Collaborative grid infrastructure for molecular simulations: The *e*Minerals minigrid as a prototype integrated compute and data grid

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

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

Abstract

This paper describes a prototype grid infrastructure, called the "*e*Minerals minigrid", for molecular simulation scientists. which is based on an integration of shared compute and data resources. We describe the key components, namely the use of Condor pools, Linux/Unix clusters with PBS and IBM's LoadLeveller job handling tools, the use of Globus for security handling, the use of Condor-G tools for wrapping globus job submit commands, Condor's DAGman tool for handling workflow, the Storage Resource Broker for handling data, and the CCLRC dataportal and associated tools for both archiving data with metadata and making data available to other workers.

Keywords

Grid computing, escience, virtual organization, storage resource broker, eMinerals, Condor, Globus

1. Introduction

The traditional way of working in the area of molecular simulations is for one person (independent or managed) to run his/her own simulations on resources for which he/she has access (private or shared, ranging from desktop to supercomputer), to manage his/her own data files, and for the only interaction between the simulation scientist and the outside world to be via summaries of results in talks and papers, and perhaps through discussions with a small number of pertinent individuals. In several respects there are aspects of this model that are being changed by a number of diverse external considerations:

1. It is increasingly the case that research funding is being targeted towards supporting consortia rather than individual scientists. Unfortunately it is too easy for such consortia to just work to a low level of collaboration, in which scientists only interact through occasional meetings, through emails and telephone calls as required, and by reading about each other's work in documents produced by consortia members. Of course, many consortia rise above this base level, but new developments in grid technologies (to be defined later) will enable consortia to work in completely new ways.
2. To some extent related to the first point, it is going to become increasingly common for consortia to work on topics that are more challenging than can be tackled by individual simulation scientists. It may be the case that in a particular study, various team members will run different simulations, which will need to be compared, and team members may have different analysis tools they can run on the configurations generated by the simulations. In these cases, easy sharing of simulations and data will become increasingly important.
3. Unrelated to either of the other two points is that the nature of computing has changed radically over the past few years. Until recently, a lot of simulation work was carried out on large computers, often supercomputers based at central facilities. Moore's Law, technically defined as the number of transistors that can be built into an integrated circuit will double every 18 months, but popularly stated that the power of computers will double in the same time scale, has meant that the power of desktop computers has changed dramatically. In 1999 Apple produced the first desktop computer (using the G4 chip from Motorola with the additional vector processor) that could achieve 1 Gflops performance, which was then the USA Government's definition of a supercomputer. Modern supercomputers, now called "high performance" or "high capability" computers, are built as large arrays of powerful processors, with fast interconnections between them. Although the simulation scientist always needs more computer power, he/she is now faced with the question of what type of

computing power. Often the need is not for the best high-performance computing, which may in fact be inappropriate if the simulation does not scale well across the number of individual processors users are expected to make use of. Instead, simulation scientists often prefer what is called "high throughput" capability, in which they can run many different simulations (e.g. running one system at many simulation temperatures) on different computers. It is this case that grid computing will have a lot to offer simulation scientists.

4. Moore's law for computer power has similar laws for both the increase in hard disk capacity and bandwidth of networks. The law for disk capacity has a time constant for doubling capacity of around 12 months. Nowadays our desktop computers come with disks with capacity of order 100 GB as standard. In the past it would have been difficult to store regular dumps of atomic configurations from a large simulation, and much of the analysis would have been carried out during the simulation. Now the problem is that we have much more disk capacity than can be backed up by institutes' standard methods. The time constant for bandwidth is lower than for processor power and disk capacity, typically doubling every six months. There are two immediate implications of these two laws: first that it is possible to manage data using commodity disk store in a distributed sense (and it is possible to manage backups using similar systems), with access to distributed data no longer being limited by network constraints, and second that if we are going to store data across distributed systems simulation scientists are going to need new data management tools.

The concept of "grid computing" arose in the USA in the 1990's [1,2]. Initially the idea was to develop grids of national supercomputers, and to explore how scientists could make use of such massive capability. In order to develop such computing grids, it was necessary to develop a computing application layer, called the "middleware", that would operate between the operating systems of the individual components of the grid and the layer seen by the user, handling issues such as user authentication and authorisation, job submission and data recovery, and resource discovery. Today's grid systems will be tailored for individual needs, and are likely to include desktop computers and small clusters. The initial idea of grid computing now extends to include distributed data management. Since the grid concept involves distinct groups of individuals sharing resources, the idea of the "virtual organisation" has emerged simultaneously with the idea of grid computing.

In this paper we will discuss how the *e*Minerals project [3,4] has set up a minigrid infrastructure designed to meet the needs of a molecular simulation community. The next section discusses some of the motivations for incorporating grid computing methods into molecular simulation work. Section 3

gives a description of a prototype minigrid structure for molecular simulations developed by the *e*Minerals project. Section 4 discusses further developments that are now possible. An Appendix provides brief descriptions of some of the technological components used in the minigrid that are not described in the main body of the paper. This paper is restricted to the resources used for running simulation jobs and managing data; other aspects of running a virtual organisation for a molecular simulations community are discussed in a separate paper.

2. Motivation for grid computing

Grid computing has had an impact in a number of fields of the physical and biological sciences [2,5], and is now emerging as a method that can have value for molecular simulations. Before we describe the *e*Minerals implementation of a number of grid tools, we first clarify some of the motivations for the use of grid computing methods.

2.1 Sharing resources

Availability of computing power has always been a limitation that the molecular simulation scientist has had to work against. High performance computing power is often available to groups of simulation scientists, but usually in some sort of cost-determined restricted sense. As noted above, high-performance computing is not the only resource that is of value to the simulation scientist; often what is needed is significant high-throughput capability to facilitate detailed sweeps through model parameters (temperature, pressure, chemical variation etc) which can be easily met by having access to many commodity computers. In some cases the required capability is for high memory. By sharing resources between several research groups, individual scientists will usually have access to many more resources. Examples of the value of shared resources to meet high-throughput computing needs are given in references 6–9.

2.2 Data management

With greater availability of computer power, there is an increased need for data management. This need becomes even more acute as computers become increasingly more pervasive in all areas of our work. One suspects that fewer scientists nowadays are preserving the art of maintaining high-quality written log books, particularly since it is now more common to use bespoke solutions to move data between various files and applications (e.g. running the Unix "`grep`" tool on a data file and pasting the output

into a spreadsheet in order to produce a graph). Moreover, we now often need to maintain data files across different systems (e.g the Unix system used to generate work, and the windows system used to generate graphs). In the extreme case the data management method used by a simulation scientist may be reduced to a combination of the scientist's memory and the Unix "`ls -l`" command to find a set of files based on chronological memory. Grid tools have the potential to provide new effective and easy-to-use data management strategies.

Grid data management tools provide the means to share data between collaborators in a transparent method, i.e. without the originator of the data needing to transfer the data onto something like a public ftp site and inform collaborators of where the data are stored, and without the need for collaborators to have to make an initial request. Data grids provide one route to support the advanced collaborations discussed in Section 1.

Use of data markup languages, such as the Chemical Markup Language [10–12], allow data to be understood by collaborators (or at least by the codes used by collaborators). These are discussed elsewhere in this collection of papers [13,14] and also in references 15 and 16.

2.3 Project integration and interoperability

A real-life example will illustrate how a project may use several codes and resources, and could be helped by the use of an integrated grid structure. The example is of simulations of cation ordering in a layer silicate such as muscovite, formula $MgSi_3AlO_8(OH)$ [17]. The various steps in a complete study might be [18,19]

1. Obtain a trial structure from a crystallographic database.
2. Most structures are obtained by X-ray diffraction, and these typically do not give the positions of the hydrogen atoms. An initial set of positions may be guessed based on some working knowledge of related structures, and refined using an empirical lattice energy minimisation code.
3. A more accurate structure may be obtained, based on a quantum mechanics lattice energy minimisation code.
4. The next stage will require running many copies of one of the lattice energy minimisation codes with different configurations of some of the cations (in our example of muscovite, this would mean different configurations of the Al and Si cations across the tetrahedral sites). Because of the need to run many configurations, it is likely that this stage will use empirical models or quantum mechanics formulations with time-saving approximations.

5. It will be desirable to check some of the results with quantum mechanics codes with fewer approximations than used in stage 4.
6. The energies extracted from stage 4 above will be analysed to parameterise a model Hamiltonian.
7. The model Hamiltonian will be used in a set of Monte Carlo simulations to study the equilibrium cation ordering as a function of temperature.

This example has a number of different requirements. Stages 2 and 6 require relatively low-level capabilities, stages 3 and 5 may require something approaching high-performance capability, and stages 4 and 7 require high-throughput capability with reasonable performance per processor (and probably high memory requirements for the quantum mechanics component in stage 4). The example also requires different types of data transfer between stages, some of which require a certain amount of human intervention. Between the first three stages the main need is to transfer details of the crystal structure. For stage 6, the need is for the transfer of computed energies from stages 4 and 5. The results of stage 6 consist of a set of energy parameters, which will need to be transferred to stage 7. Stage 7 will also need elements of the structure from the earlier stages. In this real-life example, an integrated grid structure with interoperability between codes will significantly increased the productivity of the simulation scientist, and would enhance the process of collaboration if carried out by a team.

3. The *e*Minerals minigrid: integrating compute and data capabilities

3.1 Components of the compute grid

The prototype *e*Minerals minigrid (see references 20 and 21 for earlier and more technical discussions) consists of the following purpose-built or contributed compute resources. The core middleware tools that are used to managed the shared resources are Globus [1,22] to handle communications between resources, including the important security issues, Condor [23] to link together resources distributed within an institute, and the Portable Batch System (PBS) to schedule jobs on a cluster. Globus and Condor are described in more detail in Appendices A.1 and A.2 at the end of this paper.

Linux clusters: Three replica clusters are located at Bath, Cambridge and UCL (Bruin et al, in preparation), and are each called 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 RAM per processor. The nodes have Gigabit ethernet interconnections. Each master node acts as the job manager, supporting both PBS and MPI. 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. A fourth Linux cluster, based in Cambridge and called Pond, has 40 nodes with Intel Pentium 4 processors running at 1.7 GHz and with 512 MB RAM per processor. At the time of writing, we are also incorporating a small cluster of Apple G5 Xserve dual-processor nodes, each having 8 GB RAM, and called Lagoon. The comparison between these different clusters highlights the fact that a minigrid structure contains provision for studies that have different requirements, some having high memory requirements and some having high-throughput requirements. The master nodes on each cluster also act as the Globus gatekeepers to other resources on their local networks; in the case of UCL and Cambridge these nodes are also 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, each with either 256 or 512 MB RAM. Since each of these machines act as a client to a Windows Terminal Server, little of their individual processing power is used by student users. The UCL Condor pool has a small number of submit nodes, of which the UCL lake gatekeeper is one.

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, Windows PCs and Macintosh G4 eMac desktop computers, with various RAM configurations. These machines are either classroom computers or individual researchers' desktop computers (we now put every desktop into the Condor pool, and insist that every individual researcher should submit jobs to the overall pool using Condor job submit commands rather than running on their own desktop computer). Each machine in the pool can act as a submit node. External access to this pool is currently through the Globus (v2.4.3) gatekeeper on the Lake cluster.

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; 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 [24], which wraps up Globus job submission commands in the form of more standard Condor scripts – this is described in more detail in Appendix A.3.

3.2 Data grid

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 [25–28]. 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 Linux clusters 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. We will shortly be adding a sixth vault installed on the Xserve cluster, with an additional 500 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 *e*Minerals 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.

3.3 Integrated 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.

After output files have been loaded into the SRB, they can be annotated using the Metadata Editor (Blanshard et al 2004). This is a simple forms-based web application that enables details such as the purpose behind running the study and performing a particular calculation, the personnel involved, and when and where the data were generated, to be added as metadata. 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 (Blanshard et al 2003; 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 the emerging *Web Services Resource Framework* (WSRF), using the *e*Minerals minigrid as our testbed.

3.4 Access to the *e*Minerals minigrid

The front end to the facilities of the *e*Minerals minigrid is based around the Globus toolkit. Currently the minigrid has a mixture of 2.x and 3.2 releases (see Appendix A.2 for a description of the different versions), though we are in the process of upgrading all gatekeepers to GT3.2. There is one gatekeeper for each cluster, and 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-condorINTEL-LINUX`.

In order to facilitate the porting and building of code, one of the Lake clusters allows `gsissh` access and accepts jobs to its PBS queue by direct command-line submission. However, production runs can only be submitted to the rest of the minigrid only through Globus.

Because access to the *e*Minerals minigrid is via Globus tools, users need to have access to the Globus client tools. Installing the Globus and Condor-G client tools on every user's desktop machine has not proved to be easy (they will not work on Windows machines for example, or with machines whose IP addresses are assigned dynamically), and 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.

3.5 Job submission

To enable users to submit jobs to a grid environment using Globus in a way that they find simple and intuitive has required a separate development effort. The raw Globus command-line tools have not proved to be sufficiently user-friendly for our purposes, and the use of bespoke scripts that require users to add modifications is also not satisfactory. The approach we have taken is to develop general-purpose scripts based on the use of two Condor tools, namely Condor's Globus client tool, Condor-G, to submit jobs to the minigrid resources (Frey et al 2002), and the Condor workflow tool DAGMan (Directed Acyclic Graph manager; Thain et al 2003).

Submission of a standard job to the *e*Minerals minigrid involves a three-stage workflow implemented using Condor's DAGMan tool (see Appendix A.3):

1. The job first creates a temporary working directory on the gatekeeper and extracts any relevant job input data files from the SRB.
2. The main job executes on one of the compute resources.
3. Finally, all nominated output files are put into the SRB for the user to view from his/her desktop.

These steps represent different nodes in the workflow, which are automatically generated for the user by using our own variant of Condor's `condor_submit` command, called `my_condor_submit`, which includes extensions to the Condor submit file syntax to allow SRB-specific extensions (see Appendix A.4 for more details).

All these steps make use of the fork jobmanager, except for the actual job execution stage, which makes use of the jobmanager for the relevant resource (e.g. PBS, Condor, etc.). Hence, the user only

ever issues one command, without having to worry about the details of the underlying workflow. It is this wrapper's job to autogenerate the various scripts required to perform the workflow. 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. More details are given in Appendix A.4.

This approach maps easily onto the data lifecycle paradigm discussed by Blanshard et al (2004) and Tyer et al (2004). In addition to developing the script submission method, we are in the process of developing a web-based compute 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 at the time of writing (October 2004) 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.

4. Future developments

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 within Globus which interaction with the PBS and Condor jobmanagers, namely `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.

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 `globusjob-run` of a `showq` command to a PBS queue on a cluster, and simply echo back the output. These are threaded wherever possible to help mitigate the delays inherent when performing such queries across a grid.

The *e*Minerals minigrid is now in full service for production use by 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 doubtless 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 source of possible changes. Moreover, we are currently migrating 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).

Appendix. Components of grid computing

Here we give a brief review some of the key grid technologies that are of potential use to molecular simulations.

A.1 Condor: a tool for creating a desktop grid

Condor was developed as a means to utilise idle computing time on desktop machines. Two examples used in the *e*Minerals minigrid are groups of teaching computers and the desktop computers in a research group. A Condor setup has one machine acting as the master node, and all others acting as clients, thereby defining a Condor pool. The master node handles control of jobs submitted to the condor pool, which includes the tasks of job scheduling and resource brokering, job monitoring, and

data transfer. Any number of the machines within a pool can be configured to allow job submission. In a pool composed of teaching computers, it may be most sensible to have only one submit node, but in a pool based on owned desktop machines it is likely to be more transparent to users to allow all machines to be capable of submitting jobs.

Condor has a number of key grid facilities built in. These include:

1. Ability to handle a wide range of computing platforms and operating systems. The Condor resource brokering facilities can be used to specify a particular platform that a job will run on if executable images are only available for a limited set of platforms or operating systems.
2. Fault tolerance, namely that if one machine fails the job will be migrated onto another member of the Condor pool. In certain system configurations Condor allows for checkpointing of jobs, so that with machine failure the job will restart close to the point it had reached at failure. At the time of writing, it is not possible to use checkpointing in some critical parts of the *e*Minerals minigrid, namely for Windows computers and for the use of the Intel Fortran compiler on Linux machines.
3. Respect for the owners of contributed resources. For example, jobs on the Condor pool can run with low priority so that their use as a desktop machine is not compromised. It is also possible to establish rules such that Condor jobs will only run outside of office hours, or after a fixed period of inactivity.
4. Condor jobs and data generated by Condor jobs are secure from the view of the desktop machine on which they are running. Similarly, the desktop machine is secure against potentially hazardous commands run through the Condor system. For example, it is not possible to run a Condor command that manipulates (views, deletes, alters) data on the desktop.
5. Condor only requires that the client software be installed on the desktop, and from that point onwards nothing more needs to be done on the client computers. The important point in this respect is that it is not necessary to create user accounts on any of the desktop computers: all jobs handled by Condor are run under a generic account.
6. Condor is relatively easy to install and administer, and there is now a sufficiently large user base from which help can be obtained.

It may be questioned whether, by some definitions, Condor is a true Grid tool, in that it operates on locally-controlled resources. It is in fact possible to join separate Condor pools together in a process known as “flocking” to form what would be called a true grid. Unfortunately this process does not yet handle well the problems caused by firewalls or the use of private IP addresses, and hence has not been used extensively.

A.2 Globus and GSI security

In establishing a grid between distributed resources with separate ownership, it was necessary to develop a set of tools, known as the middleware, whose role across the grid is analogous to the operating system on a single computer. The Globus toolkit is one of a small number of middleware tools for supporting grid computing. The main features are of the Globus toolkit for the *e*Minerals minigrid are

1. handling issues of security, including authorisation and authentication of identify
2. handling job submission
3. handling data transfer

These are the main components of version 2 of the toolkit, and are being propagated through subsequent versions of Globus. Versions 3 and 4 (the latter is not available at the time of writing, but is set to completely replace version 3) are attempts to incorporate web services within the Globus framework; it should be noted that the *e*Minerals minigrid does not yet make use of web services except in the data portal. However, at the time or writing, Globus toolkit 2 is no longer supported, which motivates the need to upgrade the *e*Minerals minigrid to v3.2 even though we do not have immediate plans to use the web services functionality.

Security is clearly important in a grid environment. On one hand, it is essential that users only gain access to resources to which they are entitled, and that these limitations are controlled effectively. On the other hand, since users will be accessing many resources within a grid structure, it is important to avoid the need for maintaining a long list of usernames and passwords. The approach adopted within the Globus toolkit is to use standard X.509 digital certificates based on private/public key cryptography. In the UK, these certificates are issued by the central UK eScience Certification Authority. A digital certificate demonstrates two things to another computer the user may attempt to access: it identifies who the user is, and it demonstrates that the user really is who he/she claims to be. Authorisation to use remote resources is handled by the same digital certificates; a user's certificate will be listed on any computer for which he/she is permitted access.

A.3 Condor wrapping: Condor-G and DAGman

The Globus toolkit provides commands for job submission to remote computers, but experience is showing that these are often difficult for users to come to grips with. On the other hand, the Condor commands are proving to be much easier to come to grips with. The Condor developers have wrapped

Globus commands up within the Condor framework, making some script development (or at least script management) somewhat easier; this is called Condor-G.

Central to many grid applications is the concept of workflow. A simple example is the case when job C depends on the output of jobs A and B, which therefore need to be executed first. In a grid sense, jobs A and B are free to run on any resources in the grid, but requiring that both complete before job C can run. A simpler workflow is that jobs A, B and C have to run sequentially. Although this sounds trivial, it is particularly important for data management. In the *e*Minerals minigrid, job A retrieves data from the SRB (see below), job B runs the computation, and job C sends the resultant output files to the SRB.

Workflow patterns can be achieved using Condor's workflow management tool, known as DAGman (Directed Acyclic Graph Manager). DAGman handles dependencies between jobs, so that if one job depends on the other, the DAGman will ensure that the jobs run in the correct order. The DAGman operates at a higher level than the Condor scheduler, and submits jobs to Condor in the order set by the workflow dependencies. The scripting for the DAGman is relatively straightforward for many workflows.

A.4 Job submit scripts

The submission of a job to the *e*Minerals minigrid requires the use of a script developed by one of the authors of this paper (MC), called `my_condor_submit`. This script handles the running of the job and the transfer of data between the SRB and the compute resources. It is available as a download from `www.eminerals.org`. The user requirements are met through a simple file whose name is given as the argument to the execution of the script. The file has the form:

```
Universe              = globus
Globusscheduler       = <minigrid resource>/jobmanager-<jobmanager>
Executable            = <name of executable binary or script>
Notification          = NEVER

# Next line is example RSL for a single-processor PBS job
# Modifications are required for other job managers
GlobusRSL = (arguments=none)(job_type=single)(stdin=<filename>)

Sdir          = <some directory in the SRB>
Sget          = <list of input file names, or * for wildcard>
Sput          = <list of output file names, or * for wildcard>
Output        = <standard output file name>
transfer_output = False
```

```
Log             = <name of log file>
Error           = <name of standard error file>
Queue
```

The values of parameter given in <*angle brackets and italics*> can be altered by the user. The `Sdir` directory is a directory in the user's SRB space. The `Sget` parameter is a list of input files in the SRB that need to be fetched at the start of the job. The `Sput` parameter is a list of output files that are to be put into the SRB after the execution of a job. These two parameters can be `*` for a wildcard list, which is particularly useful when the exact list of output files is not known in advance. The `Executable`, `stdin`, `Output`, `Log` and `Error` parameters are the names of files that are held or created on the computer from which the job has been submitted. The *`executable`* file, for example, will be transferred to the minigrid resource as part of the job submission process. This can be a binary or a script; the latter would be used if the executable binary file will be obtained from the SRB. The *`minigrid resource`* will be one of the compute resources within the minigrid, and would be assigned the name of the computer, e.g. `lake.geol.ucl.ac.uk`. The *`jobmanager`* parameter would be `PBS` for one of our linux clusters, or `condor-INTEL-linux` for a Linux computer within a Condor pool (see section 3.4).

Acknowledgements

We are grateful to NERC for funding through the eScience thematic programme. We are grateful to other colleagues on the eMinerals project for helping to test and improve the capabilities of the *e*Minerals minigrid.

References

[1] Foster I, and Kesselman C, ed (1998) The Grid: Blueprint for a New Computing Infrastructure, 1st edition (Morgan-Kaufman)

[2] Foster I, and Kesselman C, ed (2003) The Grid: Blueprint for a New Computing Infrastructure, 2nd edition (Morgan-Kaufman)

[3] Dove MT, Calleja M, Wakelin J, Trachenko K, Ferlat G, Murray-Rust P, de Leeuw NH, Du Z, Price GD, Wilson PB, Brodholt JP, Alfredsson M, Marmier A, Tyer RP, Blanshard LJ, Allan RJ, Kleese van Dam K, Todorov IT, Smith W, Alexandrov VN, Lewis GJ, Thandavan A, and Hasan

SM (2003) Environment from the molecular level: an escience testbed project. *Proceedings of UK e-Science All Hands Meeting 2003,* (EPSRC, ISBN 1-904425-11-9) pp 302–305

[4] Dove MT and de Leeuw N (2005) Grid computing and molecular simulations: the vision of the *e*Minerals project. Molecular Simulations **XX**, 1234–5678

[5] Berman F, Hey AJG and Fox G, ed (2003) *Grid Computing: Making The Global Infrastructure a Reality*, (John Wiley),

[6] Calleja M and Dove MT (2004) Calculating activation energies in diffusion processes using a Monte Carlo approach in a grid environment. *Lecture Notes in Computer Science*, **3039**, Part 4, pp 483–490

[7] Wells S, Alfredsson M, Bowe J, Brodholt J, Bruin R, Calleja M, Catlow R, Cooke DJ, Dove MT, Du Z, Kerisit S, de Leeuw N, Marmier A, Parker S, Price GD, Smith B, Spohr H, Todorov I, Trachenko K, Wakelin J and Wright K (2004) Science outcomes from the use of Grid tools in the *e*Minerals project. *Proceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 240–247

[8] Du Z, de Leeuw NH, Grau-Crespo R, Wilson PB, Brodholt JP, Calleja M and Dove MT (2005) A computational study of the effect of Li-K solid solutions on the structures and stabilities of layered silicate materials – an application of the use of Condor pools in molecular simulation. Molecular Simulations **XX**, 1234–5678

[9] Marmier M, Spohr H, Cooke DJ, Kerisit S, Brodholt JP, Wilson PB and Parker SC (2005) Self Diffusion of Argon in Flexible, Single Wall, Carbon Nanotubes. *Molecular Simulations* **XX**, 1234–5678

[10] Murray-Rust P and Rzepa HS (1999) Chemical markup, XML, and the Worldwide Web. 1. Basic principles. *Journal Of Chemical Information and Computer Sciences*, **39**, 928–942

[11] Murray-Rust P, Rzepa HS and Wright M (2001) Development of chemical markup language (CML) as a system for handling complex chemical content. *New Journal of Chemistry* **25**, 618–634

[12] Murray-Rust P and Rzepa HS (2003) Chemical Markup, XML, and the World Wide Web. 4. CML Schema. *Journal Of Chemical Information and Computer Sciences* **43**, 757–772

[13] Jon's paper. *Molecular Simulations* **XX**, 1234–5678

[14] Chapman C, Wakelin J, Artacho E, Dove MT, Calleja M, Bruin R and Emmerich W (2005) Workflow issues in atomistic simulations. *Molecular Simulations* **XX**, 1234–5678

[15] All Hands Jon

[16] All Hands Peter

[17] Palin EJ, Dove MT, Redfern SAT, Bosenick A, Sainz-Diaz CI and Warren MC (2001) Computational study of tetrahedral Al-Si ordering in muscovite. *Physics and Chemistry of Minerals* **28**, 534–544,

[18] Bosenick A, Dove MT, Myers ER, Palin EJ, Sainz-Diaz CI, Guiton B, Warren MC, Craig MS and Redfern SAT (2001) Computational methods for the study of energies of cation distributions: applications to cation-ordering phase transitions and solid solutions. Mineralogical Magazine **65**, 193–219

[19] Warren MC, Dove MT, Myers ER, Bosenick A, Palin EJ, Sainz-Diaz CI, Guiton B and Redfern SAT (2001) Monte Carlo methods for the study of cation ordering in minerals. Mineralogical Magazine **65**, 221–248

[20] Building the eMinerals minigrid

[21] Calleja M, Blanshard L, Bruin R, Chapman C, Thandavan A, Tyer R, Wilson P, Alexandrov V, Allen RJ, Brodholt J, Dove MT, Emmerich W and Kleese van Dam K (2004) Grid tool integration within the eMinerals project. *Proceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 812–817

[22] Foster I, and Kesselman C (1997) Globus: A Metacomputing Infrastructure Toolkit. *International Journal of Supercomputer Applications*, **11**, 115–128

[23] 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

[24] 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

[25] Moore RW and Baru C (2003) 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 11

[26] Rajasekar A, Wan M, Moore R (2002) MySRB & SRB – components of a data grid. In *11th IEEE International Symposium on High Performance Distributed Computing, Proceedings*, pp 301–310

[27] All Hands 2003 paper on SRB

[28] All Hands 2004 paper on SRB

[29] Blanshard L, Kleese van Dam K, Dove M (2003) Environment from the molecular level e-science project and its use of CCLRC's web services based data portal. In *ICWS'03: Proceedings of the International Conference on Web Services*, (ed Zhang LJ), pp 164–167

[30] Blanshard L, Tyer R, Calleja M, Kleese K and Dove MT (2004) Environmental Molecular Processes: Management of Simulation Data and Annotation. *Proceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 637–644

[31] Drinkwater G, Kleese K, Sufi S, Blanshard L, Manandhar A, Tyer R, O'Neill K, Doherty M, Williams M, Woolf A. The CCLRC dataportal. In *ICWS'03: Proceedings of the International Conference on Web Services*, (ed Zhang LJ), pp 285–291

[32] Edenhofer M, Wintermeyer S, Wormser S & Kehl R (2003) "OTRS 1.2 Manual", http://doc.otrs.org/1.2/en/html/

[33] Foster I, Kesselman C and Tuecke S (2001) The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal or Supercomputer *Applications*, **15**, 200–222

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

[35] Tyer R, Calleja M, Bruin R, Chapman C, Dove MT and Allan RJ (2004) Portal Framework for Computation within the eMinerals Project. *Proceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 660–665

Figure captions

1. The structure of the *e*Minerals minigrid. The diagram is organised to highlight the middleware tools as the core of the infrastructure.
2. The structure of the data component of the *e*Minerals minigrid.

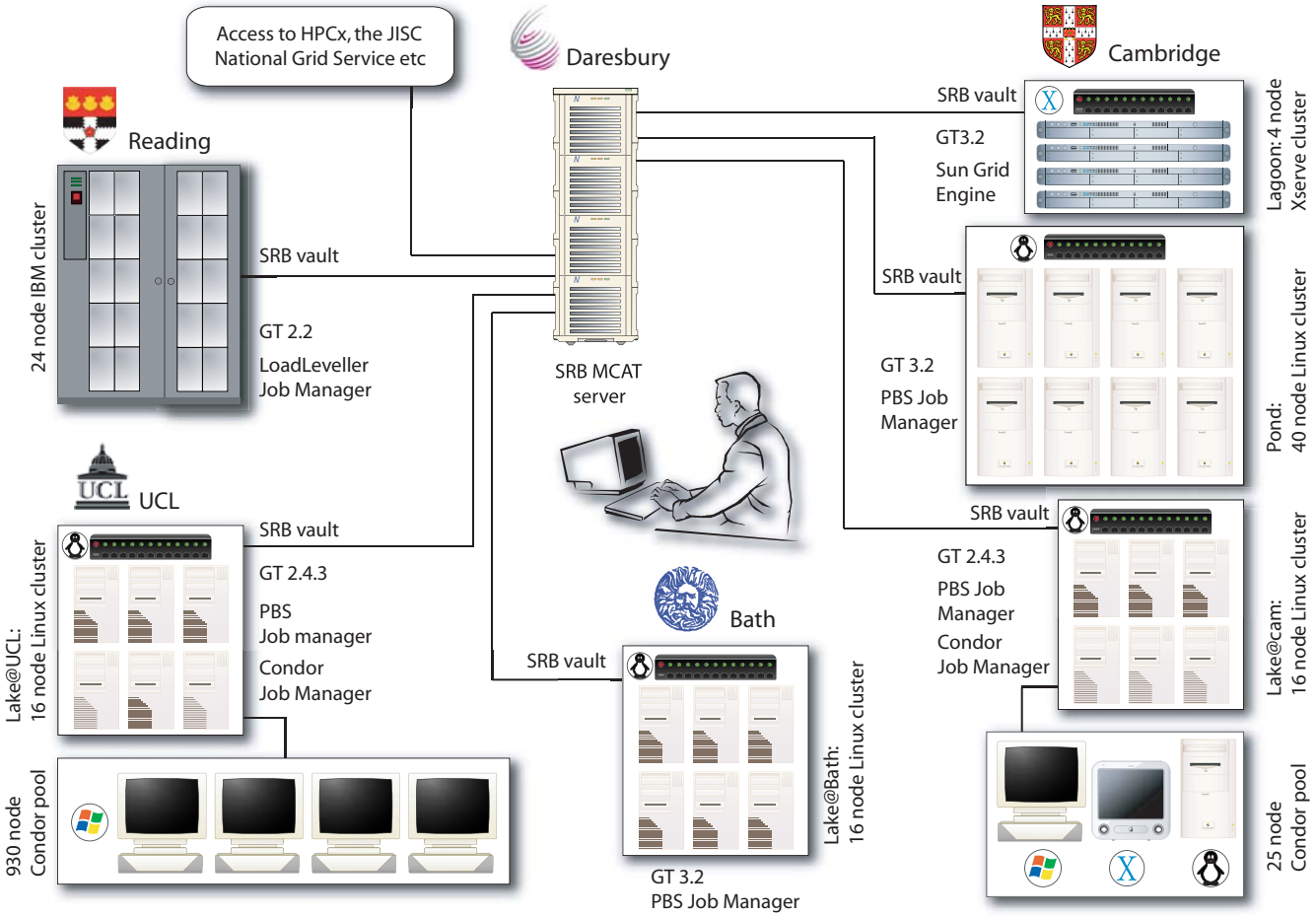
Access to HPCx, the JISC National Grid Service etc
Daresbury
Cambridge
Reading
24 node IBM cluster
SRB vault
GT 2.2
LoadLeveller
Job Manager
SRB MCAT
server
SRB vault
GT3.2
Sun Grid
Engine
Lagoon: 4 node
Xserve cluster
SRB vault
GT 3.2
PBS Job
Manager
Pond:
40 node Linux cluster
UCL
Lake@UCL:
16 node Linux cluster
SRB vault
GT 2.4.3
PBS
Job manager
Condor
Job Manager
930 node
Condor pool
Bath
SRB vault
Lake@Bath:
16 node Linux cluster
GT 3.2
PBS Job Manager
SRB vault
GT 2.4.3
PBS Job
Manager
Condor
Job Manager
Lake@cam:
16 node Linux cluster
25 node
Condor pool

Figure 1

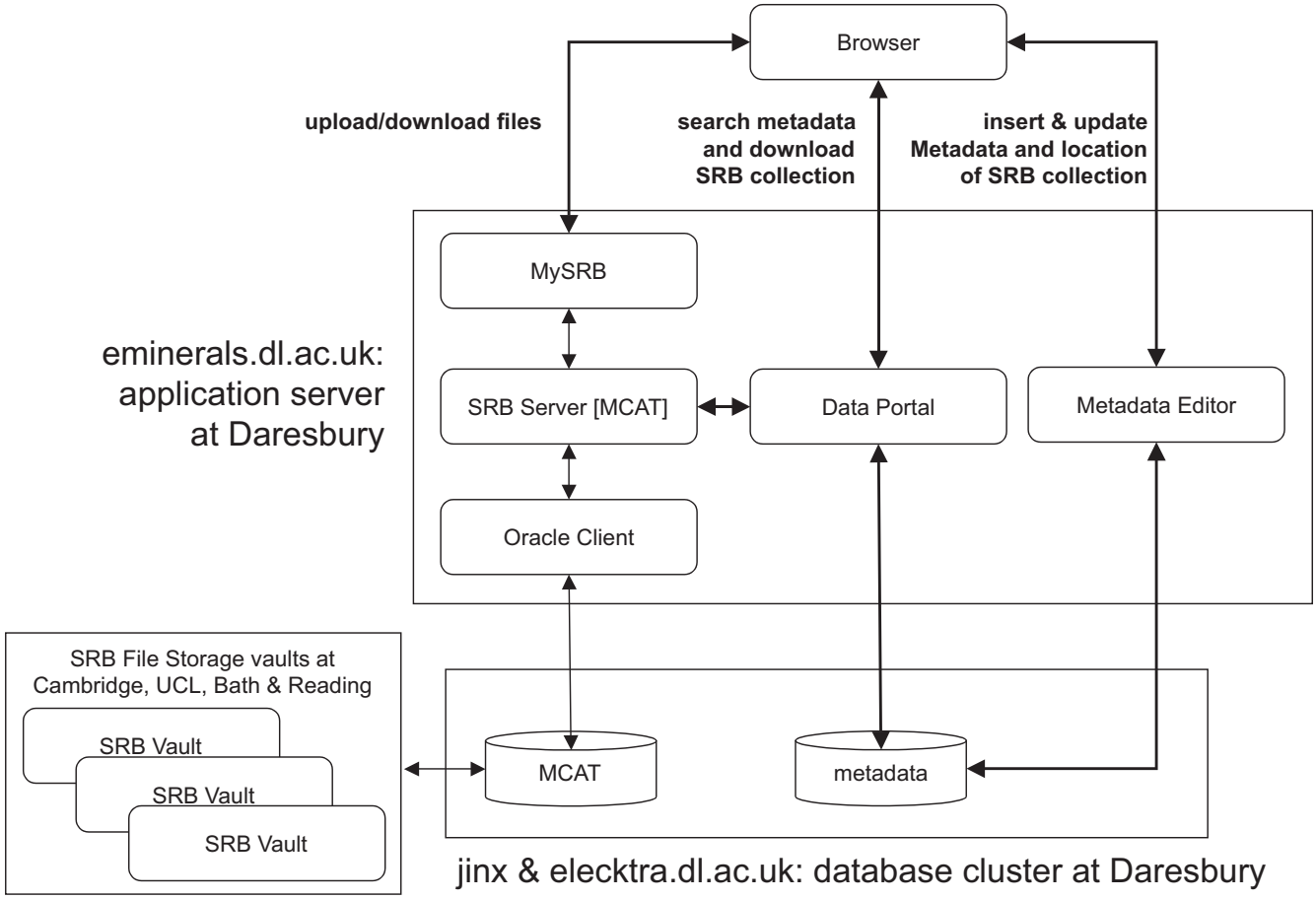
Browser
upload/download files
search metadata and download SRB collection
insert & update Metadata and location of SRB collection
MySRB
eminerals.dl.ac.uk: application server at Daresbury
SRB Server [MCAT]
Data Portal
Metadata Editor
Oracle Client
SRB File Storage vaults at Cambridge, UCL, Bath & Reading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

Figure 2