The Australian BioGrid Portal: Empowering the Molecular Docking Research Community

Hussein Gibbins[1], Krishna Nadiminti[1], Brett Beeson[3], Rajesh K. Chhabra[3], Brian Smith[2], and Rajkumar Buyya[1]

[1] **Gri**d Computing and **D**istributed **S**ystems Lab
Dept. of Computer Science and Software Engineering
The University of Melbourne, Australia
Email: {hag, kna, raj}@csse.unimelb.edu.au

[2] Structural Biology
Walter and Eliza Hall Institute
Parkville, Melbourne
Email: bsmith@wehi.edu.au

[3] High Performance Computing, Information Technology Services
Queensland University of Technology, Australia
Email: {b.beeson, r.chhabra}@qut.edu.au

ABSTRACT

This paper presents the Australian BioGrid Portal that aims to provide the biotechnology sector in Australia with ready access to technologies that enable them to perform drug-lead exploration in an efficient and inexpensive manner on national and international computing Grids. Access will be provided to docking applications and a wide variety of chemical databases. In addition, analysis of the screening results will be made possible using web-based tools, along with archival of these results. The portal aims to become a complete molecular docking e-Research platform, from user management, through to experiment composition, execution over grid resources, and results visualization. One of the most sought after functionalities in the Grid Portals today is 'Persistence' and this paper presents a solution which not only offers persistence but also provides portability across the other JSR168 compliant portal containers. This portal also offers a mechanism for users to easily manage multiple projects.

1. INTRODUCTION

Grid computing [1] is not simply a means for researchers to do existing research faster, it promises them a number of new capabilities. While the ability to carry out existing experiments in less time is definitely beneficial, other facilities such as working in collaborative environments, reducing costs, and gaining access to an increased number of resources and instruments, allows for more advanced research to be carried out.

In order to achieve these goals, a lot of work has been put into Grid-enabling technology, including Grid middleware [4], authentication mechanisms [3], resource schedulers [5], data management [6] and information services [7]. These technologies form the basic services for achieving the higher goals of the Grid – creating e-Research environments [2]. A number of initiatives, such as APAC Grid [8], EGEE [9] and NGS [10], have been started in country or continent wide efforts to build Grid infrastructure, which will offer

these basic services for use by research communities. However, providing this infrastructure is only part of the solution. It is only really once all components of the Grid are integrated seamlessly behind a single user-interface, that we can begin to fully empower research communities, researchers can go back to focussing on their research, and the true value of e-Research can be realised.

The Australian BioGrid Portal, a support project of the APAC Grid, is a web portal that aims to provide the biotechnology sector in Australia with ready access to technologies that enable them to perform drug-lead exploration in an efficient and inexpensive manner using grid-based methods. It aims to build on the previous efforts of The Virtual Laboratory [11], providing access to docking applications and a wide variety of chemical databases. In addition, analysis of the screening results will be made possible using web-based tools, along with archival of these results. This will be a complete molecular docking e-Research platform, from user management, to experiment composition, execution over grid resources, and eventually results visualization.

In recent years JSR168 [25] has emerged as a specification for developing portlets, and has been widely adopted by industry and within the portal community in general. Industry leaders in this arena such as IBM, Sun, BEA, Apache and Vignette are supporting this specification and have developed JSR168 compliant portlet frameworks. Many open source portlet containers have also been developed from various parts of the world showing growing community support for the JSR168 standard. GridSphere [26] is one of the open source JSR168 compliant portlet containers, and has been adopted as a standard toolkit for Grid portals development in the Australian APAC Grid Program. We have chosen GridSphere to develop the BioGrid Portal. Most importantly we are supporting the JSR168 standard and are open to evaluating other portlet containers in the future. Today GridSphere is essentially one of the best open source toolkits available for developing 'Grid' Portals.

JSR168 specification was developed as a generic portlet environment and it doesn't deal with the "persistence" requirements that most Grid Portals will require. Persistence is important for Grid Portals since Grid-based jobs are expected to be lengthy and ensuring that information is not lost at runtime is critical. GridSphere uses an open source object/relational persistence service for Java called Hibernate [27] as a layer to bring persistence to their Grid Portals. Some other Portal developers have also used Castor [28] to bring that persistence layer to the Grid Portals. This portal presents a persistence solution which is portable to other portlet containers since it is separated from the container itself.

The rest of the paper is organized as follows. In Section 2, we provide an overview of what molecular docking is and identify some of its challenges, and the types of issues it presents, to better understand the requirements (Section 3) of the portal. In Section 4 we present the overall system architecture, followed by the design and implementation (Section 5). In Section 6 we give a walkthrough of how a biologist would interact with the portal. Finally we discuss the current status of the project and outline the future direction of the portal (Section 7).

2. MOLECULAR DOCKING

Drug discovery is an extended process that can take as many as 15 years from the first compound synthesis in the laboratory until the therapeutic agent, or drug, is brought to market.

In silico, or computer-based, screening techniques [17] involve screening very large numbers (of the order of a million) ligands or molecules in a chemical database (CDB) to identify a set of those that are potential drugs. This process, called molecular *docking*, helps scientists in predicting how small molecules, such as substrates or drug candidates, bind to an enzyme or a protein receptor of known 3D structure. Docking each molecule in a chemical database is both a compute and data intensive task.

Since the process of docking individual molecules is independent from one-another, this problem lends itself to parallelisation and can thus be implemented as a master-worker parallel application. This means that we can take advantage of HPC technologies such as clusters and Grids to improve overall execution time and allow for large-scale data exploration. It is our goal to use Grid technologies to provide cheap and efficient solutions for the execution of molecular docking tasks on large-scale, wide-area parallel and distributed systems. This project will improve accessibility to these *in silico* techniques to regular biologists by reducing the cost of utilising HPC technology and of licensing the docking software, as well as reducing the level of technical expertise needed to conduct such advanced experimentation.

3. REQUIREMENTS

The broad requirement for the Australian BioGrid Portal is the creation of an e-Research environment; one that enables biologists to perform their molecular docking experiments through a web portal and to eventually be deployed over the APAC Grid infrastructure. Further details on some of the project's requirements, which for the most part are also applicable to other research domains, help support our approach.

User Management

The portal needs to provide support for simultaneous access by multiple biologists, single sign-on, and individual workspaces in which biologists can experiment safely.

Project Management

Biologists need the ability to run multiple simultaneous experiments. Long running experiments need to be able to continue running after a biologist logs out.

Improve Accessibility

To hide complexities of the Grid infrastructure, integration of a service to support automated resource discovery, allocation and access control details (Virtual Organisation [12]) is required.

Security

The molecule data within chemical databases and experimentation results are often

sensitive, and need to be protected. Therefore Grid security is important, and data communication between resources should be secure.

Visualisation
Once an experiment is complete, it is useful for biologists to have immediate access to visualization tools that allow them to visualise the resulting interactions between the screened molecules and the target.

Accounting and Quality of Service
In production, services will not be free, so records of resource usage need to be kept so that usage can be billed accurately. Biologists should then have some control over how much they are willing to pay for a given experiment as well as how long it should take to finish.

4. ARCHITECTURE

The high level architecture for the system involves a web portal and its interaction with the underlying Grid infrastructure, shown below in Figure 1.

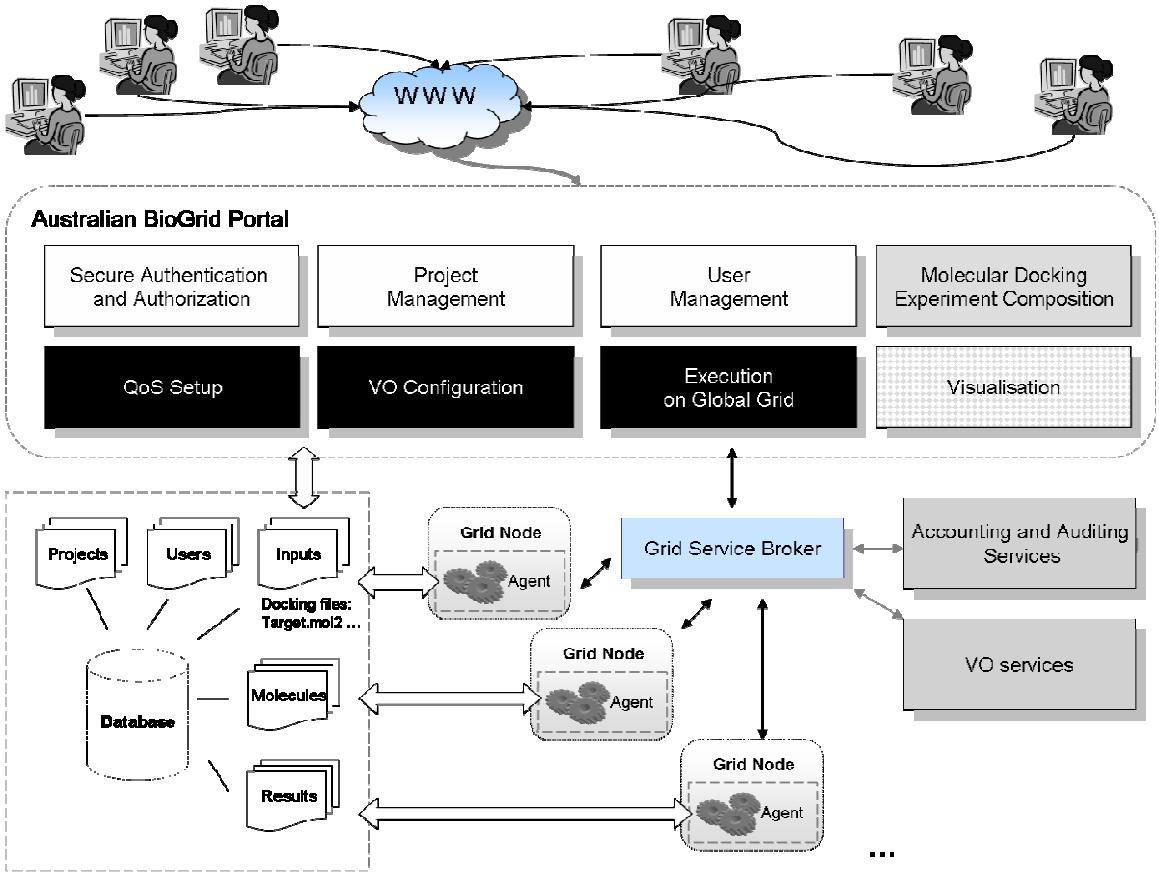

Figure 1 : Architecture of The Australian BioGrid Portal

Biologists interact with the portal, which in turn coordinates their work and interacts with the Grid infrastructure on their behalf. There are numerous components within the

system, which can lead to it becoming complex and difficult-to-maintain. We strive for an architecture which groups these components into subsystems with well-defined interfaces. For example, we use the Grid Service Broker to handle all Grid execution details. A quicker alternative would be to tightly couple these details within the application specific portal. However, this means we would have to change user interface when the underlying Grid software changes causing code reuse and sharing to become much more difficult.

Following is a description of the different components which make up the architecture.

4.1 The Portal

For the implementation of the Portal, we've decided to use portlet technology. Portlets were chosen because they are reusable Web components that are used to compose web portals. Having reusable components will be beneficial in building new portals in the future. Each portlet may be completely independent but are organised and presented to the user together as a single web page. Our web portal can be divided into four major components: user and project management, experiment composition, experiment execution, and visualization and analysis.

The great benefit of the portlet paradigm is the ability to reuse components. Only the Molecular Docking Experiment Composition and Visualisation portlets are domain specific. Some configuration will always be required in order to describe how experiments in different research domains behave, but we can keep portlets such as Execution non-domain specific by passing experiment descriptions to a Grid Service Broker and allowing it to coordinate execution on our behalf.

4.1.1 User and Project Management

These components are not specific to the domain of molecular docking, and deal with the generic entities applicable to any type of research: users and projects.

The portal will provide a mechanism for Grid based security while hiding the details from the biologist. The Authentication module will be used to log the user into the portal, while at the same time obtaining their Grid proxy [18]. The Grid proxy will then be used as a means of authentication to Grid resources and allowing for secure communication.

The user management module will be used to manage biologists and similarly the project management module will be used by individual biologists to manage projects. During experiment composition and execution, links will be made between experiment data and the project being worked on, and this data will be stored in the database.

Grid middleware such as the Globus Toolkit provides the capability to perform low-level Grid tasks such as copying files, executing processes and monitoring process output. Scientists work at a higher level, dealing with 'Experiments' and their related data. Experiments will be a set of numerous individual tasks using and producing experiment data. The user interface will provide an environment where biologists can work at the

level of Experiments and we let the Grid Service Broker handle the communication with Grid middleware to perform individual operations.

4.1.2 Experiment Composition

Depending on implementation, these components may or may not be entirely specific to molecular docking. Within the experiment composition, a biologist is able to set up their experiment by uploading input files and assigning values to docking-specific parameters. Actions such as uploading input files could of course be made generic and used within other e-Research portals. During experiment composition the biologist builds a blueprint for the experiment with all the details being stored in the database for later retrieval. This means that once the experiment has been designed by the biologist, the full details have been stored and are later available to other components.

4.1.3 Experiment Execution

The job of the execution component is to retrieve the details of the experiment from storage, as described by the biologist, and use these details to coordinate the experiment over the Grid. This component will also provide execution monitoring, scheduling based on users quality of service needs, and fault tolerance. Interaction with the Grid accounting mechanism is also included here. Results of execution will be stored along with other project information creating a complete description of the experiment.

4.1.4 Visualisation and Analysis

This component provides the ability for researchers to view the results of their experimentation. Post-processing and visualization tools can be invoked from within the portal to help with analysis.

4.2 Virtual Organisation

The Virtual Organisation (VO) service is used for user authentication and authorization as well as resource discovery. The VO will contain information about which resources the biologist has access to. This will remove the need for individual biologists having to setup and maintain accounts on various resources. Access to Grid resources will be made transparent.

4.3 Accounting and Auditing Services

An audit trail needs to be kept to provide a record of who, what, where and when execution has occurred during an experiment [13]. Not only will this assist biologists in understanding exactly what occurred during their experimentation, but this service is also important for billing usage.

4.4 Grid Service Broker

The Grid Service Broker coordinates the execution of the experiment on the Grid. It will interact with the VO service to discover available resources, manage the execution of docking on each of the compute nodes, record execution details with the auditing service and make sure usage of resources is being charged, amongst other things.

Executing work on the Grid is a complicated task. Even though low-level Grid middleware provides us with the infrastructure for submitting and monitoring single jobs, managing the execution of many jobs on many resources for many projects of many users is difficult. Add other complexities like resource discovery, accounting, retrieving results, fault tolerance and optimization, and it becomes a lot of work. We aim for this complexity to be absorbed within the Grid Service Broker, providing a simple API on top of which we can develop.

5. DESIGN AND IMPLEMENTATION

Our architecture is not specific to our application. For the application-specific *components,* our technology choices are limited. For the Grid components, the immature state of Grid technology means there are numerous portlet containers, Grid middleware systems and security systems. Technology choices are both high-risk and time-consuming, so we detail our choices and possible alternatives.

For our system we have chosen to use GridSphere [14] for our web portal, MyProxy to perform the user authentication duties of the VO, GridBank for accounting and auditing, the Gridbus Broker for our Grid service broker, and Globus 2.4 is used as the Grid-enabling middleware on each of the compute nodes. This information is presented in Table 1. The system is primarily Java based.

Component	***Technology Used***	***Comments***
Portal	GridSphere Portlet Framework	Reusable web components. Standards compliant. Open source. Offers some support for running Grid applications.
Virtual Organisation (User Authentication)	MyProxy	Currently only considering authentication duties of VO
Accounting & Auditing	GridBank	Provides the ability to charge for resource usage. This has not been integrated yet.
Resource Service Broker	Gridbus Broker 2.0	Coordinates execution on the Grid
Compute Nodes	Globus 2.4	Provides support for authentication and submitting and monitoring jobs to compute nodes.
Database	MySQL [19]	Open source, relational database
Visualisation	AstexViewer [15]	Applet for visualizing chemical structures
Docking Software	DOCK 4.0 [16]	Software to perform molecular docking.

Table 1 : Technology Choices

Portal Implementation

The portlet framework chosen was GridSphere because it was open source and JSR 168 compliant. Recent surveys [22][23] of portlet containers showed that GridSphere is comparable with other popular portlet containers (such as uPortal, LifeRay and Pluto) across a broad range of criteria. As the name implies, GridSphere has many Grid-specific

features unlike other containers which are more generic. While many of these features are not portable, discussions with the developers have reassured us the inter-container operability will be implemented in the coming months. GridSphere already complies with JSR 168, which means that our code can be shared and deployed in different portlet containers.

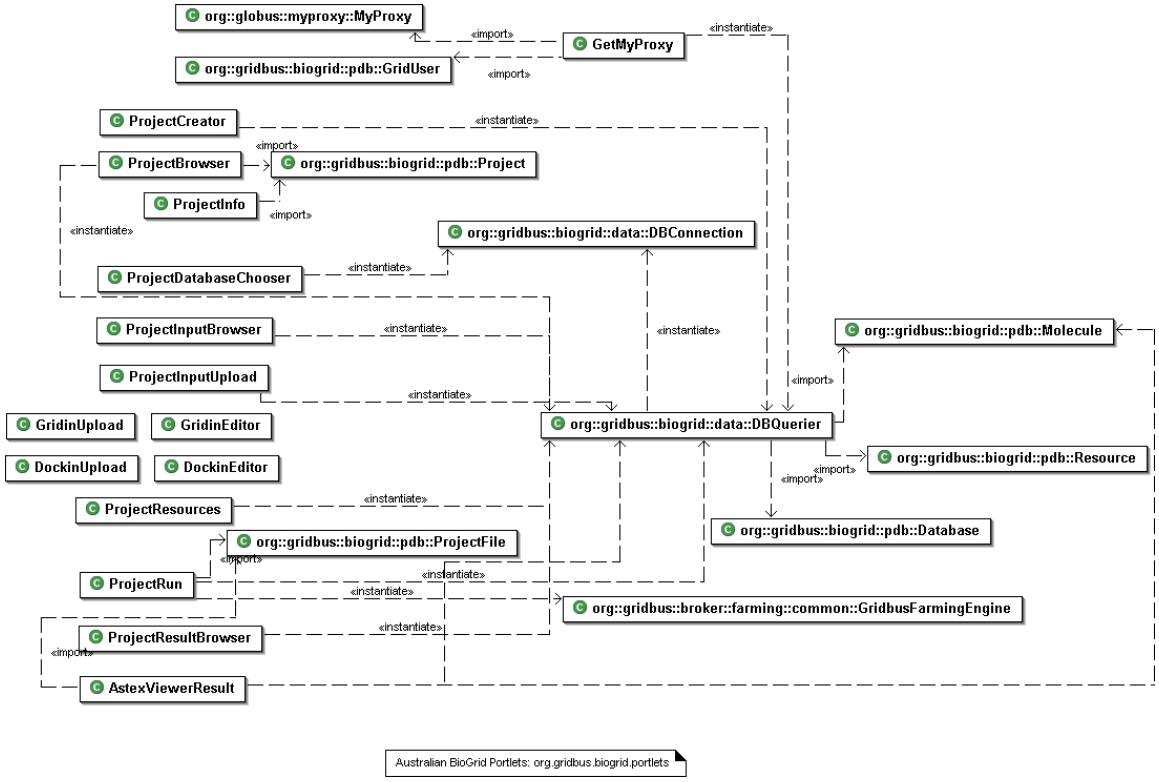

Figure 2 : UML diagram of The Australian BioGrid Portal.

Each piece of functionality was developed as a separate portlet or a small group of related portlets. To increase portability, inter-portlet communication is restricted, and information sharing occurs via the database. Figure 2 illustrates the separation between each component and also their interaction with the database via the DBQuerier class. This independence means that if a new improved set of experiment composition portlets were developed, they could easily be plugged in to replace the existing ones.

Authentication and single sign-on

GridSphere offers the ability to create custom authentication modules that are invoked once a user logs into a generic login screen. When logging into a GridSphere portal, a user is authenticated using the authentication module defined for that portal. There is also the concept of chaining modules, so that if one module fails to authenticate the user, other methods can then be tried. In order to achieve single sign-on, we needed to create our own module that would retrieve a biologist's proxy certificate and store it in the database to be accessed later. A custom authentication module was made for the portal that contacted a MyProxy server, downloaded the biologists proxy certificate based on the username and password supplied at login, and saved it to the database.

Data Storage
Although we use a single database for both user and experiment data, this data could be separated if required. For example, we may later wish to user a dedicated database server to store experiment data such as molecule structures. MySQL is open source and is a popular choice for Java development, although other implementations such as PostgreSQL (Posgres) could be used in its place.

The molecules to be used for screening are imported into the database along with user, project and experiment data. There was no convincing reason to use multiple databases (one for the molecules and one for other application data) at this stage, nor was there an advantage of using a combination of SQL database and SRB (for files). It was decided that if scalability ever became an issue, techniques similar to those adopted by high-demand web sites or Grid databases, could always be applied later. We also made sure that nothing was written to the database used by GridSphere, as this would reduce the portability and flexibility of the portal.

Project and Experiment Composition
Experiment composition is just a means of getting all the data required to perform the experiment into the database. A number of portlets were created to allow the biologist to upload input files, modify experiment parameters, and select a set of molecules to be used for screening.

Experiment Execution and Monitoring
The changing nature of Grid software means the underlying middleware like Globus will soon change. In addition to our architecture requirements, we must be shielded from such change. Most Grid development has focused on lower level middleware, so Grid Service Brokers are relatively immature. The Gridbus Broker has been used in a number of projects already [24][29] and has been designed with extensibility in mind. Other brokers, such as Nimrod-G, could be used but a Java interface makes development within the portlet environment easier. This type of functionality is provided by resource brokers such as the Gridbus Broker, which is why we have chosen to use it for the experiment execution within the Australian BioGrid Portal. Persistence of user data (such as jobs running) is essential. If our portlet container crashes we must not lose this information. As mentioned earlier, GridSphere provides persistence via Hiberate, but it's currently not portable. By keeping persistence at the Grid Service Broker level, we can change portlet container *and* still have persistence.

When told to execute the experiment, the resource broker identifies which Grid resources it should use, gets the user's proxy out of the database for authentication with Grid resources, creates a grid application description based on properties of the project, and then begins running the experiment on the Grid. Results of the execution are stored back into the database along with other experiment information. We store executing threads of the experiment within the portlet container's Java Virtual Machine (JVM). That is, instances of the Gridbus Broker are stored in the Portlet Context. However, experiment management can be system resource intensive and managing a large number of

experiments at once within the portlet container can put unnecessary burden on the web server. This also creates a dependency between the portlet container and the service broker. To avoid this, the Gridbus Broker will be migrated to a web-service and will keep the existing interface. With such a system, we can have dedicated resources for experiment management and ensure that restarting or changing the portlet container doesn't destroy experiment execution.

It should be obvious from this description that we could always simply plug in a different set of experiment execution portlets if we wanted. The new portlets would do the same job of extracting the experiment details from the database, but code would need to be written to submit, monitor jobs on the Grid.

Efficient Data Management

When docking is performed, it is common for multiple results to be generated for each molecule in a database. With general usage, the resource broker would send input files to Grid nodes and then retrieve results after execution. In our scenario however, we wanted to reduce unnecessary data communication as much as possible. The broker acting as a middle man between the database and the compute resources causes a problem. The broker needs to transfer files locally and then forward these onto the required recipient. If the broker is running on its own separate Grid resource, then the overhead of passing inputs and results via the broker creates a great overhead. This led to the creation of an agent which is sent to a resource the first time it is asked to do work. The agent then allows the resource to talk to the database directly for retrieving input and storing execution results. The job of the broker then becomes one of informing the agent of which inputs to retrieve and where to store the results.

Result Visualisation and Analysis

The AstexViewer applet is currently used to visualize the output. To visualize a result, the result stored in the database for a particular molecule needs to be extracted, the target protein also needs to be extracted, the result is superimposed onto the target, this is then run through a molecule file format converter called Open Babel [20] to convert it into a format readable by AstexViewer, and finally given to a Java applet for visualization. It is intended for other visualization options to be available. Results can also be downloaded locally if required.

Docking Software

The DOCK molecular docking software from UCSF was installed on each of the compute nodes. Optionally the executables could be staged to a resource at the start of execution. Two reasons for avoiding this are: 1) the executables are potentially very large and may require installation, 2) because of licensing issues it is assumed compute nodes have the software installed.

6. A DOCKING EXPERIMENT WALKTHROUGH

Here we will walk through the process taken by a typical biologist as they interact with the portal (in its current state of implementation) to conduct a molecular docking experiment.

We have set up the molecular docking environment, and have deployed it on different Grid resources distributed both nationally and internationally, shown in Table 2. In this experiment we have chosen to use the proteolytic enzyme 'thermolysin' as the target, and have screened it against a sample chemical database containing 100 molecules.

Component	***Resource***
Biologist's Desktop	gieseking.cs.mu.oz.au
Grid Service Broker & Portal	manjra.cs.mu.oz.au
Authentication (MyProxy server)	its-hpc-ibm1.its.tils.qut.edu.au
Database	bart.cs.mu.oz.au
Grid Resources (compute nodes)	brecca-2.vpac.org, lc1.apac.edu.au, c20.besc.ac.uk, c21.besc.ac.uk, belle.anu.edu.au, belle.physics.usyd.edu.au, belle.cs.mu.oz.au, manjra.cs.mu.oz.au

Table 2 : The Molecular Docking Grid Environment

6.1 Getting started

A biologist has just found a new target protein and wants to search an existing chemical database to find any molecules that dock favourably with it. The biologist prepares the input files and accesses the portal.

Logging in

The biologist logs into the portal by supplying their username and password. The username will first be used to verify that the biologist is a registered user of the portal, and then both the username and password are used to download the biologist's Grid proxy from the MyProxy server. Once the proxy is obtained it is stored in the database for later retrieval. This achieves a single sign-on mechanism that also hides the usage of grid proxies from the biologist.

Figure 3 : Logging into the portal.

Creating a new project

Once logged in, the biologist is presented with a listing of their projects. Since the biologist will be conducting a new experiment, a new project is created by specifying the

new project name in the 'New Project Name' text field and then selecting 'create'. The biologist creates a new project called 'my experiment'. 'my experiment' is then selected as the project that is currently being worked on.

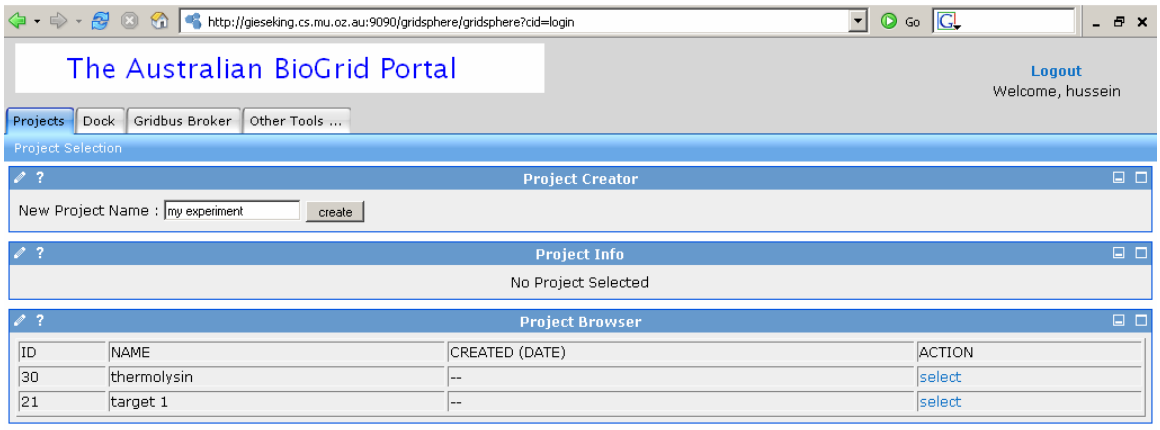

Figure 4 : Creating a new project.

6.2 Experiment composition

Now the biologist needs to set up the docking experiment. The biologist selects the 'Dock' tab to move onto the experiment composition.

Chemical database

The biologist uses the drop-down list to pick a database from those available, to be used in the experiment. 'new_molecules.db' is selected and the biologist clicks the 'select' button. A link is made in the database between the selected database and the current project.

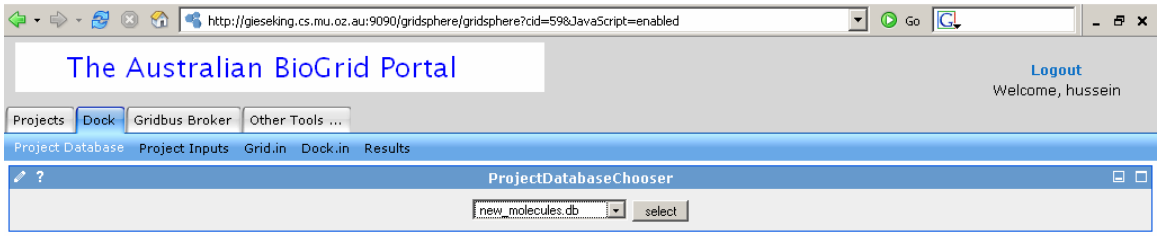

Figure 5 : Selecting molecules to screen.

Input files

The biologist moves on to uploading input files. A list of standard molecular docking files is presented to the biologist, each of which must be uploaded into the project before the experiment can be executed, including files describing the target protein. The biologist uploads each of the required files. When uploaded, each of these input files is stored in the database and linked to the current project.

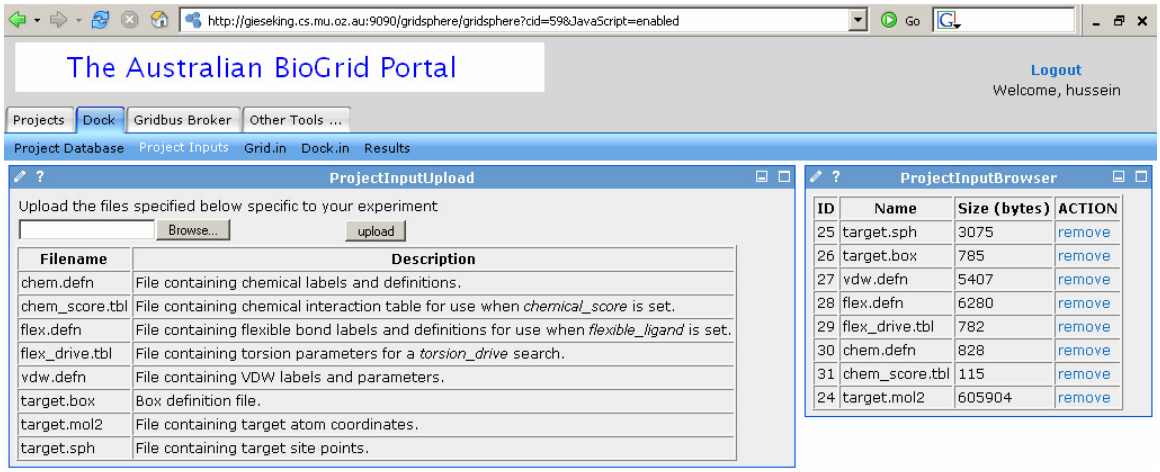

Figure 6 : Uploading input files.

Grid.in and Dock.in parameter files

Another set of input files to be uploaded by the biologist are the grid.in and dock.in file. These files contain a large number of parameters used for controlling the way each molecule is docked. Once uploaded, the biologist is given the ability to modify any of these parameters within the portal interface. As with the other input files, these files are stored in the database and are linked to the current project.

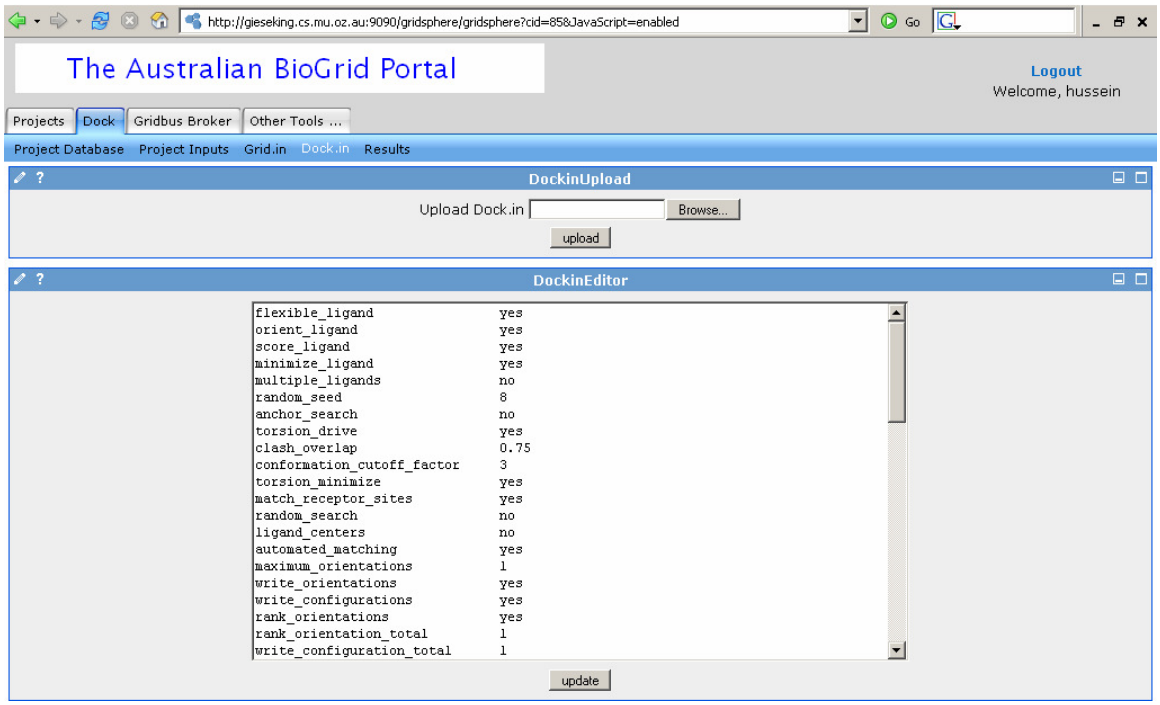

Figure 7 : Editing parameter file.

6.3 Executing the experiment on the Grid

Once the experiment has been set up, all details have been stored in the database. The biologist is now ready to run the experiment on the Grid.

Setting quality of service requirements

Because the results of this experiment need to be made available as soon as possible, the biologist decides to enforce a strict deadline (must be complete within the next 2 hours), while allowing for a relaxed budget (spend as much money as necessary).

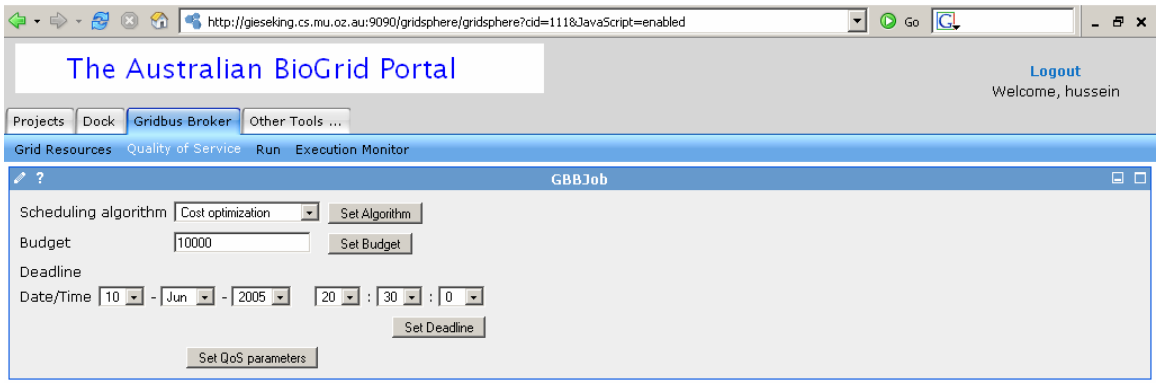

Figure 8 : Setting QoS attributes

Running experiment

The biologist now selects 'run' to start running the experiment.

Monitoring execution

Once the experiment execution has begun, the biologist is keen to see that the experiment gets started successfully.

The biologist is able to view the status of all the jobs in this experiment. He notices that two of the jobs have a 'pre-stage' status, meaning that they are waiting for the inputs to be staged onto the Grid resources. The biologist waits a little while and the statuses of the jobs change to 'running'. The biologist waits a little longer and eventually one of the jobs is marked 'done' and another few have entered the 'running' state. It appears that the execution is going fine. The biologist can also find out further details about each job or resources.

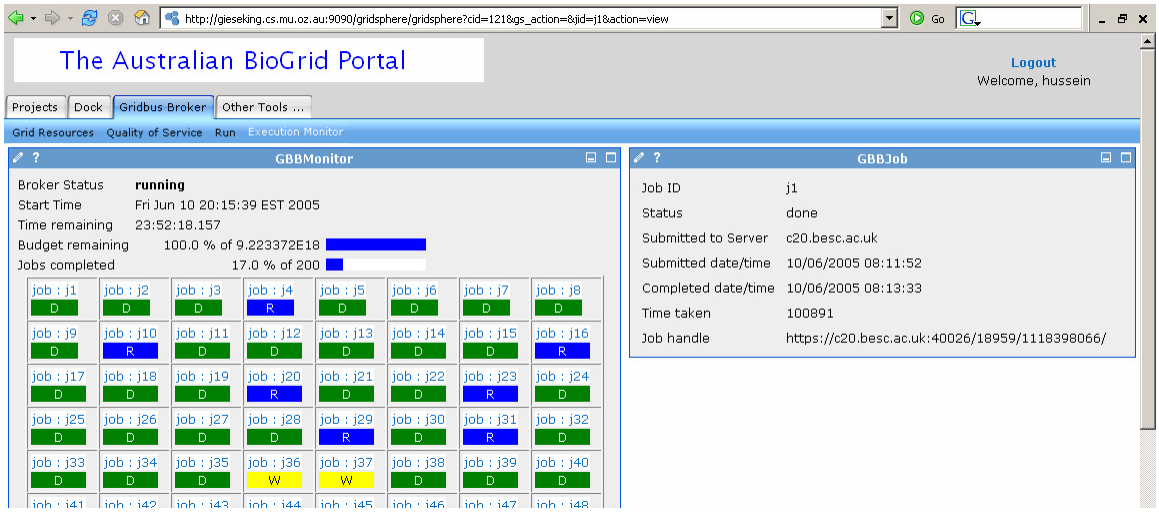

Figure 9 : Monitoring execution.

Logging out
Satisfied that the experiment has been started successfully, the biologist logs out of the portal and continues on with other work.

6.4 Results and Visualisation

Recalling experiment
The biologist can log back into the portal to view the results. The biologist logs in the same way as before, but instead of creating a new project they select 'my experiment'. The biologist can then view the results.

Results Visualisation
The biologist is presented with a listing of the molecules in the database selected for 'my experiment' and selection of output files for each of these molecules, uploaded into the database during execution. The biologist is interested in visualizing the result. By selecting the view option on the result for the first molecule, the visualization software is invoked and displays the first molecule docked onto the target protein.

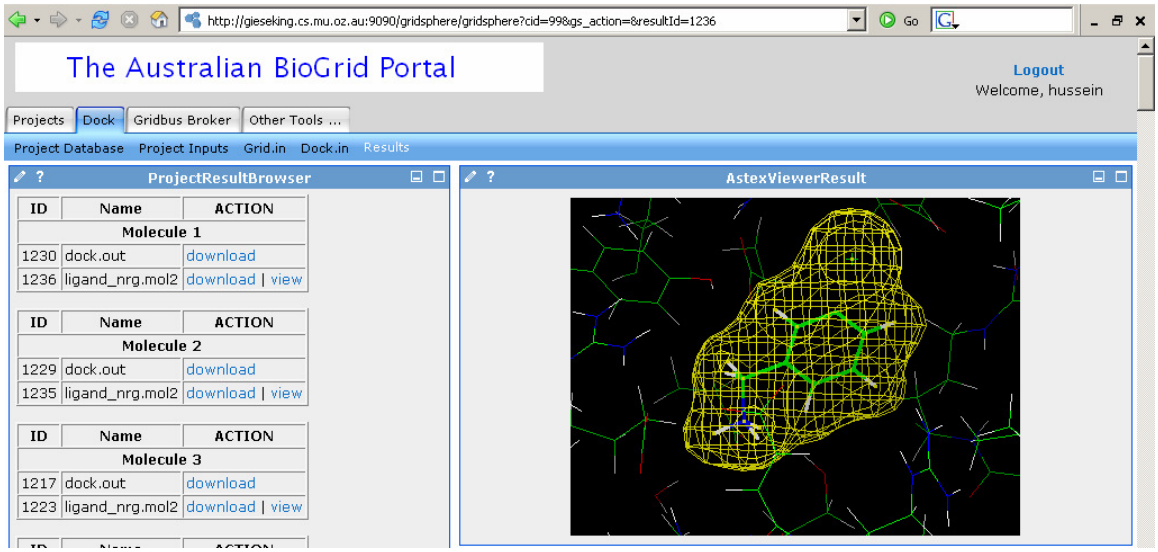

Figure 10 : Visualisation of results.

Experiment Statistics
The statistics for the experiment execution are shown below in Table 3. Since the fork-based execution service provided by Globus on grid nodes was used, only the head nodes of the clusters were utilized.

Experiment start time: June 9, 2005 10:47:44 AM
Experiment end time: June 9, 2005 11:16:48 AM
Total time taken: 29 minutes, 4 seconds.
Average job computation time: 70.5 seconds

Site	*Hostname*	*Configuration*	*Molecules Docked*
APAC, Canberra	lc1.apac.edu.au	150 node, x86 Linux cluster with based on Pentium 4 Nodes	11
VPAC, Melbourne	brecca-2.vpac.org	94 node, 194 CPU Linux Cluster based on Xeon 2.8 GHz CPUs.	9
GRIDS Lab Melbourne Univ.	belle.cs.mu.oz.au	SMP with 4 Intel Xeon CPUs	3
GRIDS Lab Melbourne Univ.	manjra.cs.mu.oz.au	13 node, x86 Linux cluster	3
Belfast e-Science Centre, UK	c20.besc.ac.uk	48 node, Intel x86 based IBM SP2 cluster	9
Belfast e-Science Centre, UK	c21.besc.ac.uk	48 node, Intel x86 based IBM SP2 cluster	65

Table 3 : Experiment Statistics

7. CONCLUSIONS AND FUTURE WORK

Currently The Australian BioGrid Portal allows biologists to conduct molecular docking experiments in a simple e-Research environment. We have successfully met a number of the requirements set out for this project, and we will be refining our solution based on feedback from real biologists.

While the complexity of building rich e-Research environments, from the hardware up to the end user-interface is high, the immense benefits to research communities are clear. By reducing costs, reducing time, promoting collaboration, and increasing the scope of the research, there is bound to be a great advance in the research carried out by various research communities. We have made progress towards achieving this vision with The Australian BioGrid Portal, however we plan to continue improving upon our work in collaboration with the molecular docking research community within Australia. We intend on having a test group of biologist trial the system and feed us with new requirements for the system. Once the system is running at a satisfactory level we will make it live to other biologists in the community.

Initially we intend on improving the integration with Virtual Organisation services to allow for user access control and resource discovery. We will also work towards including accounting services via the GridBank Grid banking service. A number of other features such as tracking of experiment history and complete data security are still to be incorporated into the system. Grid security is continually being added to supporting technologies, so software like GSI enabled MySQL [21] may be incorporated in future.

Features such as Quality of Service and may require further developments in Grid infrastructure before they can be fully applied and useable in a production environment.

Optimisation will also soon become a consideration. Combining the new MPI functionality offered with DOCK version 5.2 with the Grid implementation will offer further optimisation for molecular docking experiments.

ACKNOWLEDGEMENTS

We would like to thank those involved with the User Interface & Visualization Infrastructure Support Project of the APAC Grid. We would also like to thank Ashley Wright of QUT for providing us with access to a MyProxy server, all of those who provided access to compute nodes used in our test bed for running docking experiments, and Srikumar Venugopal his valuable comments.

REFERENCES

[1] I. Foster and C. Kesselman, The Grid: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann Publishers, 1999.

[2] T. Hey and A. E. Trefethen, The UK e-Science Core Programme and the Grid, Journal of Future Generation Computer Systems (FGCS), vol. 18, no. 8, pp. 1017-1031, 2002.

[3] J Novotny, S Tuecke, V Welch, An Online Credential Repository for the Grid: MyProxy, Proceedings of the 2001 International Symposium on High Performance Distributed Computing (HPDC 2001), IEEE CS Press, Los Alamitos, California, USA, 2001.

[4] I. Foster and C. Kesselman, The Globus Project: A Status Report, Proceedings of IPPS/SPDP'98 Heterogeneous Computing Workshop, 1998, pp. 4-18.

[5] S. Venugopal, R. Buyya, and Lyle Winton, A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids, Proceedings of the 2nd International Workshop on Middleware for Grid Computing (Co-located with Middleware 2004, Toronto, Ontario - Canada, October 18, 2004), ACM Press, 2004, USA.

[6] C. Baru, R. Moore, A. Rajasekar, and M. Wan, The SDSC Storage Resource Broker, in Proceedings of CASCON'98, Toronto, Canada, Nov 1998.

[7] J. Yu, S. Venugopal, and R. Buyya, A Market-Oriented Grid Directory Service for Publication and Discovery of Grid Service Providers and their Services, Journal of Supercomputing, Kluwer Academic Publishers, USA, 2005

[8] APAC Grid Program, http://www.apac.edu.au/programs/GRID/, accessed June 2005.

[9] Enabling Grids for E-SciencE (EGEE), http://public.eu-egee.org/, accessed June 2005.

[10] The UK National Grid Service (NGS), http://www.ngs.ac.uk/, accessed June 2005.

[11] R. Buyya, K. Branson, J. Giddy, and D. Abramson, The Virtual Laboratory: Enabling Molecular Modeling for Drug Design on the World Wide Grid, The Journal of Concurrency and Computation: Practice and Experience (CCPE), Volume 15, Issue 1, Pages: 1-25, Wiley Press, USA, January 2003.

[12] I. Foster, C. Kesselman, and S. Tuecke, The anatomy of the grid: Enabling scalable virtual organizations, International Journal of High Performance Computing Applications, vol. 15, no. 3, pp. 200-222, 2001.

[13] A. Barmouta and R. Buyya, GridBank: A Grid Accounting Services Architecture (GASA) for Distributed Systems Sharing and Integration, Workshop on Internet Computing and E-Commerce, Proceedings of the 17th Annual International Parallel and Distributed Processing Symposium (IPDPS 2003), IEEE Computer Society Press, USA, April 22-26, 2003, Nice, France.

[14] J. Novotny, M. Russell, O. Wehrens, GridSphere: a portal framework for building collaborations, Journal of Concurrency and Computation: Practice and Experience, Volume 16, Issue 5, Pages 503 - 513, 2004.

[15] AstexViewer, http://www.astex-technology.com/AstexViewer/, accessed June 2005.

[16] Ewing A (ed.). DOCK Version 4.0 Reference Manual. University of California at San Francisco (UCSF), U.S.A., 1998. http://www.cmpharm.ucsf.edu/kuntz/dock.html.

[17] Kuntz I, Blaney J, Oatley S, Langridge R. Ferrin T. A geometric approach to macromolecule-ligand interactions. Journal of Molecular Biology 1982; 161:269–288.

[18] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, F. Siebenlist. X.509 Proxy Certificates for Dynamic Delegation. *3rd Annual PKI R&D Workshop*, 2004.

[19] MySQL, http://www.mysql.com/, accessed June 2005.

[20] Open Babel, http://openbabel.sourceforge.net/, accessed June 2005.

[21] GSI Enabled MySQL, http://www.star.bnl.gov/STAR/comp/Grid/MySQL/GSI/, accessed June 2005.

[22] R. Allan, C.Awre, M. Baker, A. Fish, Portals and Portlets 2003, Technical Report UKeS-2004-06

[23] R. Crouchly , A. Fish, R. Allan, D. Chohan, Sakai Evaluation Exercise – Summary

[24] B. Beeson, S. Melnikoff, S. Venugopal, D.G. Barnes. A Portal for Grid-enabled Physics, 5th IEEE/ACM International Workshop on Grid Computing

[25] JSR168, http://www.jcp.org/en/jsr/detail?id=168, accessed June 2005

[26] GridSphere, http://www.gridsphere.org/gridsphere/gridsphere, accessed June 2005

[27] Hibernate, http://www.hibernate.org/, accessed June 2005

[28] Castor, http://castor.codehaus.org/index.html, accessed June 2005

[29] B. Hughes, S. Venugopal and R. Buyya. Grid-based Indexing of a Newswire Corpus, Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing (GRID 2004, Nov. 8, 2004, Pittsburgh, USA), IEEE Computer Society Press, Los Alamitos, CA, USA.