



Advanced Prototype Implementation of Monitoring & Steering Methods Documentation and Test Report¹

Deliverable	D6.5
Authors	Thomas Radke (AEI)
Editors	Thomas Radke (AEI)
Date	12 December 2007
Document Version	1.0.0
Current Version	1.0.0
Previous Versions	0.1.0

A: Status of this Document

Officially approved document for project deliverable D6.5.

B: Reference to project plan

This deliverable document refers to the tasks TA VI-IV "*Generalisierte Version der Zugriffsmethoden*" and TA VI-VI "*Anpassung der Anwendungen und Testen der entwickelten Middleware*" of work package WP-6 in the project plan.

¹This work is part of the D-Grid initiative and is funded by the German Federal Ministry of Education and Research (BMBF).

C: Abstract

This document describes the second prototype implementation of grid-enabled monitoring methods in selected AstroGrid-D applications, available as of month 28 into the project (December 2007).

D: Changes History

Version	Date	Name	Brief summary
0.0.1	13 August 2007	Thomas Radke	Working Draft Creation
0.0.2	23 October 2007	Thomas Radke	Description of Cactus monitoring/steering functionality
0.0.3	29 October 2007	Thomas Radke	Test Results section
0.1.0	19 November 2007	Thomas Radke	Announcement as Official Working Draft
1.0.0	12 December 2007	Thomas Radke	Publication as official AstroGrid-D Deliverable Document

E:

Contents

1	Introduction	4
2	Advanced Version of Monitoring & Steering Methods for Cactus	5
2.1	Connecting to Cactus Simulations in Restricted HPC Environments	6
2.2	Interactive Simulation Control and Online Steering of Parameters	6
2.3	Access to temporary stdout/stderr Logfiles	8
2.4	Cactus Thorn VISUALISATION	9
3	Test Report Summary	11
3.1	Specific Test Results	11
3.2	Dissemination	12
	References	13

1 Introduction

Following the design of the basic structure of grid-enabled monitoring & steering methods for AstroGrid applications and a first prototype implementation with still limited monitoring functionality available for a subset of Grid resources, the next task in the project workplan for working group WG-VI "*Grid Job Monitoring & Steering*" was to provide a second, more advanced version which incorporates the test experiences obtained with the first prototype, includes feedback from users, and offers a generalised set of both monitoring and steering functionality which is then also made available on a broader range of Grid resources.

This deliverable document describes the grid-enabled monitoring and steering methods implemented for the AstroGrid use case *Cactus* [1], a simulation framework used in gravitational wave analysis at AEI to numerically solve Einstein's equations of general relativity. The methods' basic design was defined in working group WG-VI's architecture design document [3]. The implementation of a first prototype followed and was described in [4]. The second prototype which is documented in this deliverable has also been presented at the most recent AstroGrid project meeting in Göttingen[5] and in various occasions during hands-on sessions and practical tutorials for the astrophysicists in the *Numerical Relativists* groups at the *Albert Einstein Institute* (AEI) and the *Center for Computation and Technology* (CCT) at the Louisiana State University.

The methods described in this document have been specifically developed for and are closely integrated into the *Cactus* framework in order to enable astrophysicists to monitor and steer *Cactus* simulations, however their design should be generic and flexible enough to be incorporated in other AstroGrid numerical simulations use cases as well. Two important preconditions for this are the availability of sufficient user documentation for the software modules and the interfaces implemented (as described in this document), together with free public access to the code for other software developers (as provided through the AstroGrid-D SVN).

2 Advanced Version of Monitoring & Steering Methods for Cactus

This section describes the newly available grid-enabled monitoring/steering methods which are provided for the *Cactus* use case[1] as they have been implemented in the second prototype version. It builds on an and extends the functionality of the Cactus webserver thorn HTTPS in the first prototype which still had a limited set of functionality (see [4] for a detailed description of HTTPS).

The second prototype version completes the existing monitoring/steering functionality, focussing especially on grid-enabled methods to interactively *control* Cactus simulations and emphasising on providing a user-friendly interface to intuitively follow the progress of a running simulation or to visualise intermediate results. The functionality of the second prototype was also generalised in such a way that it can be used by the community for production-mode Cactus simulations on a broad range of computing resources (both within D-Grid and other Grid infrastructures).

2.1 Connecting to Cactus Simulations in Restricted HPC Environments

At simulation startup, thorn HTTPS creates a server socket on the first processor of a parallel simulation, using an available TCP port, and offers this `https : // < hostname > : < port >` URL as connection point to login to the running simulation (eg. `https://ic0092:5555/` in a simulation's stdout logfile example below). Users then simply point their standard web browser to this URL in order to get to the simulation's homepage.

```
HTTPS web server started on:

    https://ic0092:5555/

HTTPS proxy server started on:

    https://peyote.aei.mpg.de:24000/
```

However, it often happens in supercomputer environments that the compute nodes (where the simulation is running) may be firewalled or not be visible from the outside world. This has the effect that users cannot *directly* connect to a simulation using the URL pointing to the HTTPS webserver location. As a solution for this known problem, a proxy server needs to be installed on some publicly accessible frontend machine (eg. the headnode of a cluster) which would then redirect all incoming client requests to the actual compute node where the simulation is running. Such a proxy is launched automatically by thorn HTTPS for a known list of AEI's PBS clusters (`https://peyote.aei.mpg.de:24000/` in the example above). Via parameters in the parameter file the user can also manually set the name of the proxy host and provide an available port range; HTTPS will then use this information and launch the web proxy accordingly.

2.2 Interactive Simulation Control and Online Steering of Parameters

As any other Grid service, thorn HTTPS uses X.509 certificates for user authentication and authorisation: when a user connects to a running Cactus simulation, as described in the previous section, the webserver thorn HTTPS asks the connecting webclient to present a user certificate from which the simulation can find out who is connecting to it, and whether (s)he is allowed to do so.

This standard method of user authentication is optional for just monitoring the status of a simulation; it can be turned off by the owner of the simulation in order to allow anybody (even people without a user certificate) to connect. However, it is required that every user, who also wants to control the simulation and steer parameters, *must* present a valid certificate which matches an entry in the list of authorised users with steering privileges as defined by the owner of the simulation in the parameter file, like in the following parameter file excerpt:

```
HTTPS::authorised_users      = "^/O=GermanGrid/OU=AEI/CN="
HTTPS::authorised_superuser = "^/O=GermanGrid/OU=AEI/CN=Thomas Radke$"
```

One would typically allow all users of a collaboration (in this case all astrophysicists from AEI with a GermanGrid user certificate) to connect to and access a simulation with monitoring privileges;

only the owner of a simulation would add her/his own certificate to the list of *superusers* in order to grant her/him also steering privileges. Both user authentication parameters of thorn HTTPS are *steerable* which means they can be changed during runtime. Therefore a user with steering privileges can dynamically add or remove other people from the list of authorised users to monitor and/or steer a simulation.

Simulation Home

Environment:
Time: 18:57:41+0200
Date: Oct 26 2007

Simulation:
Neutron Star Collapse
ns.par
Iteration: 0
Physical time: 0

Options:
[Simulation Control](#)
[Parameters](#)
[Thorns](#)
[Groups and Variables](#)
[New: Logfiles](#)
[Visualisation](#)

Check/Modify Parameters

On this page you can check the values of parameters used by this simulation.

Parameters can be either searched for and listed by name (or a part thereof), or displayed as a complete parameter set for the thorn which defines them.

Parameters which have been designated as *steerable* are shown in a form. To modify steerable parameters, just edit their value and then hit **Enter** or click the corresponding submit button.

Find parameters matching:

HTTPS::authorised_superusers	/CN=Thomas Radke	List of distinguished names of users authorised to login with monitoring and steering privileges
HTTPS::authorised_users	^/O=GermanGrid/OU=	List of distinguished names of users authorised to login with monitoring privileges

Display all parameters for a selected thorn / implementation:

Thorn	Implementation
Cactus	Cactus
CoordBase	CoordBase
HTTPS	HTTPS
IOUtil	IO
PUGH	Driver
Socket	Socket
Visualisation	Visualisation

You are logged in as *Thomas Radke* [About this Web Server](#)

Figure 1: Snapshot of the *Parameter Steering* page of thorn HTTPS

Interactive steering of parameters is done on HTTPS's *Parameter Steering* webpage (see figure 1) on which the user can search for specific parameters (by specifying a regular expression filter string) or follow on to the list of all parameters for an active Cactus thorn.

A more specific method to dynamically interact with the simulation and change its runtime status is provided on a *Simulation Control* page. Here the user can conveniently select – via a customised HTML formular – to pause a running simulation, single-step to the next iteration (eg. for debugging purposes), continue it, or terminate the run after the current iteration in a controlled way, possible triggering a termination checkpoint to be generated from which the simulation may then be restarted at a later time.

Each parameter steering request by a user (that is, the full name of the user as obtained from her/his certificate, the name and new the value of the parameter, and the simulation time at which the request was processed) is also logged by thorn HTTPS in the simulation's stdout logfile so that it is possible for other users to query this change of state and potentially reproduce the dynamic behaviour in a different run.

2.3 Access to temporary stdout/stderr Logfiles

One of the requirements of most AstroGrid-D use cases on job/application monitoring[1, 2] was that users can follow the progress of their running Grid jobs by watching the `stdout/stderr` logfiles. When jobs are not run interactively on the Grid but instead submitted as batch jobs, the problem arises that, while the job is running, its `stdout/stderr` messages are typically written into temporary logfiles determined by the local queuing system, and moved only afterwards to their final destination (eg. as specified in a job description). Depending on the configuration of the local queuing system the names and locations of these temporary logfiles are site-specific and thus may differ from Grid resource to Grid resource.

In order to access the temporary `stdout/stderr` logfiles, a service is necessary which has some knowledge about the local queuing system, queries at runtime its type and configuration parameters, and derives from this information the concrete location and names of the files. For Cactus, this functionality has been implemented in thorn HTTPS as a *Logfile Access* webpage (see figure 2).

The screenshot shows a web browser window with the URL `https://deg001.hpc.lsu.edu:5555/Logfiles/index.html?maxLines=20&log=stdout&filterBy=`. The page content includes:

- Simulation Home** sidebar with links: Environment, Simulation (Neutron Star Collapse), Options, Simulation Control, Parameters, Thorns, Groups and Variables, New: Logfiles, Visualisation.
- Environment:** Time: 11:10:21CDT, Date: Oct 22 2007.
- Simulation:** Neutron Star Collapse, https.par, Iteration: 0, Physical time: 0.
- Options:** Simulation Control, Parameters, Thorns, Groups and Variables, New: Logfiles, Visualisation.
- Simulation Logfiles** main heading.
- Text: "Below you can see the most recent lines of output in the simulation's stdout/stderr logfiles:"
- Log file paths: `/home/tradke/cactus/par/test.out` and `/home/tradke/cactus/par/test.err`.
- Text: "You can specify the maximum number of lines to display and a filter string to search in the logfile."
- Form: "show most recent 20 lines of stdout & stderr" with an "update page" button.
- Form: "filter by" with an "ignore case" checkbox.
- Log output snippet:


```
[CCTK_PARAMCHECK]
Boundary: Check dimension of grid variables
CartGrid3D: Check coordinates for CartGrid3D
IDScalarWaveC: Check parameters

Initialisation
[CCTK_BASEGRID]
CartGrid3D: Set up ranges for spatial 3D Cartesian coordinates
CartGrid3D: Set up spatial 3D Cartesian coordinates on the GH
PUGH: Report on PUGH set up
SymBase: Check whether the driver set up the grid consistently
WaveToyC: Schedule symmetries
if (NOT (recover initial data AND recovery_mode is 'strict'))
[CCTK_INITIAL]
IDScalarWaveC: Initial data for 3D wave equation
IOBasic: Initialisation routine
[CCTK_POSTINITIAL]
[CCTK_POSTSTEP]
HTTPS: Working routine
endif
if
```
- Footer: "You are logged in as Thomas Radke" and "About this Web Server".

Figure 2: Snapshot of HTTPS's *Logfile Access* page for a Cactus simulation running on a super-computer in the US-american LONI network

At simulation startup, the thorn will determine from its shell environment whether it was started as a batch job via a local queuing system. If so, it checks the type of queuing system (currently PBS/OpenPBS/Torque and LoadLeveler are supported), assembles the names of the corresponding temporary `stdout/stderr` logfiles, and verifies that they exist. In the case of success, the user

can then follow the simulation's most recent messages to either `stdout` or `stderr`, repeatedly updating the contents simply by reloading the webpage. Through a web form, the maximum number of lines to be displayed can be chosen, as well as an optional string search parameter (as a regular expression) to filter the logfile contents for user-specified keywords.

2.4 Cactus Thorn VISUALISATION

Extensive work has been put into providing useful visualisation methods for the physicists to graphically analyse intermediate simulation data and judge the quality of the results while the application is still running. Driven by feedback from production-mode users in the Cactus community, different visualisation methods for different types of Cactus simulation output data have been identified, designed, and coded. All implemented visualisation methods are bundled together in a single Cactus thorn `ASTROGRID/VISUALISATION`. It uses the webserver infrastructure of thorn `HTTPS` to provide its online data visualisation functionality as dynamically generated webpages.

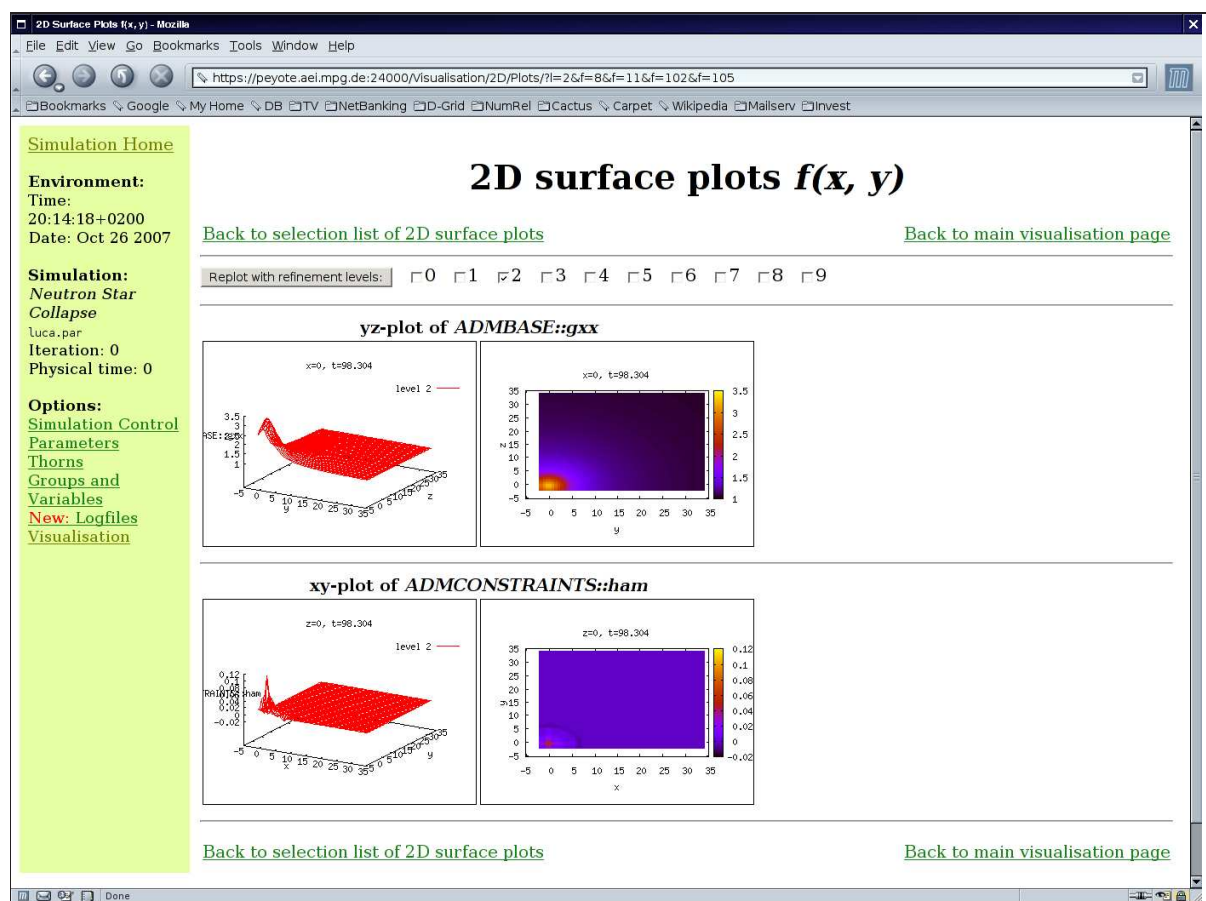


Figure 3: Snapshot of online data visualisation with 2D surface plots

When the first data output has been written to files (typically after initial data generation during simulation setup), thorn `VISUALISATION` will iterate through the known list of all output directories and query the names of output files it is able process. According to their filenames and extensions, all output files are then sorted into type classes of different online visualisations. Currently the following types are implemented:

- scalar data plots $f(t)$
- 1D line plots $f(x)$
- 2D surface plots $f(x, y)$ (see figure 3 as an example)
- Black Hole diagnostics
- gravitational wave extraction
- timing statistics

Each type is represented as a separate visualisation webpage which the user can choose on the main visualisation page. For some of these visualisation types, the user can also select from a table which quantities (eg. a specific (group of) variables, a certain norm, or a list of detector output data for gravitational wave extraction) should be visualised in the following step.

In order to actually generate images from the simulation's output data, the standard visualisation package `gnuplot`[6] is invoked internally by thorn `VISUALISATION`: it compiles a script of specific `gnuplot` commands to visualise the output data of the given visualisation type class and then pipes this script into `gnuplot` which in turn will generate one or more PNG images from it. These images are embedded as HTML tags into a dynamically generated webpage which is finally sent back to the user's browser for display.

3 Test Report Summary

The Cactus simulation monitoring & steering methods described in section 2 have been thoroughly tested so far on cluster production machines of the *Numerical Relativity* group at AEI and their collaborators at the Center for Computation and Technology (CCT) at Louisiana State University, Baton Rouge, USA. The Cactus thorns `HTTPS` and `VISUALISATION` which implement this functionality were made available to the Cactus developers community in May 2007, and have been used since then by several Cactus users in their production-mode simulation runs.

3.1 Specific Test Results

In order to use the simulation steering features provided by thorn `HTTPS`, Cactus users had to obtain a valid grid certificate. This was done using the newly deployed D-Grid VOMRS service where they registered as new members of the AstrGrid-D community and requested a *GermanGrid* certificate issued by the Forschungszentrum Karlsruhe. The process of applying for a grid certificate in VOMRS went smoothly; however it was found that the user interface could be improved to provide *simple-to-understand* information and guidelines for non-Grid users. A corresponding feature request was sent back to the VOMRS maintainers.

The proxy webserver solution in thorn `HTTPS` was gradually deployed and – for the AEI clusters (which are mostly used by the *Numerical Relativists* at AEI) – has been automated in such a way that users do not have to manually set any proxy parameters in their Cactus parameter files. The automatic detection by `HTTPS` of resources where Cactus is running on has simplified the usability of the thorn substantially.

The online visualisation functionality implemented in thorn `VISUALISATION` has been developed in close contact with the Cactus community. Starting from the knowledge about standard post-processing and analysis steps of Cactus simulation data, several prototype versions of online visualisation methods were built, incorporating the users' feedback on which methods are most useful, what type classes should the output data be sorted in, and how the generated images should be presented. Because users typically have their own individual style of analysing and visualising data from their simulations, a compromise had to be found in the end between personal configurability of thorn `VISUALISATION` on one hand versus code complexity and simplicity of use on the other hand.

The *Simulation Control* page was improved based on the experience and feedback of users running production simulations on a day-to-day basis. Simply by adding small features such as a *Checkpoint Next* and a *Checkpoint on Terminate* toggle, the web interface for Cactus simulation control already received an added value which turned out to be very useful for power users.

`HTTPS`'s logfile access functionality has been tested on all of AEI's PBS clusters² and on LRZ's HLRB2 supercomputer³ in Germany as well as on various Torque/LoadLeveler systems in the LONI network⁴ and TeraGrid⁵ in the U.S.A. Since queuing system configurations are typically site-specific, adoptions of the code may be necessary when new resources should be supported too.

²<http://supercomputers.aei.mpg.de/>

³<http://www.lrz-muenchen.de/services/hpc/hlrb/intro/>

⁴<http://www.loni.org/systems/>

⁵<http://www.teragrid.org/userinfo/hardware/index.php>

3.2 Dissemination

The implemented monitoring & steering functionality has been presented during several practical demonstrations of the code and in face-to-face hand-on sessions with the Cactus users at AEI. Through the official *Cactus Users* and *Cactus Developers* mailing lists, the work has also been disseminated to the collaborating *Numerical Relativity* group at the CCT as well as other external physicists in the Cactus community.

F: References / Bibliography

References

- [1] AstroGrid-D Use Case inquiry. AstroGrid public webpage;
<http://www.gac-grid.org/project-documents/UseCases.html>
- [2] Thomas Radke: Requirements on grid-enabled Monitoring & Steering Methods in AstroGrid-D Applications. Requirements Specification, Work Group VI deliverable document D6.2, AstroGrid project;
http://www.gac-grid.org/project-documents/deliverables/wp6/WG6_D6_2.pdf
- [3] Thomas Radke: Architecture of generic grid-enabled Monitoring & Steering Methods in AstroGrid-D Applications. Architecture Specification, Work Group VI deliverable document D6.3, AstroGrid project;
http://www.gac-grid.org/project-documents/deliverables/wp6/WG6_D6_3.pdf
- [4] Thomas Radke: Prototype Implementation of grid-enabled Monitoring Methods. Documentation and Test Report, Work Group VI deliverable document D6.4, AstroGrid project;
http://www.gac-grid.org/project-documents/deliverables/wp6/WG6_D6_4.pdf
- [5] Thomas Radke: Status WG6. Presentation at the 8th AstroGrid Project Meeting, 12. September 2007, Göttingen;
<http://www.gac-grid.org/project-overview/events-meetings/meetings/meetinggoettingen-1/WG6-Report-09-2007.pdf>
- [6] gnuplot homepage <http://www.gnuplot.info/>