The Einstein Toolkit

Frank Löffler

Center for Computation and Technology
Louisiana State University, Baton Rouge, LA

May 22th 2011

Supported by NFS grants 0903973/0903782/0904015 (CIGR)
The Einstein Toolkit

Collection of software components and tools
Targeted at simulating and analyzing general relativistic astrophysical systems
Freely available as open source
See http://einsteintoolkit.org
Guiding Principles

- Open, community-driven software development
- Separation of **physics** software from **computational science** infrastructure
- Well thought out and stable interfaces
- Providing core computational tools to
  - facilitate interdisciplinary research
  - enable new science
  - broaden community
  - take advantage of emerging Petascale computers and advanced cyberinfrastructure
Current State

- Last release on April 21, 2011
- New releases roughly every six months
- Atm. mostly Cactus (framework), Carpet (AMR)
- Production quality features (Ninja, NRAR)
- Not all new - grown from other projects
- About 50 contributors over the past decade, both physics and CS
- Currently 62 registered users from 23 research groups worldwide
- 9 maintainers from 6 different institutions
- > 200 publications, > 30 theses building on these components
Science Capabilities

- McLachlan (BSSN, up to 8th order)
- GRHydro (formerly based on WhiskyCode; Valencia formulation)
- BH/NS initial data (TwoPunctures, Lorene)
- Turduckening
- MoL + e.g., Runge-Kutta
- Carpet (Adaptive Mesh Refinement)
- Black hole horizon finder (AHFinderDirect)
- Gravitational wave extraction
- Parallelization: MPI, OpenMP
- Tools: e.g., EOS reader, HDF5 output, Visualization import
Open source ≠ offering download

Requirements for (new) Einstein Toolkit components:
- sufficient quality (actually being used)
- documentation (e.g. publication)
- community interest (maintained)

Regular, tested releases

Providing easy step-by-step instructions for first-time users
# Einstein Toolkit testsuite status

## Testsuite status of einstein-toolkit.th

<table>
<thead>
<tr>
<th>Test</th>
<th>Status</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>Passed</td>
<td>Description 1</td>
</tr>
<tr>
<td>Test 2</td>
<td>Failed</td>
<td>Description 2</td>
</tr>
<tr>
<td>Test 3</td>
<td>Pending</td>
<td>Description 3</td>
</tr>
</tbody>
</table>

## Test Results

<table>
<thead>
<tr>
<th>Module</th>
<th>Status</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 1</td>
<td>Passed</td>
<td>Description 1</td>
</tr>
<tr>
<td>Module 2</td>
<td>Failed</td>
<td>Description 2</td>
</tr>
<tr>
<td>Module 3</td>
<td>Pending</td>
<td>Description 3</td>
</tr>
</tbody>
</table>

## Test Environment

- Operating System: Ubuntu 18.04
- Python Version: 3.7.8
- TensorFlow Version: 2.2.0

---

**Frank Löffler**

The Einstein Toolkit

2011-05-22
Get account on cluster “QueenBee” (fill in web form)
Download (4 shell commands)
Configure (3 commands) [user name, email address, allocation]
Build (1 command)
Run simulation (1 command)
General Relativity

- Why we need to consider it?
- Curvature of Space-time
- Einstein Toolkit / Cactus
Science vs. Simulation

Doing science >> Running a simulation

Need to know about meaningful initial conditions, numerical stability, accuracy/ resolution, have patience, have curiosity, have a “gut feeling” for what is “right”...

Einstein Toolkit cannot give that!

But, with open codes that are easy to use, can concentrate on science!
Tools: GetComponents

Task: Collect software from various repositories at different sites

Example simulation assembly:

- Core Einstein Toolkit (svn.einsteintoolkit.org)
- Cactus Flesh and Toolkit (svn.cactuscode.org)
- Carpet AMR (carpetcode.org, hg)
- Tools, Parameter Files and Data (svn.einsteintoolkit.org)
- Group Modules (x.groupthorns.org)
- Individual Modules (x.mythorns.org)

x: cvs, svn, darcs, git, hg, http
Tools: Simulation Factory

http://www.simfactory.org/

Task: Provide support for common, repetitive steps:

- Access remote systems, synchronize source code trees
- Configure and build on different systems semi-automatically
- Provide maintained list of supercomputer configurations
- Manage simulations (follow “best practices”, avoid human errors)
Task: Ensure that simulations are and remain repeatable, remember exactly how they were performed

Take snapshots of source code, system configuration; store it in executable and/or git repository

Tag all output files
The Einstein Toolkit

- Does not want to provide “the” best code
- Rather wants to offer different codes with are
  - of high quality
  - maintained
  - open source
  - easy to use

→ Computational General Relativity for the Masses
Support

- Web site http://einsteintoolkit.org
- Mailing list users@einsteintoolkit.org
- Bug tracking system http://trac.einsteintoolkit.org
- Weekly public phone meetings
- Also blog, wiki, code repositories, ...
Acknowledgements

- More than 4 dozen contributors (over past years)
- NSF projects: CIGR, XiRel, Alpaca, PetaCactus (LSU, GA Tech, RIT, Caltech, AEI)
- LONI “ioni_cactus” allocation
Looking for users and contributions
Don’t want to take over software
Currently Cactus-centric, want to expand in other directions
Also looking for analysis tools / scripts (e.g. gravitational wave postprocessing tools)
Working on GRMHD
Improve existing numerical methods (scaling)
Investigate alternative methods for GR