

Anatomy of a grid-enabled molecular simulation study: the compressibility of amorphous silica

Andrew M Walker¹, Martin T Dove^{1,2}, Lucy A Sullivan¹, Kostya Trachenko¹, Richard P Bruin¹, Toby OH White¹, Peter Murray-Rust³, Rik P Tyer⁴, Phillip A Couch⁴, Ilian T Todorov^{1,4}, William Smith⁴, Kerstin Kleese van Dam⁴

1. *Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EQ*
2. *National Institute for Environmental eScience, University of Cambridge, Downing Street, Cambridge CB2 3EQ*
3. *Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW*
4. *CCLRC, Daresbury Laboratory, Warrington, Cheshire WA4 4AD*

Abstract

We report a case study in grid computing with associated data and metadata management in which we have used molecular dynamics to investigate the anomalous compressibility maximum in amorphous silica. The primary advantage of grid computing is that it enables such an investigation to be performed as a highly-detailed sweep through the relevant parameter (pressure in this case); this is advantageous when looking for derived quantities that show unusual behaviour. However, this brings with it certain data management challenges. In this paper we discuss how we have used grid computing with data and metadata management tools to obtain new insights into the behaviour of amorphous silica under pressure.

Introduction

It is now well-established that grid computing comes into its own in the physical sciences when it enables simulation studies to be carried out across a sweep of the input parameters. Examples might be studies of a system as a function of external conditions such as temperature or pressure. Whilst the existence of a grid of computers facilitates the parallel running of many separate simulations, to make effective use of the potential of grid computing it is essential to have appropriate workflow and data management tools. In this paper we report on a case study that has used a set of tools developed within the *eMinerals* project.

The particular case concerns a study of the properties of amorphous silica (SiO₂) as a function of pressure. Our interest concerns the way that volume varies with pressure. In almost all materials, relative volume changes become smaller with increasing pressure, which is equivalent to the statement that most materials become stiffer under pressure. Usually this can be explained by the fact that the atoms are being

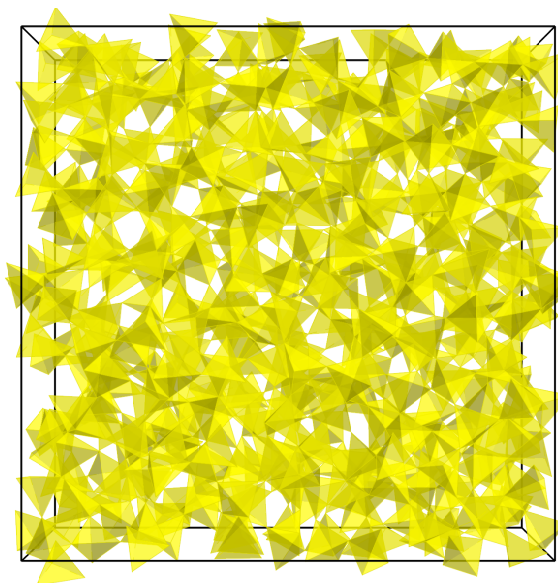
squeezed closer together, and the closer they are the stiffer the structure. However, amorphous silica behaves differently. On increasing pressure, amorphous silica initially becomes softer, until it crosses over to normal behaviour [1]. Formally the stiffness is defined by the inverse of the compressibility, κ^{-1} , where

$$\kappa = -V^{-1}(\partial V/\partial P).$$

Here V is the volume, and P is the pressure. In most materials, κ decreases on increasing pressure, but in amorphous silica κ has a maximum at a pressure of around 2 GPa.

Our approach is to use the classical molecular dynamics simulation method to study the pressure-dependence of amorphous silica. Because we need to calculate a differential, it is important to obtain a large number of data points on the volume/pressure graph, and it is in this regard that grid computing plays an important role. Using a grid enables the many separate jobs to be run at the same time, increasing the throughput by more than an order of magnitude so that collecting many data points becomes a viable process.

Figure 1. Configuration used in the simulations described in this paper, with SiO₄ polyhedra represented as tetrahedra rather than representing the individual atoms.



In this work we make use of the following technologies:

- ▶ Methods to create and submit many jobs in which one or more parameters are varied [2];
- ▶ Metascheduling within a minigrad compute environment, with jobs distributed over clusters and Condor pools [2,3];
- ▶ Use of the San Diego Storage Resource Broker (SRB) for data archiving and the sharing of data files [4];
- ▶ Use of XML output data and associated tools to aid data analysis and sharing of the information between collaborators [5];
- ▶ Incorporation of workflows within the job submission procedure to enable analysis to be performed on the fly [2];
- ▶ Automatic metadata capture using the recently-developed RCommands [6].

The purpose of this paper is to describe how these tools were combined to facilitate a detailed molecular dynamics simulation study of the compressibility of amorphous silica using grid computing.

Science background

Amorphous silica, SiO₂, is a random network of corner-linked SiO₄ tetrahedra, Figure 1. We work with configurations of 512 tetrahedra generated from initial configurations of amorphous elemental silicon [7] and tested against neutron total scattering data [8].

The issue of compressibility concerns the inherent flexibility of the network of connected tetrahedra. This is a subtle issue, because standard engineering methods of counting constraints and degrees of freedom do not capture the whole story. We have previously demonstrated [7] that the silica network has an inherent network flexibility in which the SiO₄ tetrahedra can rotate and buckle the network without the tetrahedra themselves needing to distort. Such motions will cost relatively little energy; the higher-energy processes are those that cause the SiO₄ tetrahedra to distort, either through bending of the O–Si–O bond angles or stretching of the Si–O bonds. There are two ways in which buckling of the network of corner-linked SiO₄ tetrahedra can happen. One is through fast vibrations, and the other is

through larger jump motions in which several tetrahedra change their orientations together. Animations of both processes are available from references 9 and 10 respectively, and are surprisingly instructive.

Our approach is to consider the behaviour of amorphous silica in the two extremes of large negative and positive pressures in comparison with intermediate pressures. First we note that the compressibility, as defined earlier, can also be defined in terms of the second derivative of the free energy G :

$$\kappa = -V^{-1}(\partial^2 G / \partial P^2).$$

Thus compressibility is related to changes in energy, and our hypothetical extreme end states are both states in which any changes are necessarily accompanied by large changes in energy as compared to the intermediate state. At large negative pressures (corresponding to stretching the material) the bonds are themselves stretched tight and the flexibility of the network is accordingly reduced. To change the pressure in this extreme will involve distorting the SiO₄ tetrahedra – either by changing bond lengths or bond angles – which as noted above is quite a high energy process. At the high-pressure extreme, atoms are pushed tightly together and further changes in volume can again only be accomplished by distorting the SiO₄ tetrahedra. But in the intermediate region, where there is more flexibility of the network, volume changes can be accomplished by crumpling the network without any distortions of the SiO₄ tetrahedra. Since this is a low energy process, the compressibility is a lot higher.

The task we set ourselves was to demonstrate the reasonableness of this hypothesis, and the chosen tool is molecular dynamics simulation. We have two good sets of interatomic potential energy functions for silica, both based on quantum mechanical calculations of small clusters of silicon and oxygen atoms; these are described in references 11 and 12 respectively. For the present paper we will only present results using the model of reference 11, but will refer to the other set of results later. We use the DL_POLY_3 molecular dynamics simulation code [13], which has been adapted for working within a grid computing environment.

We ran simulations for pressures between ± 5 GPa and at a temperature of 50 K. Our aim was to capture data for many pressures within this range, and to analyse the resultant configurations at the same time as running the simulations. At each pressure we ran one simulation with a constant-pressure and constant-temperature (*NPT*; *N* implies a constant number of atoms) algorithm in order to obtain data for the equilibrium volume, followed by a simulation using a constant-volume and constant-energy (*NVE*) algorithm for the more detailed analysis (see later in this paper). The analysis was performed using additional programs, and was incorporated within the workflow of each job carried out within the grid computing environment.

eScience methodology

Grid computing environment

The simulations were performed on the combination of the *eMinerals* minigrid [3,14], which primarily consists of linux clusters running PBS, and CamGrid [15], which consists of flocked Condor pools. Although the *eMinerals* minigrid and CamGrid are independent grid environments, they have overlapping resources, and the tools developed to access the *eMinerals* minigrid [3,14] have been adapted to work on CamGrid, and, incidentally, to also enable access to NGS resources, in the same manner.

Access to the *eMinerals* minigrid is controlled by the use of *escience* digital certificates and the use of the Globus toolkit. We have developed a metascheduling job submission tool called `my_condor_submit` (MCS) to make the process easier for end users [2,3,15]. This uses the Condor-G interface to

Globus to submit three separate jobs per simulation. The first job takes care of data staging from the SRB to the remote computing resource, whilst the last job uploads output data back to the SRB as well as capturing and storing metadata. The middle job takes care of running the actual simulation and corresponding analysis as part of a script job.

Data management

Data management within the *eMinerals* minigrid is focussed on the use of the San Diego Storage Resource Broker. The SRB provides a good solution to the problem of getting data into and out of the *eMinerals* minigrid. However, it also facilitates archiving a complete set of files associated with any simulation run on the minigrid. The way that the *eMinerals* project uses the SRB follows a simple workflow:

1. The data for a run, and the simulation executable, are placed within the SRB. It is not necessary for the files to be within the same SRB collection.
2. A submitted job, when it reaches the execute machine, first downloads the relevant files.
3. The job runs, generating new data files.
4. At the end of the job, the data files generated by the run are parsed for key metadata which is ingested into the central metadata database, and the files are put into the SRB for inspection at a later time.

This simple workflow is managed by the MCS tool using Condor's DAGman functionality. We have built into the workflow a degree of fault tolerance, repeating any tasks that fail due to unforeseen errors such as network interruption.

Combinatorial job preparation tools

One of the key tasks for *escience* is to provide the tools to enable scientists to access the potential benefits of grid computing. If scientists are to run large combinatorial studies routinely, they need tools to make setting-up, submitting, managing and tracking of many jobs nearly as easy as running a single job. To enable the scientists within the *eMinerals* project team to run detailed combinatorial studies one of the authors (RPB) has developed a set of tools that generate and populate new data collections on the SRB for all points on the parameter sweep, and then generate and submit the set of MCS scripts [2].

All that is required from the user is to provide a template input file from which all

necessary unique input files will be created. The user specifies the name of the parameter whose value is to be varied, the start and end values, and the number of increments. For example, a user could specify that they wish to vary pressure between -5 and $+5$ GPa in 11 steps which would result in input files being created for pressures of $-5, -4, -3, \dots, +4, +5$ GPa. The tools then create a simple directory structure, with each sub-directory containing the necessary input files. A similar directory structure will also be created on the SRB. Each directory on the submission machine also contains the relevant MCS input script, with appropriate SRB calls and metadata commands.

The subsequent stage is job submission. The user runs one command, which walks through the locally created directory structure and submits all of the jobs using MCS. It is MCS that takes care of the issue of deciding where to run the simulation within the *eMinerals* minigrid or Camgrid environments, using a metascheduling algorithm explained elsewhere [2]. This algorithm ensures that the user's simulations do not sit in machine queues for longer than is necessary, and that the task can take full advantage of all available resources.

The tools also track the success of the job submission process, and any errors that occur as part of the submission are recognised. For example, if one of the Globus gatekeepers stops responding, it will cause Condor-G to fail in the submission. This will be noticed, and a user command will provide information on all failed jobs and will resubmit them as appropriate.

Workflow within the job script

Usually MCS is used to submit a binary executable, but can also submit a script (eg shell or Perl) containing a set of program calls. In our application, a standards-compliant shell script is used to control the execution of a number of statically linked executables and a simple Perl script to run the analysis. In detail the script runs the following codes in order on the remote host: first DL_POLY_3 is run with the *NPT* ensemble in order to generate a model with the appropriate density for the pressure of interest, then the output of this run is used as input for a second DL_POLY_3 run in the *NVE* ensemble in order to sample the vibrational behaviour of the system at this density. Following the second molecular dynamics simulation, pair distribution functions are extracted for the Si-Si, Si-O and O-O

separations, and configurations are extracted for analysis of the atomic motions based on a comparison of all configurations. Finally a Perl script collates the results of this analysis for later plotting. This analysis will not be reported in this paper, but is mentioned here to make the extent of the workflow clear.

Use of XML

The *eMinerals* project has made a lot of use of the Chemical Markup Language (CML) to represent simulation output data [5]. We have written a number of libraries for using CML within Fortran codes (most of our simulation codes are written in Fortran), and most of our simulation codes now write CML. We typically write three blocks of CML, one for "standard" metadata (some Dublin Core items, some specific to the code version and compilation etc), one to mirror input parameters (such as the input temperature and pressure, technical parameters such as cut-off limits), and one for output properties (including all computed properties step by step and averages over all steps). This is illustrated in Figure 2.

The XML files stored within the SRB are transformed to HTML files using the TobysSRB web interface to the SRB, with embedded SVG plots of the step-by-step output [16]. This enables the user to quickly inspect the output from runs to check issues such as convergence.

XML output also allows us to easily collect the desired metadata related to both input and output parameters using the AgentX library, developed by one of the authors (PAC) [17]. This has been integrated into the MCS workflow, enabling the user to easily specify the metadata they wish to collect as part of their job submission process without needing to know the ins and outs of XML file formats.

Results management

When many jobs are run as part of a single study, it is essential to have tools that collate the key results from each run. For example, one key output from our work will be a plot of volume against pressure, with each data point obtained from a single computation performed using grid computing. In this we exploit the use of XML in our output files, because the required averaged quantities can be accessed by retrieving the value of the relevant XML element as per Figure 2. This value can be retrieved for each of the individual files using a simple XSLT transform, combining all of the values together

```

<?xml version="1.0" encoding="UTF-8"?>
<cml xmlns="http://www.xml-cml.org/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:dc="http://purl.org/dc/elements/1.1/title"
  xmlns:dl_poly="http://www.cse.clrc.ac.uk/msi/software/DL_POLY/dict"
  xmlns:dl_polyUnits="http://www.cse.clrc.ac.uk/msi/software/DL_POLY/units">

<metadataList>
  <metadata name="dc:contributor" content="I.T.Todorov & W.Smith"/>
  <metadata name="dc:source"
    content="cclrc/ccp5 program library package, daresbury laboratory molecular dynamics
    program for large systems"/>
  <metadata name="identifier" content="DL_POLY version 3.06 / March 2006"/>
  <metadata name="systemName" content="DL_POLY : Glass 512 tetrahedra"/>
</metadataList>

<parameterList title="control parameters">
  <parameter title="simulation temperature" name="simulation temperature"
    dictRef="dl_poly:temperature">
    <scalar dataType="xsd:double" units="dl_polyUnits:K"> 5.0000E+01 </scalar>
  </parameter>
  <parameter title="simulation pressure" name="simulation pressure"
    dictRef="dl_poly:pressure">
    <scalar dataType="xsd:double" units="dl_polyUnits:katms"> -3.0000E+01 </scalar>
  </parameter>
  <parameter title="simulation length" name="selected number of timesteps"
    dictRef="dl_poly:steps">
    <scalar dataType="xsd:integer" units="dl_polyUnits:steps"> 50000 </scalar>
  </parameter>
</parameterList>

<propertyList title="rolling averages">
  <property title="total energy" dictRef="dl_poly:eng_tot">
    <scalar dataType="xsd:double" units="dl_polyUnits:eV_mol.-1"> -2.7360E+04 </scalar>
  </property>
  <property title="volume" dictRef="dl_poly:volume">
    <scalar units="dl_polyUnits:Angstroms.3">2.2316E+04</scalar>
  </property>
</propertyList>
<propertyList title="execution time">
  <property title="run time">
    <scalar dataType="xsd:double" units="dl_polyUnits:s"> 17475.422 </scalar>
  </property>
</propertyList>
</cml>

```

Figure 2. Example CML output from DL POLY showing the key data lists.

then results in a list of points. This can easily be plotted as a graph using a further XSLT transformation into an SVG file.

These transformations can be done very quickly, and more importantly they can be done automatically, which means that the user and his/her collaborators simply need to look at a graph in a web page to quickly analyse trends within the data, rather than having to open hundreds of files by hand to find the relevant data values to then copy into a graph plotting

package. In our experience, this is the sort of thing that makes grid computing on this scale actually usable for the end user, and facilitates collaborations.

Metadata

With such quantities of data, it is essential that the data files are saved with appropriate metadata to enable files to be understood and data to be located using search tools. We have developed tools, called the RCommands, which

MCS use to automatically collect and store metadata [6]. The metadata are culled from the CML output files, collected from the metadata, parameter and property lists. Property data such as the computed average volume are used as metadata because they will be parameters within the metadata against which users are able to run search commands on.

Other information, including metadata content regarding the execution machine and directory, as well as the default metadata created as part of any of our XML output file creation, are also captured. These items of metadata provide a valuable audit trail for each simulation, in part replacing the scientist's traditional log book with a much more searchable and efficient means of tracking their submitted simulations. Scientists cannot reasonably be expected to keep track of each of several hundred simulations performed in this way without efficient and automatic metadata capture, since doing so by hand would result in more time being spent recording this sort of information than in actually analysing the science identified by the results.

It should be noted that the use of the RCommands and the AgentX tools did not give a significant overhead to the running of the the jobs. The DL_POLY_3 simulations typically took 8 hours, whereas the metadata extraction only took 30 minutes or less.

Results

Volume curve

The first key result we were aiming at was a plot of volume *vs* pressure, which is show in Figure 3. What is clear from this diagram is that the slope of the graph is greatest for intermediate pressures, indicating that amorphous silica is softest around ambient pressures. The virtue of having many points is that we were able to fit a polynomial to the data in order to extract the derivative dV/dP with reasonable confidence. We plot the compressibility, $\kappa = -V^{-1}(dV/dP)$ in Figure 4. The maximum in the compressibility occurs at a pressure of around 1 GPa. This is a bit less than the experimental value (2 GPa), but given that this is a second-order effect, the difference between experiment and simulation is not

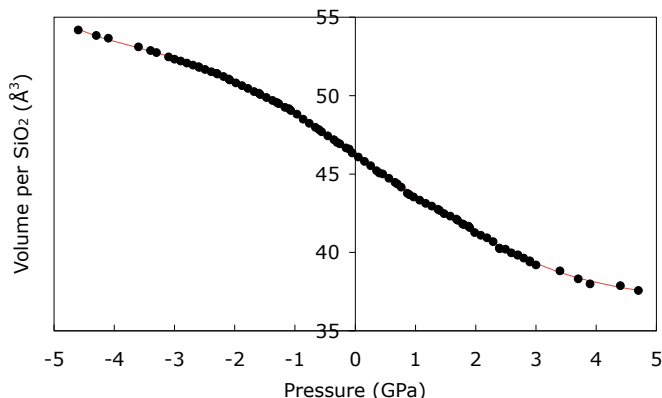


Figure 3. Pressure dependence of the volume (points) of amorphous silica, fitted with a polynomial (red curve).

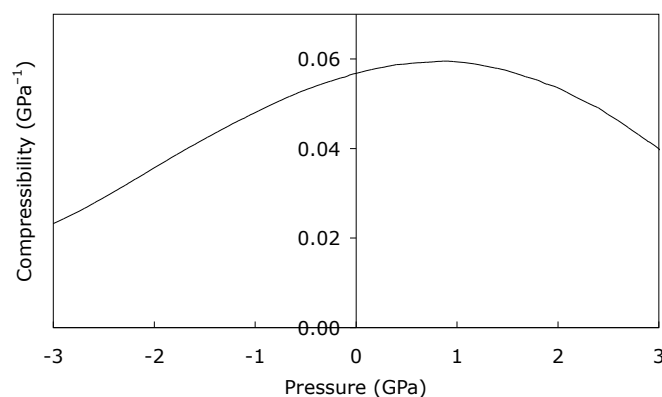


Figure 4. Compressibility of amorphous silica generated by differential of the fitted polynomial to the simulation volume.

significant. What is important from this plot is that we have successfully modelled the compressibility maximum in amorphous silica, and we note here that we have reproduced this result with the model interatomic potential of reference [12] as well. This implies that the compressibility maximum is not a subtle feature peculiar to the details of the interatomic potentials (real or model) but is a consequence of the nature of the structure and atomic connectivities.

Interatomic distances

The distribution of interatomic distances is described by the pair distribution function, $g(r)$, such that the number of pairs of atoms with separation between r and $r + dr$ is given by $4\pi r^2 g(r) dr$. $g(r)$ is thus a normalised histogram of interatomic distances, which for close neighbours is typically peaked around a well-defined mean separation. The mean separations for Si-O, O-O and Si-Si nearest neighbours are

plotted in Figure 5, normalised to unity at zero pressure to facilitate comparison. The Si–O and O–O distances are defined by the relatively-rigid SiO₄ tetrahedra – the ratio of the O–O to Si–O mean distance is equal to $(8/3)^{1/3}$ – and it can be seen that their mean distances barely change with pressure. Thus we see that the SiO₄ tetrahedra on average barely change their size or shape across the range of pressures. On the other hand, the mean Si–Si distance varies much more with pressure, almost scaling with the length scale of the simulation sample, i.e. the cube root of the sample volume (also shown in Figure 5). The pressure-dependence of the Si–Si distance suggests that the network is buckling with a folding of groups of connected tetrahedra.

The variances of the distributions of specific interatomic distances in $g(r)$ are shown in Figure 6. These show a number of interesting features. Most clear is that the Si–O distribution is very sharp (small variance); this reflects the fact that this is a very strong bond with a high frequency, and hence low amplitude, stretching vibration. The variance of the Si–Si distribution clearly increases on increasing pressure, consistent with buckling of the network. The variance of the O–O distribution is interesting. Although the mean distance (figure 5) varies no more than the mean Si–O distance, there is greater variation of the O–O distance. This means that all deformations of the SiO₄ tetrahedra mostly involve flexing of the O–Si–O bond angles. It is interesting to note that the variance of the O–O distances has a minimum around the pressure of the maximum in the compressibility.

Summary points

- ▶ We have seen how grid computing enables us to run many concurrent simulations, which enable us to obtain data with sufficient points to be able to extract derived quantities that show anomalous behaviour.
- ▶ Launching many jobs is a challenge that requires an automated solution. This work has been carried out using the parameter sweep tools developed within the eMinerals project.
- ▶ The actual mechanism of submitting the jobs to Globus resources and then managing the workflows, including interaction with the

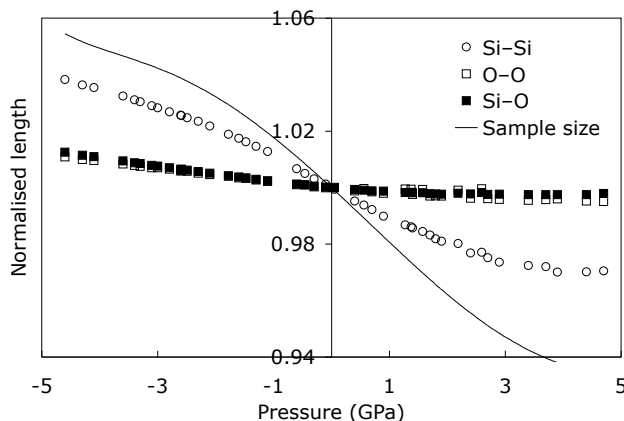


Figure 5. Comparison of the mean interatomic nearest-neighbour distances, normalised to unity at zero pressure in order to aid comparison of the dependence on pressure. The curve is the cube root of the volume, and represents the linear size of the simulation sample.

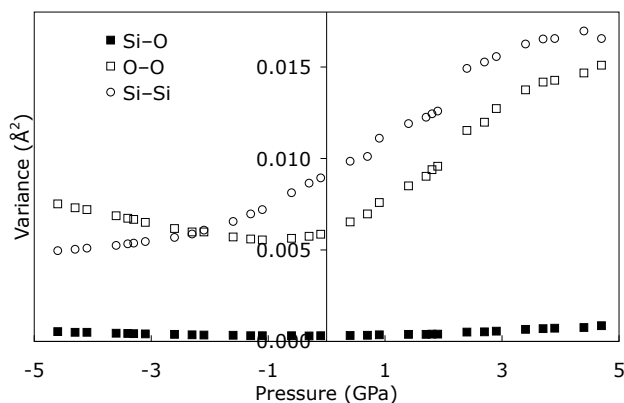


Figure 6. Pressure dependence of the variances of the distributions of nearest-neighbour interatomic distances.

SRB, was enabled using the MCS tool developed by the eMinerals project.

- ▶ Data management presented several challenges. The use of the SRB for data storage was already in use within the eMinerals project, and this case study showed the value of the SRB for data archiving.
- ▶ This work was greatly helped by the use of XML (CML) file outputs. Examples were the use of the TobysSRB tool to inspect file output stored in the SRB in XML format, the use of XML to gather key data from many data files stored in the SRB, and the use of the SRB in gathering metadata.
- ▶ With the large number of data files generated in this study, automatic metadata capture is essential. We have used the RCommand framework as used within the MCS tool together with the Rparse tool to add metadata associated with each set of files.

- ▶ This work was collaborative in nature, involving at times sharing of data between colleagues who were in physically different locations. Sharing of data was enabled using the above tools and methods.
- ▶ Finally, it is worth remarking that a large part of the simulation work reported in this study was carried out by a third year undergraduate student (LAS) as her degree project. We were confident, as proved to be the case, that the escience tools developed to support this work could easily be picked up by someone with no previous exposure to escience or computational science and who was working under some considerable time pressure.

Acknowledgements

We are grateful for funding from NERC (grant reference numbers NER/T/S/2001/00855, NE/C515698/1 and NE/C515704/1).

References

1. OB Tsiok, VV Brazhkin, AG Lyapin, LG Khvostantsev. Logarithmic kinetics of the amorphous-amorphous transformations in SiO₂ and GeO₂ glasses under high-pressure. *Phys. Rev. Lett.* **80**, 999–1002 (1998)
2. 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*
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. *Mol. Simul.* **31**, 303–313 (2005)
4. RW Moore and C Baru. Virtualization services for data grids. in *Grid Computing: Making The Global Infrastructure a Reality*, (ed Berman F, Hey AJG and Fox G, John Wiley), Chapter 16 (2003)
5. TOH White, P Murray-Rust, PA Couch, RP Tyer, RP Bruin, MT Dove, IT Todorov, SC Parker. Development and Use of CML in the eMinerals project. *Proceedings of All Hands 2006*
6. R.P.Tyer, P.A. Couch, K Kleese van Dam, IT Todorov, RP Bruin, TOH White, AM Walker, KF Austen, MT Dove, MO Blanchard. Automatic metadata capture and grid computing. *Proceedings of All Hands 2006*
7. KO Trachenko, MT Dove, MJ Harris, V Heine. Dynamics of silica glass: two-level tunnelling states and low-energy floppy modes. *J Phys.: Cond. Matt.* **12**, 8041– 8064 (2000)
8. MG Tucker, DA Keen, MT Dove, K Trachenko. Refinement of the Si–O–Si bond angle distribution in vitreous silica. *J Phys.: Cond. Matt.* **17**, S67–S75 (2005)
9. <http://www.esc.cam.ac.uk/movies/trid4t.html>
10. <http://www.esc.cam.ac.uk/movies/glass1.html>
11. S Tsuenyuki, M Tsukada, H Aoki, Y Matsui. 1st principles interatomic potential of silica applied to molecular dynamics. *Phys. Rev. Lett.* **61**, 869–872 (1988)
12. BWH van Beest, GJ Kramer, RW van Santen. Force fields for silicas and aluminophosphates based on *ab initio* calculations. *Phys. Rev. Lett.* **64**, 1955–1958 (1990)
13. IT Todorov and W Smith. DL_POLY_3: the CCP5 national UK code for molecular-dynamics simulations. *Roy. Soc. Phil. Trans.* **362**, 1835–1852 (2004)
14. M Calleja, L Blanshard, R Bruin, C Chapman, A Thandavan, R Tyer, P Wilson, V Alexandrov, RJ Allen, J Brodholt, MT Dove, W Emmerich, K Kleese van Dam. Grid tool integration within the eMinerals project. *Proceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 812–817
15. M Calleja, B Beckles, M Keegan, MA Hayes, A Parker, MT Dove. CamGrid: Experiences in constructing a university-wide, Condor-based grid at the University of Cambridge. *Proceedings of the UK e-Science All Hands Meeting 2004*, (ISBN 1-904425-21-6), pp 173–178
16. TOH White, RP Tyer, RP Bruin, MT Dove, KF Austen. A lightweight, scriptable, web-based frontend to the SRB. *Proceedings of All Hands 2006*
17. PA Couch, P Sherwood, S Sufi, IT Todorov, RJ Allan, PJ Knowles, RP Bruin, MT Dove, P Murray-Rust. Towards data integration for computational chemistry. *Proceedings of All Hands 2005* (ISBN 1-904425-53-4), pp 426–432