

# Community Software Development with the Astrophysics Simulation Collaboratory

<sup>5</sup>Gregor von Laszewski, <sup>2</sup>Michael Russell, <sup>2,5</sup>Ian Foster, <sup>3,4</sup>John Shalf, <sup>1</sup>Gabrielle Allen, <sup>4</sup>Greg Daues, <sup>3</sup>Jason Novotny, <sup>1,4</sup>Edward Seidel

*1 Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, Golm, Germany*

*2 University of Chicago, Chicago, IL, U.S.A.*

*3 Lawrence Berkeley National Laboratory, Berkeley, CA, U.S.A.*

*4 National Center for Supercomputing Applications, Champaign, IL, U.S.A.*

*5 Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 60439, U.S.A.*

*6 Washington University, St Louis, MI, U.S.A.*

*[addresses to be completed in final submission]*

Corresponding author: Gregor von Laszewski: [gregor@mcs.anl.gov](mailto:gregor@mcs.anl.gov), phone 630 252 0472, fax 630 252 5896

This paper addresses issues given by the template developed by the GCE WG for the journal. The paper can be shortened if necessary.

Submitted to:

*Journals Production Department, John Wiley & Sons, Ltd.,  
Chichester, West Sussex, PO19 1UD, U.K.*

---

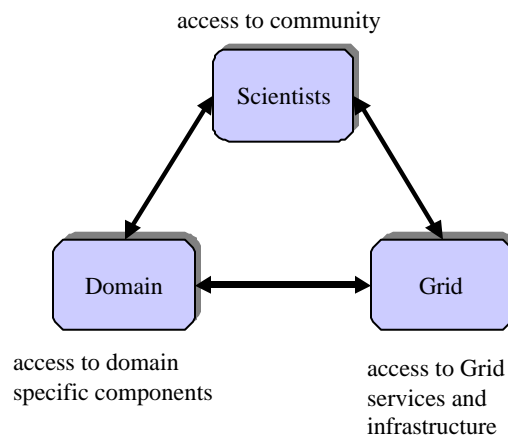
## SUMMARY

**We describe a Grid-based collaboratory that supports the collaborative development and use of advanced simulation codes. Our implementation of this collaboratory uses a mix of Web technologies (for thin-client access) and Grid services (for secure remote access to, and management of, distributed resources). Our collaboratory enables researchers in geographically dispersed locations to share and access compute, storage, and code resources, without regard to institutional boundaries. Specialized services support community code development, via specialized Grid services, such as online code repositories. We use this framework to construct the Astrophysics Simulation Collaboratory, a domain-specific collaboratory for the astrophysics simulation community. This Grid-based collaboratory enables researchers in the field of numerical relativity to study astrophysical phenomena by using the Cactus computational toolkit.**

KEY WORDS: Collaboratory, Grid Computing, Globus, Cactus, Java CoG Kit, Astrophysics Simulation Collaboratory

## 1 INTRODUCTION

The solution of complex problems often requires coordinated effort by geographically distributed participants. The term *collaboratory* was created in the early 1990s to denote an infrastructure supporting such collaborative endeavors. [3] Typically, collaboratories have emphasized interpersonal communication and collaborative access to relatively simple resources. We are interested in extending the collaboratory concept to address problem-solving methods and environments that require the sharing and coordinated use of large numbers of distributed resources. In this context, we see considerable value for *Grid* technologies, which address authentication, resource discovery, and resource access issues that arise when sharing distributed resources. Grids emerged over the past decade as a distributed computing infrastructure to promote the sharing of information, knowledge, and expertise across institutional and disciplinary boundaries [14, 15]. We term this combination of collaborative and Grid technologies a *Grid-based collaboratory* (see Figure 1).



**Figure 1:** A Grid-based collaboratory enables domain scientists to share geographically distributed resources and collaborate on the development of new knowledge.

Although the results of our research can be applied to a variety of domains, we focus our attention here on computational astrophysics, a discipline that has challenging requirements for large-scale collaborative development and application of simulation,

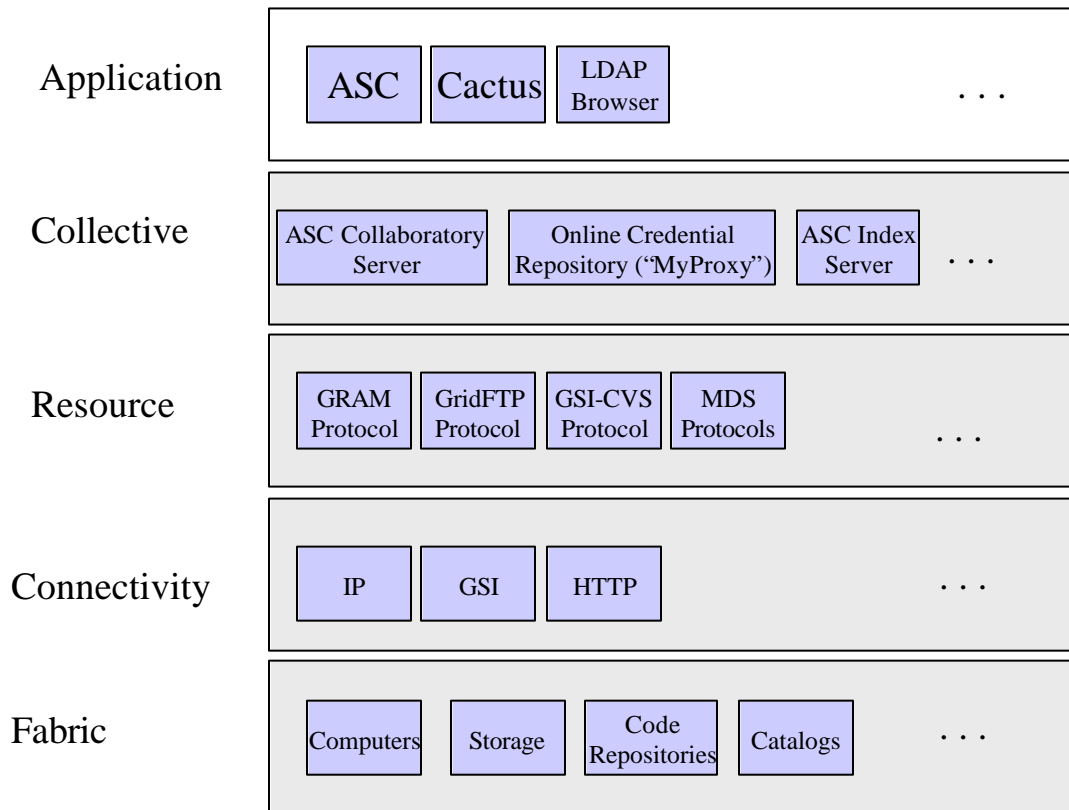
data analysis tools. For example, a typical simulation experiment might involve the collaborative creation of a simulation code from a dozen distinct components, developed by different groups; the acquisition of computer resources at one or more remote sites; the collaborative monitoring and steering of the computation as it runs; and then collaborative access to data resources at one or more output locations. Our goal is to develop an *astrophysics simulation collaboratory* (ASC) that allows astrophysics and computational science researchers to cooperate as they develop and share software, computational, and data resources.

Discipline scientists are typically no experts in distributed computing. Hence, it is necessary that our collaboratory hide as much as possible of the complexity of using multiple remote resources. In the past decade Web technologies have been accepted by many scientific communities as a means of distributing and accessing services. Web pages present the information needed in an orderly and structured fashion; Web protocols can be used to access remote information and services. Yet at the same time, Web protocols and technologies are inadequate for the sharing and coordination issues that arise when running simulations on remote systems, sharing multiple resources, and so forth. Here, Grid technologies [15] can play an important role. The resulting need to integrate Web and Grid technologies is addressed by Commodity Grid (CoG) Kits [22] that provide a middleware component framework allowing easy integration of Grid services with commodity technologies such as Java, JSP, CORBA, and Python. These CoG Kits promote the reuse of Grid technologies as part of the middleware that is used to architect Web-based Grid collaboratories.

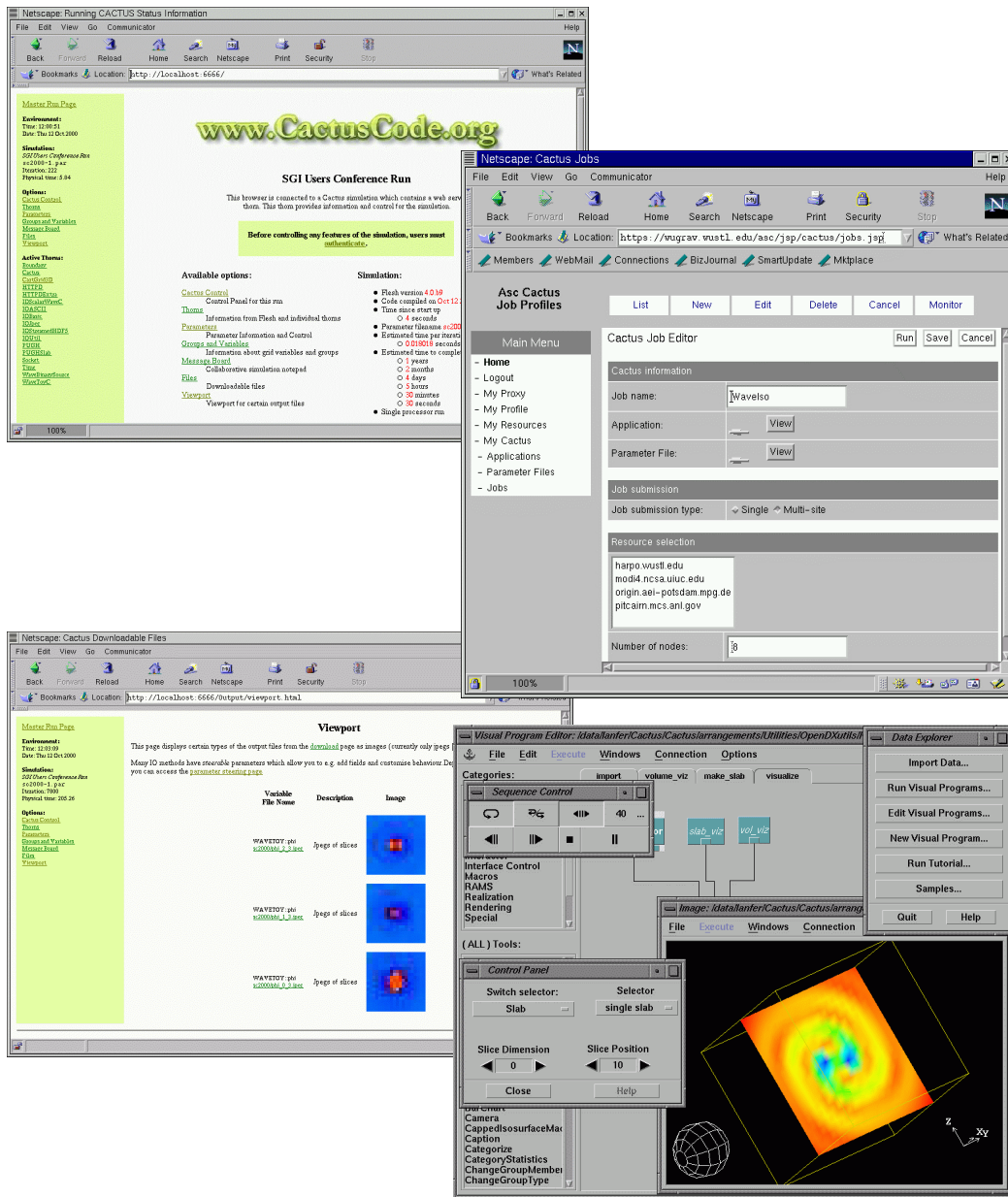
In this paper we describe both the framework, services, and components needed to construct Grid-enabled collaboratories, and the particular application of these technologies within our ASC. The ASC [1, 21] fulfills the following general requirements posed by the astrophysics simulation community:

- **Promote the creation of a community** for sharing and developing simulation codes and scientific results. The Grid infrastructure promotes the creation of virtual organizations [15](VOs) within which resources are shared across institutional boundaries through Grid services.
- **Enable transparent access to remote** resources, including computers, data storage archives, information servers, and shared code repositories.

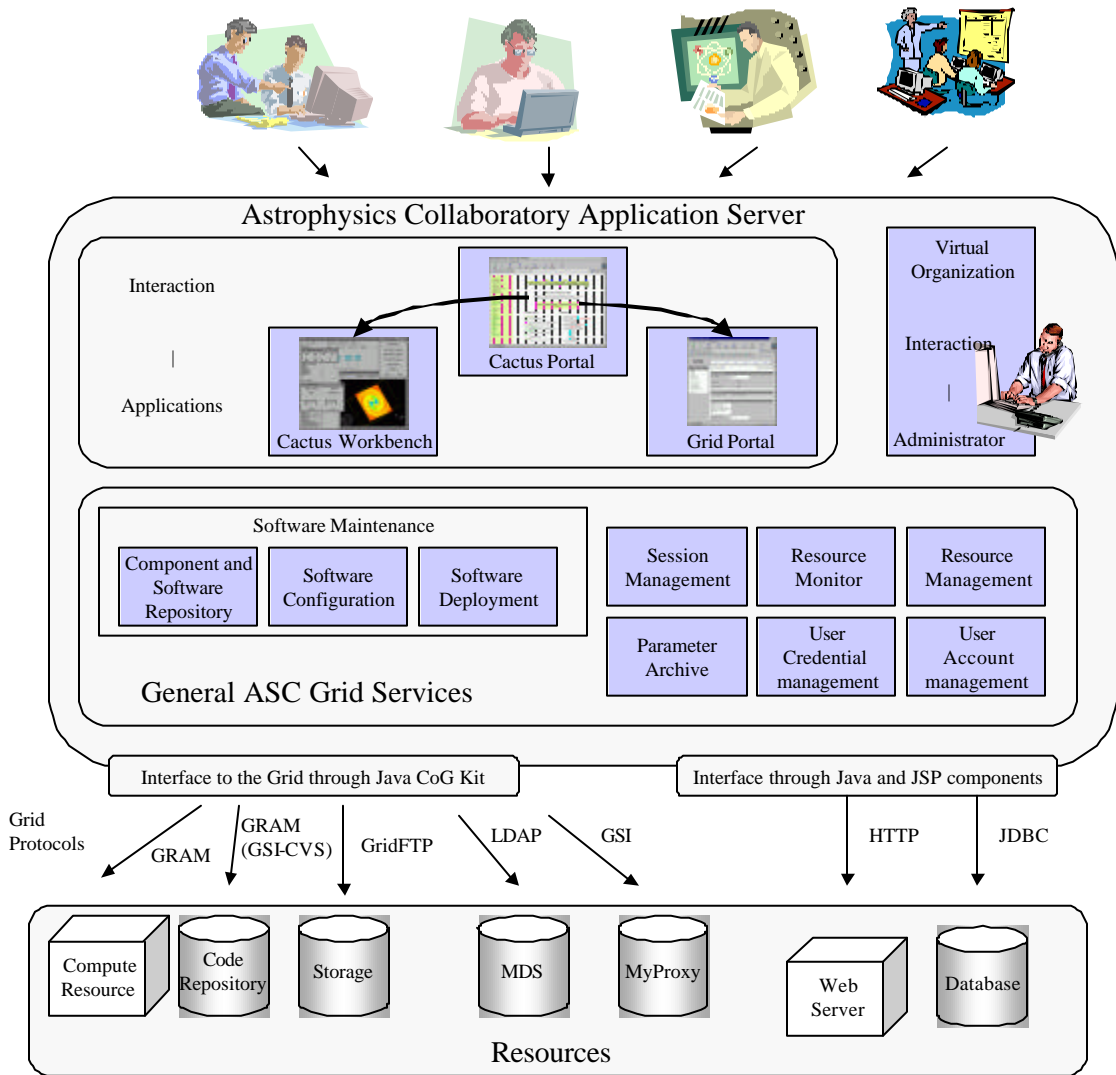
- **Enhance domain-specific component and service development** supporting specific problem-solving capabilities, such as the development of simulation codes for the astrophysical community or the development of advanced Grid services reusable by the community.
- **Distribute and install programs onto remote resources** while accessing code repositories, compilation, and deployment services.
- **Enable collaboration during program execution** to foster interaction during the development of parameters and the verification of the simulations.
- **Enable shared control and steering of the simulations** to support asynchronous collaborative techniques among collaboratory members.
- **Provide access to domain-specific clients** that, for example, enable access to multimedia streams and other data generated during the execution of the simulation.



**Figure 2:** A protocol-oriented view of Grid architecture, showing the principal components used to build the Astrophysics Collaboratory (ASC). Various components, in particular the ASC Application Server, are described in more detail in the text.



**Figure 3:** The look and feel of the ASC collaboratory exposed through a Web browser and a customized visualization component.



**Figure 4:** The structure of the ASC Application Server, showing the various components used to meet the requirements identified in the text.

## 2 ARCHITECTURE OVERVIEW

As illustrated in Figure 2, the ASC has a modular structure and makes heavy use of Grid protocols and services to access and manage distributed resources. The principal components and protocols used are as follows:

- Web browsers serve as the primary end-user interface, serving dynamic HTML (D-HTML) and Java applets obtained from the ASC Application Server. Other end-user applications are also used: for example, for more advanced visualization functions.
- The ASC Application Server is the middle-tier entity that implements most ASC-specific logic, interacting with:
  - other Grid services for purposes of resource discovery and authentication;
  - with resources to initiate and manage simulation and/or access and analyze data, and
  - with web browsers and other applications for user interaction.
- An online credential repository, MyProxy, [19] is used as a secure cache for user proxy credentials, hence reducing barriers to the use of PKI authentication.
- An ASC Index Server supports registration and subsequent discovery of resources contributed to the ASC, using MDS-2 protocols [10].
- Standard Grid protocols [15] provided by the Globus Toolkit [13] are used to negotiate secure access to remote resources, including CVS code repositories.

The vast majority of ASC-specific logic is incorporated within the ASC Application Server. This component is responsible for handling all interactions with ASC users, for maintaining state associated with user interactions, and for coordinating the use of remote resources to meet user requests. Figure 4 shows its internal structure. Major components include the following

- We make extensive use of commodity Web technologies such as servlets and JSP.
- We use software development kits (SDKs) and APIs provided by the Java CoG Kit to execute the protocols used to access remote resources.



- We use new SDKs developed for this application (but of general utility) to address issues of code management, deployment, and simulation execution on Grid resources.

The final major component of our collaboratory is the Cactus code, [5] a modular simulation development framework that allows us to encourage and leverage the development of reusable components within the astrophysical simulation community.

### 3 TECHNOLOGIES USED

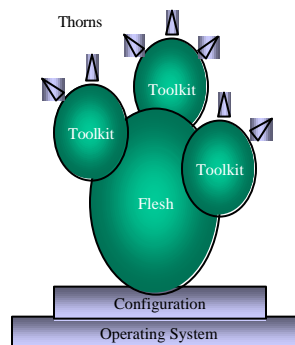
We use three primary technologies to construct our collaboratory: the Cactus framework, to support development of modular simulation codes; the Globus Toolkit, for Grid services; and various Java and Web technologies, to facilitate construction of the ASC Application Server.

#### 3.1 The Cactus Framework: Modular Simulation and Code Creation

Cactus is a modular simulation code framework [6] that uses component technologies to facilitate the construction and management of complex codes. The Cactus project has established its own terminology for naming different components (Figure 5). They term the code that coordinates the activities of modules the *flesh*, and a module performs activities and is coordinated by the flesh a *thorn*.

Thorns interoperate via standard interfaces and communicate directly with the flesh, but not with one another. Hence, the development of thorns can be performed independently of one another, which is essential for coordinating component development in large geographically dispersed research groups while being able to integrate specialized thorns developed by a large variety of groups. This separation of the thorn development promotes, for example, the independent development of *physics thorns* to model the black hole or hydro evolutions, and *information technology thorns* that provide access to I/O, parallelism programming frameworks, or performance monitoring. Thus the Cactus component model permits astrophysicists with different levels of IT expertise to participate in collaborative code development. The Cactus framework also supports sharing and cloning of thorns as part of multiple code repositories. Cactus is distributed with tools that support access to Cactus modules that are generally applicable to a variety of problem domains. An example provides a thorn that supports remote visualization and steering of running simulations through an HTTP

interface to enable seamless integration with Web-based portals. Specialized components for the astrophysics community are readily available. Furthermore, the Cactus framework is used as part of the specialized service to perform astrophysics simulations. This includes distributing and running Cactus simulations on remote resources and is supported on a wide variety of platforms. We use Cactus as part of ASC to create a simulation code assembled from various thorns, contribute new thorns through a community process, and execute and monitor them on remote resources.



**Figure 5:** The Cactus component model. Cactus gets planted on a computer through the use of standardized configuration and compile services.

### 3.2 Globus Grid Services: Enabling Resource Sharing

By using Grid technologies, in our case Globus [12] middleware, we minimize the difficulties introduced by the tremendous variety of resource types, mechanisms, and access policies, and the complex coordination issues (e.g., relating to authentication and authorization), that we would otherwise have to handle ourselves as part of developing a collaboratory.

The Globus Toolkit [12, 15] is a set of services and software libraries that support Grids and Grid applications. The Toolkit includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability. It is packaged as a set of components that can be used either independently or together to develop useful Grid applications and programming tools. Globus Toolkit components include the Grid Security Infrastructure (GSI), [9] which

provides a single-sign-on, run-anywhere authentication service, with support for delegation of credentials to subcomputations, local control over authorization, and mapping from global to local user identities; the Grid Resource Access and Management (GRAM) protocol and service, [11] which provides remote resource allocation and process creation, monitoring, and management services; the Metacomputing Directory Service (MDS) [10, 18], an extensible Grid information service that provides a uniform framework for discovering and accessing system configuration and status information such as compute server configuration, network status, and the locations of replicated datasets; and GridFTP, a high-speed data movement protocol [4]. A variety of higher-level services are implemented in terms of these basic components.

### **3.3 Java Framework and Server Pages: Developing Browser-accessible Thin Clients**

#### *3.3.1 Java Server Pages / Servelets*

To enable rapid prototyping of Web portals as part of ASC, we base much of our Web-based services on the JavaServer Pages (JSP) technology. [16] JSP uses XML-like tags and scriptlets implemented in the Java framework to encapsulate the logic that generates the content for the page. Because of the separation of the page logic from its design, we used Web-based reusable components supporting the rapid development of collaboratory based Web pages. All important logic is embedded in what we call the ASC Application Server (Figure 4). We have exposed the functionality of the application server as a set of reusable JSP components that both provide direct access to Grid services and also implement higher-level services such as monitoring of the compute resources, execution of complex jobs with file staging and cleanup abilities, and maintaining session state between ASC accesses. These components can be reused in other projects.

#### *3.3.2 JDBC*

Within the ASC Application Server we use a relational database to maintain the state of sessions. We express all database operations in terms of Java Database Connectivity (JDBC) [17] calls, which provide a generic interface to Standard Query Language

(SQL)-compliant database management systems. Thus the ASC can support all commercially viable SQL-based servers, all of which are distributed with a JDBC provider. In our current implementation, we use the popular open-source database server MySQL which is free for non-commercial use, well documented, reliable, and provides high performance.

### *Java CoG Kit*

Within the ASC Application Server, we use SDKs provided by the Java Commodity Grid Toolkit [22] to access Globus Toolkit services, as follows:

- **Security:** Access to GSI services and to the MyProxy [19] online credential repository for retrieval of certificates.
- **Information Services:** Access to MDS information services through the Java Naming and Directory Interface (JNDI) [20].
- **Resource Management:** Remote job submission and management through GRAM client APIs and protocols. This allows access to Grid resources including batch-scheduling systems on high performance computers.
- **File Transfer:** Secure access to Grid-enabled FTP servers for file transfer, [4] as well as access to GASS servers [7].

### **3.4 Other Collaboratory Technologies: Collaboration through Commodity Technologies**

We intend to integrate other collaborative technologies such as audio/video conferencing, chat rooms, news, and e-mail groups. We have refrained from doing so in this phase of the project because the access to the Grid and the code repository was of most benefit to the users community we targeted first.

## **4 PORTAL-SPECIFIC SERVICES AND COMPONENTS**

In this section we discuss services and components that we have developed to support the creation of Web-based collaboratories. All security-related services and components use GSI, and all job submission services are based on GRAM.

#### 4.1 Community Services

To foster collaboration among participants, we provide the following services:

- **ASC Index Server.** This Collective service (in the sense of Figure 2) uses MDS-2 mechanisms to maintain an index to the collection of resources for which we want to enable access through the ASC. Resources registered with this service can be discovered by other users through MDS queries.
- **Cactus CVS Repository.** We use a GSI-enabled CVS repository to enable users to acquire and distribute Cactus software onto remote machines. We provide a domain specific Cactus CVS repository ([cvs.cactuscode.org](http://cvs.cactuscode.org)) used by the Cactus development team to make the latest versions of Cactus available to the community.

#### 4.2 User Login Environment Functions

Users must authenticate with the ASC before they can be granted access to ASC resources. We use GSI mechanisms to support single sign on and delegation; we use the MyProxy online credential repository to provide access to proxy credentials when accessing the collaboratory from a Web browser. The ASC also maintains, within the ASC Application Server, persistent state information such as user preferences. The state information will be reliably recorded in a relational database.

#### 4.3 Software Maintenance Functions

As we mentioned above, a primary function of the ASC is to enable collaborative development and remote execution of Cactus applications. We hence provide a number of components to facilitate the maintenance of code and component repositories.

First, we define a new protocol, *Grid CVS*, which supports GSI-authenticated access to remote CVS repositories. We use this protocol to maintain source and component-based repositories under concurrent versioning control.

Second, we incorporate in our ASC a number of *program repositories*, which speak Grid CVS and which we use to store both source and binary versions of Cactus components. An example of such a service is provided at [cvs.cactuscode.org](http://cvs.cactuscode.org).

Third, we provide a number of SDK components for use within the ASC Application Server. These components include program installation functions, which use GridFTP to install previously compiled binaries in a desired user controlled location.

#### 4.4 Runtime Management

Once a program has been created, it must be executed on a particular machine. We provide a number of SDK components for use within the ASC Application Server, including the following:

- **Parameter file management**, which allows maintaining a set of parameter files important for the execution of an application. A database is used to maintain the various parameter files and share them within the collaboratory.
- **Remote simulation execution**, which allows executing domain and nondomain specific Cactus jobs. Cactus together with the GRAM mechanism are the basis for this service.
- **Batch file submission**, which allows using sophisticated batch files as our community needs to be able to deal with pre- and postprocessing stages for job preparation and cleanup. We base this service on GRAM.
- **Tool creation**, which allows wrapping shell commands as new services and can be exposed to other users as service through a repository.

#### 4.5 Grid Information Services

The MDS-based grid information service used within the ASC provides access to a variety of information sources. We have defined various SDKs that provide specialized views on this information (e.g., a locator for storage space) as well as a GridFTP-based remote file browser for inspecting directories located on remote resources.

#### 4.6 Workbench Services

Our user community sometimes requires external programs as add-ons or plugins into the browser. A good example for such a workbench [8] component is the MDS browsers developed by the Java CoG Kit team, demonstrating the use of thin clients

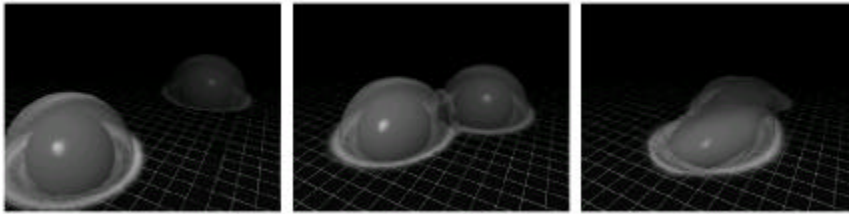
(Perl/CGI), slender clients (as Java applet) , and fat clients (as Java application). The advantage of using slender and fat clients is that they can provide more functionality than the thin client, while at the same time increase response times. Domain-specific examples of such a service include applications, such as Amira and CAVision [2] for streaming and visualizing HDF5 data. The demand on the network traffic posed by these tools is too high to attempt to generate a thin client that provides reasonable response times for user interactions.

## 5 IMPLEMENTATION

Our portal can be accessed from browsers such as Internet Explorer 4.x+ and Netscape 4.x+. The Cactus software runs on almost all major operating systems. A prototype for Windows NT is available. A remote site is brought into the collaboratory simply by implementing the Grid protocols used for remote discovery, job submission, and data transfer; the portable Globus Toolkit implementation of these protocols is available for this purpose. The Web server is based on a secure version of Apache with JSP extensions. Persistent portal session management is achieved via MySQL.

## 6 APPLICATION

ASC was originally developed in support of numerical simulations in astrophysics, and has evolved into a general-purpose code for partial differential equations in three-dimensions. Perhaps the most computationally demanding application that has been attacked with Cactus is the numerical solution of Einstein's general relativistic wave equations, in the context, for example, of the study of neutron star mergers and black hole collisions (Figure 6). The modular design of the Cactus framework and its easy exposure through a Web-based portal, as described in this paper, permits a diverse group of researchers to develop add-on software modules that integrate additional physics or numerical solvers into the Cactus framework.



**Figure 6:** The ASC project investigates astrophysical phenomena involving strong dynamic gravitational fields such as this grazing collision of two neutron stars. The surfaces represent the density of the stellar matter, while the warping of the gridded surface represents the gravitational flux on the plane of the collision.

## 7 FUTURE DIRECTIONS

We plan to improve our collaboratory in several ways, including enhancements to the user interface, advanced visualization support, data management tools, collaboration support, and development of new interaction paradigms.

While testing our framework within the astrophysics users community we have observed limits of our initial thin clients based on DHTML. Therefore, we plan to investigate a wider variety of client-side interfaces including fat client and slender client interfaces. These will also the support of using Java3D as an alternative display framework. In cases where speed is essential, we use fat clients that are traditional monolithic applications, such as the OpenDX visualization tool. In these situations, the ASC Application Server merely acts as a broker for creating direct connections between the client and various Grid services. The application server, in this sense, would serve as a sort of Napster for peer-to-peer interactions between clients and distributed resources.

Another area of development is the creation of robust data management services, such as tools to assist users in finding appropriate storage space on remote computers, archival storage management services, metadata browsing, and automated replication of data to high-performance disk caches to improve visualization system performance. This data management capability is critical for providing access to remote visualization



services as well as sharing and understanding the large volumes of archived results produces by this community of researchers. In the long term we intend to explore alternative modes of interaction with our Grid Portal through asynchronous collaborative methodologies.

## 8 CONCLUSION

We have described an online collaboratory framework that supports the collaborative development and use of numerical simulation codes. This framework uses a combination of familiar Web and Java technologies; Grid services for secure access to and coordinated use of distributed resources; and custom components for collaborative code development and distributed simulation. We have described an application of this framework within the astrophysics community, and suggest that the framework can also be applied within other application domains with similar requirements.

## ACKNOWLEDGMENTS

We thank the Numerical Relativity Group at the Max Planck Institute in Potsdam, Germany for developing the Cactus code. This work was supported in part by the National Science Foundation's Knowledge and Distributed Intelligence project "Astrophysical Simulation Collaboratory", and by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

## REFERENCES

- [1] "ASC Portal Home Page," <http://www.ascportal.org>.
- [2] "LCAVision Homepage," <http://zeus.ncsa.uiuc.edu/~miksa/LCAVision/>.
- [3] *National Collaboratories: Applying Information Technology for Scientific Research*: National Research Council, 1993.

- 
- [4] W. Allcock, I. Foster, S. Tueke, A. Chervenak, and C. Kesselman, "Protocols and Services for Distributed Data-Intensive Science," 2001, [http://www.sdsc.edu/GridForum/RemoteData/Papers/gridftp\\_spec\\_gf5.pdf](http://www.sdsc.edu/GridForum/RemoteData/Papers/gridftp_spec_gf5.pdf)  
<http://www.globus.org/research/papers/ACAT3.pdf>.
- [5] G. Allen, W. Benger, T. Dramlitsch, T. Goodale, H. C. Hege, G. Lanfermann, A. Merzky, T. Radke, E. Seidel, and J. Shalf, "Cactus Tools for Grid Applications," *accepted for publication in Intl. Journal of Cluster Computing*, 2001 <http://www.cactuscode.org/Showcase/Publications.html>.
- [6] G. Allen, W. Benger, T. Goodale, H.C. Hege, G. Lanfermann, J. Masso, A. Merzky, T. Radke, E. Seidel, and J. Shalf, "Solving Einstein's Equations on Supercomputers," *IEEE Computer*, pp. 52-59, 1999 <http://www.cactuscode.org>.
- [7] J. Bester, I. Foster, C. Kesselman, J. Tedesco, and S. Tuecke, "GASS: A Data Movement and Access Service for Wide Area Computing Systems," in *Proc. IOPADS'99*, 1999.
- [8] K. Bishop and J. Alameda, "A Prototypical Example of the Chemical Engineer's Workbench," presented at AAAS/SURA/NCSA Meeting on Networking Resources for Collaborative Research in the Southeast, 1998.
- [9] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, and V. Welch, "Design and Deployment of a National-Scale Authentication Infrastructure," *IEEE Computer*, vol. 33, pp. 60-66, 2000.
- [10] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid Information Services for Distributed Resource Sharing," presented at Proc. 10th IEEE International Symposium on High Performance Distributed Computing, 2001.
- [11] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, "A Resource Management Architecture for Metacomputing Systems," presented at Proceedings of IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, 1998.
- [12] I. Foster and C. Kesselman, "The Globus Project: A Status Report," in *Proc. Heterogeneous Computing Workshop: IEEE Press*, 1998, pp. 4-18.
- [13] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *International Journal of Supercomputer Applications*, vol. 11, pp. 115--128, 1997 <http://www.globus.org>.
- [14] I. Foster and C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure," Morgan Kaufmann, 1999.

- [15] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Intl. J. Supercomputer Applications*, vol. (to appear), 2001 <http://www.globus.org/research/papers/anatomy.pdf>.
- [16] M. Hall, *Core Servlets and JavaServer Pages (JSP)*, 1 edition ed: Prentice Hall PTR/Sun Microsystems Press, 2000.
- [17] G. Hamilton, R. G. G. Cattell, and M. Fisher, *JDBC database access with Java : a tutorial and annotated reference*. Reading, Mass.: Addison-Wesley, 1997.
- [18] G. v. Laszewski, S. Fitzgerald, I. Foster, C. Kesselman, W. Smith, and S. Tuecke, "A Directory Service for Configuring High-Performance Distributed Computations," in *Proc. 6th IEEE Symp. on High-Performance Distributed Computing*, 1997, pp. 365-375.
- [19] J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy," presented at to be published in HPDC, 2001.
- [20] S. S. Rosanna Lee, *JNDI API Tutorial and Reference: Building Directory-Enabled Java Applications*: Addison-Wesley, 2000.
- [21] M. Russell, G. Allen, G. Daues, I. Foster, T. Goodale, E. Seidel, J. Novotny, J. Shalf, W.-M. Suen, and G. V. Laszewski, "The Astrophysics Simulation Collaboratory: A Science Portal Enabling Community Software Development," presented at Submitted to the Tenth IEEE International Symposium on High Performance Distributed Computing (HPDC10), San Francisco, 2001.
- [22] G. von Laszewski, I. Foster, J. Gawor, and P. Lane, "A Java Commodity Grid Kit," *Concurrency and Computation: Practice and Experience*, vol. 13, pp. 643-662, 2001 <http://www.globus.org/cog/documentation/papers/cog-cpe-final.pdf>.