Application Portals: Practice and Experience

Mary Thomas, Maytal Dahan, Kurt Mueller, Steve Mock, Cathie Mills, Ray Regno,
University of California at San Diego, San Diego Supercomputer Center
{mthomas, mdahan, kurt, mock, cmills, rregno}@sdsc.edu

Abstract

The implementation of multiple Grid computing portals has led us to develop a methodology for Grid portal development that facilitates rapid prototyping and building of portals. Based on the NPACI Grid Portal Toolkit (GridPort) and the NPACI HotPage, all portals inherit interactive Grid services, share a single account and login environment, and share the infrastructure required to support and provide services used by each portal. We have demonstrated that the GridPort software can be used in production application portal environments and that the software can be configured to extend multiple sites. In this paper, we describe our experiences gained in building Grid portals and developing software for the Grid. We describe the architecture and design of the portal system, Grid services and systems employed, as well as the unique features of the system. We present descriptions of several application portals and the driving design choices. Finally, we discuss the new and emerging architecture system that is based on the web services architecture.

1. Overview

Computational science portals provide the scientific community with familiar and simplified interfaces to high-performance compute resources. The Grid [1] is a term that is applied to the infrastructure being constructed to interconnect these highly distributed compute, archival, instrumentation, and other resources into a large, parallel, computational resource [2]. Based on our experiences implementing user portals such as the PACI and NPACI HotPages [3], and application portals including the GAMESS, Pharmacokinetic Modeling (LAPK), Protein Database Combinatorial Extension (PDB/CE), and Telescience portals [4,5,6,7], we have found that a non-trivial level of effort is required to set up a Grid portal. Additionally, we find that a significant level of effort is required in order to establish inter-organizational resource sharing, often required when implementing community models.

Our solution for building these portals has been to develop toolkits such as the Grid Portal Toolkit (GridPort) [8,9] and the Perl Commodity Grid (CoG) Toolkit [10,11]. To simplify and generalize the task of building and hosting multiple portals, we have developed a methodology that has evolved into an integrated, layered portal "services" approach. GridPort is designed so that multiple application portals share the same installation of GridPort, and inherit connectivity to the computational Grid that includes interactive services, data, file, and account management, and share a single accounting and login environment.

We have demonstrated that toolkits such as GridPort can be used in production application portal environments, that the software can be configured to extend 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 shown that GridPort software can be extended to support the *web-services* architecture [12] that is being developed for commercial purposes and is being explored by the Grid Computing Environments research area of the Global Grid Forum [13,14].

In this paper, we describe our experiences gained in building Grid portals and developing software for the Grid. First, we discuss the project background along with a description of the architecture and design of the portal system, and what Grid services and systems are currently employed as well as what features are supported by the software. This discussion is followed by a discussion of a variety of application portals and the driving design choices, which demonstrates the usefulness of this software system by examining GridPort-based production portals. Finally, we discuss the new and emerging architecture system that is based on the web services architecture.

1.1. Background

The NPACI HotPage, created in 1997, was developed to provide NPACI users with web-based access 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). The website initially provided users with centralized access to documentation about and status of the resources. In response to requests from other centers for similar capabilities, the process of generalizing and packaging the software was begun, and the system was ported to other centers.

With the introduction of the Globus Toolkit in 1998, interactive features were added to the HotPage around 1999 [14,15]. The Globus Toolkit provides a uniform interface to remote resources and encapsulates the complexities and differences behind a programmers API. Globus software provides an authorization system that supports authentication and delegation, allowing applications such as the HotPage webserver to perform tasks as the user on the remote resources. This was an important advance in the ability of web-based portals to provide secure, transparent access to these resources, without having to store and pass around usernames and passwords. By adding real-time, interactive connectivity to the portal, HotPage users were able to access their accounts on the HPC resources directly and securely in real-time, and the HotPage was expanded to take advantage of the many interactive operations enabled by the Globus Gram gatekeeper.

Around this time the word 'portal' began to emerge as an accepted term to use when referring to web-based systems, hence the HotPage was transformed to Computational Grid Portal. As requirements for other application portals emerged, the HotPage code base was separated into two distinct parts: front-end interface components such as HTML pages and CGI scripts (the HotPage Portal), and middleware software. The Grid Portal Toolkit (GridPort) is a collection of back-end scripts that the portals use to perform operations on the Grid employing Globus and other grid technologies [9]. Separation of the Globus related functionality into a Globus Perl commodity grid toolkit (CoG) is underway, and will be used by GridPort (see related paper [10]).

More recently, we have begun to experiment with a simple web services architecture and protocol based on HTTP/CGI and FORMS elements. With this system, clients can host their application web pages on local machines and use the services provided by the portal system at NPACI to access NPACI resources. This is described in more detail in section 5.

1.2. Relevance of this Work

Our experiences have shown that creation of new portals using similar GridPort functions as HotPage but serve different users and purposes can be easily built and easy to use. GridPort-based portals require no software downloads or configuration changes on the client side, and run on common web browsers, allowing users to run specific applications on the web through a simple and familiar web interface. In Section 3, we describe these portals in more detail.

The GridPort software is also easy to install, since it employs common Web and Grid software. To date, 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, the San Diego Supercomputer Center, and a growing number of other sites.

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.

1.3. Grid Resources Supported

For the portals discussed in this paper, all are based on the GridPort toolkit, which employs the Globus Toolkit as the primary mechanism for connecting to remote computational resources. In particular, the Grid Resource Application Manager (GRAM) and the Grid Information Services (GIS) are used for job execution and information services. Thus, any resource running Globus can be added to the portal system by incorporating machine relevant data into the resource configuration files. This is discussed in more detail below. Currently, the NPACI Portals connect to the following HPC compute systems:

IBM (Blue Horizon, SP); Compaq (TCS1); CRAY (T3E, T90); Sun (E10K); SGI (0rigin); Hewlett Packard (V2500); and various workstations and clusters running Globus. These resources are located at several sites, including SDSC, NCSA, Pittsburgh Supercomputing Center, NASA/IPG, UT Austin, Univ. of Kentucky, and others. File archival systems are accessible that run software compatible with the PKI/GCI certificate system (SGI/DMF, HPSS), and the SDSC Storage Resource Broker (SRB) service [21].

1.4. Organizations, Sites and User Communities Supported

The application portal system described in this paper provides access to portal developers and users with valid accounts on resources supported by a large number of organizations: the NSF, including the PACI, NPACI, Alliance, and DTF programs [17]; the Cal(IT)2 project [18]; a variety of SDSC/UCSD research projects [19]; and resources provided by the NASA/IPG project [20]. The NSF PACI partnerships span resources that are located at universities that include UCSD, CalTech, UT Austin, University of Michigan, Boston, Kentucky, Indiana University. The current requirement for creation of a portal account is a valid user account at any of these centers and an acceptable certificate from a trusted certificate authority (CA). Since the community is continually changing at the resource, partner, site, and organizational levels, the software system must be highly flexible.

2. Application Portal Architecture

As stated above, by building multiple application portals, we have evolved a 'portal system' that supports the generalized requirements of these portals, and is shown in Figure 1. In this diagram, we represent the different parts of the system as layers. Each layer represents a logical part of the portal where data and service requests flow back and forth, and which handles some specific aspect or function of the GridPort portal system. The primary Grid software used by the NPACI application portals is the Perl based GridPort toolkit. These layers are:

Clients: These are typically web browsers and PDA's or other mobile devices. We have also included other portals as clients, based on the web-services software that is under development.

NPACI Portals: NPACI application portals are hosted on multiple web servers, and they all use the same instance of GridPort. This allows them to share data, libraries, file space, and other services. The bottom instance is where the *web-service*s are provided, and is intended to be used by other applications and clients via an HTML/HTTP/CGI interface.

Portal Services. In addition to mediating between client requests and Grid services, the GridPort software performs services for the portals and users such as managing session state, portal accounts, file collections, and monitoring the GIS system. In addition, the toolkit supports job and command execution via specialized modules and can be easily extended to meet the needs of a specific portal.

Grid Services and Compute Resources. These layers are the standard middle and backend tiers of the Grid, and contain Grid services such as Globus, SRB [21], the Network Weather Service (NWS) [22], Apples [23] or meta-schedulers. In this system, access is provided via GridPort.

Based on the layers defined above, and the separation of the portal software into a toolkit and the scripts needed to support the portal, multiple application portals running on the same system use the same installation of GridPort, share account, authentication and filespace, and hence, inherit a single login environment. We briefly describe GridPort, and other Grid software systems that are used by the toolkit below. For more detailed discussions about GridPort, we refer the reader to more detailed publications [8,9,10].

2.1. Grid Services Provided

The portal system described above supports two categories of services that can be accessed by a particular application portal: informational and interactive. The data for the informational services includes machine status, load and usage (queues, node maps) and is based on the data required by the NPACI HotPage. Most of this data is gathered via background scripts or from the Grid Information Service (GIS) daemons running on the HPC systems [16]. 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. GridPort based portals inherit the following functionality:

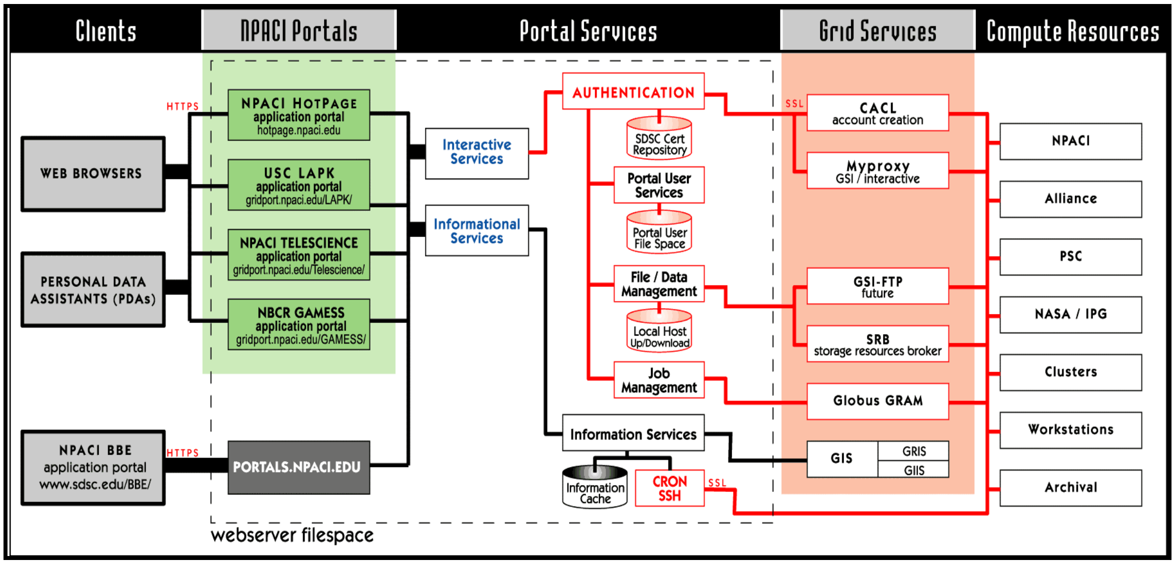

Figure 1. The diagram shows the layered approach used to construct the portal services, and indicates the types of client devices and portals and software/services currently in use (or that we plan to install in the near future).

Accounts: The portal system manages the basic portal user account, and keeps track of sessions, user preferences, and portal filespace. For the basic account, all portal users must create a portal account, and 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, there is the HotPage 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 in another paper [9].

Jobs and Commands: All remote tasks are performed using functions that are part of the GridPort Toolkit, which supports compiling and running programs, performing job and batchscript submission and deletion, and viewing of job status and history.

Files: The GridPort Toolkit supports file and directory access to compute and archival resources, portal file space, and enables file transfer between the local workstation and the HPC resources, and common file management operations on remote files.

2.2. Grid Software and Services Currently Used

The application portals developed with this system connect to the Grid via GridPort, which is built using the following Grid components: the Storage Resource Broker (for file collections), GSI-FTP for file transfers, Grid Security Infrastructure (GSI) for authentication, and MyProxy for proxy storage [24]. GridPort plays such a key role in portal design and development, we briefly describe it below, but refer the reader to more detailed documentation available at the project website [25].

2.2.1. The NPACI Grid Portal Toolkit (GridPort)

The application portal software uses the GridPort toolkit in the form of a shared library. Each application portal, such as the HotPage, must first manage its own file space, and then have access to GridPort (see Figure 2 above). This model evolved from the need to support multiple instances of portals, where we found that it became extremely complex and cumbersome to install and maintain copies of GridPort for each portal. This presents maintenance challenges in the event that the local portal administrator has to modify some part of the code, or move the portal home directories.

A significant part of the US portal community is currently using a similar set of Grid services that includes Globus, the PKI based Grid Security Infrastructure (GSI), the Grid Information Services (GIS), and the Storage Resource Broker (SRB). This is evidenced by the fact that these technologies have been adopted for use on the "Interoperable Web Services Testbed" being built by members of the Grid Computing Environments (GCE) research area of the Global Grid Forum (GGF), the Grid standards setting organization [13]. For communications between the web server and the grid, the secure socket layer (SSL) RSA X509 certificate technology that is a feature of the GSI Toolkit authentication system is employed.

GridPort Security: 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 HotPage on-line certificate creation system is based on CACL, software developed at SDSC [26].

User login sessions are tracked via a browser cookie that is assigned a long 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. Session files and other sensitive data like user proxies are stored in a restricted access portal repository, with encrypted names. 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.

As a result of these security mechanisms, all application portals that share the same filespace, and share the same domain, have inherited a single login environment. This is these portal systems can share the cookies. This is an important step on the road to deploying single login environments across the Grid. For the purposes of completeness, we also briefly describe some of the more relevant Grid software systems and services that are employed by the GridPort toolkit.

Globus GRAM Gatekeeper: The GridPort Toolkit uses the Globus Resource Allocation Manager (GRAM) gatekeeper to remotely execute and manage applications on computational resources. The GRAM gatekeeper is a daemon developed by Globus that processes and manages job requests made in a particular format called Resource Specification Language (RSL) [16]. Using GRAM allows GridPort to enable portals with the ability to submit and cancel jobs securely. These job submissions may be simple file manipulation commands like "ls" and "rm", or they may be complex job submissions for large applications that run through that computer's local batch queueing system. Once jobs are submitted, GridPort uses GRAM to enable the management of these jobs. GRAM provides GridPort with job status information, and the ability to cancel jobs, and GridPort in turn provides these services to portals.

Grid Information Services (GIS): Information makes up a valuable component of the computational Grid, and information services is the mechanism used to gather information on the Grid and Grid resources. Globus provides the Metacomputing Directory Service (MDS) as the system for acquiring information about computational resources, and is used by the portal system described in this paper [16]. The MDS provides an extensible framework for managing and providing both static and dynamic information about computational resources and the grid itself. The Grid Resource Information Service (GRIS) is the service that runs on the computational resource which collects and provides this information. Information is gathered from the GRIS via requests sent using the LDAP protocol. Data is returned from these queries in LDIF format. Portals can use these information services to gather information about resources on the grid for display to the users like in the HotPage portal, or the information could be used to influence decisions made about jobs and tasks that the portal will execute.

Data Management: Data for portal users is managed using 3 mechanisms: the SDSC Storage Resource Broker (SRB) for collection management [21]; GSI-FTP for secure file transfers [16]; and simple file upload, download, and transfers via GridPort software [25].

The SRB storage management system

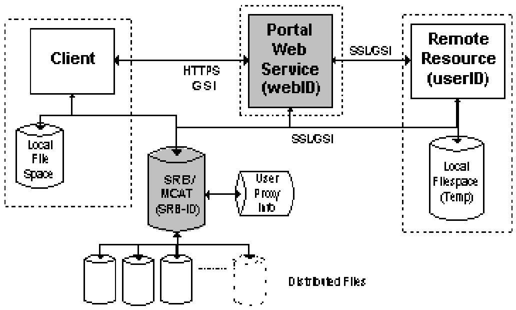

Figure 2. SRB mediates between portals and storage resources.

provides a consistent access mechanism to diverse storage systems. Each portal/SRB user has a collection in which their files are stored, and the physical location of the files is determined by intelligent rules within the SRB mechanism. SRB is able to manage small files such as batch scripts equally as well as 20GB tomography data files used by the Telescience portal. Many different storage systems may be accessed by SRB, including disk-based file systems, tape archive libraries, and relational databases, on hardware ranging from desktop PCs to supercomputers (as shown in Figure 2). This flexibility allows efficient and ubiquitous file management.

Without software systems such as the SRB, portal file management requires complex file transfers between the portal user's workstation, the portal itself, and the HPC system being accessed through the portal. SRB makes it possible for a portal to directly access a user's files in place, eliminating the staging steps that were sometimes previously necessary and allowing files to be stored in the most appropriate place based on various file and system parameters.

Using SRB provides advantages to portal developers, because the same file management code may be used across portals with minor modifications. SRB recently gained the ability to use GSI authentication (the Globus security protocol), allowing secure file transfer and single login via the same mechanisms used in GridPort. At this time we are in the early stages of integrating SRB into GridPort portals, and we plan to offer it as a feature of production portals in the near future.

For file transfer mechanisms between sites, we employ a few GSI authenticated technologies. Globus GRAM has the ability to do simple file transfers. For more complex transfers, and transfers of larger amounts of data, we employ GSI-FTP. GSI-FTP is a version of the normal ftp, which can use GSI authentication. It also has advanced features like third party transfer, which is not found in the general installation of ftp.

File upload to the servers is handled using browser forms and specifying the encoding type as 'multipart/form-data.' This is the standard mechanism for uploading small files through the browser. File download from the browser is handled by specifying the correct mime type for the file, and letting the browser and the user decide what to do with the file.

2.3. Application Portal Security

Application portal security is inherited from the GridPort system: at login, a valid username and passphrase will result in the storage of the cookie, as described above. This cookie is used by GridPort to track the user session state. Security between the client web browser and the web server is handled by SSL using a 56- or 128-bit key. Note that this is one of the primary methods used by commercial technologies such as Java Server Pages or Servlets [27].

Portal users must create a portal account. The creation process requires the user 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, this user must already have privileges on that computational resource. Note that 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.

Some applications require addition restrictions to control the user community (see the LAPK portal). In this case, the implementation details are left to the developers, and GridPort only processes requests that are passed by the portal software. However, it is important to realize that even thought the application portal system may validate a user via their own mechanisms, this use must still authenticate through the GridPort system.

2.4. Implementing Application Portals

There are a wide variety of ways to design and build computational science portals, as evidenced by the papers that are included in this

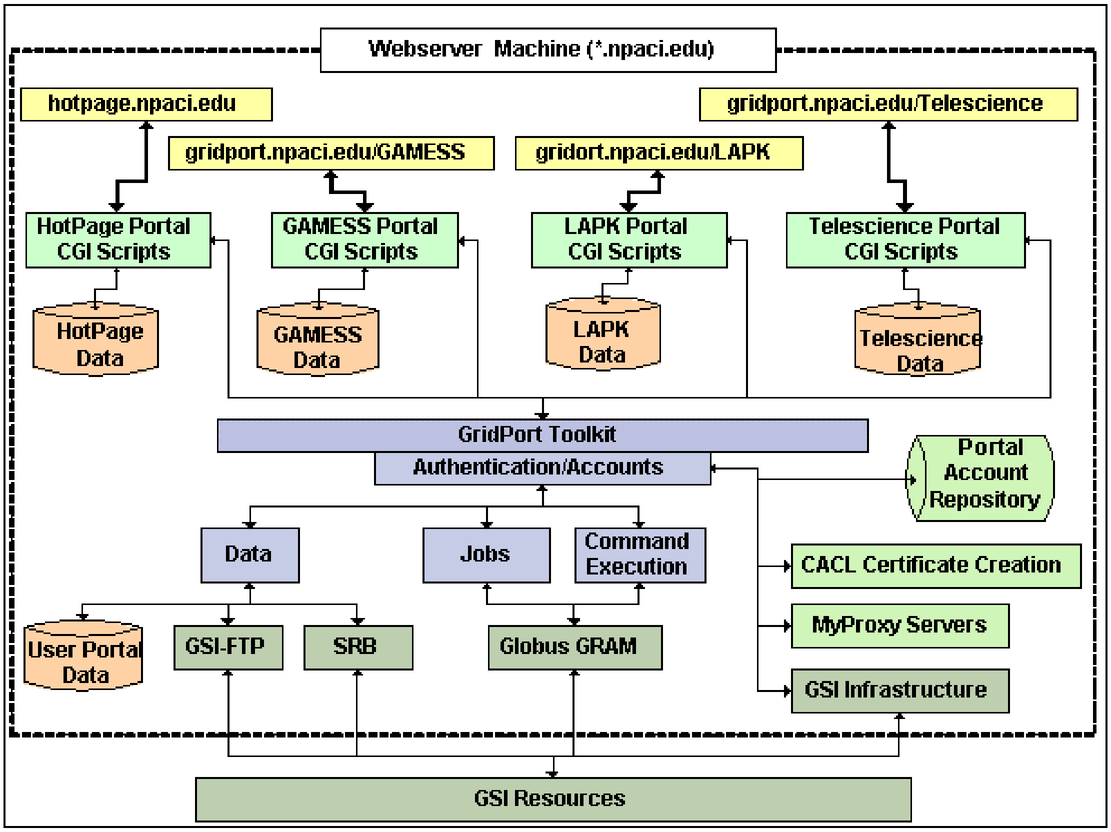

Figure 3. Portal implementation architecture diagram showing multiple portals installed on the same webserver machine. In this design, each portal has its own filespace, and shares the same instance of the GridPort modules. All access to the Grid is done through functions provided by GridPort. Each request is authenticated against common account and authorization information.

special issue. Figure 3 depicts the approach used for implementing portals. The webserver and application portals all have access to the same logical filespace. In addition, each portal has its own file space, where specialized scripts and data are located. In the case of the work done at NPACI, all scripts are implemented in Perl/GCI. The portal scripts are responsible for processing the incoming CGI data and constructing the correct data formats for the specific GridPort functions. We intentionally decoupled the GridPort software from handling the CGI data, so that GridPort could support other types of applications. Each request is checked by authentication mechanisms built into GridPort. Some portals overlay specialized access restrictions, which do not affect GridPort functionality.

The general approach used for a specific application portal is to provide a frames-based template, which allows for rapid project prototyping. The project then evolves along with requirements and user needs. We have found that developing portals with this system is part of a useful startup cycle to help teach and train developers on how to use the grid. The application portal developer incorporates the GridPort libraries *directly* into their code, makes subroutine calls to GridPort software to access the functionality that GridPort provides, or modifies GridPort to suit their needs. Once GridPort has been installed and configured, building tools based on the GridPort code is simple, yet enables complex behavior, with minimal impact or changes to existing or legacy applicaitons. A batch job submission tool from the HotPage, for example, contains only three lines of code that reference GridPort. The rest of the code is specific to the needs of the HotPage portal, and consists mostly of parsing CGI requests and formatting results.

A survey of the portals discussed below showed that in general, the portal scripts based on our templates use between 3 and 6 lines of

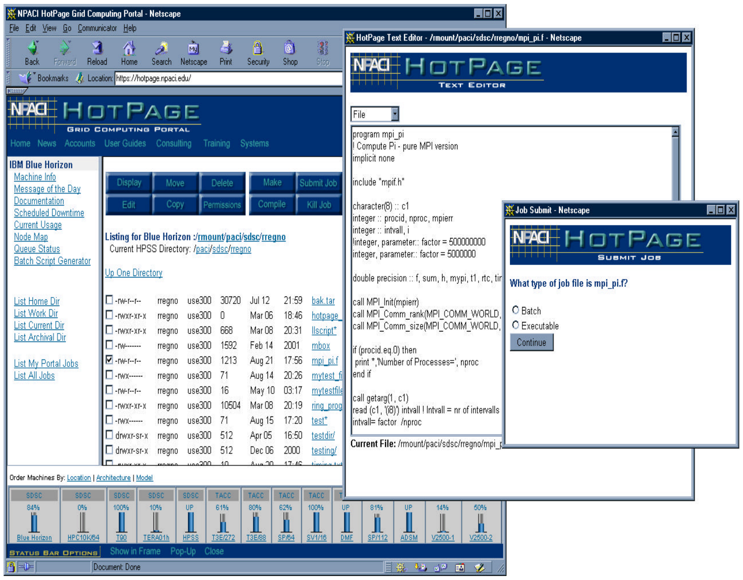

Figure 4. NPACI HotPage display showing interactive session and demonstrating that within this framework, multiple interactive tools such as directory listing, file editing or job submission can be used simultaneously.

Perl code to access GridPort functions. Note that we have found that most of these portals required some development assistance from NPACI staff, mainly due to learning curves and the complexity of working with the Grid. The application portal development team may choose to support their own portal services [25].

3. An Overview of NPACI Application Portals

In this section we describe in some detail several of the portals that are in production at NPACI, and demonstrate the variety and scope of applications and functionality that can be supported and integrated into GridPort based portals. For each of these portals, we describe the motivation and requirements for these projects, design requirements and goals, and the features and services provided. In general, we find that our current software system is most suitable for portals that support community models, rather than being a code developer's environment, although future plans include basic code compilation and optimization tools.

In this paper, there is not time to delve into detail for each portal that we have built, so we cite specific examples of portals developed by the SDSC portals team: the NPACI HotPage [3] and the National Biomedical Computation Resource (NBCR) GAMESS portals [5]. In addition, we discuss portals that were primarily built by project application developers, including the Telescience [4], NBCR Protein Data Bank [7], and the web services based Basin Bays to Estuaries portals [28].

3.1. NPACI HotPage

The HotPage portal is a common interface to all compute and archival Grid resources at NPACI, and it provides access to both informational and interactive services [3]. Informational services are comprised of web pages and links to existing documentation and dynamically updated status data. Interactive services enable secure transactions that provide users with direct access to HPC resources and allow users to access their HPC accounts, where they can submit, monitor and delete jobs, manipulate directories and files, and manage accounts.

The informational services provide a user-friendly interface to NPACI resources and services. They consist of on-line documentation, static informational pages, and links to news and events within NPACI. The informational services provide real-time information for each machine, including 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.

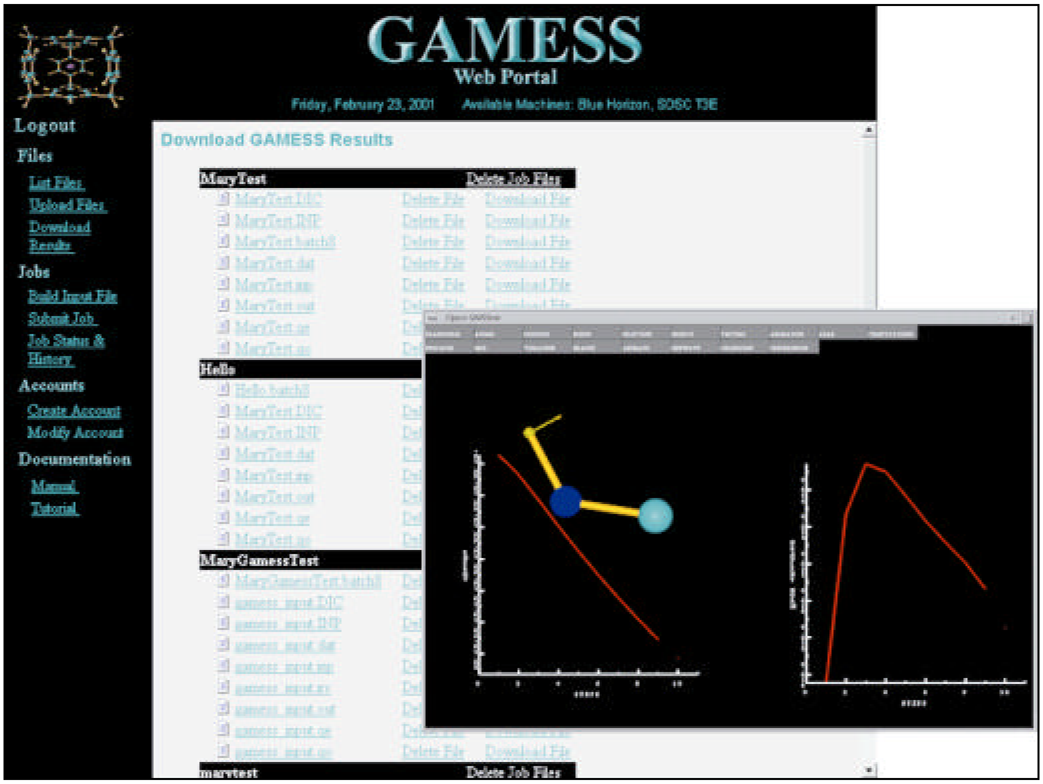

Figure 5. GAMESS Portal display showing a listing of completed results, stored in the users SRB collection. In addition, this view shows the 3D visualization software that is available as a plug-in to the portal.

Interactive services provide direct access to HPC resources and allow the web server to perform tasks for the user on those resources. To access the interactive services, the user needs to login and authenticate. Authentication is based on the Grid Security Infrastructure (GSI) provided by the Globus Toolkit through GridPort. After authentication, the user is able to access accounts with the same privileges as if he or she had logged directly onto the system via telnet or ssh. Recent modifications to the interactive portion of the web site include a new file navigation paradigm, new tools for editing remote files, job submission wizards, and multiple file selection capabilities.

3.2. NBCR Computational 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 [29]. The ultimate purpose 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 five 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 [30]. While the original version of the GAMESS/QMView infrastructure provided a complete system for carrying out computational quantum chemistry on a variety of platforms, it was not accessible through a secure and robust web interface. The GAMESS portal was created to simplify the process of using the GAMESS program on sophisticated supercomputer architectures by enabling file transfer, input file creation, intelligent job

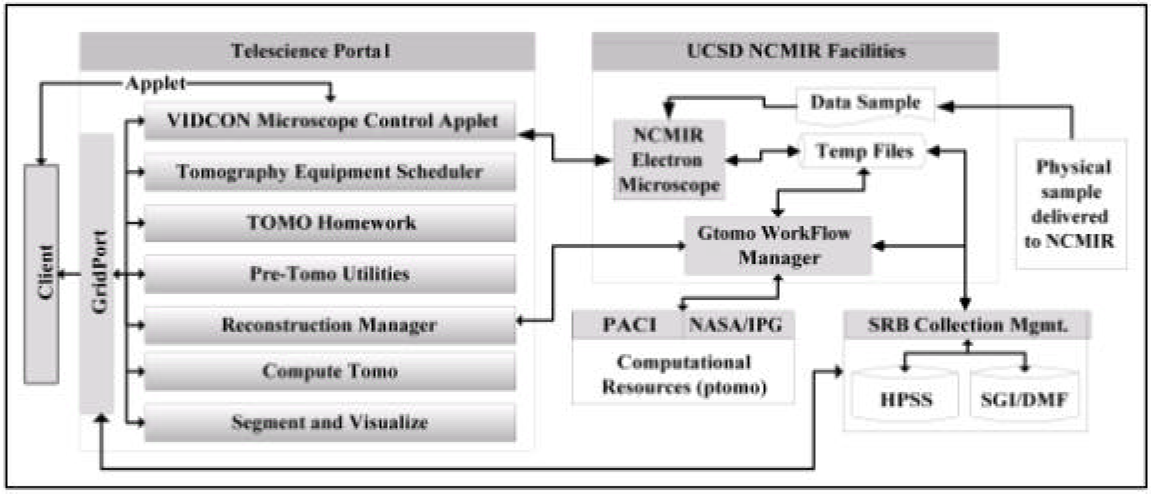

Figure 6. Basic architecture diagram of the Telescience portal, in which portal users will have access to remote instrumentation (microscope), automatic SRB collection access for storage of raw results and analyzed data; portal tools for creating and managing projects; and remote visualization applications.

building and submission, job history and tracking, and in the near future, the ability to use the visualization component of the package (QMView) through the web browser [5].

Because the portal is available online scientists can use GAMESS from any web browser. We have begun work on a version that is accessible from portable wireless devices including personal digital assistants (PDAs) and mobile phones. We are also working to integrate the SDSC Storage Resource Broker (discussed in more detail below), providing advanced file handling and storage management.

Another NBCR funded project is the PDB Combinatorial Extension (CE) Grid Computing Web Portal [7] that enables users to perform structural comparisons of proteins using the Combinatorial extension algorithm on high-performance computing resources at NPACI. Just like GAMESS the CE portal can be accessed from any web browser. The CE Portal gives users online capabilities to upload input files to query the CE database, submit searches to the CE database, and display a status of the database searches and view results of the searches. The portal enables users to upload their input files that contain their protein. They can select a chain in their protein and query a CE database against it. The query looks for similar protein structures and aligns the proteins so the user can view similarities and differences. It is unique from the other portals because it allows 2 forms of login: anonymous and registered. Anonymous login gives users general functionality such as starting a CE search and viewing results of the search. By registering for a portal account the portal user has additional benefits such as a personal and password-protected account in which to store queries and results, the ability to track the progress of searches after submission, and offers more computing power by reducing restrictions on the usage of the HPC platform. The portal also supports molecular visualization programs such as the Compare 3D, which renders a simplified 3D view of a structural alignment with a java applet, QMView and Rasmol. Future plans include the addition of enhanced file handling capabilities using the Storage Resource Broker and automatic computational resource selection.

3.3. Telescience Portal

The Telescience portal is a web interface to tools and resources that are used in conducting biological studies involving electron tomography at NCMIR, the National Center for Microscopy and Imaging Research at UCSD [4]. The basic architecture of the Telescience portal is shown in Figure 1: portal users have access to remote instrumentation (microscope) via an Applet, automatic SRB collection access for storage of raw results and analyzed data; portal tools for creating and managing projects; and remote visualization applications. The portal guides the user through the tomography process, from an applet based data acquisition program to analysis.

A central component of the portal is a tool known as FastTomo, which enables microscopists to obtain near real-time feedback (in the range of five to fifteen minutes) from the tomographic reconstruction process, online and without calibration, to aid in the selection of specific areas for tomographic reconstruction. By using SRB, the portal provides an infrastructure that enables secure and transparent migration of data between data sources, processing sites, and long-term storage. Telescience users can perform computations and use SRB resources at NCMIR and SDSC, and on NASA's Information Power Grid (IPG). The tomography process generates large quantities of data (several hundred megabytes per run, on average), and the portal moves data between these locations over high-speed network connections between compute and archival resources such as the Sun Ultra E10k and the HPSS storage system).

4. Web Services Architecture 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 based on SOAP and XML [31] and published as part of a service or used for a service request using a protocol such as WSDL [32]. 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.

A key, long-term goal of the GridPort project is to enable portal clients to access portal services from a website hosted on local machines or from other portal services systems. The emerging web services architecture facilitates achieving this goal, and we designed a simple experiment designed to determine the efficacy of the 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. 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 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 are available [25]. We present an implementation example below.

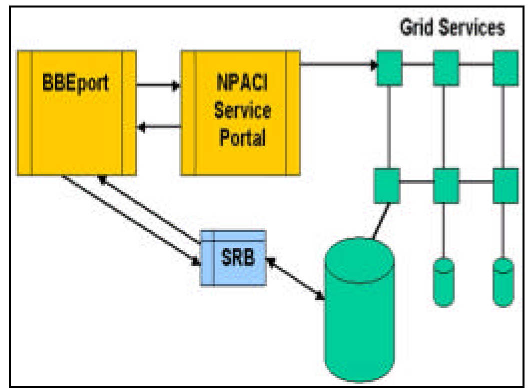

Figure 7. Shows the framework used for the BBE Portal. The portal-to-portal communication layer is used for authentication and for submitting the model simulation jobs to the Globus computational grid.

4.1. The Bays, Basins, and Estuaries (BBE) Portal

The Bays, Basins, and Estuaries (BBE) project is 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. Figure 7 outlines the web services based system for the project. The BBE portal runs on the BBE project webserver, 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 [28]. Recently, the portal functions that provided authentication, job submission and file migration were redesigned to use the GridPort toolkit. This change to using HPC resources has been 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

The NPACI portal system has proven to be a robust and flexible grid application portal development environment. Based on an integrated architecture approach and the GridPort Toolkit, multiple application portals have been built using this system, and are running in production. We find that these portals are fast and relatively easy to build (often taking only a few weeks to deploy), and that this system can support a wide variety of applications. In addition, we have found that the software integrates into existing portals fairly easily, and that it can be merged with other existing programs such as Applets. We anticipate that the cost and time factors associate with deploying these portals will make portal development easier and more appealing to scientists to migrate their applications to the Grid.

5.1. Limitations

For developing applications, the GridPort Toolkit has proven to be very useful and modular software, especially since it is written in Perl. The portal software is based on Perl/CGI, which is known to have lower performance on webservers than technologies such as Java and Servlets. This is mainly due to the fact that each time a new request arrives, the webserver must instantiate and load a new Perl process (or multiple processes), which takes time. To date, in the day-to-day operation of the HotPage and other portals, we have not noticed any unacceptable latencies but as the usage increases this is a potential issue. There are software systems available that incorporate Perl processes into the webserver process (such as Apache modPerl), and we are evaluating using them. More importantly, a primary limitation of Perl seems to be in the area of security. Currently, the Perl module that is used for encryption (CRYPT) does not support PKI proxy delegation, which is a minimal requirement for grid software systems. Unless this module is updated (last update was around 1999) Perl will not be useful for emerging technologies such as web services.

Due to the design constraints and chosen technologies, there are other limitations to the system. Latencies associated with the portal system become 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 the time it takes to execute a command on a remote host, file upload, download, or transfer rates, network latencies (out of the control of the portal software), and known latencies associated with using the Globus GRAM gatekeeper. Based on discussions with the Globus developers, this is due to the time required to generate security proxies and they are working on a solution

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, users now have the option to install a local version of OpenGL visualization software, and download the QMView plug-in and interactively view results.

Other areas of limitation for portals include a lack of grid solutions for accounts and scheduling. Getting portal accounts within the NPACI environment is cumbersome: the current GridPort toolkit supports users with GSI certificates, but those certificates can only be made for users with NPACI accounts. These users need to get individual allocations and accounts on each machine. Thus, there is a need for meta-computing accounts, and the allocations process must be changed form focussing on assigning compute time associated with single machines to Grid systems. There are also no metascheduling or advanced reservation systems that are working at the production levels. This is a key piece of grid software that portals could make effective use of when the capability is developed.

5.2. Future Work

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 was presented at the Euroglobus 2001 workshop this summer [10].

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 [33]. Other plans include the integration of web services into the GridPort architecture, participation in the GCE Interoperable Web Services Testbed, and exploring the use of other languages such as Java since there appear to be serious limitations with Perl.

In summary, 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 believe that systems like GridPort 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

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. In addition, 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.

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 NPACI and PACI HotPage Grid Computing Portals. Last Accessed on 8/1/01 at: https://hotpage.npaci.edu, https://hotpage.paci.org

[4] Telescience Portal: The Laboratory for Pharmacokinetic Modeling. Last Accessed on 8/1/01 at: https://gridport.npaci.edu/Telescience

[5] The GAMESS Portal: The Laboratory for Pharmacokinetic Modeling. Last Accessed on 8/1/01 at: https://gridport.npa;ci.edu/GAMESS

[6] Botnen, X. Wang, R. Jelliffe, M. Thomas, and N. Hoem. Population Pharmacokinetic/ Dynamic (PK/PD) Modeling via the World Wide Web. National Library of Medicine, May, 2001. LAPK Portal last accessed on 8/1/01 at: https://gridport.npaci.edu/LAPK

[7] The Protein Data Bank Combinatorial Extension Portal. Last Accessed on 8/1/01 at https://gridport.npaci.edu/CE

[8] 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.

[9] M. Thomas, S. Mock, M. Dahan, K. Mueller, D. Sutton, J. Boisseau. The GridPort Toolkit: a System for Building Grid Portals. Proceedings of

the Tenth IEEE International Symposium on High Performance Distributed Computing, August, 2001.

[10] S. Mock, M. Dahan, G. von Lesweisky, M. Thomas The Perl Commodity Grid Toolkit. Grid Computing environments: Special Issue of Concurrency and Computation: Practice and Experience, to be published Fall, 2001.

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

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

[13] The Grid Computing Environments research area, Global Grid Forum. Accessed on 8/1/01 at http://www.computingportals.org.

[14] The Global Grid Forum. Accessed on 8/1/01 at http://www.gridforum.org.

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

[16] The Globus Project. Accessed on 8/1/01 at: http://www.globus.org.

[17] The National Science Foundation Partnership for Advanced Computational Infrastructure (PACI). Accessed on 8/1/01 at http://www.nsf.gov.

[18] San Diego Supercomputer Center (SDSC). Accessed on 8/1/01 at http://www.sdsc.edu.

[19] Cal(IT)2 Project. Website last Accessed at on 8/1/01 at: http://www.calit2.net.

[20] NASA Information Power Grid Project. Webstite last Accessed at on 8/1/01 at: http://www.ipg.nasa.gov.

[21] Baru, C., R, Moore, A. Rajasekar, M. Wan,"The SDSC Storage Resource Broker, Proc. CASCON'98 Conference, Nov.30-Dec.3, 1998, Toronto, Canada. Accessed on 8/1/01 at: http://www.sdsc.edu/SRB.

[22] Wolski R., N. Spring, and J. Hayes, "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing," *Future Generation Computer Systems*, 15(5-6): 757-768, (1999).

[23] Berman, F., and R. Wolski, "The Applies Project: A Status Report, Proceedings of the Eighth NEC Research Symposium," German, (1997).

[24] "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 8/1/01.

[25] The GridPort Project. Accessed on 8/1/01 at: http://gridport.npaci.edu.

[26] CACL Software Project. http://www.sdsc.edu

[27] Sun Microsystems, Java Website. Accessed on 8/1/10 at: http://java.sun.com.

[28] D. Sutton. The Bays, Basin, and Estuaries Portal. Accessed on 8/1/01 at: http://bbe.npaci.edu.

[29] National Biomedical Computation Resource. http://www.sdsc.edu

[30] K. Baldridge, J. H. Jensen, N. Matsunaga, M. W. Schmidt, M. S. Gordon, T. L. Windus, J. A. Boatz, T. Cundari. Applications of Parallel GAMESS. In Parallel Computing in Computational Chemistry, Tim Mattson, Ed., 1995, Chpt. 3.

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

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