

Building and managing the *e*Minerals clusters - a case study in grid-enabled cluster operation.

Richard P Bruin, Mark Calleja*, Matthew G Tucker† and Martin T Dove
Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge, CB2 3EQ

March 24, 2005

Abstract

The need to perform large numbers of complex simulation computations forms a large part of the NERC funded *e*Minerals project [1, 2]. As such it has been necessary to provide extra computing power for the project through the production and use of three compute clusters, each of which are integrated into the *e*Minerals minigrid [3, 4, 5].

The integration of these clusters into a single grid infrastructure means that each of them may be accessed in a uniform manner, consistent with the other machines within the minigrid. This paper discusses the configuration, monitoring, management and usage of the compute clusters in this innovative manner where users need not even be given user accounts on the machines. Specific attention will be paid to the manner in which users access and use the clusters and how the data generated by users is kept secure from hardware failure or accidental deletion.

1 Introduction

The *e*Minerals project is one of the UK *e*Science testbed projects. Its aim is to create a mid-size multi-institution collaborative infrastructure to encourage more diverse interoperability between project members through the use of grid technology.

The project team consists of both computer sci-

entists and physical scientists (chemists, physicists, mineralogists and environmental scientists) who work together to create realistic molecular-scale computer simulations of important scientific reactions and processes. These simulations have significant computational requirements, and it is necessary for the *e*Minerals project to leverage not only its own clusters but also other contributed resources which are all connected using the internet to form a grid – the *e*Minerals minigrid.

This paper discusses the structure of the *e*Minerals project clusters, including issues such as their installation and configuration, and how the clusters are monitored and managed within a distributed grid environment. We will also discuss the methods we use to access the clusters and to backup the data created by our simulations, both of which have been developed specifically by *e*Minerals project members.

To provide the context for this work, we show the current structure of the *e*Minerals minigrid and all of the resources available to the *e*Minerals project members in Figure 1. The clusters referred to within this paper are labeled Lake@Cambridge, Lake@UCL and Lake@Bath in the figure. The *e*Minerals minigrid also consists of other compute resources, including desktop grids built using the Condor [6] system. The minigrid handles data management using the Storage Resource Broker (SRB) [7, 8], as described below. The integration of the compute and data components is critical to the operation of the clusters within our grid environment.

*Present Address: Cambridge *e*Science Centre, Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WA

†Present Address: ISIS Facility, Rutherford Appleton Laboratory, Chilton, Didcot, OX11 0QX

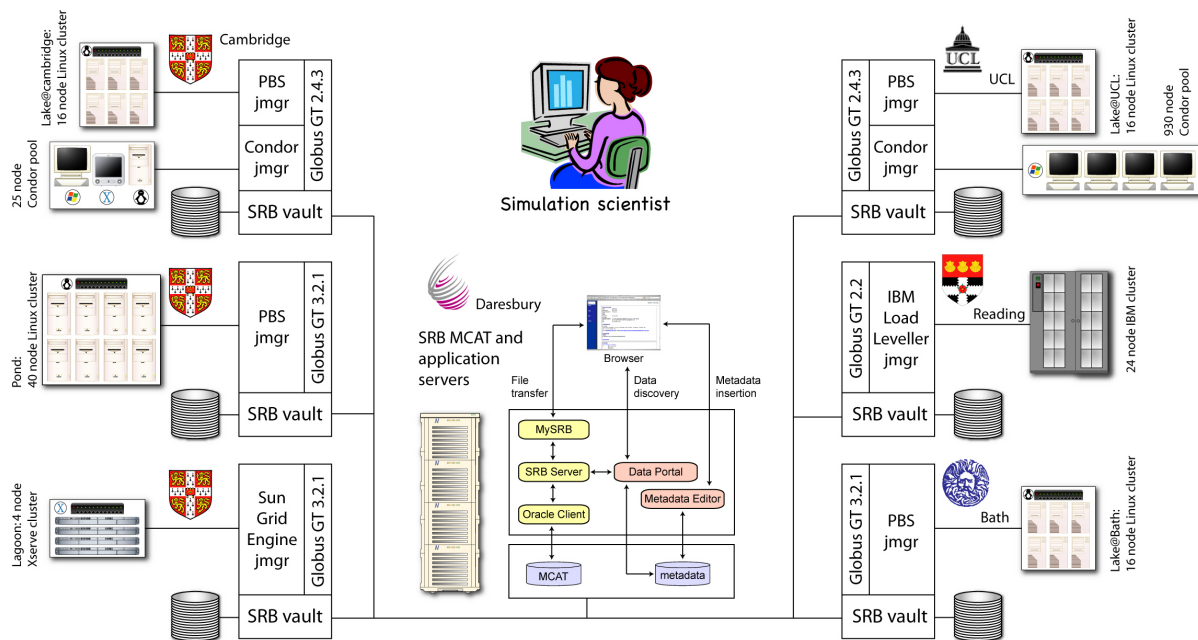


Figure 1: Representation of the eMinerals minigrid. The compute infrastructure consists of the three PC/Linux clusters discussed in this paper, namely Lake@cambridge, Lake@UCL and Lake@Bath, together with a similar cluster (Pond) adapted in the light of our experience with the Lake clusters, a small high-memory Apple Xserve cluster (Lagoon), an IBM pSeries parallel computer, and Condor pools in Cambridge and UCL. Most of these resources contain SRB vaults, and the data component of the minigrid is managed by the MCAT server in Daresbury. Access to the compute resources is through the Globus and Condor/Condor-G client tools, with security handled through the use of X.509 digital certificate. Users are prohibited from accessing most resources via traditional login procedures such as ssh. Similarly sftp is not used by users to extract files, with data access being provided by the SRB.

2 Cluster structure and operation within the grid environment

2.1 Cluster structure and basic installation

Each of the three clusters is assembled from standard commodity components. They consist of sixteen compute nodes, each with a single Pentium IV processor, 2 GB of RAM and a 120 GB system hard drive. The nodes are connected together using gigabit Ethernet and are controlled by a separate master node of identical specifications. The master node also contains a RAID array to provide 720 GB of data storage, with the capability of further expansion later when needed. This storage is only used for users' files, and is not used for operating system files.

Most of the codes users run on the *eMinerals* clusters do not require constant inter-node communication, and indeed some of the applications are embarrassingly parallel and thus require only book-keeping communications at the start and end of runs. The main requirement is to the speed/bandwidth of bulk data transfer rather than the need for ultra-low latency, and thus the appropriate inter-node network communications is provided by gigabit Ethernet. This has a slight latency penalty as compared with 100 Mbit Ethernet, which was minimised by careful optimisation of the Linux kernel TCP/IP configuration. An average latency of approximately 100 μ s was achieved with 64-byte packets, which was deemed to be acceptable.¹

At the time of writing, the clusters each run Mandrake Linux 9.2 [9] and Oscar 2.3.1. Oscar [10] provides all the necessary cluster installation tools, together with tools for all the basic operations such as job scheduling and basic monitoring via the Portable Batch System (PBS) [11] and Maui job scheduler [12].

Most of the community of users use code written in Fortran, and so the clusters were provided

with several Fortran compilers and MPI (Message Passing Interface) libraries (both MPICH and LAM implementations). This was driven by the fact that the experience of our users is that the performance of different codes is affected by the choice of compiler and MPI implementation. Finally, we also installed various Maths Libraries including ScaLAPAC, BLACS, etc., each of which was installed with their standard configuration.

The three clusters are located in Cambridge, UCL and Bath. Routine day-to-day maintenance is performed remotely from Cambridge with extra assistance provided by project members local to the cluster.

2.2 Middleware for user access within the minigrid environment

Running and managing computations within a grid environment is quite unlike running jobs on a standalone cluster. We note that Oscar is primarily designed for use in standalone clusters which users access through the use of ssh or some other direct connection mechanism. This access model does not directly fit with the grid model employed by the *eMinerals* project, since direct access is not permitted². Our approach has been to use the Globus Toolkit (herein referred to as Globus) [13]. Within the *eMinerals* project we use both versions 2.4.3 and 3.2.1 of the Globus Toolkit; however, we mostly use only the functionality of the v2 toolkit, which has been propagated into v3. At the time of writing, we are anticipating the release of Globus toolkit v4 as a replacement for v3, and will upgrade all clusters to v4 when it is available.

The key thing about Globus is that it provides access to the clusters through the use of X.509 certificate driven authentication without the users needing to directly log in. Jobs can be submitted from the users' own desktop computers using the Globus client tools, or from designated submit machines running the Globus toolkit. User accounts have been created on each cluster, but users do not

¹The view was taken that these clusters were not designed to run as mini-high-performance machines, but that they should instead be optimised for high-throughput computations. The *eMinerals* users have access to other resources for their high-performance requirements.

²Since the *eMinerals* minigrid is a prototype grid infrastructure, it is essential to enforce rules such as access mechanisms which will reflect larger-scale grid structures of the future. However, we do allow direct access to the Cambridge cluster via gssh for code developers to compile and test their applications.

actually know (and are unable to change) the passwords associated with their user accounts on the cluster machines since they are not needed when using certificate authentication such as Globus provides.

Once jobs have been submitted to a cluster via Globus, they are passed onto the cluster's own internal scheduling system which in the case of the *eMinerals* clusters is provided by PBS for internal job submission and Maui which handles the scheduling of jobs until required resources are available.

We did find the use of several different MPI libraries leads to difficulties within the Globus implementation. As a result it was necessary to extend the standard installation of Globus to enable the user to specify which MPI installation should be used at runtime. By default the Globus PBS interface uses a default MPI implementation as specified by an environment variable on the cluster being used. We extended the perl implementation of the interface between the Globus job specification and the PBS job specification so that it can handle an extra value in the Globus job's RSL (Resource Specification Language) specifying which MPI implementation to use. This extra RSL tag is then passed to the PBS scheduler (Maui) from Globus and the specified MPI implementation is then used. If the user does not specify the extra tag, then the default MPI implementation is used, as with the standard Globus behaviour.

2.3 Data management within the minigrid environment

Since users are not able to directly log in to the grid clusters, what users can do with their output files is restricted. If users know the files being produced, they can use the gridftp tool provided with the Globus toolkit, but there are cases where users do not know exact details of files being produced. Overall, we decided that most users needs would be best met by employing a distributed data management infrastructure, and the best product for this is the SRB. The SRB provides a single logical file structure even though data are distributed over several locations, and the user sees a single point of access to this file structure. The geographical location of any file is reduced to a mere file attribute. The SRB has a central metadata cat-

alogue (MCAT) server that maintains information about all files within the SRB. In the case of the *eMinerals* project, the MCAT server is located at the Daresbury Laboratory (see Figure 1). In addition to the central MCAT server, the SRB system requires a set of storage vaults. One vault has been setup on each of the three clusters (the 720 GB raid arrays referred to above). Moreover the SRB client tools are installed on each cluster.

The operational approach is that users manage their data on the SRB. They put their files onto the SRB before beginning their calculations. Their jobs download the relevant files from the SRB prior to running the main application code, and place all output files into the SRB at the end of the job. It is possible to use wildcard file specifications, which means that users do not need to know details of files produced within a run in advance.

One advantage of this way of working is that the job lifecycle process results in an archive of the entire process being maintained on the SRB. This is particularly useful for collaborative workers.

2.4 User access

As discussed above, users do not have login access to the clusters within the minigrid, and need to use both the Globus and SRB tools to manage job submission and data handling. With this infrastructure, the main challenge concerns usability issues, particularly since the Globus tools are not straightforward to work with.

Our experience within the *eMinerals* project has been that users are able to cope much better with the interface to the Condor desktop-grid infrastructure. Fortunately Condor and Globus have been implemented within the Condor-G toolkit [14], which provides a Condor-like interface to the Globus toolkit. We have used Condor-G to develop a script-based interface to our minigrid and SRB. This handles the job submission and job queueing, and the data transfer between the clusters and the SRB. Specifically, the various stages in a job life cycle are handled using Condor's DAGMan workflow tool [6], as described in more detail elsewhere [4]. The user only needs to modify a very small script, with a 'hidden' larger general script managing the setup and execution of all the processes within a job. The small user-supplied script which closely resembles a standard condor job submission

script, allows the user to provide details of file locations within the SRB. The executable image can be transferred to the minigrid from the user's desktop or from the SRB.

Subsequent to the development of this command line interface we have developed a web-based portal interface to the *eMinerals* minigrid [15], which allows project members access to all of the project's compute resources, including job submission, monitoring and data retrieval. This portal has two key advantages over the script-based interface. Firstly, it does not require the installation of client tools on the user's machine, and secondly it avoids problems experienced with communication through firewalls between project sites³.

3 System Monitoring

Since the *eMinerals* minigrid is comprised of a large number of compute resources, it became apparent that a single tool to monitor them all automatically would be required. We developed a monitoring tool comprised of two distinct sections. The first section is the actual monitoring system, which periodically queries each of the compute resources for their queue and disk usage statistics. This querying of the minigrid resources works by submitting Globus jobs to each of the different resources (both the clusters described in this paper and other machines within the minigrid as shown in Figure 1). These jobs run queue querying and disk usage commands, returning the output to the querying machine. The output from these commands is then parsed and all useful information stored within a database. This process is performed every 30 minutes giving all users very recent if not necessarily completely current information. The second section of the tool displays the data collected from the first section, making it simple for any *eMinerals* project member to quickly check the status of a machine on the minigrid to ensure that they can target their jobs at any unused resources. On accessing this tool, the

³In practice firewall problems can be fixed by coordination with the relevant firewall administrators. However, firewall policies are outside the control of the *eMinerals* project team, and we have found that they are liable to change without notice, causing part of the minigrid to become inaccessible for the time taken for the implications of policy changes to be reversed.

database is queried, and an image similar to that shown in Figure 2 is displayed in the web browser.

The next version of the monitoring tool will include email notification to system administrators should any compute nodes stop responding to queries and increased amounts of queue statistics including actual queue status command output so that users can retrieve more information regarding the state of particular queue resources.

4 System Maintenance and Management

Although the clusters are reliable and do not require much outside intervention, occasionally a compute node will stop responding to connection requests from its master node. In this case project members at the cluster's location simply reboot the node. If this fails to trigger the node to resume communications with the master node then a clean network reinstall is performed from system images stored on the master node. Note that the install images are constantly maintained so that a reinstall results in a fully functioning installation and no further steps are required to configure a node once a reinstall has occurred.

In the majority of cases these steps will reset the misbehaving node into a state in which it can again accept jobs from users and can continue as normal. However occasionally they experience hardware failure, in which case project members who have more system administration experience will troubleshoot and fix the node, however as much as possible this troubleshooting will be performed remotely with the assistance of the local project members.

We have also provided a redundant system disk in the master node allowing the system to be swapped over to the backup configuration in a matter of minutes even by a project member without computing training. This returns the system to an operational state, allowing other project members to connect to the system and perform troubleshooting tasks from remote locations, independent of the project members within the system's location. We did not, however, implement this using a raid system, in order that the backup system disk will remain in a stable state and that any system corrup-

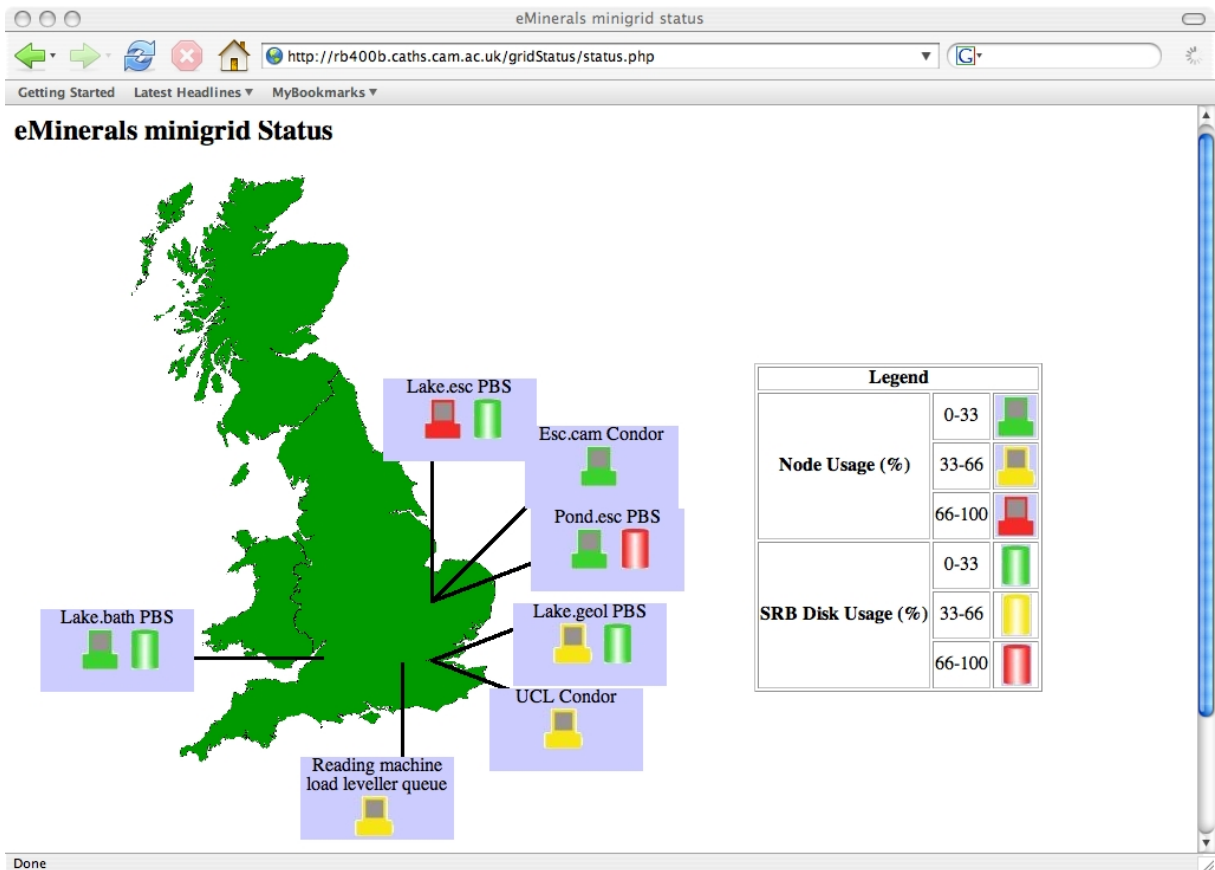


Figure 2: Representation of the status of the eMinerals minigrid as given to a user on checking with the system monitoring tool discussed in the text. Each of the blue boxes represents a compute resource on the minigrid. The colour of the computer image gives a simple representation of how heavily the machine is being used. The display and colour of storage cylinders represents the existence of an SRB storage vault and the approximate degree of utilisation of the vault. This page can be found at <http://www.eminerals.org/gridStatus/>.

tion is not propagated.

5 Backup System

The backup system employed on the eMinerals clusters was written especially for the project by project members and makes unique use of the otherwise unused hard drive space on the compute nodes. Every night an archive is made of every file that each user has changed since the previous day. Then once a week, every file in the homespace of each cluster is archived and compressed into several files, one or more for each user of the system depending on the amount of data to be backed up.

Each compressed archive is stored on the relatively-unused hard drive of one of the cluster nodes. Each week, the node to which the backup is stored is changed in a round robin fashion. In this way we can keep a complete backup of all users' files up to a maximum of sixteen weeks before we start to overwrite previous backups (depending on the total size of each week's backup). All of the archives containing files that users have changed that week are then copied over to the compute node which already stores the previous week's complete backup. The node to which any further backups are to be made is then incremented, such that the next compute node will store the next week's backup files and the complete backup performed at the same time. This means that the files backed up to any one node can be combined and manipulated to produce any of the files stored on the next node in the backup sequence.

There are however two possible drawbacks to the use of the backup system as described above. One is increased network communications due to the same communications channel being used for both inter-node communication and for moving the backed up data. However this has not been a problem with our clusters due to the largely embarrassingly parallel nature of our calculations. Secondly, there is no protection from physical risks such as fire. Our clusters do not fully address either of these problems because we believe them to be of relatively low risk when applied to our situation.

6 Discussion and conclusions

The main point of this article is to describe the implementation of high-throughput cluster computing within a grid environment, in which users are expected to access resources using grid middleware tools rather than through direct logins. The most novel feature of this work has been the integration of the compute and data grids through the use of the SRB for data management. As described above, the data management and distributed data grid components have been integrated into the clusters in both the hardware and software layers. We do not know of previous integration of this type. In addition to describing the cluster setup, we have discussed several usability issues, and described how we have developed interfaces and methods to enable the simulation scientists to make easy use of the clusters within the grid environment.

The other key message of this article, which we have only commented on in passing, is that this grid infrastructure has not proved to be difficult to maintain. Running three clusters over three sites has not required dedicated staff effort, and it has been possible to use untrained team members to help perform occasional troubleshooting tasks.

Our main conclusion, therefore, is that it is practical for consortia to set up and manage clusters within an integrated compute and data grid environment without requiring large investments in support manpower or user training.

7 Acknowledgements

We are grateful to NERC for funding the eMinerals project, and to CEH for additional financial support for Richard Bruin.

8 About the authors

Richard Bruin is a postgraduate student in the Department of Earth Sciences, University of Cambridge. He graduated with a degree in Computer Software Theory from the University of Bath. Contact him at rbru03@esc.cam.ac.uk and http://www.eminerals.org/people/richard_bruin.html
Mark Calleja works in the Cambridge eScience Centre, but was a post-doctoral worker in the Department of Earth Sciences in the University

of Cambridge when this work was undertaken. His degree and PhD were obtained from the Physics Department at the University of Kent. Contact him at M.Calleja@damtp.cam.ac.uk and <http://www.escience.cam.ac.uk/~mcal00/>.

Matt G Tucker now works at the ISIS Neutron Scattering Facility (Oxfordshire, UK), but held a post-doctoral position in the Department of Earth Sciences in the University of Cambridge when this work was undertaken. He is a physics graduate from Cardiff, and obtained his PhD in condensed matter physics from the University of Kent. Contact him at matt.tucker@physics.org.

Martin T Dove is Professor of Computational Mineral Physics in the Department of Earth Sciences, University of Cambridge, and Director of the National Institute for Environmental eScience. His PhD was in condensed matter physics, and his research in recent years has been concerned with applying a range of computational simulation methods to explore the behaviour of materials at a molecular level. Contact him at martin@esc.cam.ac.uk and www.esc.cam.ac.uk/~astaff/dove.

References

- [1] MT Dove, M Calleja, J Wakelin, K Trachenko, G Ferlat, P Murray-Rust, NH de Leeuw, Z Du, GD Price, PB Wilson, JP Brodholt, M Alfredsson, A Marmier, RP Tyer, LJ Blanshard, RJ Allan, K Kleese van Dam, IT Todorov, W Smith, VN Alexandrov, GJ Lewis, A Thandavan and SM Hasan. *Environment from the molecular level: an escience testbed project*, Proceedings of UK e-Science All Hands Meeting 2003, ISBN 1-904425-11-9 pp 302-305, September 2003.
- [2] MT Dove and NH de Leeuw. *Grid computing and molecular simulations: the vision of the eMinerals project*, Molecular Simulations (in press)
- [3] M Calleja, L Blanshard, R Bruin, C Chapman, A Thandavan, R Tyer, P Wilson, V Alexandrov, RJ Allen, J Brodholt, MT Dove, W Emmerich, K Kleese van Dam. *Grid tool integration within the eMinerals project*. Proceedings of the UK e-Science All Hands Meeting 2004, (ISBN 1-904425-21-6), pp 812817, 2004
- [4] M Calleja, R Bruin, MG Tucker, MT Dove, RP Tyer, LJ Blanshard, K Kleese van Dam, RJ Allan, C Chapman, W Emmerich, PB Wilson, JP Brodholt, A Thandavan, VN Alexandrov. *Collaborative grid infrastructure for molecular simulations: The eMinerals minigrid as a prototype integrated compute and data grid*, Molecular Simulations (in press)
- [5] M Dove. *The eMinerals Minigrid: An Infrastructure to Support Molecular Simulation Scientists*, ERCIM News No. 59, October 2004, 4849
- [6] Thain D, Tannenbaum T, and Livny M (2003) *Condor and the Grid*, in Grid Computing: Making The Global Infrastructure a Reality (ed Berman F, Hey AJG and Fox G, John Wiley), Chapter 11, 2003; see also <http://www.cs.wisc.edu/condor/>.
- [7] Moore RW and Baru C, *Virtualization services for data grids*, in Grid Computing: Making The Global Infrastructure a Reality, (ed Berman F, Hey AJG and Fox G, John Wiley), Chapter 16, 2003; see also <http://www.sdsc.edu/srb>.
- [8] Rajasekar A, Wan M, Moore R *MySRB & SRB components of a data grid*. in 11th IEEE International Symposium on High Performance Distributed Computing, Proceedings, pp 301–310, 2002
- [9] <http://www.mandrakesoft.com>.
- [10] <http://oscar.sourceforge.net>
- [11] <http://www.openpbs.org>.
- [12] <http://www.clusterresources.com/products/maui>.
- [13] Foster I, and Kesselman C (1997) , 11, 115128 *Globus: A Metacomputing Infrastructure Toolkit*, International Journal of Supercomputer Applications **11**, **115–128**, **1997**; see also <http://www.globus.org>.

- [14] Frey J, Tannenbaum T, Foster I, Livny M, and Tuecke S (2002) *Condor-G: A computation management agent for multi-institutional grids*, *Journal of Cluster Computing*, **5**, 237–246, 2002
- [15] R Tyer, M Calleja, R Bruin, C Chapman, MT Dove, RJ Allan, *Portal Framework for Computation within the eMinerals Project*, Proceedings of the UK e-Science All Hands Meeting 2004, ISBN 1-904425-21-6, September 2004.