Capabilities of, and simulation software for, new high performance computers

Mike Norman

Scalability Gap (N-body envy)

ENZO

PKDGRAV2

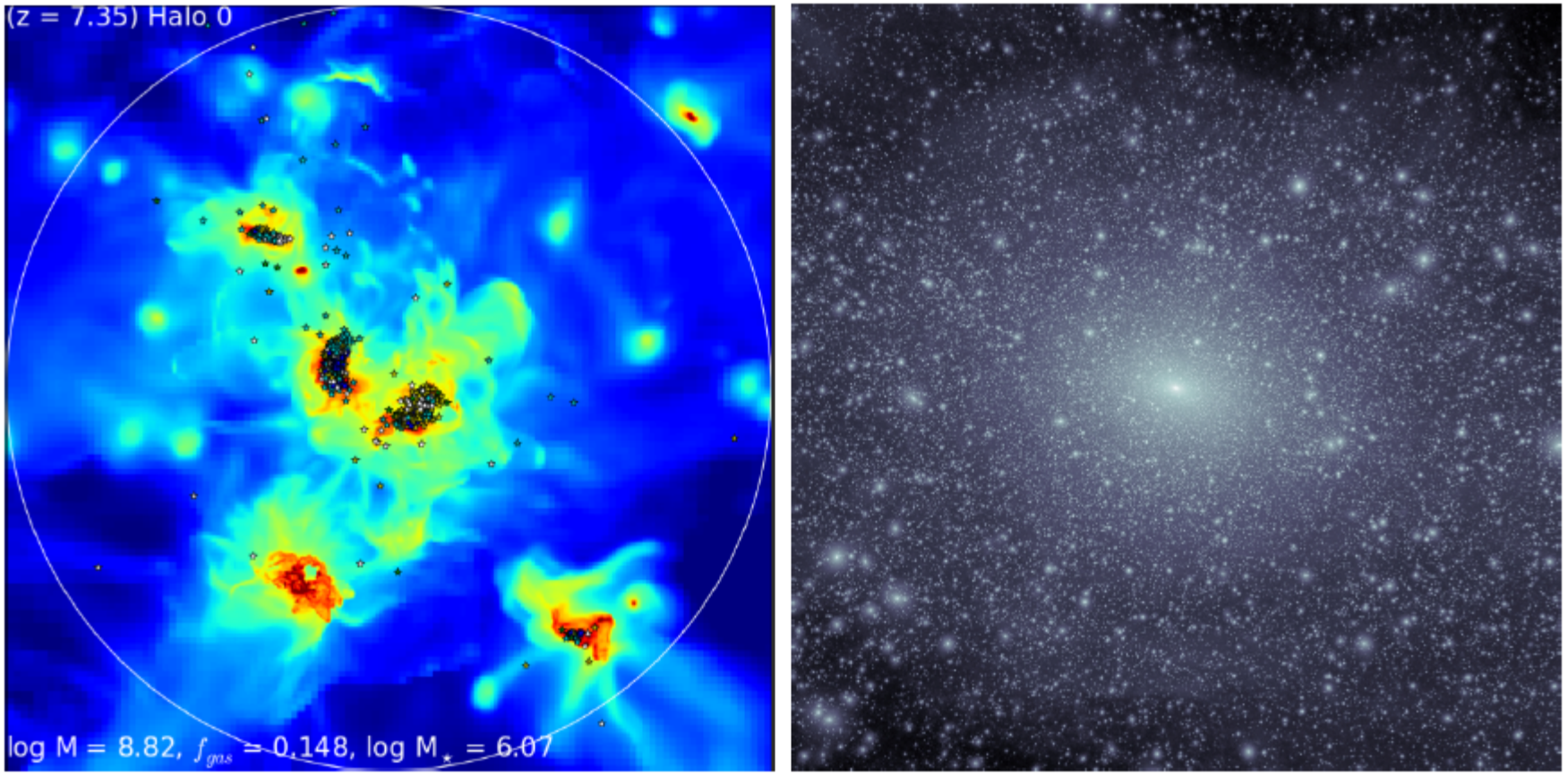
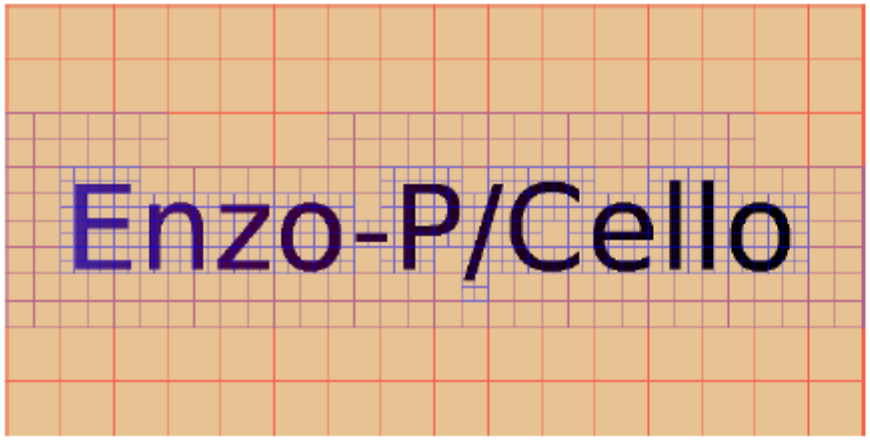
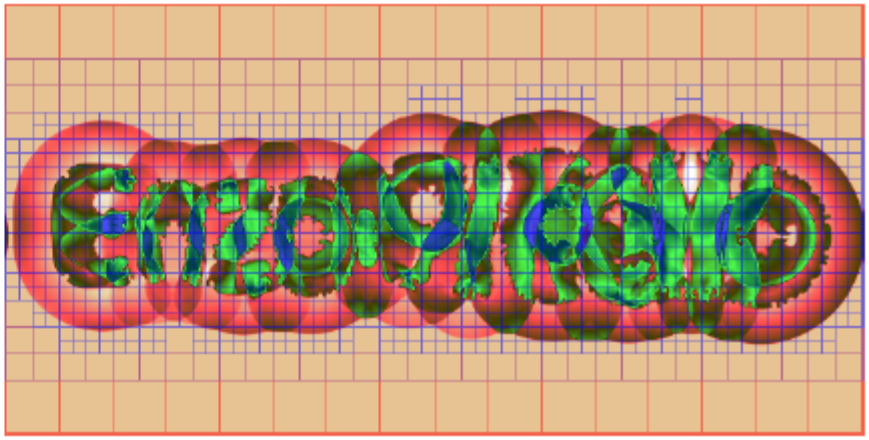


Figure 1: Current capabilities of cosmological simulations. **Left**: EnzoAMR simulation of a primeval galaxy at z=7.35. From [86] **Right**: PKDGRAV2 simulation of dark matter substructure of a Milky Way size halo at z=0. From [66].

Cello AMR: Extreme Scale Forest of Trees



initial conditions (density)



density after 300 cycles

Figure 1: A simple $P = 4$ process "explosion" test problem with Enzo-P/Cello PPM hydrodynamics

SHOW MOVIE

http://client64-249.sdsc.edu/~bordner/dm-2.html

Excellent Scalability (preliminary)

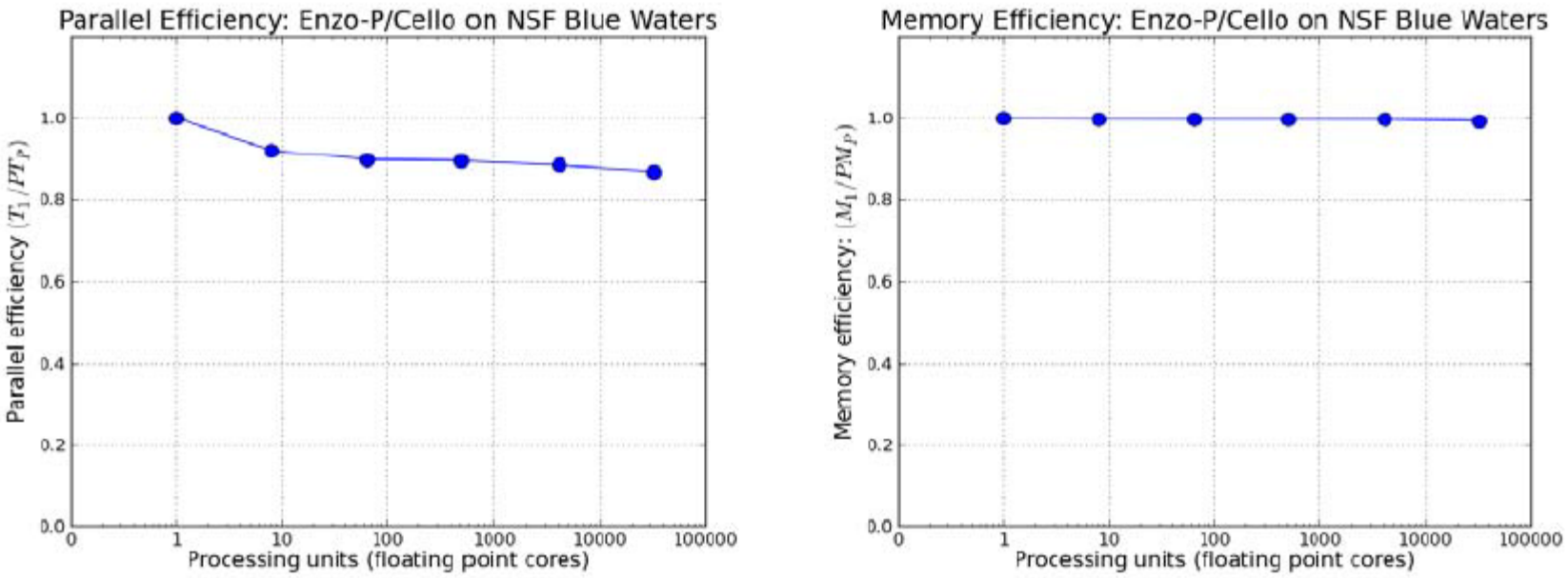
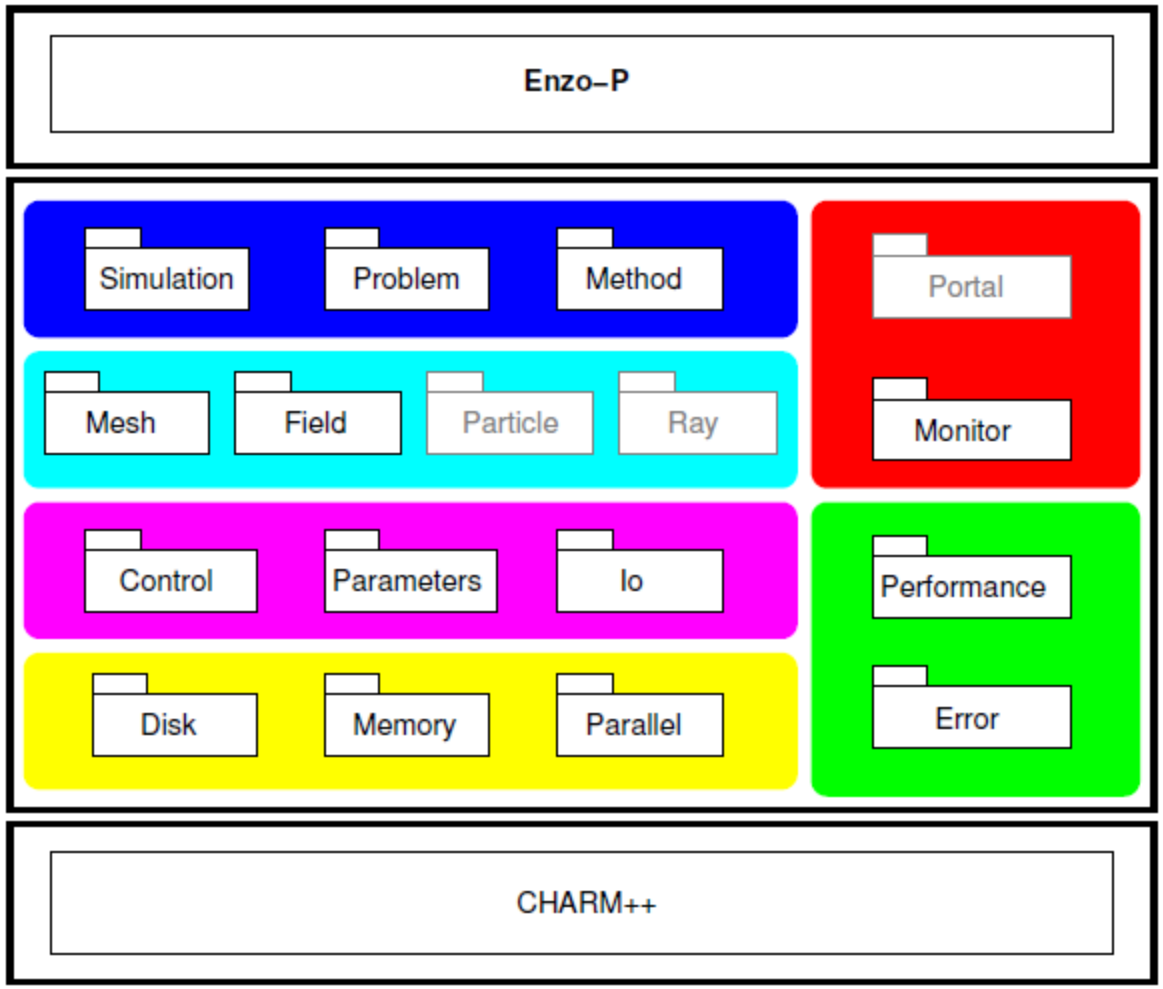


Figure 2: Enzo-P weak scaling on NSF Blue Waters: parallel efficiency (left) and memory efficiency (right)

Cello Component Diagram



astrophysics application
- physics methods

Cello
Extreme scale AMR software library (fluid, particles, rays)

Parallel layer
- asynchronous execution
- dynamic load balancing
- fault tolerance

Charm++: Parallel Objects (UIUC)

Charm++ parallel programming system

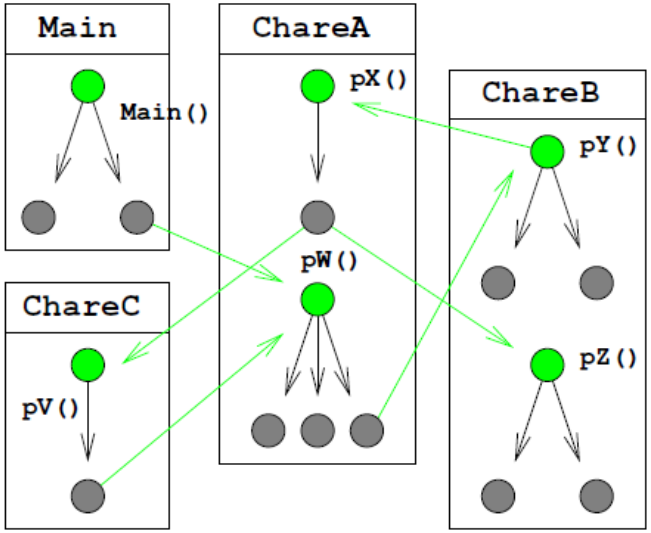
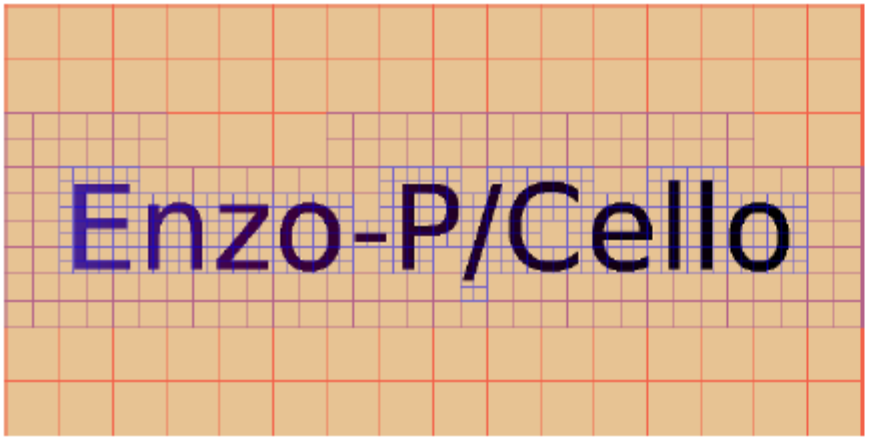
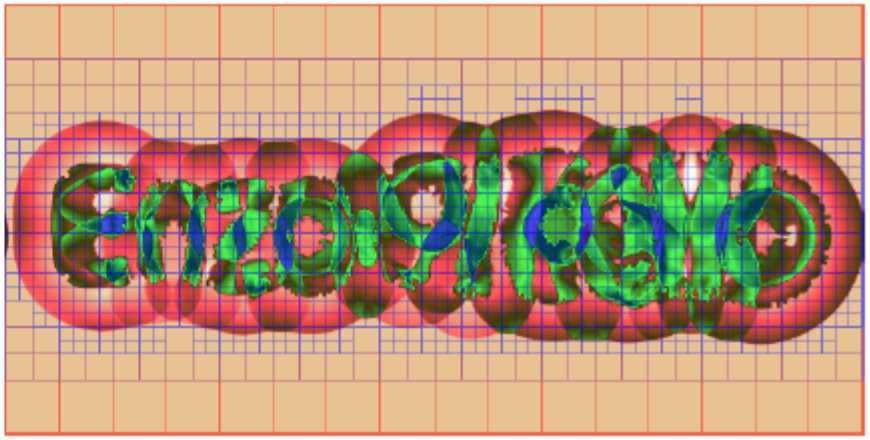


Figure 4: Charm++ data-driven asynchronous parallel programming

Cello AMR: Forest of Quad/Oct-Trees



initial conditions (density)

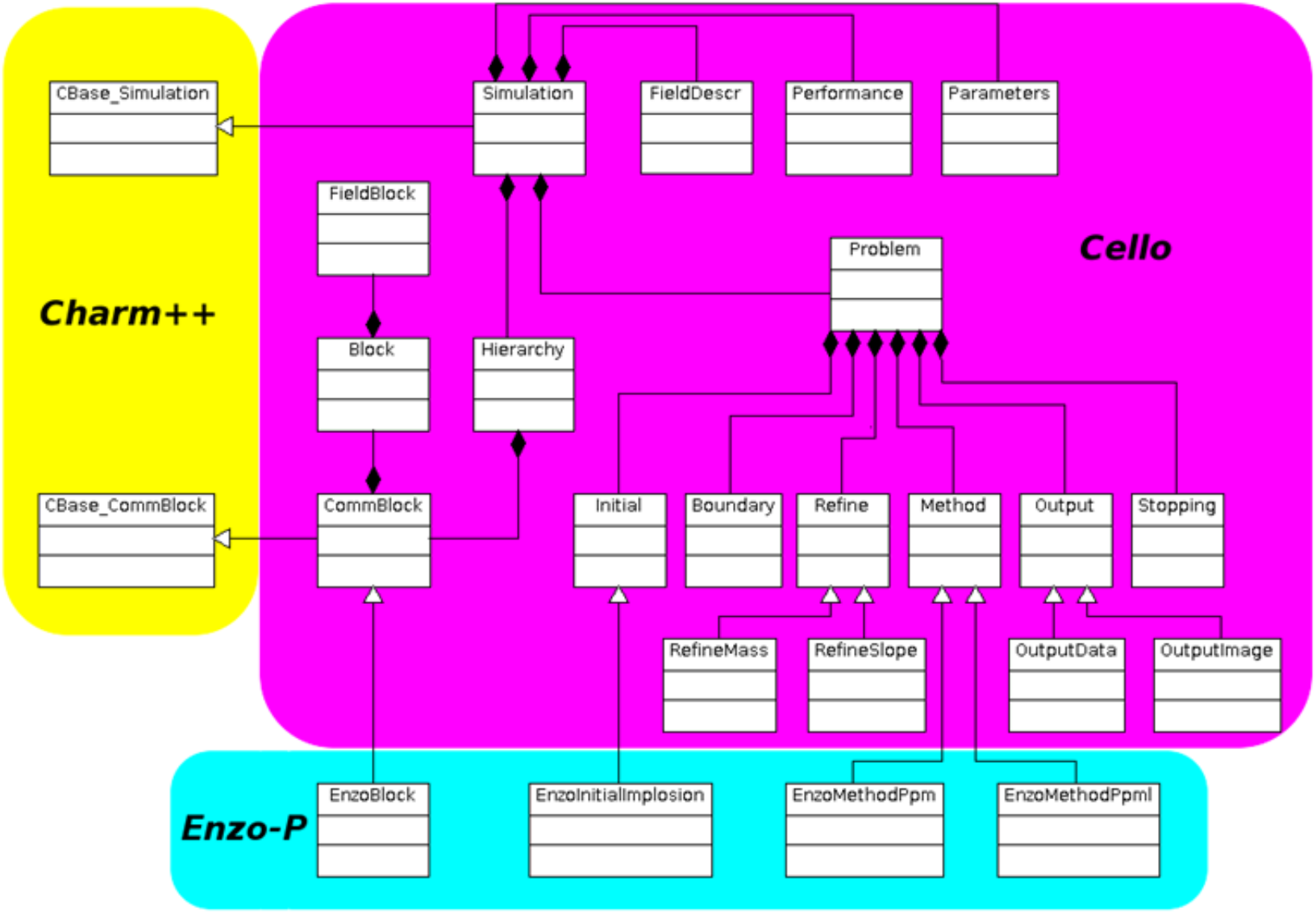


density after 300 cycles

Figure 1: A simple $P = 4$ process "explosion" test problem with Enzo-P/Cello PPM hydrodynamics

- each leaf node is a block of fixed size NxN(xN)
- Each block is a Charm++ *chare*
- *Chares* execute dynamically once they have the data they need
- *Chares* are migrated dynamically based on load (CPU and network)

Class Diagram



Key Points

- OO design confers a "separation of concerns"
  - Physics solvers (Enzo-P) separate from data structures (Cello) separate from parallelism (Charm++)
  - This permits "separation of development" amongst those best equipped
  - Also permits software reuse
- Plan: large Enzo dev-user community will migrate Enzo solvers onto Cello (UCSD), resulting in Enzo-P

Pipelined Development Timetable (very optimistic!)

Month 06: Version 2.0 (*chemistry*) Enzo-P developers will migrate chemistry solvers from Enzo to Enzo-P using the existing Cello framework; Cello developer will assist Enzo-P developers and implement support for linear solvers and adaptive time stepping.

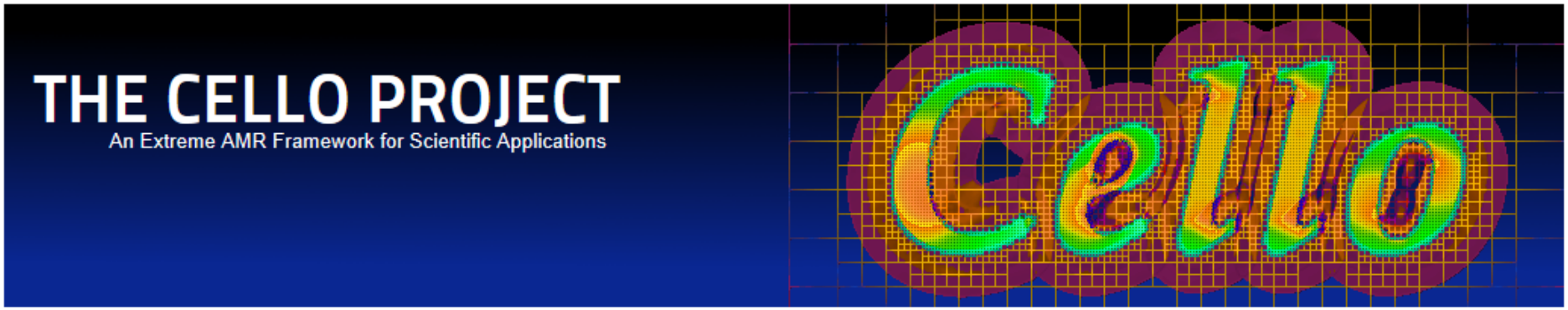
Month 12: Version 3.0 (*gravity*) Enzo-P developers will implement cosmological self-gravity and coupled-implicit flux-limited diffusion radiation transport; Cello developer will assist Enzo-P developers, implement particles in support of particle+mesh methods, and optimize Cello's linear solvers.

Month 18: Version 4.0 (*particles*) Enzo-P developers will migrate Enzo's PM method and implement PPPM; Cello developers will implement "face-methods" in preparation for migrating Enzo's MHD to Enzo-P.

Month 24: Version 5.0 (*magnetism*) Enzo-P developers will migrate Enzo's MHD methods to Enzo-P; Cello developers will implement support for adaptive ray tracing.

Month 36: Version 6.0 (*radiation*) Enzo-P developers will implement Enzo+Moray radiation hydrodynamics; Cello developers will optimize performance and scalability.

http://cello-project.org



You are here: Home

Main Menu

- **About**
- Getting Started
- Download Source
- Source Browser
- Bug Tracking
- Documentation
- Regression Tests
- Trac Site (developers)

What is Cello?

Published on Wednesday, 26 October 2011 11:40

Cello is an object-oriented "Extreme AMR" (Adaptive Mesh Refinement) software framework for high performance scientific applications, currently under development with funding from the National Science Foundation (PHY-1104819, AST-0808184). The framework will be scalable, easy to use, and portable to systems ranging from PC's and laptops to the largest HPC systems available.

Development of Cello is driven by the Enzo parallel astrophysics and cosmology application. The goal is to efficiently map Enzo's multi-resolution multi-physics capabilities onto large parallel computers with millions of computational units. This "petascale" fork of Enzo built on the Cello framework is called **Enzo-P**.

The current version of Cello is 0.5, which is undergoing beta testing. Version 0.5 supports unigrid meshes, includes both Enzo's unigrid PPM hydrodynamics and PPML MHD, and can be configured for either MPI or CHARM++ parallelism.