Computational Relativity and Gravitation at Petascale: Simulating and Visualizing Astrophysically Realistic Compact Binaries

Scott C. Noble

PI: M. Campanelli

J. Faber, B. Mundim, Y. Zlochower

Center for Computational Relativity and Gravitation
Rochester Institute of Technology (RIT)

NEIS-P² Symposium * Blue Waters * NCSA * May 21, 2013
Circumbinary Accretion Problem:

Galactic Merger → Binary Formation → Inspiral → Merger → Re-equilibration

Other Codes:
- Zeus
- Harm3d
- ET/Lazev
- Harm3d
- Bothros

Hopkins, Hernquist, Springel et al.

Noble++2012

Farris++2011

Physical Time (not to scale)
Multimessenger Synergy

Electromagnetic Surveys

Pan-STARRS:
• 2010–??
• 4 skies per month

Large Synoptic Survey Telescope (LSST):
• 2021–2032
• 1 sky every 3 days

Gravitational Wave Observatories

eLISA/NGO

Advanced LIGO

• GW Detection/Localization \(\iff\) EM Detection/Localization;
• GW and light are connected theoretically but originate in wholly different mechanisms
  • \(\iff\) independently constrain models;
• Either GW or EM observations of close supermassive BH binaries would be the first of its kind!
• Follow up (X-ray, sub-mm) observations can often be made via coordinated alert systems;
• Cosmological “Standard Sirens”: New Distance vs. Redshift Measurement

Black Hole Accretion Anatomy

• Ideal Magnetohydrodynamics (MHD)
  • General Relativity (GR)
  • Radiative Transfer, Ray-tracing
  • Multi-species thermodynamics
The Codes

Harm3d

- Ideal-MHD on curved spacetimes (does not evolve Einstein’s Equations)
- 8 coupled nonlinear 1st-order hyperbolic PDEs; 1 constraint eq. (solenoidal): Constrained Transport, FluxCT method;
- Finite Volume, methods, Lax-Friedrichs, HLL fluxes (approx. Riemann solvers); PPM reconstruction; “Method of Lines”: 2nd-order Runge-Kutta;
- “Mesh refinement” via coordinate transformation: Eqs. are solved on uniform “numerical” coordinates related to “physical” coordinates via nonlinear algebraic expressions;
- Parallelization via uniform domain decomposition; 1 subdomain per process
- No threading, simple MPI distribution;
- Computationally & memory access “intensive”, little I/O and MPI overhead;
- \( O(10^7 - 10^8) \) cells evolved for \( O(10^6) \) time steps;

Bothros

LazEv & Einstein Toolkit
The Codes

Harm3d

Bothros

- Predict electromagnetic emission from relativistic gas simulations;
- Solves the Radiative Transfer and Geodesic Equations in curved spacetimes;
- RT Eq: 1 nonlinear ODE; Geodesic Eqs: 8 coupled linear ODEs;
- Post-processes Harm3d simulation data;
- \( O(10^3) \) time frames of \( O(10^5) \) rays that travel through 4D data cube of \( O(10^{10}) \) spacetime points from which \( O(10) \) functions are interpolated onto light ray’s path;
- Very Data (I/O) Intensive -- processes Terabytes of data!
- Originally trivially parallelized, i.e. no MPI or OpenMP support;
  - \( \rightarrow \) Many redundant disk reads!

LazEv & Einstein Toolkit
The Codes

Harm3d

Bothros

LazEv & Einstein Toolkit

- ET = “an open, community developed software infrastructure for relativistic astrophysics”;
- Comprised of Cactus, Carpet, Whisky, McLachlan, (parts of Harm3d);
- E.g., solves Einstein’s equations, w/ or w/o Hydro/MHD;
- Block structured adaptive mesh refinement;
- www.einsteintoolkit.org
- LazEv = RIT’s unique set of “thorns” that formulate and discretize Einstein’s equations;
Approximate Two Black Hole Spacetimes
Yunes++2006, Mundim++2013

- Solve Einstein’s Equations approximately, perturbatively;
  \[ \varepsilon_i = m_i/r_i \sim (v_i/c)^2 \]
- Used as initial data of Numerical Relativity simulations;
- Closed-form expressions allow us to discretize the spacetime best for accurate matter solutions;
- Physically valid up until the last few orbits prior to merger;

Metric Analytic Approximation: Initial Data

Global analytic approximation for the metric describing the late quasimcircular inspiral of two comparable black holes \( g_{(Yunes et al.)} \); JohnsonmMcDaniel et al. \( g_{(ppv)} \)

Inner Zone \( g_{(r_i)} \ll b_{(hzw)} \) well described by black hole perturbation theory \( \varepsilon_i \approx r_i/b_{(hn)} \) Use Detweiler’s Schwarzschild perturbed metric in \( g_{(CookmScheel)} \)

Near Zone \( g_{(r_i)} \gg m_i \) and \( r_i \leq \lambda/r_{(πhzw)} \) \( g_{(slowmmotionoweak fieldz)} \) \( \varepsilon_i \approx m_i/r_i \sim (v_i/c)^2 \) post-MN theory of point particles in \( g_{(BlanchetmFayemPonsot)} \)

Far Zone \( g_{(r_i)} \geq \lambda/r_{(πhzw)} \) post-Minkowskian theory; Harmonic coordinates; Expansion in terms of radiative multipole moments; Non-perturbative gravitational radiation treatment

\[ g_{(global)}^{(μν)} = F_2(R_2)F_1(R_1)g_{(μν)}^{(3)} + [1 - F_1(R_1)]g_{(μν)}^{(1)} + [1 - F_2(R_2)]g_{(μν)}^{(2)}. \]
- Significant effort spent optimizing subroutine, with the aid of symbolic manipulation software (Maple);
- Metric evaluation accounts for ~35% of runtime;
Load Balancing Domain Decomposition

- Different zones of the spacetime vary in computational cost of evaluating metric;
- Strategy: decompose costlier regions into smaller domains, balancing effort across MPI processes;
- Black Holes (or zones) move through the grid --> “dynamic” load balancer;
- Need to alter static array definitions to dynamic allocations to handle nonuniform decomposition across processors;

<table>
<thead>
<tr>
<th>Zone</th>
<th>Relative Cost Per Cell</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner</td>
<td>3</td>
</tr>
<tr>
<td>Inner/Near Buffer</td>
<td>4</td>
</tr>
<tr>
<td>Near</td>
<td>1</td>
</tr>
<tr>
<td>Near/Far Buffer</td>
<td>2</td>
</tr>
<tr>
<td>Far</td>
<td>$\sim 1$</td>
</tr>
</tbody>
</table>
Harm3d Goals

• **Solve the Load Imbalance Problem:**
  - Static Memory Allocation --> Dynamic Memory Allocation;
  - Nonuniform domain decomposition (different subdomain sizes across processors):
    - Generalized subdomain boundary conditions (passing of ghost zone data);
    - Generalized data reduction routines;
  
• **Load Balancing Algorithm:**
  - Method to distribute cost evenly;
  - Ability to re-evaluate cost distribution and redistribute;
  - Profile complete package on BW with a production run;

• **Incorporate OpenMP:**
  - Preliminary tests suggest only modest performance improvement;

• **May incorporate GPUs ala Jian Tao’s talk & (Zink 2011)**
Load Balancing Algorithm

1. Start with global domain with cost estimates for each cell;
2. Order subdomains by cost;
3. Bisect most expensive domain along longest extent (maintain cubical domains);
4. Assign processor to new subdivision;
5. Determine neighbor relationships;
6. Repeat Steps 2 - 5 until all processors have been assigned;
Validation of Load Balancer through Simulated Cost Distribution:

### Domain Count

<table>
<thead>
<tr>
<th>Count</th>
<th>Uniform</th>
<th>Non-uniform</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td><img src="Image1" alt="Image" /></td>
<td><img src="Image2" alt="Image" /></td>
</tr>
<tr>
<td>128</td>
<td><img src="Image3" alt="Image" /></td>
<td><img src="Image4" alt="Image" /></td>
</tr>
<tr>
<td>256</td>
<td><img src="Image5" alt="Image" /></td>
<td><img src="Image6" alt="Image" /></td>
</tr>
<tr>
<td>512</td>
<td><img src="Image7" alt="Image" /></td>
<td><img src="Image8" alt="Image" /></td>
</tr>
</tbody>
</table>

Colors differentiate domains.

### Decomposition

### Cost Imbalance

\[
R_i = \frac{C_i - \bar{C}}{\bar{C}}
\]

Where:
- \(R_i\) is the cost imbalance for domain \(i\).
- \(C_i\) is the cost for domain \(i\).
- \(\bar{C}\) is the average cost across all domains.

### 2-d Cost Model

- **I**: Inner region
- **4**: Medium region
- **3**: Outer region

- **0**: Uniform distribution
- **4**: Non-uniform distribution
Validation of Load Balancer through Simulated Cost Distribution:

Perfect Balance

Load Balancer

Speedup Factor over Uniform Decomp.

Number of Processors
Validation of Load Balancer through Simulated Cost Distribution:

Saturation of Domain Decomposition

--> In practice, more processors will not be added at saturation point
Performance on Blue Waters:

Runtime Efficiency

TACC/Ranger
2400 procs
300x160x400 cells

Blue Waters
2400 procs
300x160x400 cells

Why the difference?

gprof

CrayPat

Why the difference?
Runtime Efficiency

Performance on Blue Waters:

- Little difference between PGI and Cray compilers;
- Little difference between Static and Dynamic memory allocation;
- Decrease in rate with more zones consistent with prior profiles;

* Static code seg. faults with Cray’s default optimizations;
  * Static = statically allocated grid functions;
  * Dynamic = dynamically allocated grid functions;
Performance on Blue Waters:

Good Scaling Performance

- Using Dynamic Harm3d

Strong

- PGI

Weak

- Cray

More Cells per FPU

Fewer Cells per FPU
Bothros Goals: Towards Radiative Transfer in Time-dependent General Relativity

- Parallelize post-processing tool via MPI and OpenMP;
- Will explore how GPUs can offload effort in the future;

X-ray Emission from Single Black Hole Disk
Noble & Krolik 2009

Binary Black Hole System in Photon “Cloud”
Schnittman, Krolik, Noble 2012

• Thesis Project of Billy Vazquez (grad student);
### Bothros’s Parallelization Model

<table>
<thead>
<tr>
<th>Master Unit (MU)</th>
<th>Evaluates problem extent; Assigns duties; Broadcasts what data is available on IOUs;</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Unit (IOU)</td>
<td>Reads time slices from disk; Serves data to CUs when its needed; Replaces processed slices with new ones;</td>
</tr>
<tr>
<td>Compute Unit (CU)</td>
<td>Requests data from IOUs; Interpolate data onto rays; Integrate radiative transfer eq.; Advance rays to next data slice;</td>
</tr>
</tbody>
</table>

#### Strategy 1:
One unit per core, each threaded

- MU: Evaluates problem extent, Assigns duties, Broadcasts what data is available on IOUs.
- IOU: Reads time slices from disk, Serves data to CUs when its needed, Replaces processed slices with new ones.
- CU: Requests data from IOUs, Interpolate data onto rays, Integrate radiative transfer eq., Advance rays to next data slice.

#### Strategy 2:
One unit per thread

- MU: Evaluates problem extent, Assigns duties, Broadcasts what data is available on IOUs.
- IOU: Reads time slices from disk, Serves data to CUs when its needed, Replaces processed slices with new ones.
- CU: Requests data from IOUs, Interpolate data onto rays, Integrate radiative transfer eq., Advance rays to next data slice.

* Requires facility for master threads to make MPI calls (MPI_THREAD_FUNNELED)
Einstein Toolkit/LazEv Goals

- Implement threading via OpenMP throughout new GRHydro code of ET;
- Analysis routines;
- Reconstruction at cell interfaces;
- Stress-energy calculation;
- Inversion of nonlinear algebraic equations to find primitive variables from conserved variables

- Evaluate the performance gain on Blue Waters;
Conclusions:

- Blue Waters provides a singular facility and opportunity for us to calculate the most accurate electromagnetic predictions of coalescing supermassive black hole binaries;
- We are close to finishing our NEIS-P$^2$ version of Harm3d;
  - Experiments on Blue Waters confirm our earlier performance models;
  - Dynamic code scales well on Blue Waters;
  - Load balancer is expected to at least HALVE effort!
- Bothros development underway;
- LazEv & Einstein Toolkit development done, but need to profile more on BW;
- Soon, we will have circumbinary disk simulations at unprecedented accuracy, longevity, and physical realism!

Questions? Discussion....