



The Center for Astrophysical Thermonuclear Flashes

High Performance Reactive Fluid Flow Simulations Using Adaptive Mesh Refinement on Thousands of Processors

A.C. Calder, *University of Chicago*, B.C. Curtis, LLNL, L.J. Dursi*, B. Fryxell, *University of Chicago*, G. Henry, *Intel*, P. MacNeice, K. Olson, *NASA GSFC*, P. Ricker, R. Rosner, F.X. Timmes, *University of Chicago*, H.M. Tufo, *University of Chicago/Argonne National Laboratory*, J.W. Turan, M. Zingale, *University of Chicago*



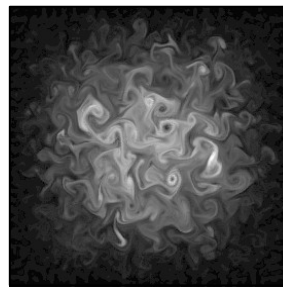
An Accelerated Strategic Computing Initiative (ASCI)
Academic Strategic Alliances Program (ASAP) Center
at The University of Chicago





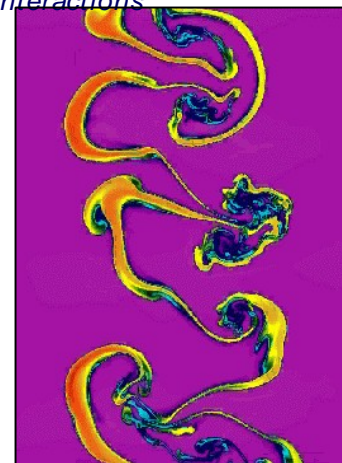
Simulations the FLASH code must handle

- Wide range of compressibility
- Wide range of length and time scales
- Many interacting physical processes
- Large variety of problems, machines
- Fairly simple geometries
- Many people in collaboration

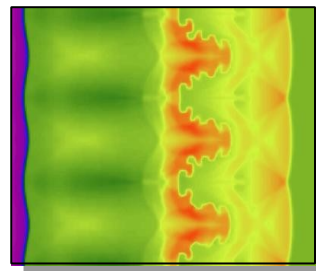


Compressed turbulence

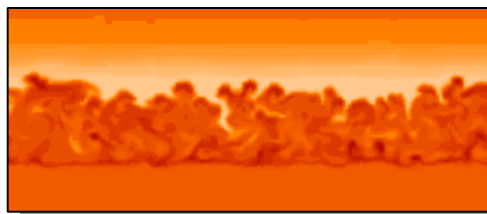
Flame-vortex interactions



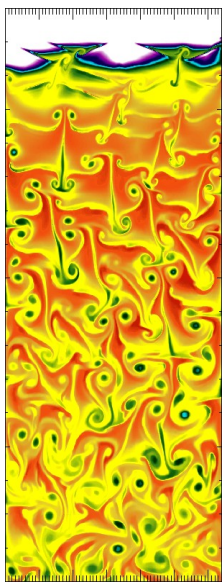
Richtmyer-Meshkov instability



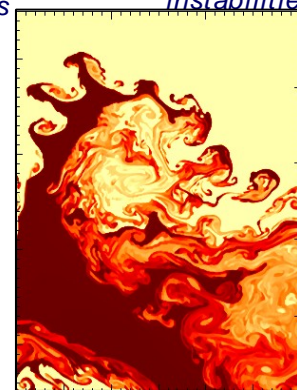
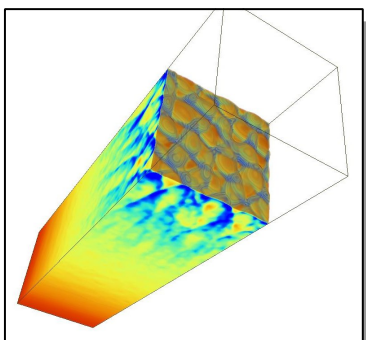
Laser-driven shock instabilities



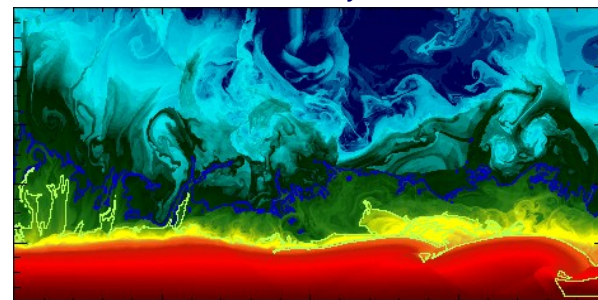
Nova outbursts on white dwarfs



Cellular detonations



Rayleigh-Taylor instability



Helium burning on neutron stars



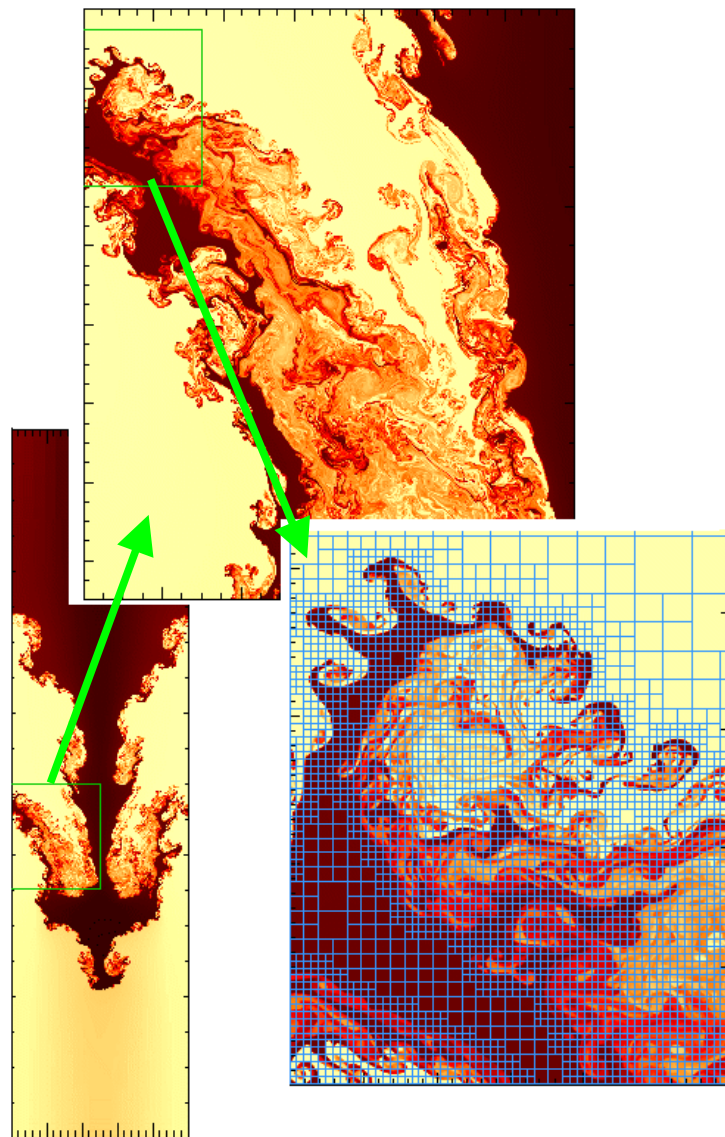
FLASH code

- The FLASH code is:
 - Modular
 - Adaptive Mesh (PARAMESH)
 - Massively Parallel (PARAMESH, MPI, HDF5/ MPI- IO)
- It solves:
 - Compressible Hydro (Euler equations) (PPM)
 - Realistic equation of state
 - Nuclear Reaction Network
 - Other modules since added or being added:
 - Gravity, Radiation Transport, Interface tracking
- Emphasis has been on:
 - Portability - just needs MPI and F90,F77,C compilers
 - Flexibility - easy to add modules and set up new problems
 - Performance - optimized for general distributed memory cache-based machines



AMR

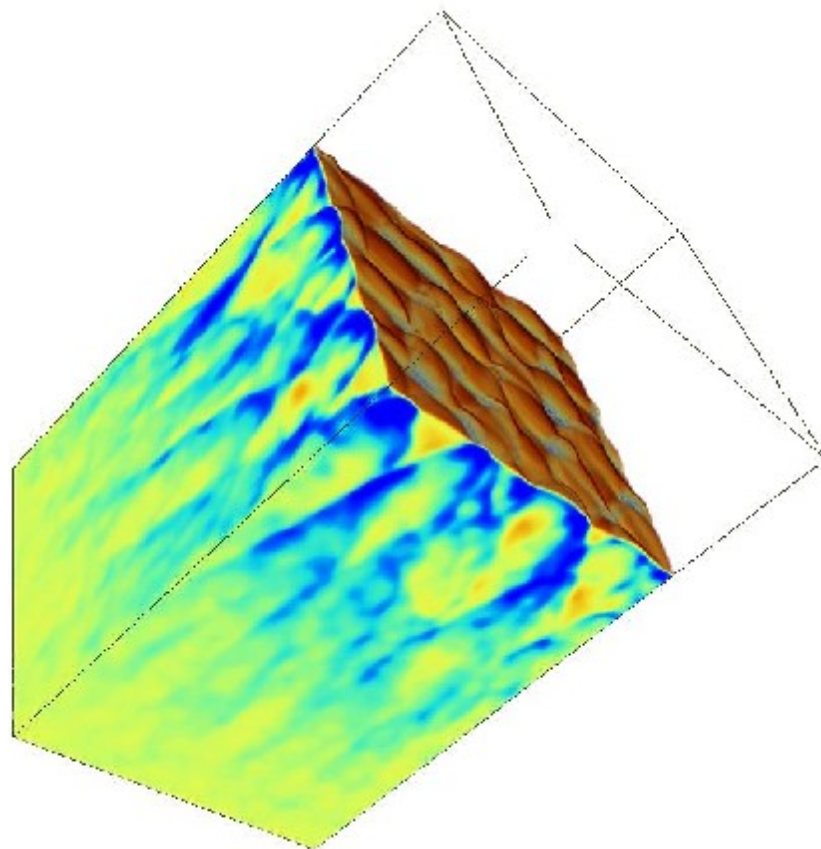
- **Required** because of very large dynamic range of scales.
- **Benefits:**
 - Resolution only where needed
 - Memory, compute time decrease
 - Can do **bigger problems**
- But AMR is really really hard because:
 - Loss of data locality
 - Frequent redistribution
 - Load balancing
 - Irregular, unpredictable memory/ message patterns; hard to precompute things
 - Refinement/ derefinement a black art.





The problem: Cellular Detonation

- In multidimensions, a detonation is not `flat':
- Thermodynamic instabilities cause detonation shock front to break into many smaller reflecting shocks; burning occurs primarily at triple points.
- `Cells' form of less- burned material.





Why it matters

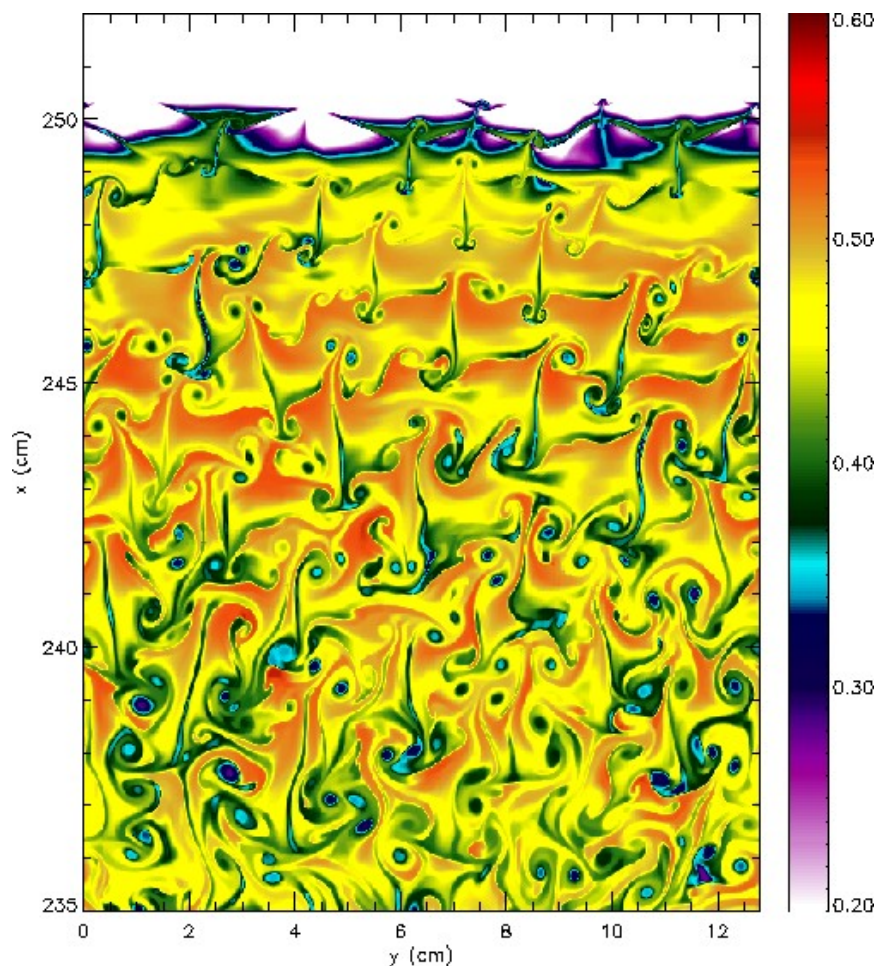


SN1994D (image from HST)

- Our center studies phenomenon involving either astronomical thermonuclear flames or detonations
 - Understanding detonations crucial
- This model detonation relevant for a model of a Supernova Type Ia
- The detonation speed determines the rate of energy output for the supernova
- The multidimensional nature of the detonation can influence the detonation speed



Why it matters



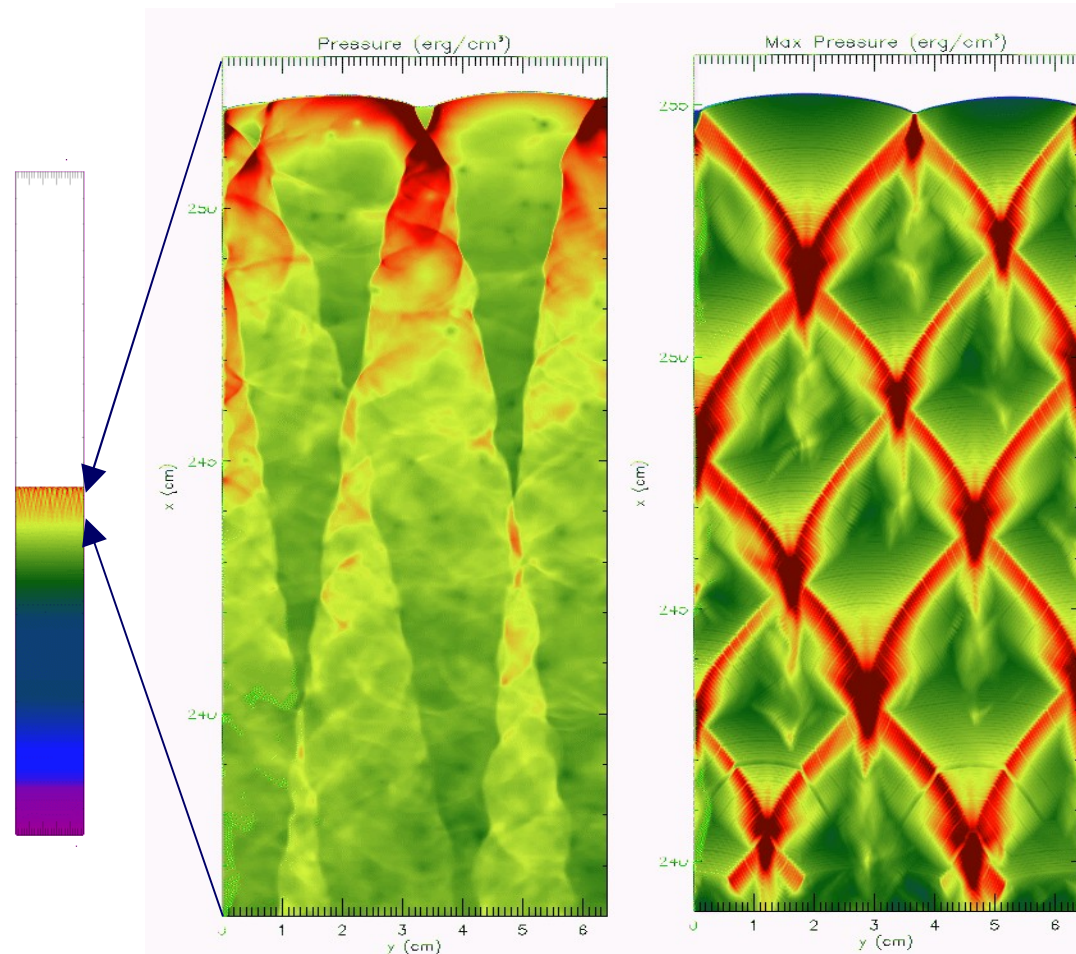
- Further, structure of detonation front can affect burning, leaving `pockets' of unburned material
- Unburned pockets affect materials produced.
- This is important:
 - Different products means the supernova looks different
 - Almost all heavy elements (eg, planets, people, computers) come from supernovae
 - Understanding products necessary to understand chemical evolution of universe.

^{28}Si (intermediate product of burning) abundance behind detonation front.



AMR in the Cellular Detonation Problem

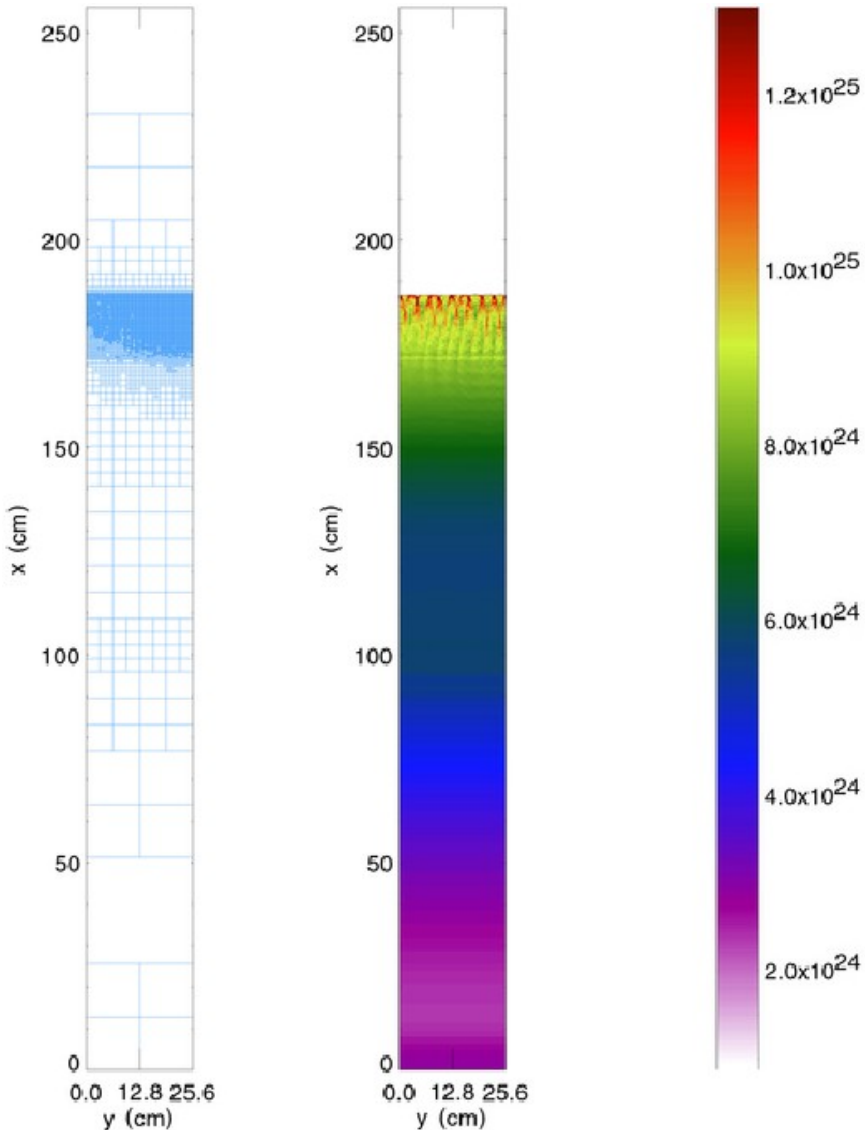
- A very good problem for AMR: all the 'action' occurs near a thin detonation front which propagates through the domain.
- Benefit to us:
 - speedup,
 - 'sizeup': can do bigger problems. (Resolution **crucial** to this problem, to resolve post-front mixing.)



Pressure

Max Pressure

AMR in the Cellular Detonation Problem

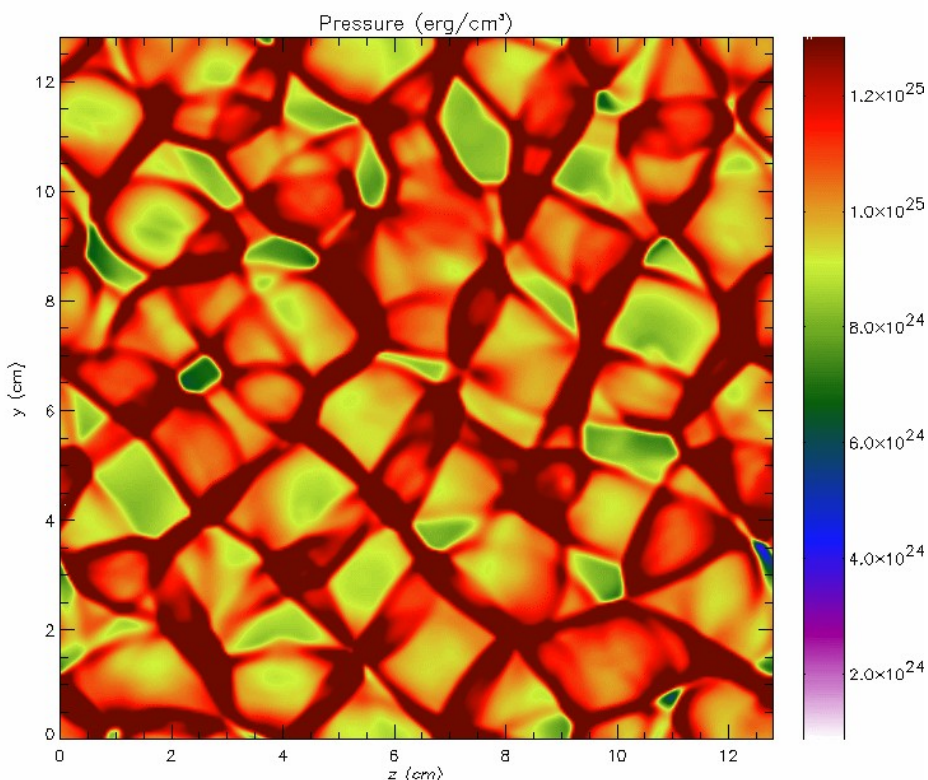


- AMR completely necessary for this problem:
 - Huge problem
 - Can't use moving mesh (susceptible to numerical errors associated with slow-moving shocks)
 - Much larger than any previously done.
 - Completely fills up memory on all ASCI machines, even with AMR
 - Impossible even to fit problem in memory without.



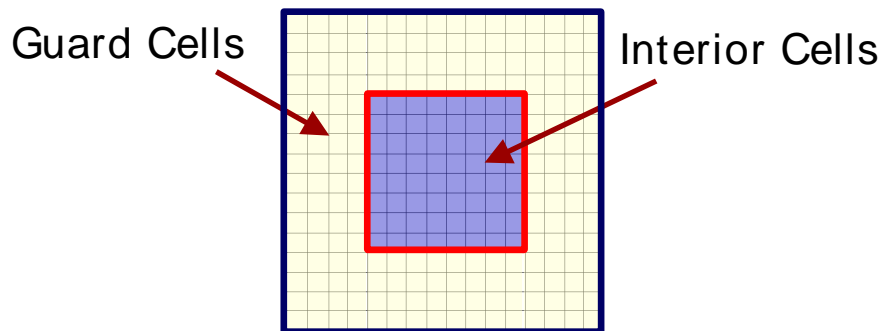
Science Learned Cellular Detonation Problem

- We've learned: (Timmes et al., 2000, submitted ApJ; Timmes et al., 2000, in preparation)
 - Sub-structure affects the 1D velocity very little
 - We know where in the star during the supernova explosion that the cellular structure becomes important
 - An estimate for the scale and amount un- or partially- burned materials, which in principle can be observed





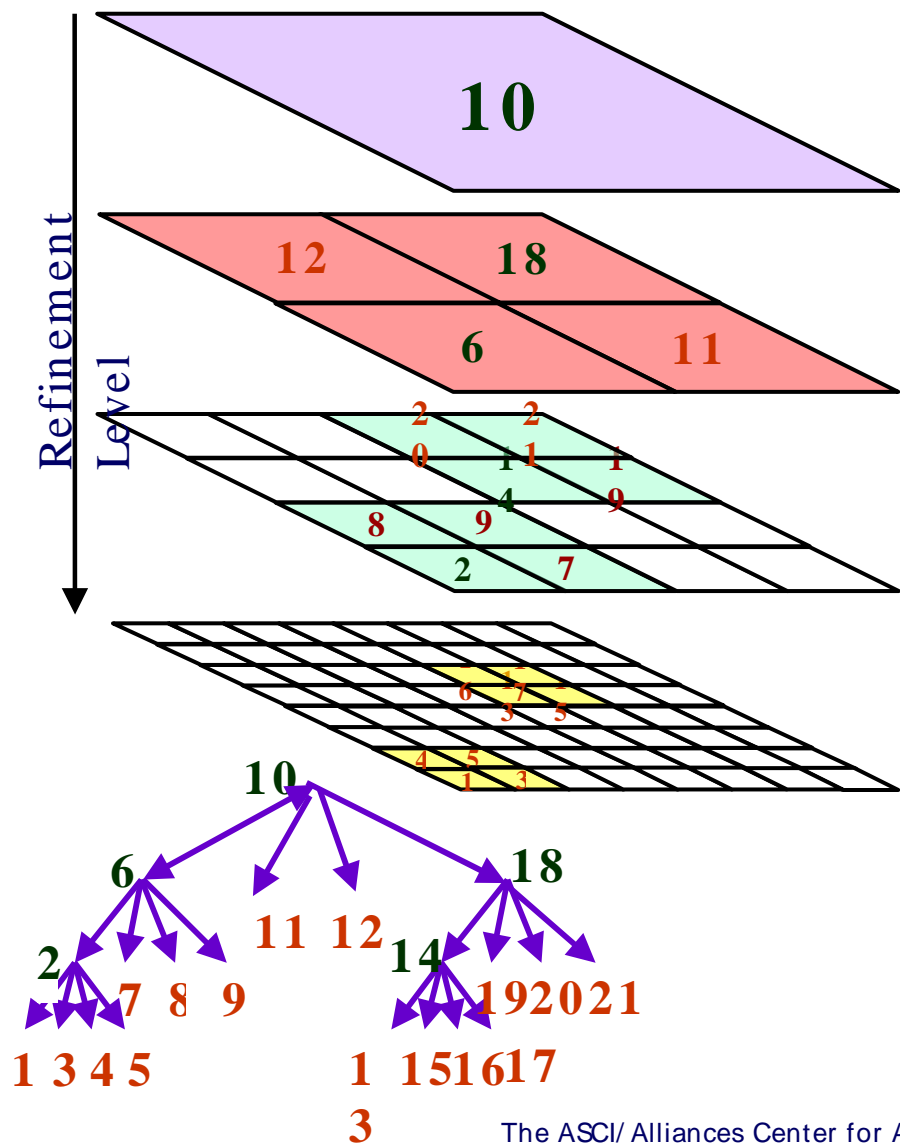
AMR library: PARAMESH



- Developed at NASA/ GSFC
- Local physics (hydro, eos, burn, etc.) occurs on a block. When guardcells are filled, operators can act on each block as if isolated.
- Number of guardcells depends on stencil size.
- Number of interior points must balance tradeoff:
 - More cells - more efficient (until block too big for cache)
 - Fewer cells - can refine more quickly in smaller area.



Refinement

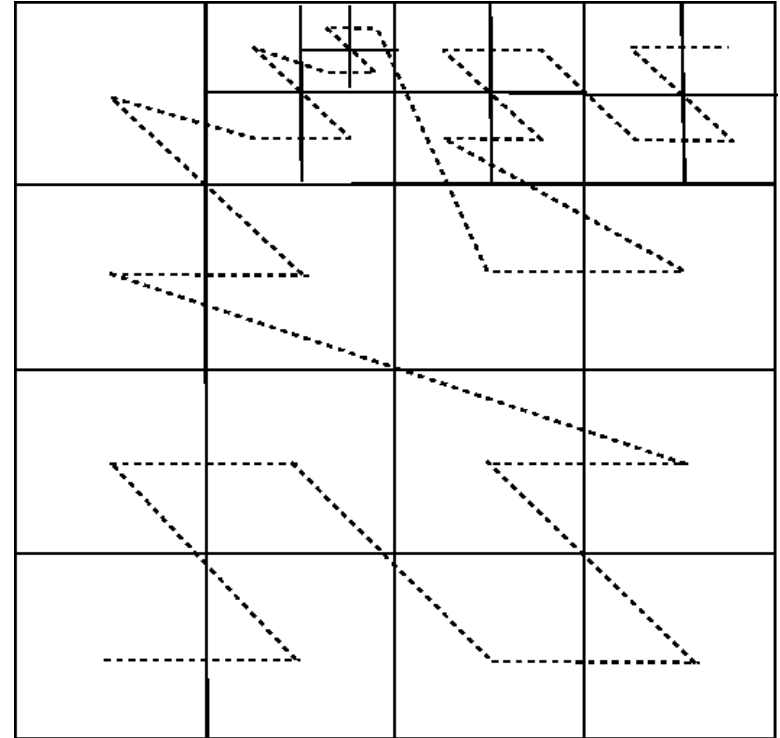


- Blocks and refinement are arranged in an oct-tree structure.
- When a block is refined, 2^d child blocks are created, each with a factor of 2 resolution increase over its parent.
- Neighboring blocks must differ by at most one level of refinement.
 - Drawback: resolution can only fall off linearly in distance.
 - Feature: simplifies, speeds up accurate calculation of 'boundary conditions' (guardcells)



Data Distribution

- Blocks are distributed across processors using a space-filling curve method.
- A curve (typically a Morton curve) is threaded through the mesh, with weights according to how much each block 'costs'.
- 'Cost' could be CPU use, communication, memory.
- Other curves (Hilbert, Peano) with better clustering properties have been tried; no performance enhancement.





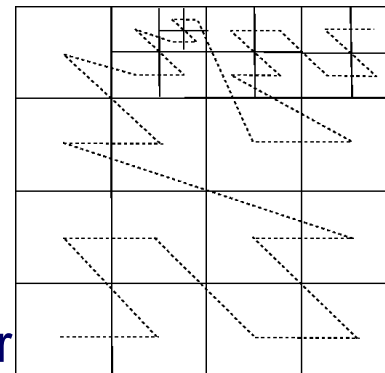
PARAMESH in action: Xray Burst

Mike's movie goes here



AMR/ Parallel Optimizations

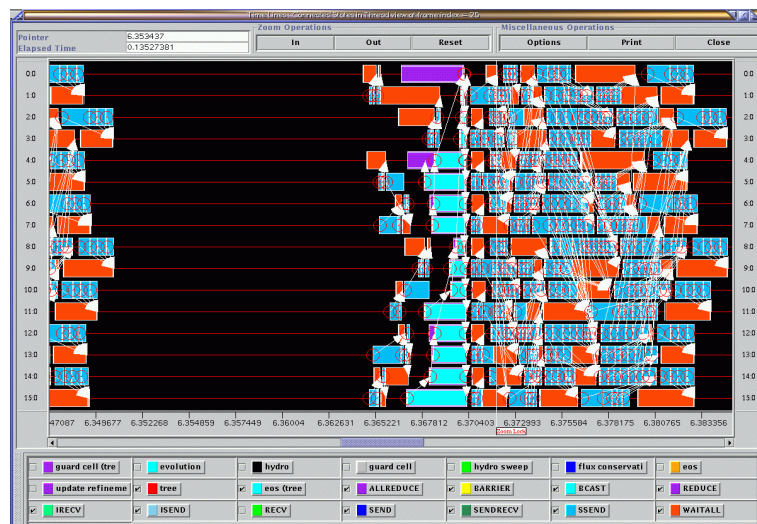
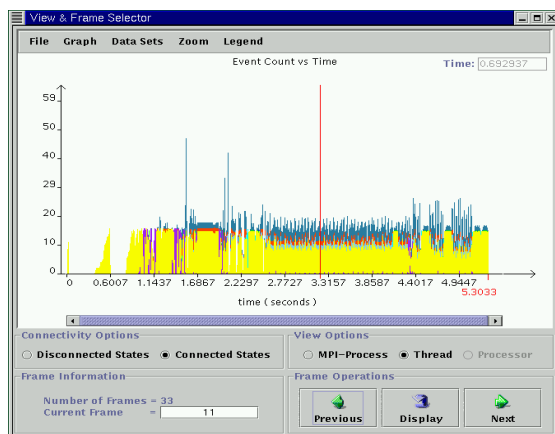
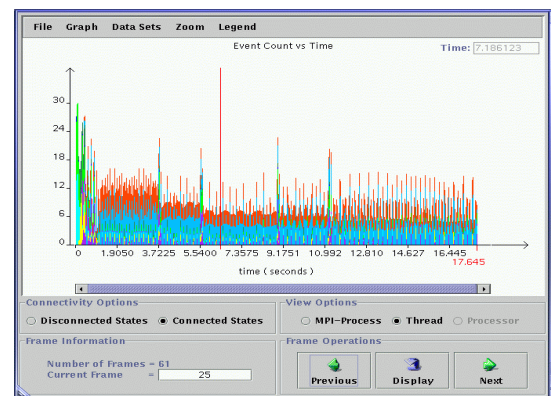
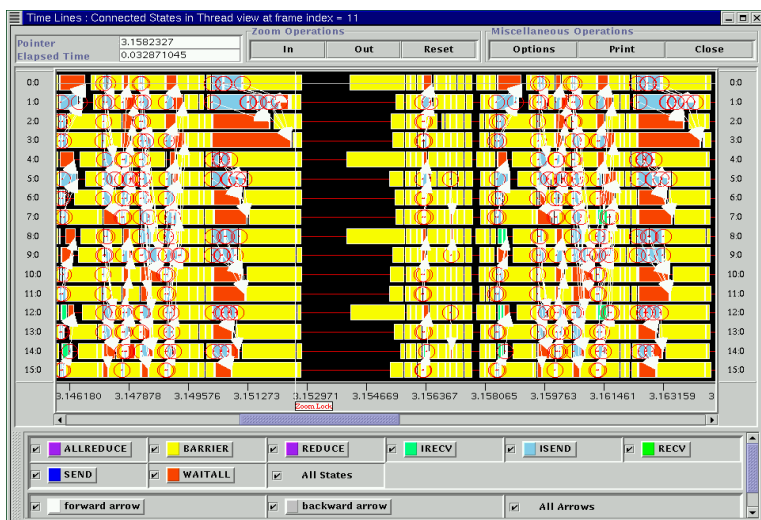
- Removal of global sort; maintain PARAMESH structures though less costly operations
- Previously:
 - Space filling curve calculated per block
 - Values sorted globally to find ordering
 - This ordering is used to redistribute
- But tree, Morton imposes structure on order
- Children of a block always have ordering between parent and next block in ordering
- If one's careful, can maintain global ordering by correct initial distribution and then ordering is maintained by induction.
- No global parallel sort!





Extraneous barriers: before/ after with Jumpshot

- Removal of extraneous barriers; perform synchronization during data communication





Physics Modules

- Hydro: directionally- split PPM, based on PROMETHEUS (Fryxell, Müller, Arnett, 1989)
 - 9- pt stencil in 1D
 - Shock- capturing
 - Expensive, but accurate
 - Works on vectors: columns taken from blocks
- Nuclear burning: (Timmes, 2000)
 - 13 isotopes, 110 (non- zero) rates
 - Uses routines generated by GIFT (sparse operations, but dense matrices)
 - Routines then hand- optimized
 - Operates on entire block at once
- EOS (Timmes and Swesty, 1999)
 - Thermodynamically consistant table- lookup scheme
 - Extremely efficient (~100 Mflops on Red)
 - Operates on entire block



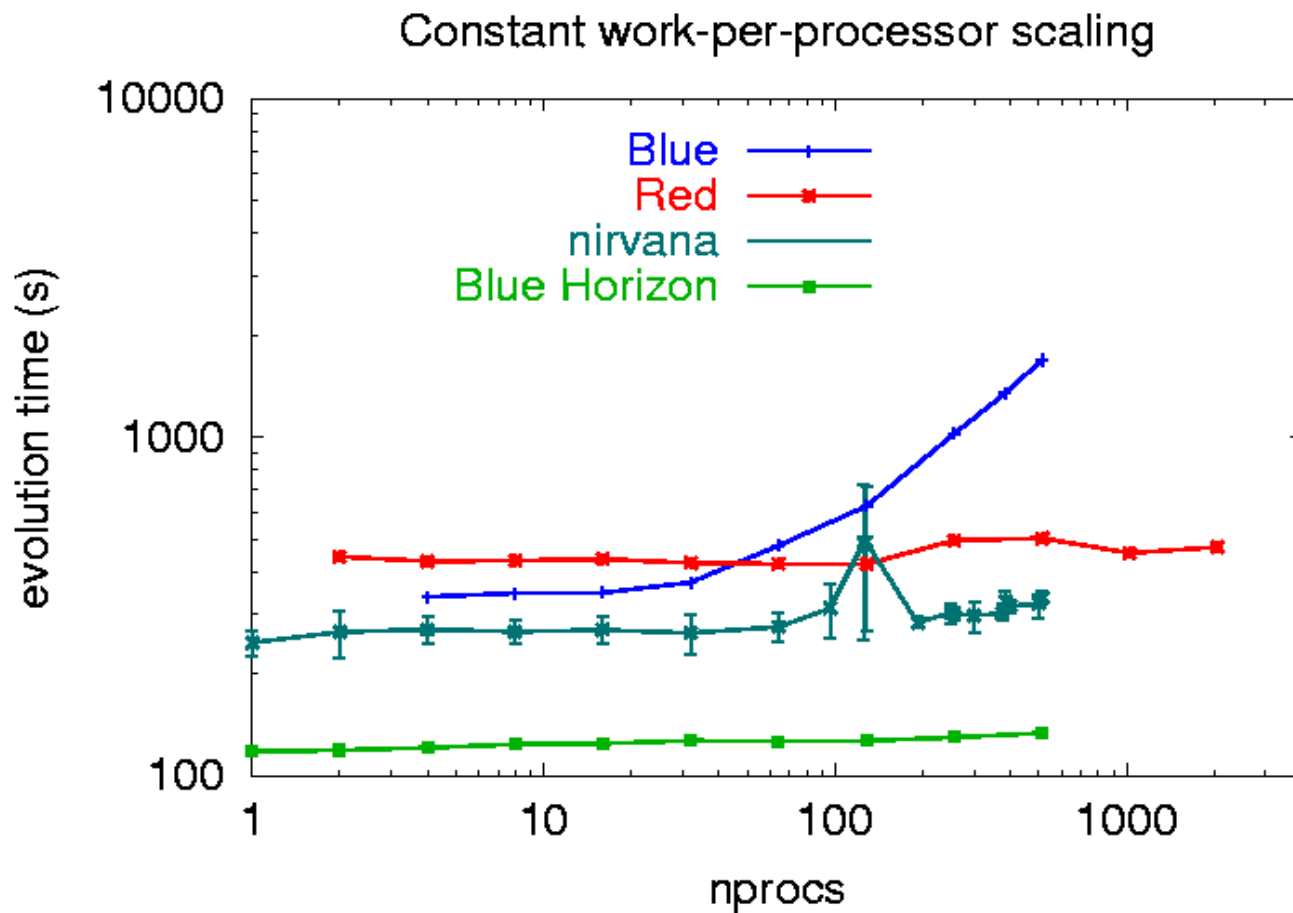
Physics Module/ Single- Proc Optimizations

- Extensive (but not problem- or platform- specific) single-processor optimizations
 - Rewriting and reordering some math
 - Restructuring code to reduce memory copies
 - Making as much known at compile time as possible (let optimizers do their thing)
 - Loop fusion to reduce memory use for arrays of intermediate values



FLASH code scaling: 2D hydro

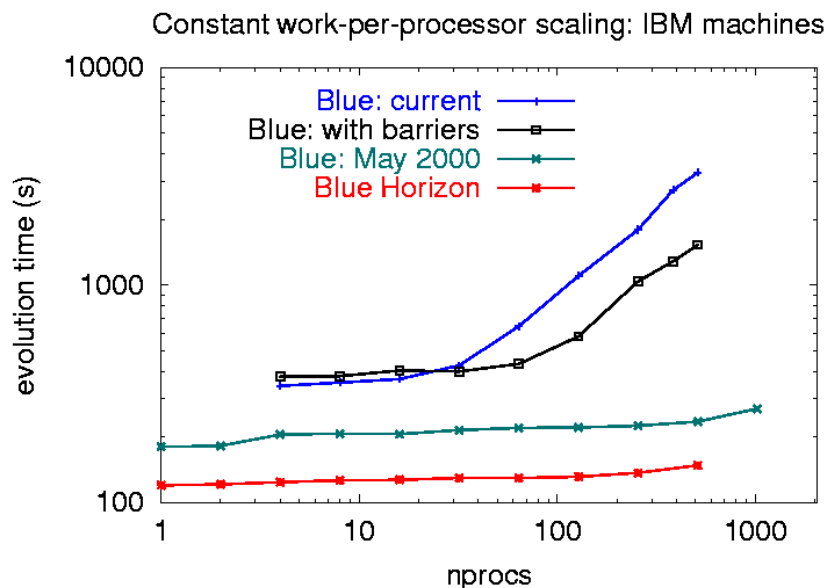
- Constant work-per-processor test: Ideal is flat line.



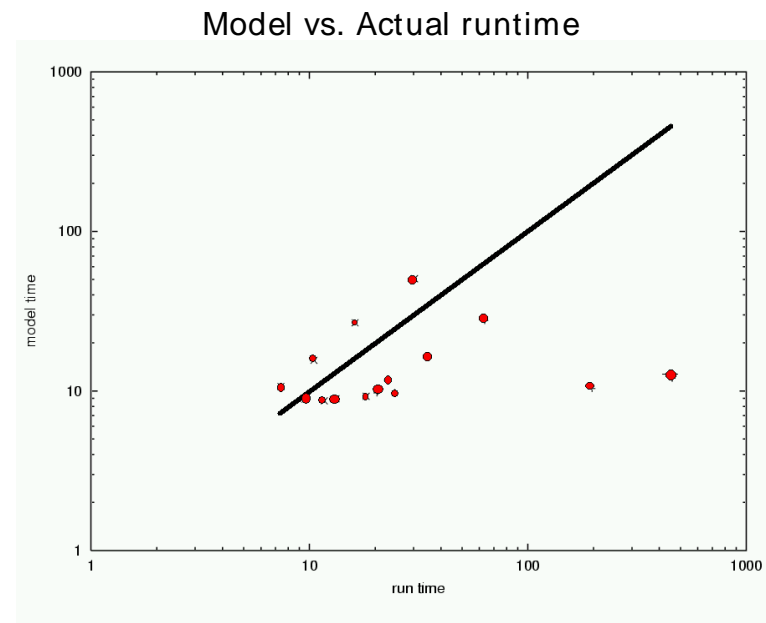


FLASH code scaling: 2D hydro

- Blue Pacific has been a mystery for the 2d hydro- only scalings:



Blue scaling over time: removal of extraneous barriers this summer **increased** runtime



Blue scaling compared to a preliminary performance model; model ok up to ~64 procs, then blue greatly underperforms model.

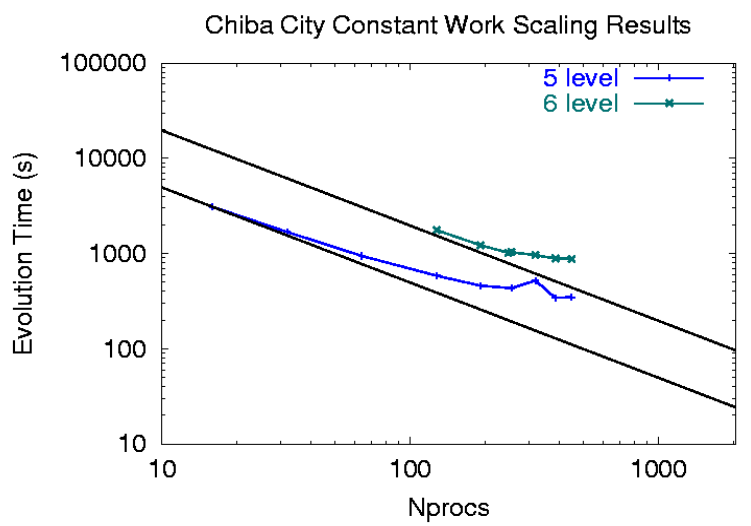
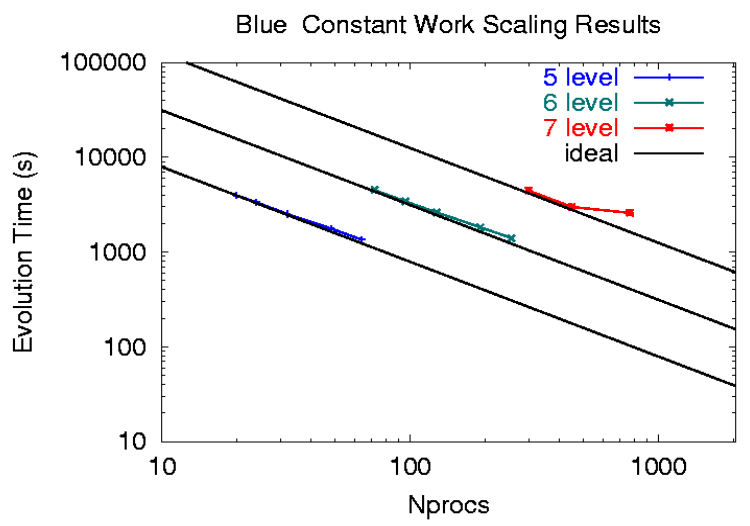
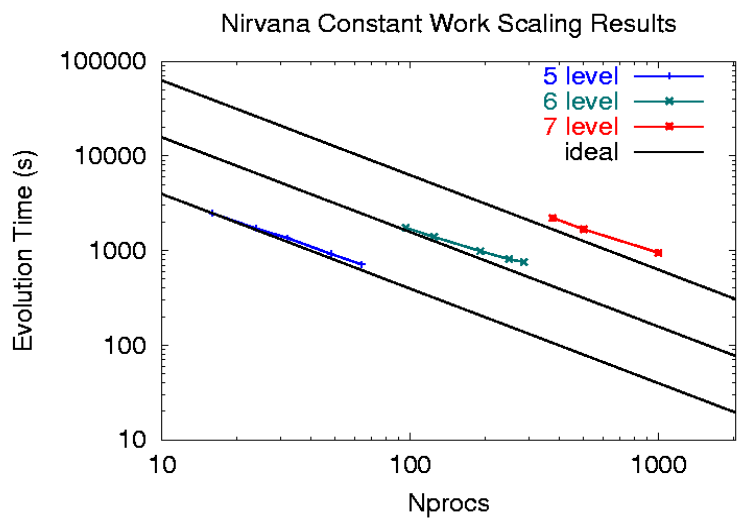
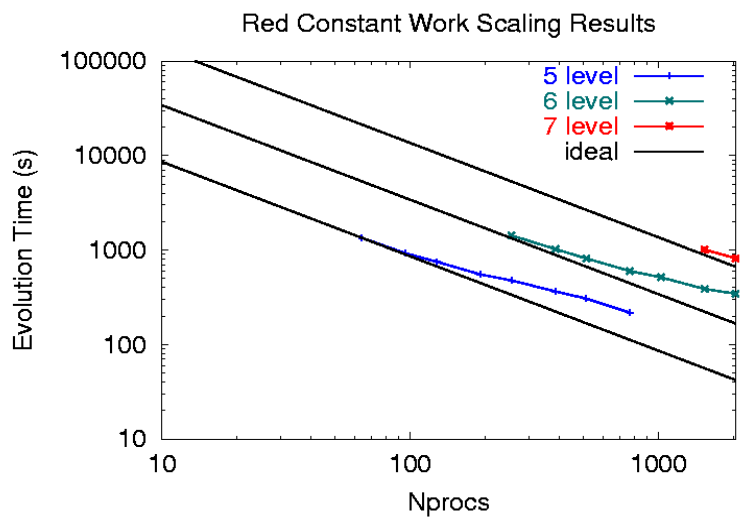


Choices made for this run

- Chose to run on Red:
 - Blue most promising memory-wise, but performance not understood
 - Nirvana: takes huge amount of effort tweaking MPI parameters to get good performance; even so, huge variability in run-to-run.
 - Red: good performance as long as processors have enough work
- Performance Tweaks:
 - Modification of code to allow ASCI Red's compiler to inline routines from within modules (platform specific)
 - Hardcoding block-sizes (problem specific)



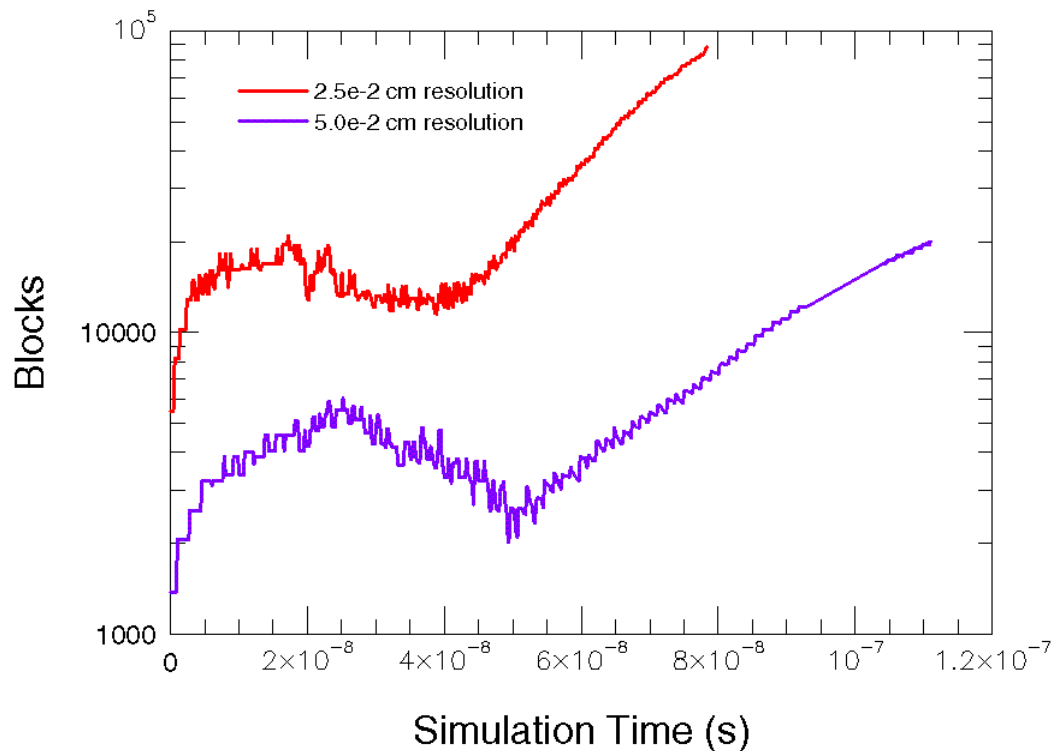
3D cellular detonation: Fixed problem size scaling





AMR in the Cellular Detonation Problem

- Many blocks created, destroyed, and transferred over the run
- Nearly 65% of mesh changed every update (8 hydro steps)
- Block count goes up over time as detonation 'goes cellular' and cellular structure approaches steady state.
- Even with AMR, not enough memory to follow entire evolution at 7 levels.





Performance Results

Number of Processors	Refinement Levels	Blocks per Processor	MFLOPS per Processor	Total GFLOPS
Two Processors per Node				
6420	8	33.1	37.0	238
4096	8	36.1	39.2	161
2048	7	33.7	40.2	82
One Processor per Node				
3072	8	58.2	47.4	146
2048	8	72.1	45.2	93
1024	7	63.5	50.2	51

- ~11% of peak (from ASCI Red hardware counters)
- Used ~70% of memory for largest run.
- Very good scaling (6% loss of efficiency from 4096 to 6420)



FLASH code: Performance Results

Only platform- or problem- specific tweaks:

- Hardwiring block size

- Code tweaks to allow compiler to inline in- module functions

Results from 6420- processor run on ASCI Red: Average over last half of run (avoids initialization, work- starvation early in evolution)

- 60%physics

 - (of which: 64%burn, 18%hydro, 18%eos)

- 40%AMR+ parallel overhead

 - (of which: 50%guardcells, 50%redistributing/ updating mesh)

Sustained performance: **238 GFlops**, 64 bit arithmetic



Conclusions

- The Flash center has developed a very scalable, high-performance, production code being used now for real astrophysical problems.
- The largest astrophysical detonation simulation ever done was performed
- New science is being done
 - Learned about structure of detonation, and effects on supernove
- Future:
 - Continuing to develop refinement/ derefinement strategies
 - Tuning space- filling curve to problem (` columns' of morton ordering)
 - Reordering processors for ASCI Red
 - Tuning redistribution of blocks algorithm
 - Performance testing on ASCI White
 - Prototype of hybrid OpenMP/ MPI code



Questions (then Lunch!)

We'd like to acknowledge our sponsors at DOE
ASCI/ ASAP program: <http://www.llnl.gov/asci/>

- Cell Det movie here
- (
http://www.mcs.anl.gov/~hensley/cell_pres.r
)

Visualization: ANL

Come see this and other movies at the Argonne

booth!

Also: <http://flash.uchicago.edu>