



IST-2001-32133

GridLab - A Grid Application Toolkit and Testbed

## D2.1 CGAT Functionality and Implementation Plan

---

Author(s):	Gabrielle Allen, Tom Goodale
Document Filename:	Gridlab-2-D2.1-0001.FunctionalityAndImplementationPlan
Work package:	Grid Application Toolkit
Partner(s):	PSNC,MPG,ZIB,MU,VU,CARDIFF
Lead Partner:	MPG
Config ID:	GridLab-2-D2.1-0001-1.0DRAFT-1
Document classification:	Internal

---

**Abstract:** This document describes the plans for the functionality and implementation of the Cactus GAT toolkit, a set of thorns which provide easy access to the Grid Application Toolkit developed by GridLab WP1





## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose of Document . . . . .	2
1.2	Structure of Document . . . . .	2
1.3	Status of this Document . . . . .	2
1.4	RFC 2119 and this Document . . . . .	2
<b>2</b>	<b>Functionality</b>	<b>2</b>
<b>3</b>	<b>Implementation Plan</b>	<b>4</b>
3.1	Design of the Cactus GAT Toolkit . . . . .	4
3.2	CGATBase . . . . .	4
3.3	IOGAT . . . . .	4
3.4	CGATSpawn . . . . .	4
3.5	Surrounding Infrastructure . . . . .	4
<b>A</b>	<b>Procedures</b>	<b>5</b>
A.1	Coding Guidelines . . . . .	5
A.2	Version Control . . . . .	5
A.3	Bug/Issue Tracking . . . . .	5
<b>B</b>	<b>Task List</b>	<b>5</b>

## 1 Introduction

GridLab WP2's goal is to specify enable applications written in the Cactus [1] framework to make use of the Grid-Application-Toolkit-API (GAT-API) for sophisticated Grid related scenarios. Cactus has been used for many years for in Grid environments, and is commonly deployed for demonstrations of the use of the Grid. Until now, all functionality to deal with Grid environments has been hand-crafted in Cactus modules, using ad-hoc mechanisms, relying on the authors' extensive experience of Grid computing. The CGAT work aims to replace this work with calls to the functionality available through the GAT API.

The GAT API provides an abstraction layer which abstracts operations which may need to be Grid-aware in such a way that an application may be developed and run independently of the underlying middle-ware actually deployed.

### 1.1 Purpose of Document

This document provides details of the functionality and implementation plans for the Cactus GAT toolkit thorns.

### 1.2 Structure of Document

This document is split into two main sections:

section 2 details the necessary functionality which the CGAT must enable.

section 3 details the specific plans and designs which will be used..

### 1.3 Status of this Document

This document is in a preliminary stage. While it identifies tasks, it does not give concrete time schedules for these tasks. This will be amended in a later release of this document.

### 1.4 RFC 2119 and this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [2].

## 2 Functionality

The general concept for the CGAT is illustrated in figure 1.

Core items:

- Checkpoint
- Export the list of created files (as per HTTPDExtra) via GAT advert and/or replica functionality.
- Get information about the current machine; e.g. cache size, memory size, size of filesystems, name of machine.
- Spawn tasks, e.g. for task farming, and monitor their status

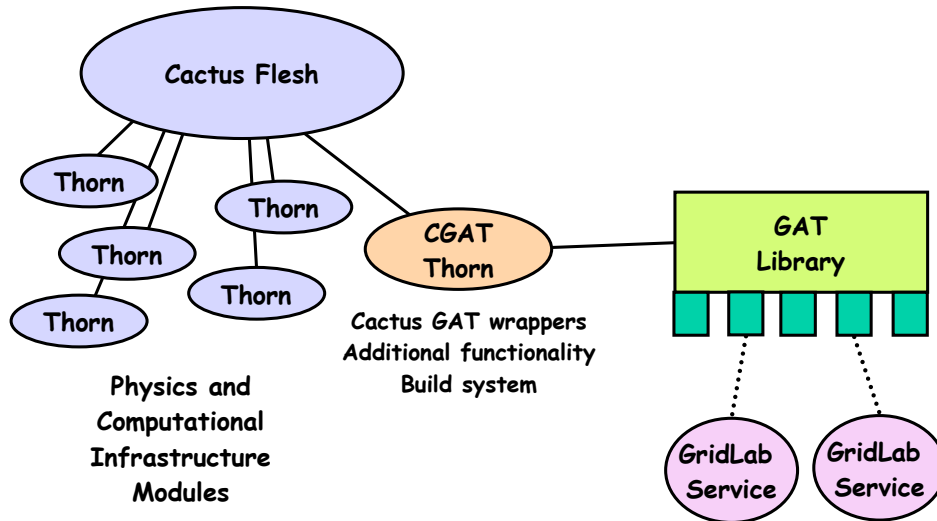


Figure 1: Conceptual model of the CGAT

- Communicate between tasks — open and use connections, stream or copy files.
- Use HTTPD functionality securely via a GAT pipe.
- Replace current announcement capabilities with stuff automatically available through the GAT.
- Provide performance and other data to external applications using GAT monitoring infrastructure.
- Get current job ID, or IDs of other jobs for archival purposes.
- Do I/O from/to remote files.
- Get file information, such as size or permissions
- Change remote file access permissions.

Other items:

- Mechanism to tar up/otherwise archive files
- Provide ways for applications running on a cluster to be able to spawn tasks hush within the nodes allocated to this job; e.g. SC2002 task farming scenario.
- Infrastructure for getting appropriate code to resources and compiling there (in collaboration with GRNS and Testbed groups).

The CGAT must be deployable on all machines on which Cactus is used; in particular:

- IBM SPs running AIX
- Hitachi SR8000s
- NEC SX-5 and SX-6
- SGI Irix 32 and 64 bit machines
- IA32, IA64, alpha, PowerPC, PS2, Samsung running Linux
- Apple OS X
- Compaq alpha running Tru64
- Windows NT, 2000

### **3 Implementation Plan**

#### **3.1 Design of the Cactus GAT Toolkit**

- Which thorns there will be and how the functionality is split across them.
- Diagram showing relationships
- Any surrounding infrastructure

#### **3.2 CGATBase**

- Main CGAT thorn
- Initialises and finalises GAT.
- Contains make stuff which allows Cactus to link to GAT
- Contains parameters which are used to control the GAT and functions to deal with adaptors.
- May contain checkpointing and monitoring stuff.

#### **3.3 IOGAT**

- Thorn to provide remote file I/O capabilities
- Will need to build upon CDAC IO-callback stuff

#### **3.4 CGATSpawn**

- Thorn to spawn remote tasks and manage them
- Would be used by Task farming infrastructure

#### **3.5 Surrounding Infrastructure**

- Infrastructure for deploying correct code

## A Procedures

### A.1 Coding Guidelines

The GridLab coding guidelines will be used in all source files.

### A.2 Version Control

Source code and document development will make use of CVS for version control and change tracking. The GridLab CVS server will be used.

Version numbering in releases will follow the standard open-source policy of a set of three numbers — major, minor, micro – of which odd numbers in the minor indicate a development sequence and even numbers indicate a stable sequence which consists of bug-fixes only.

### A.3 Bug/Issue Tracking

The GridLab Bugzilla server will be used to track the status of bugs, feature requests and other issues.

## B Task List

This is a preliminary task list, when the time-scales have been decided, this table will be updated with dates.

Task	Status
Implementation: CGATBase	
Implementation: IOGAT	
Implementation: CGATSpawn	
Implementation: Surrounding infrastructure	
Documentation: CGAT functions	
Documentation: Example scenarios	

## References

- [1] T. Goodale and G. Allen and G. Lanfermann and J. Massó and T. Radke and E. Seidel and J. Shalf The Cactus Framework and Toolkit: Design and Applications *Vector and Parallel Processing - VECPAR'2002, 5th International Conference, Lecture Notes in Computer Science*, Springer, Berlin, 2003
- [2] S.Bradner, “RFC 2119: Key words for use in RFCs to Indicate Requirement Levels”, <http://www.ietf.org/rfc/rfc2119.txt>
- [3] Allen, G., Seidel, E., and Shalf, J.: (2002). Scientific Computing on the Grid. *Byte*, **Spring**:24–32.
- [4] Cactus: <http://www.cactuscode.org>.
- [5] *Cactus Users Guide*  
<http://www.cactuscode.org/Guides/Stable/UsersGuide/UsersGuideStable.pdf>.

- [6] Allen, G., Benger, W., Dramlitsch, T., Goodale, T., Hege, H., Lanfermann, G., Merzky, A., Radke, T., and Seidel, E.: (2001). Cactus Grid Computing: Review of Current Development. In R. Sakellariou, J. Keane, J. Gurd, and L. Freeman, editors, *Europar 2001: Parallel Processing, Proceedings of 7th International Conference Manchester, UK August 28-31, 2001*. Springer. <http://www.cactuscode.org/Papers/Europar01.ps.gz>.
- [7] Anninos, P., Camarda, K., Massó, J., Seidel, E., Suen, W.-M., and Towns, J.: (1995). Three-dimensional numerical relativity: The evolution of black holes. *Phys. Rev. D*, **52**(4):2059–2082.
- [8] Walker, P.: (1998). *Horizons, Hyperbolic Systems, and Inner Boundary Conditions in Numerical Relativity*. Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois.
- [9] Allen, G., Dramlitsch, T., Foster, I., Karonis, N., Ripeanu, M., Seidel, E., and Toonen, B.: (2001). Supporting Efficient Execution in Heterogeneous Distributed Computing Environments with Cactus and Globus. In *Proceedings of Supercomputing 2001, Denver, USA*. <Http://www.cactuscode.org/Papers/GordonBell2001.ps.gz>.
- [10] Shalf, J. and Bethel, E. W.: Cactus and Visapult: An Ultra-High Performance Grid-Distributed Visualization Architecture Using Connectionless Protocols. *Submitted to IEEE Computer Graphics and Animation*.
- [11] Camarda, K., He, Y., and Bishop, K. A.: (2001). A Parallel Chemical Reactor Simulation Using Cactus. In *Proceedings of Linux Clusters: The HPC Revolution, NCSA 2001*.
- [12] Allen, G., Goodale, T., Massó, J., and Seidel, E.: (1999). The Cactus Computational Toolkit and Using Distributed Computing to Collide Neutron Stars. In *Proceedings of Eighth IEEE International Symposium on High Performance Distributed Computing, HPDC-8, Redondo Beach, 1999*. IEEE Press.
- [13] Allen, G., Benger, W., Goodale, T., Hege, H., Lanfermann, G., Merzky, A., Radke, T., Seidel, E., and Shalf, J.: (2000). The Cactus Code: A Problem Solving Environment for the Grid. In *Proceedings of Ninth IEEE International Symposium on High Performance Distributed Computing, HPDC-9, August 1-4 2000, Pittsburgh*, pages 253–260. IEEE Press. [http://www.cactuscode.org/Papers/HPDC9\\_2000.ps.gz](http://www.cactuscode.org/Papers/HPDC9_2000.ps.gz).
- [14] Benger, W., Foster, I., Novotny, J., Seidel, E., Shalf, J., Smith, W., and Walker, P.: (March 1999). Numerical Relativity in a Distributed Environment. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*. [http://www.zib.de/visual/projects/TIKSL/Papers/numrel\\_dist.ps](http://www.zib.de/visual/projects/TIKSL/Papers/numrel_dist.ps).
- [15] Allen, G., Benger, W., Goodale, T., Hege, H., Lanfermann, G., Merzky, A., Radke, T., Seidel, E., and Shalf, J.: (2001). Cactus Tools for Grid Applications. *Cluster Computing*, **4**:179–188. <http://www.cactuscode.org/Papers/CactusTools.ps.gz>.