

# CyberInfrastructure for the Analysis of Ecological Acoustic Sensor Data: A Use Case Study in Grid Deployment

Randy Butler, Mark Servilla, Stuart Gage, Jim Basney, Von Welch, Bill Baker, Terry Fleury, Patrick Duda, David Gehrig, Michael Bletzinger, Jing Tao, D. Michael Freemon

**Abstract** – The LTER Grid Pilot Study was conducted by the National Center for Supercomputing Applications, the University of New Mexico, and Michigan State University, to design and build a prototype grid for the ecological community. The featured grid application, the Biophony Grid Portal, manages acoustic data from field sensors and allows researchers to conduct real-time digital signal processing analysis on high-performance systems via a web-based portal. Important characteristics addressed during the study include the management, access, and analysis of a large set of field collected acoustic observations from microphone sensors, single signon, and data provenance. During the development phase of this project, new features were added to standard grid middleware software and have already been successfully leveraged by other, unrelated grid projects. This paper provides an overview of the Biophony Grid Portal application and requirements, discusses considerations regarding grid architecture and design, details the technical implementation, and summarizes key experiences and lessons learned that are generally applicable to all developers and administrators in a grid environment.

**Index Terms** – acoustic signal analysis, case study, grid, security

## I. INTRODUCTION

In 2005, the National Center for Supercomputing Applications (NCSA), the Long Term Ecological Research (LTER) Network Office (LNO) at the University of New Mexico, the Science Environment for Ecological Knowledge (SEEK) project, and a researcher from the Kellogg Biological Station (KBS) LTER site at Michigan State University (MSU) performed the LTER Grid Pilot Study. The Pilot Study spanned 6 months and concluded by demonstrating a functional application, the “Biophony Grid Portal”, to the LTER Coordinating Committee meeting in Cape Charles, Virginia in September of that year. Primary funding was provided by the National Science Foundation, under its NSF Middleware Initiative program.

The goals for this study included: (1) demonstration of the vision of the LTER Grid concept to the LTER ecological research community, (2) demonstration of how grid and middleware technologies might be applied to enhance collaboration and sharing within and external to the LTER community, and (3) collection of requirements for a detailed design of an envisioned fully operational LTER Grid to support the roughly 2000 ecological researchers in the NSF Long Term Ecological Research program. The primary vehicle for accomplishing these goals was to develop and deploy a proof-of-concept prototype for the envisioned LTER Grid with the Biophony Grid Portal as a driving application.

During the course of development, it became clear that some of the new features and enhancements resulting from the Pilot Study would be generally applicable and useful to other grid environments. Our modifications to MyProxy [6] enabled callouts to the Pluggable Authentication Modules (PAM) for user authentication, which allows large communities to deploy grid infrastructure in a less intrusive manner. In this role, MyProxy functions as a bridge between the Grid Security Infrastructure (GSI), a X.509 certificate-based model, and existing site-specific authentication infrastructure. This work was quickly picked up by other communities, including TeraGrid, NERSC, FusionGrid, and NCSA. We also implemented support for HTTPS POST in Tomcat that uses GSI credentials for authentication. These modifications have

---

Manuscript received February 8, 2006. This work was collaboratively supported by the National Science Foundation Middleware Initiative GRIDS Center (program 4089, award numbers 0123961, 0123973, 0330685, 0330670, and 0330634), the LTER Network Office (DEB-0236154), and the National Center for Supercomputing Applications (CISE-0330670). Metadata management was performed using Metacat and the Ecological Metadata Language, which are products of a grant from NSF KDI (DEB-9980154).

Randy Butler, Jim Basney, Von Welch, Bill Baker, Terry Fleury, Patrick Duda, David Gehrig, Michael Bletzinger, D. Michael Freemon are with the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, IL USA (e-mail: {rbutler | jbasney | vwelch | bbaker | tfleury | pduda | dgehrig | mbletzin | mfreemon}@ncsa.uiuc.edu).

Mark Servilla is with the Long Term Ecological Research (LTER) Network Office, University of New Mexico, Albuquerque, NM USA (e-mail: servilla@LTERnet.edu).

Stuart Gage is with the Kellogg Biological Station (KBS) LTER, Michigan State University, East Lansing, MI USA (e-mail: sgage@msu.edu).

Jing Tao is with the National Center for Ecological Analysis and Synthesis, University of California, Santa Barbara, CA USA (e-mail: tao@nceas.ucsb.edu).

been contributed back and incorporated into the Globus Toolkit™ [8], and are generally available to all users of the jglobus library [9]. Metacat [10] is a flexible XML-based metadata database developed by the Knowledge Network for Biocomplexity (KNB) and is used by the LTER community as their metadata repository. As part of the pilot study, a Globus Toolkit version 4 (GT4) web service was developed to wrap the legacy interfaces to the Metacat server.

In this paper, we discuss the architecture and design elements of the Biophony Grid Portal and its services, our software enhancements and contributions to open source grid middleware, and other lessons learned that may be of interest to other grid projects.

## II. BACKGROUND

### A. The LTER Program

The Long Term Ecological Research (LTER) program [1] was established in 1980 to support site-specific ecological research. It currently consists of 26 research sites located in North America, the Arctic, Antarctica, Puerto Rico, and Tahiti, and the LTER Network Office at the University of New Mexico in Albuquerque. The core areas of study include primary plant production, organism population studies, movement of organic matter, movement of inorganic matter, and disturbance patterns.



Fig. 1. LTER Sites

Each of the 26 sites has a local data center, independently managed by a local governing agency. These sites have different hardware and software environments with heterogeneous data models. When the LTER program was

first established, this organizational model worked well since the research was usually local to each of the sites. Over time, and with recent advances in computing and sensor technologies, researchers have recognized the potential opportunity to broaden their research to encompass a collection of LTER sites, including collaborative cross-site experiments involving the access and analysis of data collected across multiple LTER sites. At the same time, deployments of large scale sensor networks require more sophisticated, scalable management and analysis of large amounts of data. These emerging needs have been reflected in the LTER Network Information System Strategic Plan [2].

### B. Acoustic Sensor Networks

In order to ensure the developed prototype was relevant, an acoustical ecology science experiment was selected as the application driver for the study.

Monitoring landscapes using a microphone to record diurnal and seasonal differences in their acoustic signatures has been a developing research activity of Dr. Stuart Gage over the past five years [4]. As a co-principal investigator at the Kellogg Biological Station LTER site (Fig. 2 and 3), one of Gage's goals is to seamlessly integrate recorded observations from multiple landscapes over time into a single system that collects, stores, and analyzes time series of digital recordings in an acoustic library. The current configuration of this acoustic network consists of stations that automatically record a 30-second field recording every 30 minutes. After recording and storing the data locally, data are automatically transmitted to a remote server where the recordings are stored in an acoustic library as .wav files [5]. Acoustic signals are subsequently analyzed and the results and sounds are made accessible via [www](http://www.cevl.msu.edu) ([www.cevl.msu.edu](http://www.cevl.msu.edu)). As an order of magnitude estimate, such an acoustical sensor network can generate 4.2 million files occupying 5.5 terabytes of storage annually if running sensors at 240 sites, underscoring the need for a scalable solution for data transmission and storage. Wireless technologies make the deployment of such networks possible [3].



Fig. 2. Field equipment at the Kellogg Biological Station



Fig. 3. A field sensor at the Kellogg Biological Station

### III. APPROACH

#### A. Pilot Study Requirements

To support the mission of the LTER Network Office, and the general goals of (1) demonstrating the grid vision to the LTER community, (2) demonstrating the use of grid and middleware technologies, as applied to the LTER community, and (3) collecting requirements for a detailed design of a fully operational LTER Grid, several specific requirements were addressed in the prototype. They included:

- Simple user interface. The user should not need to download or install any client-side software.
- Secure single signon. Users should only need to enter their password once, no matter which, or how many, grid resources are accessed during their session.
- Security. Authorization, authentication, integrity must be insured.
- Data provenance. Data owners need to maintain intellectual rights and enforce the LTER Data Access policy.
- Access to High Performance Computing. Seamless access to HPC resources is essential for high-volume data analysis.
- Scalable. Large amounts of data collected from sensor networks need to be managed and analyzed.

Another important consideration was interoperability with other grids. By using standard middleware components, interoperability could be realized between internal resources of the LTER Network Information System and external resources located within other networks, such as the Science Environment for Ecological Knowledge's EcoGrid [12] or the National Ecological Observatory Network (NEON) [13].

#### B. Biophony Portal Implementation

The LTER Grid, and the Biophony Grid Portal application that runs on it, is a collection of geographically distributed servers and services spanning multiple governing agencies (Fig. 5).

The user interface is a web browser that accesses a standard J2EE web application. The web application was developed using the Jakarta Struts framework and uses the Globus Java CoG Kit library to access grid resources. The web app runs in a standard Apache Tomcat container running on a Linux server at the LTER Network Office in Albuquerque.

To access the Biophony Grid Portal, a researcher must first authenticate to the system. The system retrieves users' X.509 credentials from a MyProxy [6] server that performs the authentication using LTER's existing LDAP directory server. The application then connects to the Metacat server over an HTTPS connection that is authenticated using the X.509 credential.

Once authenticated, a researcher can select an acoustic signature from a dataset of known sounds (e.g., the call of a chipping sparrow, *Spizella passerina*) and determine whether the species is present in a selected time series of field recordings contained in the digital library.

The system computes the probability of match between the signature and each unknown sound in the selected field recordings. There is also an option to generate a sonogram, sound profile, power spectrum, and frequency histogram of specific acoustic samples (Fig. 4). All security, data access, and other events are captured in a relational database for use in audit and data provenance reporting.

The digital database of field collected acoustic signals collected at 30 minute intervals during May, 2004, a set of acoustic signatures of organisms (birds, frogs) and entities (trucks, trains), and a set of signal processing algorithms were provided by Dr. Gage to prototype the portal.

#### C. Grid Services

The Biophony Portal described in the previous section utilizes a number of Grid Services.

Metacat [10] is a flexible XML-based metadata catalog developed by the Knowledge Network for Biocomplexity (KNB) and used by the LTER community as their metadata repository. It stores the metadata in the Ecological Metadata Language (EML), a requirement of all LTER data documentation. All acoustic data, including both field recordings and signature files, are documented in EML and stored in the Metacat. Metacat queries are executed by the Biophony application to search for and retrieve metadata for data files of interest. Metacat is deployed as a separate J2EE web application that uses the PostgreSQL relational database to store XML documents. As part of the Pilot Study, GT4 web service interfaces were implemented for the MetaCat server. However, due to project scheduling constraints, those interfaces could not be used by the Biophony application during the Pilot Study.

The portal application uses standard Globus Toolkit GRAM and GridFTP protocols to submit jobs and transfer files across the grid. The LTER Grid Point-of-Presence (POP) presents a standard site interface to simplify integration of heterogeneous site equipment with the LTER Grid. The POP concept comes from NEESpop, a component of the NEESgrid project [14]. In



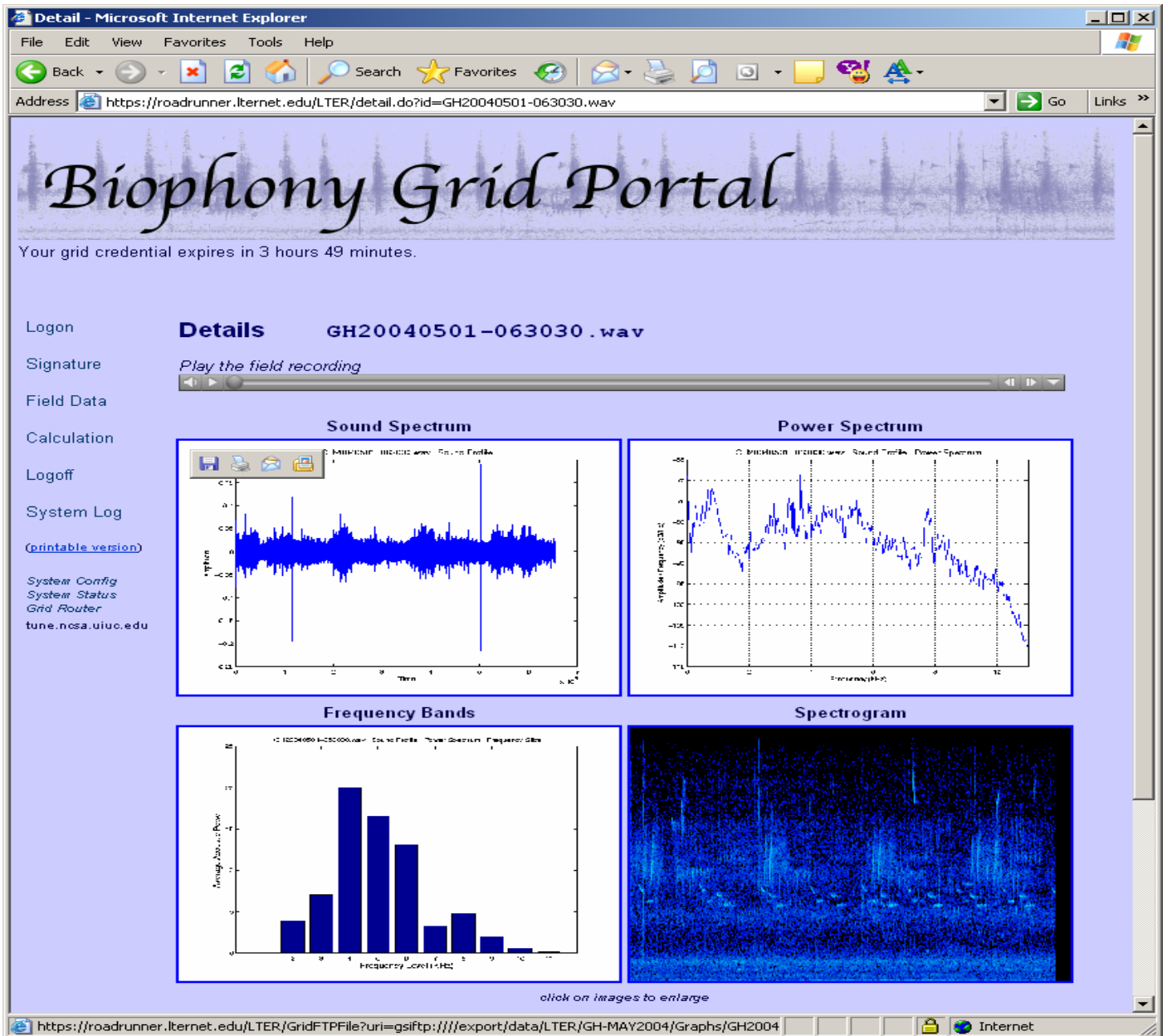


Fig. 4. Results of analyzing an acoustic data file.

our case, the POP serves as an Adapter [11] between the LTER grid and the local protocols and data formats. The main data repository is a GridFTP server at Michigan State University in East Lansing, Michigan.

The data analysis is performed on the Tungsten HPC cluster at NCSA in Champaign, Illinois. MATLAB code developed by Gage is used to perform the digital signal processing and species signature pattern matching.

Major software components of this system report key events to a centralized MySQL database to facilitate the data provenance needs of the LTER community.

#### IV. TECHNICAL CHALLENGES/ADVANCEMENTS

##### A. MyProxy PAM Integration

The LTER community has a well established LDAP-based security infrastructure that manages the user registry for all of the LTER Network. Our challenge was to layer standard grid security on top of the existing user registry, so that we could leverage existing grid and middleware tools and services and, at the same time, avoid creating an additional authentication system.

Rather than implementing LDAP support directly in the MyProxy software, we selected a more flexible solution. Pluggable Authentication Modules (PAM) [7] abstract the authentication scheme away from the application software,

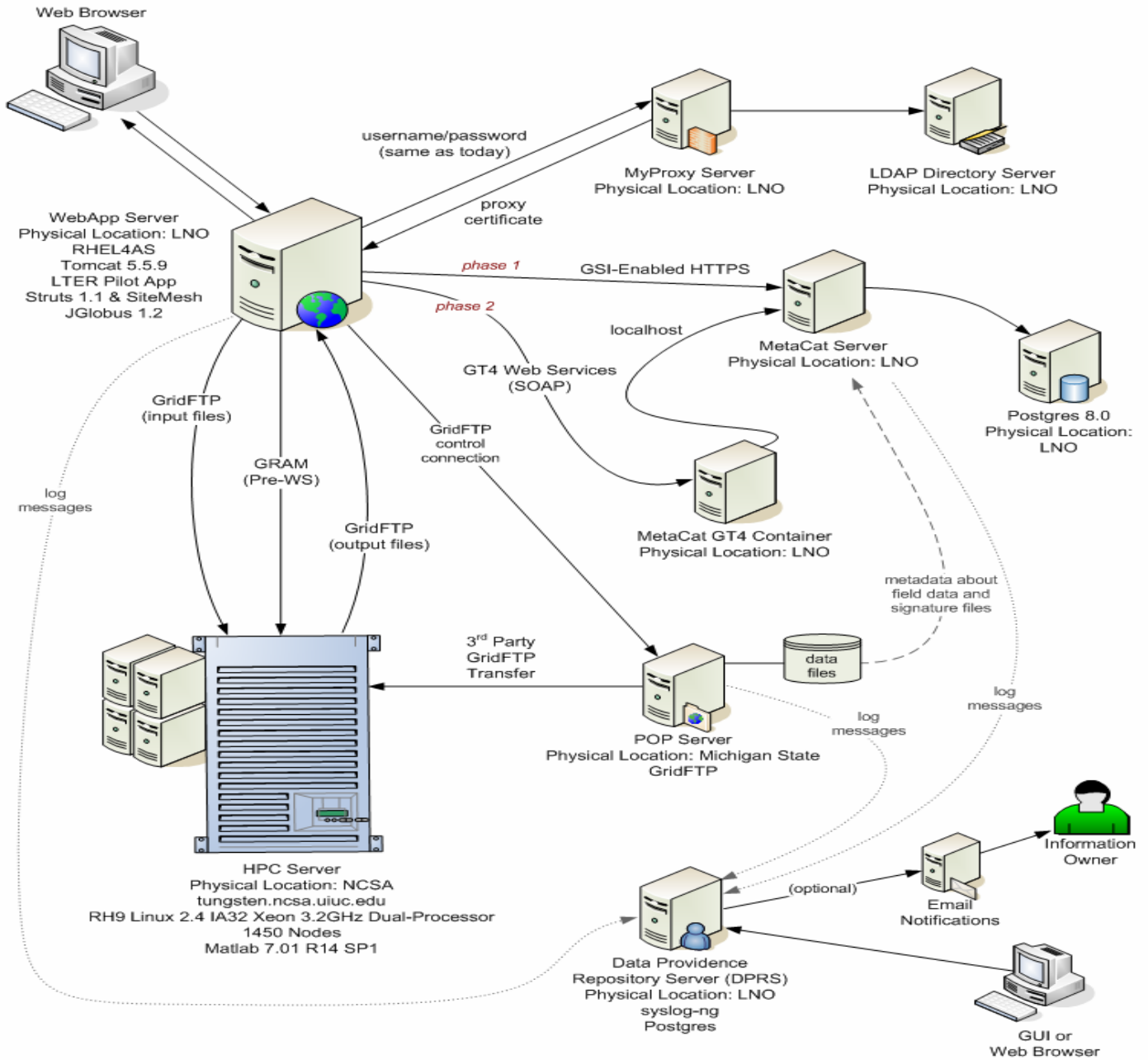


Fig. 5. Deployment and service view of the LTER Grid.

allowing the application and authentication method to vary independently. We added PAM support to the MyProxy Credential Repository software, then configured PAM to authenticate the user with LTER's remote LDAP directory server.

These MyProxy enhancements have been committed to the MyProxy CVS repository and are available to the public. As a result, MyProxy administrators running version 2.0 or higher can choose from a variety of authentication mechanisms without requiring any changes to the MyProxy source code.

Several communities have already leveraged this work. TeraGrid is using Kerberos to authenticate their MyProxy users. NERSC and FusionGrid will be using this feature for

interoperability with Department of Energy (DOE) one-time password (OTP) deployments. The MyProxy deployment at NCSA supports both Kerberos and OTP authentication.

#### B. MyProxy Online CA

Being able to use LTER's LDAP user registry as the MyProxy user authentication mechanism was vital in extending the existing security infrastructure into the grid space. However, users were still required to obtain accounts on grid resource provider servers, request and manage their user certificate, and place proxy certificates into the MyProxy server.

During the Pilot Study, we found that most errors and support requests were related to the manual processes surrounding credential management. Elimination of these burdens from the user is important for the real-world usability of the system. The MyProxy software provides a solution. Starting with version 3.0, MyProxy provides the capability of operating as an online certificate authority. In this role, MyProxy can dynamically generate X.509 credentials for authenticated users, thus avoiding all the issues surrounding user management of grid credentials.

This MyProxy capability was released just after the feature freeze date for the Biophony Portal application and so was not included as part of the Pilot Study.

### C. GSI-Enabled HTTPS

Metacat, a metadata catalog service developed by the Knowledge Network for Biocomplexity, authenticates and authorizes users via LTER's LDAP directory. One of the goals for the Pilot Study was to integrate Metacat with the Grid Security Infrastructure (GSI). This was accomplished in two ways: using a GT4 web service façade, as discussed below, and with a GSI-authenticated HTTPS connection.

We replaced the standard HTTP interface to Metacat with HTTPS, using GSI credentials to establish the SSL connection, for a fully authenticated and encrypted channel.

As part of this effort, it was necessary to implement support for HTTPS POST in the jglobus library so that Tomcat could use GSI credentials for authentication. Those modifications have been contributed back and incorporated into the jglobus codebase and are available to users of the jglobus library.

The Metacat client API was augmented to accept a GSI credential as an alternative to an LDAP userid/password pair. Within the Metacat server, a mapping facility was added between credential Distinguished Names and LDAP identities, so that existing security policies could remain in effect. As with the Globus Toolkit enhancements, these patches were submitted to the Knowledge Network for Biocomplexity (KNB) committers.

### D. GT4 Web Services to Metacat

Another approach for seamlessly integrating Metacat into the grid environment was to expose its functionality as GT4 web services by building simple wrappers around the legacy interfaces. This allows any SOAP client to invoke Metacat queries using standard web service protocols.

Fundamentally, this is nothing more than a Globus Toolkit 4 Java WS Core “wrapper” placed around the GSI-enabled Metacat client API.

The WSMetacatService GT4 web service relies on the communication between the client and server to be performed over a GT4 GSI-secure HTTPS connection. This is accomplished using an X.509 credential when connecting from a web service client to the WSMetacatService. When deployed, the web service is wrapped as in Fig. 6.

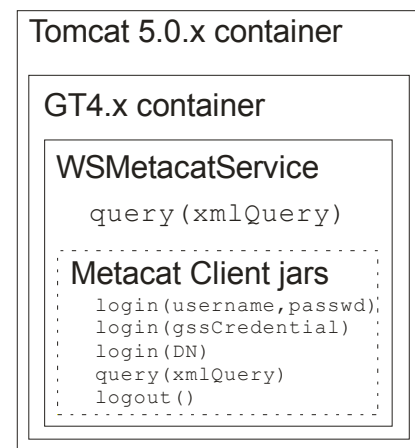


Fig. 6. Metacat GT4 Web Service

The Globus container is deployed to a modified Tomcat 5.0.x installation (to allow GSI-HTTPS connections), resulting in a web service implementation of Metacat.

A web service client gains access to the previously retrieved X.509 proxy certificate, gets an endpoint reference and porttype for the WSMetacatService, and sets the GSI\_CREDENTIALS property on the client stub. Then, by connecting to the Tomcat server on the GSI-HTTPS port, this credential gets passed to the WSMetacatService.

How *much* of the credential gets passed depends on the type of connection made. For example, the `login(gssCredential)` method requires a full credential, thus the proxy certificate needs to be delegated from the client to the server. To do this easily, GT4 has an authentication method called `GSISecureConversation`. This provides a mechanism for generating a security session, i.e., the negotiation of a shared secret which may be used to secure a set of subsequent messages. It is based on WS-Trust and WS-SecureConversation. It creates a delegation service collocated with the web service such that the `gssCredentials` specified on the client side can be accessed on the server side. Alternatively, the `login(distinguishedName)` method relies on the assumption that the WSMetacatService and the Metacat server are collocated, so that no authentication mechanism is needed. Here, all we need to do is set the transport mechanism to use encryption, still setting the GSI\_CREDENTIALS on the client stub. Since we are not doing full delegation of credentials, only the distinguished name (DN) of the proxy certificate is available to the WSMetacatService.

On the server side, only the `query(xmlQuery)` method is exposed as a public method. This is because the `query()` method implicitly calls the Metacat `login()/logout()` methods. The `login()` method to be used must be decided on the server side at Tomcat startup because the security settings for a GT4 WS service are declared in a security descriptor and cannot be changed at runtime. This server-side setting then dictates which authentication method must be used by the WS client.

It should be noted that this web service was implemented and tested, but was not used in the Biophony Portal application due to time and schedule constraints.

This work was shared with the Knowledge Network for Biocomplexity (KNB) team for possible incorporation into the Metacat distribution.

#### E. XINETD 30-second delay

During development of this grid application, we noticed an interesting phenomenon. Each time our grid client application opened a GRAM or GridFTP socket connection to one of our grid servers (a specific server), a delay of 30 seconds was incurred. This was observed only when the client was running on the Windows operating system.

After some additional investigation, the cause was isolated to the Globus Toolkit services that were configured as xinetd services (i.e., globus-gatekeeper and gsiftp). The configuration of xinetd specified `USERID` for the `log_on_success` and `log_on_failure` directives, which caused an ident request (port 113) to be issued from the server back to the client machine.

Unless the client machine actively rejects (or actually responds to) these requests, xinetd will wait until it times out after 30 seconds. Windows passively drops the packets, which results in the delay. Other operating systems actively reject such packets as their default configuration.

Note that the delay was not experienced when the Windows client was behind a NAT because the NAT server actively rejected the ident packets.

The `USERID` attribute in xinetd does not add any value to this service, so it was removed from the xinetd configurations without detrimental effects.

#### F. Application Performance

The Biophony Grid Portal application initially experienced response times in excess of what would normally be expected. A performance analysis was conducted to isolate the cause of this poor performance.

Several actions were taken to isolate this problem: (1) Debugging messages were enabled in the GridFTP server. (2) The code was reviewed to determine the “interface points” to the jglobus library. (3) The code was augmented to capture specific data that was written to log files. (4) The code was executed in a controlled environment. (5) Finally, the resulting log files were analyzed to determine the performance decrease.

Each time digital signal processing calculations are performed on data files, a number of GSI-authenticated network connections are created and destroyed. This number ( $N$ ) depends on how many files ( $M$ ) are being requested from the data server, and can be described using the following formula:

$$N = 20 + 4 * (M - 1)$$

This formula was determined from the patterns that emerged by correlating the network flows with application events reported from the instrumented code. To improve performance

the value of  $N$  must be reduced. The following paragraphs discuss our approach.

*Eliminate Unnecessary Remote Calls.* We observed that the application was issuing multiple “probe remote home” calls during each user session. This is a remote GRAM call to determine the full path name of the user’s home directory on the remote grid resource server. Since all file references can be specified as relative (to the user’s home directory), this is unnecessary information. These remote calls were removed from the application.

*Cache Derivable Data.* We also found it valuable to cache specific types of files on the portal web server. Signature files were cached since they are not updated, few in number, and read often. The generated graphs which require no input parameters were cached. We also performed “read-ahead buffering” of analysis output graphs, in anticipation of the user’s request for them.

*Avoid “Hidden” Remote Calls.* The Java CoG Kit library (jglobus) can make remote calls that are not necessary. For example, the `GridFTPClient` class will issue remote calls on method invocations, such as `exists()`, which incur overhead from TCP or GSI services. If the application checks for the existence of a file before transferring it, the performance can degrade quickly. This accounts for more than half of the connections defined in the above formula.

*Combine Small Files for Transfer.* Part of the workflow processing of the Biophony Portal application involved transferring a large number of relatively small files from the main data repository to the HPC cluster for data analysis. The overhead of creating numerous network connections to transfer each individual file also resulted in degraded performance. For example, transferring 69 separate files (approximately 90 MB total) using a 3<sup>rd</sup> party transfer required five minutes. By aggregating the files into one data package, and then performing a single GridFTP transfer to move the package to the HPC system, the time was reduced to 45 seconds.

## V. POTENTIAL FUTURE WORK

### A. Community Accounts

We did not implement “community accounts” for the Biophony Grid Portal since this concept is still evolving, and is an active area of research. There is a clear need for this capability, however, as security-related issues accounted for most of the end-user support calls, questions, and problems. Community accounts can potentially eliminate many of these problems.

### B. Interactive HPC Allocations

Another issue not addressed was the scheduling of HPC resources for on-demand (or interactive) processing. Traditionally, such resources have been dedicated to batch processing and, to a large extent, remains so today.

## VI. ADDITIONAL OBSERVATIONS

“All the old rules still apply, even in a grid environment.”

*Object-Oriented Analysis and Design.* The original code base contained many layers of abstraction that were refactored out, resulting in a cleaner and more straightforward design. It is important to always keep the architecture, design, and coding of an application as simple as possible, within the constraint of meeting the requirements.

*Distributed Systems.* Only distribute application components across data centers or platforms when necessary; doing so always adds overhead.

*Unit Testing.* Unit testing is incredibly powerful for regression testing and diagnosing problems, and can quickly pinpoint failed components. During development, simple JUnit tests were built and executed nightly, which found many broken code segments.

*Iterative Development.* Use of an iterative software development process is important to receive rapid feedback from users. This is especially helpful during prototype development in research projects.

*Psychology.* Human psychology is an important aspect of designing the user interface. Also, keep users engaged during workflow sessions by allowing them to be “interactive” while long-running processes execute in the background.

*Intergroup Communication.* The compressed timeline of this project demanded highly interactive communication between the various groups involved. Regular phone and video conferences were essential to keep the project on target.

## VII. RELATED WORK

### A. Kx509/KCA

With the extensions to MyProxy to enable PAM authentication and act as an one certificate authority, there is overlap in functionality between MyProxy and the Kerberos Certificate Authority (KCA) [15]. MyProxy at this point is a generalized KCA, allowing authentication with a range of authentication methods, selectable at deployment time. Alternately, we could have generalized KCA to achieve the same functionality and our decision to extend MyProxy instead was based on greater MyProxy expertise within the project coupled with a desire to reduce risks given the short project lifetime.

### B. TrustManager

Our work in enabling Proxy Certificate support in HTTPS for Tomcat is certainly not a novel concept. TrustManager [16] is a previous example of an implementation of such support for Tomcat. Our work however is not a reimplement of Globus functionality, but instead we determined how to use select Globus libraries in Tomcat to achieve this functionality and have subsequently documented this method [17]. A potential advantage of this method is that it will be easier to maintain compatibility with the Globus

Toolkit in the future as its implementation evolves since it leverages the Globus Toolkit libraries directly. (We note that TrustManager appears to be out of date and doesn’t support the new RFC 3820 [18] compliant proxy certificates use by current version of the Globus Toolkit.)

### C. OPAL, Generic Service Toolkit

Since our work with wrapping Metacat, several software packages to automate this process have emerged, including Opal [19] and the Generic Service Toolkit [20]. If we were to do this work today, we would investigate leveraging these packages.

## VIII. CONCLUSION

The success of the LTER Grid Pilot Study fully demonstrates the efficacy of grid middleware in support of LTER science to the LTER research community as exemplified by the Biophony Grid Portal application. It also identified (and extended) middleware components that could meet the requirements of a production version of an LTER Grid. Such insight would have been difficult to grasp without a hands-on evaluation of the technology. As developed through this pilot study, a vision for a grid computing infrastructure is an important component of the LTER Network Information System that is necessary to meet the goal of Network-level synthetic science.

Scalability of solutions has emerged as an increasingly significant issue, and grid technologies are an important approach to addressing and solving large-scale data and analytical requirements.

## ACKNOWLEDGMENT

We gratefully acknowledge the contributions of Greg Bauer, James Brunt, Duane Costa, Matt Jones, Bill Punch, Greg Shore, and Raheem Syed.

## REFERENCES

- [1] Long Term Ecological Research Network. <http://www.lternet.edu>
- [2] LTER Network Information System Strategic Plan. <http://intranet.lternet.edu/planning>
- [3] Porter, J., P. Arzberger, T. Kratz, P. Hanson, S. Gage, W. Michener, F-P. Lin, H. King, Hans-Werner Braun, T. Hanson, S. Shapiro, P. Bryant. 2005. Wireless Sensor Networks for Ecology. *BioScience* 55: 561-572.
- [4] Gage, S.H., B. Napoletano and M. Cooper. 2001. Assessment of ecosystem biodiversity by acoustic diversity indices. *The Jour. Acoustical Soc. of Amer.* 109(5). p. 2430.
- [5] Gage, S.H. and B. Napoletano. 2004. *Envirosomics equipment and operations manual*. Computational Ecology and Visualization Laboratory. Michigan State University. 30 pp.
- [6] 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.
- [7] Linux-PAM. <http://www.kernel.org/pub/linux/libs/pam>
- [8] B. Sotomayor and L. Childers, *Globus Toolkit 4: Programming Java Services*, Morgan Kaufmann, December 16, 2005.
- [9] G. von Laszewski, I. Foster, J. Gawor, and P. Lane. A Java Commodity Grid Kit. *Concurrency and Computation: Practice and Experience*, 13(89):643- 662, 2001.



- [10] KNB Software: Metacat.  
<http://knb.ecoinformatics.org/software/metacat>
- [11] E. Gamma, R. Helm, R. Johnson, J. Vlissides. 1995. *Design Patterns Elements of Reusable Object-Oriented Software*. Addison-Wesley. 139-150.
- [12] SEEK EcoGrid.  
<http://seek.ecoinformatics.org/Wiki.jsp?page=EcoGrid>
- [13] National Ecological Observatory Network. <http://www.neoninc.org>
- [14] S. Gullapalli, S. Dyke, P. Hubbard, D. Marcusiu, L. Pearlman, C. Severance, "Showcasing the Features and Capabilities of NEESgrid: A Grid Based System for the Earthquake Engineering Domain," 13th IEEE International Symposium on High Performance Distributed Computing (HPDC-13 '04), 2004, pp. 268-269.
- [15] Kornievskaja, O., Honeyman, P., Doster, B., and Coffman, K., Kerberized Credential Translation: A Solution to Web Access Control. 10th Usenix Security Symposium, 2001.
- [16] TrustManager, <http://edg-wp2.web.cern.ch/edg-wp2/security/trust-manager.htm>
- [17] Fluery, T. Using X.509 Proxy Certificates With Tomcat And Windows XP,  
<http://security.ncsa.uiuc.edu/research/wssec/gsihttps/gsitomcat.php>
- [18] Tuecke, S., Welch, V. Engert, D., Pearlman, L., and Thompson, M., Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, RFC 3820, IETF, June 2004.
- [19] Krishnan, S., Stearn, B., Bhatia, K. Baldrige, K., Li, W., and Arzberger, P. Opal: Simple Web Services Wrappers for Scientific Applications, SDSC TR-2006-5.  
<http://www.sdsc.edu/techreports/SDSC-TR-2006-5-opal.pdf>
- [20] Gopi Kandaswamy, Liang Fang, Yi Huang, Satoshi Shirasuna, Suresh Marru, and Dennis Gannon. Building Web Services for Scientific Grid Applications. IBM Journal of Research and Development, 50(2/3):249-260, 2006