The GridPort Toolkit: a System for Building Grid Portals[1]

Mary Thomas, Steve Mock, Maytal Dahan, Kurt Mueller, Don Sutton
University of California at San Diego, San Diego Supercomputer Center
{mthomas, mock, mdahan, kurt, suttond}@sdsc.edu

John R. Boisseau
Texas Advanced Computing Center, The University of Texas at Austin
boisseau@tacc.utexas.edu

Abstract

Grid portals are emerging as convenient mechanisms for providing the scientific community with familiar and simplified interfaces to the Grid. Our experience in implementing Grid portals has led to the creation of GridPort: a unique, layered software system for building Grid Portals. This system has several unique features: the software is portable and runs on most webservers; written in Perl/CGI, it is easy to support and modify; it is flexible and adaptable; it supports single login between multiple portals; and portals built with it may run across multiple sites and organizations. The feasibility of this portal system has been successfully demonstrated with the implementation of several application portals. In this paper we describe our experiences in building this system, including philosophy and design choices. We explain the toolkits we are building, and we demonstrate the benefits of this system with examples of several production portals. Finally, we discuss our experiences with Grid web service architectures.

1. Introduction

The Grid [1] is a term that is applied to the infrastructure being constructed to interconnect highly distributed compute, archival, instrumentation, and other resources into a large, parallel, computational resource. Grid portals are emerging as a highly convenient mechanism for providing the scientific community with a familiar and simplified interface to the Grid and Grid services [2]. The number of Grid portal projects is rapidly growing, and each project has its own solution to the challenges of connecting disparate resources, although many of these systems are converging towards some common approaches [3].

The complexities that portal developers encounter in developing a simple user interface to the Grid include dealing with the limitations of the client browser (Netscape, IE, applets, JavaScript) and HTTP protocol, choosing the right webserver software (Apache, Netscape), and identifying programming languages and mechanisms to use on the webserver to connect to Grid services (Perl/CGI, Java Server Pages, PHP). While each solution is of interest, the lack of interoperable standards and common system components currently prevents portals from sharing information, data, and resources and arriving at common, reusable solutions.

Our experiences in implementing computational science Grid portals for both the NPACI and PACI programs [4], such as the NPACI HotPage [5,6] and other application portals [7,8,9,10], have led us to create a unique, layered system for Grid portal development that facilitates the building of multiple portals using common code. This minimizes the work required to build the infrastructure to support and provide services used by each portal.

The significance of this research is that we have demonstrated that the GridPort software can be used in production environments, that it extends to resources beyond NPACI and SDSC (Alliance, PSC, and NASA/IPG), and that it is easily ported to other centers and systems. Additionally, we have demonstrated that the GridPort software can be extended to support the *web-services* architecture [11] that is being developed for

[1] This work was supported in part by Grants from the following projects: NPACI, NSF-ACI-975249; NPACI–NSF-NASA IPG Project; Pharmacokinetic Modeling, NCRR Grant No RR11526; NBCR , NIH/NCRR P41 RR08605-07.

commercial purposes and implemented in commercial technologies such as Jxta [11], SOAP [13], WSDL [14].

The remainder of this paper presents a summary of our experiences gained in building Grid portals and developing software for the Grid, and it is organized as follows: In Section 2, we present a discussion of the project background as well as the philosophy followed, design criteria met, and choices made in creating the GridPort system currently being used for production portals. Section 3 contains a detailed description of GridPort, the technologies used, and the limitations of this system. In Section 4, we demonstrate the usefulness of this software system by examining GridPort-based production portals, followed by plans for future work and our conclusions.

2. Background and Motivation

The NPACI HotPage was first developed in 1997, and was intended to provide NPACI users with a web-based single entry point to resources maintained by the NPACI partnership (comprised of UC San Diego, Caltech, University of Texas, University of Michigan, UC Berkeley, University of Virginia, and many other partners). As an information service, the website initially provided users with centralized access to documentation about and status of the resources. As the project evolved there were requests from other centers to use the software, so we began the process of generalizing and packaging it. The software was made available to others, and a few sites installed local versions of the HotPage.

A long-term goal of the project was to devise a method by which users could directly access their accounts on the HPC resources. Several methods were tested, and in 1999, the Globus project [15] provided the key enabling technologies that were needed in order for users to have real-time, secure access to all HPC resources. Once the HotPage was expanded to take advantage of many interactive operations enabled by Globus, the need for generalized software to support the development of application portals became evident. The result was the Grid Portal Toolkit, or GridPort.

The GridPort software system continues to evolve as a result of work being done to develop the NPACI HotPage user portal and other application portals. The system contains elements that support portal development and interactivity by integrating different Grid and Web services, and it employs a variety of technologies in order to support the needs of the large and various research projects within NPACI and accommodate their diverse requirements. The system is flexible and portable to other centers, sites, and resources and can be used in a modular manner.

The driving philosophy behind the design of the GridPort portal system is the conviction that many potential Grid users and developers will benefit from portals and portal technologies that provide universal, easy access to resources, and require minimal work by Grid users and Grid application developers. There is clearly a need for large, sophisticated and advanced projects such as Gateway [16], Cactus [17], Unicore [18], and GriPhyN [19], but most of these projects require large budgets and staff to support their development and use. We believe that for a significant number of individual users and small projects, GridPort is a more appropriate system to enable Grid participation.

From the initial design and creation of the informational HotPage, to the construction of the GridPort Toolkit and our current application portals, this has remained a focus of the project, and has driven many of the design choices made for the project. For example, most computational scientists are not computer scientists, so we have focused on keeping the portal environment very simple, easy to use, and easily accessible. GridPort-based portals require no software downloads or configuration changes on the client side, and run on common web browsers.

2.1. Design Requirements

At the start of the GridPort project, we realized that design choices would impose certain limitations and require tradeoffs, but we accepted them in order to solve a particular aspect of the Grid portal challenge: that of providing a generalized infrastructure that is accessible to and useable by the computational science community. If every scientist who wished to build a portal had to install the full set of portal software and services needed to connect to the Grid, there would be a tremendous duplication of human effort as well as unnecessary complexity in the resulting network of connected systems. A simpler and more efficient solution was required. The GridPort approach attempts to solve this challenge by meeting several key design goals:

- *Universal access*: Portals will be web-based (accessed through a web browser); Portals must run anywhere, anytime; Require no downloads, plug-ins or applications, and leave no secure data on system when done; Must support 'old' web browsers that do not support recent technologies such as client side XML, for instance.
- *Technology transfer*: Develop a software system that client portal developers can download to build

portal systems with minimal programming expertise, and that is relatively easy to install.

- *Use common Grid technologies and standards*: minimize impact on already burdened resource administrators by eliminating the need for proprietary GridPort daemon on HPC resources.
- *Provide a scalable and flexible infrastructure*: Facilitate adding/removing Grid resources, services, jobs, and users.
- *Security*: support HTTPS/SSL encryption at all layers, and provide access control.
- *Single login*: Required for easy access to and navigation between Grid resources.
- *Adopt Global Grid Forum standards*: actively collaborate and promote Global Grid Forum activities.
- *Client applications and portal services should be able to run on separate webservers*: Enable scientists to build their own application portals and use existing portals for common infrastructure services.

As a result of these design goals, and lessons learned from building several production Grid portals, we have arrived at a Grid portal system that meets most of the goals mentioned above, and presents a scalable, generalized solution for Grid portal development. The project has met all of the goals set above with the exception of the last goal, though an architecture in which clients host Grid application portals on local systems and use the NPACI portal services is in development and has been successfully prototyped. It is discussed in the Application Portals section of this paper.

2.2 Related Work

There are a variety of projects designed to provide transparent access to distributed computing environments such as Gateway [16], WebSubmit [20], and the NCSA Portals Notebook and ChemWorkbench projects [21]. Note that the Computing Portals Organization web site maintains a current list of projects [3]. All of these projects are web-based interfaces to a distributed set of HPC resources, and we discuss 3 projects that are typical of current approaches to portal technologies.

Gateway employs a programming model that is implemented as a virtual metacomputer. Evolving from the WebFlow project [22], it is based on object-oriented technologies. The user interface consists of user-programmable modules that interact with a system of distributed Gateway servers, via secure applets. Gateway plans to use Globus to access HPC resources, but Gateway requires that a Gateway server run on each HPC system. GridPort avoids this complexity by using the web server to process and build Globus commands and then communicate directly with the Globus processes on the HPC systems. In addition, we have decided not to use applets and Java, since they are not supported on all browsers.

The NCSA ChemWorkbench project is a component-based approach that provides access to Grid services and resources through an application running on the desktop and employs components that are objects and that use XML to describe public interfaces and protocols. Based on software developed by the Common Component Architecture Forum, the Workbench project is testing how well these components facilitate distributed collaboration.

WebSubmit is an early portal project, and provides basic functions such as file transfer, directory manipulation, and job execution. The GridPort toolkit and WebSubmit projects are very similar: both are based on the use of CGI scripts, and both provide web-based user interfaces. Differences between the projects lie in the security model and the way in which the back-end services are implemented. Where possible, the GridPort software takes advantage of grid services such as Globus, whereas WebSubmit uses SSH and specifically built commands for each system. We note that Globus was not readily available at the time that WebSubmit was created.

At this time, we are aware of only a few projects specifically devoted to developing portal software at the toolkit level, such as the Grid Portal Development Toolkit (GPDK) [23], a Java Servlet implementation with many similar functions to those of GridPort, and the Globus Commodity Grid Toolkit project [24], a clearinghouse for many toolkits that interface to the Globus Metacomputing Toolkit.

3. The GridPort Toolkit

The GridPort Toolkit has been in use as the backend engine for the NPACI HotPage and other application portals since 1999, and has undergone continuous change as the needs of multiple application portals have evolved. Recently, the GridPort software has been updated and generalized to increase security, support a broader range of services, and make porting it to other sites easier and faster.

The toolkit has been modified so that *multiple* application portals can access the same instance of the GridPort toolkit library. This is important for two reasons. First, each application portal such as the HotPage must

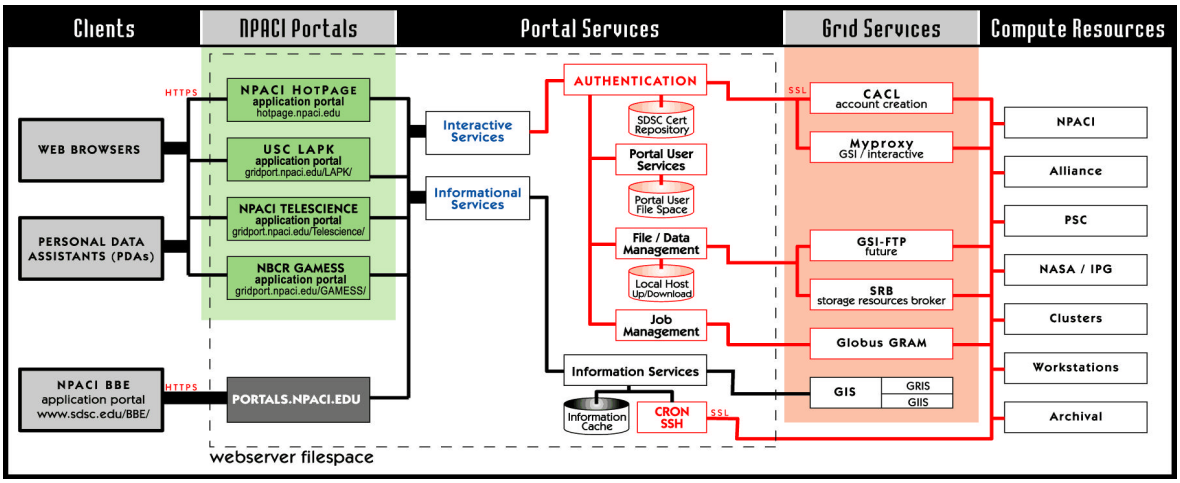

Figure 1. The diagram shows the layers discussed in the previous section, and indicates the types of client devices and portals and software/services are currently in use or that we plan to install in the near future.

be able to manage its own filespace, and also use the GridPort libraries. For multiple instances of portals, it becomes complex and cumbersome to have to install a copy of the GridPort libraries for each portal. This can present maintenance challenges in the event that the local portal administrator has to modify some part of the code, or move the portal home directories. Second, GridPort supports *single login*: this is accomplished by requiring, at present, that all portals be hosted on the same domain (for example, npaci.edu). Portals hosted by servers in the npaci.edu domain can access the same security data (e.g. cookies), which contain encrypted information about the session data. This is discussed in the Security section below.

The current GridPort portal system is shown in Figure 1. In this schematic, we represent different parts of the portal system as layers. Each layer represents a logical part of the portal where data and service requests flow back and forth, and handles some specific aspect or function of the GridPort portal system. These layers are:

Clients: These are typically web browsers. However, we have included other portals as clients, based on the web-services software that we are developing. This is discussed in detail in Section 4.3.1.

NPACI Portals: Currently, NPACI application portals exist on the same machine and are served to clients by separate virtual webservers, and they all use the same instance of the GridPort libraries. This allows them to share data, libraries, filespace, and other services on the webserver machine. The bottom portal is a *web-service* portal, and is intended to be used by other applications and clients (see Section 4).

Portal Services. In addition to mediating between client requests and Grid services, the GridPort software also performs services for the portals and users such as managing session state and portal accounts and file collections, and monitoring the GIS system. These are services that are portal specific, and are not typically addressed by Grid or web developers.

Grid Services and Compute Resources. These layers are the standard middle and backend tiers of the Grid, and represent the layer where Grid services such as Globus, Legion, SRB, NWS, Apples and Metaschedulers are available.

3.1 Technologies Employed

Wherever possible, the GridPort Toolkit employs simple technologies. On the client side, we do not require the use of client-side XML or applets, since some of our users still use older browsers and platforms that do not support these technologies. On the server side, we chose to implement the portal software in Perl/CGI since this software is easy to install under most webservers and is supported on all HPC systems, making it easy for portal developers to install and support. We realized that more complex technologies such as Java and CORBA could be used to provide more advanced services, but using those technologies would not further our goal of enabling technology transfer. However, several current portal research projects such as Gateway [16], the Missisipi Portal [25] and the Grid Portal Development Kit [23] employ these technologies and have demonstrated their usefulness and stability, making them reasonable

candidates for further investigation and possible use in GridPort.

The GridPort Toolkit v2.0 is implemented in Perl 5/CGI, and has been installed and tested under both Netscape and Apache webservers. It should run on any Unix/Linux OS/webserver combination that supports SSL encryption and the HTTPS protocol and has the required Perl/CGI libraries installed. At NPACI, the webserver software used is Netscape Enterprise running on Solaris.

Data and configuration files are used by many scripts at run-time. Typically, these are flat text ASCII files that are easily changed by site administrators. At NPACI, some of the data used in these files is generated from databases. In general, we avoid live database hits in order to reduce latencies and to ensure that the portal is able to deliver services even if the database server is inaccessible.

GridPort makes use of other key Grid technologies. The Globus/GRAM gatekeeper is used to run interactive jobs and tasks on remote resources [15]; Globus Grid Security Infrastructure (GSI) and Myproxy are used for security and authentication [15][26]; the SDSC Storage Resource Broker (SRB) is used for distributed file collection and management [27]; and the Globus Grid Information Systems/Grid Resource Information System (GIS/GRIS) is used for information services [15].

3.2. Services Supported

Portals using the GridPort system provide two categories of services to the application portals running on the system: informational and interactive. Interactive services require user authentication, while informational services are open to any users without logging in to the portal. GridPort functions are used primarily by interactive services at this time.

Interactive services are the secure transactions that provide users with direct access to HPC compute resources and allow the webserver to perform tasks for the user on those resources. To access interactive services, the user needs to login and authenticate via web pages. For this, we use the Grid Security Infrastructure (GSI) provided by the Globus Toolkit. Each interactive portal user needs a unique portal account, the creation of which requires valid accounts on the resources supported by the portal. The portal manages the user's portal account and keeps track of sessions, user preferences, and portal filespace. GridPort provides a single login environment for multiple portals (though there are certain constraints on portal setup, discussed below in the section on security issues). Once a user is logged into an NPACI portal, the user has access to any NPACI-hosted portal that is part of the single login environment.

Currently, GridPort v2.0 supports the following interactive functions:

Accounts: The portal manages the user's account and keeps track of sessions, user preferences, and portal filespace. All portal users must create a portal account, and they must have a valid PKI/GSI certificate. Currently, the NPACI portals accept certificates from several sites (such as NPACI, Alliance, NASA/IPG, Cactus, and Globus). For NPACI users, we have created an on-line certificate creation system.

Authentication: Users can log on to NPACI portals using two mechanisms: either authentication against certificate data stored in the SDSC repository, or by using the Myproxy server. Most of our users do not care to manage and handle certificates, and many of them are mobile, making it even more complex to handle these certificates. For example, if a user is logging into a portal on a public browser at the airport, she will most likely not have access to her certificate. Until the use of secure ID cards (or a similar mechanism) is universally supported, we have chosen to implement a secure repository for our users. This is discussed in detail below.

Jobs: All remote tasks are currently executed via the Globus/GRAM gatekeeper. To execute any job or task through the portal, a series of steps must be completed in order for the transaction to occur: (1) user login status must be confirmed, (2) the job command is parsed, (3) the user's environment is established, (4) a Globus RSL command is assembled, (5) the user proxy is verified or recreated, (6) the command is issued to the Globus daemon running on the remote host, and (7) the results are parsed, formatted, and returned to the web browser on the user's workstation, stored on the portal, or transferred to the user's SRB file collection. GridPort supports compiling and running programs, performing job and batchscript submission and deletion, and viewing of job status and history.

Command execution: Users may execute simple Unix-type commands such as mkdir, ls, rmdir, cd, and pwd. The steps in performing command execution are similar to those used in managing jobs.

Files: By providing file and directory access to compute and archival resources and portal file space, GridPort enables file transfer between the local workstation and the HPC resources. Users can also perform common file management operations on remote files, such as tar/untar, gzip/gunzip, and movement to archival storage.

It is easy to modify the GridPort toolkit to add more functionality, or to reflect local site policies (such as use of additional authentication mechanisms).

3.3. Resources Supported

The current version of the GridPort toolkit employs the Globus Toolkit as the mechanism for connecting to remote computational resources. Any system running Globus can be added to the system by incorporating the data about the system into a flat text configuration file. Currently, the NPACI Portals connect to the following systems: IBM (Blue Horizon, SP); Compaq (TCS1); CRAY (T3E, T90); Sun (E10K); SGI (O2K); Hewlett Packard (V2500); and various workstations and clusters running Globus. These resources are located at several sites, including SDSC, NCSA, Pittsburgh Supercompting Center, NASA/IPG, UT Austin, Univ. of Kentucky, and others. In addition, we can access file archival systems running software compatible with the PKI/GCI certificate system, and we have recently added the ability to access any file system running an SRB service.

3.4. Security Issues

Security between the client web browser and the web server is handled by SSL using a 56- or 128-bit key. For communications between the web server and the grid, the SSL RSA X509 certificate technology that is a feature of the GSI Toolkit authentication system (*gssapi_ssleay*) is employed.

User authentication is accomplished by providing the portal with a valid proxy file. The proxy may be generated from a key/certificate pair or the portal may retrieve the proxy on behalf of the user from a Myproxy server. The password used for proxy retrieval or proxy generation by the portal is never stored and travels the internet via a secure https connection.

Session files and other sensitive data like user proxies are stored in a restricted access portal repository. The repository directory structure is located outside of webserver filespace, and has user and group permissions set such that no user except the webserver daemon may access these files and directories. User login sessions are tracked via a browser cookie which is assigned a random value by the webserver when the user successfully authenticates to the portal. The random value in the cookie corresponds to a session file, which ties the cookie in the user's browser to a specific user on the portal. The session file also contains a timestamp which GridPort uses to expire user login sessions that have been inactive for a set period of time.

Portal users must create a portal account. The creation process requires the user to to supply the portal with a digital certificate from a known Certificate Authority (CA). Once the user has presented the portal with this credential, the user will be allowed to use the portal with the digital identity contained within the certificate presented to the portal. In order to access a computational resource through the portal, the user must already have privileges on that computational resource. When a portal uses GridPort to make a request on behalf of the user, GridPort presents the user's credentials to the computational resource, which decides based on the local security model whether the request will be honored or denied. The portal acts as a proxy, executing requests on behalf of the user on resources that the user is authorized to access based on the credentials they presented when they created their portal account. Portal users have the same level of access to a particular resource through the portal as they would if they logged into the resource directly.

3.5. Using GridPort

In order to install GridPort, it is necessary to have a server running an http daemon that supports CGI scripts and secure connections via the https protocol. The server must have a working installation of Perl, version 5.0 or newer. In addition, GridPort uses the Globus client tools that must be installed on the webserver. The current GridPort snapshot was developed on a Solaris 2.5.1 machine running the Netscape Enterprise web server, Perl 5.004_04, and Globus 1.1.3.

The GridPort software may be downloaded from https://gridport.npaci.edu/downloads. To install it, untar the GridPort tar file into a directory outside of webserver filespace, but that is accessible by the user the webserver daemon. Then step through each directory in the GridPort tree, read the included README files, and modify the configuration files as appropriate.

The portal software that is using GridPort shares the same filespace as the GridPort Toolkit. The application portal developer incorporates the GridPort libraries directly into the code, and makes subroutine calls to GridPort software to access the functionality that GridPort provides. Once GridPort has been installed and configured, building tools based on the GridPort code is simple, yet enables complex behavior. A batch job submission tool from the HotPage, for example, contains only three lines of code that reference GridPort. The rest of the 750 lines of Perl code are specific to the needs of the HotPage portal, and consist mostly of html formatting code. Each of the CGI scripts for all of the application

portals developed with GridPort follow this pattern closely, using between 3 and 6 lines of Perl code to access GridPort functions.

Science portals based on HotPage and GridPort technologies have been implemented at the NASA Information Power Grid, the Department of Defense Naval Oceanographic Office Major Shared Resource Center, the National Center for Microscopy and Imaging Research (NCMIR), the University of Southern California (USC), the Daresbury Labs, UK, and the San Diego Supercomputer Center. The GridPort research effort has also resulted in collaboration among portal development groups at NPACI, NCSA, the Globus project, and NASA IPG to develop a common PACI-wide grid portal, the PACI HotPage. The goal of the collaboration is to develop a common grid infrastructure so that user portals can access resources across all the collaboration sites. Many of the sites that started portal development with HotPage code have gone on to build their own portal systems, which was the goal of the technology transfer aspect of this project.

3.6. Limitations

The GridPort Toolkit has proven to be very useful and modular software. The code has been ported to several systems and is used for multiple production applications. However, due to the design constraints and chosen technologies, there are limitations to the system. First, the portal software is based on Perl/CGI, which is known to have lower performance on webservers than technologies such as Java servlets due to the fact that each time a new request arrives, the webserver must instantiate a new Perl process. There are software systems available that incorporate Perl processes into the webserver process, but we have not used them. However, in the day-to-day operation of the HotPage and other portals, we have not really noticed any unacceptable latencies associated with using Perl.

There are latency issues associated with the portal systems. This becomes a perception problem, since portal users tend to have expectations that events on the portal will happen instantaneously, which is not always the case. The latencies are due to factors such as file upload/dowload/transfer through the webserver itself, network latencies (out of the control of the portal software), and known latencies associated with using the Globus GRAM gatekeeper.

There are other latencies associated with how the webpages are built. As discussed above, due to the typical profile of users within the PACI community, we had to limit client side features to the simplest set, and so the pages are constructed of simple HTML and javascript, and all the portals are frames-based (frames are used to display results and request status). The result is that the portal itself is often much simpler in layout and design than one that would be built as an application running locally using advanced technologies such as Java. We approach this from a 'level of service' view: as a portal user community becomes more sophisticated, we add more advanced technologies. For example, in the GAMESS portals, user now have the option to install a local version of OpenGL visualization software, and download the QMView plugin and interactively view results.

Getting Grid Portal accounts is cumbersome. The current GridPort toolkit supports users with GSI certificates, and we can only make certificates for users with NPACI accounts. This is out of the control of the portal development community: there is a need for meta-computing accounts and allocations, but there currently exists no software for managing Grid accounts, and the allocations process is still focused on assigning compute time associated with single machines, rather than Grid systems.

4. GridPort-Based Portals

Currently, there are several production portals in use at NPACI that are based on the GridPort Toolkit, and other portals are still under development. In addition, the software has been ported to several sites, including NASA/AMES, Stennis Space Center (NAVO/PET MSRC site), and the Daresbury Labs, UK. At this time, most of the portals share the same file space as the GridPort software (see Figure 2, and Section 3). As mentioned above, we have begun to experiment with developing application services, discussed in the last part of this section. All portals running at NPACI share a single login environment.

4.1. The NPACI HotPage

The HotPage user portal is designed to be a single point-of-access to all Grid resources that are represented by the portal. In the case of NPACI and the PACI portals, these resources span the PACI Grid and NASA Information Power Grid. It provides access to both informational and interactive services (see Figure 2). Information services are comprised of web pages and links to existing documentation and dynamically updated status data. Interactive services are secure transactions that provide users with direct access to the HPC compute resources and allow users to access their HPC accounts,

Figure 2. Screenshot of the NPACI HotPage, during an interactive session.

where they can submit, monitor and delete jobs, manipulate directories and files, and manage accounts.

The layout of the website allows a user to view these resources from both the Grid or individual resource perspectives, in order to quickly determine system-wide information such as operational status, computational load, and available compute resources. This design provides simple, fast, and direct access to HPC systems and general information about them, with a focus on getting users working on NPACI systems quickly. Recent modifications to the interactive portion of the website include a new file navigation paradigm, new tools for editing remote files, job submission wizards, and multiple file selection capabilities.

The informational services provide a user-oriented interface to NPACI resources and services. They consist of on-line documentation, static informational pages, and links to events within NPACI, including basic user information such as documentation, training, and news. The informational services provide real-time information for each machine, including operational status and utilization of all resources; summaries of machine status, load, and batch queues; displays of currently executing and queued jobs; and a downloadable graphical map of running applications mapped to nodes. This data is published via the Grid Information Services (GIS), or via the HotPage. The GIS services allow us to share data with or to access from other portal or Grid systems such as the Alliance/NCSA or the Cactus Grids [17]. New versions of the Globus GRIS services will provide authentication via GSI security infrastructure allowing the GIS to also be used to obtain and publish user specific data such as job status, user account information, etc.

Interactive services are the secure transactions that provide users with direct access to the HPC compute resources and allow the webserver to perform tasks for the user on those resources. To access the interactive services, the user needs to login and authenticate. GridPort is the backend engine for the HotPage, so authentication is based on the Grid Security Infrastructure (GSI) provided by the Globus Toolkit. Since GridPort supports a single login environment, once a user is logged into the HotPage, the user is logged in to any NPACI hosted portal. Authentication is based on a user portal account. After authentication, the user is able to access accounts with the same privileges as if he had logged directly onto the system via telnet or ssh. By using the GridPort Toolkit, an application programmer can extend the set of basic functions provided by the HotPage. Note that security, authentication and access to the HPC systems are automatically included when any of the GridPort modules are used.

4.2. Application Portals

While the HotPage is a low-level portal, allowing authenticated users to log on and directly access accounts on HPC systems through a web browser, the GridPort Toolkit also supports developers interested in providing Web access to their scientific applications. The resulting applications approach the true promise of the grid: A scientist interacts with a standard web interface, while an application runs on potentially many high-performance computers at remote locations in response to the scientist's request. GridPort is currently being used to create web interfaces to chemistry, drug dosage, and other applications in collaboration with computational scientists at SDSC. In this section we discuss two of the portals built using GridPort that demonstrate both the usefulness and simplicity of using GridPort for building application portals, and at the same time demonstrate the effectiveness of these portals.

4.2.1. Pharmacokinetic Modeling (LAPK) Portal

The GridPort Toolkit has been used create a portal for a drug dosage modeling package from the Laboratory for Applied Pharmacokinetics (LAPK), led by Roger Jelliffe, a physician and professor of medicine at the University of Southern California [ref]. Jelliffe and the LAPK have been developing software for 30 years that enables physicians to tailor drug dosage regimens to the individual patient. Simplified statistical models run on desktop computers, but newer, more accurate, nonlinear models require supercomputing power. The physicians involved in the research have historically found working on HPC systems to be difficult. The tasks of logging onto an HPC

system, uploading input data, submitting a job and downloading results had to be kept extremely simple in order for the physicians to complete their jobs.

There was motivation to create a portal which physicians could use to run jobs on HPC resources without being exposed to the complexities of those computational resources. A key advance realized by the construction of the portal is that researchers can now run multiple tasks at the same time, whereas before they were limited to single job runs. The portal manages file names, moves data around for the clients, and allows researchers to vary simple parameters such as number of CPUs or job run time.

The GridPort Toolkit was chosen to build the LAPK portal based on the ability of developers to use GridPort to rapidly deploy a prototype portal demonstrating the viability of the portal concept. Construction of the prototype portal took less than two weeks development time. Further development has been focused on adding more sophisticated interfaces and features, including job tracking and history, user access control, and user administrative pages. The portal also allows researchers to view 2D plots of their results, and allows them to manage tests and results interactively. Most of the work on the portal is done by student interns at SDSC, and thus it is a relatively inexpensive project. Like the HotPage, the portal uses only frames and simple client-side JavaScript. The size of the portal software is relatively large, but not complex: there are approximately 50 cgi files, with an average size of about 200 lines each. Not all of these files are used to access the Grid.

4.2.2. NBCR Portal Environment

The National Biomedical Computation Resource, NBCR, an NIH-NCRR funded project at SDSC, is building software and portals to conduct, catalyze, and enable biomedical research by harnessing advanced computational technology. The ultimate goal of the Resource is to facilitate biomedical research by making advanced computational, data and visualization capabilities as easy to access and use as the World Wide Web, freeing researchers to focus on biology. The NBCR transparent supercomputing initiative was started over 5 years ago and augments the work of researchers by providing web-based access to computational tools, without requiring training in supercomputer systems.

The computational chemistry environment, GAMESS/QMView, is a package that provides all of the tools necessary to build, launch, compute, and understand computational chemistry data in one environment [8]. While the original 'web accessible' version of the GAMESS/QMView infrastructure provided a complete infrastructure for carrying out computational quantum chemistry on a variety of platforms, it did not have the features of security and file transport that are enabled with the GridPort toolkit. With those features added to the environment, the portal is now more powerful and flexible for remote computation and file manipulation. The GAMESS web portal simplifies the process of using the GAMESS program on sophisticated supercomputer architectures by enabling file transfer, intelligent job building and submission, and in the near future, the ability to use the visualization component of the package (QMView) through the web browser.

Because the portal is available online and based on the GridPort Toolkit, scientists can use GAMESS from any internet-enabled computer. Work has begun on a version that is accessible from portable wireless devices including personal digital assistants (PDAs) and mobile phones.

4.3. GridPort Web Services Experiments

The term "web services" describes a service-oriented architecture in which distributed applications that comprise some specific function are wrapped in a portable format like XML and published as part of a service or used for a service request. A client goes to an information service and locates an application service that satisfies the client request, and the client obtains the needed XML schemas and protocols, and then uses this information to submit a request to the published service. The ability to define common protocols, systems, and toolkits for Grid Portals and Grid services to publish, reuse and share is clearly important.

As stated in Section 2 above, a key goal of the GridPort project is to create a simple solution that will enable a portal client to build a website hosted on local machines, that will be able to access the portal services provided by larger organizations, such as the system built for the NPACI HotPage portal. It appears that the web services architecture will facilitate achieving this goal. We designed a very simple experiment designed to determine the efficacy of a web services oriented architecture within the GridPort portal system. The initial system was built with a very elementary protocol, using HTML and https. Form elements were used as the client mechanism for sending data to the service portal. A web service running at https://portals.npaci.edu was implemented with a set of wrapper Perl scripts to the GridPort libraries.

A set of simple client web pages were built with HTML, running on a local webserver, browser, or laptop.

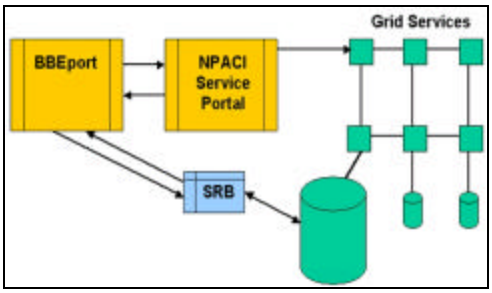

Figure 3. Shows the framework used for the BBE Portal. The portal-to-portal communication layer is used for authentication and for submitting the model simulation jobs to the Globus computational grid.

Tests with a variety of users located in the US and abroad demonstrated that these web pages install and run easily. We have shown that this system works with a set of very basic and limited functions including file listing, batch job submission, binary execution, and authentication, all managed by the web service portal.
The resulting software has been bundled into the GridPort Client Toolkit (GCT v.1), and consists of a set of client tools and services, protocols, and example web pages that interested parties can download [ref]. As a result, any NPACI user with a portal account has access to a very simple toolkit to build locally hosted portals. In the future, we plan to implement a full protocol based on XML schemas, and we are working with other researchers and the GGF/GCE working group to develop XML standards for the portal community that will support interoperability.

4.3.1. Bays, Basins, and Estuaries (BBE) Portal

The Bays, Basins, and Estuaries (BBE) project is an effort to develop a scientific portal for conducting multi-model Earth System Science (ESS) simulations that make use of previously developed resources. These simulations are run to forecast the transport of sediments within the San Diego Bay area during a storm. The technology developed for the BBE project is centered at building a *portal middle layer* that brings the individual resources together such that reusable methods are explored and the objectives of the BBE portal are met.
Figure 3 outlines the current framework system for the project. The BBE portal has been running on the BBE project webserver for some time, and employs Perl/CGI scripts to process user requests and run tasks on a local workstation, and XML encrypted schemas to store data in the SRB file collection system. To perform more detailed and advanced calculations, the project needed to run calculations on more powerful computational resources. The portal employs the GridPort Toolkit for authentication, job submission and file migration. The change to using HPC resources is transparent to the users. In the BBE portal system, users build jobs via HTML forms, and the job request is converted into a batchscript that can run on a set of HPC systems. The batchscript and job request details are sent to the NPACI application service portal via the simple protocol described above. After the job completes on the HPC resource, the batchscript is configured to move the files back to the SRB system.

Note that minimal effort was required in order to modify code to access the NPACI portals services. Developers first tested the portals connection independent of the portal using test driver code. After 10 iterations the software modifications were inserted into existing production code. Only four tests were needed to demonstrate that the new software worked. In all, only four new perl scripts were required, three for login (attempt, success, fail) and one for job submit.

5. Conclusions and Future Work

The NPACI GridPort Toolkit is a simple, robust, and flexible system. It is integral to several science portals running at multiple centers, and we expect its use to increase as Grid users and developers alike discover the benefits of portals and seek efficient tools for portal construction and operation.

We will continue with ongoing upgrades planned for the GridPort portal system, including full integration of the SRB data collection management system, and integration of information services such as the Globus GIS/GRIS model. We will develop useful and standardized tools and protocols for the GridPort Client Toolkit, in collaboration with projects such as the Common Component System project at Indiana and the Global Grid Forum. We are working with Globus to provide the initial modules for the Globus Perl CoG kit, based on GridPort. The first version of this software will be presented at the Euroglobus 2001 workshop this summer [5].

Other upgrade plans for GridPort technologies and services include: personalization and advanced accounting, including functions for handling Grid accounts; design of a service broker; expansion of infrastructure in order to make it possible for other Grid researchers and developers to work with the NPACI portal system; integration of existing meta-schedulers; conversion of all portal-related tasks to use the SRB file

collection capabilities; and improved event handling. In addition, we will be working with wireless content delivery technologies as part of our participation in the Cal(IT)2 project [28].

We believe that the GridPort system will contribute to the development of protocol standards that will promote interoperability among the various Grid portal projects, and we hope that this paper will stimulate discussions in this area.

Interested portal developers can download the software from the GridPort website (https://gridport.npaci.edu).

6. Acknowledgements

The authors would like to acknowledge the San Diego Supercomputer Center and the NSF funded NPACI program for their support (both with resources and staff) and for allowing the portals project access to HPC resources. We would especially like to thank Keith Thompson, whose dedicated and focused support of Globus on NPACI systems is remarkable, and Joshua Polterock whose advice and insights into web technologies and servers have been useful. We would also like to thank the Globus team for providing valuable input and ideas on this project. Most of all, we would like to acknowledge the contributions of our talented group of student interns: Ray Regno, Cathie Mills, Akhil Seth, Chris Garsha, and Kathy Seyama.

7. References

[1] Foster and C. Kesselman, editors, The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufman Publishers, 1998.

[2] G. Fox and W. Furmanski. High performance commodity computing. In I. Foster and C. Kesselman, editors, The Grid: Blueprint for a New Computing Infrastructure, Chapter 10. Morgan Kaufman Publishers, 1998.

[3] The Grid Computing Environments Working Group, Global Grid Forum. Accessed on 3/25/01 at http://www.computingportals.org.

[4] National Partnership for Advanced Computational Infrastructure (NPACI). Project website last accessed on June 1, 2001 at http://www.npaci.edu.

[5] M. Thomas, S. Mock, J. Boisseau, Development of Web Toolkits for Computational Science Portals: The NPACI HotPage. Proceedings of the Ninth IEEE International Symposium on High Performance Distributed Computing, August, 2000. Available at: http://gridport.npaci.edu.

[6] The NPACI HotPage User Portal. Last accessed on 6/6/01 at: https://hotpage.npaci.edu..

[7] LAPK Portal: The Laboratory for Pharmacokinetic Modeling. Last accessed on March 25, 2001 at: https://gridport.npaci.edu/LAPK.

[8] GAMESS Portal: The Laboratory for Pharmacokinetic Modeling. Last accessed on March 25, 2001 at: https://gridport.npaci.edu/GAMESS.

[9] Bays, Basin, and Estuaries Portal. Last accessed on June 6, 2001 at: http://bbe.npaci.edu.

[10] Telescience Portal: The Laboratory for Pharmacokinetic Modeling. Last accessed on March 25, 2001 at: https://gridport.npaci.edu/Telescience.

[11] J. R. Borck. Web Services: Next-Generation e-biz. Issue 20, InfoWorld, May 14, 2001, pg 77.

[12] Sun Microsystems. "Project Jxta: an open innovative collaboration." Project website http://www.jxta.org

[13] SOAP: Simple Object Access Protocol. Available at: http://www.w3.org/TR/SOAP.

[14] Web Services Description Language (WSDL). Available at: http://www.w3.org/TR/wsdl.

[15] Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. International Journal of Supercomputer Applications. 11(2):115-129, 1998.

[16] E. Akarsu, G. Fox, T. Haupt, A. Kalinichenko, K. Kim, P. Sheethalnath, and C. Youn. Using Gateway System to provide a desktop access to high performance computational resources. Proceedings of the Eight IEEE International Symposium on High Performance Distributed Computing, August, 1999.

[17] G. Allen, W. Benger, T. Goodale, H. Hege, G. Lnafermann, A. Merxky, T. Radke, E. Seidel. The Cactus Code: A Problem Solving Environment for the Grid. Proceedings of the Eight IEEE International Symposium on High Performance Distributed Computing, August, 1999.

[18] M. Romberg. The UNICORE system: seamless access to distributed resources. Proceedings of the Eight IEEE International Symposium on High Performance Distributed Computing, August, 1999.

[19] The Grid Physics Network Project Description. Available: http://www..org/info/documents/proj-desc1.0.pdf.

[20] J. E. Koontz, R. P. McCormack, and J. E. Devaney. WebSumit: a paradygm for platform independent computing. Proceedings of the Workshop on Seamless Computing, Reading, England, September, 1997.

[21] R. Bramley, K. Chiu, S. Diwan, D. Gannon, M. Govindaraju, N. Mukhi, B. Temko, M. Yechuri. A component Based Services System for Building Distributed Applications. Proceedings of the Ninth IEEE

International Symposium on High Performance Distributed Computing, August, 2000.

[22] D. Ghatia, V. Burzevski, M. Camuseva, G. Fox, W. Furmanski, and G. Premchandran. WebFlow – a visual programming paradigm for Web/Java based coarse grain distributed computing. Concurrency: Practice and Experience, 9(6):555-578, 1997.

[23] Grid Portal Development Toolkit (GPDK). National Laboratory for Applied Network Research. accessed on 6/6/01 at http://dast.nlanr.net/Projects/GridPortal.

[24] Globus Commodity Grid Toolkits. Available at: http://www.globus.org/cog.

[25] T. Haupt, Computational Web Portals Website. Accessed on 6/1/01 at http://www.erc.msstate.edu/labs/ssl/mcwp/challenge.html.

[26] "MyProxy (v1.0)," National Laboratory for Applied Network Research. Last accessed on 3/35/01 at http://dast.nlanr.net/Projects/MyProxy. Last accessed on June 6, 2001.

[27] Baru, C., R, Moore, A. Rajasekar, M. Wan,"The SDSC Storage Resource Broker,² Proc. CASCON'98 Conference, Nov.30-Dec.3, 1998, Toronto, Canada.

[28] Cal(IT)2 Project Webstite. Last accessed at http://www.calit2.net.