Development of Usable Grid Services for the Biomedical Community

Prof. Richard Sinnott
National e-Science Centre,
University of Glasgow,
Glasgow G12 8QQ
ros@dcs.gla.ac.uk

Abstract

The BRIDGES project was funded by the UK Department of Trade and Industry to directly address the needs of the cardiovascular research scientists investigating the genetic causes of hypertension as part of the Wellcome Trust funded (£4.34M) Cardiovascular Functional Genomics (CFG) project. Specifically, the BRIDGES project developed a compute Grid and a data Grid with security at its heart. This paper presents the experiences in developing usable Grid services for the bio-community and the different phases of prototypes that were refined based upon user requirements and feedback.

1. Introduction

The Biomedical Research Informatics Delivered by Grid Enabled Services (BRIDGES) project (www.nesc.ac.uk/hub/projects/bridges) was funded by the UK Department of Trade and Industry to develop a computational infrastructure to support the needs of the Wellcome Trust funded (£4.34M) Cardiovascular Functional Genomics (CFG) project (www.brc.dcs.gla.ac.uk/projects/cfg). The CFG consortium is investigating possible genetic causes of hypertension, one of the main causes of cardiovascular mortality. This consortium which involves five UK and one Dutch site (depicted in Figure 1) is pursuing a strategy combining studies on rodent models of disease (mouse and rat) contemporaneously with studies of patients and population DNA collections.
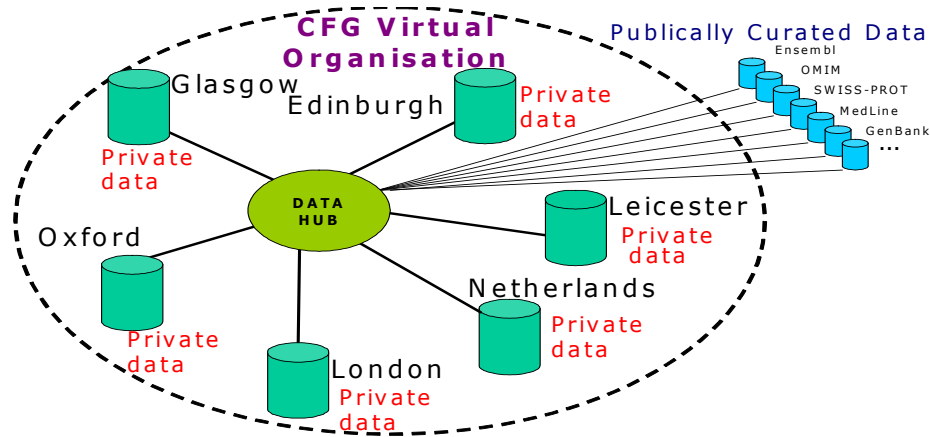

Figure 1: Data Distribution and Security of CFG Partners

The data hub provided a single repository (a DB2 database) through which access to numerous other federated genomic repositories could be made. Both OGSA-DAI and the commercial solution Information Integrator from IBM were used for this purpose and allowed to show how such remote databases could be seamlessly accessed from the users' perspective. The BRIDGES project developed various client side tools through which queries could be issued and used to access these remote databases. Given that the scientists based much of their research upon results from microarray experiments, these queries were typically based upon returning all information associated with a given gene (or set of genes).

2. Data Access Client Tools

The initial data access application developed for the scientists was MagnaVista. This application provided a completely configurable environment through which the scientists could navigate to and access a broad array of life science data sets of relevance to their research. The basic user interface to MagnaVista is depicted in Figure 2. Here the user can include the genes that they are most interested in (central pop up window). The lower left corner of Figure 2 lists the remote data sources that are accessible. The pop up window to the right of Figure 2 shows the data sets that are returned to the user upon submission of the query.
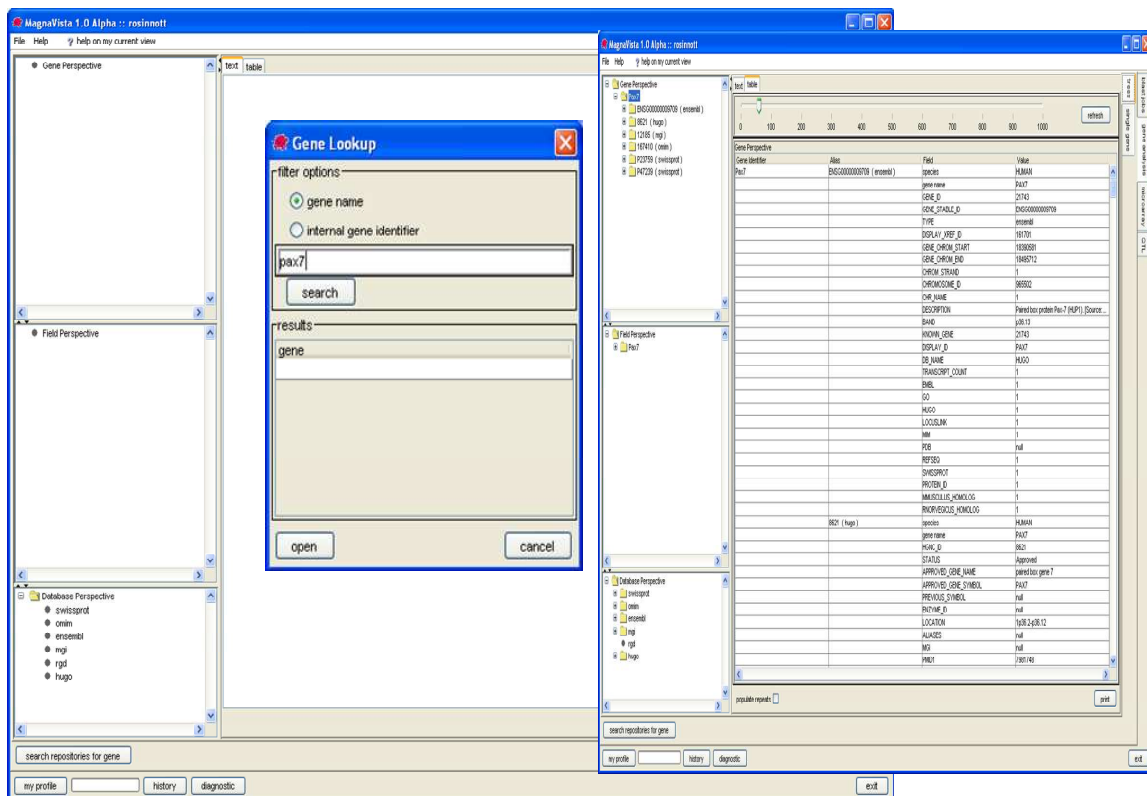

Figure 2: MagnaVista Basic Usage for Gene Query

Thus rather than the user manually hopping to each of these remote resources, a single query is used to deliver collections of data associated with the genes of interest. To support the specific targeted data needs of the scientists, the MagnaVista application could be personalised in various ways. For example, users could select specific (remote) databases that should be interrogated; select various data sets (fields) that should be returned from those databases; store specific genes of interest, and personalise the look and feel of the application itself.

The actual MagnaVista application itself was Java based and delivered to the users using Sun Web Start technology. Through launch buttons on the portal web page, a single mouse click could be used to automatically deliver the application and associated libraries, including the Web Start environment if it is not already present. However due to anomalies in Web Start with non-Internet Explorer versions of browsers used by the scientific community and issues of local firewalls blocking Web Start traffic, it was decided that a simpler version of this application was needed. It was also the case that the scientists were uncomfortable with the personalisation possibilities and having multiple panels and windows. In short, the application was not immediately intuitive and simple to use. The GeneVista was produced to address these issues.

GeneVista is a portlet based application. Portlets are Java-based Web components, managed by a portlet container, that process requests and generate dynamic content. Portals use portlets as pluggable user interface components that provide a presentation layer to information systems which enable modular and user-centric Web application access. Through a portlet based approach, the issues in firewalls and problems with Web Start with non-Internet Explorer browsers were overcome.

In essence the functionality of GeneVista is very similar to MagnaVista. However, it does not support the richness of personalisation. We note that this was at the request of the scientific end users. They simply wanted to be able to select a collection of gene names and retrieve all available information. Few of them bothered with personalisation possibilities. A Google-like front end to GeneVista was designed to reflect this (left hand side of Figure 3). The GeneVista portlet simply requires that the scientist input the gene names that they are interested in and selects submit. Following this, HTML based data sets are returned and presented within the browser window as shown on the right of Figure 3.
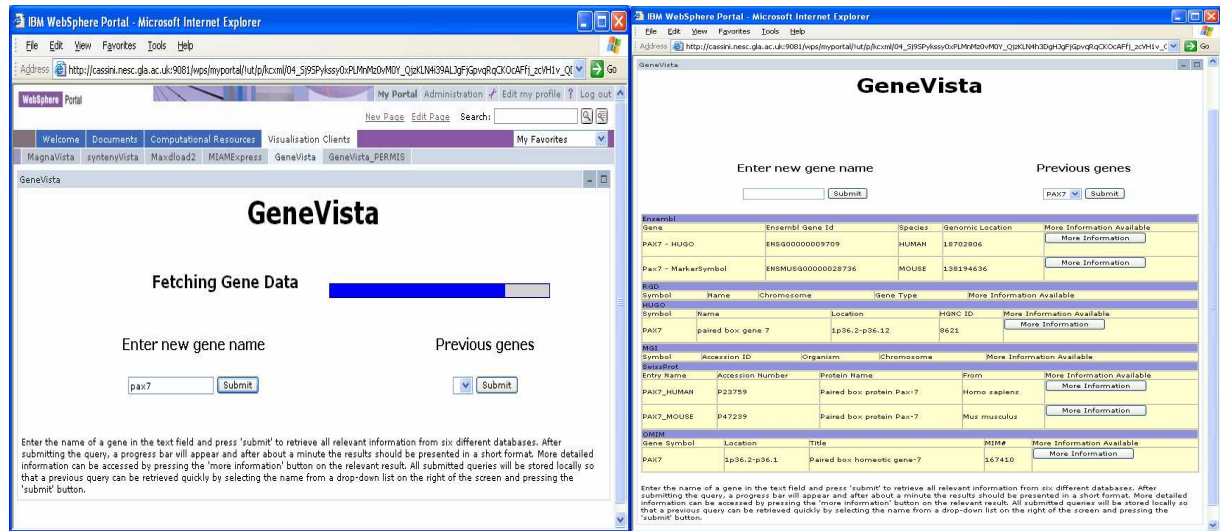

Figure 3: GeneVista Basic Usage for Gene Query

3. Client Compute Grid Tools

In their pursuit of novel genes and understanding their associated function life scientists often require access to large scale compute facilities to analyse their data sets, e.g. in performing large scale sequence comparisons or cross-correlations between large biological data sources. The Basic Local Alignment Search Tool (BLAST) has been developed to perform this function. Numerous versions of BLAST currently exist which are targeted towards different sequence data sets and offer various levels of performance and accuracy metrics. BLAST involves sequence similarity searches, often on a very large scale, with query sequences being compared to several million target sequences to compute alignments of nucleic acid or protein sequences with the goal of finding the n closest matches in a target data set. BLAST takes a heuristic (rule-of-thumb) approach to a computationally highly intensive problem and is one of the fastest sequence comparison algorithms available.

The scientists would thus like access to large scale HPC facilities. It is the case in the Grid community right now that in order to access resources such as those made available through the National Grid Service (www.ngs.ac.uk) end users are expected to have a valid UK e-Science X.509 certificate. In the experiences of the BRIDGES project, this was not something that the biological end users were comfortable with (and they did not do!). To address this, the GridBLAST service developed did not require the end users to possess such a Grid certificate. Instead the X.509 certificate associated with the machine on which the service was hosted was used for job submission to the Grid infrastructure.

It is also the case that even with a certificate, usage of BLAST on large scale HPC resources is often non-trivial to achieve and typically requires knowledge of scripting languages (for decomposing the input data sets and recomposing/merging the results data) and local job schedulers. It was recognised in BRIDGES that users should not have to learn the often complex options associated with job submission to job schedulers such as Condor (www.cs.wisc.edu/condor) or OpenPBS (www.openpbs.org). In addition, one of the primary benefits of Grid technology is the ability to dynamically select and use a variety of heterogeneous resources is essential. This in turn requires that meta-schedulers are available that can dynamically schedule jobs across a variety of heterogeneous resources utilising a variety of local job schedulers. The BRIDGES Grid BLAST service which provides such a simplified BLAST based job submission system, enabling access to and usage a collection of HPC facilities is shown on the left of Figure 4.
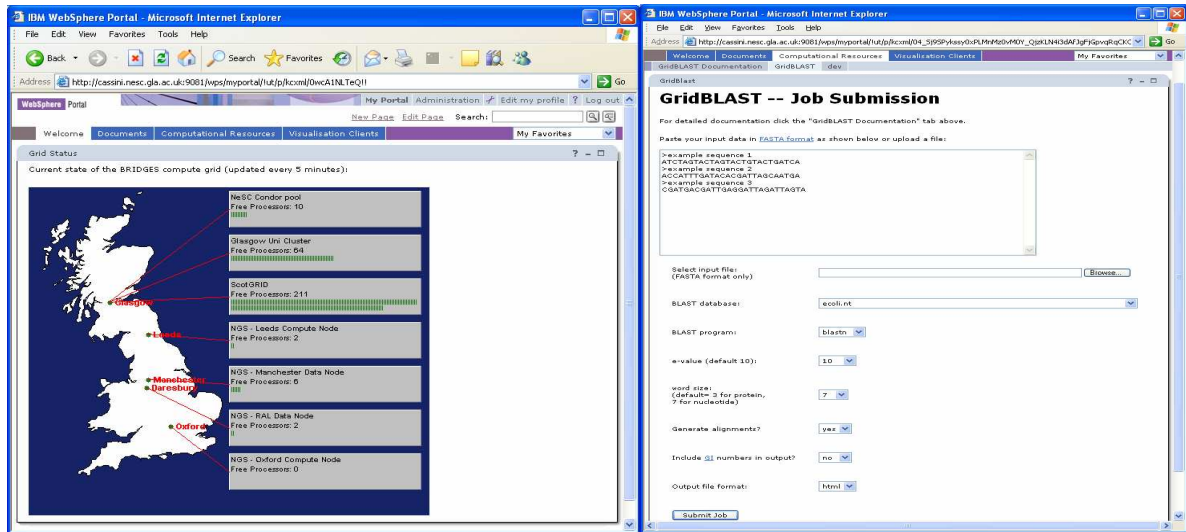

Figure 4: Grid enabled BLAST service

The current BLAST service makes use of the ScotGrid cluster (www.scotgrid.ac.uk), other HPC clusters at the University of Glasgow, Condor pools at the National e-Science Centre and all nodes of the National Grid Service. The status of these resources is shown to users (left of Figure 4). This service provides intelligent default settings for a variety of BLAST services (right of Figure 4). When used, the service checks what resources are available, where the jobs are best run and subsequently provides a prediction of how long the complete BLAST job will take to complete. In addition, monitoring of the status of the various sub-jobs is undertaken (Figure 5) and staging of the various input and output files onto the compute resources is provided. Users can see where their jobs have been submitted and their status at any given time.
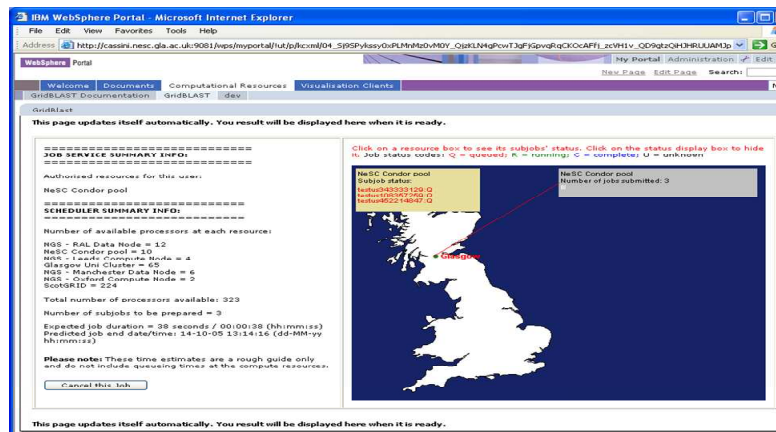

Figure 5: Monitoring the Status of the GridBLAST service

Underlying the GridBLAST service is a fine grained security infrastructure which authorises access to different resources to different users based upon their roles. To support this, the PERMIS authorisation software (www.permis.org) is used. A PERMIS authorization policy controls what resources the job can be run on for that particular user. Given that we do not mandate that end users have a UK e-Science certificate, but provide services which allow access to resources such as NGS through server certificates requires that detailed logging of user actions is made. The security infrastructure seamlessly addresses the identification of the users and submits their jobs to the appropriate resources (as defined in the policy for that user). Currently we support three policies:

- If they are unknown users the job will only be submitted to the local Condor pool (we allow anyone access to the portal, however we restrict what they are allowed to do once there).
- If we recognise the users but they do not have a local ScotGrid account the job will be submitted to the Condor pool and NGS.
- If we recognise the users and they have an account on ScotGrid then the job will be submitted potentially to the Condor pool, the NGS and to ScotGrid (based on job numbers).

4. Conclusions

Usability is crucial to the uptake and success of Grid technology and the core programme as a whole. End user scientists require software which simplifies their daily research and not make this more complex. The idea of getting training on use of Grid software and resources is quite simply not something many scientists have the time or inclination for. Grid application and software developers need to address this fact. The BRIDGES project has developed real data Grid and real compute Grids which have taken into account real biological user demands and explicitly targeted ease of use. The BRIDGES services are helping to shape the wider UK Grid activities – for example helping to define the biological data sets being deployed across the National Grid Service.

It is a fact that the customer is always right. Whilst BRIDGES has developed much richer services in terms of functionality such as MagnaVista, end user scientists did not feel comfortable with these services hence simpler services have been engineered. Simpler and more intuitive user interfaces are crucial for the success of Grid applications. Similarly, solutions which help to overcome existing requirements on Grid infrastructures, e.g. possession of X.509 certificates, are required. Why should a biologist need an X.509 certificate when they only want to run BLAST jobs on available HPC resources? Such ideas are being taken forward in many other projects at the National e-Science Centre at the University of Glasgow where fine grained security is required, but client side software has to be trivial (and not include any complex Grid middleware solutions).

The question is worth asking as to the take-up of these services by the biological community. It is the case that the effort in keeping data fresh and ensuring that data schemas are updated accordingly is an on-going challenge. Throughout the lifetime of BRIDGES various upgrades to the remote databases have taken place necessitating changes to the underlying Grid infrastructure. From a user perspective this is off-putting since a change to a remote schema will automatically cause an exception to be raised within the applications themselves. With MagnaVista for example, this was a cryptic error message needed by the developers to ascertain which SQL command caused the error to occur (which would subsequently be used to track down the remote schema change and eventual upgrade of the system). As a result the widespread acceptance of the data Grid infrastructure did not get thoroughly adopted, despite our attempts to make it as user friendly as possible as per the scientific requests.

The BLAST service was less problematic. However, even here the users had significant issues. Thus for example, the BLAST service was prototyped where a portlet was used for job submission and for tracking and monitoring of associated jobs. This works well for smaller jobs however when very large BLAST jobs are submitted (over 30,000) input sequences, then waiting for and viewing this information via a portlet was not a scalable or apposite solution. Instead the system was redesigned to capture completed jobs in a

database and users were sent emails when these all jobs were complete. The biological community at Glasgow University are quite enthused about this service and currently using it in their daily research.