Steve Parker, Jim Guilkey, Chuck Wight, Justin Luitjens, Todd Harman, Dav de St Germain, Dave Pershing, Chris Johnson + rest of CSAFE Group at Utah

Martin Berzins

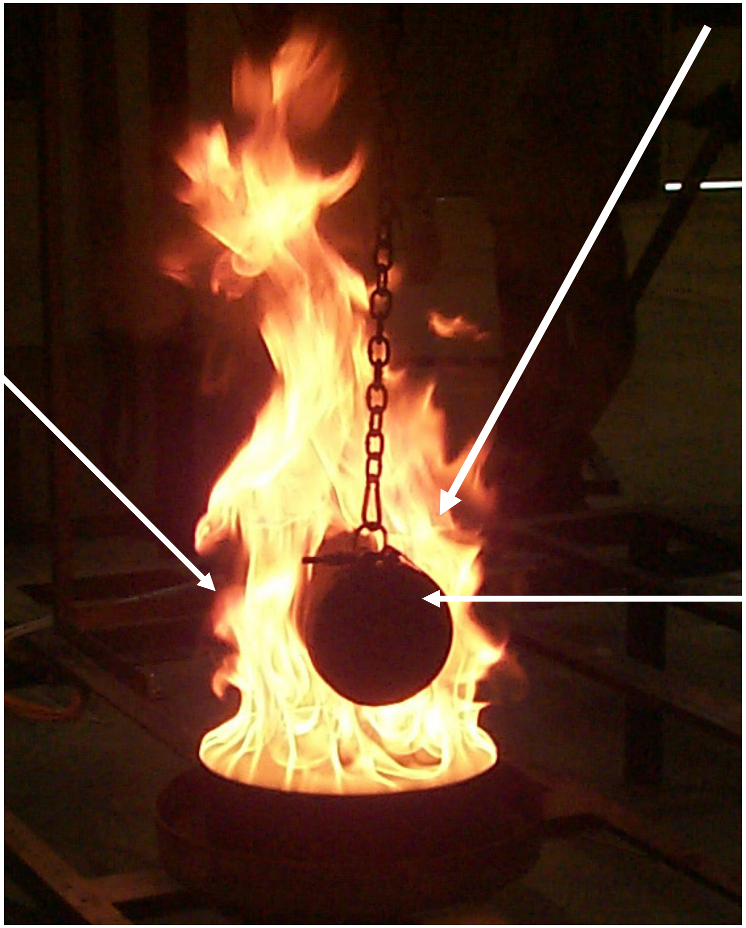
Multi-scale Simulations of Fires and Explosions (and other things)

Adaptive Mesh Refinement/ Particle Method Challenges for Parallel Computing

Thanks to DOE for funding from 1997-2008

University of Utah CSAFE DoE Project

When is the explosion?
How strong is it ?



Hydrocarbon Fire

-turbulent combustion

- complex kinetics

- soot formation

- wind allowed

- wide range of scales

Metal Container

-heat transfer from fire

- fragmentation

HE Material PBX

- surface burning

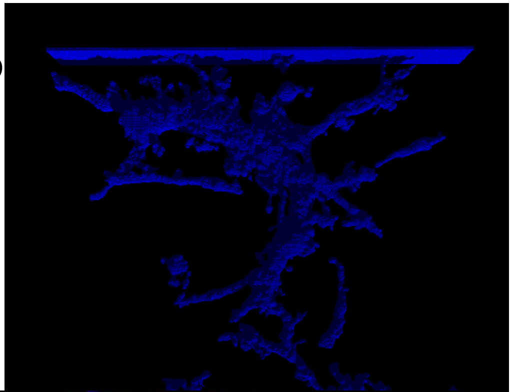
- microscopic crack formation

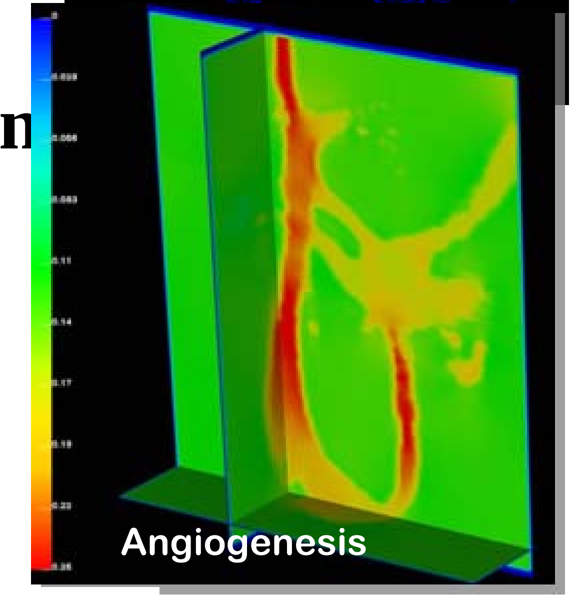
Single mesh with Navier Stokes eqns for multiphase-fluid-structure interaction problems. Material particles MPM move in a fixed grid.

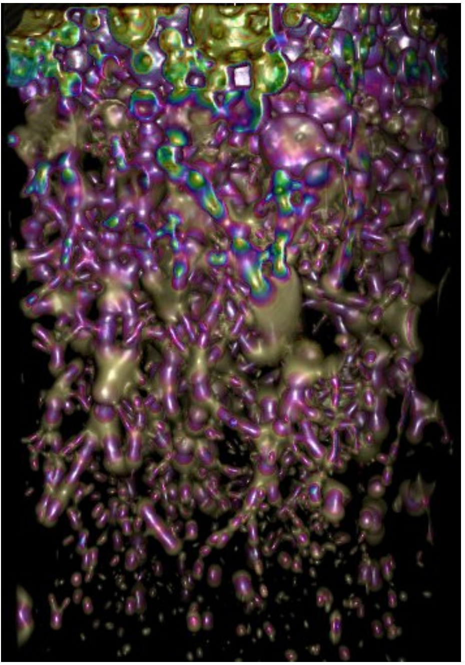
Automated parallelization including MPM and AMR

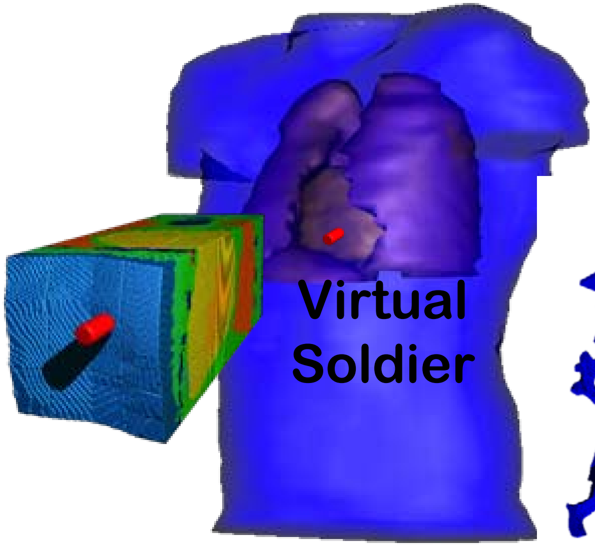
Uintah Applications

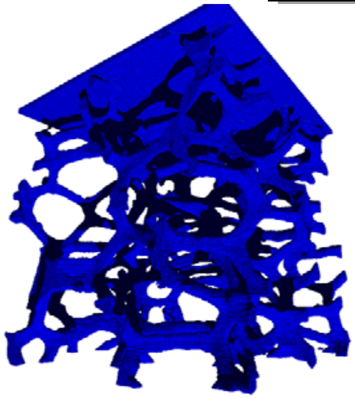
- Flare Simulation
- Angiogenesis
- Vocal modeling
- Rocket stage separation
- Bullet-torso impact
- Foam properties

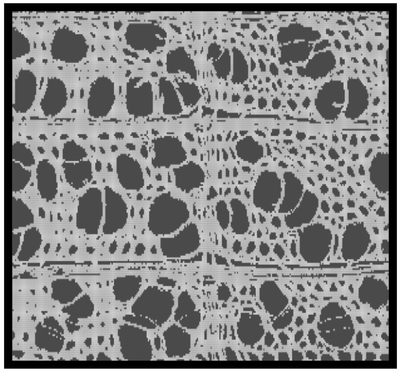










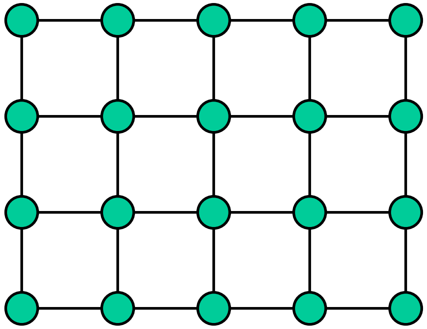


Fundamental Uintah data Structure is a patch – multiple variable types

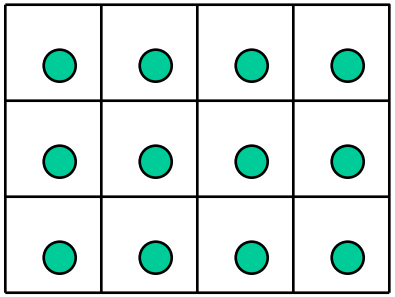
Load balancing uses patches

User writes code for a patch and its communications only - Uintah uses this information to construct communications pattern via a task graph

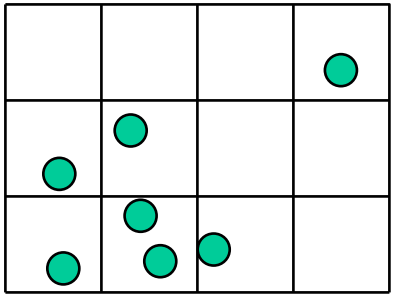
Cell –Vertex Variables

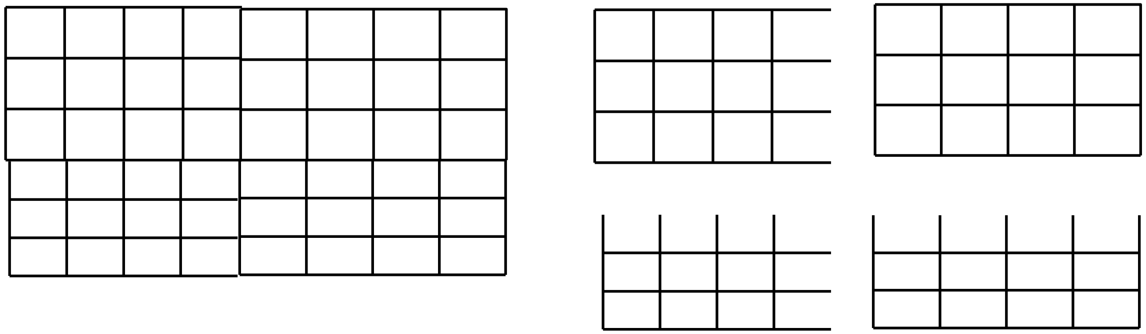


Cell Centered Variables



Particle Variables





Uintah Domain Decomposition

Scheduler component

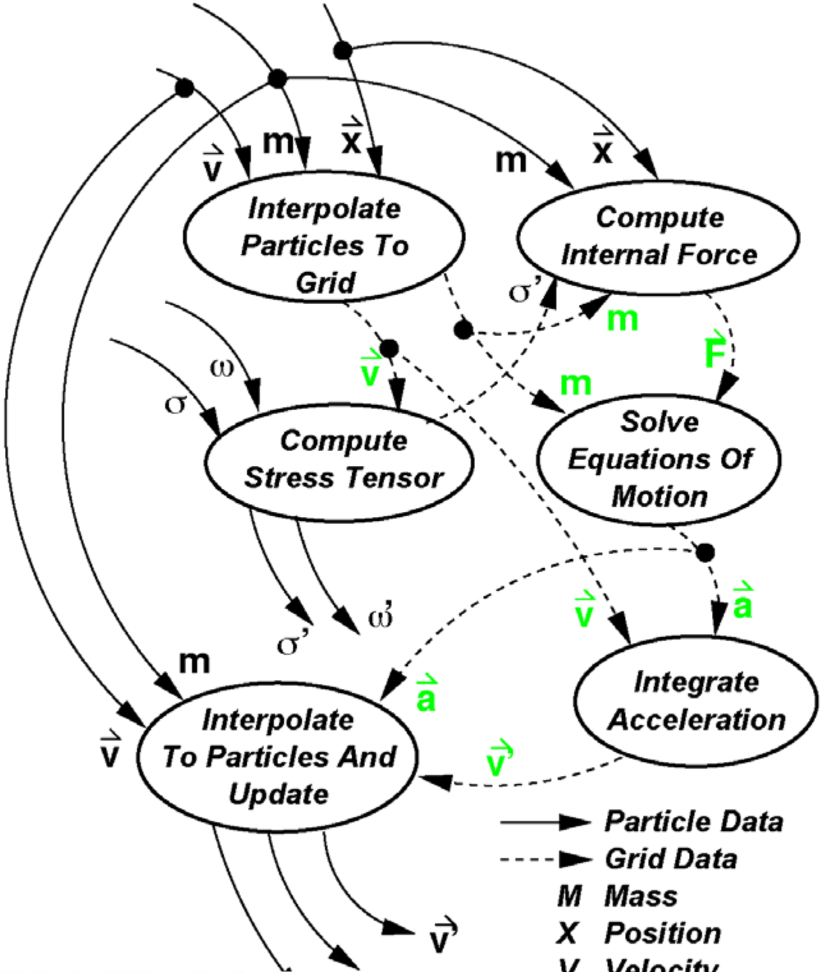
- decides which task will be executed by which processor
- is guided by cost models for computation and communication
- can be very simple or very complex
- can be static (MPI), dynamic (threads), or perhaps a mixture
- is encapsulated and composable

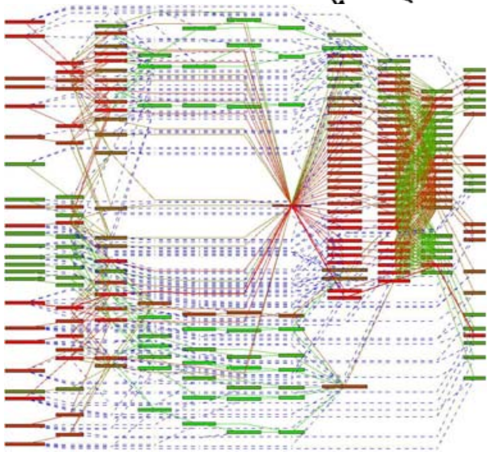
Scheduler and Load balancer are the key components for portable scalability. One size does not fit all.

Challenge is dealing with mapping onto multi-cores and increasingly novel architectures at a purely system level

Task graph

- Each algorithm (MPM, CFD, Fire, I/O, etc.) defines a description of the computation
  - Required inputs and outputs (names and spatial relationships)
  - Callbacks to perform each task on a single subregion of space (C++ or Fortran)
- Communication is performed at the edges in the graph
- Uintah uses this information to create a graph of computation and communication



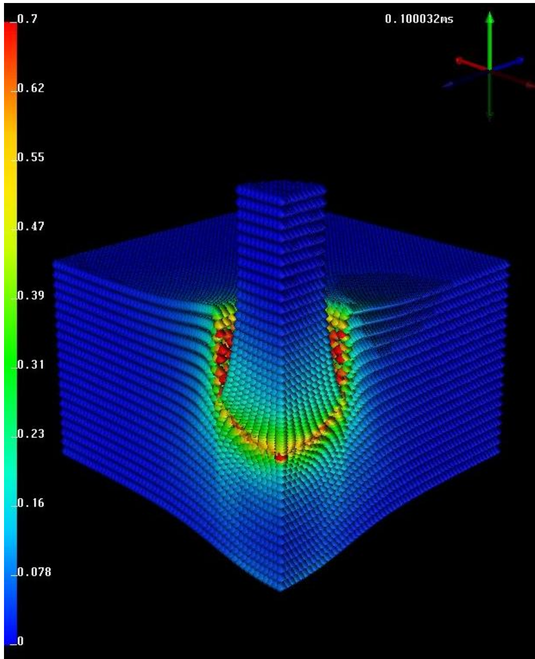


Task graph (dis)advantages

- Enables complex physics in a flexible manner
- Expresses complex communication patterns that arise from meshes/work changing dynamically
- Enables scalability
- Allows components (including the scheduler) to evolve independently

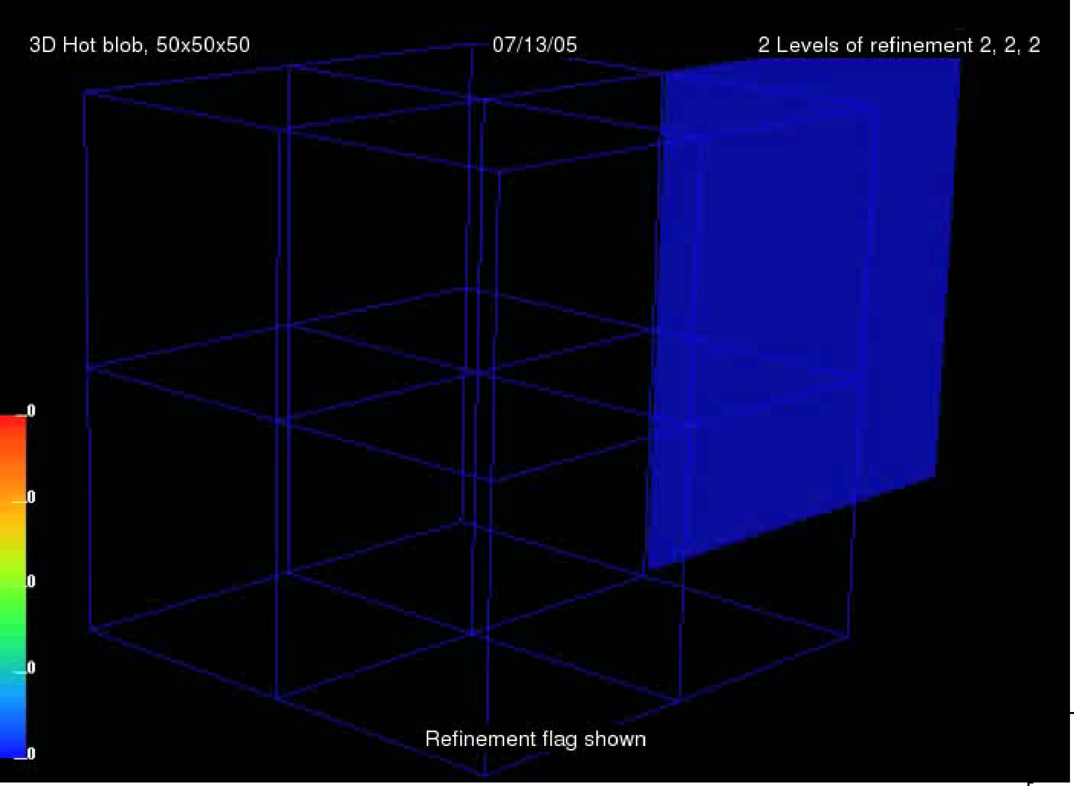
BUT

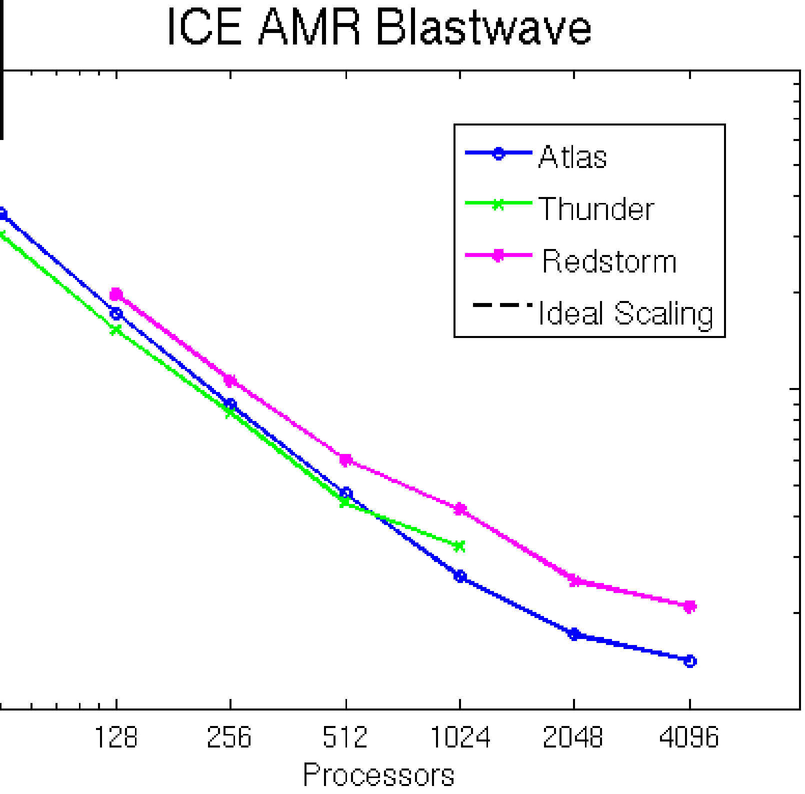
- Optimal scheduling is NP-hard
  - However, "optimal enough" isn't too hard
- Creation of schedule can be costly
  - only done periodically when needed

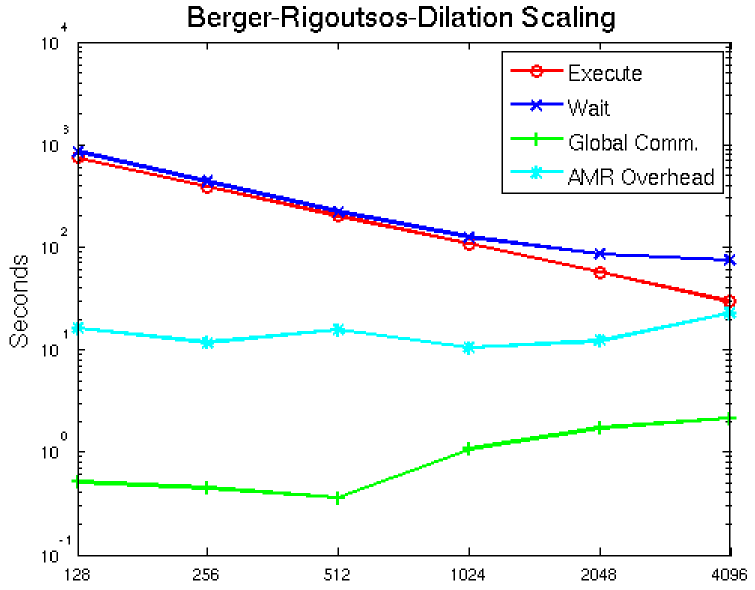


Scalability

Challenging dynamically changing workload

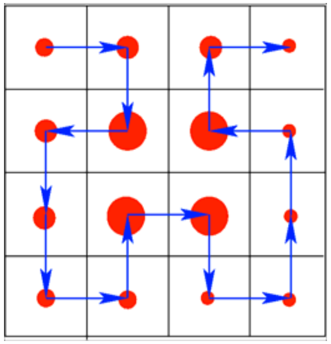


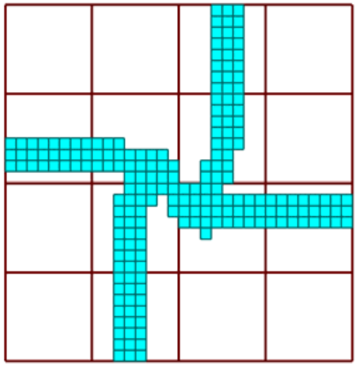


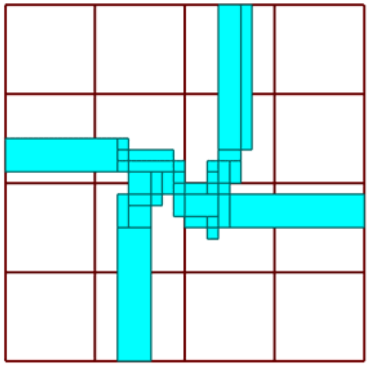


AMR Scaling Developments

New fast space filling curve load balancer plus new modified Berger Rigoutsos algorithm

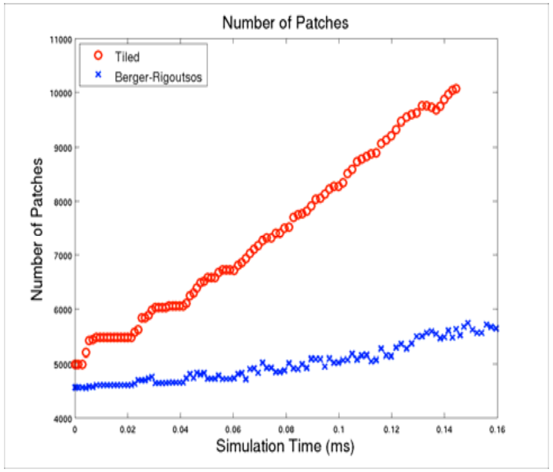


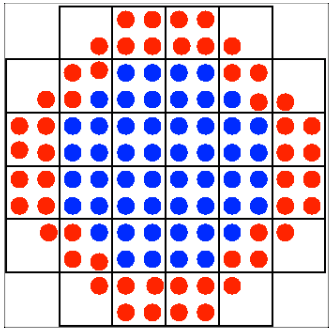




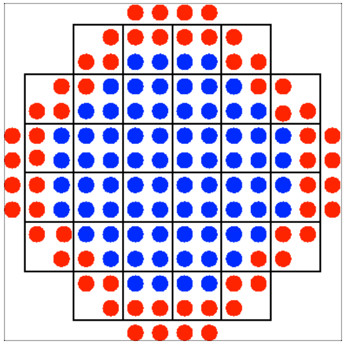
Careful splitting of patches so just enough

Speculative patch sizing so less frequent remeshes





Remesh needed



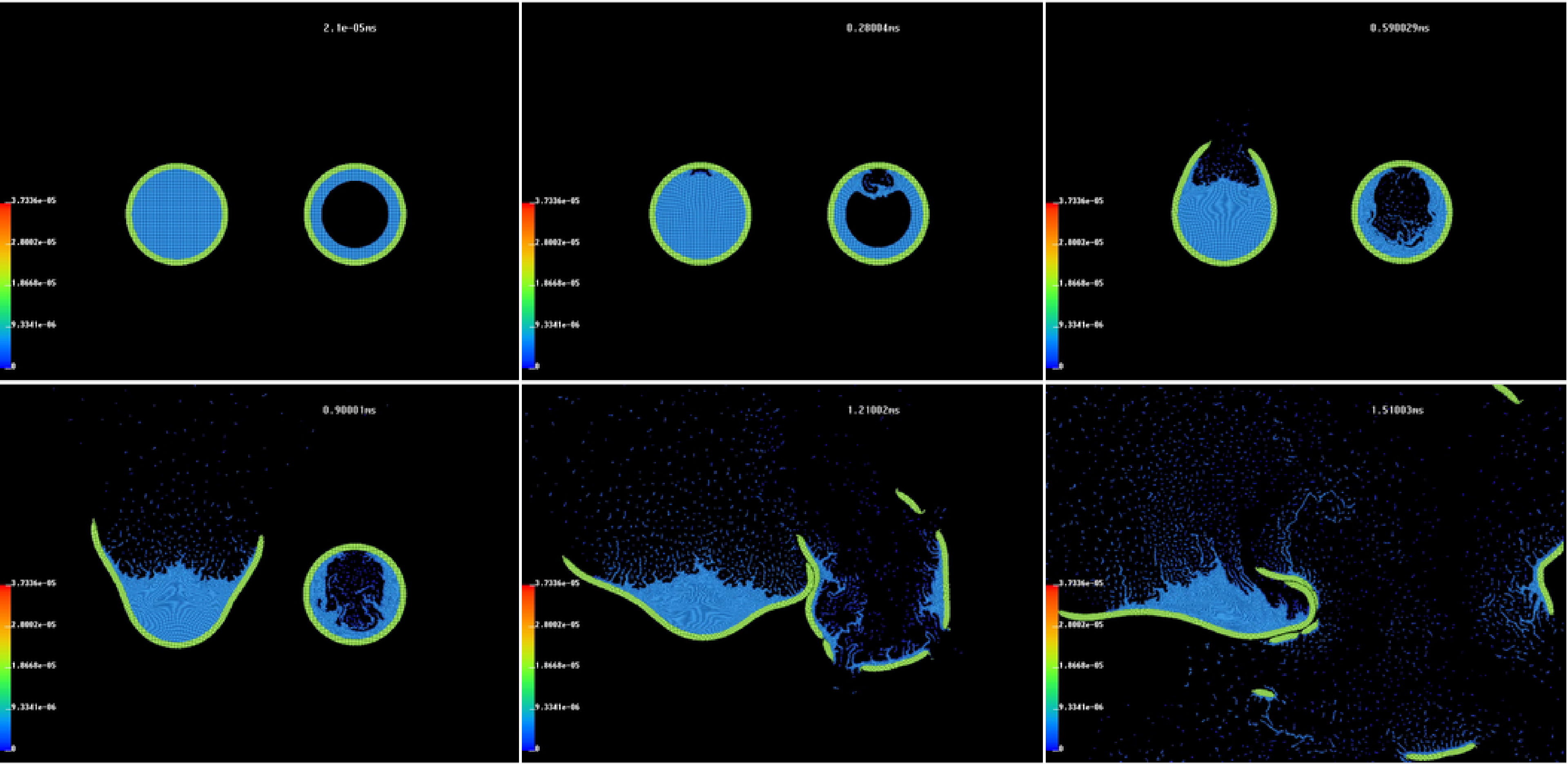
remesh not needed

● **Refinement flag**

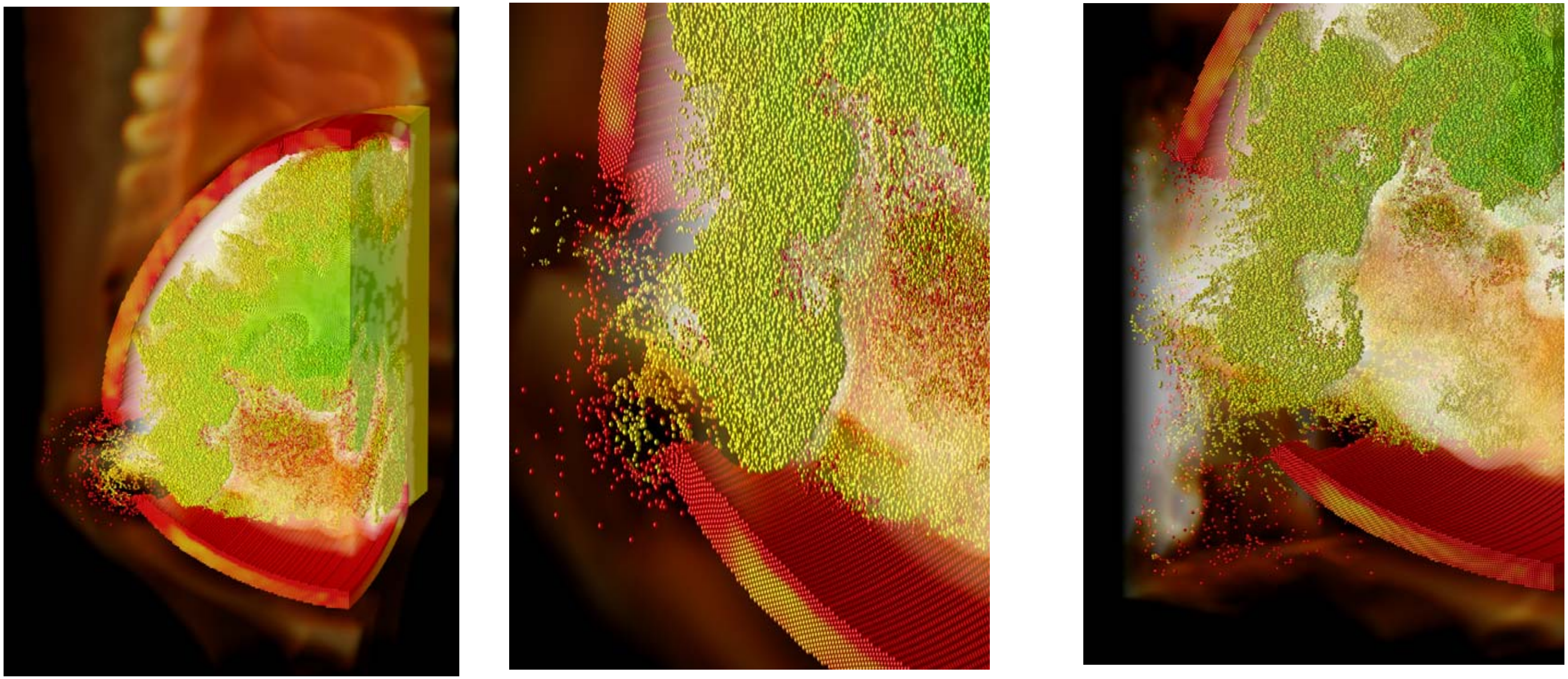
● **Dilated flag**

Dual Container Experiment

left – solid explosive, right- explosive with air



Snapshots of Container Rupture



Summary

- Uintah has automated parallelism for the user at the cost of sophisticated infrastructure (and $50M).
- Task graph as scheduled overlaps communications and computation.
- Good load balancing still requires fast advanced algorithms and precise models.

- Scalability depends a lot on availability of large machines.
- **Ease of portability is a major issue for efficient use of parallel machines –especially novel architectures**.

Center for the Simulation of Accidental Fires and Explosions (C-SAFE)

University of Utah

Supercomputing 2004