

Computational grids for mid-sized collaborative projects: the *eMinerals* experience

MT Dove, TOH White, AM Walker, RP Bruin, KF Austen, E Artacho, M Calleja, MG Tucker
Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EQ, UK
mtd10@cam.ac.uk

RP Tyer, P Couch, K Kleese van Dam, RJ Allan, IT Todorov
CCLRC, Daresbury Laboratory, Daresbury, Cheshire WA4 4AD, UK

C Chapman, W Emmerich
Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK

A Marmier, SC Parker
Department of Chemistry, University of Bath, Bath BA2 7AY, UK

MO Blanchard
Royal Institution, 21 Albemarle Street, London W1S 4BS

Z Du
School of Crystallography, Birkbeck College, Malet Street, London WC1E 7HX

GJ Lewis, V Alexandrov
Department of Computer Science, University of Reading, Whiteknights, Reading RG6 6AY, UK

M Alfredsson, JP Brodholt
Department of Earth Sciences, University College London, Gower Street, London WC1E 6BT, UK

P Murray-Rust
Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, UK

Abstract

Grid computing has the potential to revolutionise how small groups of simulation scientists work together to tackle new science problems. In this paper we report how the eMinerals project has developed a small scale integrated compute and data grid infrastructure – the eMinerals minigrid – and developed generic job submission tools that exploit this infrastructure and which enable the science users to also access other grid systems.

1. Introduction

Whilst much of the current emphasis on grid computing is on large-scale collaborations – like those found within the particle physics communities – there are many smaller-scale communities who can benefit enormously from grid computing. The *eMinerals* project [1] is an exemplar of how grid computing has enabled a smaller scientific community to tackle problems that are beyond the reach of a single academic unit.

The *eMinerals* project is a collaboration between molecular simulation scientists and grid computing specialists, with team members drawn from 9 departments within 7 institutions. The project was funded as a UK eScience testbed project, with the objective to explore how grid computing technologies can be used to support molecular simulation science. This area of science is one that is growing rapidly in line with the parallel developments in computing technologies and simulation methods, and will underpin much of our knowledge of materials, whether for technological or environmental applications, in the coming years.

One can identify two traditional work patterns of molecular simulation scientists. First, simulation scientists tend to work alone or in very small groups. Second, the more demanding simulation studies have been run on high-performance computing facilities. Whilst for the most challenging types of simulation (typically quantum mechanical studies of high accuracy or with very large systems) this is likely to remain the working pattern, the rapid growth in the power of commodity computing will create new possibilities for molecular simulation science. It is now possible to run simulation calculations on a modern PC

that would recently have required supercomputing capabilities. Moreover, grid methods may appear to the scientist to be offering unlimited compute capacity. This will affect the traditional approach. On one hand, there is new scope for combinatorial or ensemble molecular simulation studies. It has always been useful to perform parameter sweeps, such as across a range of temperatures, but these can now be performed over many more state points than previously possible. This will lead to a massive increase in data production rates, as well as challenges in managing jobs, workflow and information. Such challenges increasingly mean that simulation studies benefit from scientists working within collaborative teams.

2. The scientific challenge

The scientific challenge that the eMinerals project is tackling concerns the adsorption of pollutants on mineral surfaces [2]. The mineral-pollutant interaction is important because of its impact on the transport of pollutants within the environment. Consider the example of chlorinated aromatic organic molecules. Molecules such as the polychlorinated biphenyls (PCBs, $C_{12}H_xCl_{12-x}$) and polychlorinated dibenzo-*p*-dioxins (PCDDs, $C_{12}O_2H_xCl_{8-x}$) have been released into the environment by human activities over the past few decades. Unfortunately these molecules are both toxic and long-lived, and feed back into the human food chain. Understanding their transport is vital for developing remediation strategies. A similar problem is the release of heavy metal ions such as arsenic through mining and drilling, and chemicals released through agriculture.

Molecular-scale simulations can play a role in understanding the transport of pollutants through the calculation of the energies associated with adsorption of pollutants onto the surfaces of the minerals found in soils (such as silica and silicates, clays, carbonates, sulphides, oxides and hydroxides). Molecular-scale simulations can provide numeric data, such as the adsorption energy, and an understanding of processes such as the complexes formed by different charge state of arsenic cations interaction with water and mineral surfaces.

The challenge is to understand a wide range of processes in a coherent manner. To achieve this, we need more than one type of simulation methodology, including methods based on empirical representations of the forces between atoms, and those that calculate the forces using quantum mechanics. The empirical

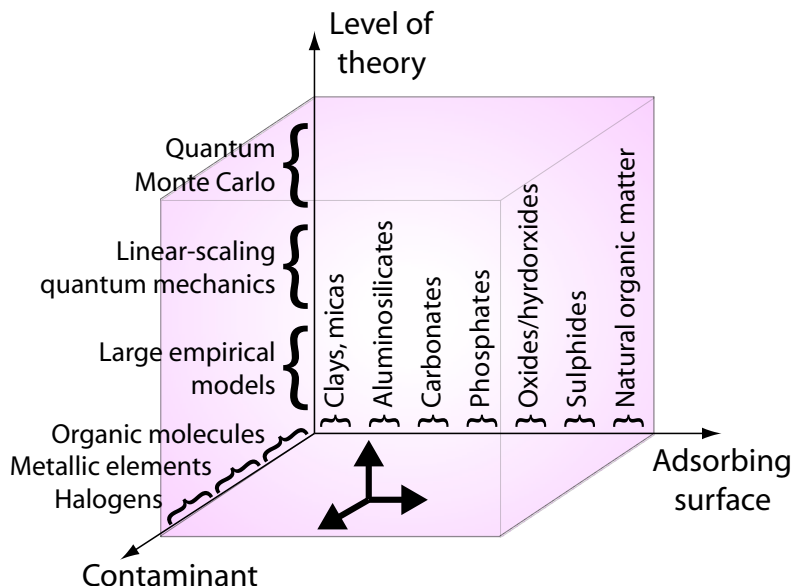


Figure 1. The three-dimensional space that defines the science challenge faced by the eMinerals project. The science space is spanned by the range of pollutants and the range of soil mineral surfaces, and the third axis concerns the level of theory used in the simulation methodology.

(© Martin Dove and the eMinerals project)

approaches are much faster and hence work better with large systems, but suffer from the problem of accurately parameterising the model. Typically simulation studies require more than one approach; this provides another driver for scientists working in teams rather than alone.

This discussion is represented in Figure 1, which shows the variables of pollutant, surface and methodology as a three-dimensional phase space. One of our objectives is to explore this phase space aided by grid computing methods. In this paper we will describe our work on developing appropriate grid computing methods to support our project scientists.

3. The eMinerals minigrid

3.1. Integrated compute and data grid infrastructure: the eMinerals minigrid

The eMinerals minigrid (Figure 2) is in some senses a proof-of-concept of the possibility to create a small-scale multi-purpose heterogeneous compute grid with an integrated data management system [3,4]. The central compute components are a set of clusters with various job managers (PBS/Maui combinations and IBM LoadLeveller) [5] and several multi-platform Condor pools. We have set up Globus gatekeepers on each resource. User authentication and authorisation are controlled by the use of the Globus Security

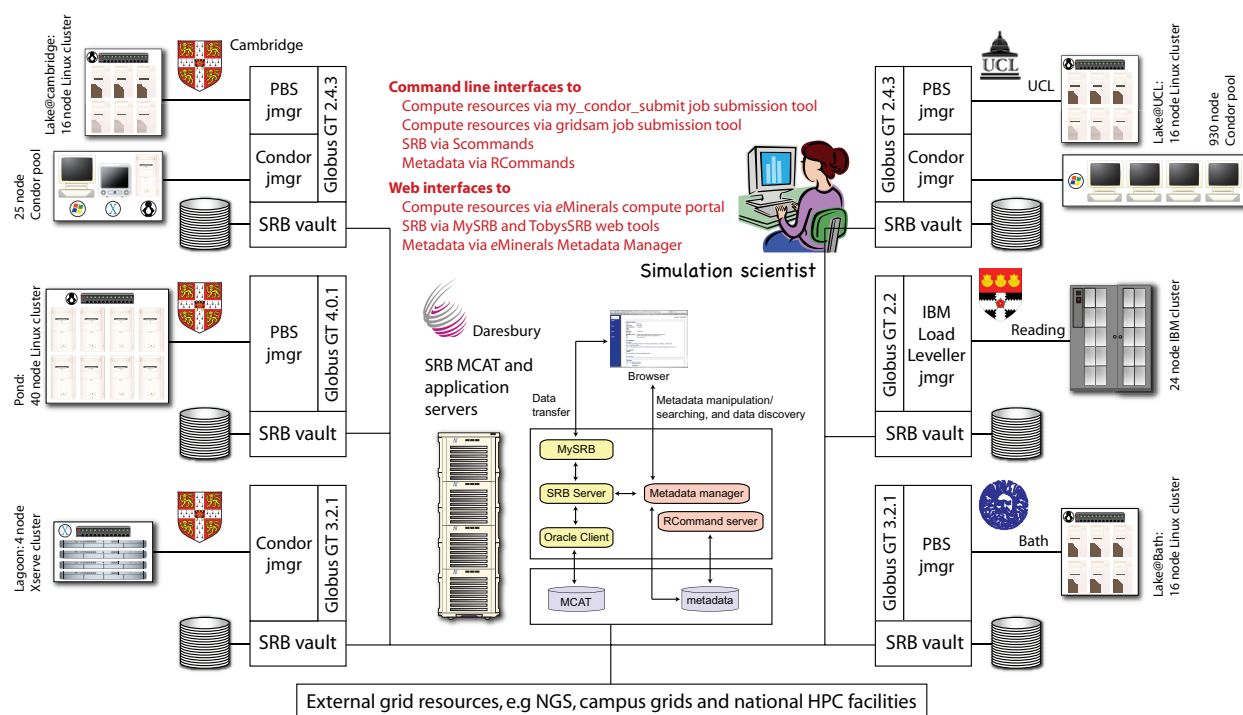


Figure 2. Representation of the eMinerals minigrid. The figure shows the various compute platforms (clusters and Condor pools), and the SRB central server and data vaults. The figure also shows the various job managers and instances of the Globus toolkit associated with each resource. The central part of the figure shows the architecture of the data management tools, including the metadata database. The simulation scientist has a number of command-line and web interfaces with the compute and data resources. The minigrid is directly linked to other resources, such as the UK National Grid Service and campus grids.
(© Martin Dove and the eMinerals project)

Infrastructure with individual user X.509 digital certificates. We do not use Globus resource monitoring and discovery tools because the eMinerals minigrid resources are expected to be running all the time: the eMinerals virtual organisation, being relatively small, is not dynamic in the sense that larger virtual organisations are. We will describe job submission below, but for the moment it should be clear that job submission is via the use of the Globus toolkit.

At an early stage in the development of the eMinerals minigrid it was decided to integrate the San Diego Storage Resource Broker (SRB), with data vaults being set up on each compute resource (see Figure 2). The original reason for a close integration was to use the SRB for data transfer in and out of the minigrid, rather than rely on file transfer tools such as GridFTP (which had some complications); subsequently we have made considerable use of the SRB for forming complete archives of combinatorial studies, and for the obvious objective of sharing data across the eMinerals virtual organisation. With regard to sharing data, not only do we need to share data, we also need to make the information content accessible between collaborators. This area of work and the

innovations developed by the eMinerals project are discussed in more detail in a companion paper [6].

The first objective of developing the eMinerals minigrid was to demonstrate that building and maintaining such a minigrid is a viable prospect for a small-scale collaborative project. The second objective was to explore how such a minigrid infrastructure can be exploited by simulation scientists. To this end we have developed a number of tools, including the job submission tools discussed in the next section and data access tools [6]. The guiding principle has been “usability”, namely that for simulation scientists the use of the minigrid should give easily recognisable advantages without the new technology being unreasonably intrusive.

3.2. Job submission to the minigrid compute environment

As a testbed project, it was decided from the outset that we would not allow users to log in to the compute resources to submit their jobs directly to specific queues; this was prompted by our ambition to demonstrate the feasibility of operating a true

distributed grid infrastructure. In our early tests, we learned that our scientist users were very comfortable with Condor job submission scripts, but found scripting with Globus tools was too intrusive. In order to make access to the compute resources of the minigrid easy for our scientists regardless of the specific applications they want to run, we developed a job submission tool, programmed in perl, called `my_condor_submit` (MCS) [3,4,7]. In outline, MCS has the following features:

- ▶ MCS will metaschedule over a group of compute resources using a simple round-robin resource utilisation check and a database cache;
- ▶ MCS interacts with the SRB to locate and download data (and job executables) at the start of a job and to upload output files at the end of a job;
- ▶ The user can either specify the names of files to be downloaded or uploaded from/to the SRB, or can specify wildcards;
- ▶ MCS can automatically harvest metadata, using the AgentX and RCommand utilities [8];
- ▶ MCS works with a range of job schedulers;
- ▶ MCS is a generic tool and will work with other grid infrastructures, including the UK National Grid Service and similar external grid infrastructures, and University campus grids;
- ▶ MCS works with the Condor-G wrapping of Globus, and uses Condor's DAGman workflow tool to handle the data and job management contiguously.

To use MCS, the user prepares a relatively simple Condor-like job script, examples of which are given in Figures 3 and 4. To understand these examples, it is helpful to know that MCS envisages a simple three stage workflow:

1. The user places all input files and executables into the SRB; they do not need to be in the same directory (collection). When the MCS job runs, the first task within the workflow is to download all the files from the SRB onto the compute resource.
2. The main job then runs. The job could be a single executable, such as one of our simulation codes, or

```
# Specify the name of the executable to run
Executable      = gulp

# Specify where the executable should get stdin from and
# put stdout to
GlobusRSL = (stdin=andalusite.dat)(stdout=andalusite.out)

# Specify an SRB collection to get the relevant
# executable from
pathToExe      = /home/codes.eminerals/gulp/

# Specify a metadata dataset for all metadata
RDatasetId     = 55

# Specify a directory to get files from, put files to
# and relate to metadata created below
Sdir           = /home/user01.eminerals/gulpminerals/
Sget           = *
Sput           = *

# Creates and names a metadata data object
Rdesc         = "Gulp output from andalusite at ambient P/T"
# Specify metadata to get from files with Agent-x;
# get environment and default metadata only
AgentXDefault = andalusite.xml
GetEnvMetadata = True
```

Figure 3. A simple example of an MCS script. In this example the executable (gulp) is held within a general collection in the SRB. The user (user01) is anticipating all input and output files to be held within one collection within the SRB, as specified by the `Sdir` directive, and the wildcards on the `Sget` and `Sput` directives imply that MCS should download all files within the selected collection at the start of the job and place all generated files into the same collection at the end of the job. The `AgentX` call specifies that a default set of metadata will be gathered from the output xml file.

(© Martin Dove, Richard Bruin and the eMinerals project)

a shell script with a series of commands or programs to execute.

3. After the main job has run, MCS performs two main functions before closing. First it places copies of all output files into the SRB (collection specified by the user). Second, MCS automatically collects metadata from the key output files.

(Note that before running MCS, users also need to run the `grid-proxy-init` Globus command to set up their certificate-based access to the minigrid resources.)

First we look at the example in Figure 3, which is the simpler of our two example MCS scripts. In this case the user is gathering the executable from a general codes area on the SRB, and is using wildcard settings for the directives to download files from the SRB and to upload the generated output files. The final directive in the script concerns the collection of metadata. MCS will extract metadata from the output xml file. The specific form of the directive implies that MCS will collect a default set of metadata from the xml file.

```

# Specify the executable to run
Executable      = siesta
# Instruct condor to not tell us the job outcome by email
Notification    = NEVER

# Specify which file to use for stdin and stdout
GlobusRSL = (stdin=chlorobenzene.dat)(stdout=chlorobenzene.out)

# Force overwriting when uploading / downloading files
SForce          = true

# Specify an SRB collection to get the relevant executable from
pathToExe       = /home/codes.eminerals/siesta/
# Specify a list of machines that we are happy to submit to
preferredMachineList = lake.bath.ac.uk lake.esc.cam.ac.uk lake.geol.ucl.ac.uk
pond.esc.cam.ac.uk
# Specify the type of machine to be submitted to
# (throughput for a condor pool and performance for a cluster)
jobType         = performance
# Specify how many processors to use on the remote machine
numOfProcs      = 1

# Specify a metadata study to create a dataset within
RStudyId        = 1010
# Create and name a metadata dataset to contain data objects
RDatasetName    = "chlorobenzene on clay surface"

# Specify an SRB collection to do some transfers to / from
Sdir            = /home/user01.eminerals/clay_surface/
# Specify that we want to get every file from this collection
Sget            = *

# Specify another SRB collection to do some transfers to / from
Sdir            = /home/user01.eminerals/chlorobenzene
# Specify to put all files into the specified SRB collection
Sput            = *

# Create and names a metadata data object
Rdesc           = "chlorobenzene molecule on clay surface: first test"
# Specify metadata to get with Agent-x (Tied to the previous Sdir line)
# Get environment metadata
GetEnvMetadata  = true
# Get default metadata from the specified file
AgentXDefault   = pcbprimfixed.xml
# Get z coordinate information and store as zCoordinate in the metadata database
AgentX          = zCoordinate, pcbprimfixed.xml:/molecule[1]/atom[last]/zCoordinate
# Get lattice vector information and store in the metadata database
AgentX          = latticeVectorA, pcbprimfixed.xml:/Module/LatticeVector[1]
AgentX          = latticeVectorB, pcbprimfixed.xml:/Module/LatticeVector[2]
AgentX          = latticeVectorC, pcbprimfixed.xml:/Module/LatticeVector[3]
# Get the final energy and store in the metadata database
AgentX          = finalEnergy, pcbprimfixed.xml:/Module[last]/PropertyList[title='Final
Energy']/Property[dictRef='siesta:Etot']
# Store an arbitrary string of metadata
MetadataString  = arbString1, "Test of molecule height and separation"

# Leave the code's stderr on the remote machine, to be uploaded to the SRB
Transfer_Error  = false
# End the file (taken from the condor input file)
queue

```

Figure 4. Caption on next page.

Figure 4 (figure on previous page). More complex example of an MCS input script. This example shows the metascheduling over a selected group of clusters, with a directive as to whether this is a cluster (assumed to be running MPI if required) or a Condor pool. Unlike the example in Figure 3, this example shows the use of more than one SRB collection for the data. Much of the script is concerned with metadata collection. In this specific example, which is part of a study to compute the energy of a PCB molecule using the siesta quantum mechanics code, the metadata to be collected concern the z coordinate of the molecule, the size of the simulation box (the latticeVector parameters), and the final energy, using calls to the AgentX library. The user is also specifying to write some additional metadata.

(© Martin Dove, Richard Bruin and the eMinerals project)

At this point it is worth noting how the metadata collection is carried out [8]. AgentX is a set of libraries for extracting information from xml files. Since most of our simulation codes are written in Fortran, we have developed a set of libraries to enable Fortran codes to write XML output [9]. We use a protocol of three types of output data: basic metadata associated with the simulation code and run-time environment; reflections of the input parameters that control the simulation (such as temperature or pressure); and finally the property values generated by the simulations (such as final energies and volumes). MCS passes instructions to MCS to read selected output data, which in turn are written to the eMinerals metadata database and associated with file locations in the SRB. Although the SRB has its own metadata system, it was not found to be adequate for our requirements. More details of our metadata tools are given in [7,8]; the point made here is that metadata are collected for each grid job, saved as part of a study/dataset/dataobject hierarchy, and available for collaborators to user to search for data.

Effort has been expended to make MCS generic. We have used MCS on other grid environments in which there is access to the SRB (in the case of external Condor pools, this simply means that the Condor pool is accessible via a Globus gatekeeper on which the SRB Scomand scripting tools are installed). To exploit the metadata tools, it is also necessary for AgentX and the RCommand scripting tools to be installed on the external grid resource or the associated Globus gatekeeper.

3.3. Combinatorial job submission for parameter-sweep studies

As noted earlier, the main area in which simulation scientists can exploit grid infrastructures is in the ability grids offer to run large-scale combinatorial or

ensemble studies, including studies involving a sweep across a range of parameter values. The grid enables the many simulations within a single study to be run concurrently, but the challenge is to enable the user to set up and manage all the jobs in a way that is as simple as running just one job. To this end we have developed a script-based tool [7]. The user supplies a template job input file and provides information on which parameter values are to varied from one job to another in another file. By running one command the tool creates all the job input files, one for each of the possible values of the varied quantities. These input files are automatically uploaded to a specially created directory within the SRB, with one subdirectory for each simulation to be submitted. At the same time, MCS input files are created for each of the different jobs on the local computer.

The actual process of submitting the jobs is performed by running a second command. This causes all of the generated MCS files to be submitted to the grid independently. Since MCS has metascheduling and load balancing, the jobs will be submitted to run on separate parts of the minigrid environment.

3.4. Desktop access

At the time of writing, MCS needs to be run on a computer on which both Globus and Condor are installed. Whilst for some users this is not a problem, for other users this is not a trivial requirement, particularly for certain operating systems. One short-term solution is to provide a set of job-submission computers for users to work from. Although this is less than ideal, in fact there are grid systems where this approach is used for security reasons. We are currently tackling the problem of desktop grid access in two ways. First, we have developed a web-services wrapping of MCS, called RMCS, which enables users to use MCS from their desktop without having to install other tools. RMCS submits the MCS file to a central server which performs the actual submission of MCS to the grid environment. Second, we have developed a web services equivalent to MCS based on Gridsam and BPEL [10]. Both tools have the same functionality as MCS, and enable users to submit large numbers of jobs as part of a combinatorial study. At the time of writing these are both in the testing stage and are yet to be rolled out across the project.

4. Case studies

4.1. The compressibility of amorphous silica

Amorphous silica has the unusual property that it initially becomes softer (i.e. the compressibility increases) on increasing pressure; most materials simply become harder due to the fact that atoms are

compacted in towards each other. In the case of silica, the atoms are connected within a network of corner-sharing SiO_4 tetrahedra. This network has a certain degree of flexibility which is lost under both high positive pressure and negative pressure; the anomalous behaviour of the compressibility of amorphous silica arises as an intermediate state between the two extremes of positive and negative pressure.

We have studied the anomalous compressibility in amorphous silica using the molecular dynamics simulation method [11]. Our first aim was to compute the volume as a function of pressure, with subsequent aims to analyse the resultant volume curve and perform detailed analyses on the generated configurations at each pressure. This study was easily broken down into many simulations, each with a different pressure but otherwise identical simulation parameters. The user provided a template input file with all of the necessary input parameters, and used the parameter sweep tools to configure the scripts to vary pressure between the lower and upper values and in the desired number of steps (for example, values might vary between ± 5 GPa in steps of 0.01 GPa, producing 100 simulations.). The user then ran the two commands as outlined in §3.3, and the jobs automatically ran on the compute resources of the eMinerals minigrid. On completion of the jobs, the values of the sample volume were extracted from all the output xml files on the SRB using tools described in the companion paper [6]. The results are shown in Figure 5.

In fact, the study involved running a number of analysis tools. These were run as part of the same job as the simulation. The workflow was constructed within a shell script which was submitted by MCS.

These simulations were performed using the DL_POLY_3 code [12], which uses our Fortran libraries [9] to generate xml output. Typically each simulation run took around half a day. We used the TobysSRB tool [6,13] to inspect output files, with relevant graphs (such as volume as a function of time to ensure that the volume fluctuated about a mean value) generated as svg files embedded within an xhtml file on the fly for viewing within a web browser.

4.2. The adsorption of pollutant molecules on mineral surfaces

The example in §4.1 represents a common case where the varied parameter represents a state of the

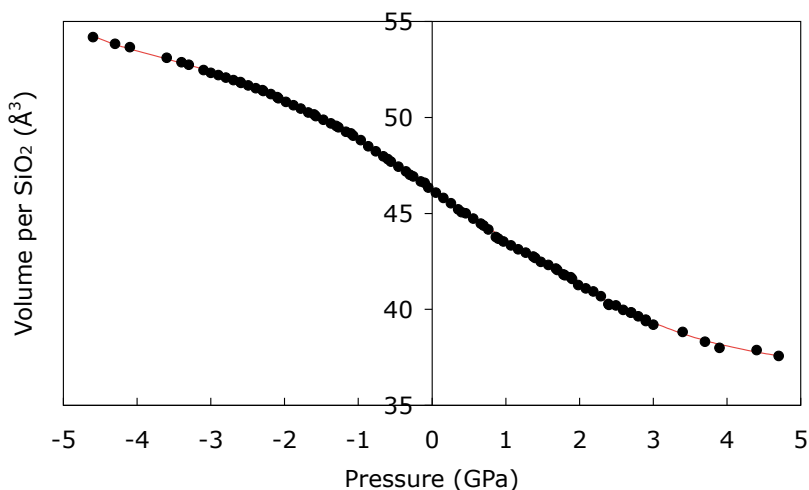


Figure 5. The pressure-dependence of amorphous silica calculated using molecular dynamics simulation, with the study carried out using the compute resources of the eMinerals minigrid. The points represents results from the simulations, and the curve is a fitted polynomial.

(© Martin Dove, Andrew Walker and the eMinerals project)

system (such as pressure or temperature). A second class of combinatorial study is where the actual molecular structure is changed between jobs. One example is the study of PCDD molecules on a clay mineral surface. The topology of the PCDD molecule is shown in Figure 6a, and the clay surface is shown in Figure 6b. There are 75 different permutations (congeners) of hydrogen and chlorine atoms in the PCDD molecule, and grid methods are essential to aid in a study involving all congeners. We have used both empirical and quantum mechanical methods to calculate the energy of adsorption of each molecule on the top surface of the clay mineral pyrophyllite. The adsorption energies calculated from the empirical models are plotted in Figure 6c, with a colour coding to show the results for different numbers of chlorine atoms. It is clear that there is stronger adsorption for larger numbers of chlorine atoms. A similar picture emerges from the quantum mechanical calculations. Typical running times for the quantum mechanical simulations are half a day per calculation, with three calculations required for each point. The empirical models run times were shorter. As in §4.1, this study was considerably aided by the use of grid computing methods coupled with our data and information management tools [6].

6. Acknowledgements

This work has been supported by NERC (grant reference numbers NER/T/S/2001/00855, NE/C515698/1 and NE/C515704/1).

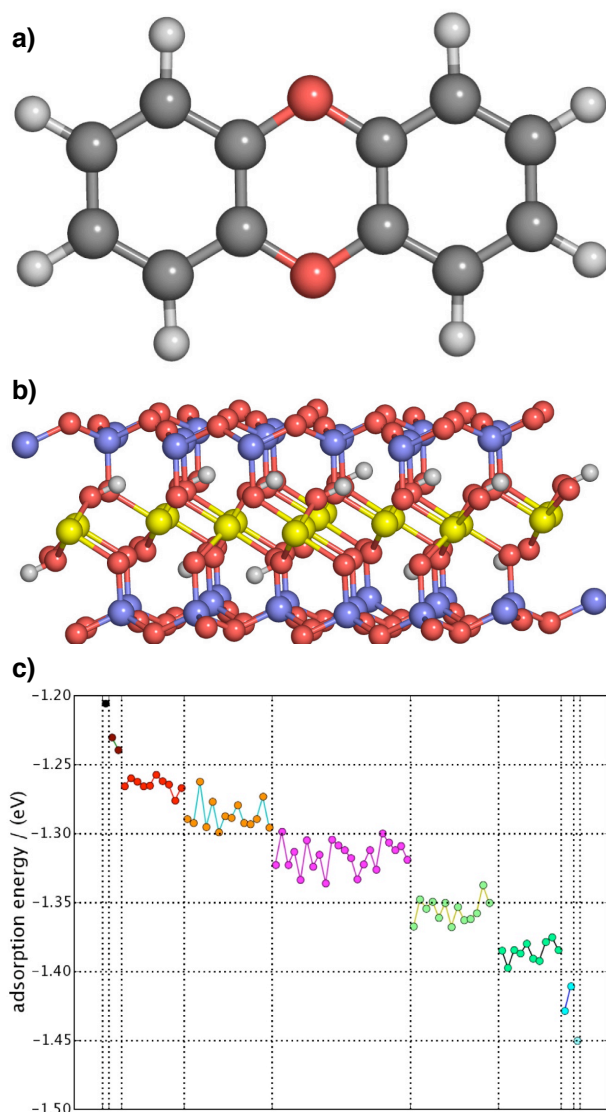


Figure 6. a) Representation of the PCDD molecule; red = oxygen, grey = carbon, white = hydrogen or chlorine. b) Representation of a layer of atoms in a pyrophyllite clay structure; red = oxygen, blue = silicon, yellow = aluminium, white = hydrogen. c) the adsorption energy plotted for different numbers of chlorine atoms, colour coded for different numbers (increasing from left to right) with more different congeners for intermediate numbers. (© Martin Dove, Toby White, Arnaud Marmier and the eMinerals project)

10. References

- [1] MT Dove, NH de Leeuw Grid computing and molecular simulations: the vision of the eMinerals project. *Molecular Simulations* **31**, 297–301, 2005
- [2] MT Dove, E Artacho, TO White, RP Bruin, MG Tucker, P Murray-Rust, RJ Allan, K Kleese van Dam, W Smith, RP

Tyer, I Todorov, W Emmerich, C Chapman, SC Parker, A Marmier, V Alexandrov, GJ Lewis, SM Hasan, A Thandavan, K Wright, CRA Catlow, M Blanchard, NH de Leeuw, Z Du, GD Price, J Brodholt, M Alfredsson. The eMinerals project: developing the concept of the virtual organisation to support collaborative work on molecular-scale environmental simulations. *Proceedings of All Hands 2005* (ISBN 1-904425-53-4), pp 1058–1065, 2005

[3] M Calleja, R Bruin, MG Tucker, MT Dove, R Tyer, L Blanshard, K Kleese van Dam, RJ Allan, C Chapman, W Emmerich, P Wilson, J Brodholt, A.Thandavan, VN Alexandrov. Collaborative grid infrastructure for molecular simulations: The eMinerals minigrid as a prototype integrated compute and data grid. *Molecular Simulations* **31**, 303–313 (2005)

[4] MT Dove, E Artacho, TO White, RP Bruin, MG Tucker, P Murray-Rust, RJ Allan, K Kleese van Dam, W Smith, RP Tyer, I Todorov, W Emmerich, C Chapman, SC Parker, A Marmier, V Alexandrov, GJ Lewis, SM Hasan, A Thandavan, K Wright, CRA Catlow, M Blanchard, NH de Leeuw, Z Du, GD Price, J Brodholt, M Alfredsson. The eMinerals project: developing the concept of the virtual organisation to support collaborative work on molecular-scale environmental simulations. *Proceedings of All Hands 2005* (ISBN 1-904425-53-4), pp 1058–1065, 2005

[5] RP Bruin, M Calleja, MG Tucker and MT Dove. Building and managing the eMinerals clusters - a case study in grid-enabled cluster operation. *Computers in Science and Engineering* **7**, 30–37, 2005

[6] TOH. White, MT Dove, RP Bruin, KF Austen, AM Walker, A Marmier, SC Parker, PA Couch, RP Tyer, IT Todorov, P Murray-Rust, AD Walkingshaw, DJ Wilson. Information delivery in computational mineral science: the eMinerals data handling system. Submitted to same volume.

[7] RP Bruin, TOH White, AM Walker, KF Austen, MT Dove, RP Tyer, PA Couch, IT Todorov, MO Blanchard. Job submission to grid computing environments. *Proceedings of All Hands 2006* (accepted)

[8] RP Tyer, PA Couch, K Kleese van Dam, IT Todorov, RB Bruin, TOH White, AM Walker, KF Austen, MT Dove, MO Blanchard. Automatic metadata capture and grid computing. *Proceedings of All Hands 2006* (accepted)

[9] TOH White, P Murray-Rust, PA Couch, RP Tyer, RP Bruin, IT Todorov, DJ Wilson, MT Dove, KF. Austen. Application and uses of CML within the eMinerals project. *Proceedings of All Hands 2006* (accepted)

[10] C Chapman, AM Walker, M Calleja, RP Bruin, MT Dove, W Emmerich. Managing scientific processes on the eMinerals minigrid using BPEL. *Proceedings of All Hands 2006* (accepted)

[11] MT Dove, LA Sullivan, AM Walker, K Trachenko, RP Bruin, TOH White, P Murray-Rust, RP Tyer, PA Couch, IT Todorov, W Smith, K Kleese van Dam. Anatomy of a grid-enabled molecular simulation study: the compressibility of amorphous silica. *Proceedings of All Hands 2006* (accepted)

[12] IT Todorov, W Smith, K Trachenko, MT Dove. DL POLY_3: new dimensions in molecular dynamics simulations via massive parallelism. *Journal of Materials Chemistry* **16**, 1911–1918, 2006

[13] TOH White, RP Tyer, RP Bruin. A lightweight scriptable web-based frontend to the SRB. *Proceedings of All Hands 2006* (accepted)