Portal Framework for Computation within the eMinerals Project

R Tyer[1], M Calleja[2], R Bruin[2], C Chapman[3], M.T. Dove[2], R.J. Allan[1]

[1]*CCLRC – Daresbury Laboratory, Daresbury, Warrington, Cheshire, WA4 4AD, UK*
[2] *Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge, UK*
[3] *Department of Computer Science, UCL, Gower Street, London, WC1E 6BT, UK*

Abstract

This paper describes the eMinerals minigrid infrastructure and motivates the need for a portal based environment as an interface to this infrastructure. The paper then describes the ComputePortal developed by the project as a portal interface to the compute resources within the minigrid. Its architecture, functionality and integration with other portals in use by the project are discussed. Finally an example use case is outlined which shows the potential functionality available via an integrated portal environment.

1. eMinerals Project

The 'Environment from the molecular level' (eMinerals) project [1] is a NERC funded eScience pilot project focused on fundamental science problems associated with key environmental issues such as nuclear waste storage and pollution. The project includes collaborators from Bath University, Cambridge University, CCLRC, the Royal Institution, UCL and Reading University.

The scientific aims of the project are primarily focused on the use of computer simulations performed on systems with molecular length and time scales in order to gain insight into key environmental issues.

Aside from the scientific issues, this research is challenging both in terms of the computational power required to tackle realistic system sizes with the required accuracy and the data management issues associated with handling large amounts of data over a distributed virtual organisation. The computational requirements of the project vary significantly depending primarily on both the simulation code and physical system in question but both HPC and HTPC resource are required. Hence the use of eScience and Grid technologies to develop an integrated infrastructure and working environment provides enticing opportunities to facilitate and enhance the scientific work performed by the project.

2. eMinerals Minigrid

The project has investigated and deployed a number of different pieces of eScience technology in order to build the eMinerals Minigrid [2], an integrated grid infrastructure combining both compute and data resources. Along with the physical resources, the minigrid includes a suite of client tools and portals which act as brokers to the compute and data servers. A schematic view of the minigrid infrastructure is presented in Figure 1.

2.1 Compute Resources

The compute infrastructure used by the project includes both HPC and HTPC resources as both are required by different computational codes and classes of problems. The HPC resources on the minigrid include three 16 node Beowulf cluster machines and a 24 node IBM pSeries parallel computer. In addition the project makes use of other HPC facilities such as the NGS machines and HPCx. These machines run different batch queuing systems and have been integrated into the minigrid using a mixture of Globus Toolkit 2 and 3 [3].

The project has also made heavy use of Condor [4] for HTPC calculations and has been central to the creation of a large (currently around 950 node) Condor pool at UCL. In addition, there is a smaller, but extremely heterogeneous, Condor pool at the Cambridge node. Again Globus Toolkit 2 has been used to integrate these resources in the grid infrastructure.

2.2 Data Grid

A data grid has been built using the Storage Resource Broker (SRB) [5] developed by San Diego Supercomputing (SDSC). The SRB allows users to organise files held on distributed systems into a single virtual filesystem thereby abstracting the user from the physical location and underlying storage media of the files. Each of the compute resources owned by the project also hosts a SRB storage vault. These storage vaults form a distributed and heterogeneous data grid which currently provides around 3Tb of storage capability for the project.
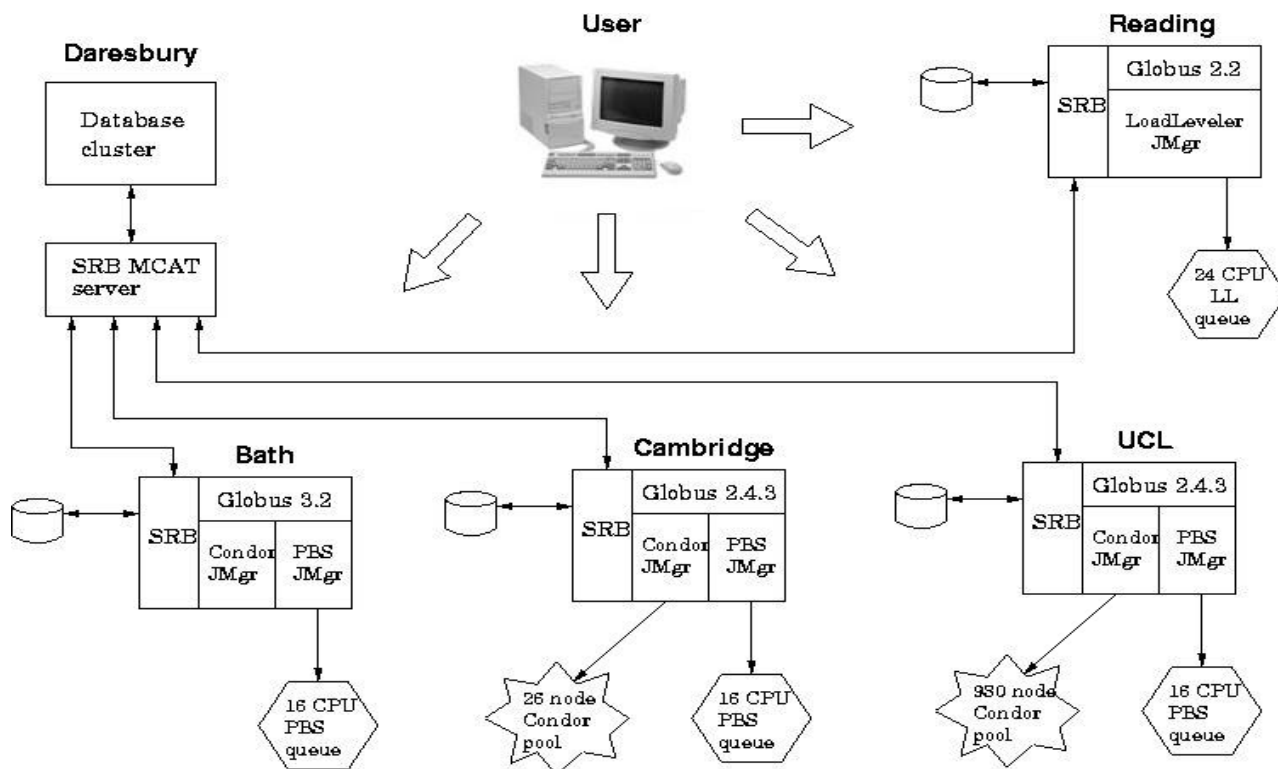
Figure 1: Schematic representation of the compute and data resources within the eMinerals Minigrid Infrastructure.

In addition to the data grid, there is a metadata database hosted on the database cluster at the Daresbury node. This holds metadata describing the data held in the SRB data grid.

2.3 Portals for Data Managagement

The project is making use of several portal interfaces in order to manage both scientific data and metadata. The MySRB portal [6] developed by SDSC is being used as a client to the project's SRB based data grid. The CCLRC Data Portal [7] is used for data discovery as it allows the simultaneous searching of multiple distributed heterogeneous metadata catalogues. In addition, the project is using the CCLRC Metadata Editor [8], which is a portal front end to metadata databases that allows users to insert studies and associate and annotate datasets held in SRB with these studies. These studies are then labelled with various topic hierarchies, which then allows the studies and datasets to be retrieved by the Data Portal during subsequent searches.

2.4 Client Tools

Currently a variety of client tools are needed in order to utilise all the functionality of the minigrid. These include:

1. SRB client tools – the most significant of these is the Scommands which are command line SRB client tools. These provide a highly efficient method of pulling input files onto

compute resources before calculation and pushing the output datasets into the data management infrastructure after simulations have been completed.
2. Condor client tools – these are required if the user wants to submit jobs to Condor directly, or use the Condor workflow engine, DAGMan. In the latter scenario Condor-G would be used to interface to the Globus gatekeepers and jobmanagers.
3. Globus client tools – these are the command line utilities which are included with the Globus Toolkits and allow users to submit jobs from the command line.

2.5 Problems with current client environment

In developing and deploying the eMinerals minigrid several obstacles have been encountered which have complicated uptake by the users.

The first is the learning curve associated with all the different pieces of middleware and client tools. It is perhaps unrealistic to expect users who are primarily focused on scientific problems to have to learn how to use all the individual client tools necessary to utilise the minigrid functionality. This is especially true when the core middleware and client tools themselves are still being developed and can be subject to change between versions. In any event, every effort must be made to provide user friendly and intuitive client tools in order to facilitate access to the grid infrastructure.

Scalability problems have been encountered with deploying and maintaining the entire suite of client tools on users desktop machines. This has currently been avoided by having designated nodes within the minigrid on which the client tools are installed and maintained on behalf of the users.

Although the use of the minigrid infrastructure is providing real benefit and added value to the scientists, more effort is required in order to make its use more seamless and automated.

Further details about the eMinerals minigrid and its evolution into a production infrastructure can be found in Reference [2].

3. Portal Technology

Portal technology offers potential solutions to the problems outlined in the previous section.

It is possible to construct highly usable and intuitive interfaces using portal technology, which abstract the users significantly from the underlying middleware and client tools. This abstraction allows the users far more focus on the functionality offered by the grid infrastructure and how it may be harnessed to solve the scientific problem they are concerned with. In addition it is significantly easier to train a user to use a portal interface compared with command line tools.

Clearly the use of web portals entirely avoids the client problem, as the only requirement for use of the grid infrastructure is the ubiquitous web browser. This is especially important for the highly mobile world academic community.

Perhaps the most exciting potential for portal technology is its use as an integration layer above the grid middleware in order to build a more seamless environment for the user. In general most of the middleware and client tools used in the eMinerals minigrid were developed independently of each other and with no thought as to how they may be integrated into a single infrastructure. This is especially true in the case of compute and data grids. By interacting with both compute and data resources via web portals it is possible to combine the functionality of both and realise much more complete workflows and use cases than is easily attainable using client tools directly.

4. ComputePortal

The eMinerals project has developed the ComputePortal as a portal interface to the compute resources on the eMinerals minigrid. In addition to interfacing with the HPC and HTPC resources on the minigrid, the ComputePortal is also being integrated with the other portal tools already deployed within the mingrid. Along with the DataPortal, Metadata Editor and MySRB, the ComputePortal forms the eMinerals Portal which provides a single portal environment from which users can manage all their computational simulations and associated data/metadata management.

The ComputePortal evolved from the initial work on the HPC Portal [9] carried out by the CCLRC Grid Technology Group. This portal was one of the first to demonstrate the possibilities of using portal technology in order to act as a broker to compute resources using Globus middleware. As this portal primarily used the Globus C API in order to interact with Globus resources, it was felt that a new portal based on the Java Cog kit [10] and building on the experiences and lessons learned with the HPC Portal would be the best approach.

4.1 Portal Architecture and Technologies Employed

Whilst the innovative work on portlets and JSR 168 [11] is felt to be the long-term future of portal technology, the decision was taken to employ a far more simplistic and lightweight approach for the ComputePortal. This was primarily motivated by the resources available within the eMinerals project to develop the ComputePortal and the desire to focus on providing functionality to the users rather than on the technology itself.

One of the key design rules and lessons learned from the earlier HPCPortal project was the need for clear separation between the presentation and application layers within the portal. Not only does this provide a more transparent and maintainable framework but it will simplify refactoring to a web or grid services based architecture if this is required at a later date.

The architecture of the ComputePortal is hierarchical as shown in Figure 2.
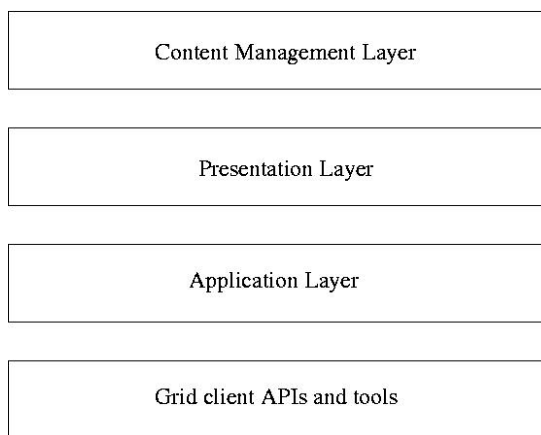

Figure 2: Schematic representation of hierarchical architecture of ComputePortal.

The base of the portal stack is composed of the client tools and libraries required in order to interact with the grid resources and middleware. The Java Cog kit is used heavily in this layer.

Above this layer are the presentation and application layers which are divided into modules of discrete functionality, which are described in Section 4.2. The code in the application layer is almost

exclusively written in Java. Java was chosen principally because of the Java Cog kit, which is the only Cog kit to provide a native implementation of the Globus client libraries.

The modules in the presentation layer are written using PHP. At the design phase it was unclear how much code from the HPCPortal project would be reused within the ComputePortal so PHP was appealing as it could bind to code written in a variety of languages. Most importantly there is a Java extension to PHP which provides an extremely clean binding between the presentation and application layers. Using this extension it is possible to treat Java objects as native PHP objects, call their methods and catch their exceptions. Hence the combination of PHP and Java allows for complete separation between the presentation and application layers without any significant loss of functionality.

The top layer of the portal stack is a content management layer which uses the PHP Nuke [12] content management system (CMS). The motivation for using a CMS system is to minimise dealing with layout and access control issues within the individual presentation layer modules. Within PHP Nuke these are handled centrally by the portal framework, which uses a local MySQL database in order to track the presentation and modules available to particular uses.

4.2 Portal Modules

The ComputePortal is modular within the presentation and application layers. Key modules include:

- Authentication – The aim of this module is to authenticate the user with the PHP Nuke framework and to obtain a valid delegate proxy for the user. In addition, this module handles single sign on with the other portals comprising the eMinerals portal which is described further in Section 5. The steps involved in login onto the portal are:
  1. User enters username and passphrase which are used to obtain their credentials from a MyProxy repository [13].
  2. If valid credentials are obtained, then the DN is used to look up the user's details in a local database which are then used to log onto the PHP Nuke framework and initialise the portal.
- Job Submission – This module is concerned with submitting computational jobs to the variety of HPC and HTC resources within the minigrid whilst abstracting the users as much as possible from the underlying queuing systems. The user requirements are used to construct the corresponding RSL string. If the requested resource has a correctly configured jobmanager then this RSL is submitted directly to the machine. If this is not the case then the RSL is used to construct an appropriate job submission script which is then transfered via gridFTP to the

resource and submitted via the fork() jobmanager. The job ID's of submitted jobs are stored in a local database which is used by the Job Monitoring module.

- Data Transfer – The role of the data transfer module is to transfer input data to and output data from the compute resources. The input data may come directly from the users client machine via HTTP, from SRB or be transferred using metadata obtained from searching the DataPortal. In the latter two cases the portal transfers the data directly assuming appropriate firewall configuration and client tools are in place for the remote resource. Alternatively the portal can act as an intermediary by initially pulling the data on the portal server before pushing it onto the remote resource if these requirements are not met.
- Resource Discovery – This module is to allow the users to dynamically discover, monitor and query the status of resources on the eMinerals minigrid. It allows the user to build up LDAP filter strings with which to query GIIS servers so that users can find appropriate resources. Although currently the list of resources used by members of the project is relatively static, dynamic resource discovery is a key concept within the grid vision and the functionality of this module will undoubtedly have to be expanded in the future.
- Job Monitoring – This module allows users to monitor the status of their jobs within the individual queuing systems and initiate post processing tasks such as uploading results files to the SRB. The portal caches the Globus job ID's from the Job Submission module and periodically polls the Globus jobmanagers in order to determine the status of the users jobs. When the user logs back onto the portal, it is possible to view all the job states of their recent jobs.
- Metadata / Logging – This module provides an API for all modules to use for all logging and interaction with the local database for handling state information. This module will be further developed in order to harvest metadata at each state of the computation simulation process to automate the metadata handling process as much as possible. It is essential result datasets are annotated with relevant metadata in order to enable this data to be discovered again at a future date. This metadata also gives the data longevity by recording the exact nature and context of its generation.

5. Integration with other portals

The ComputePortal was intended to be integrated with the other portals deployed on the eMinerals minigrid in order to form the eMinerals portal, a
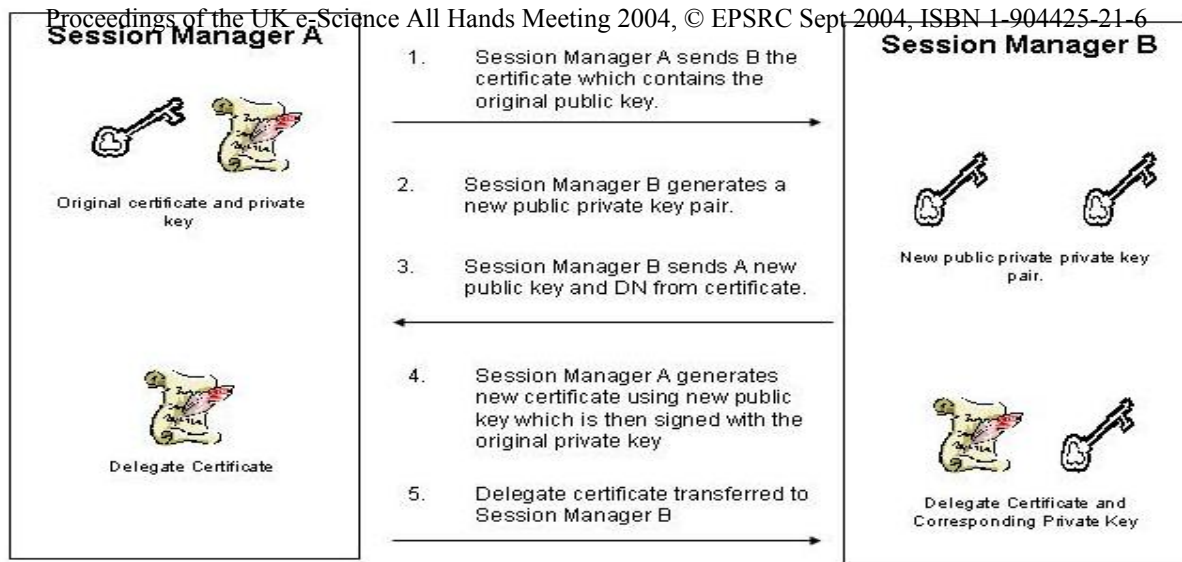
Figure 3: Transfer of user credentials between portal session managers using GSI delegation.

seamless and integrated environment from a user perspective.

The most important step in integrating the different portals is to enable single sign on between them. This is achieved using web services and GSI delegation. When a user attempts to move from one portal to another, the portal they are currently logged onto sends the current session ID in the HTTP header. When the new portal receives a session ID which does not correspond to one of its own sessions, it queries the session managers of the other portals within the minigrid to see if this session ID corresponds to one of their valid sessions. If this is the case then the user's delegate credentials are transferred between the session managers using GSI delegation over web services as shown in Figure 3. The portal can then continue its logon procedure from the point where the user credential has been obtained from the MyProxy repository.

Having established a mechanism for transferring session ID's and credentials between portals, it is relatively straight forward for the ComputePortal to integrate with both the Data Portal and Metadata Manager by exploiting their web services architectures.

6. Example Use Case

From a data centric perspective the process of scientific simulation can be represented by the workflow shown in Figure 4. It is useful to examine this workflow as an example use case of the ComputePortal as it shows the benefit of developing an integrated environment for both computational simulation and data/metadata management. The steps involved in such a use case are:

1. The user logs onto the eMinerals instance of the Data Portal and searches various metadata databases looking for data of interest, e.g. a crystal structure and pseudopotential to use as the initial input for a calculation. The metadata describing this data and including its URI is stored in the user's personal shopping cart within the Data Portal.

2. The user then seamlessly transfers to the ComputePortal, which uses the GSI delegation mechanism over web services in order to transfer the user's credentials and session from the Data Portal.

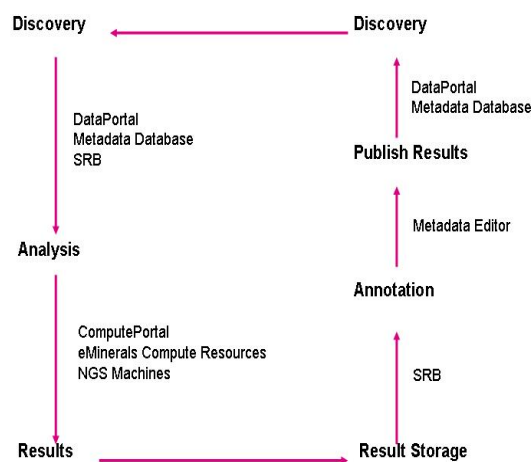3. Having logged onto the ComputePortal, the user uses the Resource Discovery

Figure 4: Data lifecycle showing data centric view of scientific simulations.

functionality in order to find a resource appropriate for the calculation they wish to perform.

4. Using the Data Transfer module the user then transfers the input data to the resource using

664

the metadata imported from the Data Portal's shopping cart web service.

5. The user then uses the Job Submission module to submit a job to the remote resource in order to perform some analysis on the input datasets.
6. Using the Job Monitoring functionality the user is able to monitor the progress of their calculation.
7. When the simulation has completed successfully, the user can transfer the output datasets to the SRB using the Data Transfer module.
8. The user would then exploit the single sign on to transfer to the Metadata Manager where they could either create a new study to describe this work or open an existing study. The result data sets would then be annotated and associated with this study.

This last step completes the workflow as the output datasets are now stored within the data grid and labelled with appropriate metadata to enable them to be retrieved during future searches.

7. Future Work

There are many potential avenues of future development. Particular areas of interest are developing a more automated mechanism of metadata handling and investigating more complex workflow enactment using DAGMan and Condor-G.

However the main focus of any future development has to be driven by user feedback with additional functionality and enhancements being added as required by the users themselves.

8. References

[1] http://www.eminerals.org; M. Dove et al. "Environment from the molecular level: an escience testbed project", Proceedings of All Hands Meeting (2003).

[2] M. Calleja et al. "Grid tool integration within the eMinerals project", Proceedings of All Hand Meeting (2004), R. Tyer et al. "Building the eMinerals Minigrid", Proceedings of All Hands Meeting (2003).

[3] http://www.globus.org; I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructre Toolkit", Intl. J. Supercomputing Applications, 11(2), 115-128 (1997).

[4] http://www.cs.wisc.edu/condor; P. Wilson, et al. "A Grid approach to Environmental Molecular Simulations: Deployment and Use of Condor Pools within the eMinerals Minigrid". Proceedings of GGF 10 (2004).

[5] http://www.npaci.edu/DICE/SRB; M. Doherty et al. "SRB in Action", Proceedings of All Hands Meeting (2003).

[6] https://srb.npaci.edu/mySRB2v7.shtml

[7] G. Drinkwater et al. "The CCLRC Data Portal", Proceedings of All Hands Meeting (2003).

[8] http://www.e-science.clrc.ac.uk/web/projects/scientific_metadatamgnt

[9] http://tyne.dl.ac.uk/HPCPortal/hpcportal.shtml

[10] http://www-unix.globus.org/cog/java

[11] http://www.jcp.org/en/jsr/detail?id=168

[12] http://phpnuke.org/

[13] http:// myproxy.ncsa.uiuc.edu