Barriers to Computing at Scale : Hardware, Algorithms, Modeling

Robert Fisher
University of Massachusetts Dartmouth

Future of AstroComputing : December 16, 2010

In collaboration with

University of Chicago	SUNY Stony Brook	Hebrew University
Donald Lamb Jim Truran Dean Townsley George Jordan Nathan Hearn Carlo Graziani Casey Meakin	Alan Calder	Shimon Asida

With Special Thanks To

University of Chicago	Argonne National Laboratory	Lawrence Berkeley National Laboratory
Anshu Dubey Brad Gallagher Lynn Reid Paul Rich Dan Sheeler Klaus Weide	Ray Bair Susan Coghlan Randy Hudson John Norris Mike Papka Katherine Riley	Katie Antypas

Outline

I. Modeling Turbulence in Astrophysical Simulations

II. Hardware, Algorithms, and Asymptotically-Large Simulations

I. Modeling Turbulence in Astrophysical Simulations

Post-Millenial Computational Astrophysics

- Large-Scale Structure
- Compact Objects, Accretion Disks
- SF at high and low z, high and low mass
- SNe Ia & II
- Galaxy Formation

The Universal Nature of Turbulent Flows

The Universal Nature of Turbulent Flows

- Turbulence at high Reynolds is *universal* - the inertial scaling laws of a homogeneous, isotropic turbulent velocity field are independent of the driving.

The Universal Nature of Turbulent Flows

- Turbulence at high Reynolds is *universal* - the inertial scaling laws of a homogeneous, isotropic turbulent velocity field are independent of the driving.

The Universal Nature of Turbulent Flows

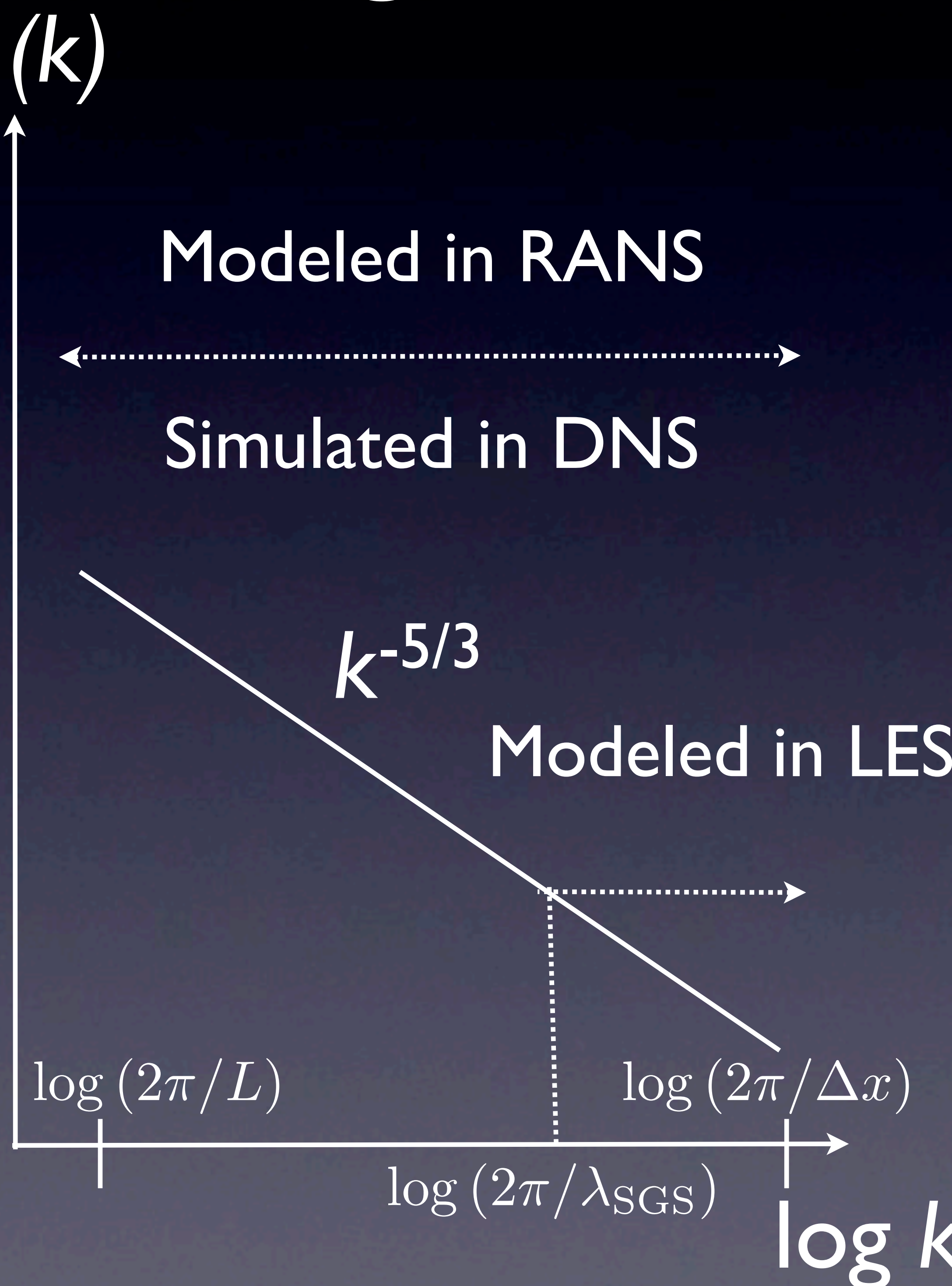
- Turbulence at high Reynolds is *universal* - the inertial scaling laws of a homogeneous, isotropic turbulent velocity field are independent of the driving.
- This is one of the deepest lessons of Kolmogorov (1941).

Hierarchy of Fidelity in Turbulence Modeling

- Direct Numerical Simulation (DNS)
  - Resolves Kolmogorov scale

$$\eta \simeq 2 - 4\Delta x$$

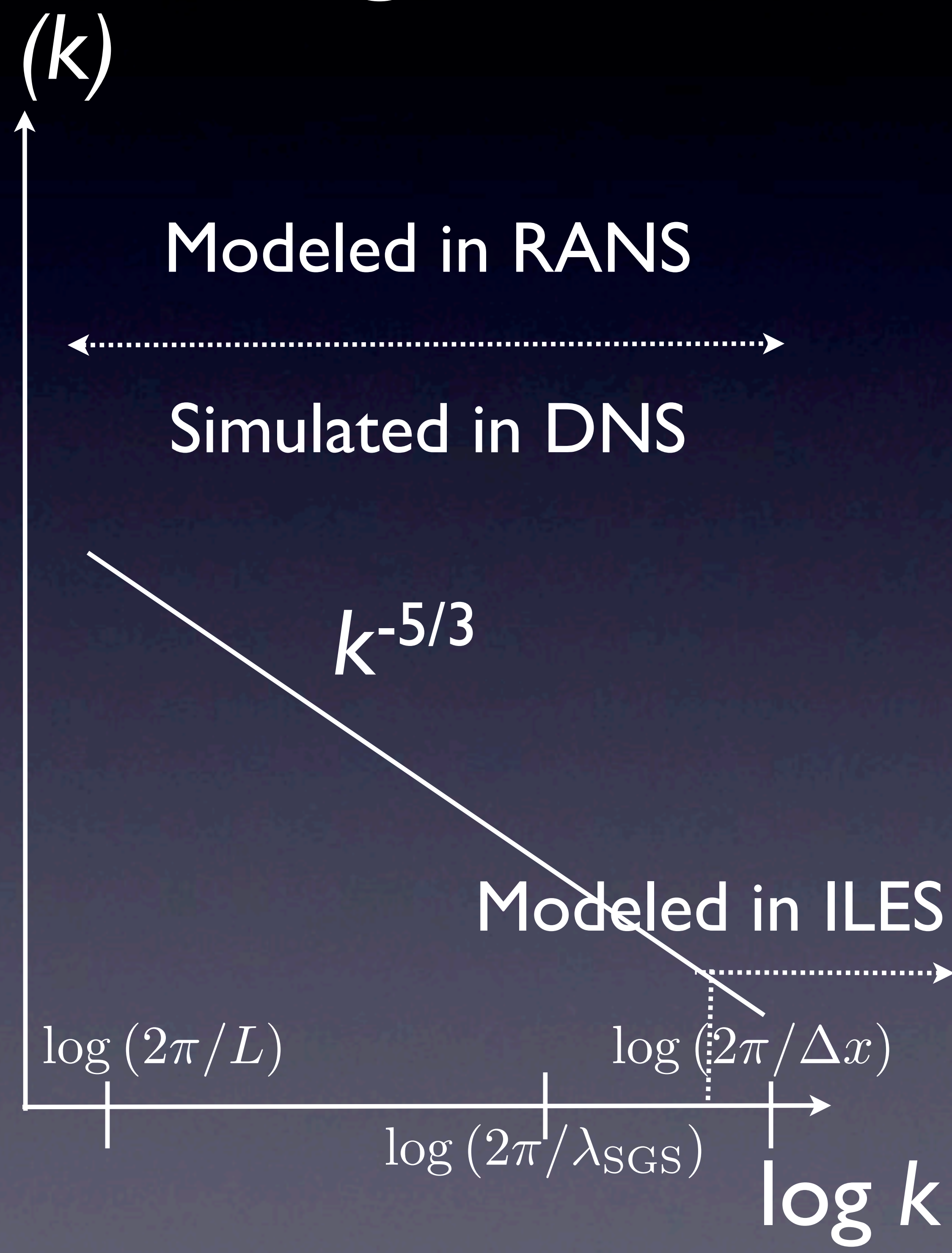
- Large Eddy Simulation (LES)
  - Introduces a subgrid model below the filter scale λ_{SGS}
- Reynolds-Averaged Navier Stokes (RANS)



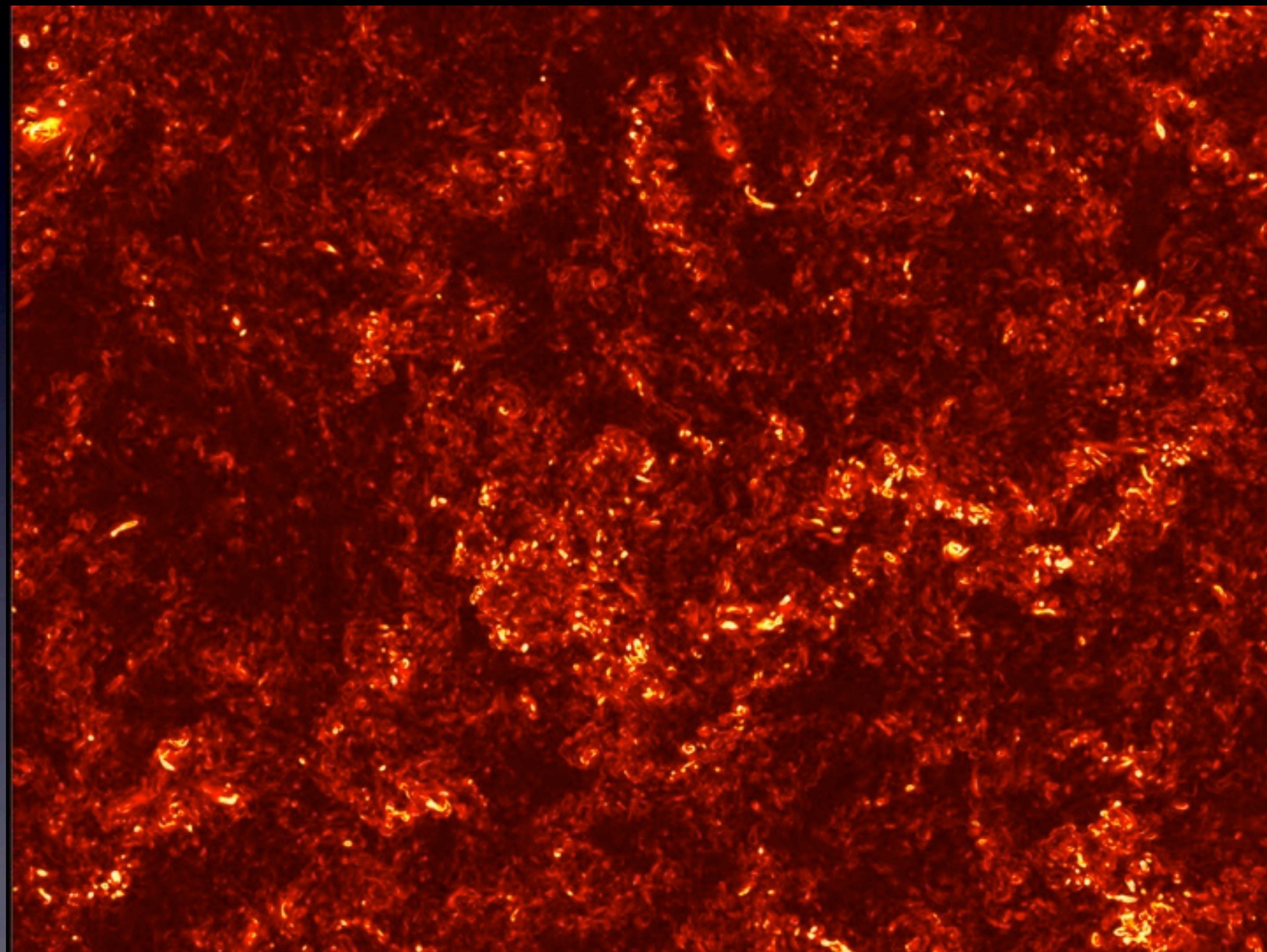
Hierarchy of Fidelity in Turbulence Modeling

- Implicit Large Eddy Simulation (ILES)
    - Numerical solution to Euler equations
    - Introduces an effective subgrid model and an effective viscosity through numerical dissipation

$$\eta \simeq \Delta x$$



Weakly-Compressible Hydrodynamic Turbulence



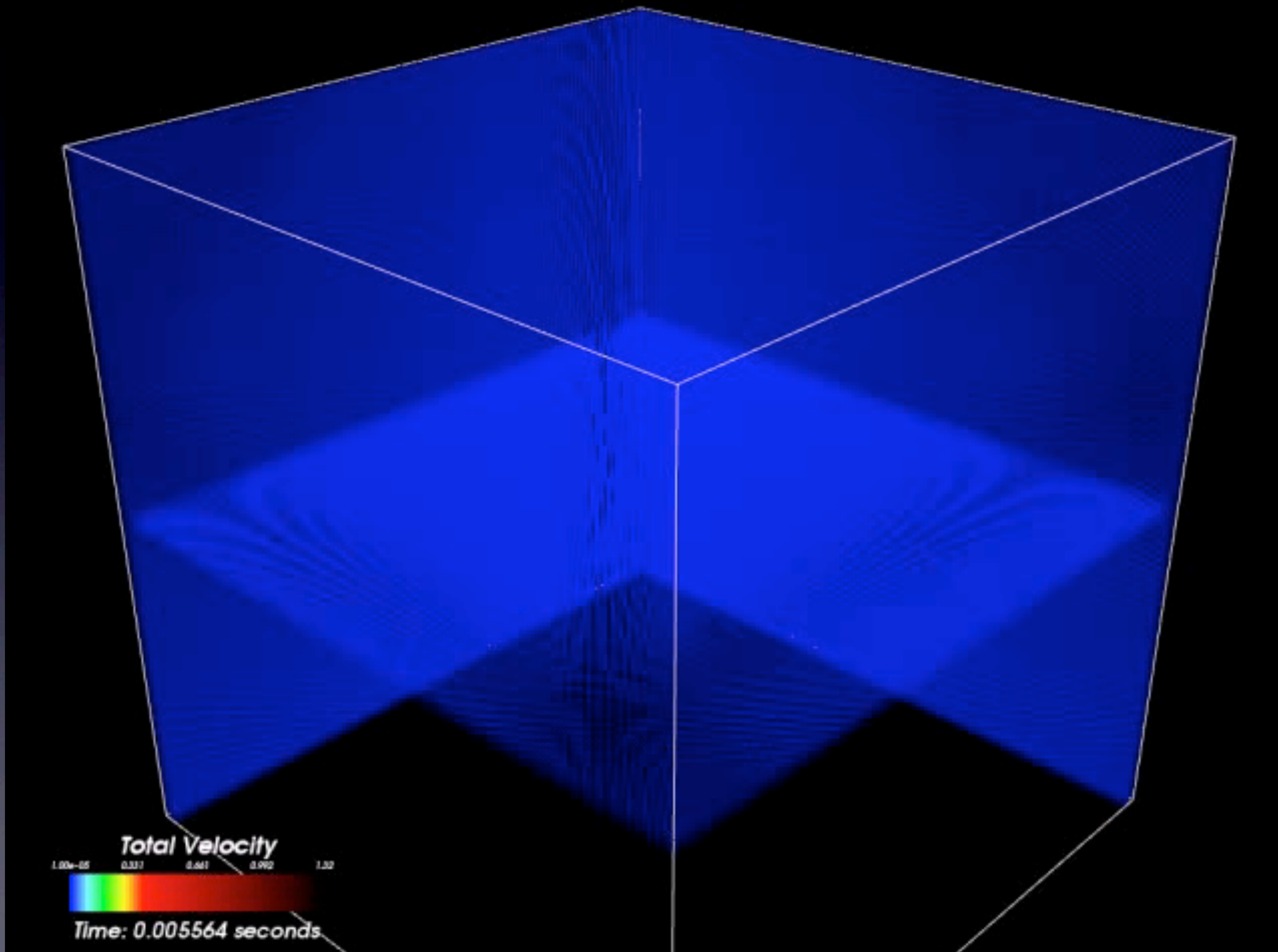
(Fisher *et al*, 2008, Benzi *et al*, 2008, Arneodo *et al*, 2008, Benzi *et al*, 2010)

BG/L Turbulence Run

- Large-scale homogeneous, isotropic compressible fully-developed turbulence :
  - 1856^3 base grid size
  - 256^3 Lagrangian tracer particles
  - 3D turbulent RMS Mach number = 0.3 (1D = .17) in steady-state
  - $Re_\lambda \sim 600$
  - Roughly one week wall clock on 65,536 processors in CO mode

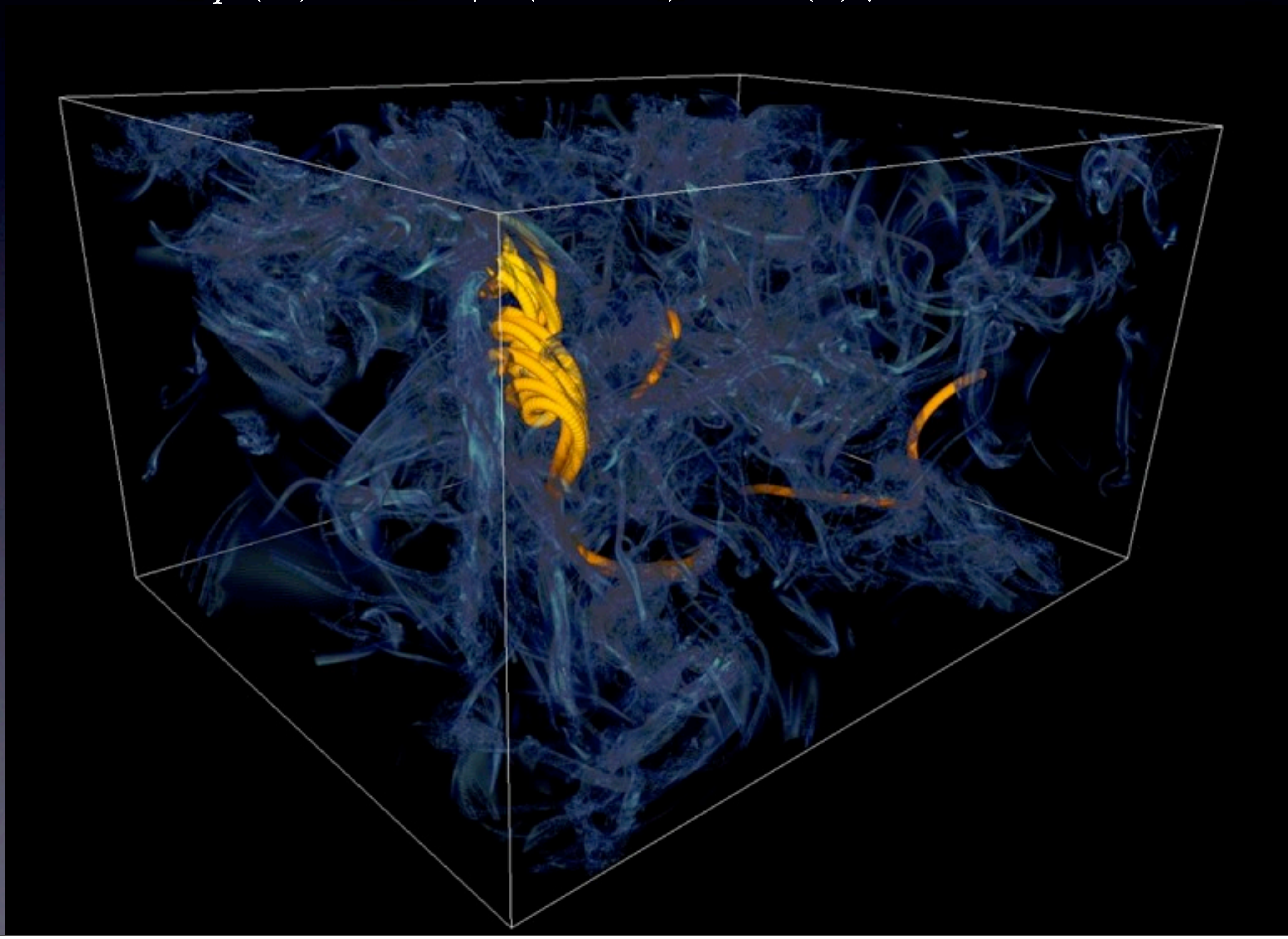
Visualization of Lagrangian Tracers

Visualization of Lagrangian Tracers



Universality of Lagrangian Structure of Turbulence

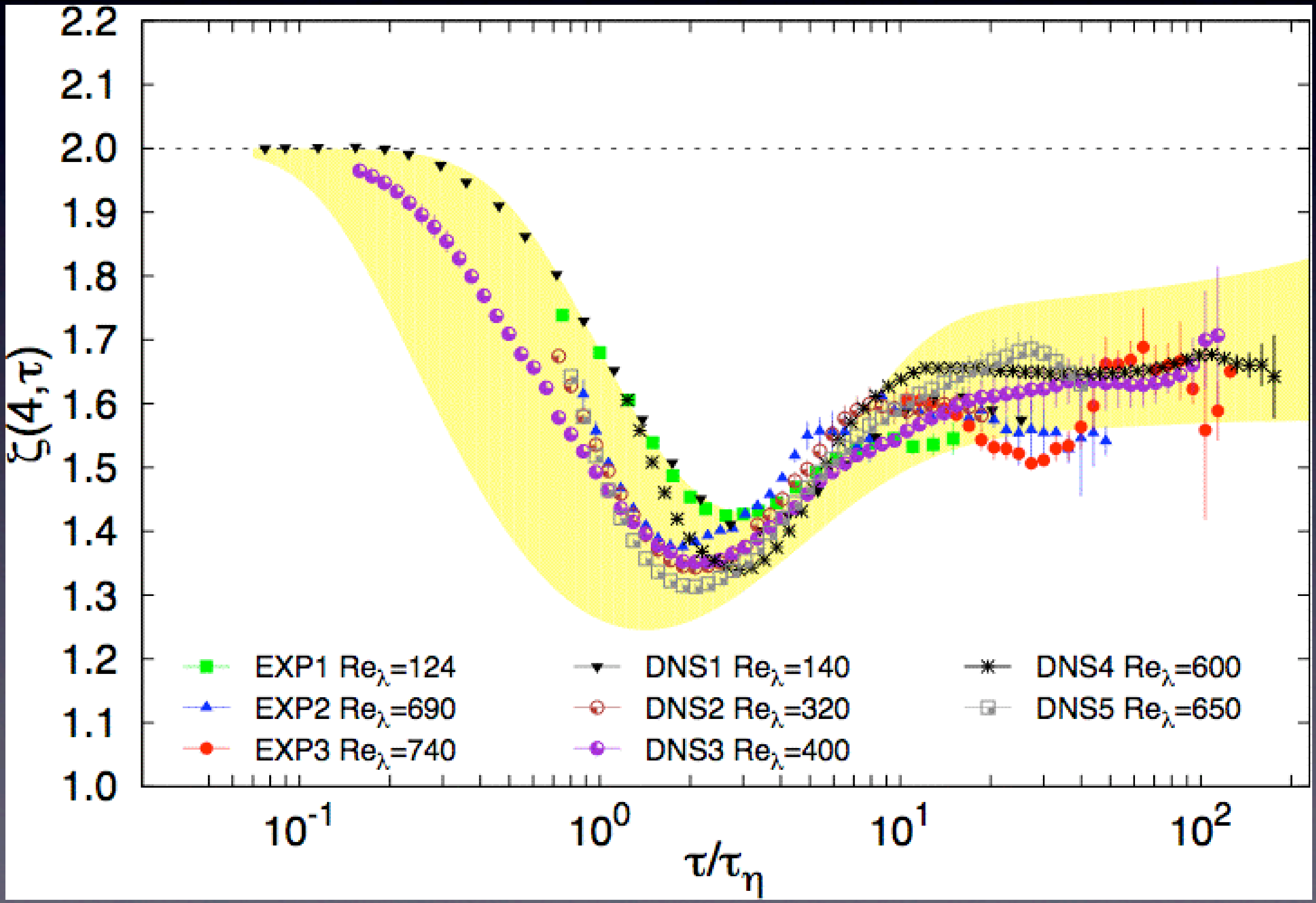
$$S_p(\tau) = < |v(t+\tau) - v(t)|^p > \propto \tau^{\zeta_p}$$



(Arneodo *et al*, 2008)

Universality of Lagrangian Structure of Turbulence

$$S_p(\tau) = < |v(t+\tau) - v(t)|^p > \propto \tau^{\zeta_p}$$



(Arneodo *et al*, 2008)

Astrophysical Simulation Modeling

Astrophysical Simulation Modeling

- Current turbulent modeling succeeds because the velocity field is *universal*

Astrophysical Simulation Modeling

- Current turbulent modeling succeeds because the velocity field is *universal*
- Turbulence modeling may pose significant challenges to future astrophysical studies of coupled multifluid, multiphysics processes :

Astrophysical Simulation Modeling

- Current turbulent modeling succeeds because the velocity field is *universal*
- Turbulence modeling may pose significant challenges to future astrophysical studies of coupled multifluid, multiphysics processes :
  - Turbulent Combustion (SNe Ia)

Astrophysical Simulation Modeling

- Current turbulent modeling succeeds because the velocity field is *universal*
- Turbulence modeling may pose significant challenges to future astrophysical studies of coupled multifluid, multiphysics processes :
  - Turbulent Combustion (SNe Ia)
  - Turbulent Mixing (Planet Form., GMCs/SF, SNe II)

Buoyancy-Driven Turbulent Nuclear Combustion

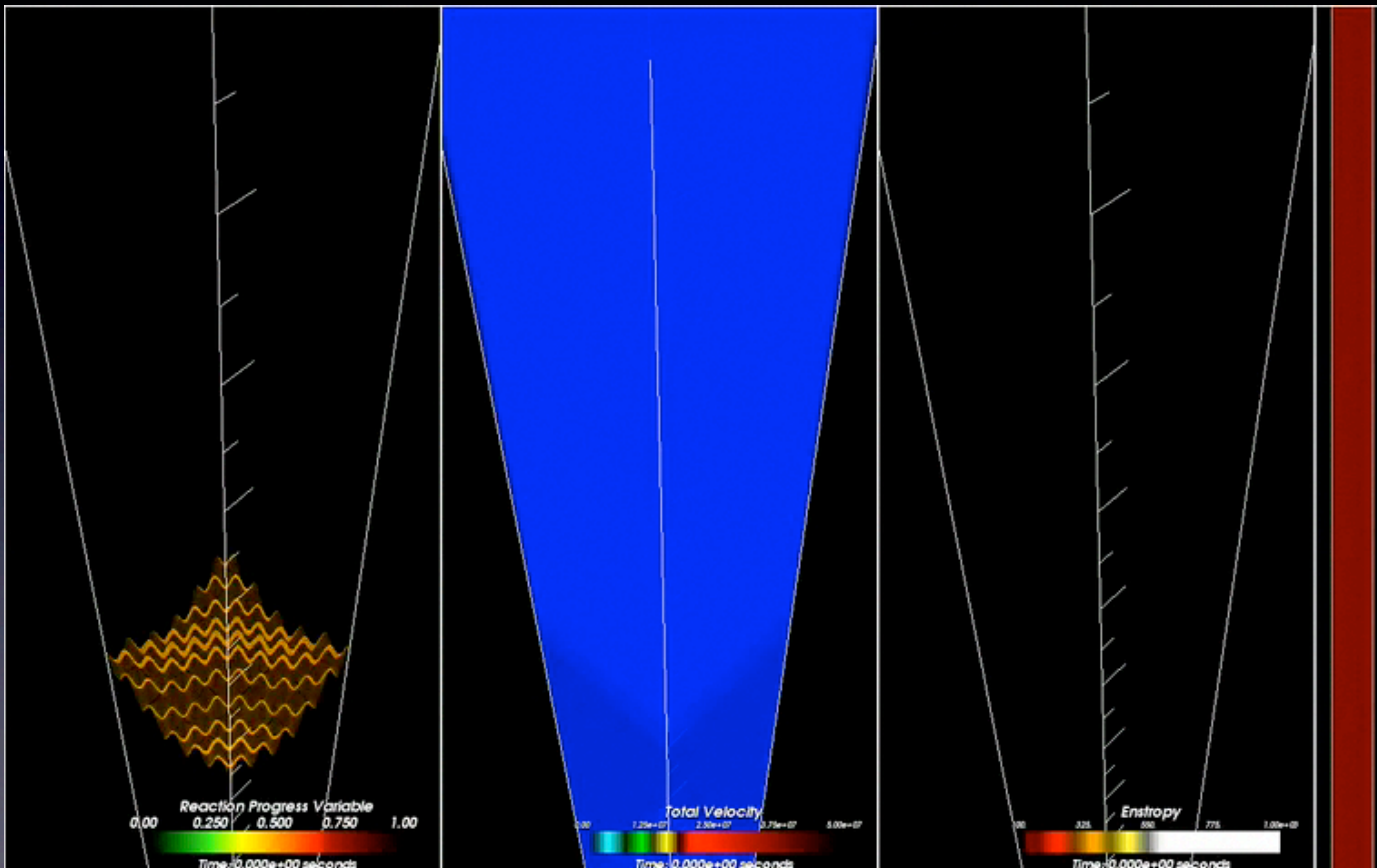
- Ongoing work targets the issue of turbulent nuclear combustion
- Simulations resolve the Gibson scale and the flame-polishing scale
- Adaptive-mesh refinement calculations using FLASH3 up to full scale of ANL BG/P *Intrepid*, ~10^5 cores and 10^5 grids

(Townsley *et al*, 2009)

Buoyancy-Driven Turbulent Nuclear Combustion

(Townsley *et al*, 2009)

Buoyancy-Driven Turbulent Nuclear Combustion



(Townsley *et al*, 2009)

Buoyancy-Driven Turbulent Nuclear Combustion

(Townsley *et al*, 2009)

Buoyancy-Driven Turbulent Nuclear Combustion

- Calculations demonstrated the feasibility of pursuing *local* AMR simulations up to the petascale

(Townsley *et al*, 2009)

Buoyancy-Driven Turbulent Nuclear Combustion

- Calculations demonstrated the feasibility of pursuing *local* AMR simulations up to the petascale

(Townsley *et al*, 2009)

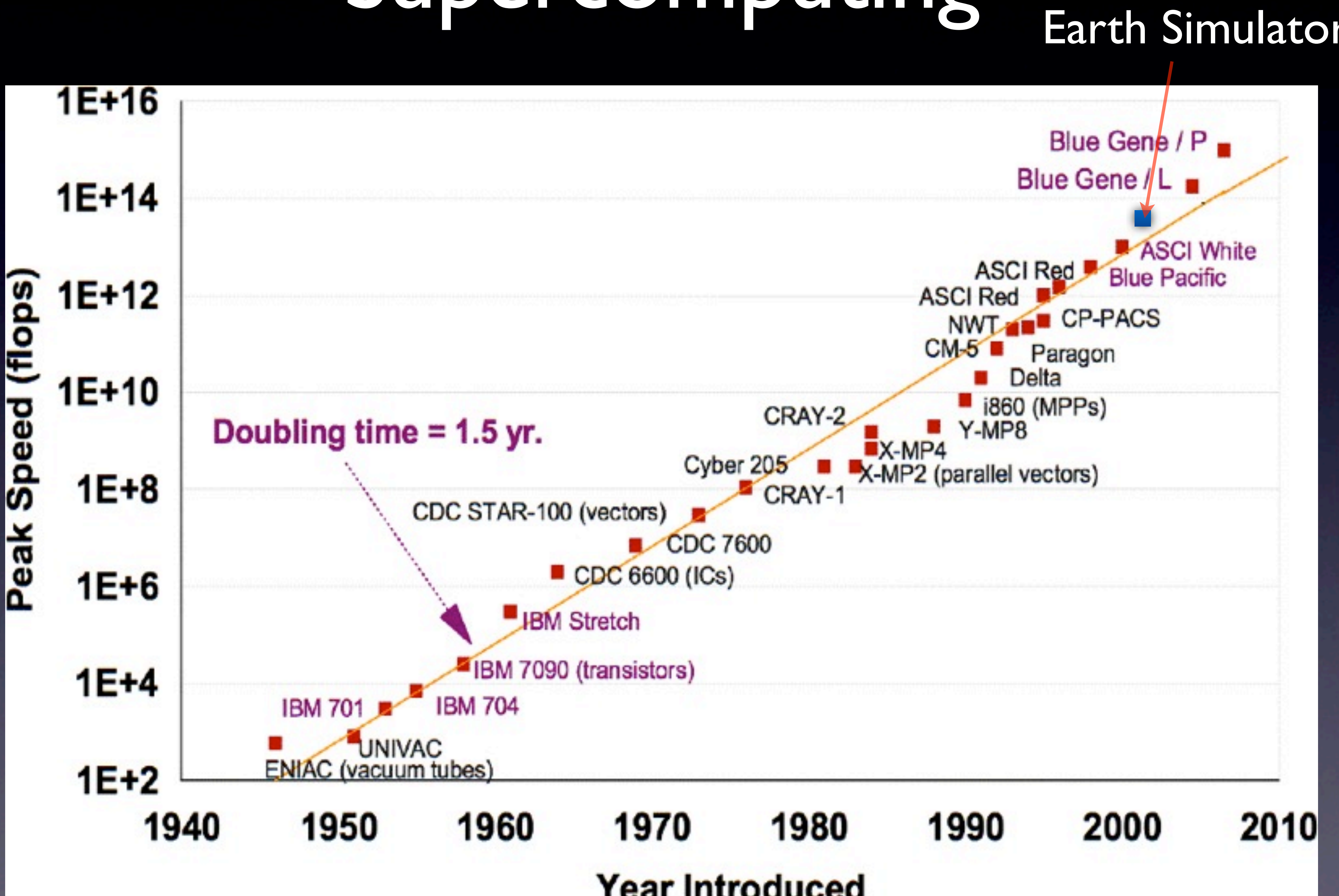
Buoyancy-Driven Turbulent Nuclear Combustion

- Calculations demonstrated the feasibility of pursuing *local* AMR simulations up to the petascale
- Lessons learned from this project can help inform progress to exascale and beyond

(Townsley *et al*, 2009)

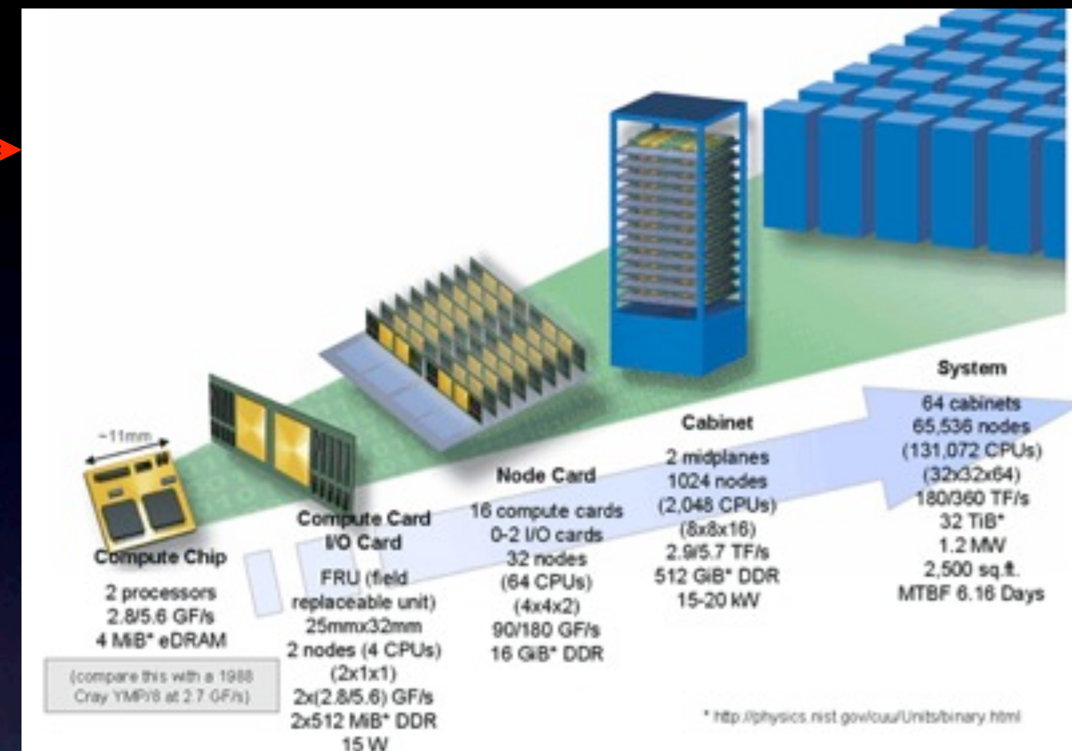
II. Hardware, Algorithms, and Asymptotically-Large Simulations

A Brief History of Supercomputing



Blue Gene Series

- BG/L, 2004
  - 2 Cores/node
  - 700 MHz/core, 512 MB/core
- BG/P, 2007
  - 4 Cores/Node
  - 850 MHz/core, 1 GB/core
- BG/Q, 2011
  - 17 Cores/node
  - 1.6 GHz/Core, 1 GB/core



Theory of Ideal, Asymptotically-Large, Explicit Simulations

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Ideal -

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Ideal -
  - Perfect load balance

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Ideal -
  - Perfect load balance
  - Perfect scalability

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Ideal -
  - Perfect load balance
  - Perfect scalability
  - Infinite memory bandwidth (no memory wall!)

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Ideal -
  - Perfect load balance
  - Perfect scalability
  - Infinite memory bandwidth (no memory wall!)
  - Neglect cost and power considerations

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Ideal -
  - Perfect load balance
  - Perfect scalability
  - Infinite memory bandwidth (no memory wall!)
  - Neglect cost and power considerations
- Explicit -

Theory of Ideal, Asymptotically-Large, Explicit Simulations

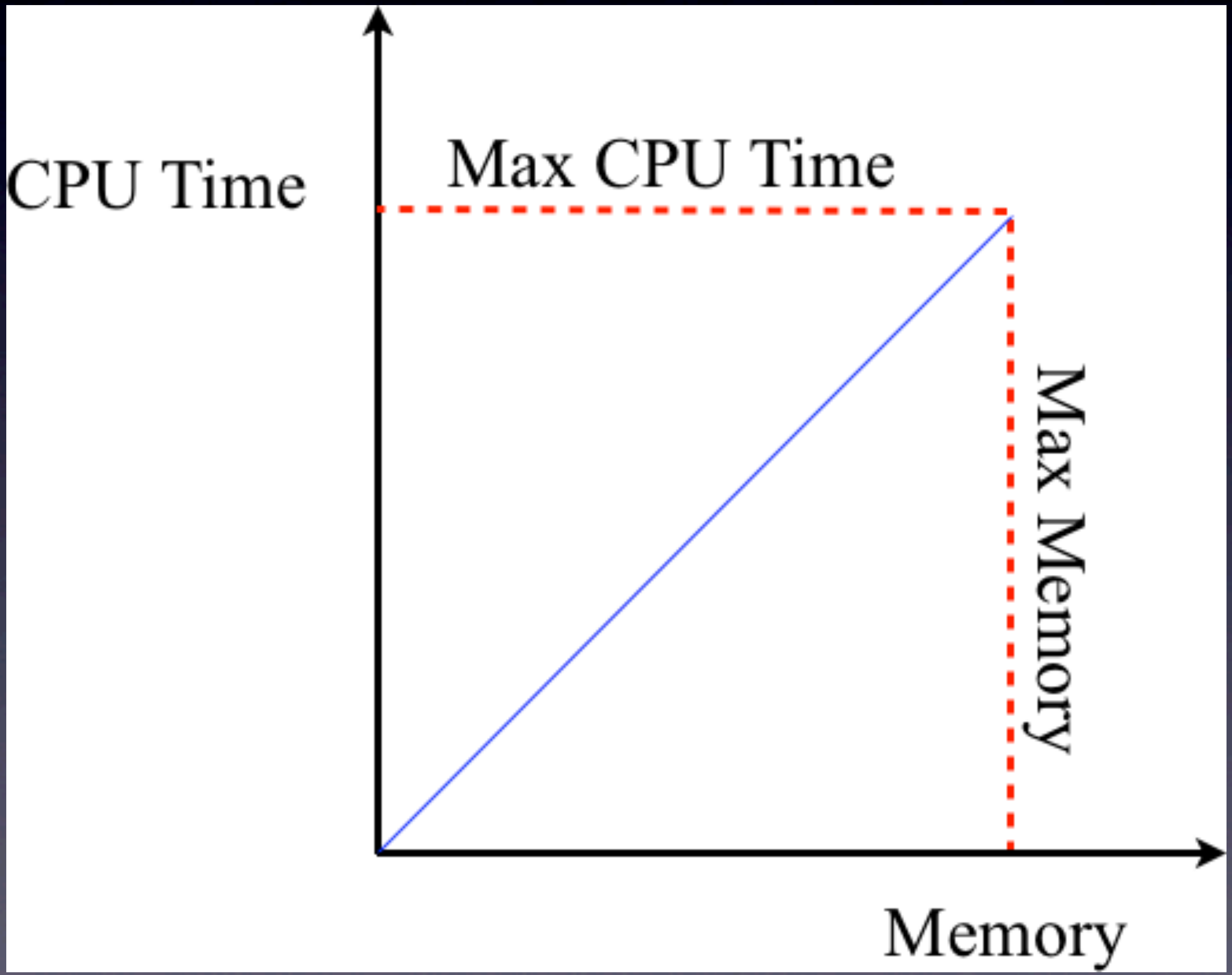
- Ideal -
  - Perfect load balance
  - Perfect scalability
  - Infinite memory bandwidth (no memory wall!)
  - Neglect cost and power considerations
- Explicit -
  - Timestep limited by CFL Condition

Theory of Ideal, Asymptotically-Large, Explicit Simulations

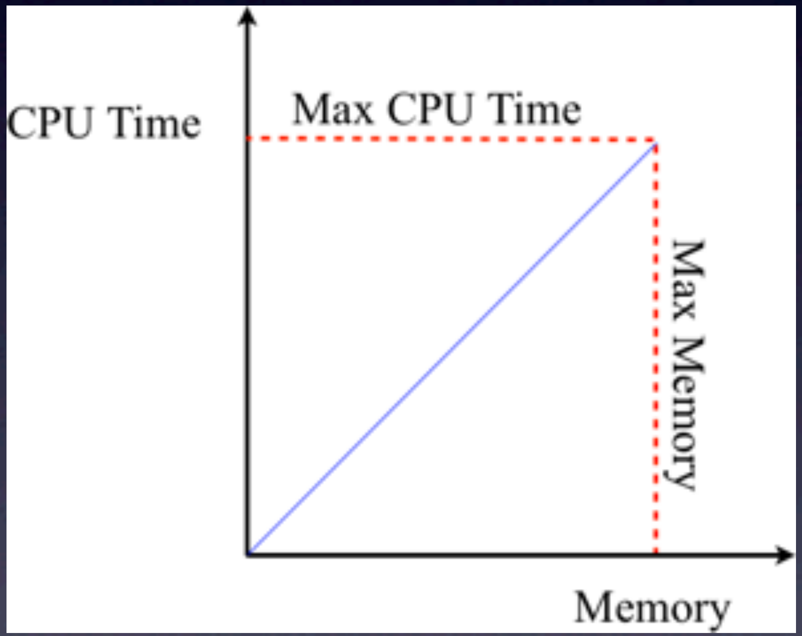
- Ideal -
  - Perfect load balance
  - Perfect scalability
  - Infinite memory bandwidth (no memory wall!)
  - Neglect cost and power considerations
- Explicit -
  - Timestep limited by CFL Condition
- Idealized assumptions allow us to focus on deep limits to scalability and strategies to address these

Theory of Ideal, Asymptotically-Large, Explicit Simulations

Theory of Ideal, Asymptotically-Large, Explicit Simulations

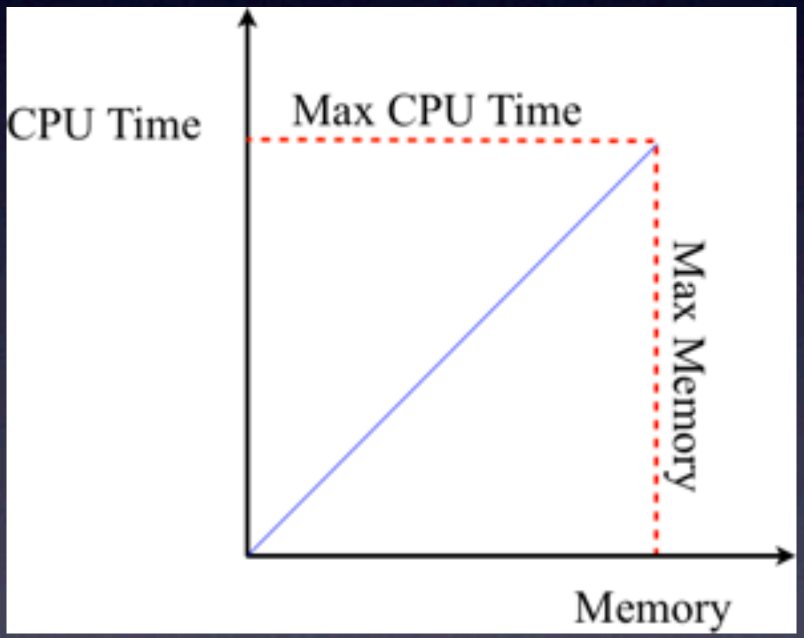


Theory of Ideal, Asymptotically-Large, Explicit Simulations



Theory of Ideal, Asymptotically-Large, Explicit Simulations

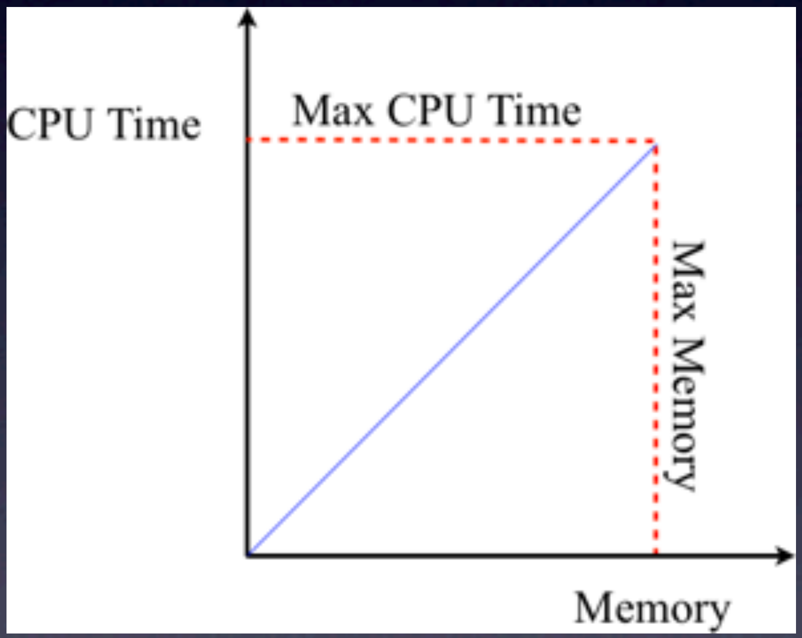
- First consider scaling behavior of a *serial, explicit, 3-D, uniform Eulerian code* with N^3 cells :



Theory of Ideal, Asymptotically-Large, Explicit Simulations

- First consider scaling behavior of a *serial, explicit, 3-D, uniform Eulerian code* with N^3 cells :

$$\text{CPU Time} = \frac{\chi_{\text{CPU}} N^4 N_{\text{dyn}}}{C}$$

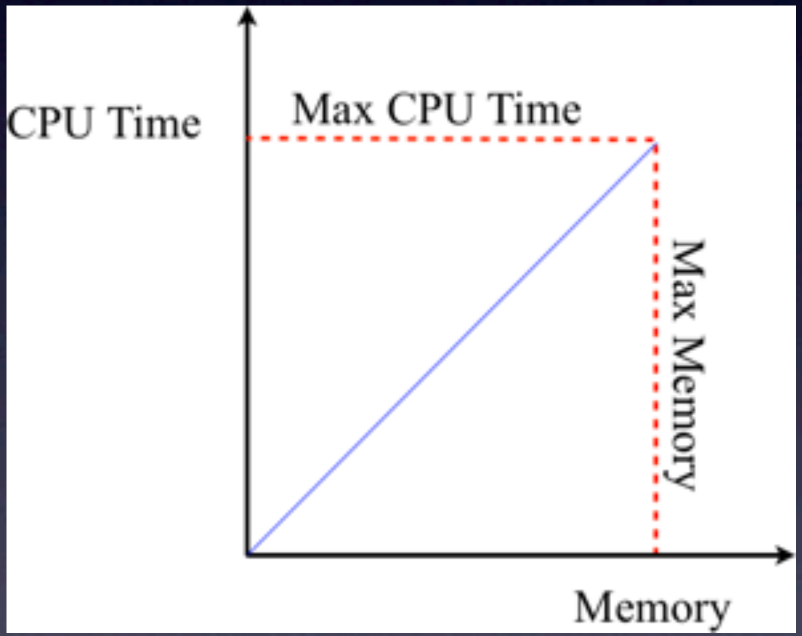


Theory of Ideal, Asymptotically-Large, Explicit Simulations

- First consider scaling behavior of a *serial, explicit, 3-D, uniform Eulerian code* with N^3 cells :

$$\text{CPU Time} = \frac{\chi_{\text{CPU}} N^4 N_{\text{dyn}}}{C}$$

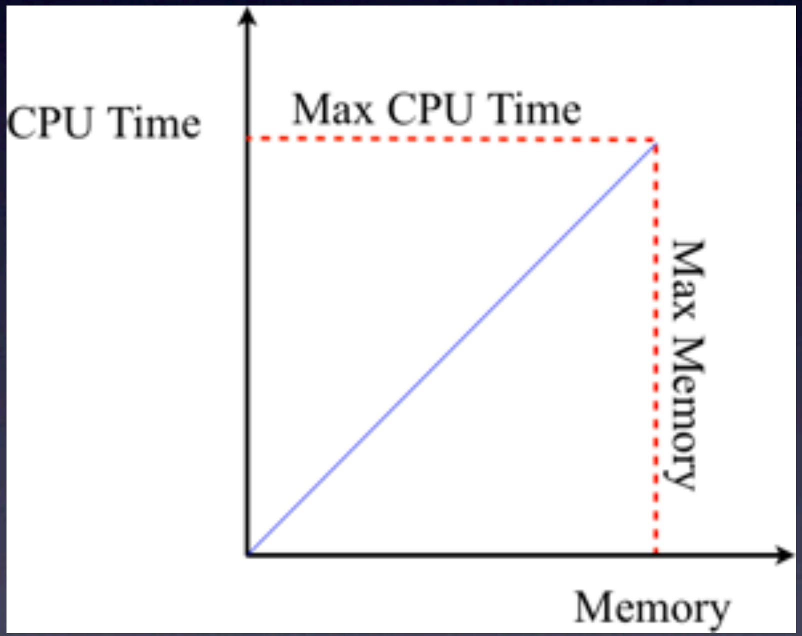
$$\text{Memory} = \chi_{\text{mem}} N^3$$



Theory of Ideal, Asymptotically-Large, Explicit Simulations

- First consider scaling behavior of a *serial, explicit, 3-D, uniform Eulerian code* with *N*^{3} cells :

$$\text{CPU Time} = \frac{\chi_{\text{CPU}} N^4 N_{\text{dyn}}}{C}$$



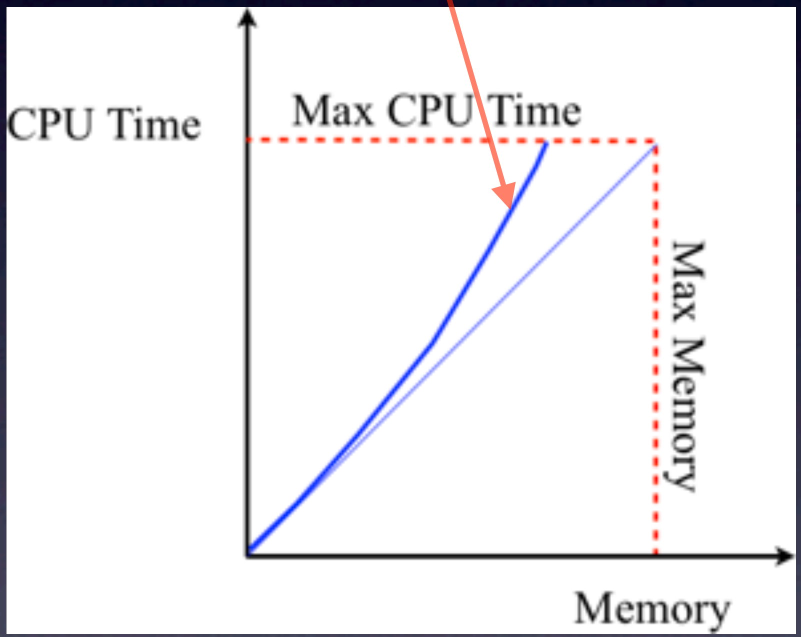
$$\text{Memory} = \chi_{\text{mem}} N^3$$

$$\text{CPU Time} = \frac{\chi_{\text{CPU}}}{\chi_{\text{mem}}^{4/3} C} N_{\text{dyn}} \text{Memory}^{4/3}$$

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- First consider scaling behavior of a *serial, explicit, 3-D, uniform Eulerian code* with *N^3* cells :

$$\text{CPU Time} \propto \text{Memory}^{4/3}$$



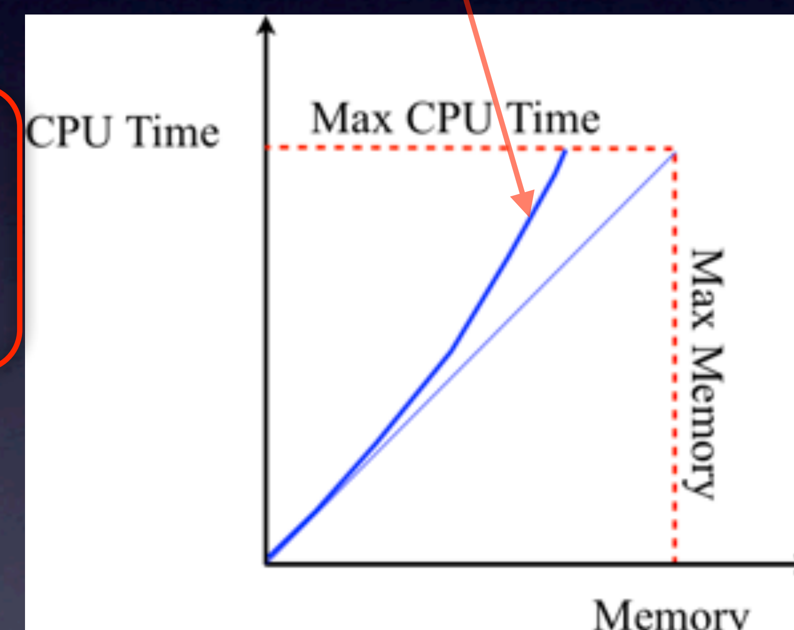
$$\text{CPU Time} = \frac{\chi_{\text{CPU}} N^4 N_{\text{dyn}}}{C}$$

$$\text{Memory} = \chi_{\text{mem}} N^3$$

$$\text{CPU Time} = \frac{\chi_{\text{CPU}}}{\chi_{\text{mem}}^{4/3} C} N_{\text{dyn}} \text{Memory}^{4/3}$$

Theory of Ideal, Asymptotically-Large, Explicit Simulations

$$\text{CPU Time} \propto \text{Memory}^{4/3}$$



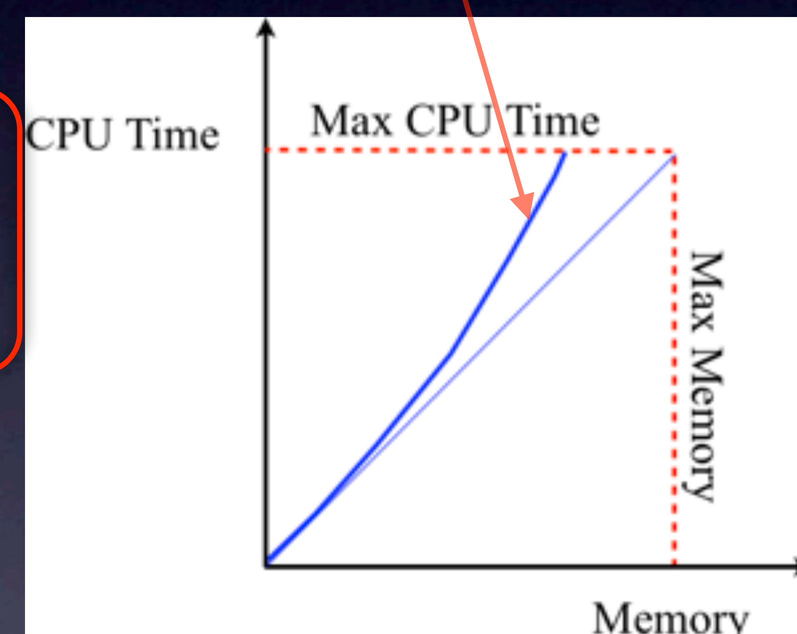
$$\text{Max Memory} < \frac{\chi_{\text{mem}} C^{3/4}}{\chi_{\text{CPU}}^{3/4}} \left(\frac{\text{Max CPU Time}}{N_{\text{dyn}}} \right)^{3/4}$$

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Given maximum memory and CPU time bounds, a *serial* simulation is memory-bound *if*

$$\text{CPU Time} \propto \text{Memory}^{4/3}$$

$$\text{Max Memory} < \frac{\chi_{\text{mem}} C^{3/4}}{\chi_{\text{CPU}}^{3/4}} \left(\frac{\text{Max CPU Time}}{N_{\text{dyn}}} \right)^{3/4}$$

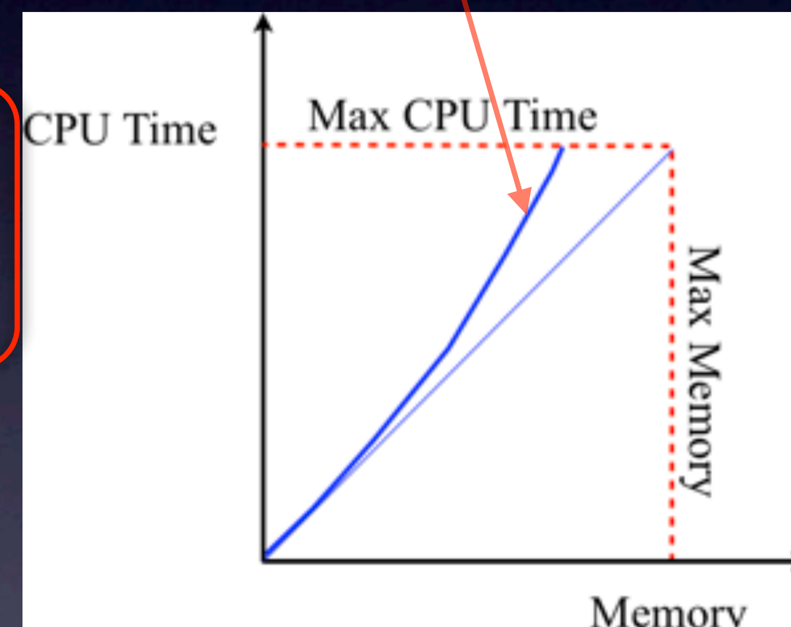


Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Given maximum memory and CPU time bounds, a *serial* simulation is memory-bound *if*

$$\text{CPU Time} \propto \text{Memory}^{4/3}$$

$$\text{Max Memory} < \frac{\chi_{\text{mem}} C^{3/4}}{\chi_{\text{CPU}}^{3/4}} \left(\frac{\text{Max CPU Time}}{N_{\text{dyn}}} \right)^{3/4}$$



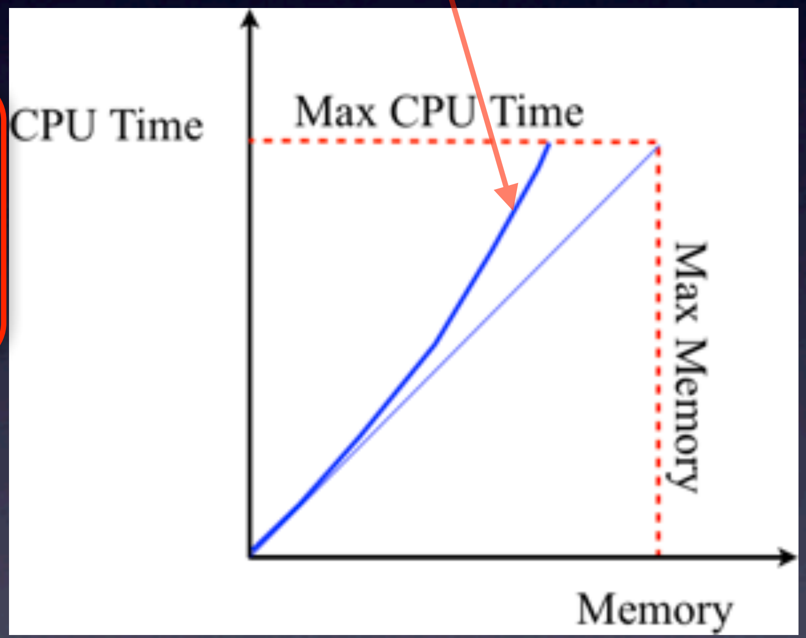
- In *parallel*, an *ideal* simulation has

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Given maximum memory and CPU time bounds, a *serial* simulation is memory-bound *if*

$$\text{CPU Time} \propto \text{Memory}^{4/3}$$

$$\text{Max Memory} < \frac{\chi_{\text{mem}} C^{3/4}}{\chi_{\text{CPU}}^{3/4}} \left(\frac{\text{Max CPU Time}}{N_{\text{dyn}}} \right)^{3/4}$$



- In *parallel*, an *ideal* simulation has

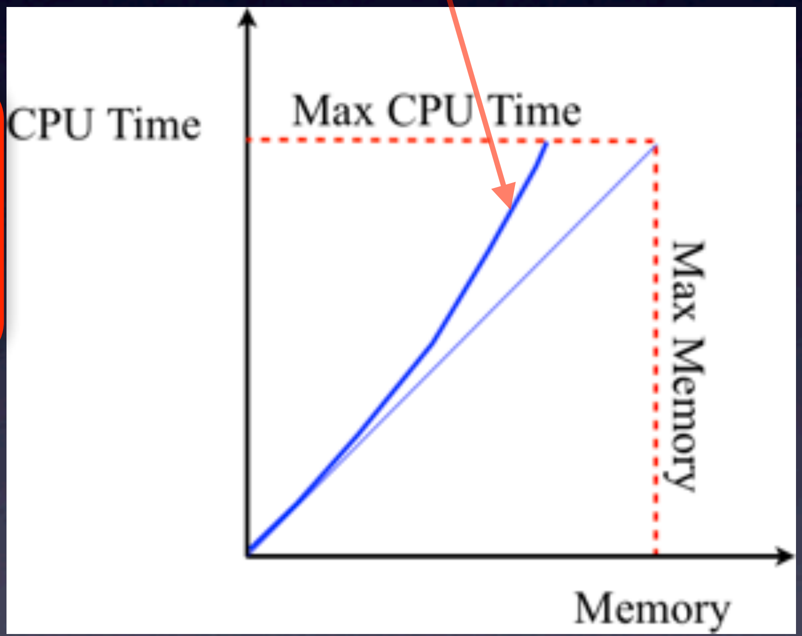
$$\text{Max Memory} = \text{Memory / CPU } N_{\text{CPU}}$$

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Given maximum memory and CPU time bounds, a *serial* simulation is memory-bound *if*

$$\text{CPU Time} \propto \text{Memory}^{4/3}$$

$$\text{Max Memory} < \frac{\chi_{\text{mem}} C^{3/4}}{\chi_{\text{CPU}}^{3/4}} \left(\frac{\text{Max CPU Time}}{N_{\text{dyn}}} \right)^{3/4}$$



- In *parallel*, an *ideal* simulation has

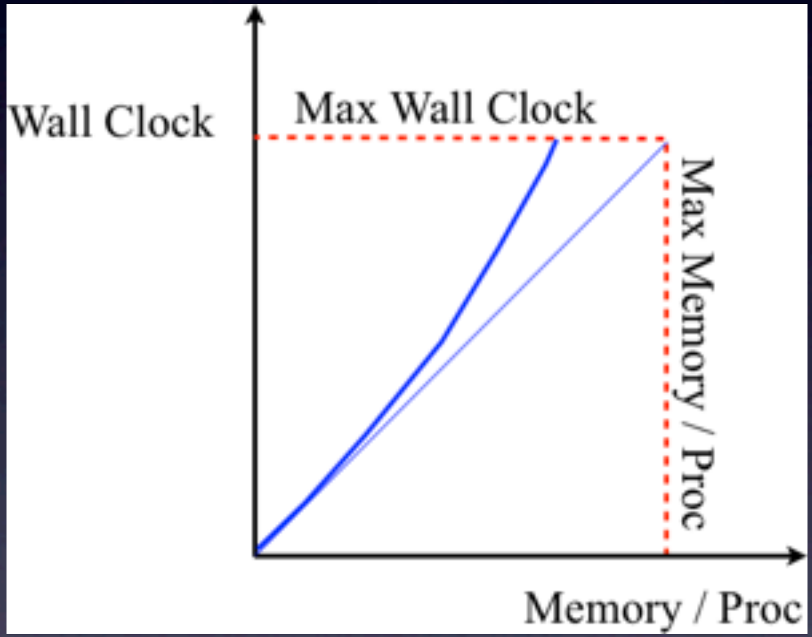
$$\text{Max Memory} = \text{Memory / CPU } N_{\text{CPU}}$$

$$\text{Max CPU Time} = \text{Max Wall Clock } N_{\text{CPU}}$$

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- The memory-boundedness criterion for a *parallel* simulation becomes

$$\text{Memory/CPU} < \chi_{\text{mem}} \left[\left(\frac{C \text{ Max Wall Clock}}{\chi_{\text{CPU}} N_{\text{dyn}}} \right)^3 \frac{1}{N_{\text{CPU}}} \right]^{1/4}$$



- Scaling to typical values on a small cluster,

$$\text{Memory/CPU} < 0.2 \text{ GB} \left[\left(\frac{(N_{\text{state}}/10)(C/0.5) \text{ (Max Wall Clock/1wk)}}{(\chi_{\text{CPU}}/10 \ \mu s)(N_{\text{dyn}}/10)} \right)^3 \frac{512}{N_{\text{CPU}}} \right]^{1/4}$$

- Asymptotically-large, explicit simulations ($N_{\text{CPU}} \rightarrow \infty$) are *always* CPU-bound.

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- Consider an ideal AMR simulation with of a *total* N_{blocks} of N_{grid}^3 cells

$$\text{Wall Clock} = \left(\frac{\chi_{\text{CPU}} N_{\text{dyn}}}{C N_{\text{CPU}}}\right) N^4$$

- *Fixing the wall clock time barrier,*

$$N \propto N_{\text{CPU}}^{1/4}$$

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- The distribution of blocks over cores, *fixing the wall clock time barrier and grind time*,

$$\frac{N_{\rm blocks}}{N_{\rm CPU}} = \frac{1}{N_{\rm grid}^3}\left[\frac{C(\text{Wall Clock})}{\chi_{\rm CPU}\mathrm{N}_{\rm dyn}}\right]^{3/4} N_{\rm CPU}^{-1/4}$$

$$\frac{N_{\rm blocks}}{N_{\rm CPU}} = 12\left(\frac{32}{N_{\rm grid}}\right)^3 \left[\frac{(C/0.5)(\text{Wall Clock}/1\text{ wk})}{(\chi_{\rm CPU}/10\ \mu\text{s})(\mathrm{N}_{\rm dyn}/10)}\right]^{3/4}\left(\frac{N_{\rm CPU}}{10^6}\right)^{-1/4}$$

Theory of Ideal, Asymptotically-Large, Explicit Simulations

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- As we go beyond the petascale, AMR simulations will face increasing tight load-balancing issues.

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- As we go beyond the petascale, AMR simulations will face increasing tight load-balancing issues.
- Possible strategies :

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- As we go beyond the petascale, AMR simulations will face increasing tight load-balancing issues.
- Possible strategies :
  - Multithreading

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- As we go beyond the petascale, AMR simulations will face increasing tight load-balancing issues.
- Possible strategies :
  - Multithreading
  - Smaller block sizes

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- As we go beyond the petascale, AMR simulations will face increasing tight load-balancing issues.
- Possible strategies :
  - Multithreading
  - Smaller block sizes
  - Improved load-balancing algorithms

Theory of Ideal, Asymptotically-Large, Explicit Simulations

- As we go beyond the petascale, AMR simulations will face increasing tight load-balancing issues.
- Possible strategies :
  - Multithreading
  - Smaller block sizes
  - Improved load-balancing algorithms
  - Faster grind times through GPU or other technologies

Conclusions

Conclusions

- Continued success for computational astrophysics at scale will hinge upon our ability as a community to
  - Think deeply about modeling of turbulence in ways not yet manifested in existing codes
  - Think deeply about the ultimate limits to scalability and beginning to take long-term strategic directions to address these