GA–A25616

SECURE FEDERATED LIGHT-WEIGHT WEB PORTALS FOR FUSIONGRID

by
D. ASWATH, M. THOMPSON, M. GOODE, X. LEE, and N.Y. KIM

OCTOBER 2006

GENERAL ATOMICS

DISCLAIMER

GA–A25616

SECURE FEDERATED LIGHT-WEIGHT WEB PORTALS FOR FUSIONGRID

by
D. ASWATH, M. THOMPSON,* M. GOODE,* X. LEE, and N.Y. KIM

This is a preprint of a paper to be presented at the Second International Workshop on Grid Computing Environments, November 12-13, 2006, Tampa, Florida.

* Lawrence Berkley National Laboratory, San Francisco, California

Work supported by
the U.S. Department of Energy
under DE-FG02-01ER25455 and DE-AC03-76SF00098

GENERAL ATOMICS PROJECT 30106
OCTOBER 2006

GENERAL ATOMICS

ABSTRACT

The FusionGrid infrastructure provides a collaborative virtual environment for secure sharing of computation, visualization and data resources over the Internet to support the scientific needs of the US magnetic fusion community. Invoking FusionGrid computational services is typically done through client software written in, for historical reasons, the commercial language IDL. Scientists use these clients to prepare input data and launch FusionGrid computational services. There are also numerous web sites throughout the US dedicated to fusion research, functioning as lightweight single purpose portals. Within the FusionGrid alone, there are web sites associated with authentication, authorization, and monitoring of services. Pubcookie and MyProxy technology were used to federate these disparate web sites by enabling them to authenticate a user by their FusionGrid ID and then to securely invoke FusionGrid computational services. As a result of this drop-in authentication mechanism, portals were created that allow easier usage of FusionGrid services by the US fusion community. The shared authentication mechanism was accomplished by the integration of Pubcookie's single sign-on mechanism with the MyProxy credential repository that was already in use by the FusionGrid. This paper will outline the implementation of the FusionGrid portal technology, discuss specific use cases for both invoking secure services and unifying disparate web sites, present lessons learned from this activity, and discuss future work.

I. INTRODUCTION

The National Fusion Collaboratory Project [1,2] teams the three major U.S. fusion physics research centers: the Princeton Plasma Physics Lab, General Atomics (GA), MIT Plasma and Fusion Science Center, with collaborators from the computer science groups at Princeton, Argonne National Labs (ANL), Lawrence Berkeley National Labs (LBNL) and University of Utah. This project created a national Fusion Energy Sciences Grid (FusionGrid) [3] to provide new capabilities to fusion scientists to advance fusion research. FusionGrid is a system for secure sharing of computation, visualization, and data resources over the Internet. The FusionGrid goal is to allow scientists at remote sites to fully participate in experimental and computational activities as if they were working at a common site thereby creating a virtual organization (VO) of the US Fusion community. The Grid's resources are protected by a shared security infrastructure including strong authentication to identify users and fine-grain authorization to allow stakeholders to control their resources. FusionGrid uses the X.509 certificate standard and the FusionGrid Certificate Authority (CA) to implement Public Key Infrastructure (PKI) for secure communication.

Fundamental to the deployment of FusionGrid into the everyday working environment of US scientists is the usage of the web browser client to deliver some of FusionGrid's capabilities. Such web browser functions include a Fusion Grid Monitor (FGM) [4], hosted at General Atomics, for monitoring the execution of FusionGrid jobs and a preliminary site hosted at LBNL [5] for user registration and management. Combining these new capabilities with the numerous existing US Fusion web sites that contain documentation and other information relevant to perform sciences on FusionGrid has resulted in a large number of web servers spread across the US that serve some aspect of FusionGrid functionality. A separate project investigated the usage of a Java portal, but having a single general-purpose portal did not correspond to the realities of the highly distributed VO with a significant number of legacy web sites.

Access to FusionGrid's computational service is done through client programs that depend on Globus Secure Infrastructure (GSI) [6] to do secure data access and secure job submission. These client programs have been written in the Interactive Data Language (IDL) [7], a commercial software analysis and programming language that is very commonly used with the experimental US fusion community. There are two problems posed by this solution: Globus Toolkit [8] is not available on Windows and requires a fairly complicated installation procedure on UNIX, and IDL, is not available on every potential client machine, since it requires buying a license for each host. Thus a simple web interface that would allow data marshalling and job submission is desirable, as it would allow easy client usage from any web browser capable computer. This web

interface also has to be enabled to leverage off a single sign-on authentication scheme to get a proxy certificate for the scientist, that is required by the grid middleware for remote job submissions and data access.

II. RELATED WORK

The majority of the community's work on creating scientific portals has been done in Java, leveraging off some popular containers for Java servlet code such as Apache Tomcat [9], Jetty [10], Jboss [11], WebSphere [12], and the Java portlet specification released in October 2003 [13]. The goal of such a portal is to provide a single point of entry to all the functions of a VO, and some of the commonly provided functions include: shared spaces such as chat, calendar and newsgroups, whiteboards, shared applications and group authorizations. Grid Portal containers such as OCGE [14] and GridSphere [15] also provide the X.509 authentications, grid style job submissions and grid data transfers. To provide a GUI interface for marshalling data and setting parameters for running a particular code and to further encapsulate the code within such a portal, the developers must be knowledgeable about the scientific code being called and the tools and libraries that are provided by the portal.

Typically portals are designed to run centrally at one site providing access to all of the VO services. Installing and maintaining an integrated portal is a non-trivial undertaking, complicated by the fact that the state-of-the-art portals are large rapidly evolving software projects, based on frequently changing third party software, e.g. portlet containers, portlet standards and authentication approaches. The National Fusion Collaboratory Project worked with the developers of an OGCE portal to deploy a full-service grid portal for FusionGrid. While deployment of the provided portal framework was accomplished, the handoff of the maintenance and further development by fusion scientists, who were neither Java programmers nor portal experts, was not successful. Addition of a new code or service could only be done by users who understood the code to be executed, its portal environment and with administrative access to the site at which the portal was run. In practice, the fusion scientists who understood how to execute a code lacked the portal expertise to integrate their interface into the portal.

Another common approach to hiding the complexities of running a scientific code is to first wrap the code with a simple command line or a GUI interface, prompting for the required parameters and then run the command as a Common Gateway Interface (CGI) script behind a web page; thereby creating a single purpose portal. To launch the code as a grid service, this approach must authenticate the user as a member of the VO permitted to run the service and subsequently retrieve a grid proxy credential on behalf of the user, for use in the Globus job submissions.

Pubcookie [16] as an open source software package uses cookies and a central secure login server to enable a set of trusted web sites to effect a single, authenticated sign-on to all the web sites. The login server is the only site that needs to see the user's password for

authentication. The authenticated user's ID is conveyed to the other web sites in encrypted cookies that can only be decrypted by the login server and the targeted sites. Notably, the user or anyone who gains possession of the cookie cannot alter its contents without invalidating it. Pubcookie is implemented on each of the trusted web sites as an Apache module. The login server allows the authentication mechanism to be provided as a plug-in module enabling the deployer to decide if user names and passwords are to be kept in a simple database, an ldap server, Kerberos or some other means. Additionally, MyProxy [17] as a standard open-source grid server provides X.509 proxy credentials, suitable for use in GSI transactions, when provided with a user name and either the user's password or a trusted credential. As the FusionGrid was already running a MyProxy server as part of the centralized certificate management service, combining it with Pubcookie was an obvious approach to provide an authentication and delegation service that allows existing single-purpose web sites to authenticate FusionGrid members and securely access remote data and submit jobs, through the GSI infrastructure.

Other work has also been done to integrate these two packages. The MyProxy server has an authentication plug-in module, which allows it to authenticate a user via a Pubcookie cookie. The National Virtual Observatory has done a similar integration of MyProxy, Pubcookie and PURSe (Portal-based User Registration System) [18] to provide proxy certificates to its portals.

III. FEDERATED WEB PORTALS

A. Overview

As described in the previous two sections, the National Fusion Collaboratory Project aimed to provide browser accessible GUIs for job submission on the FusionGrid. These single-purpose portals would help lead the user through the data preparation stages, explain and set parameters, record input for future reference or reuse, invoke the service, monitor the process and make results available to the user. In order to succeed within the FusionGrid environment, it is necessary that such interfaces be written by the service provider in a language of their choice, requiring minimum additions to the standard Apache web server installations.

The major challenges to providing such single-purpose portals is the ability to provide single sign-on across all portals a user might require; to get the necessary grid credentials that enable the client-side software to make a GSI-enabled call to a FusionGrid service, and to provide access to the Globus software from within the portal.

B. Approach

We combined several existing software modules to provide a Federated Portal Framework, namely: a Pubcookie module providing single-signon for the set of trusted web servers; a MyProxy server handling the storage of long term credentials and delegation and storage of short-term proxies needed for GSI [4]; the Credential Manager handling the registration of users and the management of long-term credentials; The combination of the four components: Credential Manager, Pubcookie login server and two instances of MyProxy one for long-term and another for short-term credentials is referred to as the Authentication and Delegation Service (ADS). Each of these servers is co-located on the same host, so that the connections between them are automatically secure. The FusionGrid authorization server, ROAM [19] is used by a FusionGrid service to check a user's access to the specific grid resource based on the Common Name (CN) in the user's X.509 certificate. Figure 1 presents an overview of the architecture and the details on how it is used.

The infrastructure for the portal architecture consists of:

- A set of servers running on a secure and trusted host (***ADS***)
- A set of trusted web-interfaces that support https, cookies and an Apache Pubcookie module,
- A FusionGrid authorization server ROAM.

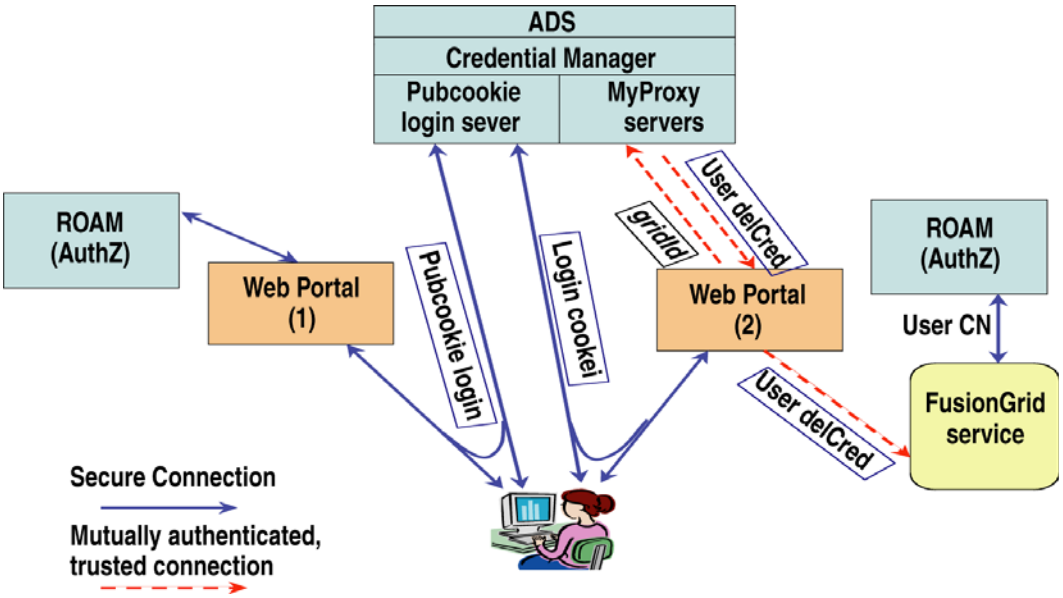


Fig. 1. Federated Portal Architecture.

C. Credential Manager

A user on registering with the FusionGrid chooses a login ID, called the *gridId* used for their subsequent logins. The first and last name provided as part of the user's registration process is used to define the Common Name (CN) that forms a part of the X.509 credential issued to the user. The Credential Manager enters these long-term credentials into a MyProxy server, indexed by *gridId* and encrypted by the user's pass-phrase.

D. Pubcookie

The Pubcookie framework consists of an Apache module deployed on all of the trusted web portals and a central login server providing the basic multi-site/single sign-on ability for web sites in the same domain, e.g. fusiongrid.org. When a user connects, the Pubcookie module looks for a session cookie and if such a signed-cookie containing the user's *gridId* is provided, it knows that the user has been authenticated. However, in the absence of such a cookie, it redirects the request to the Pubcookie login server. The first time a user is redirected to the login server, they are presented with a login form prompting them for a *gridId* and password. The login server authenticates the user with the *gridId* and password provided thereby returning two cookies: a *granting cookie* scoped to reach the web portal that was originally contacted and a *login-cookie* scoped to be returned to it on access to any other web portal.

When a different portal is first contacted, the redirection to the login server contains the login-cookie. The login server uses this cookie to authenticate the user without prompting them for a password again. It then generates a new granting-cookie, which is subsequently returned to the second web portal.

E. MyProxy server with Pubcookie

FusionGrid runs two MyProxy servers, a *MyProxy Credential Store* and a *MyProxy Proxy store*. The first MyProxy server is deployed to store long-term credentials in the *CredentialStore*. For FusionGrid jobs that are submitted directly by a user, a short-term proxy is delegated from the *CredentialStore* using the *gridId* and password provided by the user.

A second MyProxy server is deployed to store short-term proxies in the *ProxyStore* to support proxy renewals by long-running FusionGrid services. These proxies can be used for delegation by a trusted server presenting its own X.509 credential. This style of delegation can be used by portals to get the proxy certificate for an authenticated user to submit Globus jobs on their behalf.

Since Pubcookie is using MyProxy to authenticate the *gridId* and password via a *myproxy-login* interface, it is natural to store the resulting proxy in the short-term ProxyStore. These proxies are set to allow delegation only by the trusted web portals. Thus when a web portal needs a proxy certificate to do a Globus request, it can contact the short-term MyProxy server to get one. This requires that each web portal have its own X.509 service certificate, which is registered with the ADS. In order to co-ordinate the Pubcookie password authentication with that of MyProxy, an authentication plug-in was added to Pubcookie that calls MyProxy to check the password. A side effect of this call is the issuing of a proxy credential.

The process of securely launching FusionGrid computational codes with the Federated Portal Architecture has the following steps as shown in Fig. 2:

1. The user connects to the web portal to launch a specific FusionGrid service.
2. If the user has not previously authenticated, the Pubcookie module redirects the request to the Pubcookie login server.
3. The Pubcookie login server requires the user to sign-on with Fusion-Grid credentials.
4. User is authenticated with the ADS server and a short-term proxy is delegated from stored long-term credentials
5. The short-term proxy is placed in the secondary MyProxy server for subsequent portal retrieval.
6. Upon a successful authentication, the user is sent a ‘redirect’ page and is granted a login cookie. The login cookie is used on any subsequent visits by the user to the login server (single sign-on capability)
7. The ‘redirect’ page causes the user’s browser to re-connect to the original web portal, this time with the granting cookie.

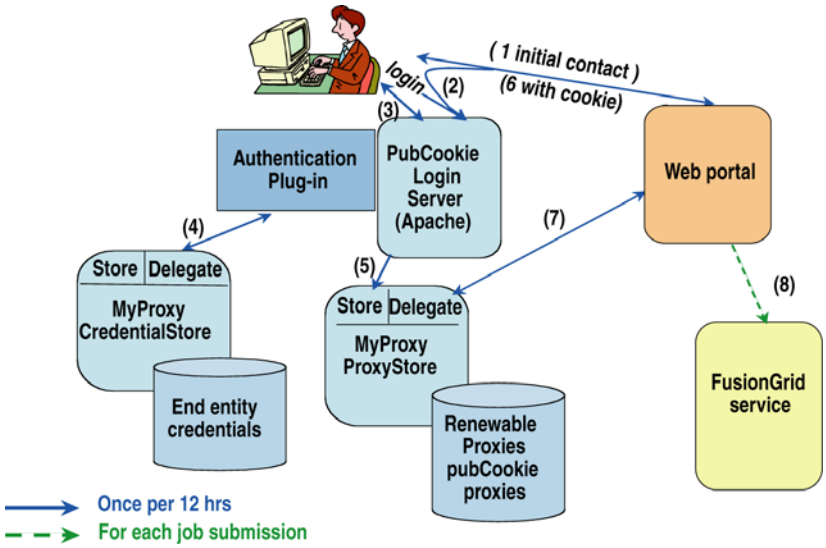


Fig. 2. Details of Authentication and Delegation.

IV. WEB PORTAL USE CASE

ONETWO [20] is a time dependent magnetic fusion analysis and simulation code that is available as a computational service on the FusionGrid. This grid-enabled code running on a cluster of Linux machines can be invoked directly from General Atomics (GA) locally or remotely via FusionGrid as shown in Fig. 3.

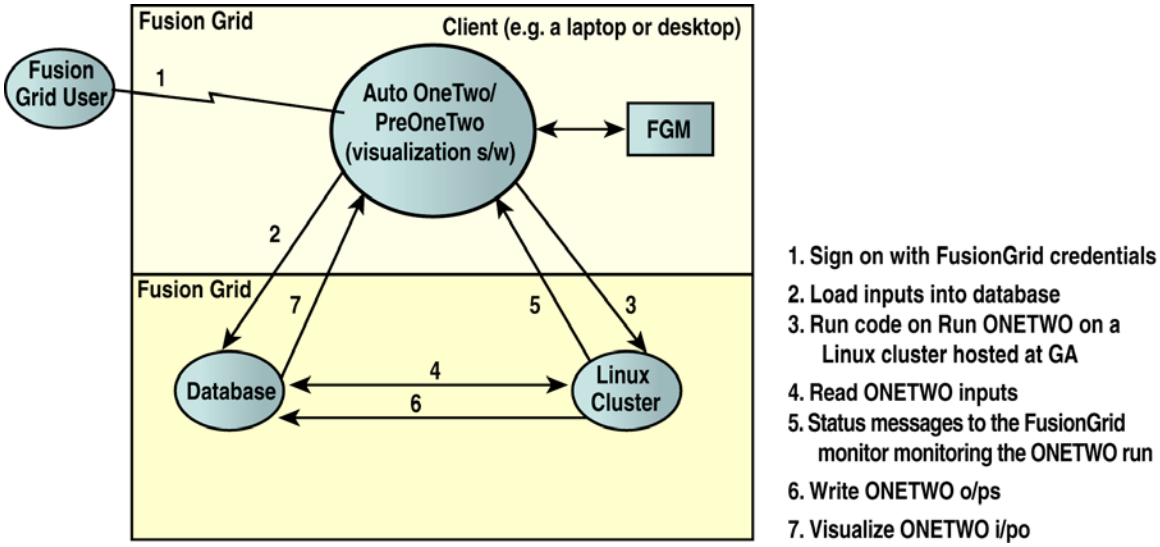


Fig. 3. ONETWO as a FusionGrid service.

AUTOONETWO and PREONETWO are IDL-based client-side GUI tools hosted at GA to help scientists in preparing ONETWO runs on the FusionGrid. Though differing in the way the inputs are gathered and processed and the specific scientific problem they address, these GUI tools manages code runs with a code run database, and upload the inputs prepared to the ONETWO computational service requesting a new code run; thereby reducing the work required by the user to launch FusionGrid code runs. However these client programs depend on both a Globus infrastructure [8] and a commercial IDL license to be made available on every host machine. With the ongoing effort to have a simple web interface to allow easy client usage from any machine supporting a web browser, we have developed a web portal (Fig. 4) with the Federated Portal Architecture described above to enable authenticated FusionGrid users to securely invoke the ONETWO computational service on the FusionGrid.

When clients attempt to access a Pubcookie protected web page hosted by the web portal, through their browsers, they are prompted for their *gridId* and password by the login server at cert.fusiongrid.org. Upon a successful authentication, a proxy is delegated from the MyProxy server for later retrieval by the portal and the user is redirected to the originally requested page. The portal uses its host credentials to retrieve the proxy

certificate for the authenticated user to start the ONETWO run. The portal queries the ROAM authorization server to check if the authenticated user has permission to access the ONETWO resource. Authorized users are presented with the option to gather and process inputs as shown in Fig. 5. The ONETWO code has hundreds of different input settings but this initial version of the portal interface has only the most commonly changed available for user adjustment. As use of the portal grows more inputs will be added. The general inputs include which fusion plasma shot, what time range within that shot to analyze, and an optional text comment string. The advanced input section includes: where to get the plasma shape (EFIT ID), the plasma temperature and density profiles (ZIPFITS and Profile directory), the input template file that specifics all possible inputs (INONE Template), and specifics about the auxiliary heating of the plasma (NBI and ECH). The inputs thus prepared on a desired shot are inserted into a database and subsequently retrieved by the ONETWO computational code during its run. The run can be monitored through a FusionGrid Monitoring system (FGM) as shown in Fig. 6. The user can then access the results of the run stored in an MDSPlus [21] data repository identified by the FGM logs.

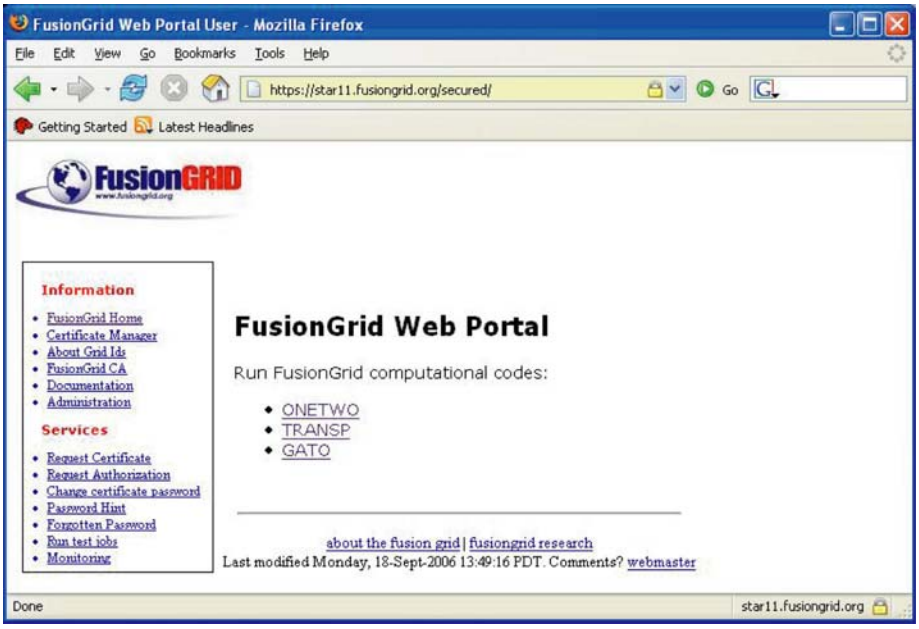


Fig. 4. Web Portal hosting the web page to launch the ONETWO service on the FusionGrid.

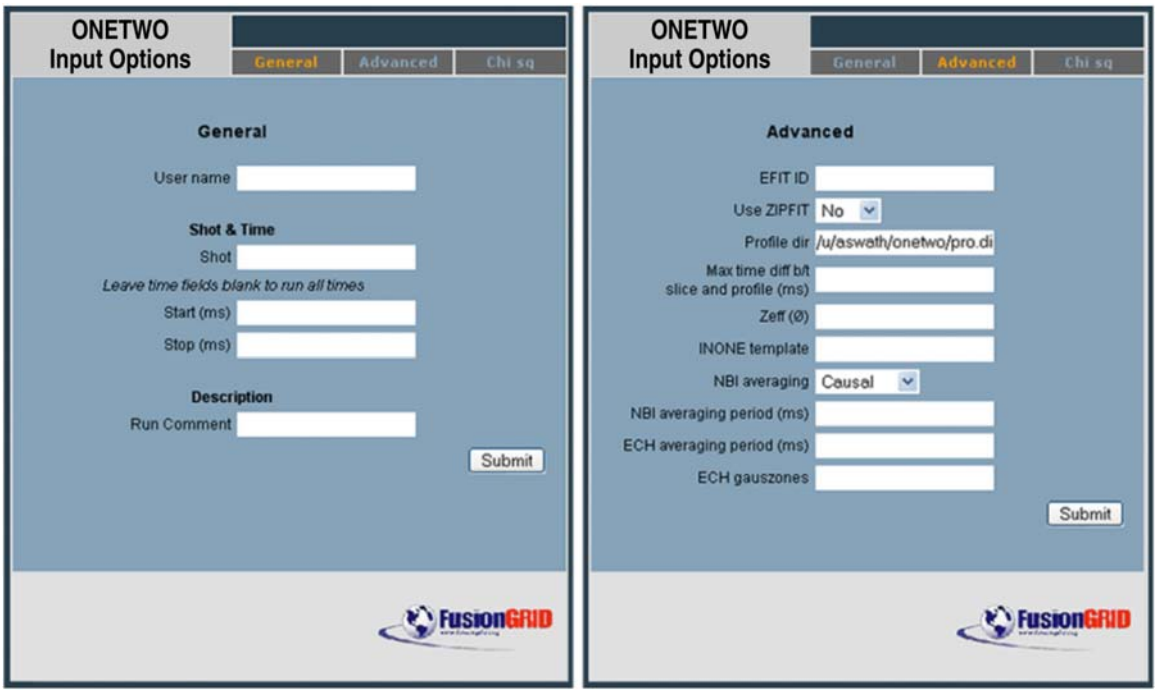
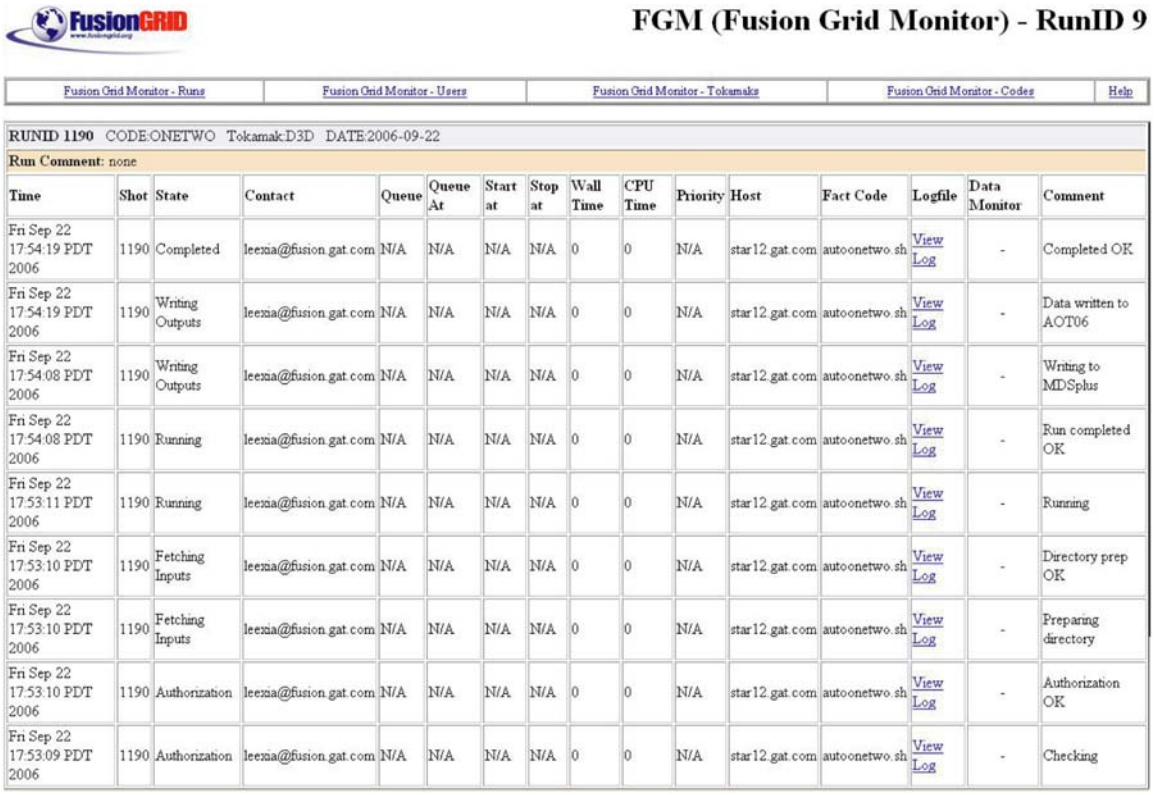


Fig. 5. ONETWO Input Preparation for FusionGrid users authorized to access ONETWO.



FusionGRID

FGM (Fusion Grid Monitor) - RunID 9

Fusion Grid Monitor - Runs | Fusion Grid Monitor - Users | Fusion Grid Monitor - Tokamaks | Fusion Grid Monitor - Codes | Help

RUNID 1190 CODE:ONETWO Tokamak:D3D DATE:2006-09-22

Run Comment: none

Time	Shot	State	Contact	Queue	Queue At	Start at	Stop at	Wall Time	CPU Time	Priority	Host	Fact Code	Logfile	Data Monitor	Comment
Fri Sep 22 17:54:19 PDT 2006	1190	Completed	leexia@fusion.gat.com	N/A	N/A	N/A	N/A	0	0	N/A	star12.gat.com	autoonetwo.sh	View Log	-	Completed OK
Fri Sep 22 17:54:19 PDT 2006	1190	Writing Outputs	leexia@fusion.gat.com	N/A	N/A	N/A	N/A	0	0	N/A	star12.gat.com	autoonetwo.sh	View Log	-	Data written to AOT06
Fri Sep 22 17:54:08 PDT 2006	1190	Writing Outputs	leexia@fusion.gat.com	N/A	N/A	N/A	N/A	0	0	N/A	star12.gat.com	autoonetwo.sh	View Log	-	Writing to MDSplus
Fri Sep 22 17:54:08 PDT 2006	1190	Running	leexia@fusion.gat.com	N/A	N/A	N/A	N/A	0	0	N/A	star12.gat.com	autoonetwo.sh	View Log	-	Run completed OK
Fri Sep 22 17:53:11 PDT 2006	1190	Running	leexia@fusion.gat.com	N/A	N/A	N/A	N/A	0	0	N/A	star12.gat.com	autoonetwo.sh	View Log	-	Running
Fri Sep 22 17:53:10 PDT 2006	1190	Fetching Inputs	leexia@fusion.gat.com	N/A	N/A	N/A	N/A	0	0	N/A	star12.gat.com	autoonetwo.sh	View Log	-	Directory prep OK
Fri Sep 22 17:53:10 PDT 2006	1190	Fetching Inputs	leexia@fusion.gat.com	N/A	N/A	N/A	N/A	0	0	N/A	star12.gat.com	autoonetwo.sh	View Log	-	Preparing directory
Fri Sep 22 17:53:10 PDT 2006	1190	Authorization	leexia@fusion.gat.com	N/A	N/A	N/A	N/A	0	0	N/A	star12.gat.com	autoonetwo.sh	View Log	-	Authorization OK
Fri Sep 22 17:53:09 PDT 2006	1190	Authorization	leexia@fusion.gat.com	N/A	N/A	N/A	N/A	0	0	N/A	star12.gat.com	autoonetwo.sh	View Log	-	Checking

Fig. 6. FusionGrid Monitor (FGM) logs monitoring a ONETWO run with a run id of 1190. Results of this run are stored in the MDSPlus tree AOT06.

V. DISCUSSION AND CONCLUDING REMARKS

With a straightforward implementation, the Federated Web Portal worked as expected to authenticate, authorize and provide a proxy credential for the user and we were successful in launching the ONETWO computational code as a secure grid service on the FusionGrid via the portal. With the MDSPlus repository storing the outputs of the code runs, output visualizations are currently presented to the fusion scientists with existing GA software tools, such as ReviewPlus run locally rather than through the portal. As the scientists have commenced their usage of the portals, we are yet to determine the types of physics codes that can be ported to our portal framework. Our preliminary phase shows that codes with intensive visualizations during the process of input preparation are not suited to be accessed via our web portals. However, those codes that do not require extended graphics such as our ONETWO codes are well suited to be used with our proposed portal architecture. A brief duration of six months should enable us to better make such a decision on the best usage of this technology. It is expected that with a straightforward procedure for a portal site to authenticate, authorize and obtain and use a proxy, the significant work in creating a code portal will be in presenting a convenient and intuitive interface for input and output to FusionGrid services.

For our future work, to further enable visualizations of the code run outputs, we plan on integrating Elvis [22], as the scientific visualization package that would allow users to view graphs in a browser window. As the Federated Portal approach requires each of the web portals to have a fusiongrid.org alias in addition to their primary name, we would like to eliminate this additional requirement. Having migrated to a Wiki-based web site for the DIII-D fusion facility and a Bugzilla system to track requests from users on software updates and possible bugs, we will examine the usage of FusionGrid credentials for the login scheme. As the DIII-D wiki site requires the user's login ID to be tracked to monitor edits on the web pages themselves, this Pubcookie model of authentication with the X.509 FusionGrid credentials would not only secure the DIII-D wiki pages, but would also eliminate the need for users to remember a separate set of login ID's and passwords to be able to access and edit such pages, and make requests via Bugzilla.

REFERENCES

[1] D.P. Schissel, *et al.*, "Building the US National Fusion Grid: Results from the National Fusion Collaboratory project," Fusion Eng. and Design **71**, 245 (2004).

[2] D.P. Schissel, *et al.*, "The National Fusion Collaboratory Project: Applying Grid Technology for Magnetic Fusion Research," Proceedings of the Workshop on Case Studies on Grid Applications at GGF10 (2004).

[3] The National Fusion Collaboratory, http://www.fusiongrid.org.

[4] S.M. Flanagan, J.R. Burruss, C. Ludescher, D.C. McCune, Q. Peng, L. Randerson, D.P. Schissel, "A General Purpose Data Analysis System with Case studies from the National FusionGrid and the DIII-D MDSPlus between pulse analysis system."

[5] J.R. Burruss, T.W. Fredian, M.R. Thompson, "Simplifying FusionGrid Security, Challenges of Large Applications in Distributed Environments (CLADE) workshop at HPDC"14, July 2005, Research Triangle Park, NC.

[6] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke, "Security for Grid Services," Twelfth International Symposium on High Performance Distributed Computing (HPDC-12), IEEE Press June 2003.

[7] The Data Visualization and Analysis Platform (IDL), http://www.ittvis.com/idl/.

[8] Globus, http://www.globus.org/toolkit/docs/2.4/.

[9] Tomcat, http://tomcat.apache.org/.

[10] Jetty, http://www.mortbay.org/.

[11] JBoss, http://labs.jboss.com/portal/jbossportal

[12] WebSphere, http://www-306.ibm.com/software/websphere/

[13] Java Portlets, Final release Oct 2003:

http://jcp.org/aboutJava/communityprocess/final/jsr168/

[14] OGGE, http://www.collab-ogce.org/ogce2/

[15] GridSphere, http://www.gridsphere.org/gridsphere/gridsphere

[16] Pubcookie, http://www.pubcookie.org/

[17] J. Basney, M. Humphrey, and V. Welch, The MyProxy Online Credential Repository, Software: Practice and Experience, Volume 35, Issue 9, July 2005, pages 801-816, also http://grid.ncsa.uiuc.edu/myproxy/

[18] M. Freemon, http://grid.ncsa.uiuc.edu/myproxy/talks.html

[19] J.R. Burruss, T.W. Fredian, M.R. Thompson, "ROAM: An Authorization Manager for Grids," to appear in the fall 2006 in Journal of Grid Computing.

[20] W. Pfeifer, R.H. Davidson, R.L. Miller, and R.E. Waltz, General Atomics Report GA-A16178 (1980).

[21] J.A. Stillerman et al., "MDSPlus," Rev. Sci. Instrum. **68**, 939 (1997).

[22] Elvis, http://w3.pppl.gov/elvis/.

ACKNOWLEDGMENT

This work funded by the SciDAC project, U.S. Department of Energy under contract DE-FG02-01ER25455 and by the Director, Office of Science, Office of Advanced Science, Mathematical, Information and Computation Sciences of the U.S. Department of Energy under contract number DE-AC02-05CH11231. The authors wish to thank D. Schissel for his valuable suggestions on this paper.