Using Dynamic Accounts to Enable Access to Advanced Resources through Science Gateways

Joseph A. Insley
University of Chicago / Argonne National Laboratory
9700 S. Cass Ave.
Argonne, IL 60439
insley@ci.uchicago.edu

Ti Leggett
University of Chicago / Argonne National Laboratory
9700 S. Cass Ave.
Argonne, IL 60439
leggett@ci.uchicago.edu

Michael E. Papka
University of Chicago / Argonne National Laboratory
9700 S. Cass Ave.
Argonne, IL 60439
papka@ci.uchicago.edu

ABSTRACT

Science Gateways have emerged as a valuable solution for providing large numbers of users with access to advanced computing resources. Additionally, they can hide many of the complexities often associated with using such resources effectively. Many gateways make use of a community account, which is shared by all gateway users on the backend compute resource. In some cases this can lead to problems when it comes to segregation of user data. To address this issue we have investigated the use of dynamic accounts, where each gateway user is dynamically allocated their own account on the backend resource. We will describe some of the features of the Dynamic Account service and explain how it has been integrated into the TeraGrid Visualization Gateway. We will also discuss problems encountered, identify remaining open issues, and conclude with directions for future work.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: *Web-based services*

General Terms

Documentation, Design, Security.

Keywords

Science Gateways, Dynamic Accounts, Community Access.

1. INTRODUCTION

An increasing number of scientific disciplines can benefit from the use of advanced computing resources. However, the process for gaining access to such resources can often be a lengthy and competitive process, posing a barrier to many researchers. The National Science Foundation's TeraGrid project [1] has greatly simplified this process through its start-up allocations, allowing researchers to easily request up to 200,000 service units (a service unit being approximately one CPU hour). However, such allocations can still take up to several weeks to process. Additionally, once an allocation is awarded there are many complexities associated with using such resources efficiently. In recent years the concept of science gateways has been increasingly used to remove this barrier to access, as well as hide many complications often encountered with using the resources. Typically focused on a specific scientific domain, science gateways are aimed at enabling entire communities of users, providing access to common applications and services through a simplified interface. In many cases, users of the gateway are not even aware that such high-end resources are being used behind the scenes.

2. MOTIVATION

Our work has been largely motivated by the need for simplified access to visualization and data analysis resources and services. Consider, for example, the following scenario. A computational astrophysicist conducts large-scale calculations that simulate the evolution of the universe. As the simulation advances it is important that he monitor its progress. As each time step of the simulation completes, the resulting data is moved to a resource equipped with specialized graphics hardware and large amounts of memory, which is required for processing the complex data. Here images of the data are produced, using predefined viewpoints and transfer functions that highlight anticipated regions of interest in the data. Images are also combined with renderings from previous time steps and made into animations showing the evolution thus far. As the renderings are completed the images and animations are posted to the researcher's gallery within a gateway. Along with the renderings statistical graphs produced from the data are generated and posted. Information about the initial conditions and variables used in the calculations, as well as how the renderings were produced are also stored here. At any time he can visit the gateway to see the latest data. Because of the large amounts of compute power required to do the computation, the simulation progresses slowly, producing one time step of data every few hours. For this reason he may additionally want register to be notified via email or other communication mechanism (e.g. RSS, Twitter, SMS, Jabber, etc.) when new data becomes available. He can set personal preferences to dictate when and how often notifications should be sent. He may choose to keep all of this information private, or share it with collaborators, allowing them to view the gallery and register for notifications as well.

While viewing one of the images the scientist may notice a feature that he hadn't expected, and would like to investigate in more detail. At this point he launches an interactive visualization session from within the gateway, starting with the same view that was used to generate the image. He interactively explores the data, revealing a new region of interest. From here he can save new images, and create camera paths for generating additional renderings from this newly found perspective.

3. SCIENTIFIC GATEWAY WITH DYNAMIC ACCOUNTS

There are several basic functionalities required to facilitate the preceding scenario. The first is simple access to resources. Many science gateways enable access to advanced backend resources through the use of a community account. Users have individual accounts on the gateway itself, and when advanced computational resources are required the gateway will use a single community account, shared by all gateway users, to access and utilize that resource. This can be an effective method for providing such resources to a large number of users. Another core capability that is required is that of data management, both within the gateway, and on the backend resource. One important aspect that the community account method lacks is that of data segregation. Individual gateway users may each have their own data, but because all users access the backend resource as the community user, the community user owns all of this data on the backend resource. This means that any gateway user can read and write any other gateway user's data. While in many instances it is possible for the gateway to avoid complications that this can pose by managing all user data on the backend resource, this is not always the case. In particular, this can be problematic for third-party interactive applications, such as the ones used in our example above, where the user can browse directories and read and write data. To address this issue we have explored the use of dynamic accounts, where each gateway user is dynamically allocated an individual Unix account on the backend resource. In the remainder of this paper we will describe features of the Dynamic Account service, and detail how this service has been integrated into the TeraGrid Visualization Gateway. We will also discuss issues encountered, identify remaining open issues, and conclude with directions for future work.

3.1 Dynamic Account Service

The Dynamic Account (DA) service [2] is an Incubator Project of the Globus Toolkit [3] that can be used to provision Unix accounts on a Grid resource. These accounts can then be utilized through other Grid services. The DA service can allocate accounts based on a number of different configuration options. It can dynamically create each Unix account on the fly as it is requested, or can draw from a pool of previously created accounts. When the pool of accounts method is used, each time an account is requested the next available account in the pool is used. The DA service then checks the number of available accounts remaining in the pool, and sends a notification to service administrators if that number has fallen below the configured threshold.

The DA service gives the option of recycling accounts, or allowing them to be assigned to just a single user and never reused. There are cases when it would be beneficial to recycle accounts. For instance, it is possible to use the DA service as part of a workflow in which a gateway may manage a large throughput of jobs. Part of the workflow may include requesting an account from the DA service that can then be used to run the job, in effect, using a different user account for each job submission. This, in fact, is one of the use scenarios for which the DA service was initially implemented. In such a case, one would clearly want to reuse accounts, returning them to the pool of available accounts after each use.

The DA service provides a capability for handling inactive accounts. When the account is initially created, a time to live (TTL) can be set for that account. The DA service can then be configured to disable the account and run a clean up script once that TTL has expired. Once the account has been disabled, if the service has been configured to reuse accounts, the account is returned to the pool of available accounts.

The DA service uses Grid credentials for authentication and authorization of users that are allowed to request, and subsequently utilize, accounts through the service. Once an account has been provisioned, the service keeps a list of credentials that are allowed to use the account to interact with other Grid services. For example, the GRAM service [4] is used for remote job submission on a compute resource. Typically when a job request is submitted GRAM consults a local grid-mapfile to map the credential that was used to make the request to an end user account on the resource. Additionally it can make use of an authorization plug-in that interfaces to the DA service. If no match is found in the grid-mapfile, it then contacts the DA service, asking it for the correct mapping. Once this is done, the service can continue processing the request as it normally would. The GridFTP [5] service can similarly consult the DA service when processing data transfer requests.

3.2 TeraGrid Visualization Gateway

The TeraGrid Visualization Gateway [6] provides users with access to advanced visualization resources and services through a Web-based portal interface. Users in need of such resources can take advantage of community access by creating an account on the gateway. Gateway users can then run interactive applications on the University of Chicago/Argonne National Laboratory (UChicago/Argonne) TeraGrid Visualization resource, which is available exclusively through the Gateway. As described previously, the use of interactive applications makes data segregation for community accounts essential. To enable this, we have integrated the Dynamic Account service into the Visualization Gateway, using it to provision local Unix accounts on the UChicago/Argonne resource for each Gateway user. The process for creating these accounts is completely automated, making it painless for users, and requiring human intervention by administrators only for periodic maintenance.

3.2.1 Configuration Considerations

In order to integrate the DA service into the Visualization Gateway a number of configuration and implementation decisions needed to be made. The first of these is related to credentials. Each community user would need an individual Grid credential. To provide these, the gateway runs a local MyProxy [7] server coupled with its own Kerberos Certificate Authority (KCA) [8]. This MyProxy server runs on a virtual host on the same machine that serves the gateway, and is accessible only from this machine. This enabled us to keep the service secure while at the same time remove the human from the loop, streamlining this process.

Next, we needed to decide how accounts would be allocated and later disabled. For this we wanted to take somewhat of a hybrid approach. We wanted a persistent, one-to-one mapping of gateway account to backend Unix account. This would allow for

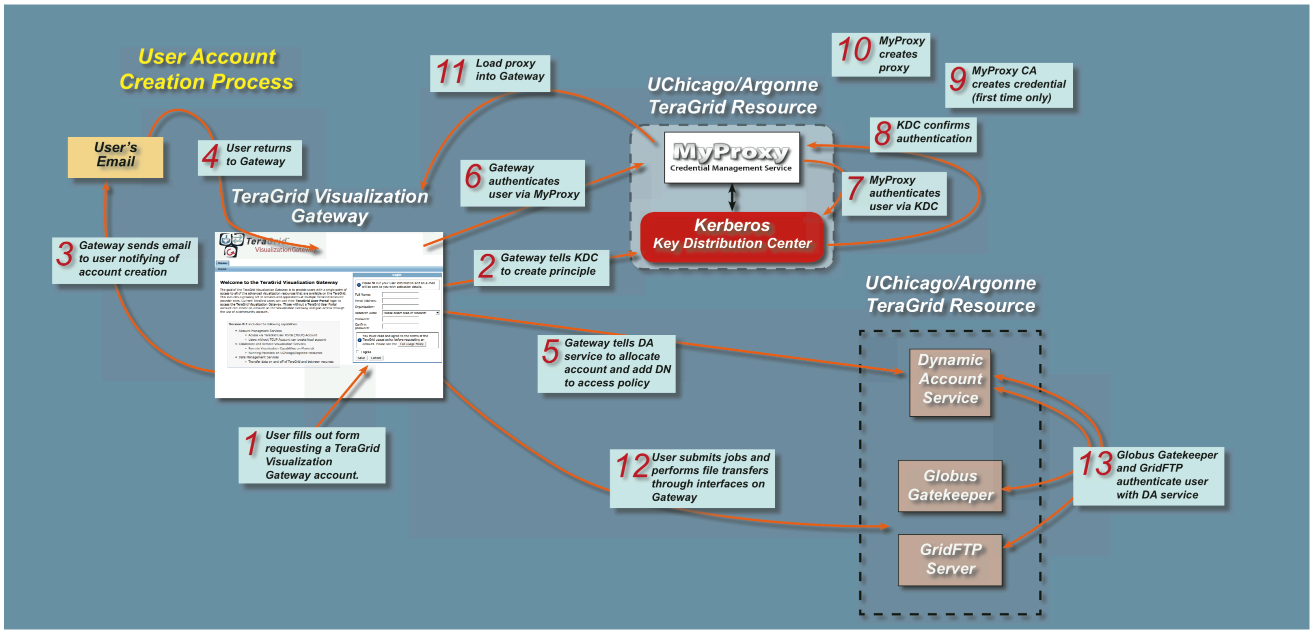

User Account Creation Process
User's Email
4 User returns to Gateway
11 Load proxy into Gateway
UChicago/Argonne TeraGrid Resource
10 MyProxy creates proxy
9 MyProxy CA creates credential (first time only)
8 KDC confirms authentication
MyProxy Credential Management Service
6 Gateway authenticates user via MyProxy
7 MyProxy authenticates user via KDC
TeraGrid Visualization Gateway
Kerberos Key Distribution Center
3 Gateway sends email to user notifying of account creation
2 Gateway tells KDC to create principle
UChicago/Argonne TeraGrid Resource
5 Gateway tells DA service to allocate account and add DN to access policy
Dynamic Account Service
1 User fills out form requesting a TeraGrid Visualization Gateway account.
12 User submits jobs and performs file transfers through interfaces on Gateway
Globus Gatekeeper
13 Globus Gatekeeper and GridFTP authenticate user with DA service
GridFTP Server

Figure 1. User account creation process on the TeraGrid Visualization Gateway using the Dynamic Account service.

simplified usage accounting. All usage by an individual backend account could be attributed to a single gateway user. If accounts were recycled, the gateway would need to keep track of what user was assigned to a particular backend account at a particular time, complicating this accounting process. Because we disable inactive accounts, should a user later return to the gateway we want to reactivate the same backend account that they were previously assigned.

To do this we use the pool of accounts method. We initially created a modest number of accounts and, later, added additional accounts as those were used up. When a user's TTL expires, the DA service is configured to purge any of that user's data and deactivate the backend account but not return the account back to the pool of available accounts. If the user returns to the portal at some later date, the DA service re-enables the mapping to the same backend account automatically. The TTL for the user's account is reset each time the user logs into the gateway. This means that as long as the user frequently visits the gateway, his or her data will remain in place.

Because there is limited vetting of user information when the user's gateway account is created, the backend accounts allocated by the DA service have restricted access. First, they have no access to a login shell; this has been disabled for these accounts on the backend resource. So, users cannot login to the backend resource directly and run arbitrary commands. Similarly, from the gateway side they are also limited to using the specific applications and services exposed by the gateway, and cannot submit jobs to the resource to run arbitrary commands.

When it comes to recording usage for these community users of the gateway, it is charged against a community visualization allocation. All job submissions by each unique gateway user are recorded in a local database, and can be used to audit each user's individual usage.

3.2.2 Account Creation Process

Figure 1 shows the steps taken when a user creates an account on the TeraGrid Visualization Gateway. The user first visits the gateway and fills out a short form, including their choice of password. When that form is submitted the gateway stores this account request information in a local database, identifying it is a 'new' request. It then uses the Kerberos key distribution center (KDC) to create an identity for this user (step 2). A cron job periodically runs on the gateway host, processing requests in the database. It first checks for any 'new' requests. If any are found, it checks to see the whether the user's identify has been propagated throughout the system. If it has not, the request is left marked as 'new'. If the identity has been propagated, the gateway then sends an email message to the user to verify the user's email account (step3), and the status of the request in the database is updated to 'verifying'. Users have a limited time to verify their email account, in our case this is set to two days. When it is done with 'new' requests, the cron job that is processing requests looks for any marked as 'verifying'. If any of these requests are more than two days old, they are removed from the database, and the user would need to submit another request in order to obtain an account.

When the user returns to the gateway by following the link provided in the email message (step 4), the gateway updates the status of the user's request in the database to 'active' then contacts the DA service to request an account on the backend resource for this user, as shown in step 5. Once this account has been allocated the process is complete. At this point the user can login using the password they provided when they requested the account. This password is used to authenticate the user with the MyProxy server (step 6), which in turn verifies the user's identify with the KDC (step 7). If successful (step 8) the MyProxy server checks to see if this is the first time that this user has accessed the gateway. If so, the MyProxy server's Certificate Authority (CA) is used to generate a credential for this user (step 9). MyProxy then creates a proxy credential for this user (step 10) and it is loaded into the gateway (step 11). The user can now make use of the services exposed through the gateway, which contacts the appropriate Globus service on the UChicago/Argonne resource, GRAM for job submission and GridFTP for data transfer (step 12). As described earlier, these services then contact the DA service to map this user's credential to the appropriate account on the back end (step 13).

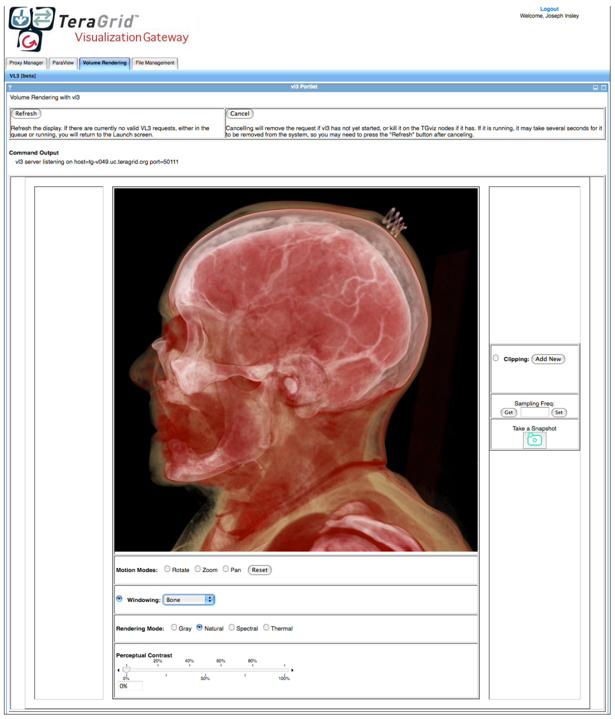
3.2.3 Issues Encountered

One issue that was encountered involved reactivating a disabled user account. The DA service supports this, but does so based on the credential that was used to request the account. Initially, the gateway credential was used to request all backend accounts from the DA service. The credential for the individual gateway user was then added to that account. Since the same gateway credential was always used to make the request, a new user may be assigned to an account that was previously marked as disabled, rather than to the next available, unused account. For example, a user creates an account on the gateway, and the gateway credential is used to request an account from the DA service. The user is assigned to account user00001. The user stays away from the gateway for too long, and the account is inactivated. A different user then requests a new account on the gateway. The gateway again uses its credential to request an account. The DA service recognizes it as the credential used to request account user00001, so instead of taking the next unused account from the pool, this user is assigned to user00001. Clearly this is not the desired result. To resolve this problem, the gateway-supplied credential for the individual gateway user must be used to request the backend account from the DA service. This process was somewhat complicated by the fact that this request happens when the user returns to the gateway after confirming their email address, but before actually logging into the gateway. At this point, the user's credential has not yet been retrieved from the MyProxy server. Therefore, we must temporarily store the user's MyProxy passphrase in the gateway account activation request, and use it to retrieve their credential for them, and request the backend account from the DA service. Once the user's account has been activated, the passphrase is removed from the database.

Another difficulty that was encountered was with login shell access. For security purposes, this was originally disabled on all nodes of the cluster. However, this caused problems when trying to run parallel jobs. The head node of the job logs in to each of the other nodes to run the job executable. With login access to all nodes disabled, this was failing. To remedy this situation, login access was re-enabled on the compute nodes, which are only accessible from within the cluster and not from the outside world, and kept disabled on all publicly accessible hosts.

3.3 File Transfer Service

Another core functionality required to enable the scenario outlined in section 2 is data management. Users need to be able to move data onto and off of the visualization resource. There are a number of file management portlets available from the Open Grid Computing Environments (OGCE) collaboration [9] that could be used for this purpose. Using the user's proxy certificate for authentication and authorization they enable users to transfer data to and from GridFTP servers on various resources. However, using the Dynamic Account service to provision the backend accounts on the UChicago/Argonne resource complicates this process a bit. Because there is limited vetting of users' identity before the gateway issues users a certificate, other resource



Welcome, Joseph Insley

Volume Rendering with vl3

Refresh the display. If there are currently no valid VL3 requests, either in the queue or running, you will return to the Launch screen.

Cancelling will remove the request if vl3 has not yet started, or kill it on the TGviz nodes if it has. If it is running, it may take several seconds for it to be removed from the system, so you may need to press the "Refresh" button after canceling.

Command Output

```
vl3 server listening on host=tg-v049.uc.teragrid.org port=50111
```

Motion Modes: Rotate Zoom Pan

[✓ Windowing]: Bone

Rendering Mode: Gray [✓ Natural] Spectral Thermal

0%

Figure 2. View of the vl3 portlet on the TeraGrid Visualization Gateway being used to investigate anatomical data sets.

providers may be hesitant to accept certificates issued by the gateway's CA. To get past this limitation users need to be able to use their gateway-issued certificate on the gateway side of a transfer, and a different identity on the other side. To enable this we augmented the Comprehensive File Management portlet available from OGCE.

The Proxy Management portlet, also from OGCE, enables users to load proxy credentials into the portal from a MyProxy server. The user then designates one credential to be the default. The file management portlet then uses the current default credential when a transfer is initiated. We extended this capability by presenting the user with a list of all of the credentials that have been loaded, if more than one, and allowing them to choose the desired identity to use for both the source and destination of the transfer.

4. CASE STUDY

In addition to the scenario outlined in the introduction, another environment that can benefit from the use of dynamic accounts within science gateways is that of classrooms and educational settings. For example, we are working with a professor at the University of Chicago who uses volume rendering visualizations to replace cadaver dissection in his undergraduate anatomy class. However, the visualization resources used by the professor during class for instruction and demonstration are the same resources that are used by faculty and staff to conduct their research. Thus, these resources are not available to students outside of class time. To enable students in the anatomy class to explore datasets made available to them by the instructor on their own outside of class, an existing volume rendering service on the Visualization Gateway was expanded to include controls specifically targeted for use with medical data (see Figure 2). Students will now be able to create an account on the Visualization Gateway and make use of TeraGrid resources to do this investigation.

5. FUTURE WORK

As mentioned in section 3.3, some resource providers may be reluctant to accept certificates issued by the gateway CA because of its lack of vetting of user identity. One possible way to remedy this situation, while maintaining minimal human in the loop effort on the part of the gateway, is through the use of federated identity, such as InCommon [10]. The gateway could enable a user requesting a new account on the gateway to present their campus-issued InCommon ID, which likely leverages a strong identity vetting process. Once verified, the gateway CA would issue the user's certificate, augmented with an attribute that indicates that the user's identity was verified via the campus InCommon identity provider. Knowing that the user's identity had been vetted by the campus issuing the InCommon identity may increase the resource provider's trust in the gateway's CA, such that it would accept the certificates that it issues.

A gateway could leverage the resources at a large number of resource provider sites, as long as those sites trust the gateway CA. Each resource provider could run its own dynamic account service to provision local accounts on its resources. Once the gateway validates the user's identity, either by simply verifying their email address or through the federated identity management described above, the gateway could request accounts from all of the DA services using a single credential. Users could then use that same credential to move data between such resources, and continue to use the multi-credential GridFTP portlet for those resources that do not accept the gateway's credentials. Access to the gateway's MyProxy server could remain limited to the gateway host, since the other resources would simply need to trust the credentials it served, and would not need to access it directly.

Currently data movement on the gateway is manual. The researcher must login to the gateway to initiate data transfers. However, the Gateway could be enhanced to provide an interface that utilizes a data movement service. Here the researcher could configure transfers to happen at regularly scheduled intervals, or construct a pipeline where his simulation would send a notification to the gateway when the next time step is complete, triggering the data transfer.

The OpenSocial framework [11] enables Web 2.0 applications to be easily embedded into social networking sites such as iGoogle and MySpace. OAuth [12] security handling enables users to give such applications access to their credentials without sharing their password with the application. An OAuth interface to MyProxy allows users to talk directly to a MyProxy server, and delegate his proxy credential to an OpenSocial gadget, without sharing his proxy passphrase with the gadget. Since the credentials for accessing dynamic accounts on advanced resources are stored in MyProxy, OpenSocial gadgets could then potentially be configured to take advantage of those resources. In order to enable this however, access to the gateway's MyProxy server would need to be allowed from hosts other than just the gateway host.

6. CONCLUSIONS

Science Gateways have become an important vehicle for enabling communities of users to leverage the power of advanced computing resources. We have illustrated the case where data segregation among community users is required on these advanced resources. The Dynamic Account service has been identified as a mechanism for providing this capability. Features of the Dynamic Account service have been highlighted, and details of how it was integrated into the TeraGrid Visualization Gateway have been described. Problem areas that were encountered along the way were discussed, as well as areas for future investigation.

7. ACKNOWLEDGMENTS

The authors wish to thank the Dynamic Account service developers from the Globus project for their invaluable help with incorporating the service into the gateway, and for providing requested enhancements to the service. This work was supported in part by the National Science Foundation under Grants OCI-0504086 and OCI-0503697, and in part by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

8. REFERENCES

[1] Catlett, C. et al. "TeraGrid: Analysis of Organization, System Architecture, and Middleware Enabling New Types of Applications," HPC and Grids in Action, Ed. Lucio Grandinetti, IOS Press 'Advances in Parallel Computing' series, Amsterdam, 2007.

[2] K. Keahey, M. Ripeanu, and K. Doering, "Dynamic Creation and Management of Runtime Environments in the Grid," in *Workshop on Designing and Building Web Services (GGF 9)*. Chicago, IL, 2003

[3] I. Foster and C. Kesselman, "Globus: A metacomputing infrastructure toolkit," International Journal of Supercomputer Applications, 11(2):115-128, 1997.

[4] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, "A Resource Management Architecture for Metacomputing Systems". *Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing*, pg. 62-82, 1998.

[5] GridFTP Data Management, www.globus.org/toolkit/data/gridftp.

[6] M. Dahan, J. A. Insley, M. E. Papka, T. Uram, and K. P. Gaither, "Enabling Science through the TeraGrid Visualization Gateway" TeraGrid '07 Conference, Madison, WI, June 14 - 18, 2007.

[7] MyProxy, grid.ncsa.uiuc.edu/myproxy.

[8] J. T. Kohl, B. C. Neuman, and T. Y. T'so, "The Evolution of the Kerberos Authentication System," in Distributed Open Systems: IEEE Computer Society Press, 1994, pp. 78-94

[9] M. Pierce, J. Alameda, M. Christie, G. Fox, J. Futrelle, D. Gannon, M. Hategan, G. v. Laszewski, M. A. Nacar, E. Roberts, C. Severance, and M. Thomas, "The Open Grid Computing Environments Collaboration: Portlets and Services for Science Gateways," *Spec. Edtn, Concurrency & Computation: Practice & Experience*, 2005.

[10] InCommon Federation, www.incomonfederation.org.

[11] OpenSocial, www.opensocial.org.

[12] OAuth, www.oauth.net.