Grid tool integration within the *e*Minerals project

Mark Calleja[1], Lisa Blanshard[2], Richard Bruin[1], Clovis Chapman[3], Ashish Thandavan[4], Richard Tyer[2], Paul Wilson[5], Vassil Alexandrov[4], Robert J Allen[2], John Brodholt[5], Martin T Dove[1,6], Wolfgang Emmerich[3], Kerstin Kleese van Dam[2]

1. Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EQ
2. Daresbury Laboratory, Daresbury, Warrington, Cheshire WA4 4AD
3. Department of Computer Science, University College London, Gower Street, London WC1E 6BT
4. Department of Computer Science, University of Reading, Whiteknights, PO Box 225, Reading RG6 6AY
5. Department of Earth Sciences, University College London, Gower Street, London WC1E 6BT
6. National Institute for Environmental *e*Science, Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge CB3 0EW

Abstract

In this article we describe the *e*Minerals mini grid, which is now running in production mode. This is an integration of both compute and data components, the former build upon Condor, PBS and the functionality of Globus v2, and the latter being based on the combined use of the Storage Resource Broker and the CCLRC data portal. We describe how we have integrated the middleware components, and the different facilities provided to the users for submitting jobs within such an environment. We will also describe additional functionality we found it necessary to provide ourselves.

Introduction

The *e*Minerals project (otherwise called "Environment from the molecular level"; Dove *et al*, 2003) is one of the NERC escience testbed projects. It is primarily concerned with the challenge of using computer simulations performed on molecular length and time scales to address important environmental issues such as including the effects of radiation damage in high-level nuclear waste encapsulation materials, the adsorption of pollutants on surfaces, and weathering effects (Alfredsson *et al*, 2004). The project consists of approximately 20 workers distributed over six geographic locations within the UK, with the computational resources available to this team being similarly distributed. The scientists in the project run a number of different simulation codes, which are based on being able to describe the interactions between atoms using either empirical model potential energy functions or a fuller quantum mechanical approach. Many of the simulations are based on Monte Carlo or molecular dynamics algorithms. All have high computational demands.

As a testbed project, one of the objectives of the *e*Minerals project is to create an enabling grid-based infrastructure appropriate for the science drivers. Our approach has been to build upon established standards such as Globus and Condor. One key feature has been to integrate compute and data middleware tools analogous to how compute and data operations are integrated at the operating system level. The approach has been to construct the *e*Minerals minigrid with close collaboration with the science users as a high priority, both to ensure that the minigrid best meets the need of the scientists and to help the users learn to use the new system – we consider the close interaction between the project grid developers and the scientists to have been particularly important in setting up the *e*Minerals minigrid. It should be appreciated that the use of a shared grid resource is a big change in how the scientists represented in the project would previously have carried out their work. Typically members of the molecular simulation community will work with a small set of individual compute resources, and will manage their data on these resources through the usual unix tools.

The purpose of this paper is to describe the construction of the *e*Minerals minigrid. We describe the various tools used, with focus on their integration in order to make the user's job lifecycle appear as seamless as possible. We also discuss a number of shortcomings of the tools and some difficulties we encountered in setting up the *e*Minerals minigrid.

2. Components of the eMinerals minigrid

2.1 Compute resources

The *e*Minerals minigrid comprises the following shared or contributed compute resources:

Three Linux clusters: These are located at Bath, Cambridge and UCL, and are each given the collective name Lake. Each cluster has one master node and 16 slave nodes, all with Intel Pentium 4 processors running at 2.8 GHz, and with 2 GB memory per processor. The nodes have Gigabit ethernet interconnections, they run PBS queues, and support MPI jobs. Each master node also hosts a data vault for the Storage Resource Broker (see below), and acts as a Globus Gatekeeper. At the present time, the clusters in Cambridge and UCL run v2.4.3 of the Globus Toolkit, and the cluster in Bath runs v3.2;

we are planning to soon update all clusters to v3.2. We are also planning to add a second linux cluster in Cambridge (40 nodes, similar configuration, called Pond) to the minigrid. The master nodes on each cluster act as the Globus gatekeepers to other resources on their local networks; in the case of UCL and Cambridge these nodes are the gatekeepers for access to the Condor pools described below.

IBM pSeries parallel computer: This machine is located in Reading, and consists of three IBM pSeries p655 nodes, each with eight POWER4 1.5 GHz processors and 16 GB memory. They have a dedicated 250 GB of storage and are linked via a private Gigabit Ethernet switch. The nodes run AIX 5.2 at the latest maintenance levels and the LoadLeveler batch job scheduler. In addition to IBM supplied C, C++ and Fortran compilers, IBM's Grid Toolbox v2.2 (which is based on Globus Toolkit v2.2) is installed and configured to run within the *e*Minerals minigrid.

UCL condor pool: A large Condor pool at University College London was put together by members of the *e*Minerals project in collaboration with the Information Systems group at UCL (Wilson *et al*, 2004). This pool consists of 930 teaching PCs running Windows. Since each of these machines act as a client to a Windows Terminal Server, their processing power is not heavily used by student users. The UCL Condor pool has a single master.

Cambridge condor pool: We have pooled around 25 computers into a small production/testbed condor pool in Cambridge. This is a heterogeneous pool, containing Silicon Graphics Irix workstations, Linux PCs and Windows PCs. We will shortly be adding a group of Macintosh G4 machines running Mac OS X. External access to this pool is currently through a Globus (v2.4.3) gatekeeper.

Grid middleware for the compute grid: As noted above, we have designed the *e*Minerals minigrid around the core tools of Globus and Condor. We have restricted our work to date to the functionality of the Globus 2 toolkit (as also embedded in GT 3); this decision was influenced by the use of Globus v2 in the construction of the UK Level 2 Grid, and the fact that the *e*Minerals science users are primarily working with legacy codes and do not want to wrap up their codes to fit in with another middleware paradigm. As we will remark below, the Globus toolkit 2 has a number of restrictions for which we have had to develop work-arounds. The Condor toolkit provides functionality that overcomes some of the restrictions in the user interaction with the compute resources in the form of the Condor-G toolkit, which wraps up globus job submission commands in the form of more standard Condor scripts.

2.2 Data resources

The *e*Minerals minigrid comprises the following shared data resources:

Storage Resource Broker: The Storage Resource Broker (SRB), developed at the San Diego Supercomputing Center, provides access to distributed data from any single point of access (Drinkwater *et al*, 2003). From the viewpoint of the user, the SRB gives a virtual file system, with access to data being based on data attributes and logical names rather than on physical location or real names. Physical location is seen as a file characteristic only. One of the features of the SRB is that it allows users to easily replicate data across different physical file systems in order to provide an additional level of file protection.

The SRB is a client-server middleware tool that works in conjunction with the Metadata Catalogue (MCAT). The MCAT server preserves the information about files as they are moved between different physical files systems. The SRB configuration employed within the *e*Minerals minigrid consists of the MCAT server held at CCLRC Daresbury, and 5 data storage systems (the SRB vaults) located in Cambridge (2 instances), Bath, UCL and Reading, giving a total storage capacity to the minigrid of around 3 TB. The first four use a RAID array on standard PCs with Intel Pentium 4 processors, with each vault on the Lake clusters providing 720 GB of storage and a further 500 GB on the Pond cluster. The Reading SRB vault is on a Dell Poweredge 700 server running SuSE Linux 9.0, providing 400 GB of storage.

The use of the SRB overcomes some of the limitations experienced when using the Globus toolkit for retrieval of files generated by applications running on the minigrid. As we will discuss below, the approach we take is to handle the interaction of the user and the minigrid with data through a job lifecyle entirely through the SRB.

Application server: The *e*Minerals minigrid application server is an IBM Bladecentre with a dual Xeon 2.8 GHz architecture and 2 GB memory per node, and is located at CCLRC Daresbury. The application server has a number of functions. It runs the MCAT server for the SRB, the web server for the eMinerals portals (see below), the MySRB web interface for the SRB, and the metadata editor (also see below) that runs alongside the data portal and the SRB.

Database cluster: The database cluster consists of two mirror systems acting as a failover server. Again, this is located at CCLRC Daresbury. It runs the Oracle Real Application Cluster Technology to hold the SRB MCAT relational database containing data file locations and the metadata database. The use of the Oracle Dataguard system is currently being implemented with an equivalent database cluster at the CCLRC Rutherford Appleton Laboratory in order to further increase the resilience of the database cluster.

2.3 The *e*Minerals integrated minigrid

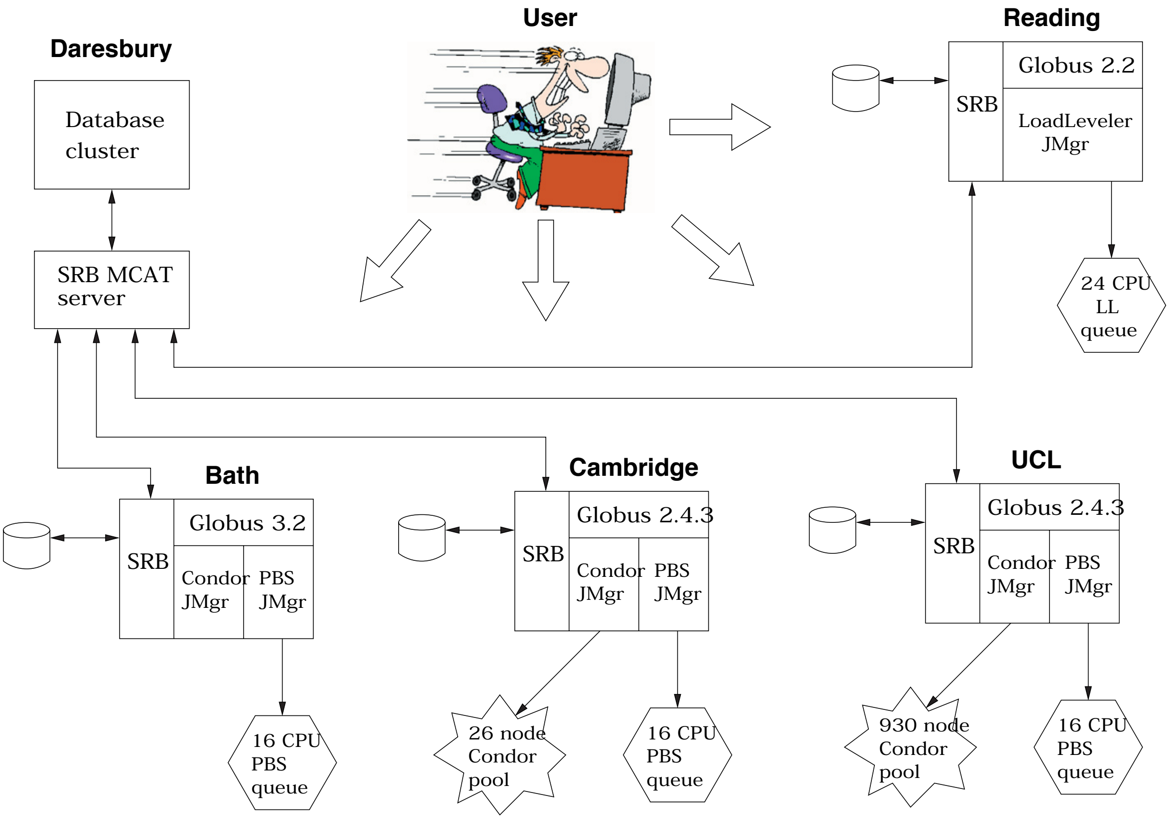

Figure 1: Representation of the architecture of the eMinerals Minigrid at the time of writing.

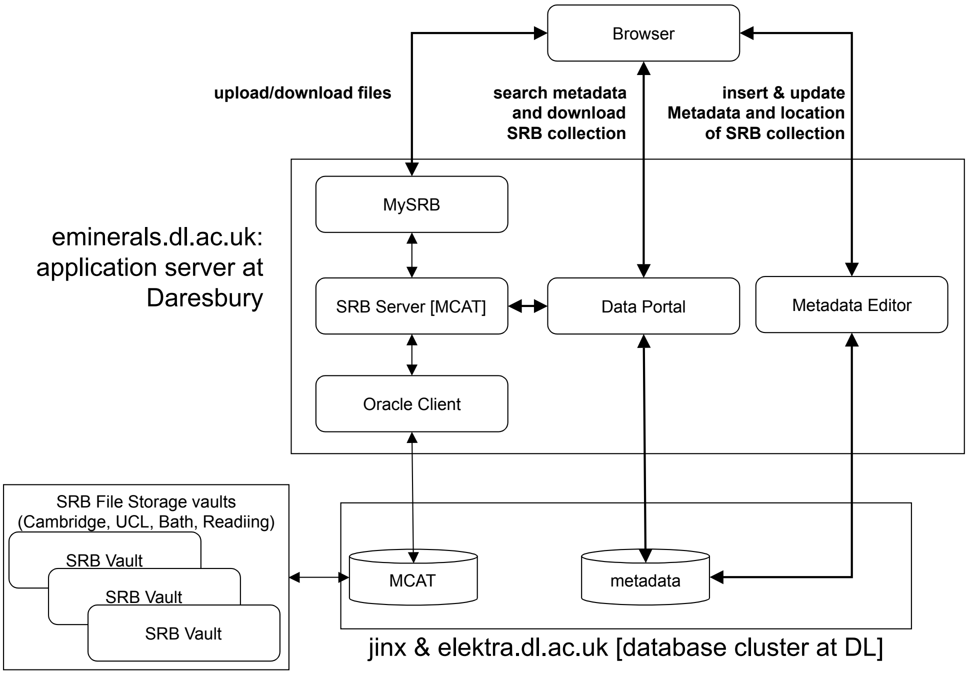
Browser
upload/download files
search metadata and download SRB collection
insert & update Metadata and location of SRB collection
eminerals.dl.ac.uk: application server at Daresbury
MySRB
SRB Server [MCAT]
Data Portal
Metadata Editor
Oracle Client
SRB File Storage vaults (Cambridge, UCL, Bath, Readiing)
SRB Vault
SRB Vault
SRB Vault
MCAT
metadata
jinx & elektra.dl.ac.uk [database cluster at DL]

Figure 2. Representation of the data component of the eMinerals minigrid.

The architectural arrangement of the *e*Minerals minigrid, composed of the integrated compute and data resources outlined above, is depicted in Figure 1. The architecture for data management within the project is shown in Figure 2.

The primary advantage of this distributed architecture is that all data files within the project are immediately available to all compute resources. Users upload input data files to the SRB prior to starting a calculation, and these data are then available wherever they choose to run the job. Similarly, on job completion, output data files are automatically stored within a nominated SRB vault, making them accessible to the user via any of the SRB's interfaces (InQ for Windows, MySRB for any web browser, or the SRB unix S-command line tools if installed locally). The SRB is also used to store executable images of applications. At the time of writing the project vaults house over 40 GB of data, made up of some 10,000 files. However, usage is rising steadily as team members become more confident with the technology.

After output files have been loaded into the SRB, they can be annotated using the Metadata Editor. This is a simple forms-based web application that enables details such as the purpose behind running study and performing a particular calculation, who was involved, when and where the data were generated, and where the data are stored in SRB to be entered. As a result, members of the *e*Minerals project can search for the study details and datasets using the Data Portal, another web application that provides uniform search capabilities and access to heterogeneous data resources (Drinkwater *et al*, 2003). Data files can also be downloaded through the Data Portal if desired.

Although the *e*Minerals minigrid is firmly rooted in the tools of Globus v2, with job submission handled through Globus, Condor and Condor-G toolkit commands and data accessed through the SRB, the architecture of the *e*Minerals minigrid retains the possibility to graft on a service-oriented work paradigm if this should prove useful for workflow issues. We are, for example, beginning to work with the Condor development team in order to integrate Condor with WSRF, using the *e*Minerals minigrid as our testbed.

2.4 Access to the *e*Minerals minigrid

The front end to the facilities of the *e*Minerals minigrid are based around the Globus toolkit. Currently these are a mixture of 2.x and 3.2 releases, though we are in the process of upgrading all gatekeepers to GT3.2. Hence there are four such gatekeepers, one on each Linux Lake cluster master node, and one on the IBM machine in Reading. All minigrid resources are accessed via one of these gatekeepers. Hence, the PBS queues on each cluster are accessed by requesting the corresponding jobmanager on that cluster in a Globus or Condor-G command. Similarly, the Condor pools at UCL and Cambridge are reached by requesting the correct Condor jobmanager from the gatekeeper, e.g. to request a Linux machine with an Intel architecture in a Condor pool one would nominate jobmanager-condor-INTEL-LINUX.

In order to facilitate the porting and building of code by users, one of the Lake clusters allows gsissh access and accepts jobs to its PBS queue by direct command-line submission. This is particularly useful when porting MPI-enabled applications. However, production runs are submitted to the rest of the minigrid across Globus.

Because access to the *e*Minerals minigrid is via Globus tools, users need to have access to the Globus client tools. We have found that installing the Globus and Condor-G client tools on every user's desktop machine has been an unsatisfactory experience. Because of this, we have provided a small number of dedicated machines to be used as job submission nodes within the minigrid. Indeed, only a small number of users have a full suite of client tools on their desktops, the reasons for which are mainly two-fold: *a*) installing these tools is not a trivial affair, and *b*) such tools require major configuration changes in local firewalls.

Although the architecture of the *e*Minerals minigrid represents a successful minigrid implementation, it does require that any firewalls present be suitably configured to allow the relevant traffic to pass. Such traffic occurs on well defined port ranges, but it has been necessary to work closely with institution computer support staff in order to investigate and solve a number of associated problems. One way to mitigate against such problems is to have all traffic propagate over a single, well defined, port such as port 80 for HTTP. The SRB web interface (MySRB) and the DataPortal take this approach, and we are developing a compute portal to assist users submit jobs to the minigrid and monitor their progress.

The architecture of our minigrid enables *e*Minerals grid developers and administrators to directly assist users with the usage of Grid resources. Indeed, a ticket-driven helpdesk system based on the OTRS software (Edenhofer *et al*, 2003) has been set up in order to systemise troubleshooting such problems. In effect, the deployment of a number of submission nodes, which act as gateways to these resources, allows administrators to configure, test and manage grid tools on behalf of users, limiting their actual need to deal with the complexities of installation (although some users have chosen to also install Globus and Condor-G client tools on their desktop machines). The user can then submit jobs either via these pre-configured nodes or from their own desktop PCs.

3. Job submission

Submitting jobs to such a grid environment in a manner that users find simple and intuitive has proven to be relatively tricky. The raw Globus command-line tools have not been particularly well received, and hence we have undertaken to provide users with a number of alternatives with varying complexity and functionality. These tools also provide access to grid resources outside our minigrid, such as the National Grid Service and the high-end national computing facilities.

From configured desktops or one of the minigrid submit machines, users can use Condor's Globus client tool, Condor-G, to submit jobs to the minigrid resources. Condor-G provides users with client-side job scheduling, effectively enabling them to manage their submissions to Grid resources in a local queue. When used with the Condor workflow tool DAGMan (Directed Acyclic Graph manager), preprocessing and post-processing scripts can be utilised to transfer the associated data in and out of the SRB. Hence a typical job would first start by the user uploading input data, and possibly even the executable file to the SRB if that is not already available on the minigrid, and then submitting a Condor DAG using condor_submit_dag which has the following steps in the workflow:

1. A perl script is launched on the remote gatekeeper using the Fork jobmanager that creates a temporary working directory and extracts into it the relevant input files from the SRB.
2. The executable that needs to be run is then passed to the appropriate jobmanager by the next vertex in the workflow. The job is run in the working directory created in the previous step. This is communicated

by passing the correct value to the relevant field in the GlobusRSL string in the Condor-G script for this step in the workflow, "Initialdir" for a Condor job and "directory" for a PBS job.

3. On completion of the previous step, the final part of the workflow is launched and another perl script is passed to the gatekeeper's Fork jobmanager to deposit the output data into the user's SRB area and clean up the temporary directory structure on the compute machine.

In fact it is sometimes necessary to have one DAGMan job spawn off another one since it may need to make use of run-time information, but the user need not worry about this level of detail. Once the workflow has been correctly encapsulated in the relevant DAGMan script(s) then the user only ever issues one command to execute the process. The main point here is that all data handling is done on the server side (and the execute machine), with that data being available to the user from any platform that supports one of the SRB's many client tools, such as the MySRB web browser interface.

This approach maps easily onto the data lifecycle paradigm, as discussed by Blanshard *et al* (2004) and Tyer *et al* (2004).

It is unfortunately true that dealing with such submission scripts as those mentioned above can be frustrating for some users. Hence, we are in the process of developing a web portal (Tyer *et al*; 2004), which will provide a browser interface for accessing all of the current functionality, as well as introducing some new services (e.g. job monitoring, resource discovery, accounting, etc.). Although this work is currently in progress, the aim is to provide a fully integrated workspace, capturing not just the functionality mentioned above but also other collaborative tools being developed within the project. At the time of writing (July 2004), this facility has limited functionality, but we hope to roll out a useful service within the next two months.

4. Problems encountered

The main limitations encountered while knitting together these various technologies have generally been related to the lack of functionality associated with the various Globus jobmanagers. Indeed, we have found that we have had to extend the perl modules for both the PBS and Condor jobmanagers, pbs.pm and condor.pm. The main problem with the PBS jobmanager is that it doesn't currently allow for different MPI distributions to be nominated, e.g. LAM or MPICH, compiled with GNU or Intel compilers, etc. For the Condor jobmanager extensions were necessary in order for output files to be returned to the submit machine, although that mechanism has been superseded now that output is uploaded into a SRB vault on the server side upon job completion.

Getting users to fully understand the steps involved in constructing, or at least editing, the DAGMan workflow scripts has not been trivial, and though most of the gory details are hidden (or at least pre-coded) for them, some involvement by the user is necessary. Although we hope that the introduction of the portal will circumvent most of these problems, it is unlikely that such a tool will completely replace the functionality offered by such scripts.

Load balancing across the minigrid is currently entirely at the users' discretion, which is not an ideal situation. This has meant that sometimes jobs have been queued on one resource while another resource was free to service their request. We have provided some rudimentary resource discovery tools to aid users in deciding where to submit their jobs, but the user still has to actively decide which cluster/pool to send that job to. These tools take the form of simple script wrappers for native scheduler commands, e.g. they might wrap a globus-job-run of a showq command to a PBS queue on a cluster, and simply echo back the output.

5. Conclusions and Future Work

The eMinerals minigrid is now in full service for production use for the project scientists, with only highly parallelised jobs requiring very low levels of interprocessor communication latency (e.g. as afforded by Myrinet interconnects) needing to be submitted elsewhere, e.g. the National Grid Service compute clusters or national high-performance facilities. The vast majority of the jobs in the project can be handled by the resources in the minigrid, from small single-node tasks on the Condor pools to parallel, MPI-type applications on the clusters. The use of the SRB has greatly facilitated data access throughout the minigrid, and it is its integration with the job-execution components of the architecture that has been the most obvious value-added feature of the project so far. The idea that a job can run on some unknown host (e.g. a node in a Condor pool) while using data stored in some unknown repository (one of the SRB vaults) has constituted a very novel *modus operandi* for most team members, but one whose benefits have become clear.

Future work will follow a number of strands, and improving the user-interface to the resources of the *e*Minerals minigrid is certainly a necessity. The intention is that the job submission portal being developed for the project will address these issues (Tyer *et al*; 2004). We will doubtlessly also have to take on board any changes that are implemented within the middleware we use, with the forthcoming introduction of WSRF standards within the Globus toolkit being the most obvious change. However, we also intend to migrate to newer versions of the SRB software that use certificate based authentication, and are monitoring developments within the Condor project, especially for proposed new features that facilitate the use of such pools in the presence of firewalls and private IP addresses (Son & Livny, 2003).

Acknowledgements

We are grateful to NERC for financial support, and to the computer support personnel at our various institutes.

References

Alfredsson M *et al* (2004), “Science outcomes from the use of Grid tools in the eMinerals project, Proceedings of UK e-Science All Hands Meeting 2004

Blanshard L, Tyer R, Calleja M, Kleese K & Dove MT (2004) “Environmental Molecular Processes: Management of Simulation Data and Annotation”, Proceedings of UK e-Science All Hands Meeting 2004

Doherty M *et al* (2003), “SRB in Action”, Proceedings of UK e-Science All Hands Meeting 2003, pp 51–59

Dove MT *et al* (2003) “Environment from the molecular level: an escience testbed project”, Proceedings of UK e-Science All Hands Meeting 2003, pp 302–305

Drinkwater G *et al* (2003) “The CCLRC Data Portal”, Proceedings of UK e-Science All Hands Meeting 2003, pp 540–547

Edenhofer M, Wintermeyer S, Wormser S & Kehl R (2003) “OTRS 1.2 - Manual”, http://doc.otrs.org/1.2/en/html/

Son S & Livny M (2003) “Recovering Internet Symmetry in Distributed Computing.” Proceedings of the 3rd International Symposium on Cluster Computing and the Grid

Tyer R *et al* (2004) “Portal Framework for Computation within the eMinerals Project”, Proceedings of UK e-Science All Hands Meeting 2004.

Wilson P *et al* (2004) “A Grid approach to Environmental Molecular Simulations: Deployment and Use of Condor pools within the eMinerals Mini Grid”, Proceedings of GGF 10