

Building the Bioscience Gateway

Alan Blatecky, Kevin Gamiel, Lavanya Ramakrishnan, Daniel Reed, Mark Reed

{alan, kgamiel, lavanya, dan_reed, markreed} @renci.org

Renaissance Computing Institute (RENCI), University of North Carolina at Chapel Hill

1. Introduction

Bioinformatics, biology and biomedical research have advanced rapidly in the last few years, with increased involvement of interdisciplinary teams, large-scale computational models, distributed data archives and high-throughput instrumentation. The major biomedical breakthroughs can only be made possible through the use of high performance computing, grid computing and data management technologies and tools. For the biomedical research community to leverage next generation computing resources, we need robust software tools that provide interfaces that are easy to use as well as having highly integrated environments which simultaneously manage data resources, computer models and applications.

This paper describes the Bioscience Gateways we are building to cater to the needs of specific user groups while building an infrastructure useful to national biomedical users and research. The motivation for this infrastructure is based on three separately funded projects:

- *North Carolina Bioinformatics portal* [1] This project is funded by the state of North Carolina to build a bioinformatics portal that combines access to standard databases and computational analysis tools. The portal leverages the Grid infrastructure under development across the state and nation.
- *Evolutionary biology* [2] As part of the NSF funded National Evolutionary Synthesis Center (NESCent), which is a five year, \$15M joint Duke, UNC and NCSU effort, we are developing federated data models and portals for access to diverse data resources used to construct evolutionary biology trees.
- *Exploratory genetics* [3] The NIH funded Carolina Center for Exploratory Genetic Analysis (CEGA) is targeting data federation and correlation for understanding the interplay of multiple genes in disease. Understanding the function of specific sequence variations will lead to the identification of

genes and pathways that play a critical role in human disease.

The remainder of this paper is organized as follows. Section 2 describes the general notion of building communities with local gateways and shared resources. Section 3 describes the portal, grid and bioinformatics technologies that our implementation is based on. Section 4 discusses practical issues and specific policies that are being addressed in the context of the North Carolina Bioinformatics gateway. Section

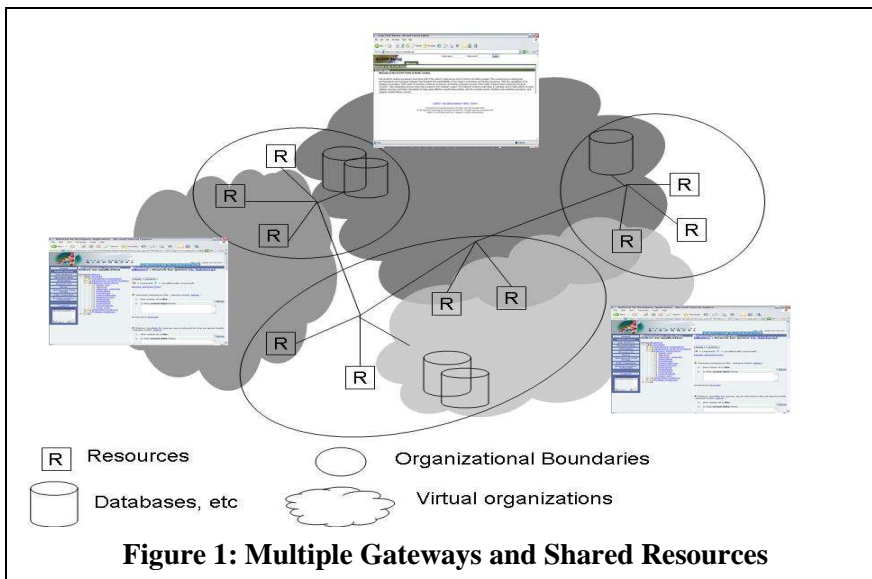


Figure 1: Multiple Gateways and Shared Resources

5 is the conclusion.

2. Building Communities

The Bioportal is based on the grid architecture as initially described by Foster, Kesselman and Tuecke [4]. Each of the individual communities is likely to have some dedicated infrastructure (e.g. cluster, data collection) and runs a local gateway catering to the needs of its users. The gateways provide transparent access to user-accessible resources, both as individuals and members of virtual organizations [5]. For example, the evolutionary biology community may have access to TeraGrid resources at certain times in addition to their own resources. The North Carolina Bioportal community may have access to statewide resources and possibly to some TeraGrid resources to meet peak needs. This grid architecture allows the community to grow and add new partners to their virtual organizations as well as benefit from additional resources as they become available.

```
<parameter ismandatory="1" iscommand="1" issimple="1" type="Excl">
  ...
  <code>"blastall -p $value"</code>
  .... <vlist>
    <value>blastn</value> <label>blastn: nucleotide query /
nucleotide db</label>
    <value>blastp</value><label>blastp: amino acid query /
protein db</label>
    .. <value>psitblastn</value> <label>psitblastn: protein
query / transl. nucleotide db</label>
  </vlist>
</parameter>
```

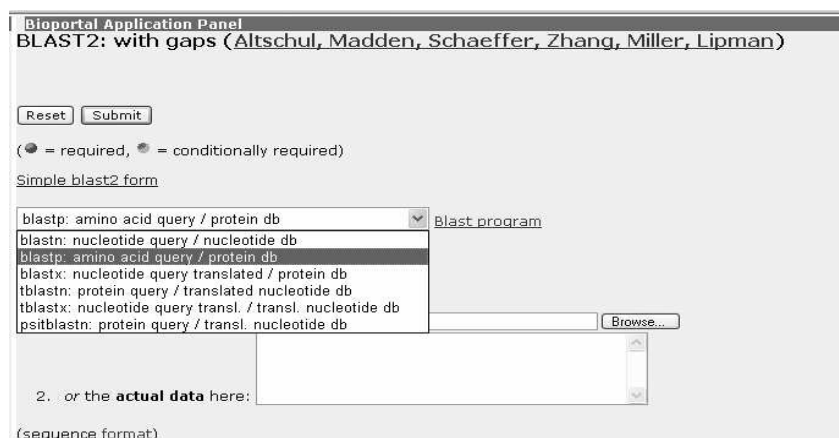


Figure 2: Sample of Pise XML for BLAST. Note how the <vlist/> is used to generate the drop down menu. The <code/> section describes how this parameter is used in the command line (RSL) generation.

Using separate gateways allows service providers to accommodate individual community needs while reusing the backend infrastructure. The replication of the gateways helps provide load balancing and assures QoS guarantees made to individual user communities.

3. Bioportal Framework & Technologies

Our Bioportal framework is based on the Open Grid Computing Environment (OGCE) framework[10]. The portal implementation also integrates a suite of biology applications and adds other supporting portlets to enhance the user experience.

3.1. Portlets

The portal leverages the grid (provided by OGCE) and collaboration portlets (provided by CHEF). The Biosciences Gateway expands this framework to integrate biology applications and data. In the current deployment of the portal, each user has a “personal workspace” and a “shared workspace.” The

shared workspace is shared by all the users in the community and fosters collaborations through the chat, discussion, and group schedule portlets. The personal workspace, akin to a personal desktop, gives access to the OGCE portlets such as the latest news through RSS feeds, personal calendar, or provides the ability to store documents and portlets to submit and monitor biology jobs to the grid. Jobs are submitted via the Globus gatekeeper using the Java CoG interface to Globus. We have also built a job history portlet allowing users to view previous jobs and associated output files. The job history maintains information about the time, job id, job directory and other metrics in a backend MySQL database.

3.2. Grid Infrastructure

The current prototype of the Bioportal is deployed on a 34 node Linux cluster (1 head node, 32 compute nodes, 1 storage node) each with 3.06 GHz dual Xeon processors, 4 GB memory/node and connected by Gigabit Ethernet. The compute infrastructure is supported by 1.73 TB storage array. The cluster is managed using Rocks cluster management software that is based on Red Hat Linux.

The grid infrastructure is based on the Globus [6] middleware. We use Globus 3.2.1, primarily the pre-web services tools such as GridFTP and the Globus gatekeeper. The gatekeeper is used as the standard interface for job submission, and GridFTP is used for managing file transfer. The gatekeeper is connected to the Torque/Maui scheduler (OpenPBS) to manage scheduling on a cluster. The user certificates are stored in MyProxy [7], a credential repository used to manage grid credentials. When the user logs into the portal, the OGCE MyProxy portlet retrieves the credential, and the credential is then used in the context of the user for job submissions.

3.3. Bioinformatics Tools

The bioinformatics interfaces are based on PISE [8], an XML tool that generates web interfaces for bimolecular applications. PISE has XML descriptions for each application that describe the interface and the logic to process these applications (See Fig 2). PISE generates application specific HTML forms. The HTML forms are transformed and imported as velocity templates into the portal framework. The Perl-CGI logic in PISE has been reproduced in Java and integrated into the OGCE framework. The user input in the form is processed in conjunction with the XML for the application to generate the RSL for the job to be submitted to the gatekeeper.

The current deployment of the portal has about 140 bioinformatics applications including suites such as EMBOSS (European Molecular Biology Open Software Suite), GLIMMER (gene identification in microbial DNA), HMMER (Hidden Markov Model program for profile-based sequence analysis) NCBI (diverse set of tools), PHYLIP (PHYLogeny Inference Package for inferring phylogenies) and other applications such as ClustalW, FASTA, etc. (See complete list of currently supported applications at [11]) The backend databases (about 330 GB) to support the applications include NCBI, GenBank, PDB, GenPept, Prints, RepBase, UniProt, PFam, ProSite, TransFac.

4. Practical Issues

The initial prototype has been made available to users in the state of North Carolina as part of the North Carolina Bioportal project. The target audience includes students, researchers and educators.

4.1. User Accounts

One of the main challenges was with respect to managing the user accounts, credentials and associated passwords. Every user needs to have a portal account, a user certificate and an actual account on the physical machine. The passwords for the first two need to be the same to enable user transparency in acquiring the user certificate during logon. Thus, an account request procedure was a multi-step process with the user needing to ssh to the machine to request the grid certificate. The user account request procedure was a nightmare for both administrators and end-users. We built a simple account management system that allows the user to pick a password. A public-private key pair is generated for that user and an email is sent to the user asking him/her to confirm the request. After the request is confirmed, the administrator approves the account and creates a unix account for the user, loads the certificate in MyProxy and populates the OGCE databases with both user account information and a customized set of portlets that the user can access based on his/her rights. The account management system allows the user to change his/her password at any time, a capability that cannot be easily done today. PURSE[9] (released recently) solves part of the above problem and we are investigating its use. From our preliminary analysis it seems that there is still significant integration and customization effort

to make it work in our environment. In the future, the account management interface could be expanded to allow users to specify other MyProxy servers to use to access his/her credential. During user registration we use a very simple validation process. If the user has a valid email from an educational institution in the state, they are granted an account. Out-of-band methods such as emails and telephone are used to validate other users. A stronger validation mechanism will be needed in the longer run

4.2. Security

We use traditional SSL to encrypt user communications with the portal server and use Grid Security Infrastructure(GSI) for protecting access to grid resources. A large amount of biomedical data has very strict privacy requirements. Managing such data in the portal is still an open issue to be addressed. In addition, capabilities to enable auditing and accounting in the portal are important. We are developing a very simple log based system to collect the data

4.3. Job Management

We run the portal under a separate user account – “portal” to restrict damage that be can caused by malicious users or runaway processes. Most bioinformatics applications take an input file as a parameter. The file (uploaded through the user interface) needs to be available to the job during execution. This file is written to the portal server as user "portal." This needs to be in a shared directory that is accessible by the user's job. We also had to ensure that every job is run in a unique directory to avoid jobs overwriting themselves. Thus we create a unique job directory named as <applicationName>.<timestamp> for every job run. This is created by a unix call right now because the portal server and the compute nodes have a shared file system. Ideally, this should be controlled during job submission and could be a complex issue if the portal server does not have prior knowledge of what directories the user has access to in the remote systems.

In our current deployment users are allocated directories in the scratch space that is used to run the jobs. There are no limits on the file or job size. However, because the portal does not provide an easy interface to delete their previous runs, we have an automated script that deletes job files that are more than 14 days old. Users are responsible for saving files that they need by saving it to their home directory. This policy is subject to change based on usage in the coming months. Using scratch space for job runs allows heavy users to use the space not used by others. A semi-automated mechanism to save the results to a user's home directory will help accommodate large space and longer history requirements. But the ability to reflect when a user has exceeded their unix quota through the portal is necessary in the long-term to completely solve this problem.

4.4. Science Issues

There are a small number of challenges that are specific to the application domain. We had to work through issues resulting from conflicts in the RSL syntax and the application's command-line e.g: “=” is considered a special character by RSL and used by the application to pass parameters. This would result in the job submission to fail. We post-processed the command-line generated from the application logic (represented in PISE XML) to escape the characters appropriately.

The bioinformatics applications depend on large databases and these will need to be hosted on all the grid resources. These databases are updated periodically. Some of these updates are fairly large and may take a number of hours every single day. Sites may have different update policies based on disk space or bandwidth constraint. The variation in the output can be significant based on the frequency of the updates. Maintaining global knowledge of the data and using it to make resource decisions during scheduling is important when considering load balancing across clusters.

5. Conclusions and Future Work

The North Carolina Bioportal gateway is currently available to the user community in the state. The gateway provides the tools necessary for scientists to access distributed data and resources. Today's grid and portal technologies provide the different software pieces required to enable the scientist to take advantage of the grid. However, deep understanding of the technologies and a significant time investment is still required to enable the integration of the software, site customization and building interfaces for specific science communities. Also individual sites are largely responsible for ensuring scalability to a large number of users, accounting and auditing practices. This slows down the large scale adoptability of grid technologies especially in smaller institutions around the country. As part of the North Carolina Bioportal we are trying to reduce this gap by distributing the software stack and conducting hands-on workshops for users and for grid infrastructure administrators.

We are building a simple log based auditing and accounting system into the user interface to enable tracking user accounts and also to enable collecting data on popular applications, etc. We are also investigating simple load balancing techniques to enable choosing other shared resources. Data models and federation of distributed data sources will be made available through the portal.

6. Acknowledgements

The current prototype Bioportal was developed in part with seed funding from the University of North Carolina Office of the President for development of advanced research and education applications in high-performance computing, information systems, and computational and computer science. The initial prototype was a team effort and the authors would like to thank the entire NCBioportal team (consisting of the Renaissance Computing Institute, UNC-Chapel Hill's Information Technology Services and Center for Bioinformatics and Wake Tech Community College) including Timothy Chagnon for developing the account management module; Mike Seda and Michael Shoffner for development, testing and deployment of the initial prototype of the Bioportal.

7. References

1. North Carolina Bioportal (<http://www.ncbioportal.org>)
2. National Evolutionary Synthesis Center (NESCent) (<http://www.nescent.org>)
3. The Carolina Center for Exploratory Genetic Analysis (<http://www.renci.org/nih>)
4. I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3), 2001.
5. Dennis Gannon, Randall Bramley, Geoffrey Fox, Shava Smallen, Al Rossi, Rachana Ananthkrishnan, Felipe Bertrand, Ken Chiu, Matt Farrellee, Madhu Govindaraju, Sriram Krishnan, Lavanya Ramakrishnan, Yogesh Simmhan, Alek Slominski, Yu Ma, Caroline Olariu, and Nicolas Rey-Cenvaz. Programming the Grid: Distributed Software Components, P2P and Grid Web Services for Scientific Applications. In *Special Issue on Grid Computing, Journal of Cluster Computing*, volume 5(2002) No. 3, pages 325-336. Kluwer Academic Publishers, 2002.
6. I. Foster, C. Kesselman. Globus: A Metacomputing Infrastructure Toolkit. *Intl J. Supercomputer Applications*, 11(2):115-128, 1997.
7. J. Novotny, S. Tuecke, V. Welch. An Online Credential Repository for the Grid: MyProxy. *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*, IEEE Press, August 2001.
8. Pasteur Institute Software Environment (PISE) (<http://www.pasteur.fr/recherche/unites/sis/Pise/>)
9. Portal Based User Registration Service (<http://www-unix.grids-center.org/r6/ecosystem/security/purse.php>)
10. Open Grid Computing Environment (<http://www.collab-ogce.org/nmi/index.jsp>)
11. Supported applications in the Bioportal (<http://www.ncbioportal.org/SupportedApplications.txt>)