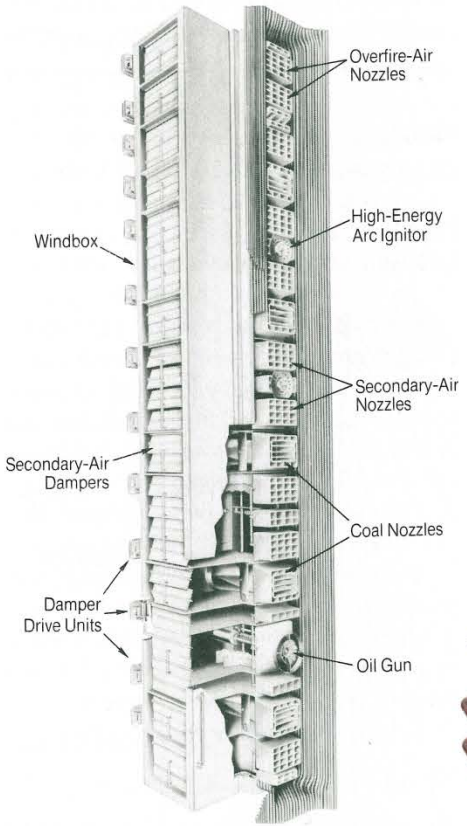
present in 100 countries
- *2011/12 sales:*

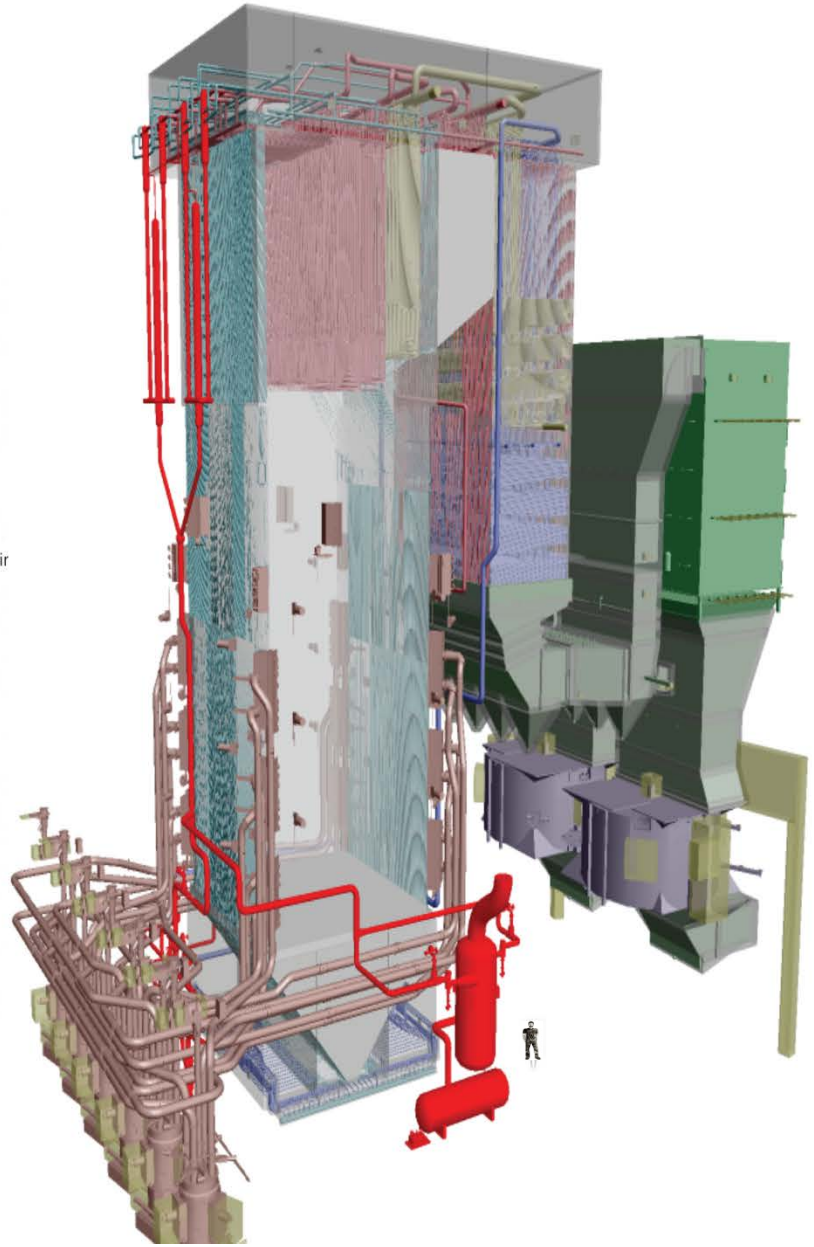
$26.5 billion
- *93,000 employees*

GE takeover in progress

Pulverized Coal Power Generation







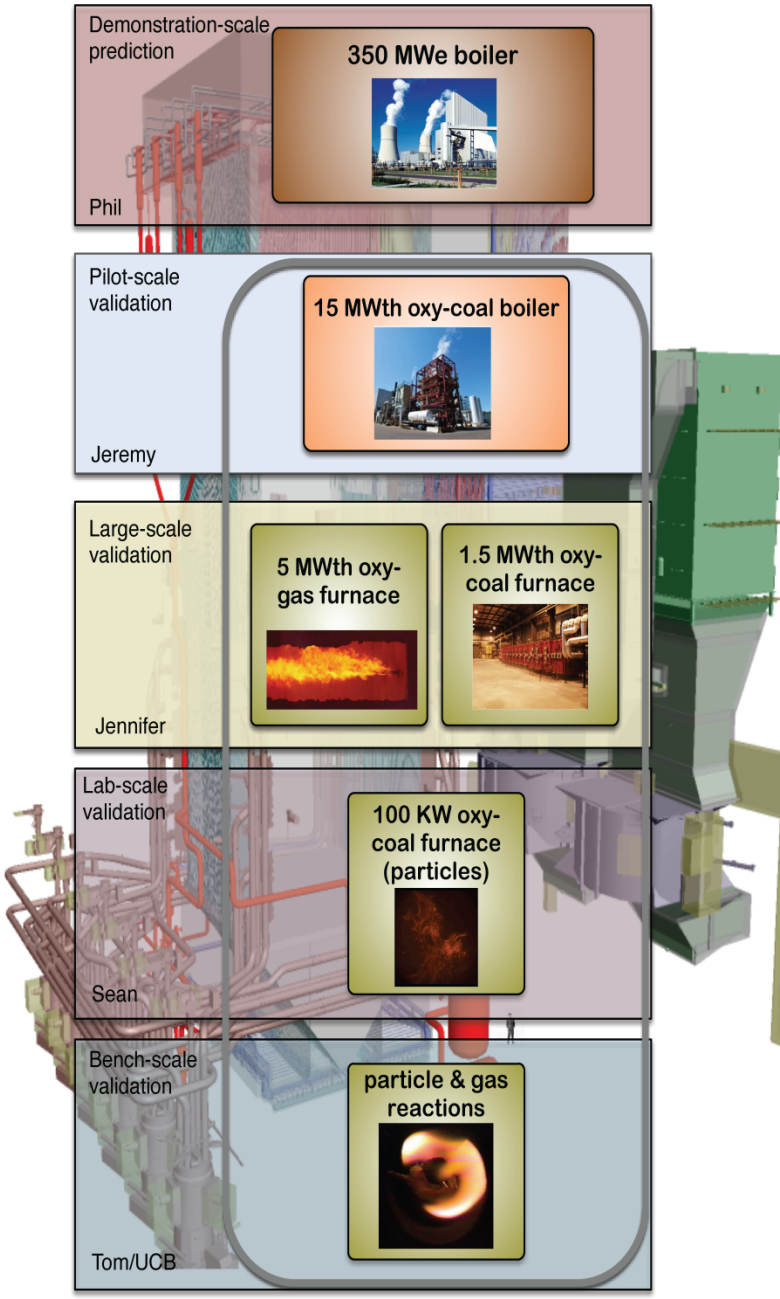
CARBON CAPTURE MULTIDISCIPLINARY SIMULATION CENTER

Specific Goal

overarching prediction

- predict *heat flux profile* for 350MW oxy-coal AUSC
  - *extrapolation*
  - *no experimental data @ demonstration scale*
  - *UQ predictive design: produce uncertainty in QOI that is 'consistent' with all experimental observations in hierarchy*
- secondary QOIs:
  - *boiler efficiency*
  - *exhaust NOx*
  - *unburned carbon in ash*
- "embrace uncertainty"
  - *quantify uncertainties in mmts. & sims.*
  - *V/UQ process for decision making in the presence of uncertainty*
- accelerate deployment of a new technology:
  - *high efficiency carbon capture for pc power generation*

CARBON CAPTURE
MULTIDISCIPLINARY
SIMULATION CENTER



A-USC

- AUSC:
  - *760C steam temperature*
  - *>12% reduction in fuel consumption carbon dioxide emissions over 600C USC units*
  - *materials - expensive*

- what will change from USC:
  - *superheater and reheater tube banks will use materials like 740H and 230 nickel.*
  - *furnace enclosure material in the upper enclosure is creep strength enhanced ferritic steel requiring field PWHT of the tube to tube joints and possibly the membrane panel seams.*
  - *steam piping is 740H nickel or better.*

- boiler arrangement / design is evolving
  - *price of nickel is high that the position of the steam turbine must be a consideration*
  - *Siemens AG has proposed a horizontal boiler*

CARBON CAPTURE
MULTIDISCIPLINARY
SIMULATION CENTER

ALSTOM

SIEMENS

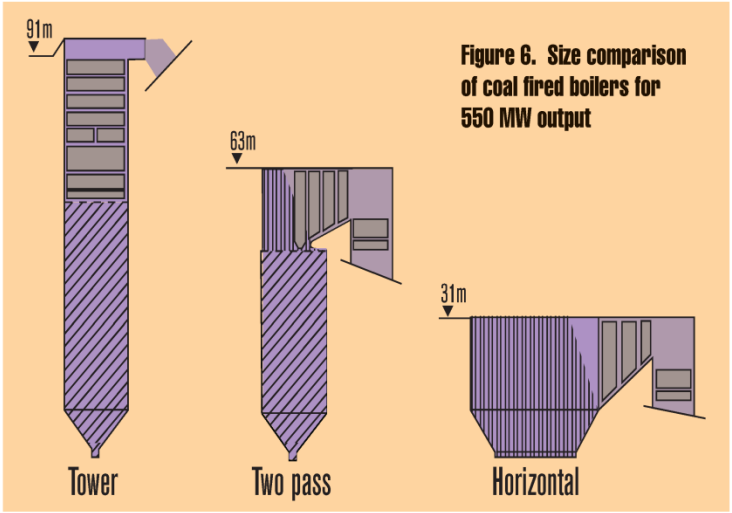
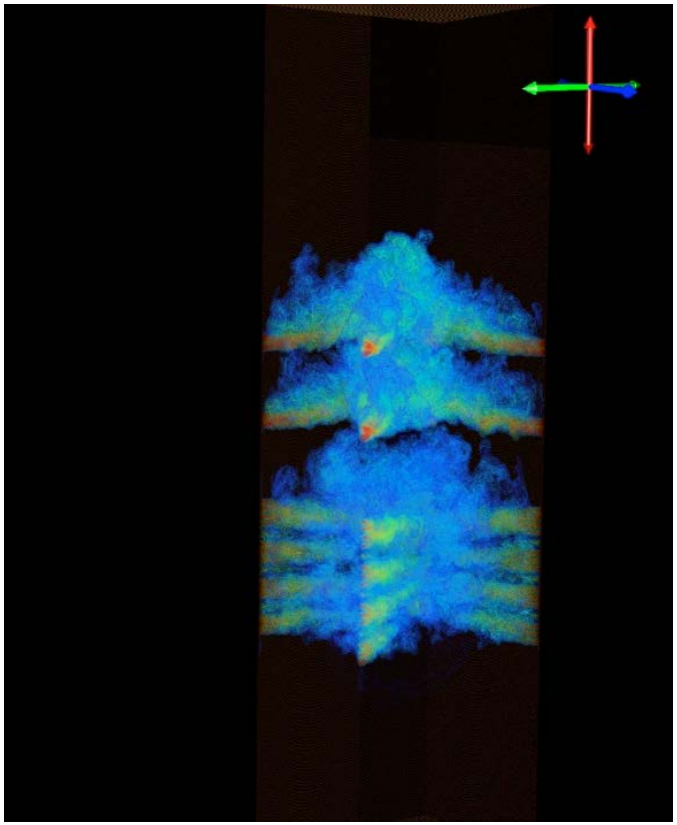


Figure 6. Size comparison of coal fired boilers for 550 MW output

Existing Simulations of Alstom Clean coal Boilers



Temperature field

For 350MWe boiler problem. LES resolution needed: 1mm per side for each computational volume = 9x 10**12 cells
This is 1000x our largest simulations on 764K cores. - to run in 48 hours of wall clock time requires 50-100M fast cores.

Dr Jeremy Thornock ICSE

Existing Simulations of Alstom Clean coal Boilers using ARCHES in Uintah

(i) Traditional Lagrangian/RANS approaches do not address well partile effects

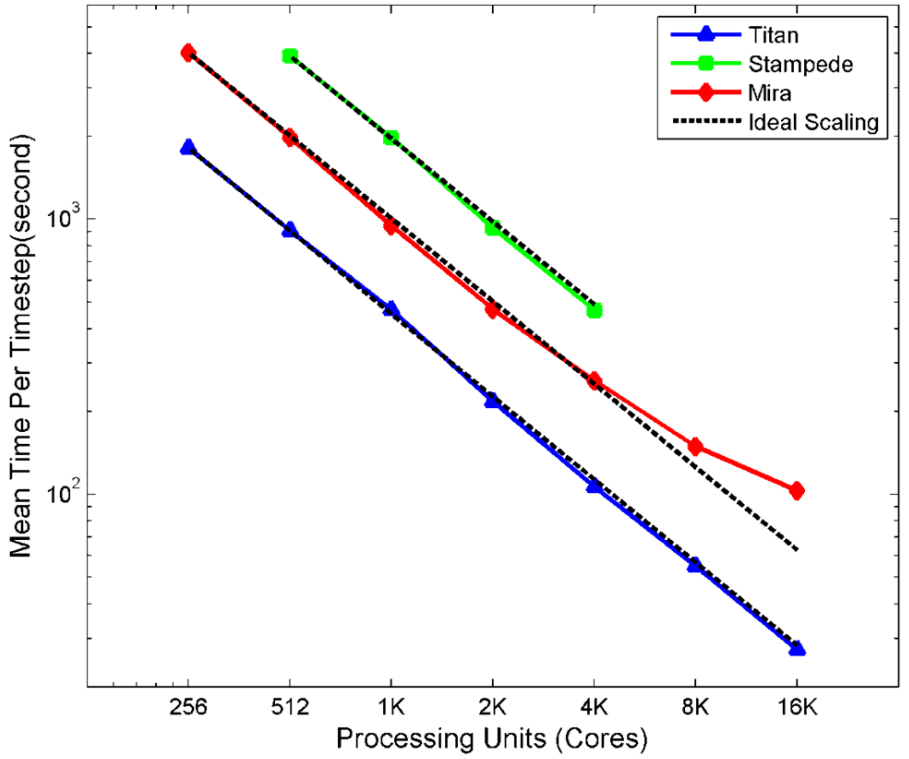
(ii) LES has potential to predict oxy---coal flames and to be an important design tool

(iii) LES is “like DNS” for coal

- Structured, finite-volume
- Mass, momentum, energy with radiation
- Higher-order temporal and spatial numerics
- LES closure
- Tabulated chemistry
- PDF mixing models
- DQMOM

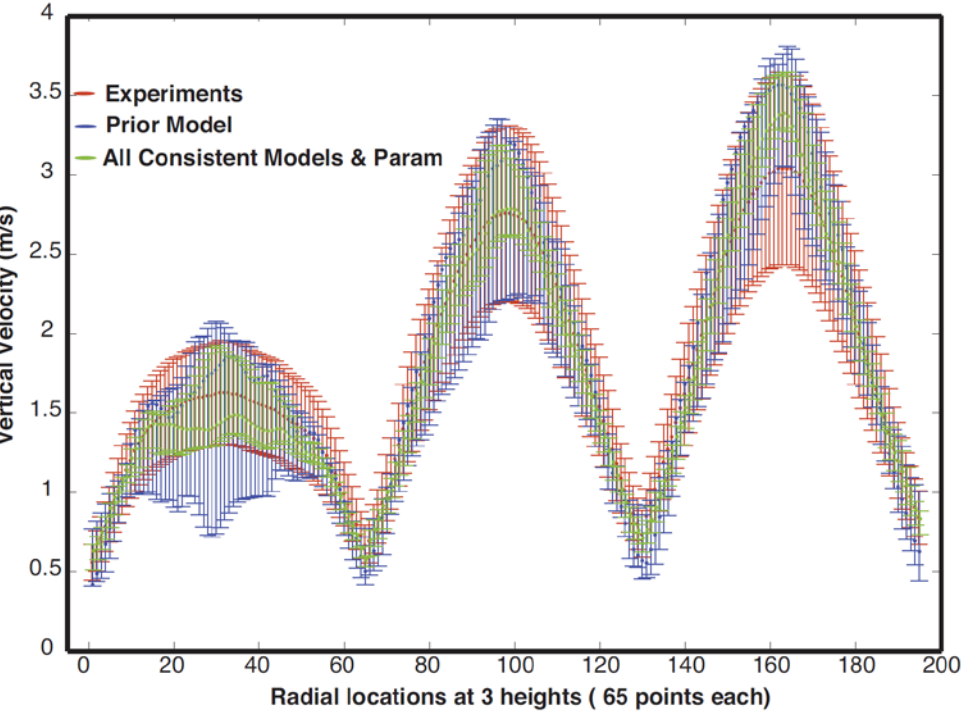
Computational challenges at these scales

- Uncertainty quantification. How reliable is it?
- Modeling Particles
- Radiation (see last years talk for Ray Tracing).
- Solving linear systems



Strong Scaling Radiation Problem

Verification Validation Uncertainty Quantification State of the Art with Buoyant Helium Plume Model

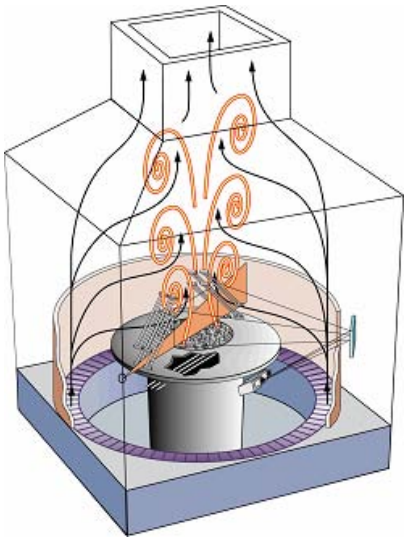


Red is experimental uncertainty

Blue is uncertainty region from simulation

Green is uncertainty in vertical velocity consistent with experimental data and input parameters

Turbulent combustion problem typical of any real life cases, experiments at Sandia Labs



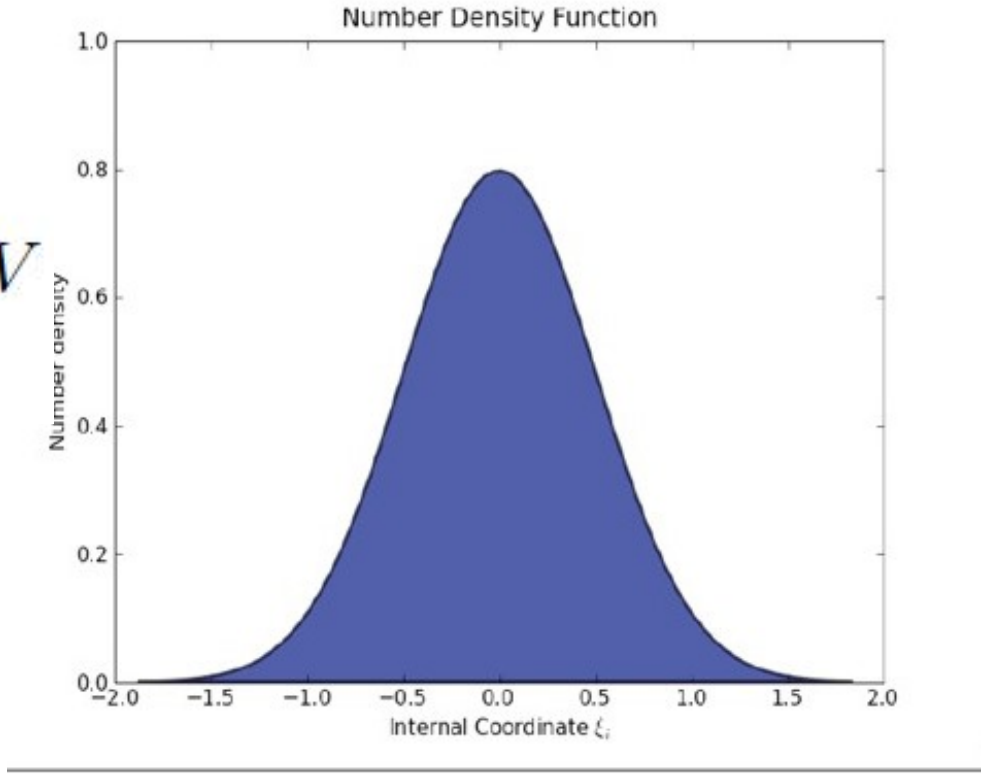
Sources: Smith Schmidt

DQMOM Equations: Number Density Function (NDF)

The NDF describes the number of particles per volume as a function of several particle independent variables (e .g . , particle diameter , particle composition, etc.) called internal coordinates.

Given a volume V and a set of internal coordinates **ξ**, the total number of particles in this volume is:

$$N = \int_V \int_\xi f(\xi, x, t) d\xi dV$$



Julian Pedel Thesis Institute for Clean and Secure Energy 2014

Population Balance

$$\frac{\partial n}{\partial t} + \sum_i \frac{\partial u_i(\xi, t) n}{\partial \xi_i} = h_i(\xi_i, t)$$

Moment

$$m^i = \int \xi^i n(\xi; x, t) d\xi$$

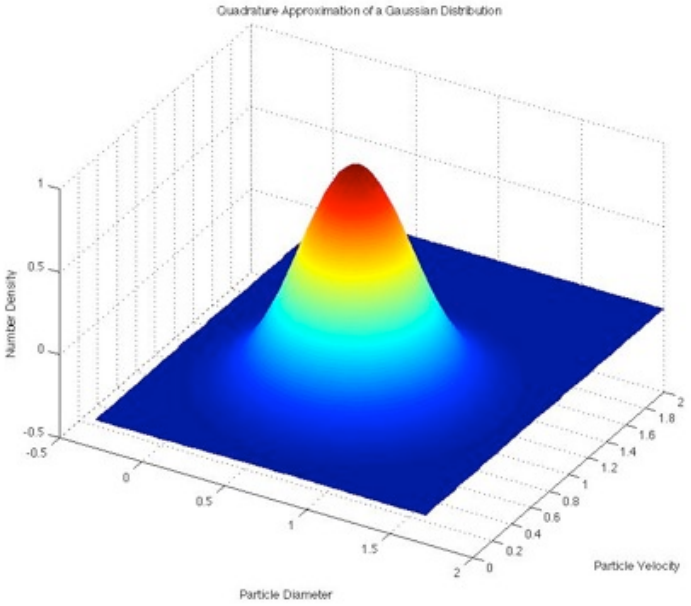
Quadrature

$$n(\xi; \mathbf{x}, t) \approx \sum_{\alpha=1}^{N} w_\alpha(\mathbf{x}, t)\, \delta\left(\xi - \langle \xi \rangle_\alpha (\mathbf{x}, t)\right)$$

Environment

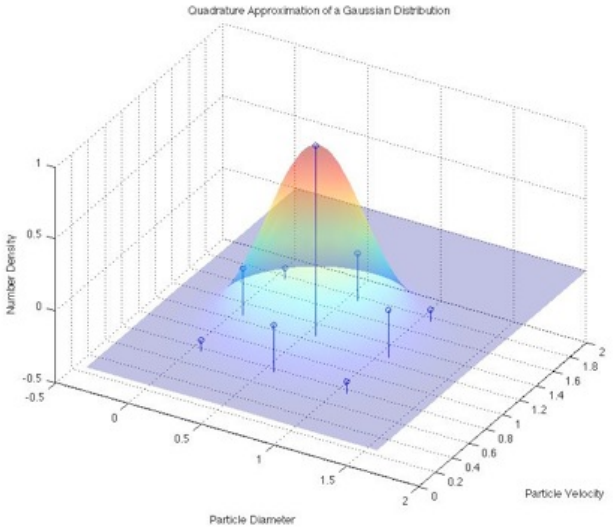
$$\frac{\partial w_\alpha}{\partial t} + \ldots = S_{w_\alpha}$$

$$\frac{\partial w_\alpha \langle \xi_\alpha \rangle}{\partial t} + \ldots = S_{w_\alpha \langle \xi_\alpha \rangle}$$



w_α number of particles per vol. assoc with node

$$\begin{aligned}
\frac{\partial}{\partial t}(w_\alpha) + \frac{\partial}{\partial x_i}(\langle u_i \rangle_\alpha w_\alpha) - \frac{\partial}{\partial x_i}((\langle u_i \rangle_\alpha - u_{i,f}) w_\alpha) &= a_\alpha \\
\frac{\partial}{\partial t}(\varsigma_\alpha) + \frac{\partial}{\partial x_i}(\langle u_i \rangle_\alpha \varsigma_\alpha) - \frac{\partial}{\partial x_i}((\langle u_i \rangle_\alpha - u_{i,f}) \varsigma_\alpha) &= b_\alpha
\end{aligned}$$



DQMOM Numerical Issues

- Abscissas values are obtained by dividing weighted abscissas by weights: problem when weights are null
- Need to know to a_α and b_α to transport weights and weighted abscissas:

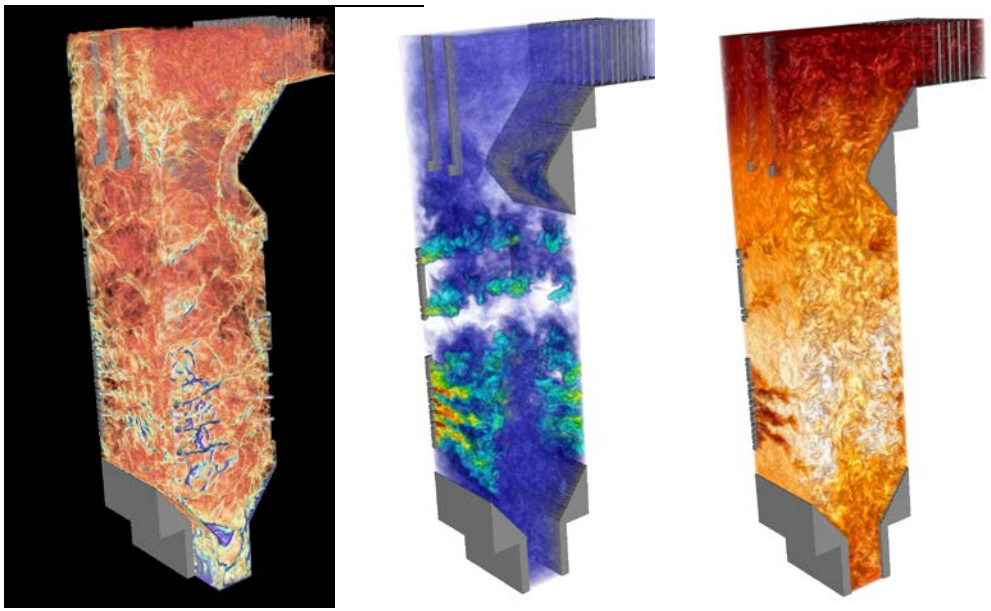
$$\begin{aligned}
\frac{\partial}{\partial t}\left(w_\alpha\right) + \frac{\partial}{\partial x_i}\left(\langle u_i\rangle_\alpha w_\alpha\right) - \frac{\partial}{\partial x_i}\left(\left(\langle u_i\rangle_\alpha - u_{i,f}\right) w_\alpha\right) &= a_\alpha \\
\frac{\partial}{\partial t}\left(\varsigma_\alpha\right) + \frac{\partial}{\partial x_i}\left(\langle u_i\rangle_\alpha \varsigma_\alpha\right) - \frac{\partial}{\partial x_i}\left(\left(\langle u_i\rangle_\alpha - u_{i,f}\right) \varsigma_\alpha\right) &= b_\alpha
\end{aligned}$$

a_α and b_α are obtained by solving a linear system: Ax = B

Matrix A:
- size $N(N_\xi+1)$
- often ill-conditioned and has to be solved in every cell

V/UQ Assessment of a Large Eddy Simulation Tool for Clean-Coal Technology

- Demonstrate LES predictivity for oxy-coal applications
- Provide reference point for high-fidelity simulation tools
- Provide a predictive tool for modern boiler design and retrofit applications
- Advance the heterogeneous scaling capabilities of Uintah Computational Framework



Images (from left to right) of large coal particle distribution, oxygen concentration, and temperature throughout the boiler.

- First full boiler scale simulation using high-fidelity LES with parameter variation over input ranges (15M cpu hrs)
- Initial validation of LES results with experimental data
- Performance 2X better for the LES capability
- First-cut demonstration of the GPU reverse Monte-Carlo for performing radiation calculations
- Scaling demonstration of the Uintah hybrid scheduler (3M)
- GPU implementation of key pieces of the DQMOM solution process

Exascale and the UINTAH FRAMEWORK

The Exascale challenge for Future Software?

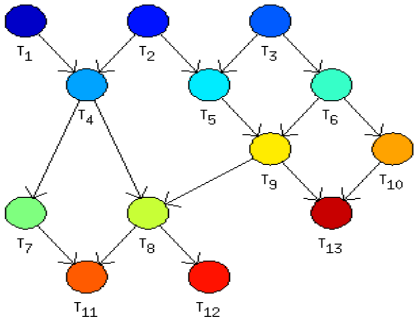
Harrod SC12: "today's bulk synchronous (BSP), distributed memory, execution model is approaching an efficiency, scalability, and power wall."

Compute

Communicate

Compute

Sarkar et al. "Exascale programming will require prioritization of critical-path and non-critical path tasks, adaptive directed acyclic graph scheduling of critical-path tasks, and adaptive rebalancing of all tasks……"

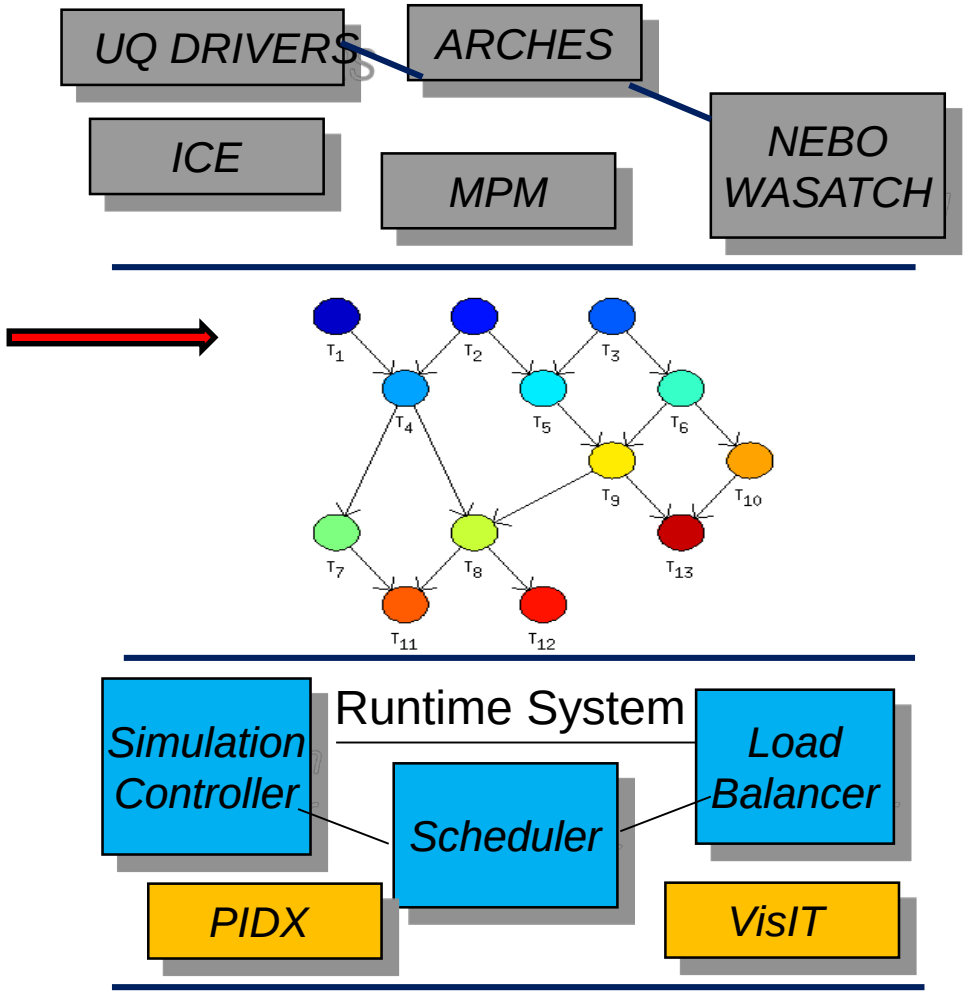


" DAG Task-based programming has always been a bad idea. It was a bad idea when it was introduced and it is a bad idea now " Parallel Processing Award Winner

Vivek Sarkar's thesis 1989 introduced many of the main ideas we use today. Of course everything is theoretically intractable see. Sinnen "Task Scheduling for Parallel Systems"

- **Application Specification** via ICE MPM ARCHES or NEBO/WASATCH DSL
- **Abstract task-graph** program that executes on:
- **Runtime System** with:Asynchronous out-of-order execution, work stealing
- Overlap communication & computation
- Tasks running on cores and accelerators

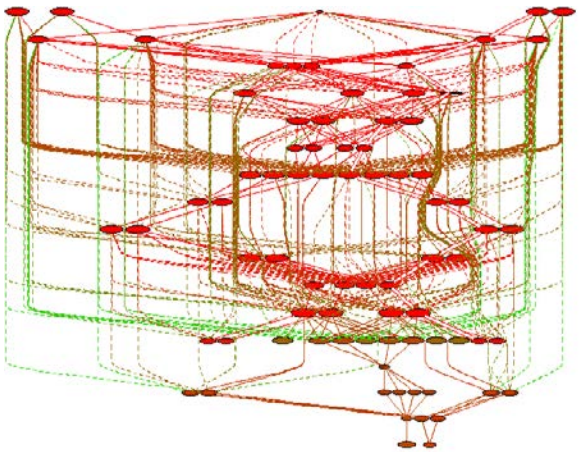
- **Scalable I/O via Visus PIDX**

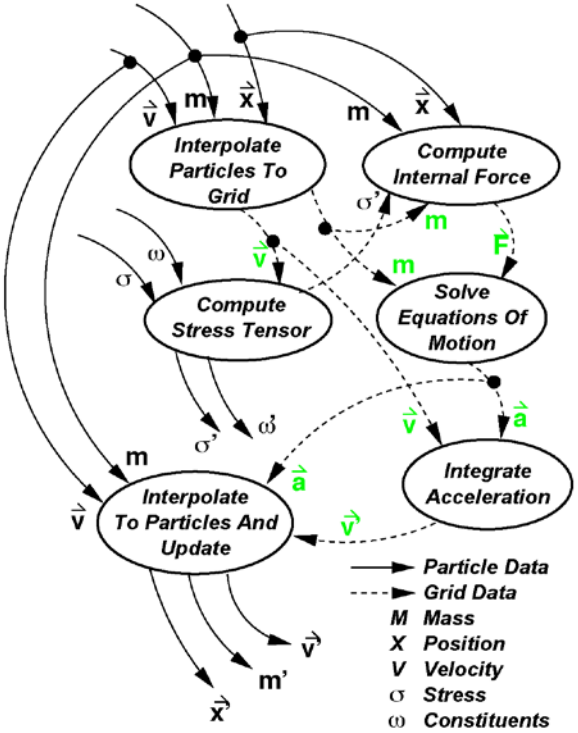


Uintah(X) Architecture Decomposition

The problem specs for some components have not changed as we have gone from 600 to 600K cores it is the Runtime System that changed

Uintah Directed Acyclic (Task) Graph-Based Computational Framework





Each task defines its computation with required inputs and outputs

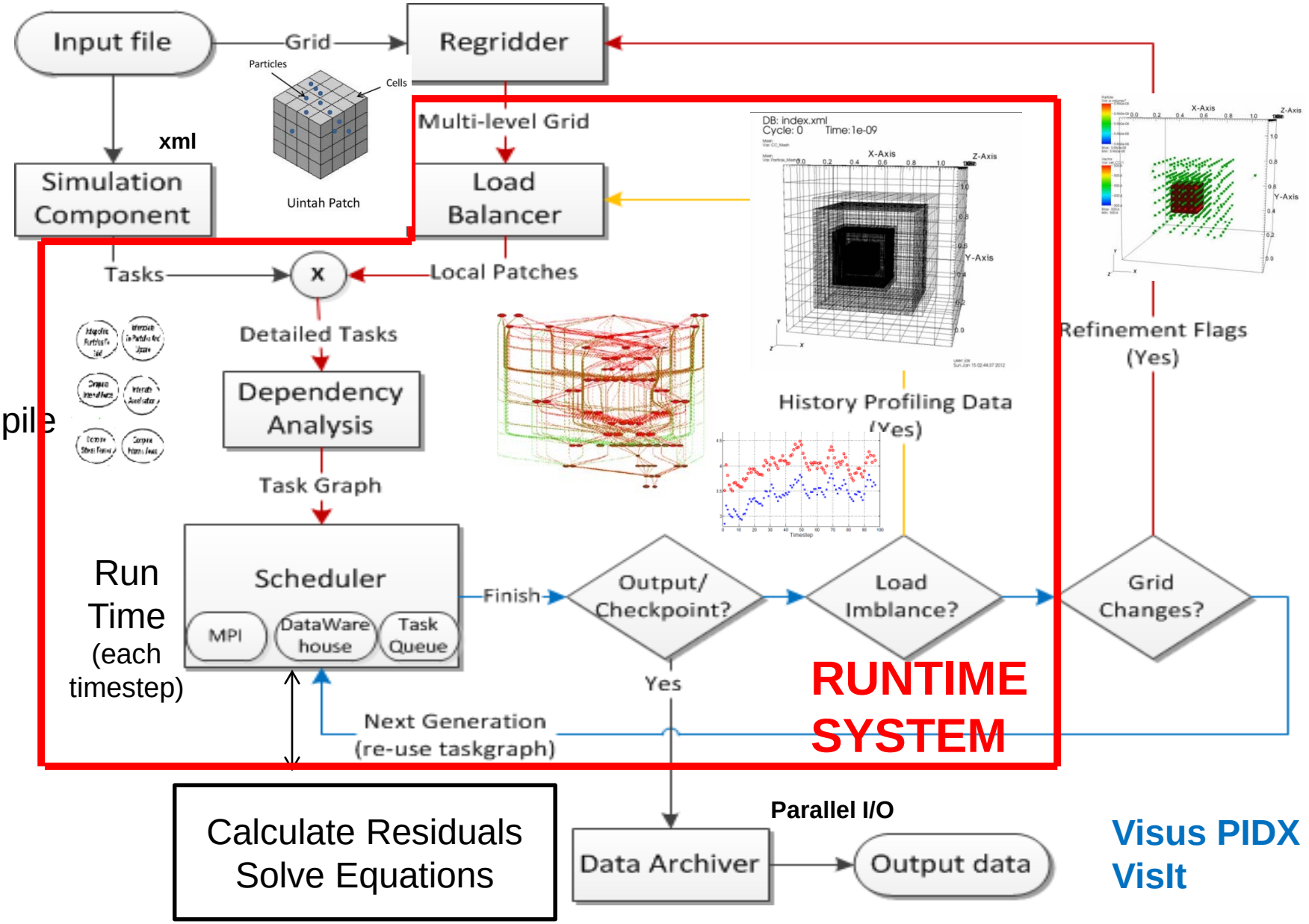
Uintah uses this information to create a task graph of computation (nodes) + communication (along edges)

Tasks do not explicitly define communications but only what inputs they need from a data warehouse and which tasks need to execute before each other.

Communication is overlapped with computation

Taskgraph is executed adaptively and sometimes out of order

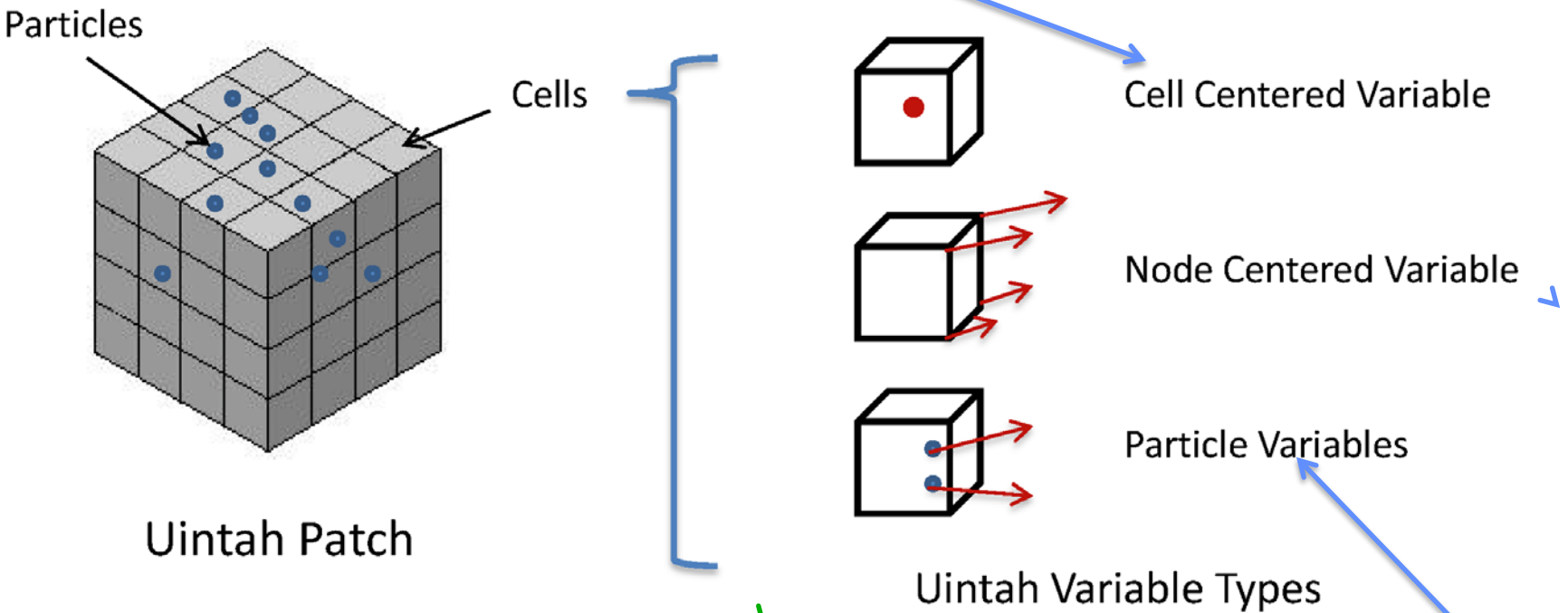
ARCHES or WASATCH/NEBO



UINTAH ARCHITECTURE

Uintah Patch and Variables

ICE is a cell-centered finite volume method for Navier Stokes equations



- Structured Grid Variable (for Flows) are Cell Centered Nodes, Face Centered Nodes.
- Unstructured Points (for Solids) are Particles

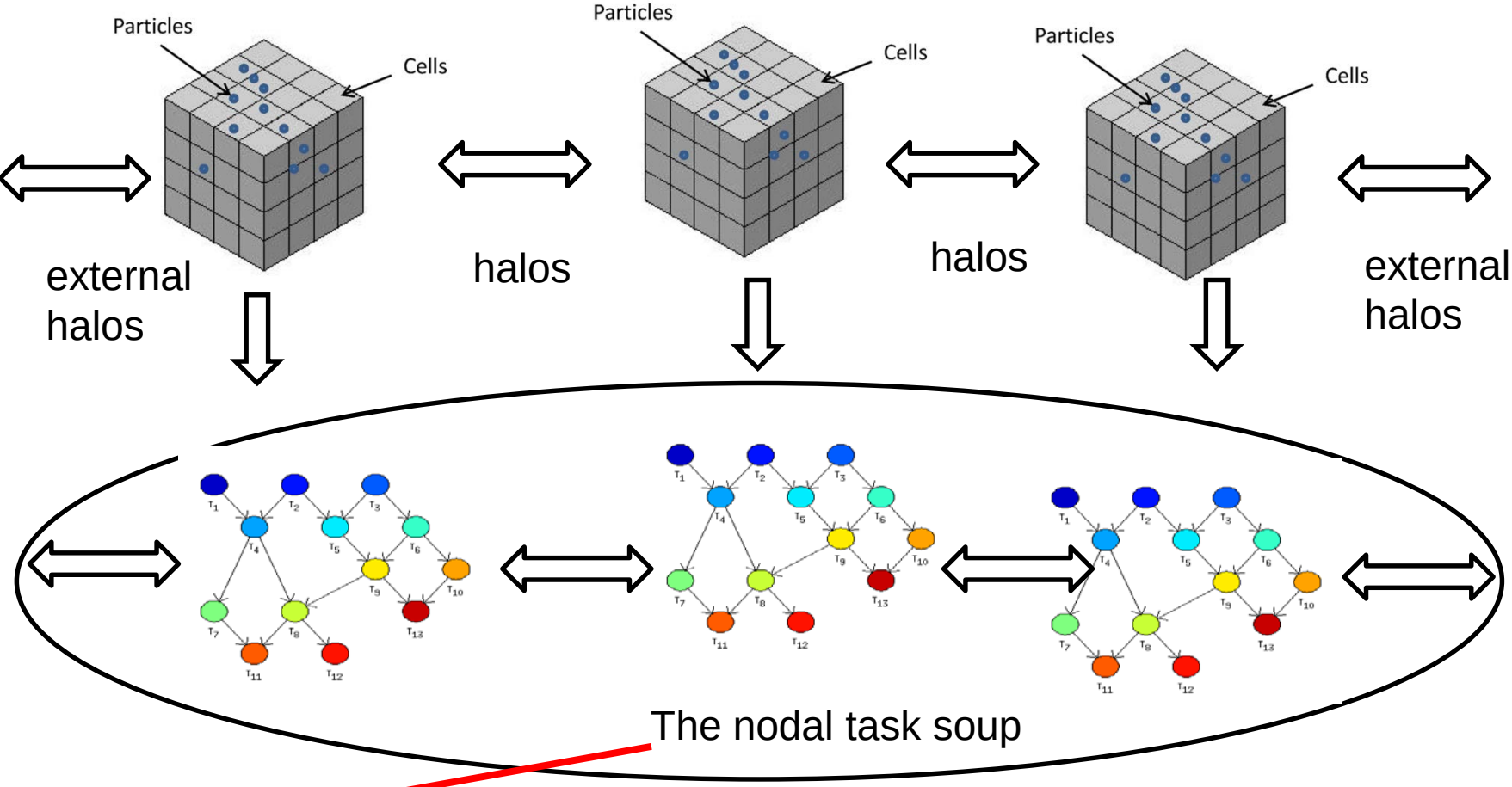
ARCHES is a combustion code using several different radiation models and linear solvers

MPM is a novel method that uses particles and nodes

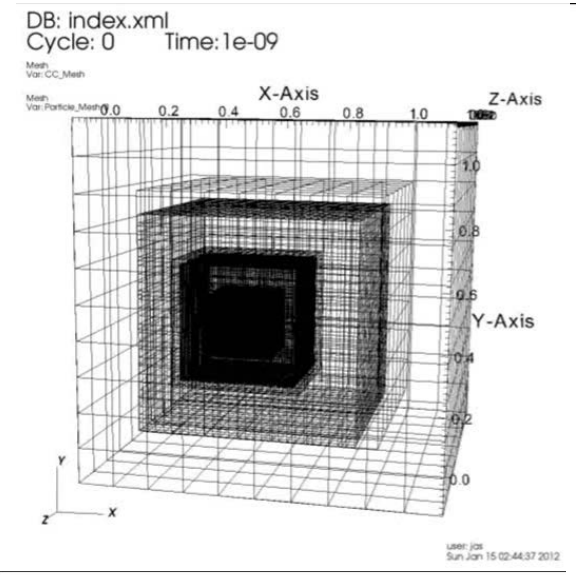
Exchange data with ICE, not just boundary condition

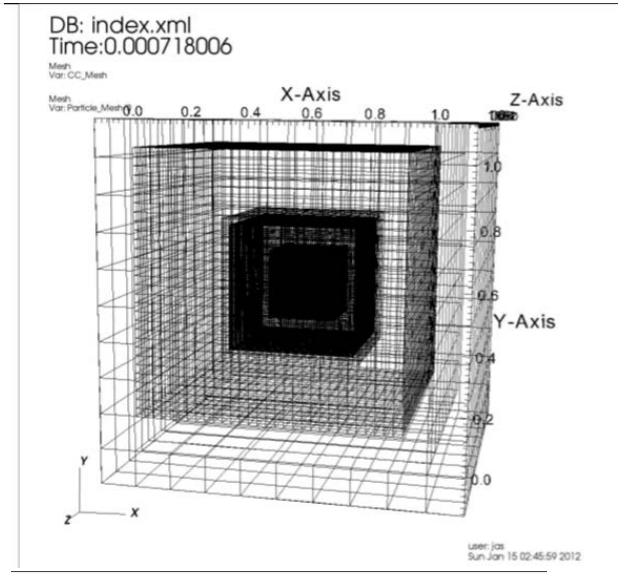
Uintah:MD based on Lucretius is a new molecular dynamics component

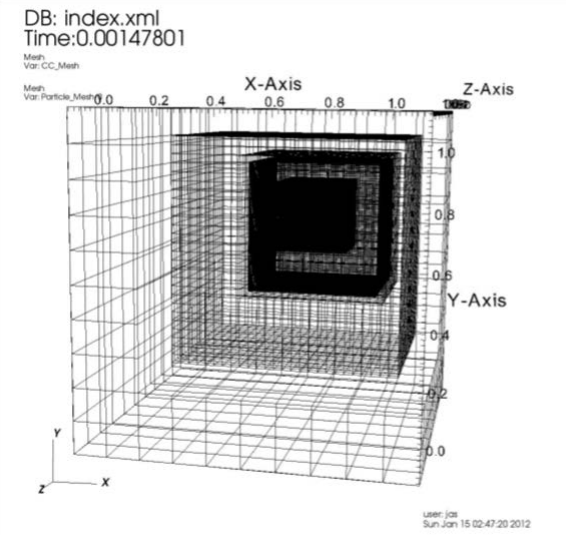
Task Graph Structure on a Multicore Node with multiple patches

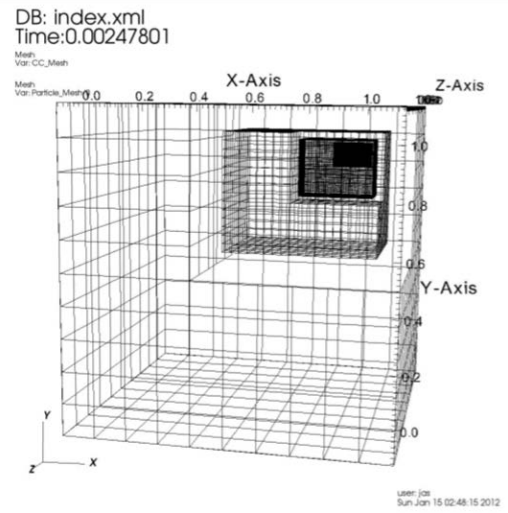


This is not a single graph. Multiscale and Multi-Physics merely add flavor to the “soup”.









Uintah's Adaptive Meshes

Structured Grid + Unstructured Points

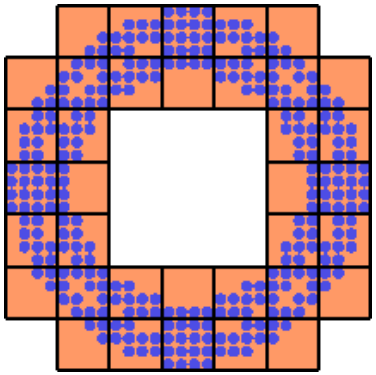
Patch-based Domain Decomposition

Adaptive Mesh Refinement

Dynamic Load Balancing

- **Profiling + Forecasting Model**
- **Parallel Space Filling Curves**

- **Works on MPI and/or thread level**
- **Scales to 768K cores**



The spatial mesh follows features of interest - in this case a moving container.

Burgers Example I

```
<Grid>
  <Level>
    <Box label = "1">
    <lower>      [0,0,0]       </lower>
    <upper>      [1.0,1.0,1.0] </upper>
    <resolution> [50,50,50]    </resolution>
    <patches>    [2,2,2]       </patches>
    <extraCells> [1,1,1]       </extraCells>
   </Box>
  </Level>
 </Grid>
```

25 cubed patches
8 patches
One level of halo elements

```
void  Burger::scheduleTimeAdvance( const LevelP& level,
                                   SchedulerP& sched)
{
  .....
   task->requires(Task::OldDW, u_label, Ghost::AroundNodes, 1);
  task->requires(Task::OldDW, sharedState_->get_delt_label());

  task->computes(u_label);
  sched->addTask(task, level->eachPatch(), sharedState_->allMaterials());
}
```

Get old solution from old data warehouse
One level of halos
Compute new solution

Burgers Equation code

$$U_t + UU_x = 0$$

```
void Burger::timeAdvance(const ProcessorGroup*,    const PatchSubset* patches,
 const MaterialSubset* matls,  DataWarehouse* old_dw, DataWarehouse* new_dw)
 //Loop for all patches on this processor
 { for(int p=0;p<patches->size();p++){
//Get data from  data warehouse including 1 layer of  "ghost" nodes from
    surrounding patches
    old_dw->get(u, lb_->u, matl, patch, Ghost::AroundNodes, 1);

   // dt, dx Time and space increments
   Vector dx = patch->getLevel()->dCell();
      old_dw->get(dt, sharedState_->get_delt_label());

  // allocate memory for results new_u
      new_dw->allocateAndPut(new_u, lb_->u, matl, patch);

 // define iterator range  l and h …… lots missing here  and Iterate through all the
    nodes
   for(NodeIterator iter(l, h);!iter.done(); iter++){
     IntVector n = *iter;
     double dudx = (u[n+IntVector(1,0,0)] - u[n-IntVector(1,0,0)]) /(2.0 * dx.x());
      double du = - u[n] * dt * (dudx);
     new_u[n]= u[n] + du;
   }
```

UINTAH SCALABILITY

Summary of Scalability Improvements

(i) Move to a one MPI process per multicore node reduces memory to less than 10% of previous for 100K+ cores

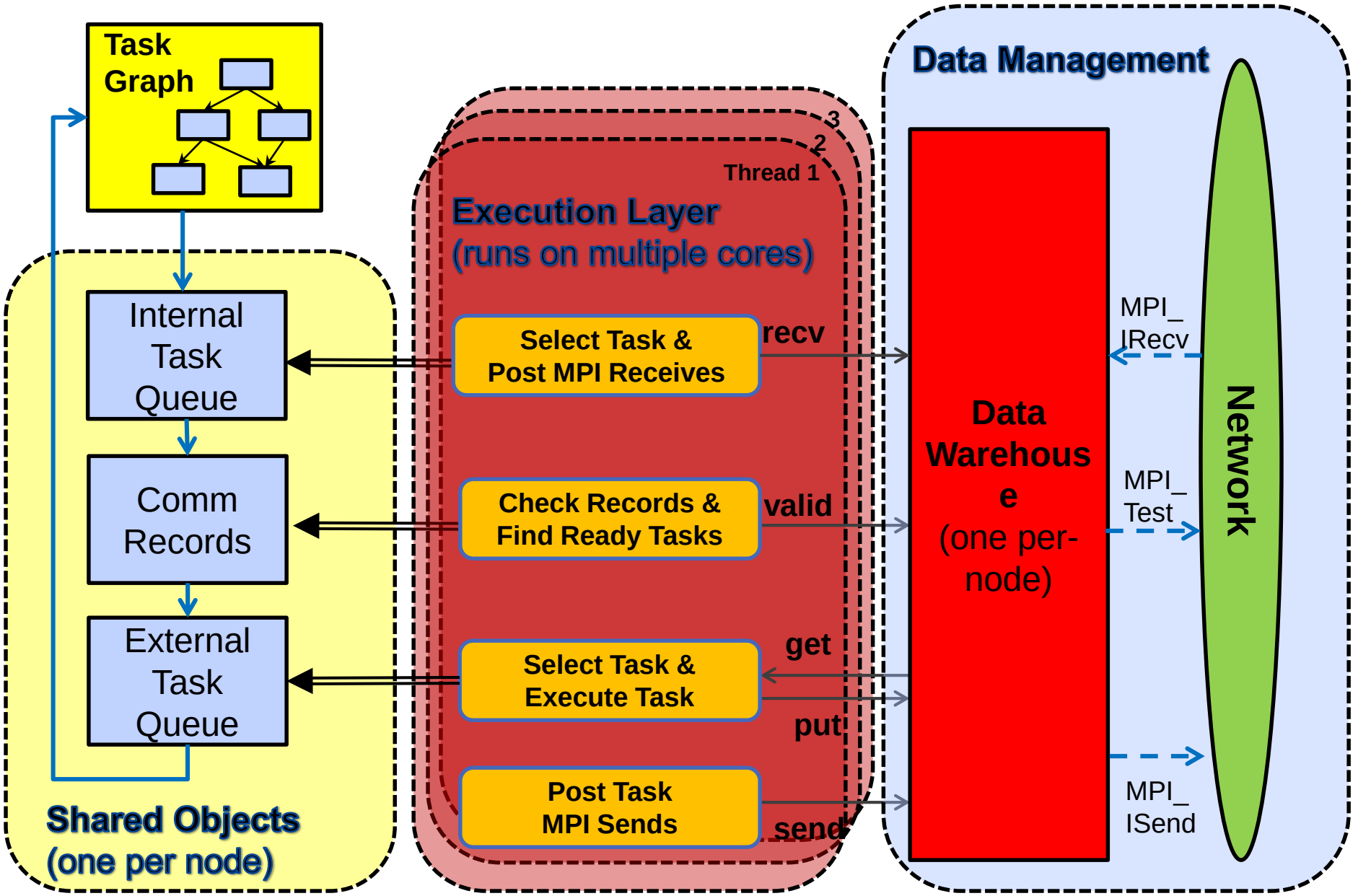
(ii) Use optimal size patches to balance overhead and granularity 16x16x 16 to 30x30x30.

(iii) Use only one data warehouse but allow all cores fast access to it, through the use of atomic operations.

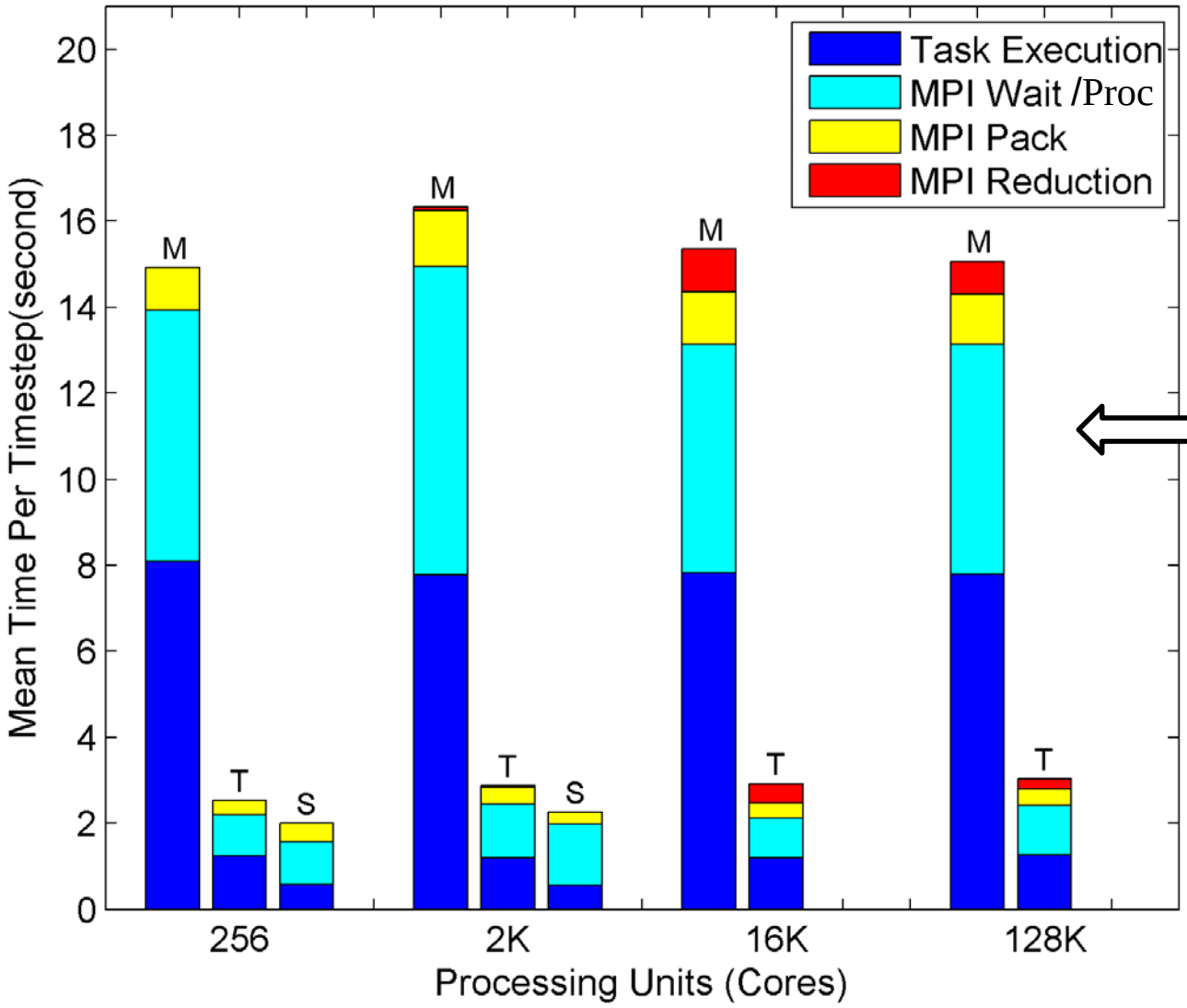
(iv) Prioritize tasks with the most external communications

(v) Use out-of-order execution when possible

Uintah Runtime System



Weak Scaling AMR+MPM ICE
M = Mira, T=Titan, S=Stampede



Only 2 patches per core Includes packing unpacking and data warehouse

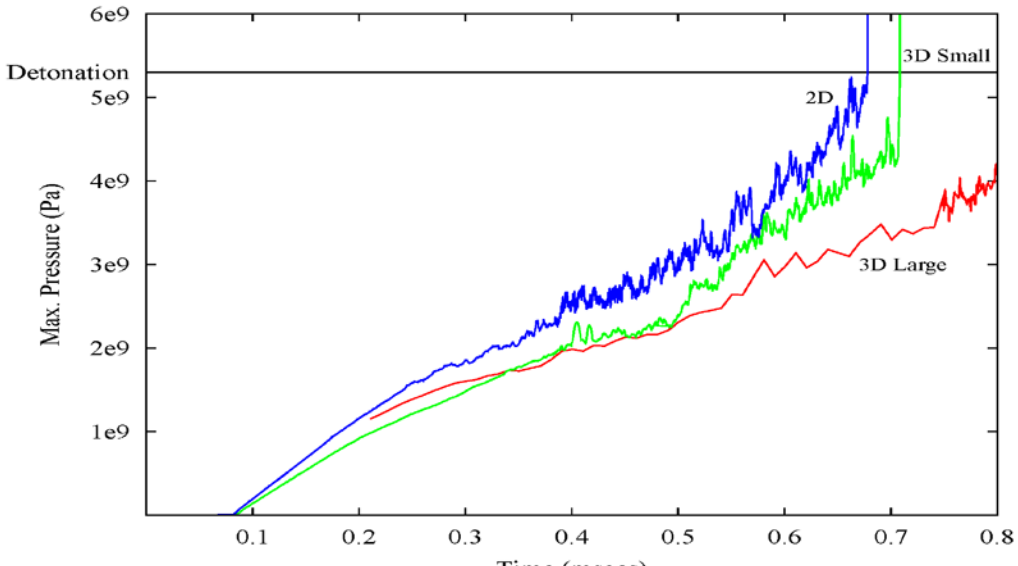
Only 8 interior patches from 32

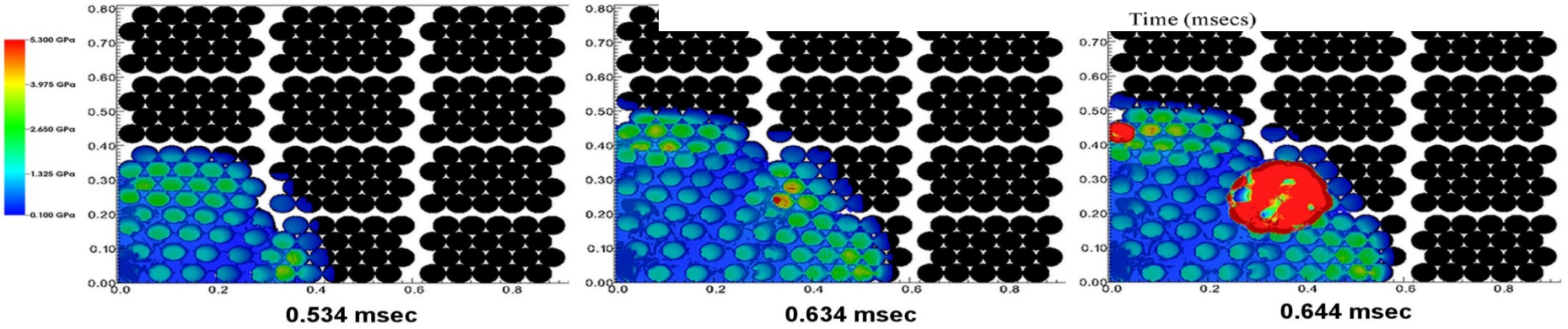
NSF funded modeling of Spanish Fork Accident 8/10/05

Speeding truck with 8000 explosive boosters each with 2.5-5.5 lbs of explosive overturned and caught fire

Experimental evidence for a transition from deflagration to detonation?

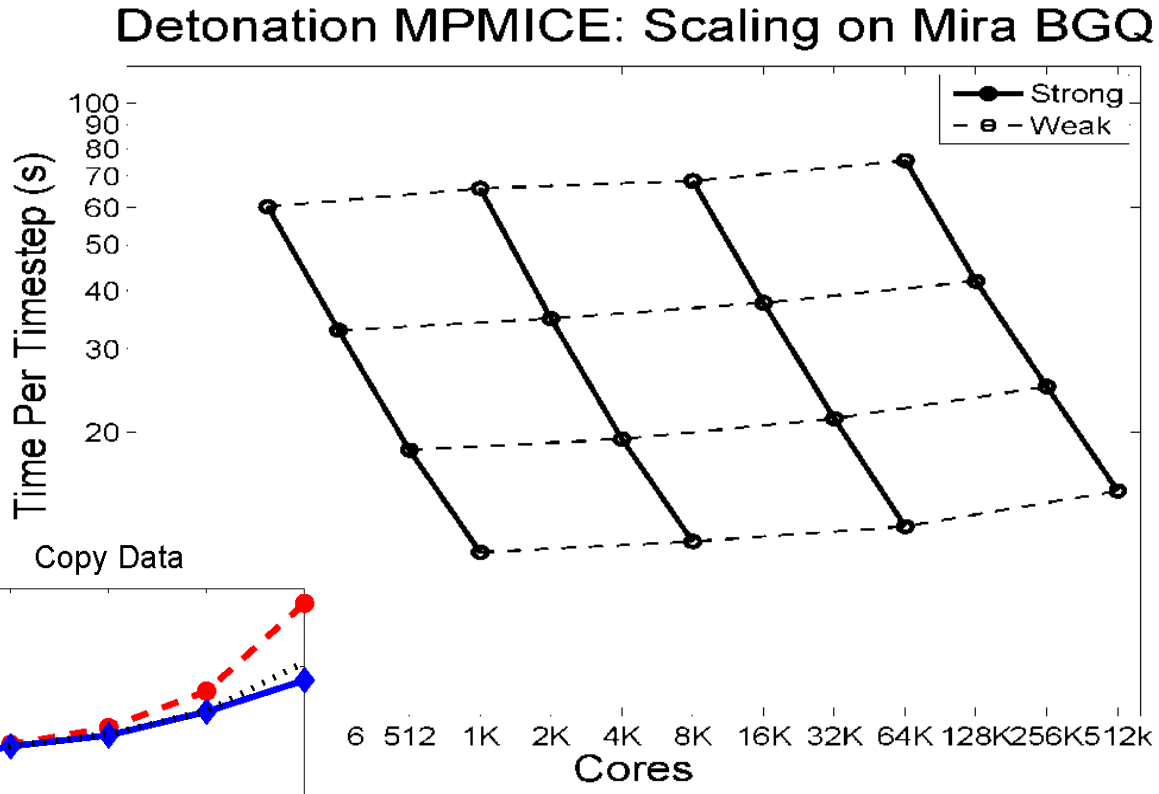
Deflagration wave moves at ~400m/s not all explosive consumed. Detonation wave moves 8500m/s all explosive consumed.

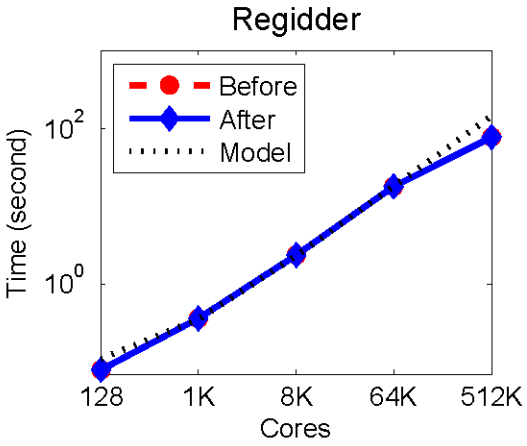


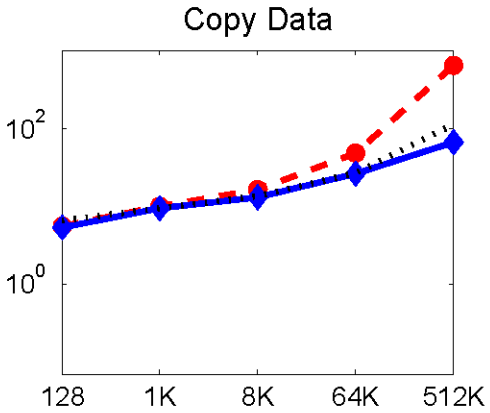


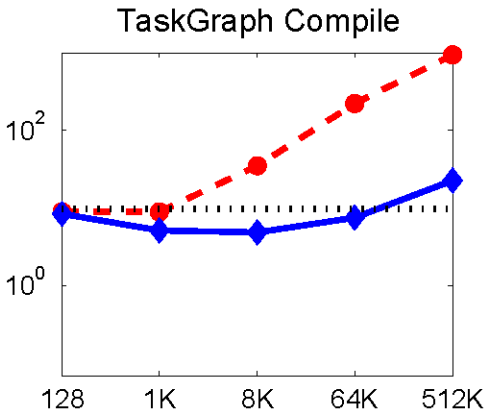
Spanish Fork Accident

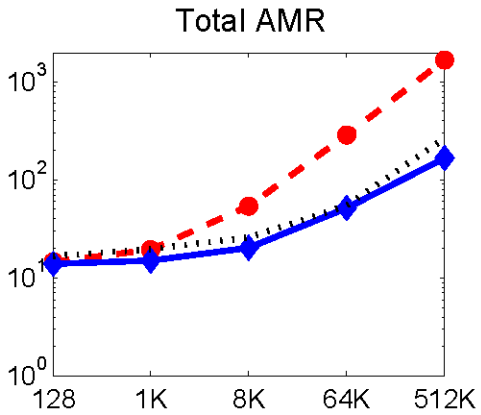
500K mesh patches
1.3 Billion mesh cells
7.8 Billion particles







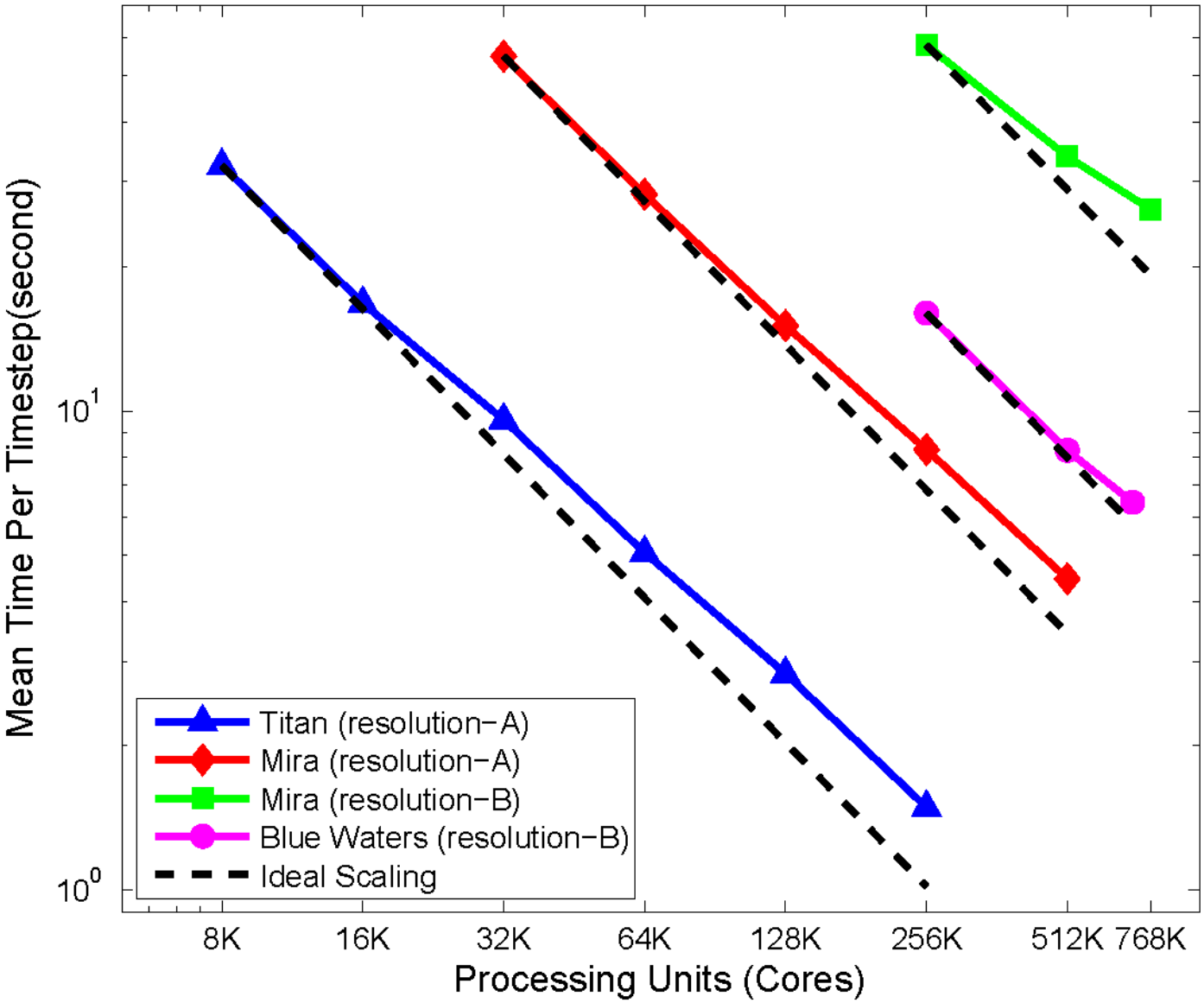




At every stage when we move to the next generation of problems Some of the algorithms and data structures need to be replaced .

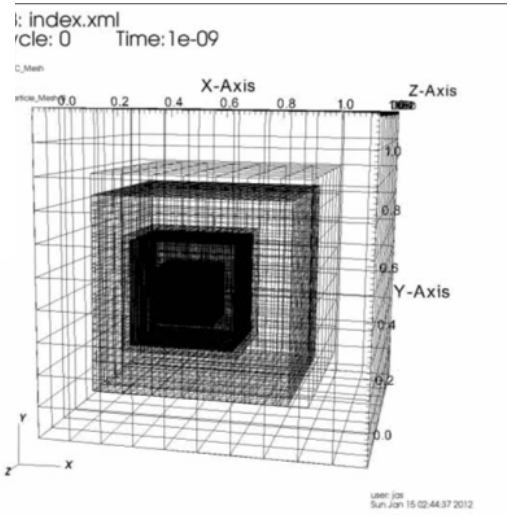
Scalability at one level is no certain Indicator fro problems or machines An order of magnitude larger

MPM AMR ICE Strong Scaling



Mira DOE BG/Q
768K cores
Blue Waters Cray XE6/XK7 700K+ cores

Resolution B
29 Billion particles
4 Billion mesh cells
1.2 Million mesh patches



Complex fluid-structure interaction problem with adaptive mesh refinement, see SC13/14 paper

NSF funding.

Solvers, EDSLs, Viz and Analysis

- Hypre Solver
- Nebo EDSL for Uintah (Sunderland Might Earl)
- Fast efficient visualization tools for Uintah (Pascucci and Hansen)

Linear Solves arises from Navier –Stokes Equations

$$\frac{\partial \rho}{\partial t} + \nabla . \rho u = 0,$$

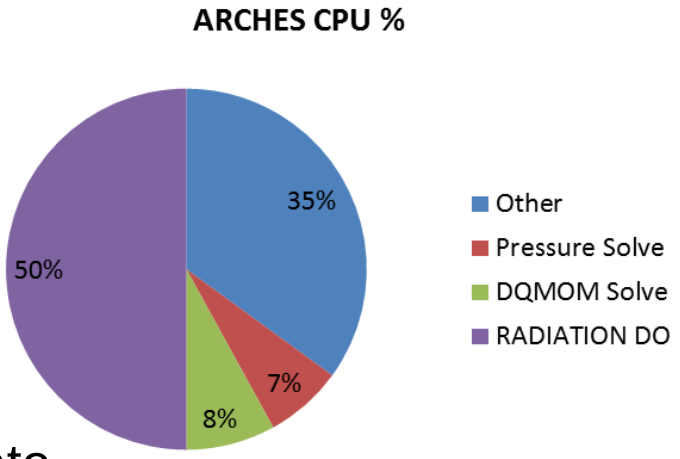
Full model includes turbulence, chemical reactions and radiation

where ρ is density, u is velocity vector and p is pressure

$$\frac{\partial \rho u}{\partial t} = F - \nabla p, \quad \text{where} \quad F = -\nabla . \rho uu + \nu \nabla^2 u + \rho g$$

Arrive at pressure Poisson equation to solve for *p*

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



Use Hype Solver distributed by LLNL
Many linear solvers inc. Preconditioned Conjugate Gradients on regular mesh patches used
Multi-grid pre-conditioner used
Careful adaptive strategies needed to get scalability
CCGrid13 paper.

One radiation solve per timestep

Linear Solves arises from Navier –Stokes Equations

$$\frac{\partial \rho}{\partial t} + \nabla . \rho u = 0,$$

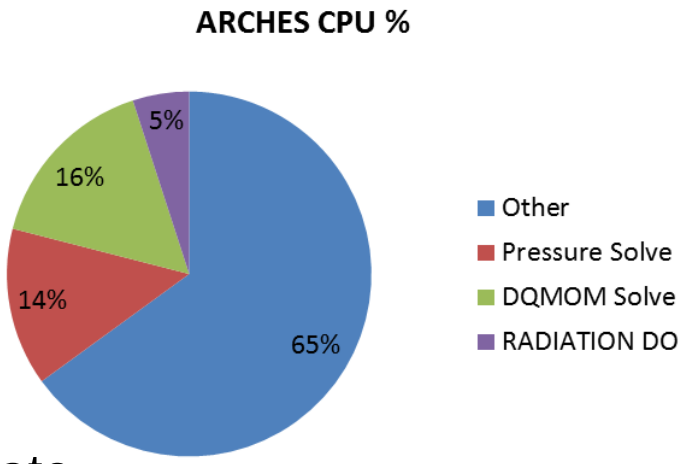
Full model includes turbulence, chemical reactions and radiation

where ρ is density, u is velocity vector and p is pressure

$$\frac{\partial \rho u}{\partial t} = F - \nabla p, \quad \text{where} \quad F = -\nabla . \rho uu + \nu \nabla^2 u + \rho g$$

Arrive at pressure Poisson equation to solve for *p*

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



Use Hypre Solver distributed by LLNL
Many linear solvers inc. Preconditioned Conjugate Gradients on regular mesh patches used
Multi-grid pre-conditioner used
Careful adaptive strategies needed to get scalability
CCGrid13 paper. **http://www.llnl.gov/CASC/hypre/**

One radiation solve Every 10 timesteps

NEBO/Wasatch Example

- **Express complex pde functions as DAG** - automatically construct algorithms from expressions
- **Define field operations** needed to execute tasks (fine grained vector parallelism on the mesh)
- **User writes only field operations code** . Supports field & stencil operations directly - no more loops!
- **Strongly typed fields** ensure valid operations at compile time. *Allows a variety of implementations to be tried without modifying application code*.
- **Scalability on a node** - use **Uintah** infrastructure to get scalability across whole system

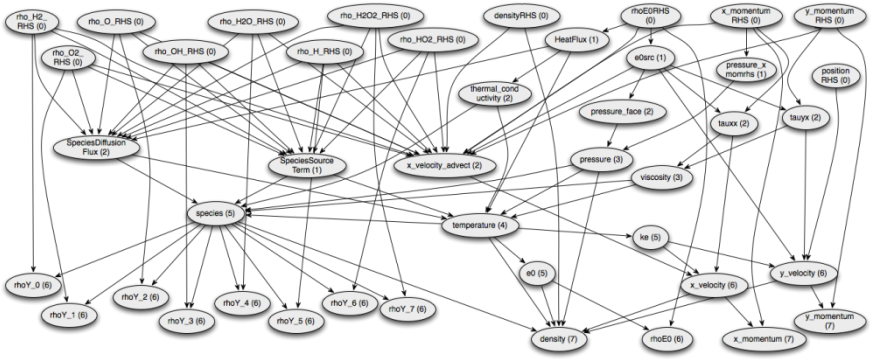
Energy equation

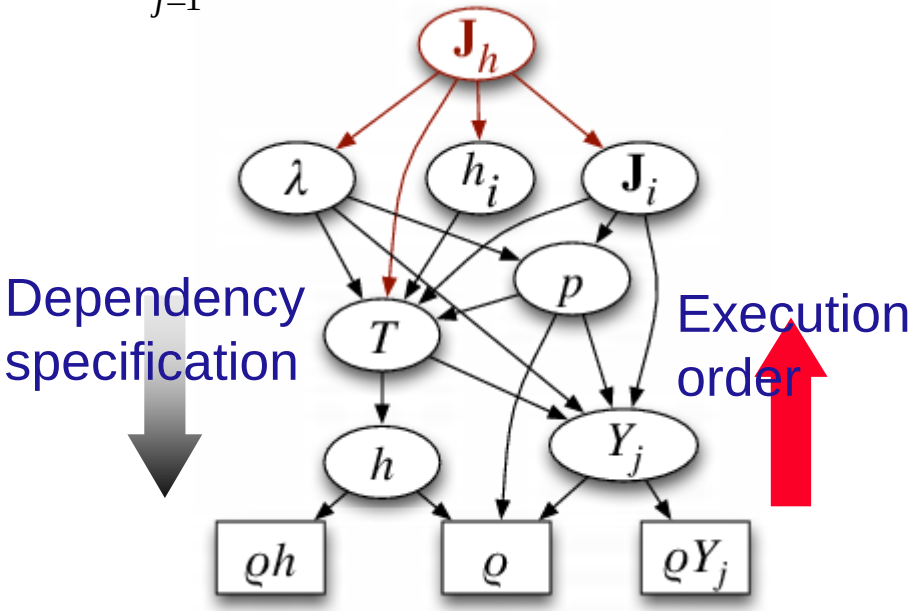
$$\frac{\partial \rho e}{\partial t} + \nabla.(\rho e \underline{u}) + \nabla.\underline{J}_h + terms = 0$$

Enthalpy diffusive flux

$$\underline{J}_h = -\lambda(T, Y_j)\nabla T - \sum_{i=1}^{n} h_i \underline{J}_i$$

$$\underline{J}_i = -\sum_{j=1}^{ns} D_{ij}(T, Y_j)\nabla Y_j - D_i^T(T, Y_j)\nabla T$$





[Sutherland Earl Might]

Multicore & GPU Performance

$$\nabla \cdot (\lambda \nabla T)$$

```
phi <<= divX( -interpX(lambda) * gradX(T) )
      + divY( -interpY(lambda) * gradY(T) )
      + divZ( -interpZ(lambda) * gradZ(T) );
```

	2	4	6	8	10	12	GPU
64x64x64	1.3	2.4	2.6	3.3	3.3	3.3	13.8
128x128x128	1.9	3.9	4.9	6.8	7.8	6.0	26.0

$$-\nabla \cdot (\rho \phi \mathbf{u} + \mathbf{J}_\phi)$$

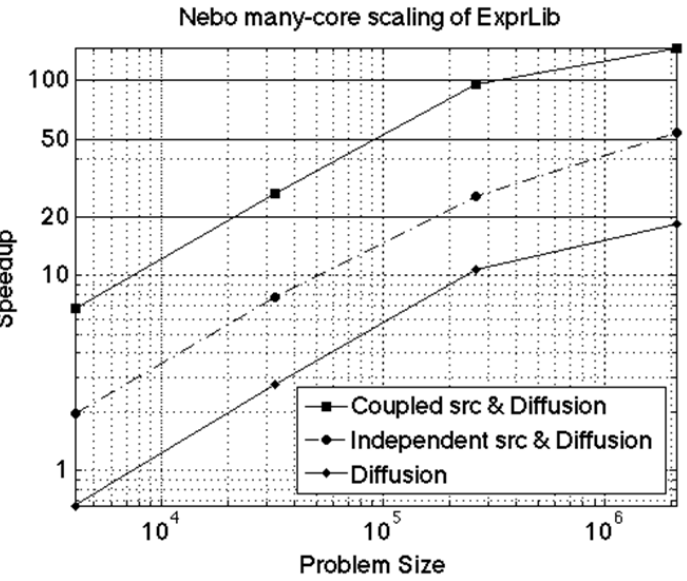
```
rhs <<= -divOpX_( xConvFlux_ + xDiffFlux_ )
        -divOpY_( yConvFlux_ + yDiffFlux_ )
        -divOpZ_( zConvFlux_ + zDiffFlux_ );
```

	2	4	6	8	10	12	GPU
64x64x64	1.8	2.9	2.9	3.4	3.6	3.6	16.3
128x128x128	2.0	3.6	5.0	6.5	6.1	4.8	13.5

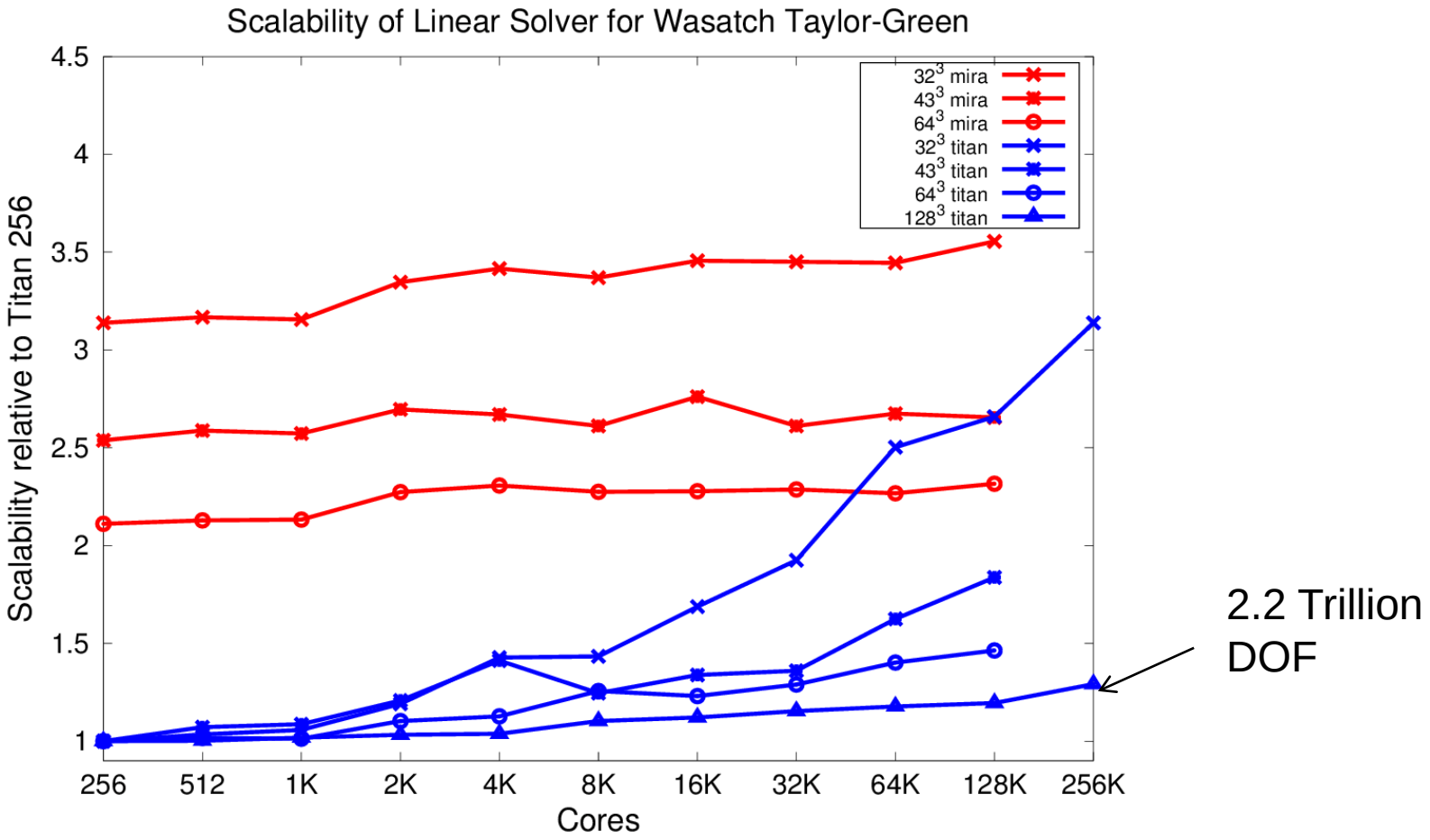
- One inlined grid loop, no temporaries.
- Better parallel performance than without chaining.
- Compile-time consistency checking (field-operator and field-field compatibility).

Wasatch – Nebo Recent Milestones

- Wasatch is solving (nonreacting miniboiler~3-4x speedup over the non-DSL approach.
- New Nebo backend for CPU resultied in 20-30% speedup in the entire Wasatch code base.
- Much of the Wasatch code base is GPU-ready
- Arches plus SpatialOps & Nebo EDSL being scoped.



Good GPU scaling with (>32^3 per patch).
Loop fusion (heavy GPU kernels) needed e.g “coupled source & diffusion”



Each **Mira Run** is scaled wrt the **Titan Run at 256 cores**
Note these times are not the same for different patch sizes.

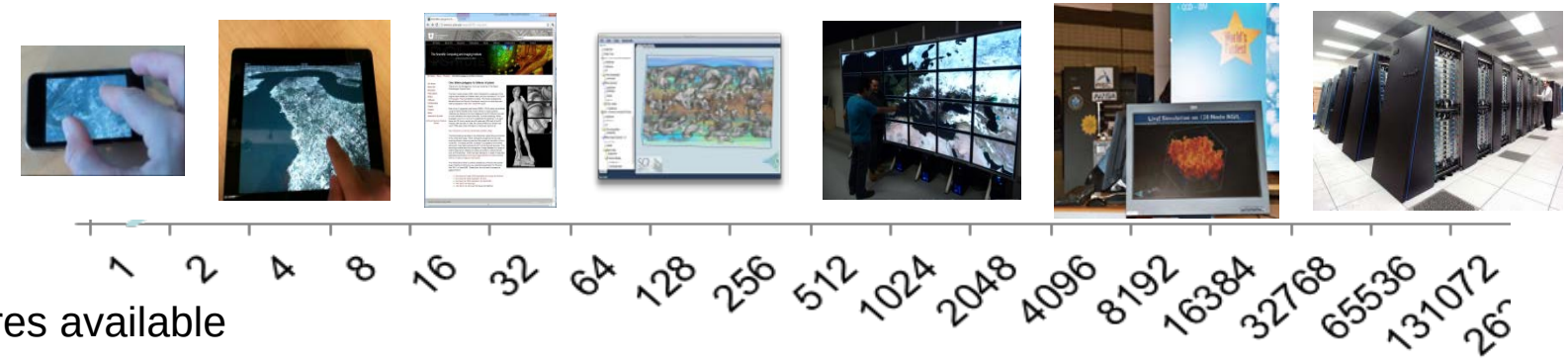
Weak Scalability of Hypre Code

We build our data management solution on most advanced technology in big data streaming analytics and visualization

Live demonstration at SC12 & 13:

- ~4TB per time step (100s of PF 3D timesteps generated on Intrepid)
- Steaming live from ANL visualization cluster
- Interactive, immersive, analysis and visualization

Infrastructure that scales gracefully with available hardware resources

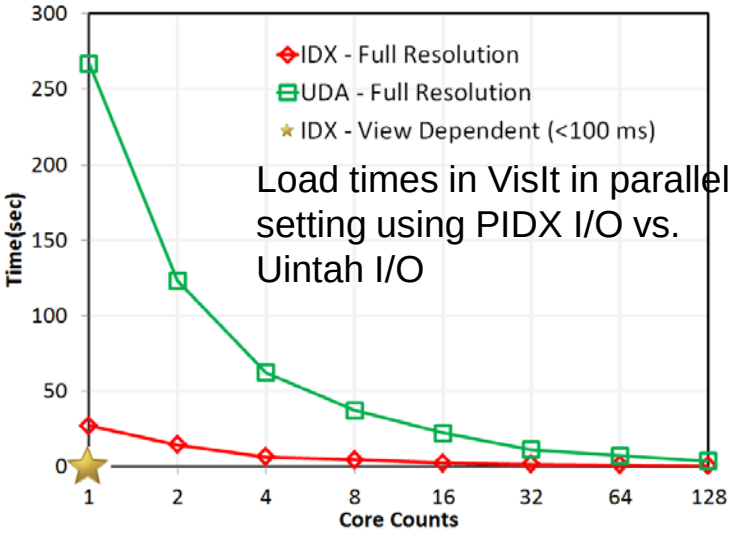


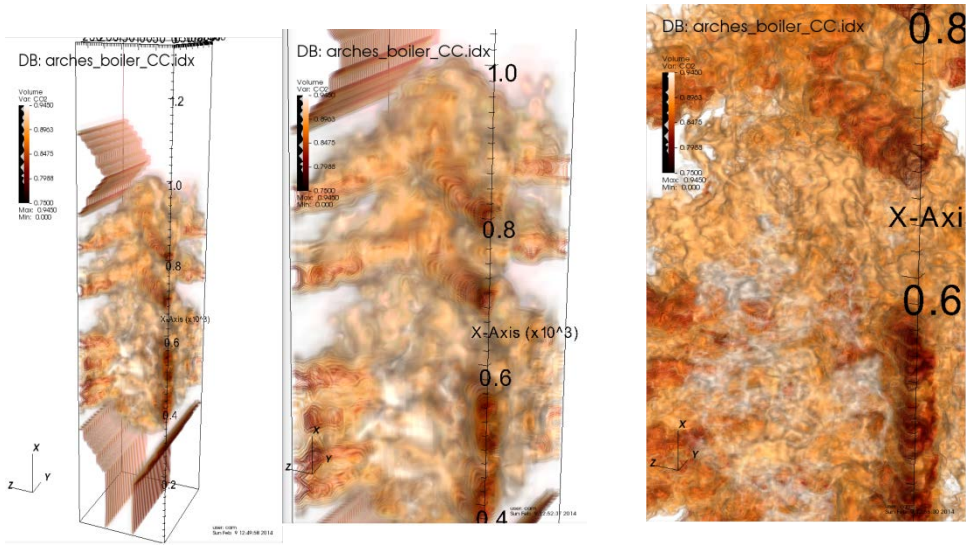
Cores available

Uintah+PIDX Integration

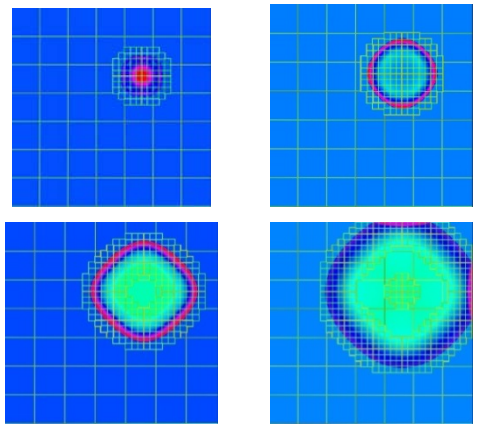
Uintah I/O One data file per patch & 1 Metadata file per patch. **Non-scalable I/O and visualization**

PIDX I/O:Multi-resolution, cache oblivious data format – IDX format. Real time interactive viz of simulation data High performance I/O – SC'13 View Dependent viz. using VisIt (ISC '14)

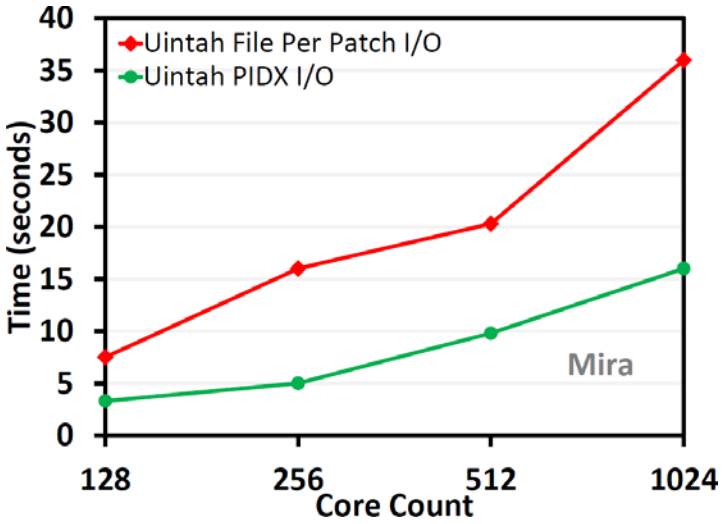




View Dependent Viz of Uintah BSF Data in IDX using VisIt



- Progression of AMR Uintah simulation for a 2-level Blast Wave
- AMR data written out using PIDX and visualized with ViSUS

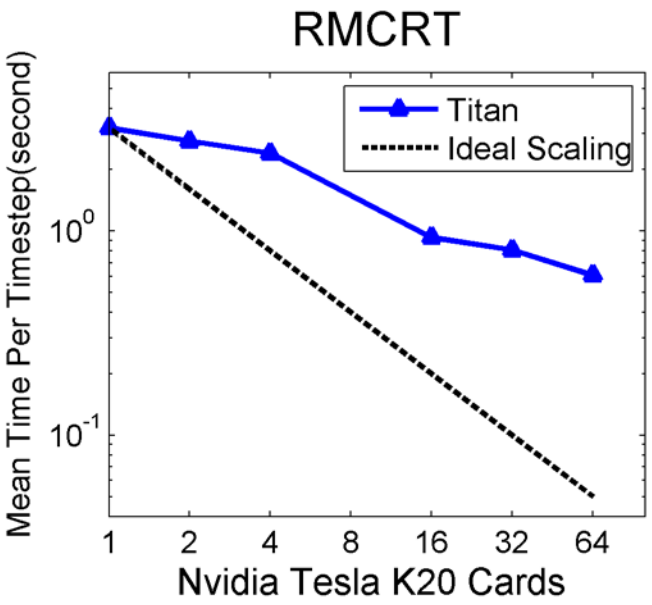


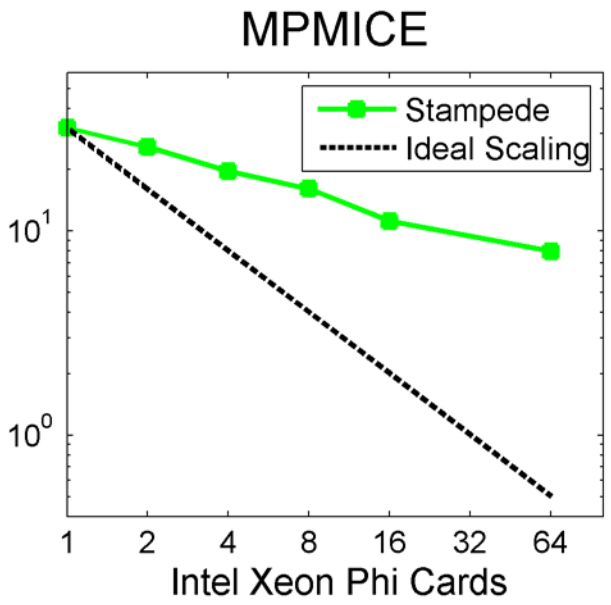
DESIGNING FOR EXASCALE

Clear trend towards accelerators e.g. GPU but also Intel MIC – new NSF "Stampede" 10-. 15PF Balance factor = flops/bandwidth - high

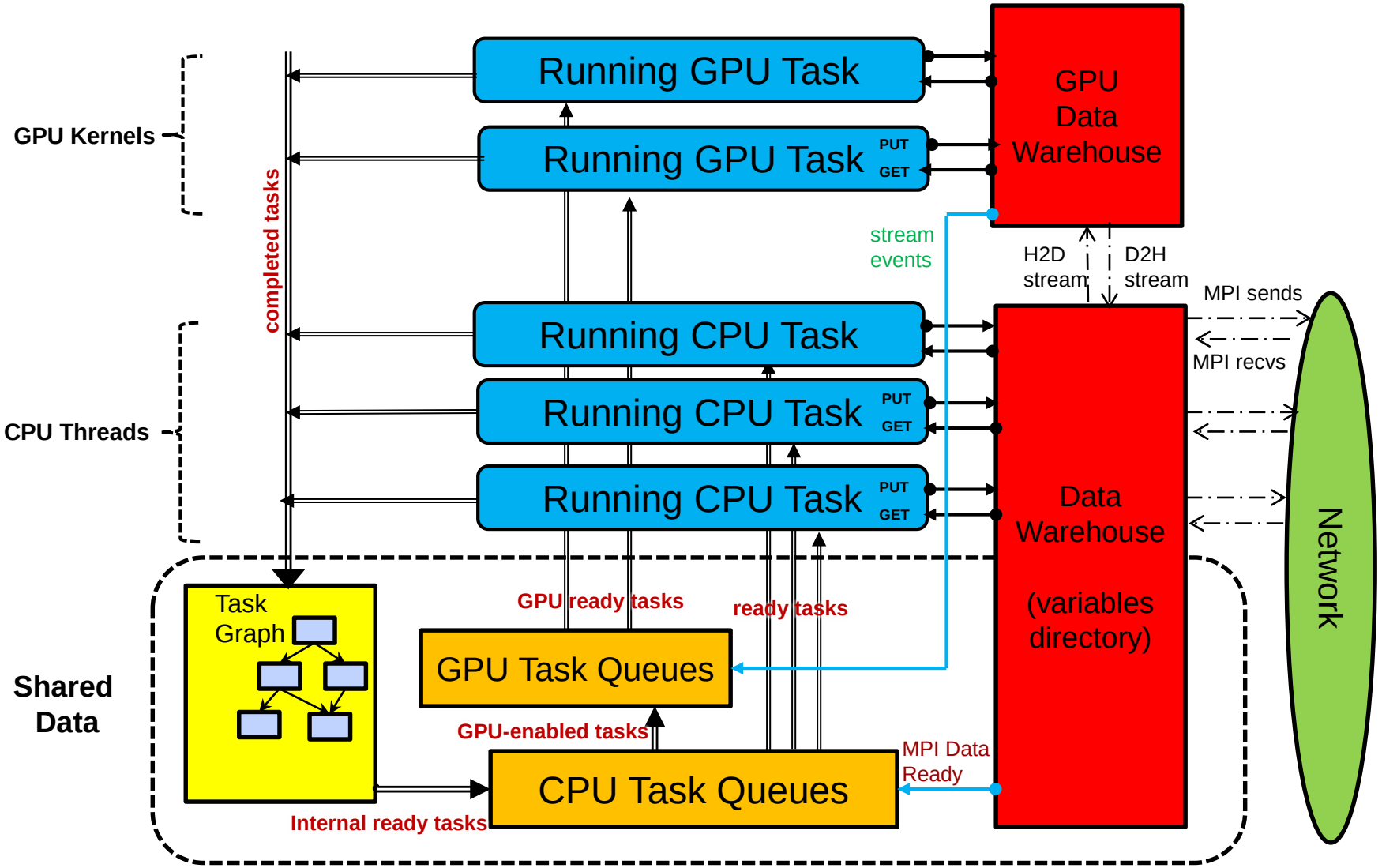
GPU performance "ok" for stencil-based codes ,2x over multicore cpu estimated and achieved for ICE . Similar results by others.
Network and memory performance more slowly growing than cpu/gpu performance. GPU perf.of ray-tracing radiation method is 100x cpu

Overlapping and hiding Communications essential





Unified Heterogeneous Scheduler & Runtime node



No MPI inside node, lock free DW , cores and GPUs pull work

NVIDIA AMGX Linear Solvers on GPUs

- Fast, scalable iterative gpu linear solvers for packages e.g.,
- Flexible toolkit provides GPU accelerated Ax = b solver
- Simple API for multiple apps domains.
- Multiple GPUs (maybe thousands) with scaling

Key Features

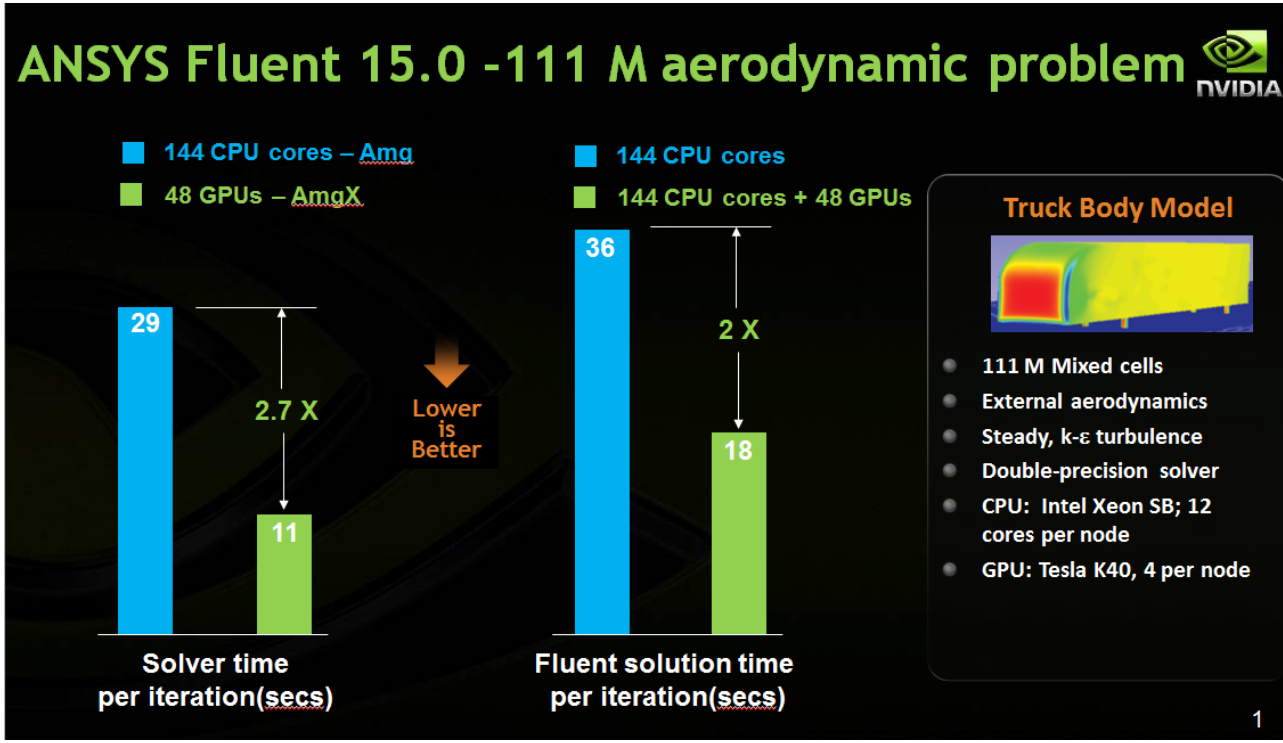
Ruge-Steuben algebraic MG

Krylov methods: CG, GMRES, BiCGStab,

Smoothers and Solvers: Block- Jacobi, Gauss-Seidel, incomplete LU,

Flexible composition system

MPI support OpenMP support, Flexible and high level C API,



Free for non-commercial use

Utah access via Utah CUDA COE.

Resilience

- Need interfaces at system level to help us consider:
- Core failure – reroute tasks
- Comms failure – reroute message
- Node failure – need to replicate patches use an AMR type approach in which a coarse patch is on another node. In 3D has 12.5% overhead – suggested by Qingyu Meng Mike Heroux and others.
- Will explore this from fall 2014 onwards

Summary

- DAG abstraction important for achieving scaling
- Layered approach very important for not needing to change applications code
- Scalability still requires much engineering of the runtime system.
- General approach very powerful indeed.
- Obvious applicability to new architectures
- DSL approach very important in future-proofing
- Scalability still a challenge even with DAG approach – which does work amazingly well
- GPU development ongoing
- The approach used here shows promise for very large core and GPU counts but using these architectures is an exciting challenge e.g. new Knights Landing NERSC8 machine