Cluster Computing 5, 297–304, 2002
© 2002 Kluwer Academic Publishers. Manufactured in The Netherlands.

The Astrophysics Simulation Collaboratory: A Science Portal Enabling Community Software Development

MICHAEL RUSSELL
University of Chicago, Chicago, IL, USA

GABRIELLE ALLEN
Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, Golm, Germany

GREG DAUES
National Center for Supercomputing Applications, Champaign, IL, USA

IAN FOSTER
University of Chicago, Chicago, IL, USA
Argonne National Laboratory, Chicago, IL, USA

EDWARD SEIDEL
Max-Planck-Insitut für Gravitationsphysik, Albert-Einstein-Institut, Golm, Germany
National Center for Supercomputing Applications, Champaign, IL, USA

JASON NOVOTNY
Lawrence Barkeley National Laboratory, Berkeley, CA, USA

JOHN SHALF
Lawrence Barkeley National Laboratory, Berkeley, CA, USA
National Center for Supercomputing Applications, Champaign, IL, USA

GREGOR VON LASZEWSKI
Argonne National Laboratory, Chicago, IL, USA

Abstract. Grid Portals, based on standard web technologies, are emerging as important and useful user interfaces to computational and data Grids. Grid Portals enable Virtual Organizations, comprised of distributed researchers to collaborate and access resources more efficiently and seamlessly. The Astrophysics Simulation Collaboratory (ASC) Grid Portal provides a framework to enable researchers in the field of numerical relativity to study astrophysical phenomenon by making use of the Cactus computational toolkit. We examine user requirements and describe the design and implementation of the ASC Grid Portal.

Keywords: Grid, Globus, Cactus, astrophysics, portal

1. Introduction

The Astrophysics Simulation Collaboratory (ASC) project seeks to provide a Web-based problem solving framework for the astrophysics community to harness computational Grids. Computational Grids are emerging as a distributed computing infrastructure to promote the sharing of information, knowledge and expertise across institutional and disciplinary boundaries as well as to promote the collective use of large-scale computational resources towards the study of complex scientific problems. The ASC is motivated by a distributed community of astrophysicists and computational science researchers working to link together software, computational and data resources to promote collaboration and problem solving in the field of numerical relativity. This community makes up a Virtual Organization (VO) [15], or more specifically what we define as the ASC Virtual Organization.

Web Portals are a common way to support communities of users with similar interests by providing Web-based mechanisms for sharing information and access to distributed resources in a secure and well controlled manner. Thus the proposed solution involves the development of an Astrophysics Simulation Collaboratory Web Portal. This solution is driven by the following high-level requirements:

- *Low-cost access to collaboratory resources from portable, low-footprint ("thin-client") interfaces.* Our community would like a more accessible and unified environment for utilizing the Cactus computational toolkit [2], described in more detail below, to construct astrophysics simulations to be run on Computational Grids. Extending upon the traditional 3-tier Web Portal architecture to deliver application services was deemed a suitable approach for building such an environment, given the pervasive nature of Web technology today. Our goal is to deliver *Web services* that utilize *Grid technologies* to link users with remote resources.

- *Ability to cooperate effectively on the creation and execution of complex simulation codes.* We wish to provide Web-based tools for accessing *shared code repositories* and tools to *compile applications* remotely. We want also to provide access to Grid *resource management services,* including scheduling systems on high-performance computers, for executing simulations developed from the Cactus computational toolkit.
- *Flexible, secure, dynamic sharing of resources across institutional boundaries.* We would like to build services that can take advantage of deployed Grid infrastructure in order to more easily obtain information about remote distributed resources and securely access these resources as needed by our applications. We can do this by building on existing *Grid technologies* including those provided by the Globus Toolkit [14] and the MyProxy certificate repository [23].

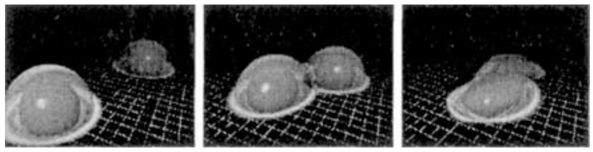

Figure 1. The ASC project investigates astrophysical phenomena involving strong dynamic gravitational fields such as this grazing collision of two neutron stars. The surfaces represent the density of the stellar matter while the warping of the gridded surface represents the gravitational flux on the plane of the collision.

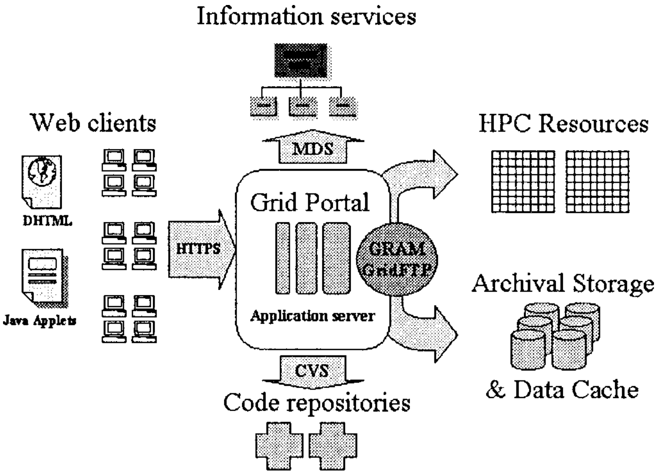

Figure 2. Grid Portals provide a central point of access to widely distributed Grid services.

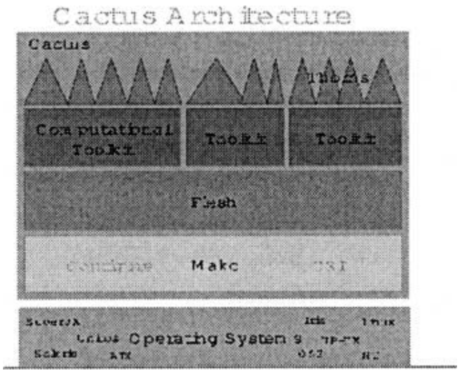

Figure 3. Diagram of the Cactus architecture.

2. Overview

The ASC Portal, in its current form, enables users to work with the Cactus toolkit in a location independent manner on supercomputing resources made available to the ASC Virtual Organization. Users construct simulation codes with Cactus and other components located at multiple sites, compile on any available computer, run on any available computer, and share results with colleagues in a controlled fashion. The ASC Portal is in fact an instance of an emerging class of Web Portals, called *Grid Portals*, that utilize Grid technologies. Grid Portals extend the Web Portal paradigm as a collection of Web services for sharing access to information and online communication to include Web-based access to high-performance computing and data-storage services. In serving the astrophysics community, the ASC Portal is also an example of what is sometimes called a Science Portal, with the NCSA Chemical Engineering Workbench [6,7], SDSC HotPage [20,25], and the Alliance Virtual Machine Room [3] being other examples.

3. Cactus

Before describing the ASC Portal in more detail, we must first take a moment to describe the Cactus computational toolkit and how it is used by the ASC to support the collaborative development of astrophysics simulation codes. The Cactus toolkit has been successfully applied to a wide range of applications, including complex astrophysical phenomenon such as the merger of neutron stars and collisions of black holes (figure 1). It is a modular simulation code framework developed primarily by the Numerical Relativity Group at the Max Planck Institute in Potsdam Germany [2,8]. Cactus consists of a "flesh", which coordinates the activities of modules, referred to as "thorns" which "do the actual work" (figure 3). The thorns interoperate via standard interfaces and communicate directly with the flesh (not with one another). This makes the development of thorns independent of one another, which is essential for coordinating large geographically dispersed research groups, and allows each group to concentrate on its own area of expertise. The separation of *physics thorns* (e.g., black hole or hydro evolutions) and *CS thorns* (e.g., I/O, parallelism, AMR and performance monitoring) permit astrophysicists with different levels of CS expertise to participate in the co-development. Furthermore, since Cactus is distrib-

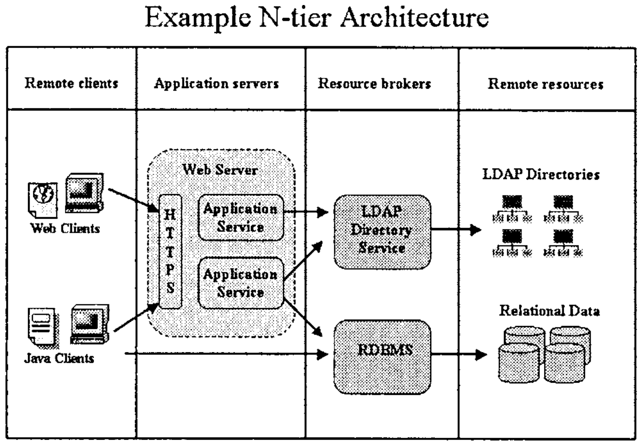

Figure 4. N-tier applications divide functionality into layers in order to maximize the sharing of resources and services among clients.

uted with tools that support access to multiple code repositories for obtaining Cactus thorns, developers and users alike can easily acquire new thorns as they become available. One Cactus thorn that we encourage all Cactus users to employ, for instance, is the HTTP thorn. This thorn enables Cactus simulations to run their own HTTP servers to support collaborative steering through Internet browsers of running simulations as well as to provide access to datasets at runtime.

4. Architecture

The ASC Portal is comprised of several technologies arranged in an N-tier architecture. N-tier applications (figure 4) divide functionality into 3 or more separate layers. Such applications are typically made up of *multiple clients,* an *application server* or a collection of application services made accessible on one host, services that act as *resource brokers* and the *resources* they mediate. This layering helps to maximize the sharing of resources among multiple clients because while the application server is designed to efficiently manage requests for resources among clients according to the *business logic* of the application, resource brokers may be designed to optimize the use of resources according to the types of resources they manage.

A critical layer for any N-tier architecture is the application server. Application servers are "always-up" processes that provide a central point of access to application services for multiple users. Application servers can provide everything from fault-tolerance, say by guaranteeing that critical tasks complete, to online chat services that facilitate better communication among users. With the aid of a persistent storage mechanism, a relational database system (RDMBS) say, application servers can record user activity and notify users when their tasks complete, with an email report for instance. Thus at the core of the ASC Portal architecture is the ASC Application Server. The ASC Application Server delivers pages and Java applets to Internet browser clients.

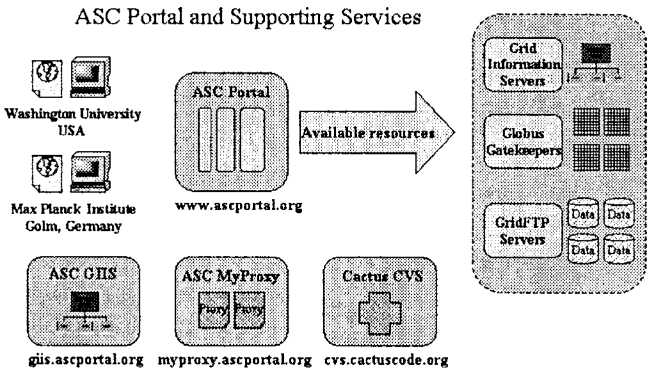

Figure 5. The ASC Portal is supported by a variety of Grid services and technologies.

4.1. Supporting services

Besides the application server, the ASC Portal relies upon several remote services and Grid resource brokers to serve our astrophysics community. These services and related technologies are described in this section.

4.2. Globus and resource sharing

In order to overcome the practical obstacles of accessing and sharing resources across institutional boundaries we employ Grid technologies in the ASC Portal. By using Grid technologies, in our case Globus [14] middleware, we are removed from the difficulties introduced by the tremendous variety of resource types, mechanisms, and access policies we would otherwise have to handle ourselves. We are therefore better able to focus our efforts on delivering a useful application environment.

The Globus Toolkit 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), 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, which provides remote resource allocation and process creation, monitoring, and management services; the Metacomputing Directory Service (MDS), 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.

4.3. Grid information index server

The ASC Portal is supported by a Grid Information Index Sever (GIIS), available as part of the Globus Toolkit. GIISs provide a central point of access to information made available about remote resources by Grid Resource Information Servers (GRIS), where GRISs may register with one or more index servers per the MDS 2.0+ protocol. The latest MDS protocol, MDS 2.1, supports authentication with GSI so that information may be published by Grid resources in a secure manner. Once registered with a GIIS, clients can discover Grid resources within a particular organization by sending a general query to the organizational GIIS. Clients may then gain information about one or more Grid resources with additional queries, or query the GRISs on those resources directly as needed.

The ASC GIIS provides an index to the collection of resources for which we want to enable access to users of the ASC Portal. When adding resources to the ASC Virtual Organization, we contact the site administrators responsible for those resources and request that they configure their GRISs to register with our GIIS. The ASC Portal can then learn about those resources by querying the ASC GIIS.

4.4. GSI certificate repository

Because the Grid technologies we rely upon support GSI authentication, our users must be able to use GSI certificates to access those services. MyProxy is a secure online certificate repository for allowing users to store short-lived GSI proxy certificates, on the order of one week, and later to retrieve shorter-lived copies of those certificates, on the order of hours, with a user-name and password. When users have obtained one or more GSI proxy certificate copies from MyProxy, other applications may use those certificates to authenticate to Grid services without requiring additional user intervention.

The ASC maintains an organizational MyProxy server where members of our virtual organization may store their GSI proxy certificates. Users retrieve shorter-lived copies of those certificates upon "logging on" to the ASC Portal, typically lasting as long as the user is logged on. This enables the ASC Portal to authenticate to remote services that accept GSI proxy certificates on behalf of users as described above.

4.5. Shared document repositories

In order to enable users to acquire and distribute Cactus software onto remote machines from the ASC Portal, we provide users with online tools for accessing CVS (Concurrent Version Server) repositories. CVS repositories manage versioning of documents among multiple authors and is the primary method by which Cactus developers maintain version control of Cactus source code. The Cactus CVS repository (cvs.cactuscode.org) is where the Cactus development team makes the latest versions of Cactus available to the general community.

However, our users often need to work with additional CVS repositories with restricted access. In order to simplify authentication to these restricted repositories, we worked with the Globus Project to develop GridCVS. GridCVS extends the CVS protocol and service to include GSI-authentication. We are currently developing a Java client and corresponding DHTML pages that will allow checkings out and checking in to be performed from/to GSI-enabled CVS repositories from third-party locations, where remote file browsing will be performed with our GridFTP Java and DHML clients, extended from the Java CoG Toolkit (see below).

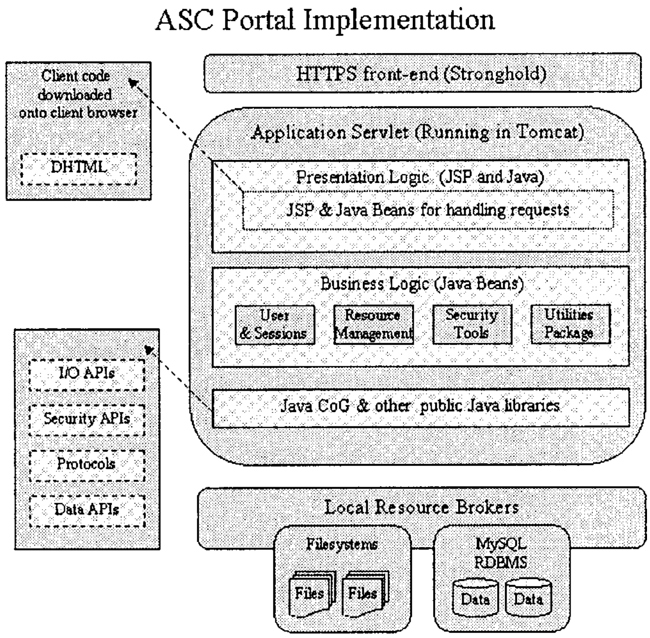

Figure 6. The ASC Portal server-side architecture and its software components.

5. Implementation

This section describes the key technologies that support the ASC Portal and the N-tier architecture we discussed earlier.

5.1. HTTPS/Apache

Communication with our application server is mediated by a front-end Secure HTTP server (HTTPS). Since users may send sensitive information to the ASC Portal at anytime, passwords for retrieving certificates from the ASC MyProxy server for example, all communication between our application server and end-user Internet browsers must be secure. We use Stronghold, a commercial based distribution of Apache, the leading open-source Web server, to secure communication with our application server.

5.2. Java Servlets/JSP

We chose to develop the ASC Application Server in Java, in part due to the popularity of Java in the Web development community and the many open-source toolkits that are available in Java. By developing in Java we are able to employ Java Servlets/JSP technologies to deliver ASC application services as Web applications. Java Servlets and JSP provide a powerful alternative to the more traditional CGI-based approach for delivering Web applications. Java Servlets hide the complexity of HTTP communication, providing a well-defined interface for manipulating HTTP sessions, while JSP provides a powerful yet easy to use mechanism for building dynamic Web-pages.

We deploy our application server with Tomcat, a popular open-source implementation of the Java Servlets/JSP specification. Tomcat acts as a container process, running its own Java Virtual Machine (JVM), supports multiple Java Servlets and serves Web documents that may include JSP instructions. Tomcat can be configured to interoperate with Apache and Stronghold, enabling secure communication between end-users and Java Servlets/JSP Web applications, as described above.

5.3. JDBC

Our application sever is supported by a relational database management system (RDBMS) called MySQL. We use MySQL to store persistent information about users and remote resources as well as to record the tasks users perform. MySQL has gained wide-acceptance in the Web community because it is free for non-commercial use, well documented, and provides very reliable performance. However, in order to support interoperability with other relational database management systems we express all database calls with the Java Database Connectivity (JDBC) package, which provides a generic interface to Standard Query Language (SQL) compliant database management systems.

5.4. Java CoG/Globus

The Java Commodity Grid (CoG) Toolkit [21] enables our application server to interoperate with Grid services. The functionality provided by Java CoG Toolkit that is most pertinent to the ASC Portal is highlighted below:

- *Grid Security Infrastructure (GSI):* The Java CoG Toolkit supports mutual authentication and delegation using IAIK libraries [16];
- *Grid Information Services (GIS):* The Java CoG Toolkit provides access to information services using the Java Naming and Directory Interface (JNDI);
- *Globus Resource and Management (GRAM):* The Java CoG Toolkit provides Grid Resource Access Management (GRAM) interfaces for submitting jobs securely to remote job managers through Globus gatekeepers, including batch scheduling systems on high performance computers [9];
- *Grid FTP:* The Java CoG Toolkit offers a GSI-FTP client for manipulating remote files through GSI-enabled FTP servers [17];
- *Myproxy:* The Java CoG Toolkit includes client classes for accessing GSI proxy certificates on remote Myproxy servers [23].

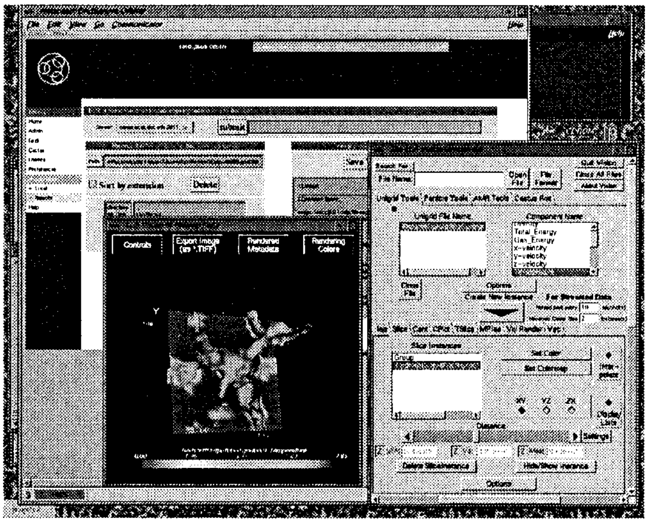

Figure 7. Screen shot of ASC Portal launching LCAVision with a GRAM job to view a remote dataset.

6. Features

In addition to providing DHTML pages that expose the basic functionality provided by the Java CoG Toolkit, DHTML pages for launching GRAM job requests and tracking their status for instance, we have developed many tools users can access from the ASC Portal.

6.1. Secure login

In order to gain access to services provided by the ASC Portal, users logon to the ASC Portal with a secure Web form. Once logged on users may retrieve their GSI proxy certificates from the ASC MyProxy server at any time. From there, users can make use of our application services without the need to supply additional passwords until either the user logs off or their GSI proxy certificates expire.

6.2. User management

We have developed classes for maintaining information about users of the ASC Portal and the tasks they perform. Users can track the remote jobs they have started for instance, or the file transfers they have performed. Users can also maintain a list of GSI proxy certificates they use to gain access to remote resources and easily determine to which services their certificates successfully authenticate.

In addition, users can be granted or denied access to the functions of the ASC Portal, in particular for enabling select users to administer the ASC Application Server, including adding or removing users from the Collaboratory, monitoring user sessions and starting or stopping application server tasks.

6.3. Grid resource management

We attempt to model computational resources and the services they support in an intuitive way by providing classes and DHTML interfaces that abstract the details of the MDS queries required to gather information from remote resources. Additional information about each resource, such as the location of software not otherwise made available to MDS, can be maintained in our application server's primary database and published to users as needed.

6.4. Grid service management

As mentioned above, we can track what services are available on remote computational resources, where for each service we attempt to provide intuitive interfaces for invoking, testing, and monitoring user requests to those services. In most cases, we extend upon the JavaCoG Toolkit to allow user activities to be written to the ASC Application Server database and provide wrappers that unify the interfaces to those services.

6.5. Development tools

We support commonly used tools for managing the development of software within the ASC. In particular, we provide Java classes and DHTML interfaces for invoking CVS and MAKE on remote computers with interactive GRAM jobs, thus taking advantage of GSI security.

6.6. Cactus tools

Since the ASC supports application development with Cactus, we have developed tools that are useful for manipulating Cactus applications and data on remote resources from the ASC Portal. Some of these tools are described below:

- *Parameter file management:* We allow users to maintain 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 ASC.
- *Simulation management:* We enable users to run Cactus simulations and to track the simulations they have run. Cactus jobs can be invoked as GRAM requests or be run from batch scripts that include pre- and post-processing instructions.

6.7. Visualization

We are developing tools that allow users to run applications for visualizing remote datasets. Currently, for those users running X-servers on their local desktop, we provide a Web interface that enables users to search for files with our DHTML-based GridFTP file browser on remote machines and subsequently launch visualization applications that exist on those machines, such as LCAVision, an application developed by the ASC for visualizing HDF5 data, with the application's display set to the user's local desktop. We have found this works well in practice.

7. Future directions

Much of our focus now is on how to provide enhanced support for managing remote datasets and visualizing these datasets from the ASC Portal. We would like to develop online tools to assist users in finding appropriate storage space on mass secondary-storage systems, working with archival data-storage management services, using meta-data browsing catalogues, and to provide support for automated replication of data to high-performance disk caches. Data management facilities are critical for providing access to remote visualization services as well as for sharing the large volumes of archived results produced by this community of researchers.

There are several ways in which we would like to extend support for visualizing remote datasets. To begin with, we have defined Multi-purpose Internet Mail Extensions (MIME) types for datasets produced by Cactus thorns. We can use those MIME types to instruct Internet browsers to launch the appropriate visualization application, if it exists on the client host, when a user attempts to access the data through the ASC Portal. More advanced visualization applications that include support for streaming only those portions of interest from large datasets are being developed within the ASC and by the Cactus development team. However, we are considering whether effective Java applet based visualization software can be developed or extended from existing Java toolkits. Since Java applets can now be easily deployed with Java WebStart technology, we are hoping with this much improved support for Java applets in Internet browsers, we will be able to deliver more advanced visualization services and other application services that are highly interactive.

We are also interested in enhancing the ASC Portal with more collaborative facilities, such as support for online chat services or even support for video conferencing. Collaborative facilities might also include support for sharing access to remote computing resources and datasets across user access privileges. However, this implies the need for Grid-wide access control mechanisms that aren't yet in place.

In the long term we intend to explore alternative modes of interaction with our Grid Portal. One such scenario, that we like to call Grid Pilot, involves a client application that would allow a user to setup tasks offline and to schedule those tasks to run once the user reconnects to the Internet. This is a familiar mode of operation for users of Palm Piots and e-mail clients like Eudora. This mode of operation might be important for users of portable computers.

8. Conclusions

The ASC project serves a widely distributed community of researchers who all hold a common interest in using the Cactus computational framework to understand complicated astrophysical phenomena. We feel that our Grid Portal is an effective model for building any solution that entails distributing and managing data and software among remote resources utilizing standard Web and Grid technologies. These and other Portal designs are critical steps in developing the software infrastructure necessary to make the widespread deployment of Grid Application Servers feasible. We expect Web-based Grid Application Servers will become increasingly pervasive as this software infrastructure matures and becomes the leading paradigm for High Performance Computing application environments. The ASC, along with other Science Portal efforts, are on the cusp of a revolution that is going to change the way we interact with HPC environments for solving the Grand Challenge problems of the future.

References

[1] 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, Int. Journal of Cluster Computing (2001), accepted for publication.

[2] 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 (December 1999) 52–59.

[3] Alliance Virtual Machine Room Home, http://www.ncsa.uiuc.edu/SCD/Alliance/VMR/.

[4] ASCportal homepage, http://www.ascportal.org

[5] W. Benger, H.C. Hege, A. Merzky, T. Radke and E. Seidel, Efficient distributed file I/O for visualization in Grid environments, ZIB Technical Report SC-99-43 (January 2000).

[6] K. Bishop and J. Alameda, A prototypical example of the chemical engineer's workbench, in: *AAAS/SURA/NCSA Meeting on Networking Resources for Collaborative Research in the Southeast* (June 1998); http://www.aaas.org/spp/dspp/rcp/gamtg/agenda.htm

[7] K. Bishop, B. Driggers and J. Alameda, Distributed collaboration for engineering and scientific applications implemented in Habanero, a Java-based environment, Concurrency: Practice and Experience 9(11) (1997) 1269–1277.

[8] Cactus homepage, http://www.cactuscode.org

[9] K. Czajkowski, I. Foster, C. Kesselman, S. Martin, W. Smith and S. Tuecke, *A Resource Management Architecture for Metasystems,* Lecture Notes on Computer Science (1998).

[10] Documentation of ANT build tool, http://jakarta.apache.org

[11] Documentation of Java Server Pages (JSP), http://www.sun.com/products/jsp.

[12] Documentation on Servlet API, http://www.sun.com/products/servlets.

[13] Documentation of TomCat JSP Server, http://jakarta.apache.org

[14] I. Foster and C. Kesselman, Globus: A metacomputing infrastructure toolkit, Int. J. Supercomputer Applications 11(2) (1997) 115–128.

[15] I. Foster, C. Kesselman and S. Tuecke, The anatomy of the Grid: Enabling scalable virtual organizations, Int. J. Supercomputer Applications (2001), to be published.

[16] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, A security architecture for computational Grids, in: *Proc. 5th ACM Conference on Computer and Communications Security Conference* (1998) pp. 83–92.

[17] Grid Forum GridFTP Specification DRAFT, http://www.sdsc.edu/GridForum/RemoteData/Papers/gridftpspecgf5.pdf

[18] Grid Portal Development Kit, http://dast.nlanr.net/Features/GridPortal

[19] Homepage for TIKSL project, http://www.zib.de/Visual/projects/TIKSL/.

[20] HotPage Home, http://hotpage.npaci.edu

[21] G. von Laszewski, I. Foster, J. Gawor, W. Smith and S. Tuecke, CoG Kits: A bridge between commodity distributed computing and high-performance Grids, in: *ACM 2000 Java Grande Conference*, San Francisco, CA (3–4 June 2000).

[22] LCAVision, http://zeus.ncsa.uiuc.edu/miksa/LCAVision/

[23] MyProxy web site http://dast.nlanr.net/Features/MyProxy/

[24] National Research Council, *National Collaboratories: Applying Information Technology for Scientific Research* (National Academy Press, 1993).

[25] M. Thomas, S. Mock, J. Boisseau, NPACI Hot-Page: A framework for scientific computing Portals, Presented at the *3rd International Computing Portals Workshop*, San Francisco (7 December 1999).

Michael Russell obtained his B.Sc. in computer science at the University of Chicago in 1997. He spent 3 years in industry, focusing primarily on user-interface design, financial applications, database programming and Web application development. He became interested in Grid computing in December of 1999, when he learned of Ian Foster's work at the University of Chicago and Argonne National Laboratory. Michael joined Ian Foster's team in May of 2000 to lead the design and development of the Astrophysics Simulation Collaboratory Portal.
E-mail: russell@cs.uchicago.edu

Gabrielle Allen obtained her Ph.D. in astrophysics at Cardiff University in 1993. For several years she worked as a researcher in Numerical Relativity at the Max Planck Institute for Gravitational Physics using supercomputers to study the physics of black holes. In 1998 she joined the Cactus Group there, developing and supporting a modular framework for large scale supercomputing. She is involved in a number of large, collaborative projects focused around both computer science and physics, and is particularly interested in techniques for large scale computing in a Grid environment.
E-mail: allen@aei-potsdam.mpg.de

Greg Daues
E-mail: daues@ncsa.uiuc.edu

Ian Foster is Senior Scientist and Associate Director of the Mathematics and Computer Science Division at Argonne National Laboratory, Professor of computer science at the University of Chicago, and Senior Fellow in the Argonne/U.Chicago Computation Institute. He currently co-leads the Globus project with Dr. Carl Kesselman of USC/ISI as well as a number of other major Grid initiatives, including the DOE-funded Earth System Grid and the NSF-funded GriPhyN and GRIDS Center projects. He co-edited the book "The Grid: Blueprint for a New Computing Infrastructure."
E-mail: foster@mcs.anb.gov

Edward Seidel is a Professor at the Max-Planck-Institut fuer Gravitations physik (AEI) in Potsdam, Germany, with Adjunct positions at Illinois Astronomy and Physics Departments and Senior Research Scientist at NCSA. Seidel has led both the computer science and the numerical relativity research groups at the AEI for the past five years, with over 120 publications spanning both disciplines. Seidel is a Co-Chair of the Global Grid Forum Applications Working Group. In the US, he is a PI for the NSF funded KDI project to establish a Astrophysics Simulation Collaboratory, and a NASA Grand Challenge Project. In Europe, he led the DFN project TIKSL: TeleImmersion of Black Holes, for which a follow-on proposal GriKSL will shortly start, is the Coordinator of the EU funded network for Astrophysical Sources of Gravitational Wave Astronomy. His research interests include general relativity, astrophysics and computational science. He received a Ph.D. degree from Yale in 1988.
E-mail: eseidel@aei-potsdam.mpg.de

Jason Novotny is currently a Computer Systems Engineer at the Lawrence Berkeley National Laboratory, where he has been involved in distributed computing research involving the DOE Science Grid effort. His interests include distributed computing frameworks and middleware, distributed system security, high performance networking, and high level tools and components for developing distributed computing applications. Jason Novotny has been involved with developing and deploying distributed applications and Grid middleware for both the NCSA/Alliance Virtual Machine Room (VMR) and the NASA information Power Grid (IPG).
E-mail: jdnovotny@lbl.gov

John Shalf
E-mail: jshalf@lbl.gov

Gregor von Laszewski is an assistant computer scientist at Argonne National Laboratory and a Fellow of the Computation Institute at University. He received his Ph.D. from Syracuse University in 1996 while focusing on Metacomputing. He is actively involved in the Global Grid Forum since its creation. Current research interests are in the areas of parallel, distributed, and Grid computing. Specifically, he is working on topics in the area of using commodity technologies that combine Web and Grid services.
E-mail: gregor@msc.anl.gov