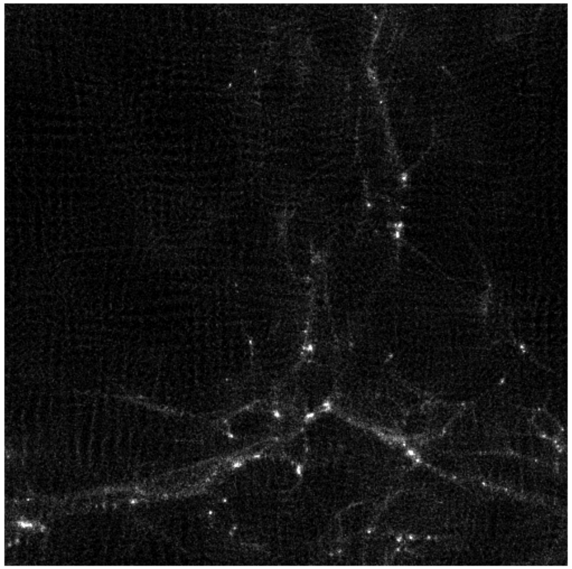
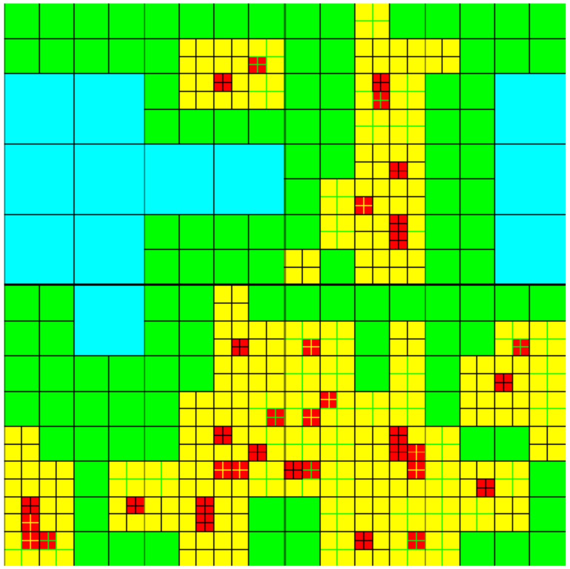




# A Scalable AMR Gravity Solver for ENZO-E (Extreme-scale ENZO)

Michael L Norman and James Bordner

San Diego Supercomputer Center

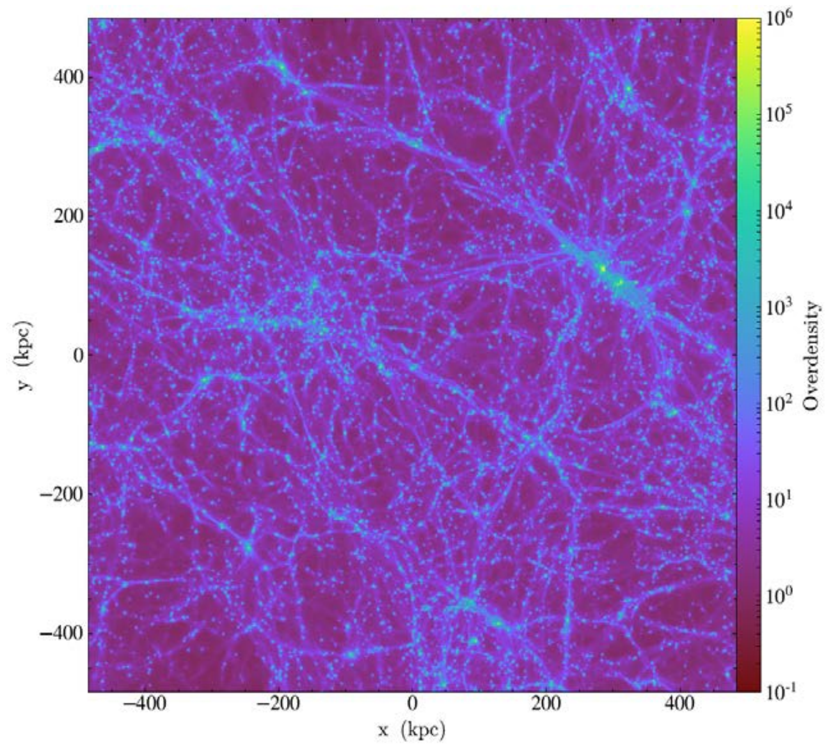
University of California, San Diego

*Supported by NSF grants CSSI-1835402, SI2-SSE-1440709, PHY-1104819, PRAC-1516003, PRAC-1810774*

# Science Motivation: Large-scale simulations of the IGM including galaxy feedback

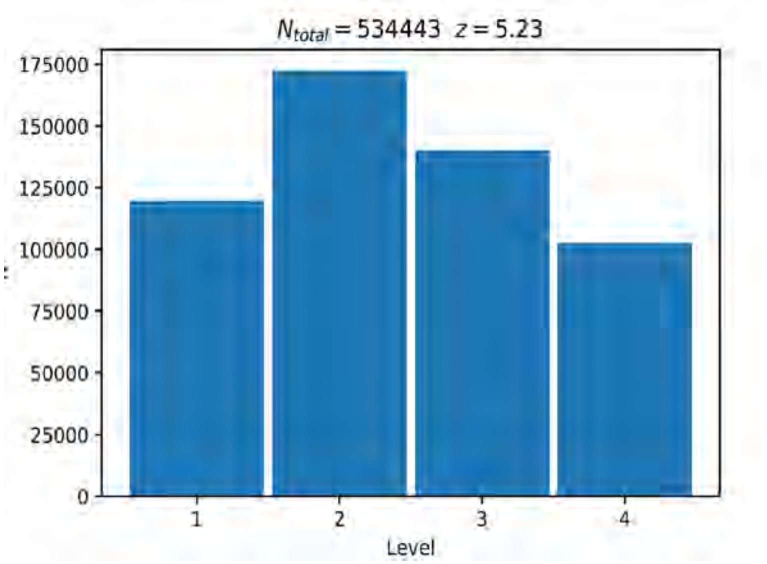
- Precision comparisons between models and simulations require large volumes and high resolution in galaxies
- Standard ENZO code struggles to do this due to limited scalability of its AMR implementation ($P<1000$)
- In 2011, James Bordner and I embarked on a from-scratch redesign and reimplementation of ENZO capable of scaling AMR to millions of cores.
- Blue Waters has been instrumental in the development and testing of ENZO-E

# **PRAC**: Realistic Simulations of the Intergalactic Medium: The Search for Missing Physics-Part 2




Projection of baryon density in a section of a $1024^3$, 25.6 Mpc box with 4 additional levels of refinement.
8K cores, Blue Waters

←ENZO simulation




#AMR grids versus level of refinement
Potential level of concurrency: >100,000

# Adopted Strategy

- Keep the best part of Enzo (numerical solvers) and replace the AMR infrastructure
- Implement using modern OOP best practices for modularity and extensibility
- Use the best available scalable AMR algorithm:
  - Array-of-Octrees (aka Forest-of-Octrees)
- Move from bulk synchronous to data-driven asynchronous execution model to support patch adaptive timestepping
- Leverage parallel runtimes that support this execution model, and have a path to exascale (Charm++)
- Make AMR software library application-independent so others can use it

# Software Architecture

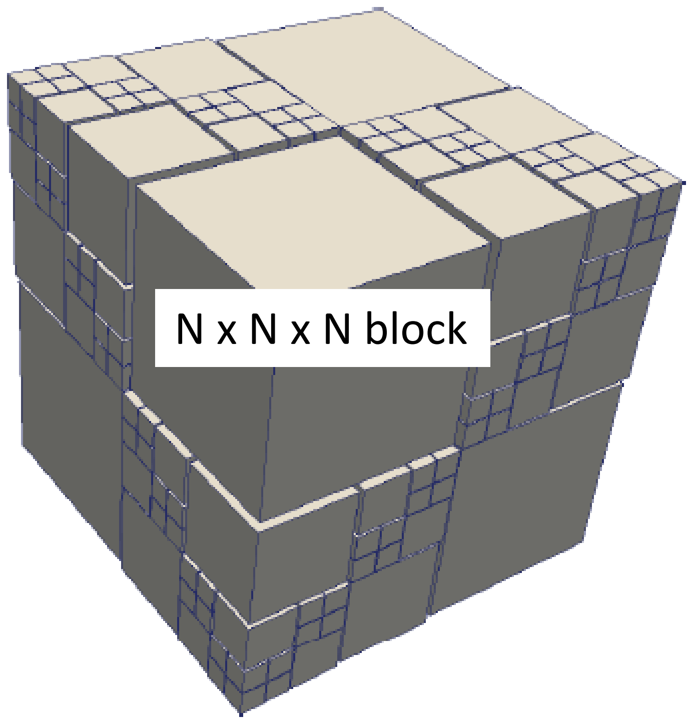
Enzo numerical solvers (Enzo-E)

Array-of-octrees AMR (Cello)

Charm++

Hardware (heterogeneous, hierarchical)

# How does Cello implement AOT?




2 x 2 x 2 array

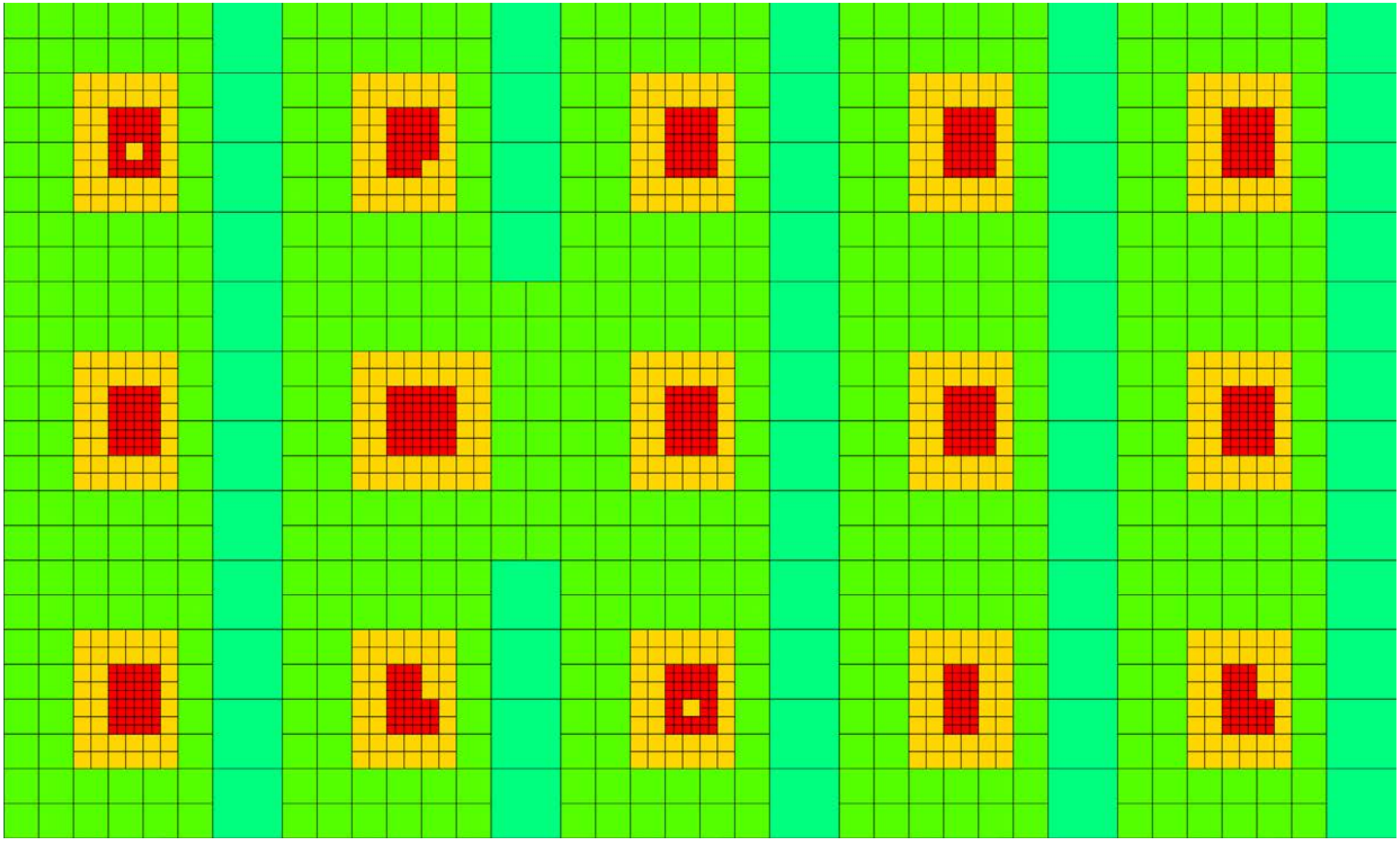
- **Array of octrees** of arbitrary size K x L x M
- An octree has leaf nodes which are **blocks** (N x N x N)
- Each block is a **chare** (unit of sequential work)
- The entire AOT is stored as a **chare array** using a bit indexing scheme
- Chare arrays are **fully distributed data structures** in Charm++

# Demonstration of ENZO-E

Interacting blast waves with PPM solver - Total energy

# Demonstration of ENZO-E

Mesh refinement level; $32^3$ blocks/chare

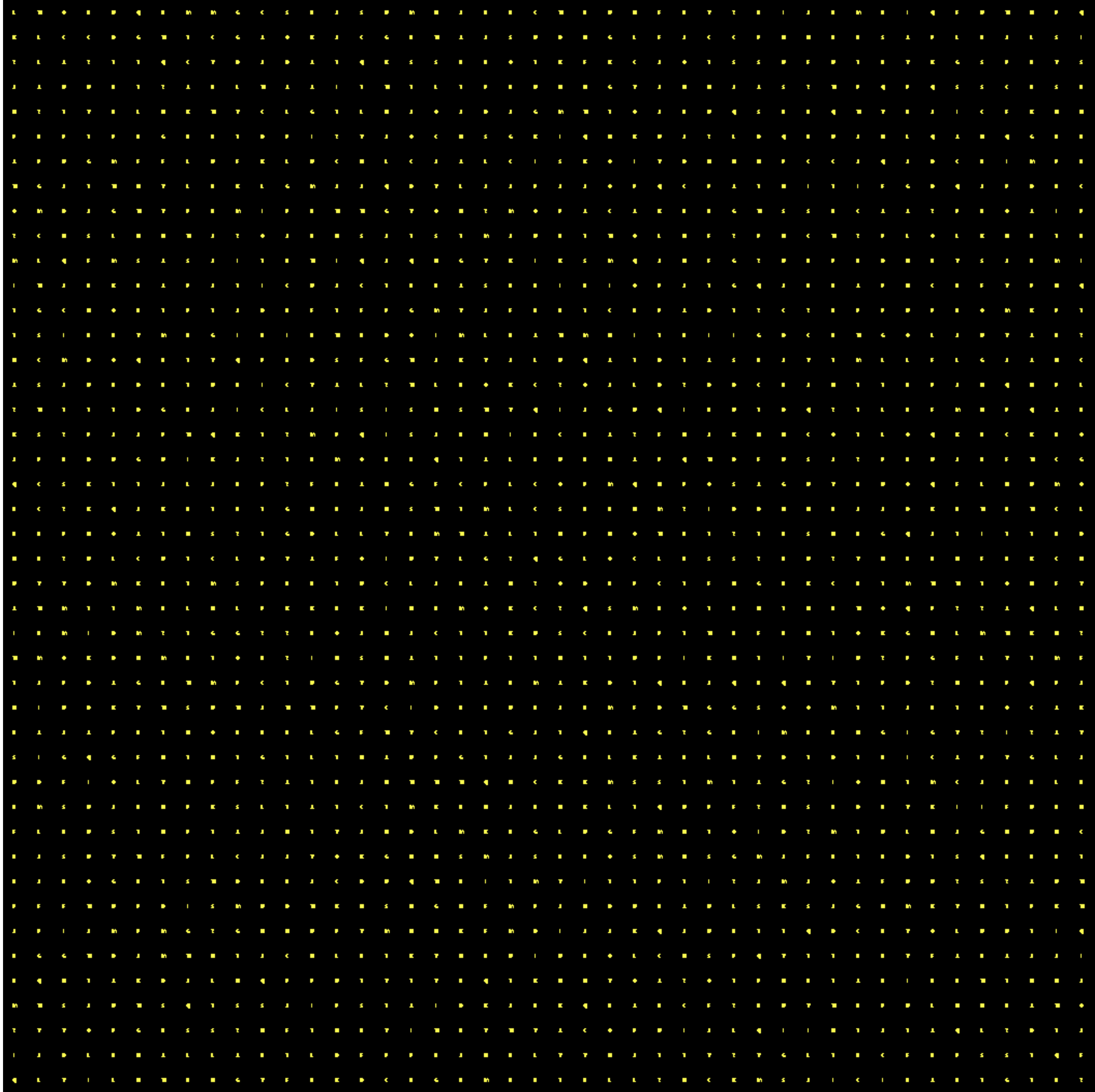


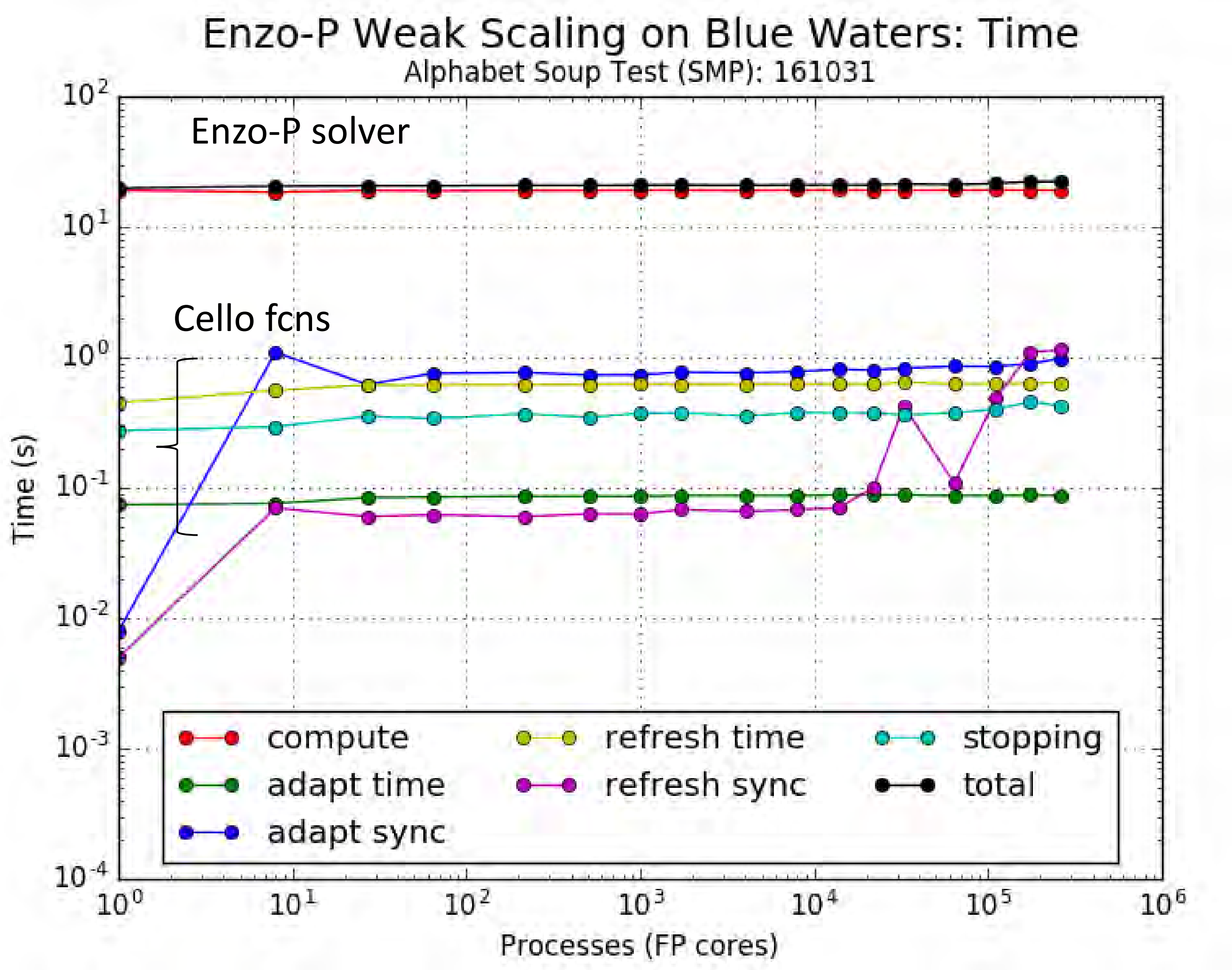
Largest AMR simulation in the world?

1.7 trillion cells

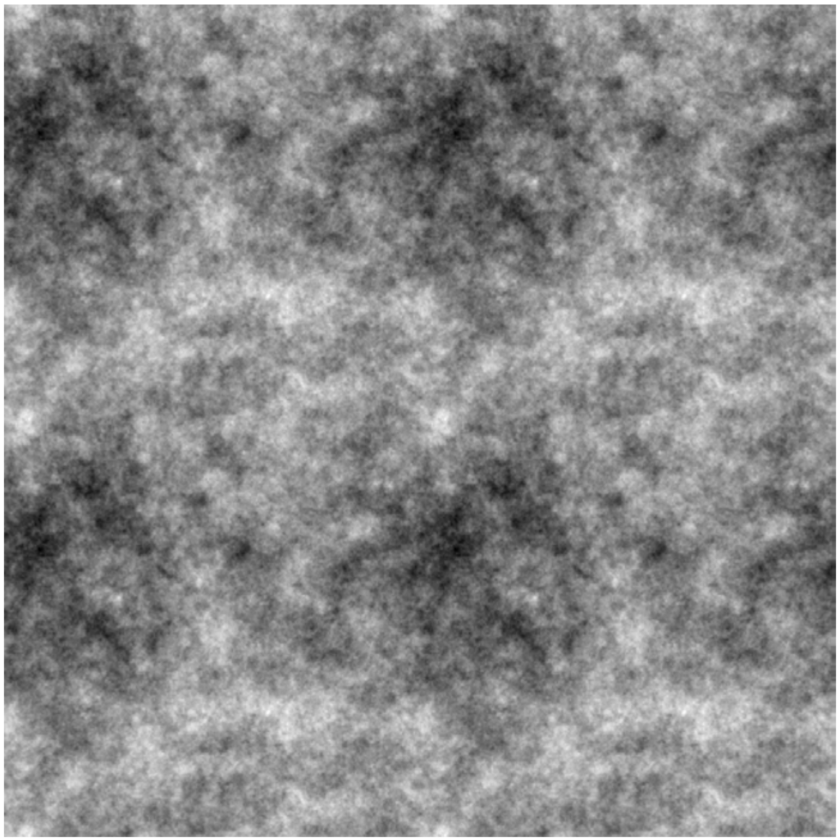
262K cores on NCSA Blue Waters

html



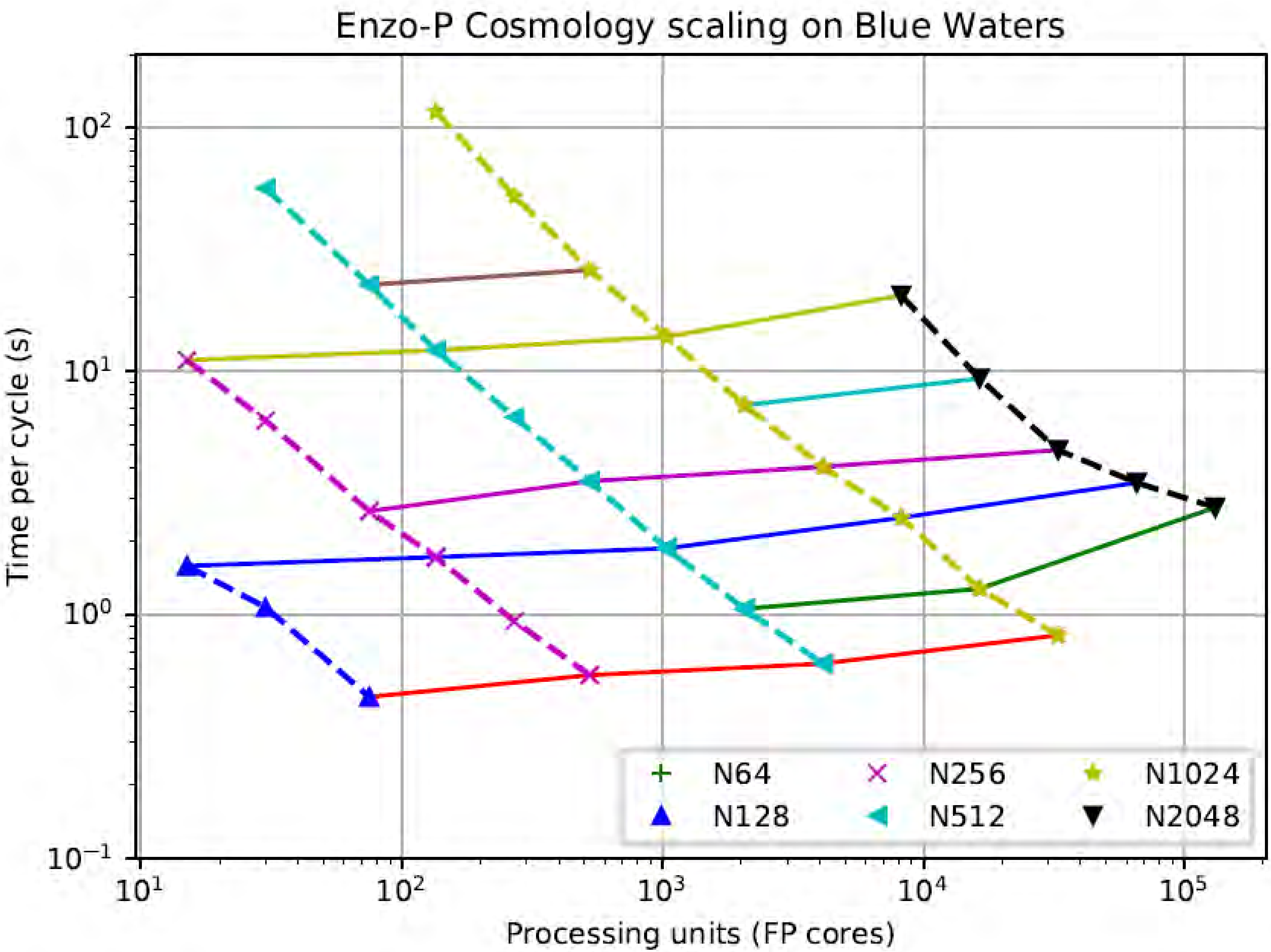

Enzo-P Weak Scaling on Blue Waters: Time
Alphabet Soup Test (SMP): 161031
Enzo-P solver
Cello fcns
Time (s)
Processes (FP cores)
compute
adapt time
adapt sync
refresh time
refresh sync
stopping
total

# Hydrodynamic Cosmology Scaling Tests

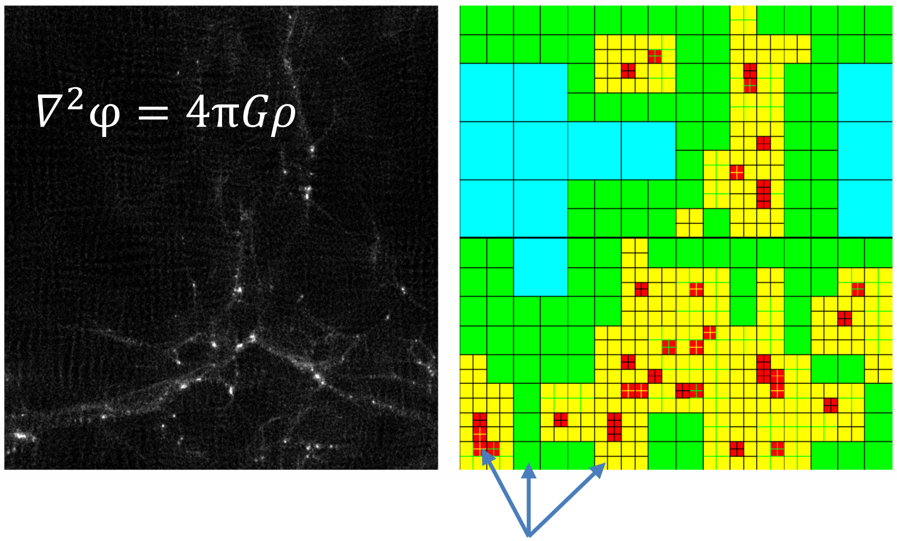


Density projection in $512^3$ simulation

- Uniform grid only
- Weak and strong scaling
- $32^3$ blocks/chare
- 1, 8, 64 chares/core
- $64^3$ to $2048^3$ meshes
- p=8 to 128k cores


Enzo-P Cosmology scaling on Blue Waters
Time per cycle (s)
Processing units (FP cores)
$10^{2}$
$10^{1}$
$10^{0}$
$10^{-1}$
$10^{1}$
$10^{2}$
$10^{3}$
$10^{4}$
$10^{5}$
N64
N128
N256
N512
N1024
N2048

# Global Multilevel AMR Poisson Solver

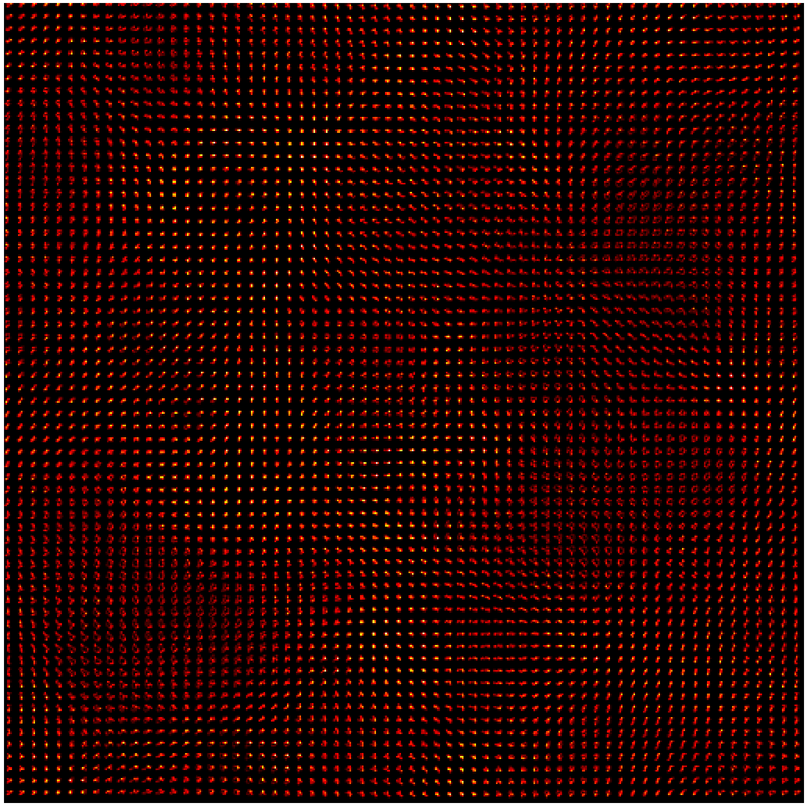


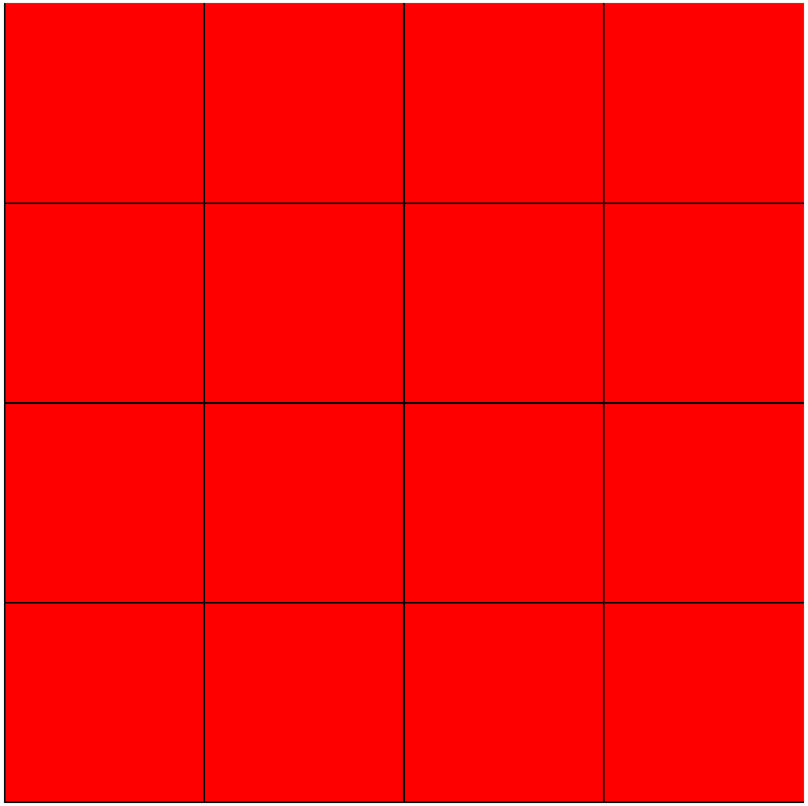

- Each square is projection of $16^3$ block

- Ax=b
  - A is non-symmetric matrix arising from discretizing Laplacian operator on multilevel mesh
  - x is gravitational potential $\phi$
  - b is matter density source term
- Algorithm
  - BiCGStab
    - Diagonally-preconditioned
    - Multigrid-preconditioned
  - Parallelize over all blocks in AOT using Charm++
- poor scalability

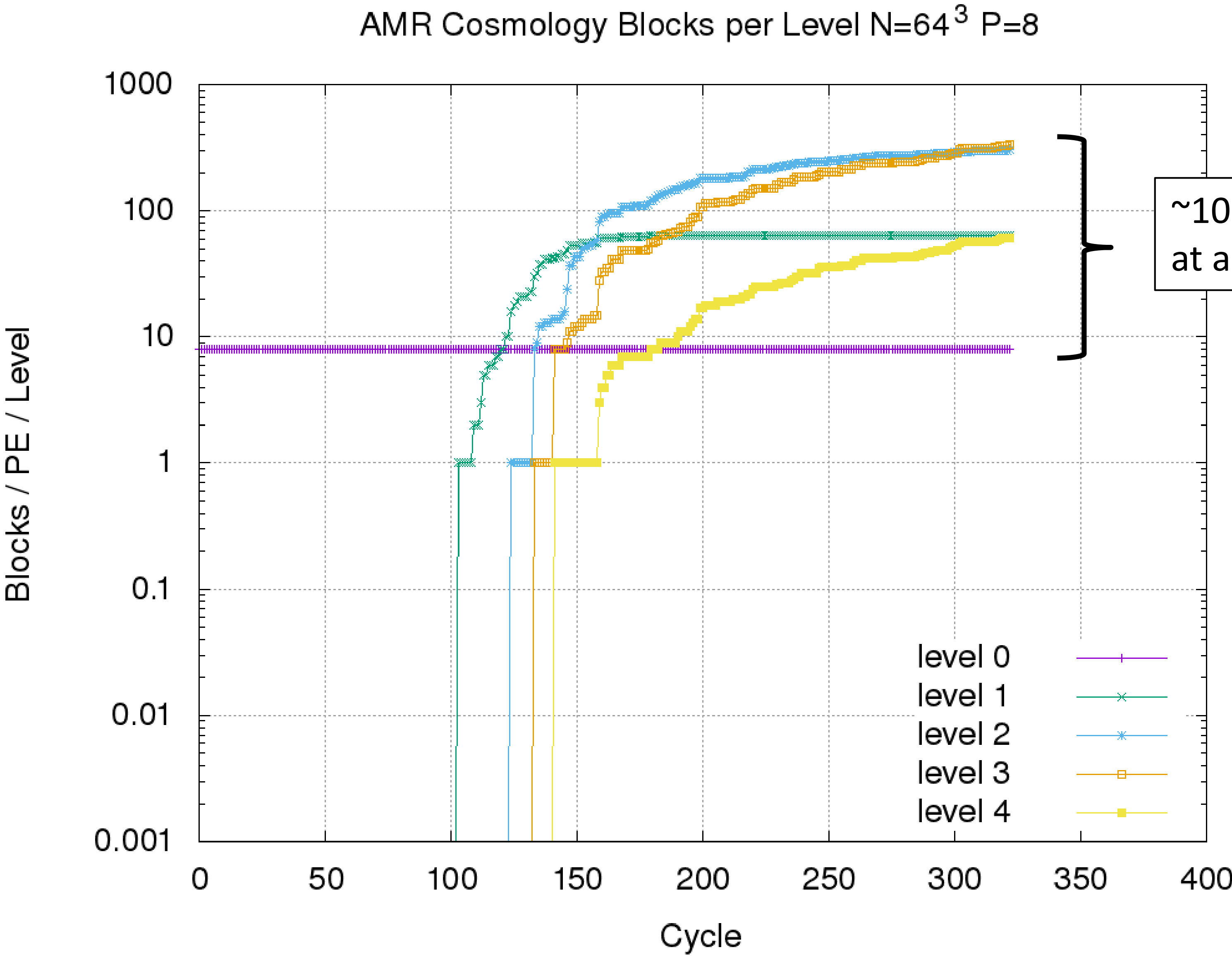
# AMR cosmology with HG* solver

$64^3$ mesh ($4^3$ array of $16^3$ blocks), 4 AMR levels ($1024^3$ eff.), PE=8



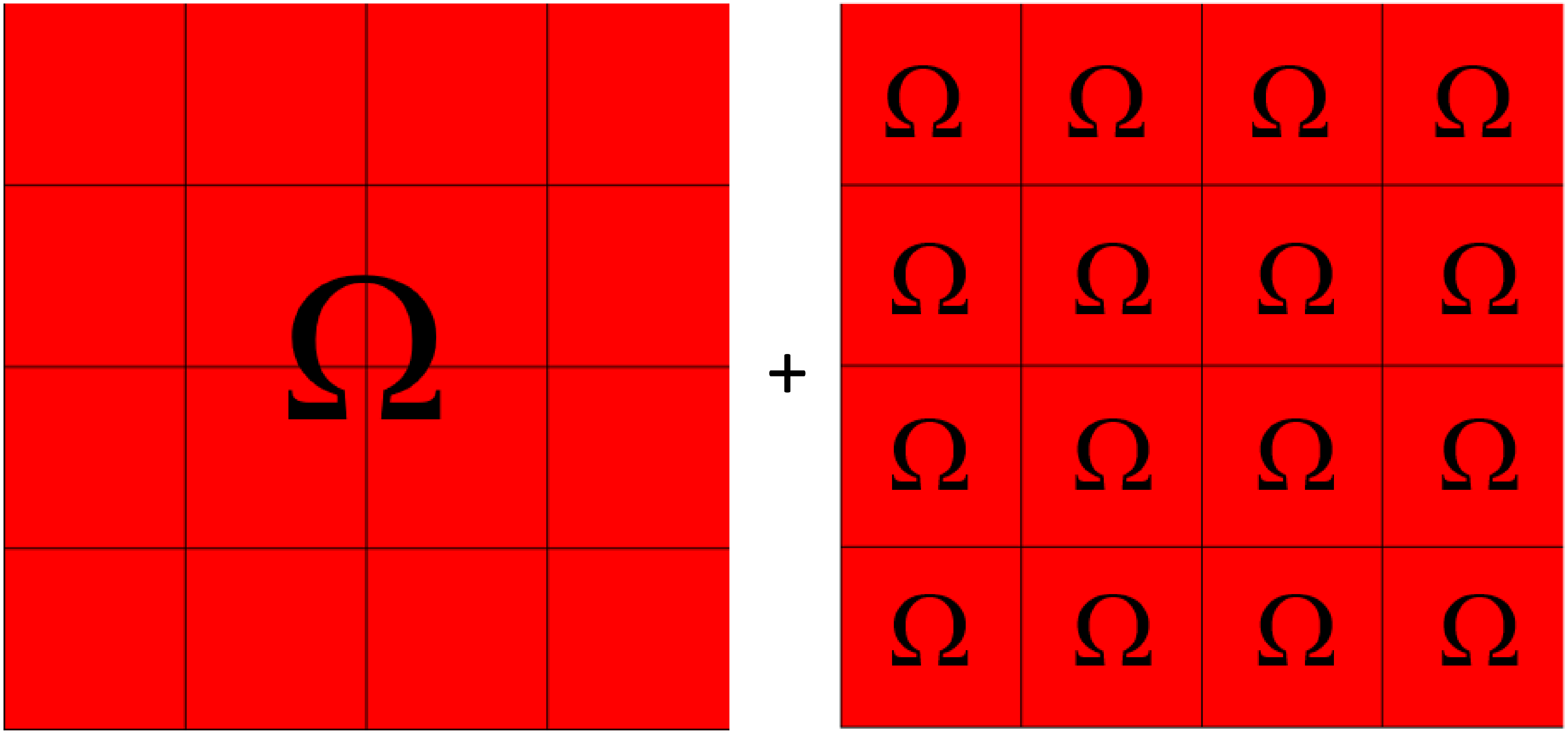


* HG = multigrid preconditioned BiCGStab



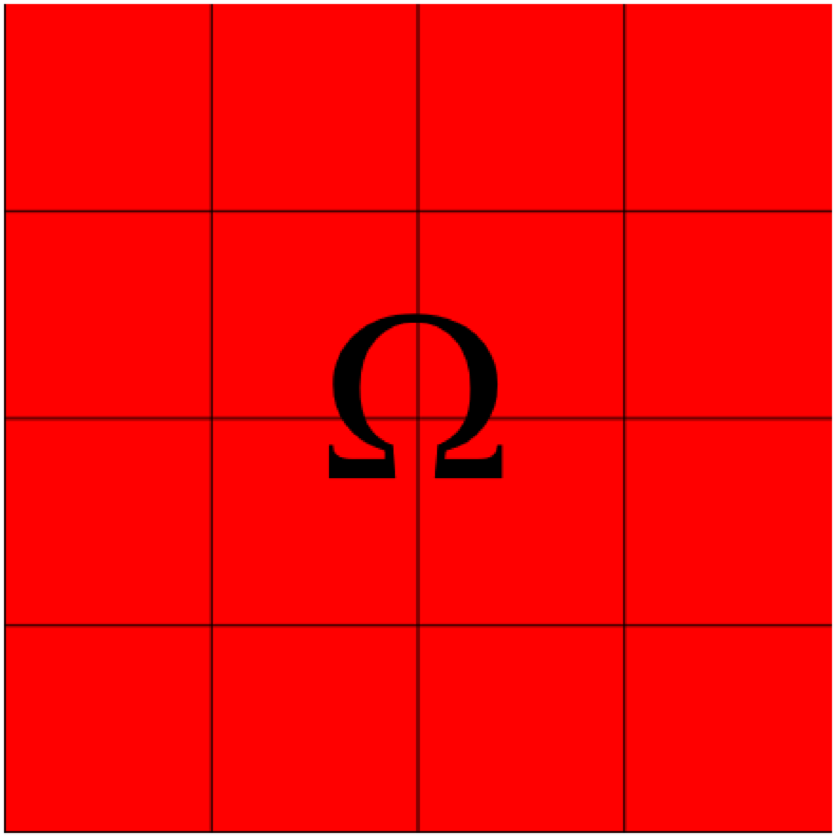

~1000 blocks/PE at all levels

# Domain-decomposed AMR Poisson Solver (DD)




Each block becomes an octree

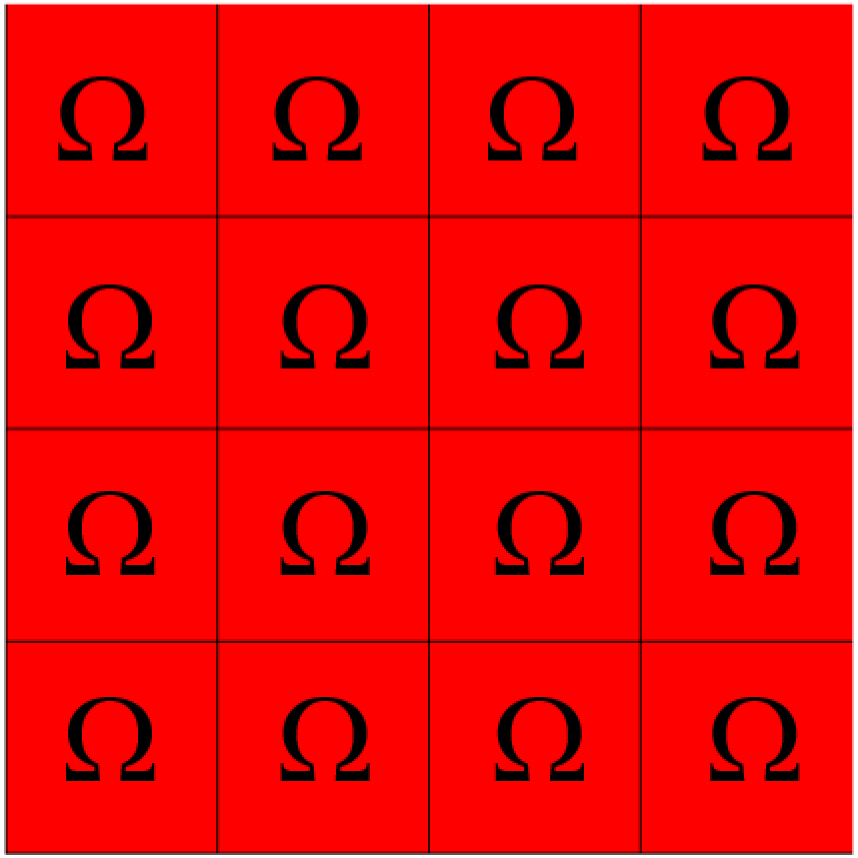
# Domain-decomposed AMR Poisson Solver (DD)




- Step 1: project density field to root grid blocks
- Step 2: global Poisson solve on root grid using multigrid solver
- Step 3: interpolate $\Phi_0$ to faces of each octree

Each block becomes an octree

# Domain-decomposed AMR Poisson Solver (DD)

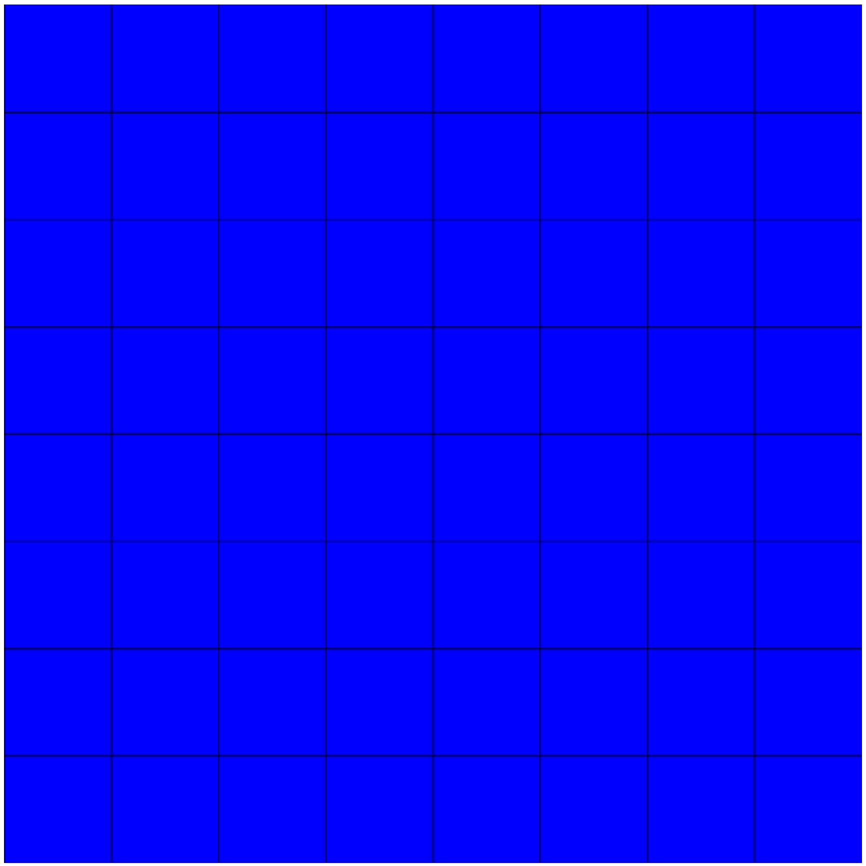


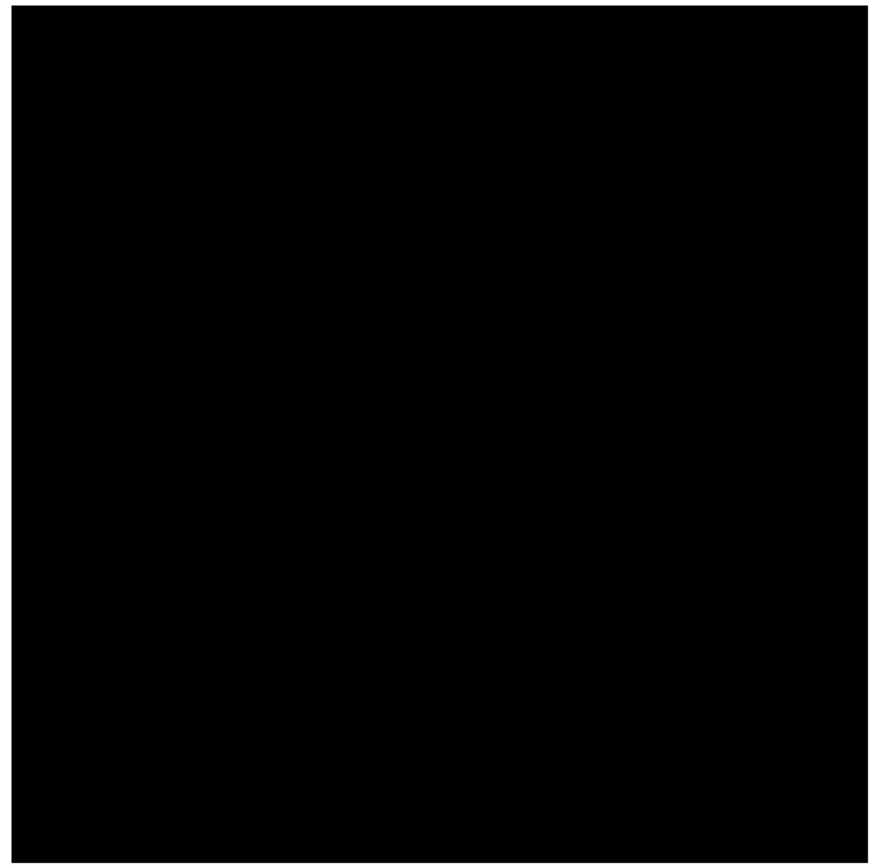

Each block becomes an octree

- Step 4: local Poisson solve on each octree using BiCGtab
- Step 5: Jacobi smooth potential $\Phi_\ell$ on leaf blocks

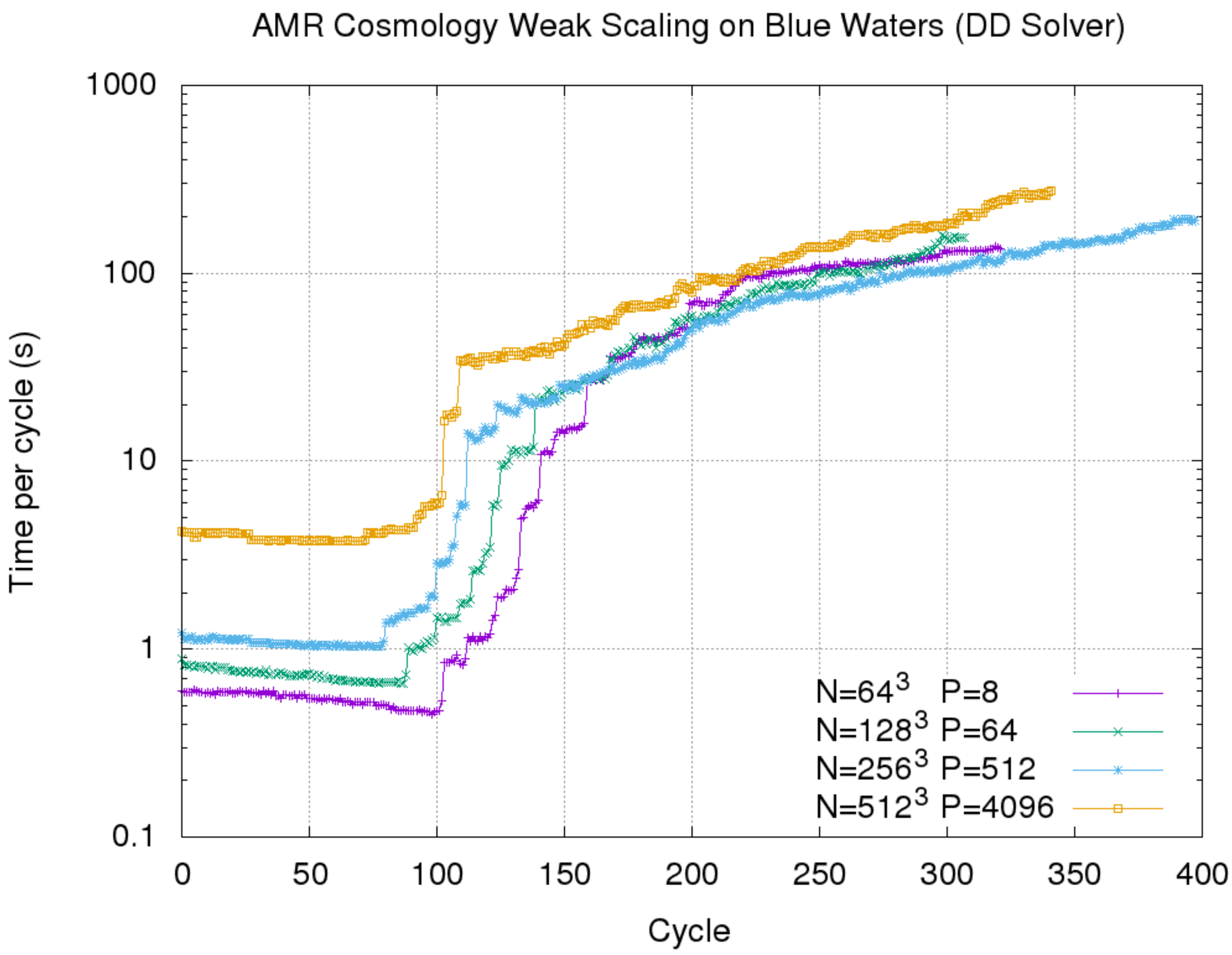
# DD in action

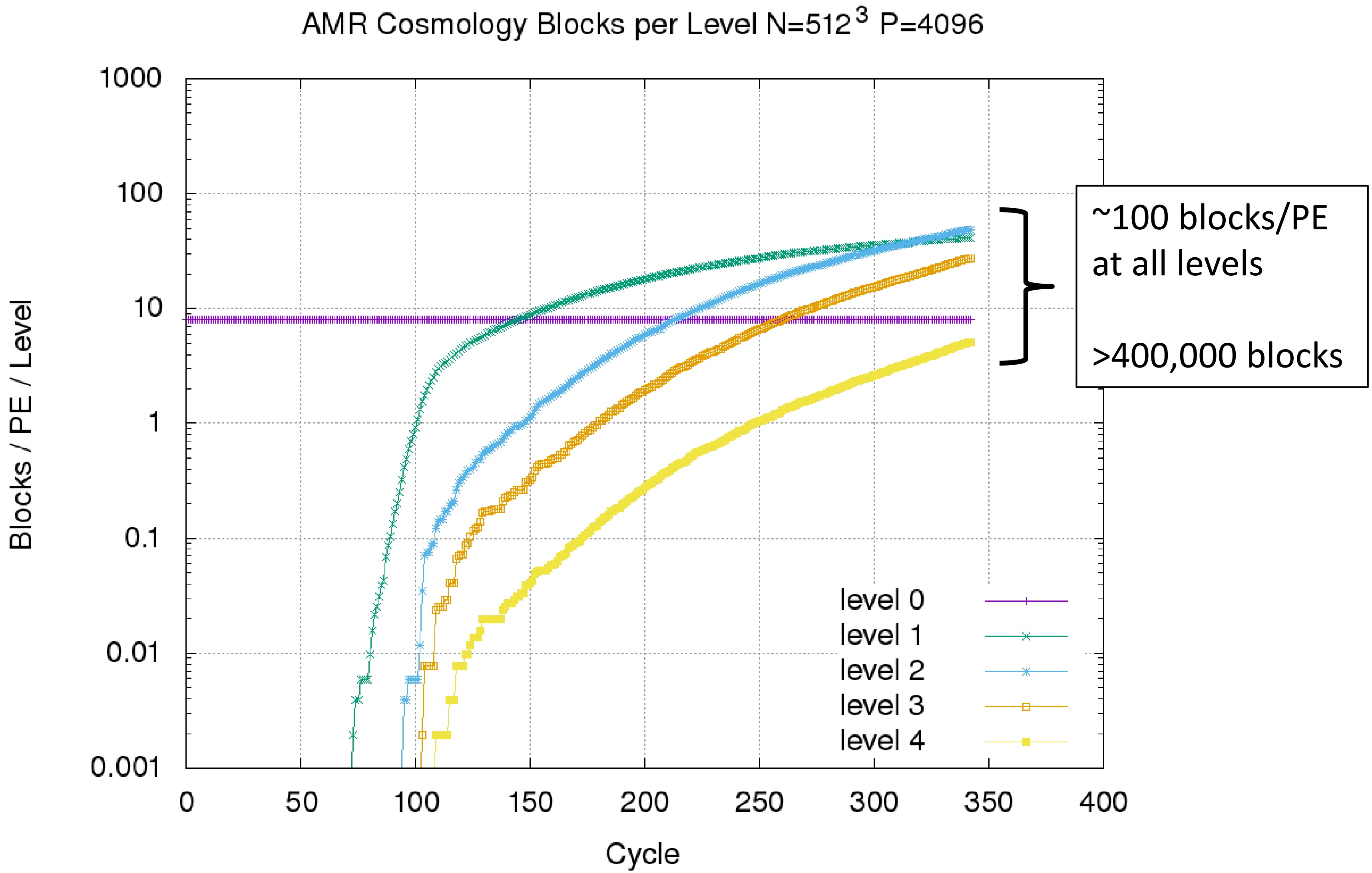
$128^3$ mesh ($8^3$ array of $16^3$ blocks), 4 AMR levels ($2048^3$ eff.), PE=64





# Is DD scalable? Yes!




AMR Cosmology Blocks per Level N=512 3 P=4096
1000
100
10
1
0.1
0.01
0.001
Blocks / PE / Level
0
50
100
150
200
250
300
350
400
Cycle
level 0
level 1
level 2
level 3
level 4
~100 blocks/PE
at all levels
>400,000 blocks

# NSF CSSI grant (ENZO-E)

- $1.9M, 3 years
- Goal: feature-complete implemention of ENZO solvers into ENZO-E
- Goal: migration of ENZO community to ENZO-E
- Goal: implementations for exascale (accelerators)
- PI team
  - Mike Norman, James Bordner (UCSD)
  - Brian O’Shea (MSU)
  - Greg Bryan (Columbia)
  - John Wise (Georgia Tech)

# ENZO-E tasks

- FMM gravity solver
- Block-adaptive local timestepping
- Adaptive ray tracing radiative transfer
- Cosmic ray transport incl. MHD
- Interfacing to GRACKLE chemistry library
- Interface to GPUs using Kokkos
- Lots of scaling/optimization work…….

# Contribution of Blue Waters

- You need a petascale platform to develop an exascale code!
  - Sheer size
  - Balanced architecture
  - Throughput (especially scaling runs)
  - Favorable Q policies
  - Mature SW environment

- Thanks for the memories!!!!