Progress in Scalable Adaptive Mesh Refinement (AMR) for Applications at Extreme Scales

Prof. Michael L Norman
Director, San Diego Supercomputer Center
University of California, San Diego

Software Developer: Dr. James Bordner

Supported by NSF grants SI2-SSE-1440709, PHY-1104819 and AST-0808184.
Extreme Scale Numerical Cosmology

- Dark matter only N-body simulations have crossed the $10^{12}$ particle threshold on the world’s largest supercomputers.

- Hydrodynamic cosmology applications are lagging behind N-body simulations.

- This is due to the lack of extreme scale AMR frameworks.

1 trillion particle dark matter simulation on IBM BG/Q, Habib et al. (2013)
**Enzo**: AMR Hydrodynamic Cosmology Code

http://enzo-project.org

- Enzo code under continuous development since 1994
  - First hydrodynamic cosmological AMR code
  - Hundreds of users
- Rich set of physics solvers (hydro, N-body, radiation transport, chemistry,...)
- Have done simulations with $10^{12}$ dynamic range and 42 levels
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
- 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
- Make AMR software library application-independent so others can use it
Forest (=Array) of Octrees
Burstedde, Wilcox, Gattas 2011

refined tree

unrefined tree

2 x 2 x 2 trees

6 x 2 x 2 trees
Software Architecture

- Enzo numerical solvers
- Forest-of-octrees AMR
- Charm++
- Hardware (heterogeneous, hierarchical)
Software Architecture

- Enzo-P
- Cello
- Charm++
- Charm++ supported platforms
Parallel Languages/Paradigms:

Charm++

Parallel Programming with Migratable Objects

Relevant links: exascale relevance, the manual, mini-apps, downloads, charmplusplus.org

Charm++ is a machine independent parallel programming system. Programs written using this system will run unchanged on MIMD machines with or without a shared memory. It provides high-level mechanisms and strategies to facilitate the task of developing even highly complex parallel applications.

Charm++ programs are written in C++ with a few library calls and an interface description language for publishing Charm++ objects. Charm++ supports multiple inheritance, late bindings, and polymorphism.

Platforms: The system currently runs on IBM's Blue Gene/P, Blue Gene/L, Cray XT3, XT4, XT5, Infiniband clusters such as Ranger, LoneStar and Abe, clusters of UNIX workstations and even single-processor UNIX, Solaris and Windows machines. It also runs on accelerators such as the Cell BE and GPGPUs.

The design of the system is based on the following tenets:
(7.1) What is Charm++?

Charm++ parallel programs: collections of asynchronously-interacting objects

- Charm++ program
  - Decomposed by objects
  - Charm++ objects called *chares*
  - invoke *entry methods*
  - *asynchronous*
  - communicate via *messages*

- Charm++ runtime system
  - maps *chares* to processors
  - schedules entry methods
  - migrates *chares* to load balance

- Additional features
  - checkpoint/restart
  - dynamic load balancing
  - fault-tolerance

*(Laxmikant Kale et al. PPL/UIUC)*
How does Cello implement FOT?

- A **forest** is an array of **octrees** of arbitrary size $K \times L \times M$
- An octree has leaf nodes which are **blocks** ($N \times N \times N$)
- Each block is a **chare** (unit of sequential work)
- The entire FOT is stored as a **chare array** using a bit index scheme
- Chare arrays are **fully distributed data structures** in Charm++
Demonstration of Enzo-P/Cello

Total energy
Demonstration of Enzo-P/Cello

Mesh refinement level
WEAK SCALING TEST – HOW BIG AN AMR MESH CAN WE DO?
Weak scaling test: Alphabet Soup
array of supersonic blast waves

<table>
<thead>
<tr>
<th>N trees</th>
<th>Np = cores</th>
<th>Blocks/Chares</th>
<th>Cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>1³</td>
<td>1</td>
<td>201</td>
<td>6.6 M</td>
</tr>
<tr>
<td>2³</td>
<td>8</td>
<td>1,608</td>
<td></td>
</tr>
<tr>
<td>3³</td>
<td>27</td>
<td>5,427</td>
<td></td>
</tr>
<tr>
<td>4³</td>
<td>64</td>
<td>12,864</td>
<td></td>
</tr>
<tr>
<td>5³</td>
<td>125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6³</td>
<td>216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8³</td>
<td>512</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10³</td>
<td>1000</td>
<td>201,000</td>
<td></td>
</tr>
<tr>
<td>16³</td>
<td>4096</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24³</td>
<td>13824</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32³</td>
<td>32768</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40³</td>
<td>64000</td>
<td>12.9M</td>
<td></td>
</tr>
<tr>
<td>48³</td>
<td>110592</td>
<td>22.2M</td>
<td>0.7T</td>
</tr>
<tr>
<td>54³</td>
<td>157464</td>
<td>31.6M</td>
<td>1.0T</td>
</tr>
<tr>
<td>64³</td>
<td>262144</td>
<td>52.7M</td>
<td>1.7T</td>
</tr>
</tbody>
</table>
Largest AMR Simulation in the world

1.7 trillion cells

262K cores on NCSA Blue Waters
Enzo-P Weak Scaling on Blue Waters: Efficiency

Alphabet Soup Test (SMP): 161031

Efficiency

Processes (FP cores)

parallel efficiency
memory efficiency
Takeaways

• **Cello** is a software framework for extreme scale AMR simulations

• **Cello** implements the most scalable AMR algorithm known: forest-of-octrees

• Parallelism is handled by **Charm++**, which supports fully distributed AMR data structures, asynchronous execution, dynamic load balancing, and fault tolerance, parallel IO

• Developing applications on top of **Cello** is easy—as simple as writing a sequential program

• It is available NOW at [http://cello-project.org](http://cello-project.org)
Path Forward (Oct 2016)

• Finish scalable gravity solver (50% complete)
• Implement block adaptive timestepping
• Experiment with Charm’s built-in DLB schemes and dynamic execution
• Do a 1 trillion cell/particle hydro cosmology simulation for the fun of it
(1.2) Scalable gravity
Parallel scaling test problem

We tested scalability of a recently implemented gravity solver

- “array of collapsing spheres” problem
- varied both block size and problem size
- multigrid V-cycle solve (uniform-mesh)
- debugging scalable AMR gravity
  - Reynolds “HG” algorithm
  - BiCG-STAB Krylov solver
  - multigrid-based preconditioner
(1.2) Scalable gravity
Gravity weak scaling

![Graph showing Enzo-P Gravity Weak Scaling on Blue Waters.](Image)
(1.2) Scalable gravity
Gravity strong scaling

Enzo-P Gravity Strong Scaling on Blue Waters

Time per MG V-cycle (s)
Processing units (FP cores)
Scalable Hydrodynamic Cosmology

Enzo-P Cosmology Weak Scaling on Blue Waters

- Time per cycle (s)
- Processing units (FP cores)

Symbols:
- + N64
- × N256
- ★ N1024
- ▲ N128
- ▼ N512
- ◀ N2048
Scalable Hydrodynamic Cosmology

Enzo-P Cosmology Strong Scaling on Blue Waters

Run in non-SMP mode

Processing units (FP cores)

Time per cycle (s)
Resources

- Project site: [http://www.cello-project.org](http://www.cello-project.org)
- Source code: [https://bitbucket.org/cello-project](https://bitbucket.org/cello-project)
- Tutorials: on project site
THANK YOU!