Abstract

We present a description of our ongoing efforts to develop a chemistry grid portal by utilizing core grid technologies for computational chemistry and molecular science applications. The portal aims to provide the Australian computational chemistry and molecular science community a uniform and integrated user interface for scientists to access and execute distributed applications through a conventional web browser. The portal will become a complete molecular science e-Research platform from user management, to experiment composition and preparation, execution over grid resources, and finally results analysis and visualization. The requirements for the portal, its architecture and design are discussed, along with implementation details and case studies, and future directions of the project.

1. Introduction

Computational chemistry and molecular science cover a broad spectrum of scientific areas including pharmacology, biology, chemistry, electronics, engineering, environment, medical materials, etc. They often require using various software packages which may be on different computer resources crossing over different physical locations and need to use high performance computing (HPC) resources for execution of their jobs. Theoretical chemistry and molecular science users are major clients of HPC facilities worldwide.

Distributed computational resources cannot be effectively utilized by using typical unix tools based on the remote login. Recent advances in computer technology, especially grid technologies make them good candidates for development of computational environments for utilizing distributed resources [1]. Computational grids enable sharing a wide variety of geographically distributed resources across multiple organizations for solving large scale computational and data intensive problems in chemistry and molecular science. In order to attract users, the computational environment needs to provide easy to access and more user friendly graphic user interfaces that are able to work on different operating systems and hardware platforms.

Some computational chemistry programs which provide graphic user interface, such as GaussView [2], are able to run user applications only on local workstations and access to remote distributed resources is limited. A dedicated user interface [3] that uses UNICORE infrastructure [4] provided an access to a number of computational chemistry codes on distributed computer resources. However, to use this interface, users need to install the UNICORE client together with the plugin software on to their workstations or PC machines. An approach that uses the existing web technology is presented by WebMo [5] which is a web client-server based submission system for quantum chemistry codes such as Gaussian [6], Gamess [7] and Mopac [8]. This system is limited to the local batch systems and does not provide grid capabilities. BioCore [9] provides a web interface to the molecular dynamics code NAMD [10] for web submission across geographically distributed systems, whereas NPACI [11] Gamess Portal and QC Grid [12] present analogous approaches for quantum mechanical codes. However, currently these systems are limited to the particular individual application code and none of them are ever likely to provide an “off-the-shelf” solution for the needs of Australian HPC facilities [13].

Grid enabled portals are becoming increasingly popular as a platform for providing access to Grid
services and resources, making distributed computational resources readily available to individual scientists and large-scale collaborations, in order to enable e-Science (or e-Research) activities through the internet. Grid portals bridge the gap between current Grid computing technology and the user community that uses reliable production systems for their research. By presenting an integrated web browser based interface, the portal hides the complexities of powerful Grid tools from ordinary users. It allows users to utilise distributed computational resources without the need to know the specifications of each computer system environment. Therefore scientists are able to focus on their research activities with improved productivity and accessibility.

As proposed for a sub-project of the APAC Grid program [13], the goal of the chemistry portal is to provide the Australian computational chemistry and molecular science community a uniform and integrated user interface for scientists to access and execute distributed applications through a conventional web browser. It aims to improve the efficiency of utilization and to facilitate seamless use of different computer resources. In addition to some generic grid functionalities such as remote job submission, monitoring, and file transfer for all the applications supported, the chemistry Grid portal will also attempt to provide additional facilities for particular applications, such as input preparation, analysis and validation, result post-processing and visualization so that it will provide the molecular science community an appropriate e-Science platform to conduct their computational research from user management, to experiment composition and preparation, execution over grid resources, and finally results analysis and visualization.

In this paper, we present a description of our ongoing efforts to develop the chemistry Grid portal by utilizing core grid technologies for computational chemistry and molecular science applications. In Section 2, we provide an overview of applications and related requirements on the portal, following this with an overall of system architecture and design. Next, our implementations to meet the requirements are illustrated. Finally, we outline the current status of the project as well as future directions for the portal.

2. Applications and Requirements

2.1. Applications

The chemistry Grid portal will provide the molecular science sector in Australia with a uniform user interface for ready access to a variety of computational chemistry applications ranging from computational quantum chemistry (QC) to molecular dynamics (MD).

The QC software is used for molecular electronic structure and property calculations by solving the Schrödinger equation for the electrons. The QC software usually takes one input file which contains molecular definitions, atomic coordinate data and associated symmetry information, computing methods, basis sets for molecular integrals, and desired property of targeting molecules. Initial target QC software packages for the portal include Gaussian [6], Gamess [7], and ADF [14].

MD simulation can be used to obtain macro-properties of molecular systems by solving the equations of motion for the system of molecules. Usually the input files to a MD simulation need to contain following information, (1) information about how to run the simulation e.g. time-step, number of steps, temperature, pressure etc.; (2) a complete description of the molecular structure of the simulation system; (3) force field information defining the nature of molecular forces; (4) initial coordinates of atoms. The number and format of input files vary from one program to another depending on how the program organises its input data and on the kind of simulations to be conducted. The candidate MD software packages for the chemistry portal include Amber [15], Gromacs [16] and NAMD [10].

Both the QC and MD software packages are very widely used to perform complex molecular modelling, simulation and analysis for the design of novel materials and molecules in fields such as computer-aided chemistry, chemical engineering, pharmacy, and biochemistry.

2.2. Requirements

The primary requirement for the chemistry portal is to create a comprehensive problem-solving environment or e-Research environment for the Australian computational chemistry and molecular science community. The environment should provide a uniform and easy-to-use user interface for accessing input and output, running various application jobs on a variety of grid computer resources without logging onto those platforms, and the ability to transfer data among various resources. Through this portal interface,
users of the APAC Grid can use the physically distributed computer resources as if they were a single system, i.e., without having to consider differences in authentication mechanism, operating system, job management system, hardware, computational chemistry applications, and so on. In detail our chemistry portal attempts to meet the following requirements.

**Resource Registration:** The portal needs to provide the portal administrator with a way to register grid computer resources and application resources that are accessible to the APAC Grid users.

**Authentication and Single Sign-on:** The portal needs to provide the capability to authenticate a user using the user’s proxy certificate and the capability to store and keep track of the user’s credentials. Thereafter, access to Grid resources is delegated on their behalf with the credentials.

**Project Management:** The portal should allow users to create and conduct multiple computational experiments and have the ability to manage their jobs.

**Input Preparation:** The input preparation involves uploading input data files from a local computer, providing application specific graphic and/or text editors for modification and edit of the input files.

**Input Analysis and Resource Estimation:** The portal will need to provide a module to check the input data against possible errors, assess the input data such as calculation method and system size, and estimate resource requirements.

**Job Construction and Submission:** The portal will provide users with options to construct and run their jobs. To improve the usability the portal needs to provide automated job construction feature for the supported applications.

**Monitoring:** Users may want to monitor the progress of their jobs either immediately after the submission, or later if the jobs are long-lived. The portal also needs to allow users to choose a particular submitted job to monitor if they have several computations running simultaneously.

**Result Retrieve and Post-processing:** Provides facilities for storing, retrieving, downloading output results, utilities for transforming the output data from one program into the input data for another program, and visualisation tools for displaying molecular structures.

### 3. Architecture and Design

The primary designing principle of our portal is to maximise the reuse of existing software components, tools and services provided by open source portal frameworks for generic tasks, while focusing our efforts on addressing application specific components, interfaces and services. In addition to the requirements addressed in Section 2.2, easy to use, flexible and reusable are key features we considered in the development of the portal.

Portlets provide a standard technique for creating web applications for deployment in a portal container. A portlet represents an application software component that can create and load a particular piece of web content. It is self-contained and can be displayed in a sub-window of the portal. In recent years JSR 168 [17] has emerged as a standard for developing portlets, and has been widely adopted by the industry and within the portal community. GridSphere [18] is one of the open source JSR 168 compliant portlet containers. It is based on the popular Apache Tomcat [19] web application container. GridSphere portal framework supports portal development by providing a set of centralised core portlets and portlet services that provide the basic infrastructure required for developing and administering Web portals such as user management, access control and portlet subscription. A key feature of the design of GridSphere is that it builds upon the Web application repository (WAR) deployment model for the easy development and integration of “third-party portlets”. Today GridSphere has become one of the best open source toolkits available for developing Grid Portals and has been adopted as a standard toolkit for the development of Grid portals for the Australian APAC Grid Program.

Another key product from the GridSphere group is the GridPortlets framework that allows end-users to make use of Grid technologies. GridPortlets provides a collection of grid portlet services that offer a high-level API and model of the Grid for performing tasks on the Grid. GridPortlets makes use of the Java Commodity Grid (CoG) Kit [20] for performing many tasks on the Grid, including credential service for user authentication and user credential management, job service for submitting jobs to Globus Gatekeepers and monitoring the jobs, file service for transferring files with Grid Ftp and setting up GASS servers to collect job output. These portlet services can be used to easily Grid-enable any GridSphere web application. Using
GridSphere and GridPortlets as the core technologies, the design of the chemistry portal has a multi-layered architecture as depicted in Figure 1.

As shown on Figure 1, on the top level is the Client layer which represents the consumers of Grid computational portals, typically Web browsers. Typically, clients interact with the portal via HTML form elements and use HTTP/HTTPs to submit requests. The second layer is an Application Portals layer, which consists of application specific portal code itself such as our chemistry portal. The chemistry portal uses GridSphere as the portlet container and runs on a standard Web server, the Tomcat server, to process the client requests and the responses to those requests. The application portals can share portal-related user and account data and thereby make a single-login environment possible. These portals can also share file space, libraries, and other services provided by the next layer below, a Grid Portal Services Layer. The grid portal services layer, built on top of the low core Grid services, such as Globus, provides common services for application portals. Our application portal uses services and libraries provided by the GridPortlets to implement both informational and interactive portlet functionalities for clients. Below the grid portal services layer is the core Grid Services layer such as Globus which consists of those software components and services that are needed to handle requests being submitted by the grid portal services layer. The bottom layer is the Resource layer that provides the HPC computer resources for executing real computational jobs.

Figure 1. The architecture of chemistry Grid portal

Figure 2 depicts the design scheme of our portal system. At the core of the system is the web server machine which is the host of our application portal server. For the core system we have used GridSphere for our portlet container, Tomcat for the Web server, GridPortlets for the grid portal services, MyProxy for storing user credentials. The MyProxy server runs in a separate process and can be on a separate machine. On the core system we also installed the Globus 2.4 client which communicates with Globus servers on each Grid resource machines for performing Grid tasks on those machines. The core system is primarily Java based.

Figure 2. Design scheme of the chemistry Grid portal system

4. Implementation

Our chemistry portal consists of a set of portlets that are integrated as a comprehensive problem-solving environment for computational chemistry and molecular science. Each piece of functionality provided by the portal can be developed as a separate portlet or a small group of related portlets, usually supported by portlet services. In this section, we describe the implementation of our portal.

Application Registration: The chemistry grid portal will provide supports for a variety of applications on a number of HPC facilities. In order to make these applications available for scientists to access from the portal and to “portalize” other computational chemistry and biochemistry software, the applications are first needed to be registered into a database by a portal administrator. Our portal provides a mechanism for the registration and manipulation of computational applications available on the target machine through software resource entries in the persistent database. The registration of applications involves the development of two components, one application mapping manager portlet and one...
Application managing service. Gridsphere provides built-in support for Role Based Access Control (RBAC) which separates portal users into guests, users, administrators and super users. Using this supported feature the application mapping manager portlet provides an interface only accessible to portal administrators or super users and allows them to view already registered applications, register new application entries, update or delete an existing application entry. The application mapping manager, which utilises functionalities provided by the application managing service, consists of three form pages as shown in Figure 3 for performing the dedicated tasks.

The application managing service manipulates persistence for software resource entries and provides functional supports required by the application mapping manager portlet. GridSphere uses an object/relational persistence and query service for Java called Hibernate [21] as its default persistence service. The application managing service provides functionalities, such as, creating new application objects that contain information about the applications, storing the objects to the persistence, querying the persistence for retrieving, updating or deleting application objects from it. The application managing service also provides our portal capabilities to check and retrieve software resource entries on the particular target systems and to use this information for job construction.

Figure 3. Application registration and management

Authentication and Single Sign-on: The user authentication and single sign-on are achieved by providing a proxy retriever portlet as shown in Figure 4. The portlet can set up a connection with a user specified MyProxy server and retrieve the user’s proxy certificate from the server based on the username and password supplied by the user. Once the proxy is obtained the proxy retriever portlet utilises the credential manager service supplied from the Gridportlets to store the user proxy in the persistence and keep track of the user’s credentials. Other portlets or portlet services that require active credentials for performing operations for the given user on the Grid can use this service for credential retrieval. This achieves a single sign-on mechanism. A new user needs to apply for and obtain an APAC user certification from the APAC certificate authority (CA) in order to use the portal. To use a particular target system from the portal an entry for the user is needed to be added to grid-mapfile on the dedicated target system to map the subject of the user certificate to the user account.

Project Creation: A new computational task to be conducted by a user is treated as a new computational project. After retrieving the proxy credential, the user can use the project creator portlet as displayed in Figure 4 to create a new project by specifying a new project name, selecting an application and specifying the user preferred computing resource machine and the working directory. Once the new project is created, it becomes the one that is currently being worked on and then the application specified portlets will be presented to the user.

Input Processing: Inputs to computational chemistry applications have text format. New input files can be generated with standard text editor or graphic tools, such as the GaussView for Gaussian input generation. Once the inputs are generated and stored they can be reused for multiple times, often with
necessary modification or editing on them by changing some parameters and/or computational job configuration conditions.

The approach using text editor requires significant experience from the user. Any mistake in the file format and parameter choice results in an error and extends the time in which results will be obtained. The preparation of the input file requires knowledge of many keywords to describe atom coordinates, molecular symmetry, geometry optimisation, electronic properties, etc. This disadvantage can be removed by providing an application specific interface to assist users for input preparation. An existing input file is first uploaded and stored on to the portal server machine. If the user chose the option of View/Edit while uploading the input file, the configuration and edit form page will be presented to the user. An example for the Gamess portlet is shown in Figure 5. A user can modify the input by choosing appropriate parameters from a number of option lists. The content in the text editor area will be updated when the update change button is clicked. The content can also be modified by experienced users, e.g. the user can add new keywords and values. An input analyser module will be developed to check the input data and their interdependencies to prevent users from entering incorrect values. The input analyser will also be able to assess the calculation method, system size and resource requirements.

![Figure 5. Input file view and edit](image)

**Job Submission:** After performing validation checks against user input a job object is constructed automatically. In this case users have no need of specifying job details, such as stdout, stderr, executable full path, etc. A particular application may be installed on more than one computer resource centre perhaps in a different path or with a different executable name (or script name). In order to hide this complexity from users, the job submission portlet checks the availability of the software on sites and retrieves the executable information from the Hibernate persistence service by using the functionalities provided by the application managing service. The portal also generates other details for constructing the job, such as input and output information.

As part of the job preparation, currently the user needs to specify the computational resource requirements required by the job through the job submission portlet as shown in Figure 6. This includes the wall time, required memory size, job type (single or parallel) and the number of processors if parallel job type is selected. Thereafter, a job specification object will be generated by the portal and filled with all the job information. After the user clicks the submit button, the validated input files will be transferred from the portal server to the target system where the particular application will be executed and the job will be submitted to the system for execution as well.

![Figure 6. Job submission](image)

Because of the large variety of molecular sizes, calculation methods and computation configuration parameters, and properties to be calculated, it may be difficult for the users to estimate appropriate computational resource requirements. An on going effort is to develop an application specific resource estimator to compute resource allocation features based on expert knowledge about the calculations, the input parameters, the characteristics and/or the state of the computer resources. We expect in future, once the user specifies the job type and the requests for the number of processors, the job submission portlet will make a proper estimation on the resource requirements.
The chemistry grid portal also provides an alternative approach to submit a job. For some advanced users who want to submit their jobs using their own application codes or some particular applications not currently supported by the portal, they can do this by using the installed job submission portlet from the GridPortlets. However, to use this portlet the user needs to specify all details for construction of a job such as full executable path, stdout, stderr, arguments, and so on. The user also needs to manually transfer input files whenever required by the particular application.

**Job Monitoring and Result Retrieve:** Once the job is submitted the user can check job status, monitor execution, and retrieve output to the local machine. All these functions can be performed from the installed job submission portlet and file browser portlet provided by the GridPortlets. Using the job submission portlet, a user can obtain the job status information on all the jobs submitted by the user. The user is also allowed to hold or delete an executing job or create a new job. With the file browser portlet, a user can download files from remote computer resources, transfer files between two machines and manipulate files on the remote machine such as rename or delete a file. We are planning to explore other approaches for input/output storage and result retrieval. One option is to use database for storing the input files and the location of all output files for later retrieving and downloading. Another option is to use the SDSC Storage Resource Broker (SRB) [22] for input and output data storage and retrieval.

5. Conclusions and Future Work

Sets of core grid technologies and tools have been used to develop the chemistry Grid portal for computational chemistry and molecular science applications. The portal approach is very compatible with the Web/Grid services model and allows new grid interactivity and services to be easily aggregated into a portal interface. Using this approach and the GridSphere framework we have met a number of requirements set up for this project [13]. Currently our project is in its mid-way through its development. Some requirements still need to be met. This includes the input validation and resource estimation for different applications.

Currently the system uses portal services provided by GridPortlets for performing Grid tasks such as job submission and monitoring. We would like to add the capability to use Gridbus Broker [23] for these tasks since the Gridbus Broker provides additional features such as resource discovery and coordination for grid computing. For input and output data storage and retrieval we will explore other approaches such as archive data files in database and/or use the SDSC Storage Resource Broker for these tasks.

We will create more application specific portlets and integrate them into the chemistry Grid portal to provide supports for a variety of computational (bio) chemistry applications. It is clear that once these portlets are enabled for the supported applications we will envision more functionality that would further enhance computational science capabilities. In particular, interoperability between one application and another or even between one and more applications and other resources would be the next goal.

**References**

[15]. http://amber.scripps.edu/
[16]. http://www.gromacs.org/

[18]. http://www.gridsphere.org/gridsphere/gridsphere
[20]. http://www.globus.org/cog/
[21]. http://www.hibernate.org
[22]. http://www.npaci.edu/dice/srb
[23]. http://www.gridbus.org/broker/