ARMY RESEARCH LABORATORY

Gateway Portal

by Marlon Pierce and Stephen Wilkerson

ARL-TR-3154 **March 2004**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-TR-3154 **March 2004**

Gateway Portal

Marlon Pierce and Stephen Wilkerson
Computational and Information Sciences Directorate, ARL

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
March 2004	Final	June 2001–July 2002

4. TITLE AND SUBTITLE
Gateway Portal

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Marlon Pierce and Stephen Wilkerson

5d. PROJECT NUMBER
JONO3UH7CC

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRD-ARL-CI-HC
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-3154

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

One problem facing researchers needing high-speed computations for their research is easy access to High Performance Computing (HPC) resources. At one time, a majority of the users in the research community had two machines: one Unix workstation for research and another personal computer (PC) type machine for word processing, e-mail, and creating publications and presentations. As the speed and popularity of PCs have grown, their roll has also changed to include pre- and post-processing of complex jobs as well as their other functions. Moreover, these machines have started replacing the more expensive Unix-based workstations for graphics and other multimedia functions. While PC machines can be made to emulate a Unix X-Windows environment, this requires users to understand both operating systems and have X-software loaded on the PC client. In some cases, unfortunately, users are becoming more PC literate and less Unix knowledgeable. Not knowing the correct commands has led to difficulties in transferring files, writing job scripts, and the like, for HPC access.

One solution is to make the access to HPC assets available through a web interface. A web server can establish a connection between the user's PC and HPC assets, allowing file transfers and job submission. This interface eliminates the need for the user to understand how to write GRD scripts and the problems associated with setting up HPC computer runs, while allowing the scientist to concentrate on the application of the computation for his or her research. However, there are security concerns associated with Web access to HPC. This report addresses the implementation of a Web solution using Java, Java Beans, and Java Server Pages (JSP) on an Apache web server running Tomcat. Examples from the Gateway interface are used as evidence of this technology as a solution to a growing problem.

15. SUBJECT TERMS
HPC, Gateway, Tomcat, Java, JSP

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Stephen A. Wilkerson
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL	22	19b. TELEPHONE NUMBER (*Include area code*) 410-278-3966

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

Access to the U.S. Army Research Laboratory Major Shared Resource Center's (ARL-MSRC's) computer facility from a desktop machine can be accomplished using a number of methods. For example, from a Unix[*] or Linux[†] workstation with the Kerberos[‡] client installed, a High Performance Computing (HPC) ticket can be requested from the command prompt. Once your credentials have been established, an Xterm[§] client can be made from your local display to the HPC host. On the other hand, the procedure from a Microsoft Windows PC platform is a little different. The user must load a stand-alone Kerberos client on the machine. Once done, the user can request tickets from an HPC asset using a small client window. After a ticket has been granted, the local host provides a Kerberos File Transfer Protocol (FTP) and Telnet graphical user interface (GUI). If an Xterm client is installed on the Microsoft Windows machine, then the machine operates much the same way as the Unix/Linux platform does through Xterm. Without an Xterm client, the user is limited to DOS-style FTP and Telnet[**] to interact with the HPC resources. Both systems have one drawback, the need to know a certain amount of Unix-based commands or antiquated DOS commands. Additionally, as systems come and go and procedures for submitting program-specific jobs change, users are forced to learn and relearn command procedures to use HPC for their computation needs. A solution to this constant flux is to write a platform-independent GUI to allow users to interface with HPC through an intuitive "point-and-click" environment. This eliminates the Scientist and Engineers' (S & Es') time-consuming task of learning new Global Resource Direction (GRD)[††] procedures or changes on the system and puts the onus on HPC personnel to keep the interface to HPC current for users. It also puts the burden of providing support and keeping the GUI up to date where it belongs, on HPC personnel.

An obvious GUI environment would be a Web page. Web pages by definition provide platform independence. The Web offers a platform-independent GUI by definition: the user interface runs in a standard browser (Internet Explorer or Netscape Navigator) using standard Internet technologies with no additional client software required. Furthermore, using a portable

[*] Unix is an operating system developed by Bell Laboratories in the early 1960s and is still regarded as an industry standard around the world.

[†] Linux is a Unix-type operating system originally developed by Linus Torvalds with assistance from others. The source code is freely available to all.

[‡] Kerberos is a network authentication protocol. It is designed to provide strong authentication for client server applications by using secret key cryptography. The protocol can be downloaded at the Massachusetts Institute of Technology or at <http://web.mit.edu>.

[§] The "xterm" program is a terminal emulation for the X-Window System and provides compatible terminal emulation similar to a DEC VT 100.

[**] Telnet is a terminal emulation program for TCP/IP networks such as the Internet. TCP is responsible for verifying the correct delivery of data from client to server. IP is responsible for moving packets of data from node to node.

[††] GRD is a program to help users submit jobs to the High Performance Computing Computers at ARL and elsewhere.

development language like Java would enable the GUI to be moved from server platform to platform as new systems are brought on line and older systems retired without redeveloping the HPC GUI. Because this is one obvious solution, other organizations have been developing or are developing approaches along similar lines. For example, the University of Minnesota has developed a Web interface to help users access their mainframe computer systems. The University of Minnesota's Web interface was named Teraweb. Teraweb offers a Web interface to the University's mainframe computers but is not written in Java and does not currently support GRD or a Kerberos interface in an open-source programming environment. Implementing Teraweb was possible, but with modification and additional costs and fees. Teraweb is not open source and is proprietary. Rather than be locked into someone else's program environment, it was desirable to create a vanilla open-source security package that could be used throughout the MSRC computer infrastructure for site-specific applications. Therefore, it was decided that Teraweb was not the solution for ARL-MSRC.

2. Approach

There are two basic approaches to the security feature of the Gateway server that can be taken. A brief description of the two methods is given here and the advantages and disadvantages are discussed as well (figure 1).

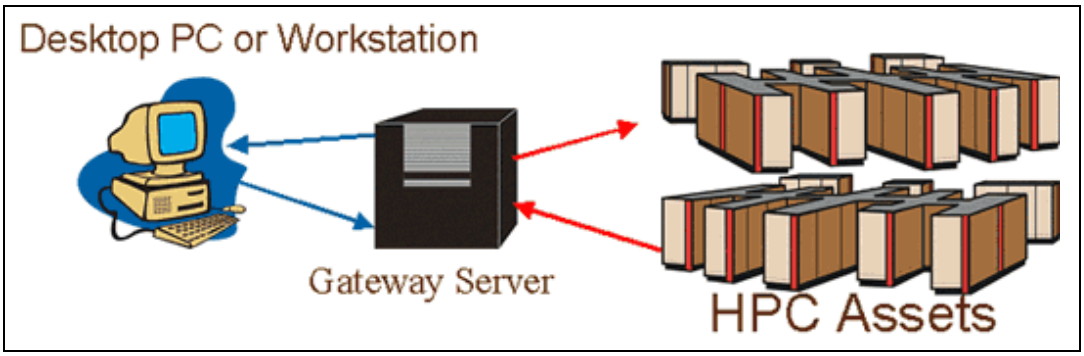

Figure 1. Gateway server approach schematic.

The first approach is purely a Web page connection. In this approach, the user connects to a server using a Secure Socket Layer (SLL) connection. Then, using the SLL connection, the user authenticates, using Kerberos encryption, and gets Kerberos tickets to use in connections with HPC. The Web page provides a number of point-and-click GUIs layered on top of this basic authentication feature. Any number of applications or additional GUIs can be layered on top of this basic secure connection to provide HPC users easy access to HPC assets. The advantage of this approach is in its simplicity. Anyone can write GUI modules to be used with the basic security package. The other advantage is that it's a Web page. Hence, it will work on a Unix

platform or with a PC environment.* However, the approach has its drawbacks. The hyper text markup language (HTML) interface provides only a relatively few simple widgets (buttons, text areas, text fields, checkboxes, etc.) to build GUI interfaces. Advanced functionality (such as interactive remote visualization) requires more sophisticated clients. Other problems include aggravating missing functionality in current browsers. For example, the hyper text transfer protocol (HTTP) allows for multiple file uploads and downloads, but neither Microsoft Internet Explorer nor Netscape Navigator support this. Multiple file transfers are a desirable feature for HPC users who often need to perform recursive directory transfers. This restriction can be overcome by using signed Java Applets, but will not work on machines where users don't have local privileges.

The second approach uses the Kerberos tickets on the local machine and creates a connection between a PC machine and a server, which then connects to HPC. However, this methodology requires that software be loaded on the local user's machine.† The advantages to both of these approaches is the platform independence. Platform independence is based on the use of Java as the primary language and the Apache Web server. Java offers a huge suite of functions and ease of portability. The drawback to this approach is that once again this requires that users are allowed to load software on their machines or the user will be required to have an administrator set up and maintain his computer with current copies of the software, which is not a bad option.

Rather than picking and proceeding with one approach over another, both approaches described here were adapted and used as required to develop specific GUI applications for users. The first approach was developed initially because it required no additional effort on the part of the user. GUIs were created to allow users a basic point-and-click file transfer system from the PC to HPC. Additionally, modules have been written to allow users to submit jobs in ANSYS,‡ ZNS-Flow,§ and Fluent** to HPC for simulation. This user interface writes a basic GRD script for the job submittal and allows users the option to modify the script or accept it as is. The script generation service that we provide is actually general in nature and can be extended to support other queuing systems such as LSF†† and PBS.‡‡ This is not important for the ARL MSRC but is an important issue for deployment at additional MSRCs and for supporting users that have

---

* A personal computer environment might include Microsoft Windows 95, 98, ME, NT, 2000, or XP, as well as Linux and Apple Computers Macintosh operating systems.

† The current approach is to use Java routines to build functionality. Java offers a platform-independent, free programming language with a lot of functionality already built in. Java JDK1.4 is currently being used for the development of users' HPC GUIs.

‡ The ANSYS suite of programs provides state-of-the-art prediction capabilities for structural, thermal, mechanical, computational fluid dynamics (CFD), and electromagnetic analysis of structures and systems. ANSYS software provides accurate modeling of physical phenomenon using accepted mathematical techniques.

§ ZNS-Flow is based on Chimera technology and allows the computational predictions of fluid and air flow problems. This is considered a CFD code.

** Fluent is a commercial software package for solving fluid flow problems in computational fluid dynamics on CFD.

†† LSF load-sharing facility is a suite of workload management products from Platform Computing Corporation.

‡‡ PBS Portable Batch System is a flexible batch queuing and workload management system by Veridian Systems for the National Aeronautics and Space Administration. It works on networked UNIX platforms.

accounts both at ARL and other MSRCs. Additionally, the process is stored in a session that can later be modified and resubmitted for further analysis. The GUI also provides a basic procedure to check on running jobs, jobs in the queue, and to delete these jobs if necessary.

The next step in this process is for the developers of the interface to work with HPC users and build in functionality to the specific application to help ease the use of HPC assets for U.S. Army-related research. Because the program is open source and was written using Java and Java Server Pages (JSP) this process can continue with any developer at any of the MSRC sites and be adapted to the specific needs of a specific organization. One final note is that the servers running the software developed for this GUI require administrative maintenance. This benefits the user in putting the onus of HPC access on the administrator rather than the user. The administrator needs to maintain the system, the GUI, and update procedures as new machines and software continually roll out, while older systems and software are retired.

3. Functionality

The target market of Gateway* is the S & Es at any of the HPC facilities. Table 1 lists the location of these facilities.

Table 1. HPC MSRCs.

ARL	ASC
U.S. Army Research Laboratory ARL MSRC ATTN: AMSRC-CI-HA Aberdeen Proving Ground, MD 21005-5006	Aeronautical Systems Center ASC/HP Building 676, Area B 2435 Fifth Street WPAFB, OH 45433-7802
ERDC	**NAVO**
U.S. Army Engineer Research and Development Center CEWES-IH 3909 Hall Ferry Road TL112, Room 207 Vicksburg, MS 39180	Naval Oceanographic Office/MSRC Program 1002 Balch Boulevard Stennis Space Center, MS 39522-5001

The functionality of the web portal includes the following:

- Secure login and access to HPC assets. Users will use their user name, password, and secure IDs over a Kerberos connection to a server using one of the two methods described in section 2.
- A master page will be maintained that connects to additional services and functions at various sites within the MSRC.

* "The Gateway Computational Web Portal" is a journal article describing the system, and "Gateway Security Infrastructure" provides additional material on security, including alternative strategies for secure Web access. It also provides a more in-depth discussion of WebFlow.

- A server running Apache* and Tomcat† will be used with JSP and Java programs to serve up HTML pages to clients logging on. Connections will be made through a proxy server on the World Wide Web to the Gateway server located behind an MSRC firewall.
- Batch script generation with a GUI interface will enable users access to HPC computers and run simulations on a variety of software packages. Currently, Gateway supports ANSYS, ZNS Flow, and Fluent.
- File transfer services are provided on several levels as described in the approach using HTML protocols and Java-signed Applets.
- Job monitoring will be allowed through the Web portal. Additionally, a job can be deleted while in the queue and stopped through this interface.
- The portal's Web interface can easily be extended to link to existing ARL Internet resources, such as code documentation and the ARL machine load status pages.

Future efforts will enable the following:

- Additional GUI interfaces for applications requiring HPC resources. Examples include CTH,‡ DYNA,§ Paradyn,** and other resource-intensive software applications.
- A Kerberized VNC†† connection to an HPC graphics workstation.
- Access between MSRC sites.
- File transfers between HPC mainframe computer workstations and user PCs. (Connect one location transfer anywhere.)

4. The Gateway Portal

Gateway architecture will initially consist of a three-tiered approach to security. This will eventually be replaced by a four-tiered system. Both will be discussed here. The first layer

* Apache is an open-source software product that serves as a web server platform on numerous operating systems.

† Tomcat is the servlet container that is used for the implementation of Java Servlet (*1*, *2*), and JSP. Simply put, Tomcat is the server-side software that interpolates JSP (*3*, *4*), and generates HTML code for the World Wide Web.

‡ CTH 3-D Eulerian Shock Code was developed by Sandia National Laboratories and can model multi-phase, elastic-viscoplastic, porous, and explosive materials.

§ DYNA is a general purpose explicit nonlinear finite element code developed by Halquist et al. for the Department of Energy.

** Paradyn is an implementation of the DYNA 3-D code optimized for parallel processing.

†† Virtual network computing provides a remote display system that allows one to view a computing desktop environment anywhere on the Internet and for a wide variety of machines. VNC was developed by AT&T.

of the three-tiered system consists of dynamically generated Web pages. These are implemented using JSP that allow Java code to be embedded into an HTML page (figure 2). Therefore, the code that is executed on the client's machine is HTML based.

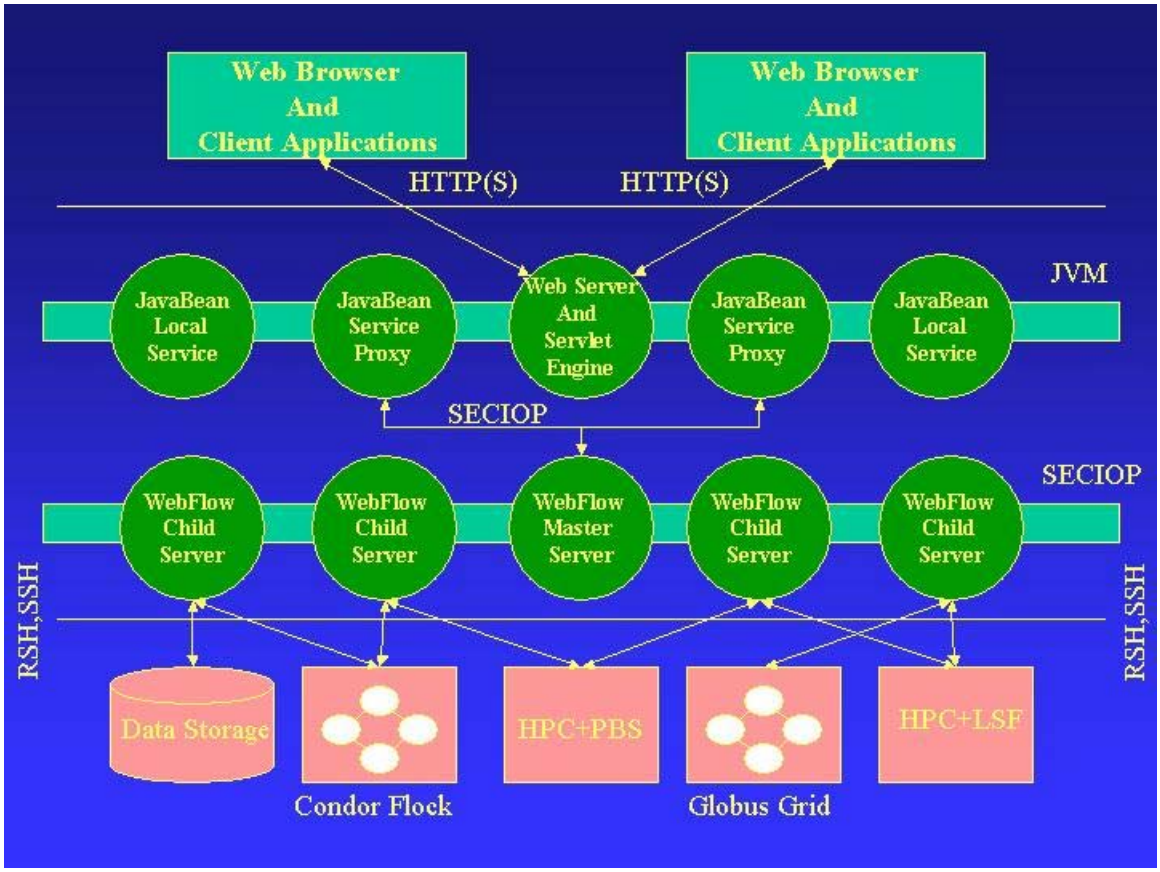

Figure 2. Gateway architecture.

In figure 2, the user interacts with the Apache server through an SLL-encrypted connection. Services are either implemented within JavaBean components in the Web server's servlet engine or by WebFlow modules. Communication between child servers and their parent uses the secure SECIOP protocol. Communications with backend resources use Kerberized rsh and rcp.

The middle tier represents the control layer and consists of two subtiers. The first subtier consists of an Apache Web Server and a servlet engine (Tomcat, also from Apache). The Tomcat server compiles and executes Java code contained in the JSP pages. Most of the functionality (such as security and session management) is encapsulated into JavaBean components rather than in the JSP pages themselves. These components are simply added to the pages. All of the JavaBean components run in the same servlet engine (a Java Virtual Machine) on the same host machine as the Apache web server. These components can provide services themselves, or they can act as proxies to WebFlow components (described next).

The second part of the middle tier consists of the distributed application layer, implemented using WebFlow, originally developed by Dr. Geoffrey Fox's group at Syracuse University and partially funded by the High Performance Computing Modernization Program (HPCMP) through programming environment and training (PET). WebFlow uses CORBA* to allow components to be distributed over multiple hosts. That is, components can be contained in other servers running on different machines besides the Apache host server. WebFlow components (or modules) implement services such as job submission and file transfer and can be extended to provide other services. WebFlow servers are hierarchical, so a single master server acts as the parent and gatekeeper server for the distributed children. Clients (in this case, a JSP page) do not contact child servers directly, but instead contact the parent server and request child servers by name. The parent then forwards requests to the appropriate child.

The final layer in the system is the HPC layer. This is a Unix layer that will be accessed through the middle tier. Commands in Unix will be scripted and issued from the middle tier using the user's credentials. The HPC layer has functionality like GRD already built into the system. Therefore, this functionality will be leveraged by the Gateway GUI. This allows the client to pay attention to the task at hand (getting CPU cycles for his simulation) rather than worrying about GRD updates or changes to the bottom layer configuration.† As new systems come on line, modulus at the center tier will be upgraded to improve Gateway functionality.

The four-tiered system will work the same way as the three-tiered system except the Web server and servlet engine will be behind a firewall. These will be accessed through a proxy server outside the firewall. This added layer of security prevents potential would-be intruders from knowing the Internet protocol address of the machine where Gateway is installed.

5. Security Model

As seen in figure 2, Gateway requires several layers of security:

(1) The browser must make a secure connection to the Web server.

(2) The JSP pages must make a secure connection to the WebFlow parent server.

(3) The WebFlow parent and child servers must have a secure connection.

(4) WebFlow connections to the back end (such as HPCs) must be secure.

* Common Object Request Broker Architecture (CORBA) is an open distributed object infrastructure being standardized by the Object Management Group (OMG) (*5*).

† The HPC facility is constantly undergoing upgrades and improvements to their systems. For example, older Silicon Graphics Incorporated (SGI) equipment is being replaced by newer IBM CPUs this year. These changes can effect the way users submit and run jobs on the machine. By creating this portal, the client's environment becomes stable and the burden of staying up to date is placed on HPC personnel.

We address the following security issues: authentication, authorization, privacy, and data integrity. We built the first two on top of the existing Kerberos security infrastructure at the MSRCs, and the last two are subsumed by encryption mechanisms (*6*).

A portal administrator starts the Apache, Tomcat, and WebFlow master servers. Presumably, this will be done in a special "Gateway" account, and we will assume that these servers run under that account. These accounts are checked and maintained as a system administration function within HPC. The server machine may or may not include network file system (NFS)-mounted directories and is assumed to be able to reach HPC machines and mass storage via Kerberized rsh and rcp. It will be assumed that the administrator has a forwardable Kerberos ticket. For production portal use, this ticket should be renewable via a cron job. The Apache server is set up to support SLL-encrypted logins using 128-bit encryption. This server was built using OpenSSL and ModSSL.

Users log in to the portal via a login page that prompts them for their principal name (such as swilker or pierceme@asc.hpc.mil), password, and passcode. When the user submits this form (via HTTP POST) the server runs a kinit on the remote system and creates a ticket for the user. The ticket is stored in /tmp and will be named krb5cc_<principal_name+timestamp> (the time stamp guarantees the ticket name is unique). A simple Expect script is used to run the kinit on the server.

One of the features of servlet engines, such as Tomcat, is that they maintain session states for users automatically. When a user accesses a particular Tomcat server (indirectly, through Apache), a session cookie is generated for him. Tomcat uses this cookie to maintain session state for the user; so for example, JSP pages requiring extensive initialization will only need to be initialized once per session, and user interactions can be stored in memory. Session lifetime is configurable, but typically lasts about 2 hr. We will take advantage of this built-in feature of Tomcat to support user authentication and identification, as we now describe.

We have developed a special authentication component to manage user session tickets. Our mechanisms are based on the ideas of Kelly Kirk (ARL PET Enabling Technologies On-Site Lead) that have been previously reviewed by the HPCMP (*6*). The authentication component is included in all Gateway JSP pages. When the user logs in and successfully gets a ticket, a unique HTTP cookie is created for him. This cookie is a message digest (using an MD5 algorithm) of the contents of his ticket file. Any subsequent access to Gateway Web pages is verified by comparing this encrypted cookie to the value stored in memory on the server. Additionally, the IP address of the user's Web browser is stored in memory. Subsequent page requests are verified to come from the same IP address as the initial request used at login.

Users may also manage their sessions through the browser by deleting their session credentials. This will remove the session ID and IP addresses stored on the server, delete the

user's Kerberos credential from the file system, and set the ticket-based session cookie's lifespan to zero (causing immediate expiration). The session credentials will also be deleted automatically if the session expires (say, in 2 hr) and the user will need to log in again to access the Gateway Web pages. Session credentials for all users will be deleted if the Tomcat server is shut down.

In summary, the Gateway pages can only be accessed if the user has performed a successful kinit. Thereafter, the user is identified by a unique Tomcat session cookie, a unique session cookie generated from his ticket granting ticket (TGT), and the IP address he used to initiate the session. All Web access is encrypted. We can also require the browser to send a properly signed client certificate as an additional means of identification.

For this application, JSP pages make use of services provided by WebFlow servers. The WebFlow servers use CORBA Object Request Brokers (ORBs) for remote client-server connections. The CORBA security service is a general set of interfaces for making these client-server connections secure, and these services can be built using different security mechanisms. We use the ORBAcus* ORB from Object Oriented Concepts in WebFlow and have purchased a security service extension that uses Kerberos from Adiron Software.

In Gateway, JSP pages act as clients to the WebFlow servers. Both the JSP client and the WebFlow server are required to have valid credentials. For the JSP pages, this is the TGT created when a Gateway administrator logged in to the account. The WebFlow parent server, which runs on the same machine as the Apache and Tomcat servers, must access a keytab file. The Kerberos implementation of CORBA security requires that all Kerberos services use a keytab for authentication. For testbed implementations of Gateway, we have used specially generated keytab files that are owned by the Gateway account and are tied to a single server machine.

WebFlow servers can potentially be distributed on multiple machines as child servers to the parent. Wire communication between the servers goes over SECIOP, a secure CORBA protocol that supports generic security services application programming interface mechanisms such as Kerberos. SECIOP supports Data Encryption Standard (DES)† encryption and MD5 hashing. These child servers are required to possess access to a keytab file and must authenticate themselves to the parent. These servers would also need access to keytab files. We do not currently make use of this feature with the ARL Gateway server.

Interactions with any remote machines are handled through Kerberized rsh and rcp. For example, if the queuing system of a remote machine needs to be queried to learn a job status, a WebFlow module runs the qsub command through rsh as an external call.

---

* ORBacus is a fully CORBA-compliant ORB that is distributed as source code. ORBAcus is a product of Object Oriented Concepts, now owned by Iona Software. See <www.ooc.com>.

† DES Encryption originally published in 1977 with the official backing of the U.S. government.

Job submission is handled in the following manner: When a user logs in to the Web server, a TGT is generated for him on the server. This file is owned by the Gateway account. The user's identity (principal name) is maintained as part of his/her session (as described previously). When the user wishes to submit a script to a particular queuing system, the server does this as a remote process with the environment variable KRB5CCNAME set to the appropriate credential. Thus the user's credential is owned by the Gateway account, but the job is submitted as the user's.

Other services may be handled in the same manner. For example, the user's file system might be accessed through an rsh command to determine the files in his/her directory. The user may move files between different remote machines using rcp, with ownership preserved. Files transferred between the user's desktop and the server machine are handled in a different manner, using encrypted HTTP upload and download mechanisms, following proper Web authentication previously described.

We finally note that WebFlow servers may accomplish these same tasks directly. For instance, by the same process just described, the master server can create a WebFlow child process via rsh that will run as the appropriate user. This process could then access the user's file system as needed, submit jobs, etc.

6. Gateway GUI Interface

The Gateway Interface consists of several screens that are reviewed here. The basic "look" and "feel" can be changed or updated at any time. The GUI starts a session with an HTTP login screen as shown in figure 3.

In the top box, the user inserts his/her user name and the realm that he/she wishes to authenticate as well (e.g., swilker@arl.hpc.mil). That is followed by a password and then the authentication ID from the user's SecureID card. The server authenticates the user and creates a ticket for the user on the realm that the user has permission to be on.

After some initial screens with the message of the day, the screen shown in figure 4 appears. This screen consists of a "submit job" button, an "archive" button, and a "portal admin" button. Along the top of the window, there are buttons for a file browser and a job monitor; these buttons are also repeated along the left-side frame. The buttons along the top provide the user with a new independent window. The buttons along the side do the same function but use the existing frames' main window. A logout window is also provided to allow the user to exit the session and kill his/her tickets. The file browser feature provides a connection between the user's machine and HPC storage. The user can then transfer files to HPC from

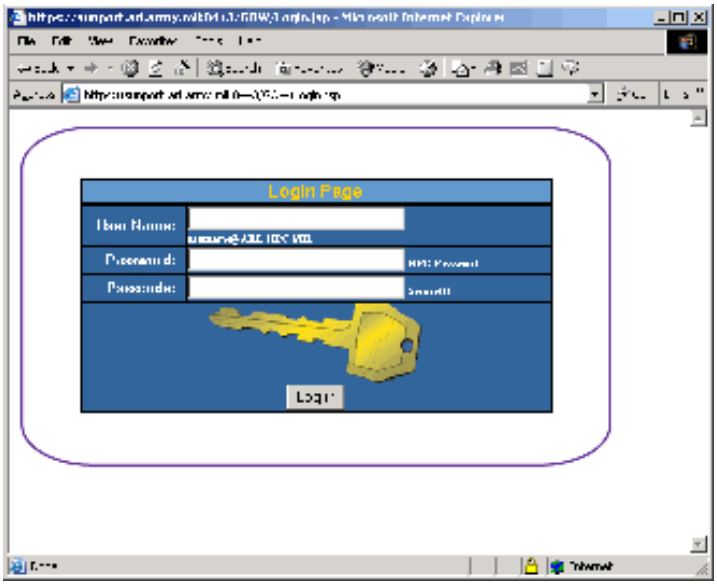

Figure 3. Login screen.

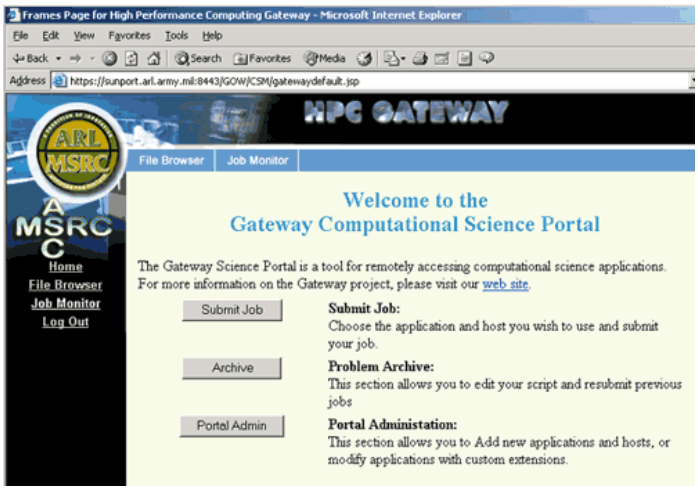

Figure 4. Main screen.

his/her local machine and retrieve files from HPC to his/her local machine. There are plans to extend this capability, and this will be discussed in the conclusion section. The Job Monitor feature queries HPC GRD for jobs that are currently submitted. All of the jobs in the queue are then displayed. The user is additionally allowed to delete a job from the queue or stop a job.

The job submission module is designed to allow other sites and users the ability to add software to the access list, write their own scripts, and customize the job submission process. The current portal includes modules for three different codes. These are ANSYS, ZNS-FLOW, and Fluent. A snapshot of the screen is shown in figure 5.

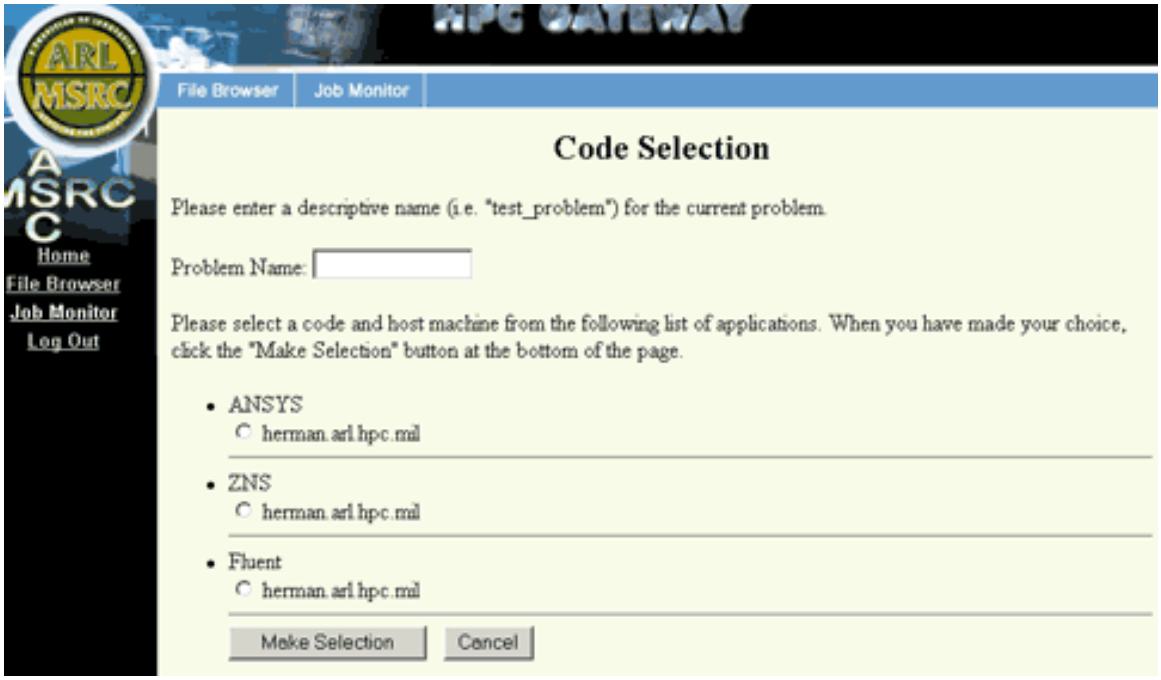

Figure 5. Submitting jobs.

For each of the options available, a problem name is supplied, and the script will create a directory on the user's HPC home directory with that name and a date as part of the name. The GRD scripts program input files and output files are all stored there. Then, using the archive feature, the user can easily return to the same job again after, for instance, he/she changes the input file. This allows users to correct mistakes and quickly resubmit jobs to HPC for simulation. A job submission using ANSYS is shown in figure 6 for reference.

In this screen, the user selects the wall time for the run and the complex, although not currently working, so anything will work here (e.g., SGI, an input file, and an output file). The script then writes a GRD script and supplies it via an editable text window. This can be changed by the user or used "*as-is*" to submit the job.

Figure 7 shows a cut-in-half view of the GRD script in a text window. After you submit the job, you can check on the job's progress using the job monitor. Jobs can be stopped and then resubmitted using the archive feature. The procedures followed here will vary between different application software packages. However, the functionality remains the same, i.e., (1) the user authenticates and gets a Kerberos ticket, (2) files are moved between the user's machine and HPC machine using the interface, (3) a series of questions, pertaining to the running of an application, are answered using some GUI interface, and (4) the user submits his/her job through the GRD interface using the script provided.

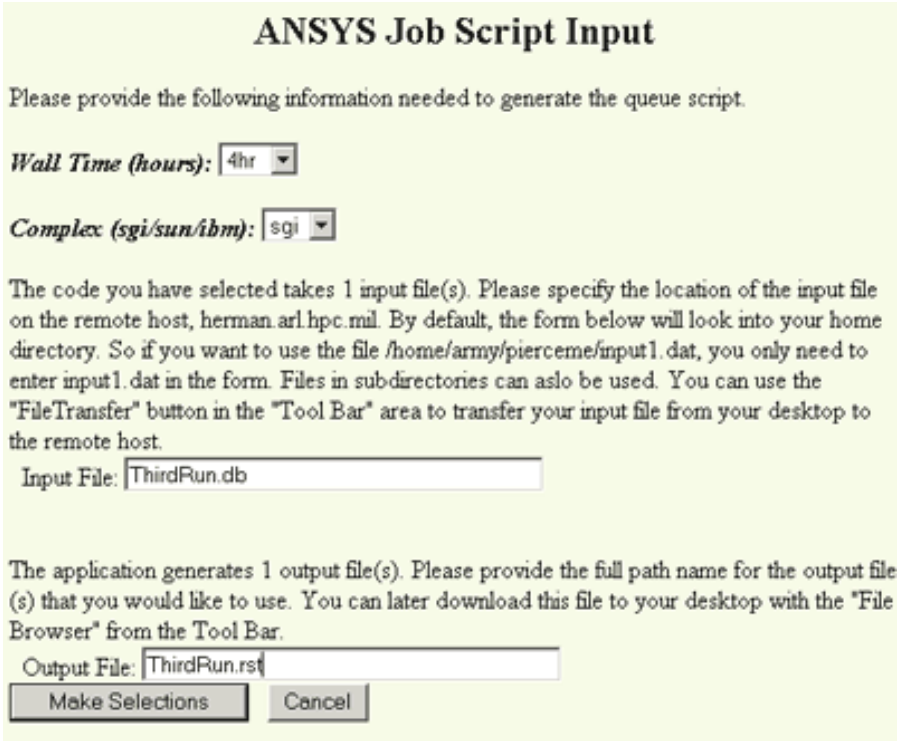

Figure 6. Submitting an ANSYS job.

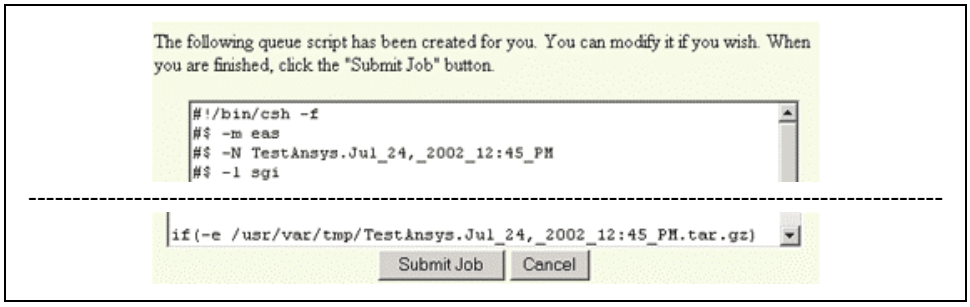

Figure 7. GRD script for ANSYS job.

7. Conclusions

An open-source vanilla Kerberos-enabled Web portal now exists in which a user can transfer files and submit computer runs to the HPC. The system is open source and can be adapted for use at any of the HPC sites and elsewhere. The portal will only be viable if users who need this HPC capability are identified and then the developers work with the users to make the interface user friendly. The advantage to having this relationship is obvious. For starters, the client buys into the system, but more importantly is the fact that the developers don't usually know the user's needs or wants. The user knows his code and knows what will help

him do his/her work more efficiently. At the same time, the developers have knowledge of the HPC architecture and can smooth over the coding difficulties to provide HPC with an interface that everyone is happy with. The marriage between the two sciences is a good one.

Future efforts will include a Kerberos VNC and a more robust version of the FTP. The VNC will allow users to access an HPC or science visualization computer, as if he/she were setting in front of it. The implementation of this feature will avoid the need for users to physically walk to other sites to complete their work. Another big advantage is that files can be left on one machine and accessed without time-consuming file transfers. Additionally, the user gets the advantage of using a powerful machine with lots of high-end graphic capabilities without the burden of purchasing expensive hardware and software. The file transfer feature will also be enhanced. Currently, file transfers can be done within the Unix environment on HPC. However, this is a command line driven procedure. With this interface, the user will be able to drag and drop whole directory structures using Java-signed Applets from one system to the next, thus greatly simplifying the process of pre- and post-processing. The Web interface will include this feature. Additionally, there will be a Java stand-alone program that will also provide the same functionality using a local machine's HPC Kerberos tickets. The advantage of the stand-alone program is in its use of the local Kerberos authentication. The disadvantage is that the users will need to have the current version of Java JDK library installed on the local machine to make the program function properly.

8. References

1. Flanagan, D. *Java in a Nutshell*; O'Reilly Press: California, 2002.
2. Fields, D. K.; Kolb, M. A. *Java Server Pages*; Manning Press: Connecticut, 2000; Java is an open-source programming language provided by SUN Microsystems. See <http://www.sun.com>.
3. Monson-Hafel, R. *Enterprise Java Beans*; O'Reilly Press: California, 2001.
4. Tremblett, P. *Instant Java Server Pages*; McGraw-Hill Press: New York, 2000.
5. Adiron LLC implements the CORBA security service in Kerberos: <www.adiron.com>.
6. Marlon, P. *Gateway Security Model*; White Paper ASC-MSRC; University of Indiana, IN, 2002.

NO. OF COPIES ORGANIZATION

1 (PDF Only) DEFENSE TECHNICAL INFORMATION CTR
DTIC OCA
8725 JOHN J KINGMAN RD
STE 0944
FT BELVOIR VA 22060-6218

1 COMMANDING GENERAL
US ARMY MATERIEL CMD
AMCRDA TF
5001 EISENHOWER AVE
ALEXANDRIA VA 22333-0001

1 INST FOR ADVNCD TCHNLGY
THE UNIV OF TEXAS
AT AUSTIN
3925 W BRAKER LN STE 400
AUSTIN TX 78759-5316

1 US MILITARY ACADEMY
MATH SCI CTR EXCELLENCE
MADN MATH
THAYER HALL
WEST POINT NY 10996-1786

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRD ARL D
DR D SMITH
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRD ARL CS IS R
2800 POWDER MILL RD
ADELPHI MD 20783-1197

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRD ARL CI OK TL
2800 POWDER MILL RD
ADELPHI MD 20783-1197

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRD ARL CS IS T
2800 POWDER MILL RD
ADELPHI MD 20783-1197

NO. OF COPIES ORGANIZATION

ABERDEEN PROVING GROUND

1 DIR USARL
AMSRD ARL CI OK TP (BLDG 4600)