

Experiences with different middleware solutions on the NW-GRID

Jens M. H. Thomas, John Kewley, Rob J. Allan,
Jamie M. Rintelman, Paul Sherwood, C.L. Bailey,
A. Wander, B.G. Searle and N.M. Harrison
STFC Daresbury Laboratory, Warrington WA4 4AD, UK

S. Mukhopadhyay
Computational Material Science Group, Department of Chemistry,
Imperial College London, South Kensington Campus,
London, SW7 2AZ

Abbie Trewin, George R. Darling and Andrew I. Cooper
Department of Chemistry, University of Liverpool,
Crown Street, Liverpool L69 3BX, UK

Corresponding author: j.m.h.thomas@dl.ac.uk

Abstract

Grid computing is becoming an increasingly important way for computational scientists to access computer resources. However, there is currently no consensus on the optimum middleware for providing access to these resources, or even a uniform way for users to interact with a particular middleware solution. As part of our work on the NW-GRID project, and in our development of our CCP1 GUI software, we have had cause to use two of these solutions (Globus and NorduGrid ARC), together with various client-side tools that are under development to support them. This paper describes our experiences to date with these different solutions, and work that has been carried out using some of them.

1 Introduction

The problem of linking disparate computational resources from different administrative domains, and providing a reasonably simple interface for users to access the resources is a highly active area for research. A number of different solutions are being developed, each with slightly different objectives and emphases on the aspects of the problem they are trying to solve. The ease of use of these technologies still leaves a lot to be desired and various projects are seeking to address this in different ways. Through our involvement in the NW-GRID project [1] we have either been involved ourselves, or have helped other users access Grid resources using these technologies. In this paper, we describe our experiences with them and science that was able to be undertaken using them. We also describe our efforts to simplify their usage by interfacing them to our CCP1 GUI code. For information about the UK Collaborative Computational Projects (CCPs), see <http://www.ccp.ac.uk>.

2 Middleware solutions

Probably the most well-known Grid middleware toolkit is the Globus Toolkit, developed by the Globus Alliance [2]. The Globus Toolkit is an open source technology for Grid computing, letting people share computing power, databases, and other resources securely online across corporate, institutional and geographic boundaries without sacrificing local autonomy. The toolkit includes software

services and libraries for resource monitoring, discovery, and management, plus security and file management.

The Globus toolkit has been used successfully on a large number of computational Grids world wide, and has been installed on the NW-GRID, which was the target for most of the work described in this paper. We are currently using GT 2.4.3 or the equivalent functionality from the pre-Web services GT 4.0.1.

The Nordugrid [3] is a Grid research and development collaboration aiming at development, maintenance and support of the free Grid middleware known as the Advance Resource Connector (ARC). The Nordugrid collaboration is centered around Scandinavia, but has collaborators as far afield as Russia, Australia and Sri Lanka.

Nordugrid ARC provides a reliable implementation of the fundamental Grid services, such as information services, resource discovery and monitoring, job submission and management, brokering and data management and resource management. Most of these services are provided through the security layer of the GSI (Grid Security Infrastructure). The middleware builds upon standard Open Source solutions like the OpenLDAP, OpenSSL, SASL and the Globus Toolkit (GT) libraries.

The technologies listed above, however, are used as replacements and extensions of the original pre-Web services of the Globus Toolkit. ARC does not use most GT services, such as GRAM, job submission commands, the WUftp-based gridftp server, the gatekeeper, GRAM job-manager scripts, MDS information providers or schemas. ARC extends Globus RSL and makes the Globus MDS

functional without altering the original code. For this reason ARC should be considered to be a Grid solution in its own right and not just a client for the Globus toolkit services.

3 Client Interfaces

The Globus Toolkit provides a number of command-line client tools for submitting and monitoring jobs and staging data to and from the resource (e.g. `gsisscp`, `globus-job-submit`) and this is the most common way to access Grids such as the UK's National Grid Service (NGS) or the NW-GRID that run the pre-Web services version of Globus. However, there are a number of issues with the client-side tools as provided, namely:

- There is currently no port of the Globus Toolkit for Windows and unfortunately this is still popular with many end-users who are not developers.
- The use of the command-line tools is rather complicated and involved, and requires manually staging files to and from resources, and building up complicated command-lines containing Globus Resource Specification Language (RSL) to submit jobs. It also requires re-typing certain parameters like the long job-contact string.
- The tools require certain firewall ports to be open on the client for callbacks, and this is often either difficult or impossible to arrange as most desktop client systems are considered by site administrators to be insecure.
- The handling and reporting of errors encountered during submission and runtime on the remote machine leaves a lot to be desired. Some of the reported error numbers require interpretation based on experience and errors can be caused by firewall issues, problems with proxies or problems with the GASS cache.
- Installing the client tools is often rather difficult, may require root privileges and usually requires a sizeable download which can also include a lot of non-client code or replace already-existing system packages such as OpenSSL.

The Nordugrid ARC software supplies a similar set of command line tools, which differ from their standard Globus counterparts as follows:

- The client does not require non-standard ports to be opened. It uses SOAP over HTTP as a wrapper for GSI or other protocols. This is done using the C/C++ `gSOAP` package [4] and a series of "connectors" are provided.
- The client functionality includes a resource broker which can automatically determine the optimal resource to submit to (the selection of the resource is enabled through extensions to Globus RSL, so unfortunately the ARC command-line is even longer than the standard Globus one).

- on encountering an error it is possible to download a directory that contains various control data enabling accurate *post mortem* analysis of the job.
- The client-side software is very lightweight and easily installed via RPMs, although as the ARC software builds on top of the Globus toolkit, it also suffers from the restriction that it will not run on Windows systems.

Many of these issues were noted in the paper by Chin and Coveney [5].

4 Working around the Limitations of the Client Interfaces

The issues raised above concerning the client-side tools for the different middleware options have caused developers of applications that run on computational Grids to explore alternative ways of accessing the resources. Two of these are explored briefly below.

4.1 Application Hosting Environment

AHE, the Application Hosting Environment [6] is a lightweight hosting environment for running unmodified applications that has been developed as part of the Reality Grid project [7]. The AHE provides a client/server architecture, with a very lightweight and firewall-friendly Java client (command-line scripts and a GUI) that can be deployed on most platforms, together with a more heavy-duty server that installs as part of the software stack from OMII, the UK's Open Middleware Initiative Institute [8].

Applications that are installed on various Grid resources are deployed as Web services on the AHE server, so the client need only use standard Web protocols to access the desired application. The server then handles the staging of data and the submission to the Grid resource, with the client periodically polling the server to receive updates on the status of the jobs. As access to the remote resource is mediated by the server, there is no need for any firewall ports to be open between the client and remote resource. Furthermore, as the AHE server supports accessing Sun Grid Engine, Condor and Unicore resources, in addition to both pre and post Web-services Globus, the same lightweight client has the potential access to a large number of diverse Grid resources.

The disadvantages of the AHE are that it currently lacks support for automatic resource discovery (as offered by Nordugrid) and that, although the server installation has been considerably simplified by incorporation into the OMII stack, it is still rather complicated to install. However we assume the server side would usually be managed by a competent Unix system administrator.

The AHE is currently being investigated as part of the NW-GRID effort. We are in the process of installing the server stack and will then use this to provide access to the GAMESS-UK molecular modelling code as a first

test. We will then look at extending this to provide access to the other codes that are hosted on the NW-GRID.

4.2 GROWL Scripts

The GROWL Project [9] set out to try and alleviate some of the problems with the pre-WS Globus client interface described above by the use of Web Services (via gSOAP) and a three-tiered architecture (similar to the approach adopted by the AHE). As part of this effort, a series of shell scripts were produced to wrap a simple subset of the Globus GT2 standard client commands, adding value and simplifying the interface to the user. Functionality enabled by use of the scripts includes:

- Incoming firewall ports need not be opened on the Client.
- The scripts provide extra transparency by discovering the type of job manager, user's home directory and paths of executables requested by the user.
- GROWL Scripts build quickly on platforms supported by VDT (a Grid middleware distribution which includes Globus and Condor, the server side of which is used on NGS and NW-GRID), and more slowly on many other linux/ unix platforms (on these platforms a source code version of the Globus Grid middleware client is downloaded and built).
- a potential restriction of GROWL Scripts is that they rely on firewall ports being open on the Grid resource for both remote login (using GSI-SSH), file transfer (using GSI-SCP) and job submission (using GRAM).

As with all Globus-based tools, GROWL Scripts are not currently supported on Windows.

Despite the limitations of the GROWL Scripts, they have proved very useful in enabling users of the NW-GRID to access the resources from their desktop workstations. Work studying Aluminum Fluorides that made use of the GROWL scripts to access the NW-GRID is described further below. A further advantage of the scripts is that the simplified and packaged middleware installation procedure can enable an intermediate Globus-enabled server to be set up with little effort.

5 The CCP1 GUI

The tools that have been described so far are focused entirely on the process of running a generic computational job, with little regard for what the job may involve. They therefore present an implicit barrier to a computational scientist interested only in running a calculation and getting the results. The running of the compute job now becomes a separate and potentially complex task, which, depending on the client used, may require significant knowledge of the underlying technologies and systems being used.

In order to address this, we have interfaced our CCP1 GUI software[10] to several of these clients, in order to hide some of the complexity of the underlying operations from the user and provide as seamless an integration as possible of the access to Grid resources into the users' normal working environment.

5.1 Background to the CCP1 GUI

The CCP1 GUI project arose as a result of demand within the UK academic community (principally through CCP1) for a free, extensible Graphical User Interface (GUI) for community codes, particularly for teaching purposes. The CCP1 GUI is designed to present a uniform interface to a range of different application codes (such as GAMESS-UK, Dalton and Molpro).

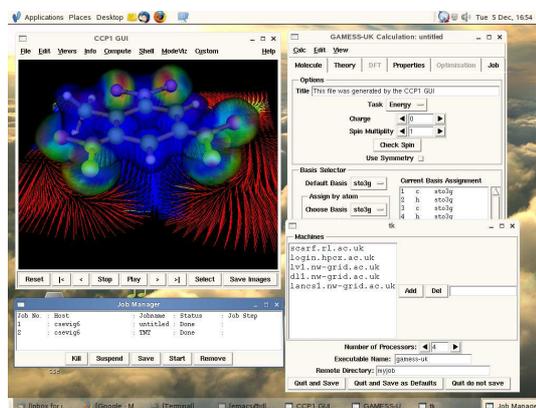


Figure 1: A screenshot of the CCP1 GUI

It currently enables users to build molecular structures (or import them from a range of different file formats), create input files for the different supported codes, invoke the code on the input, and then visualise the results from within the same environment.

The CCP1 GUI has been built around the Python open-source programming language [11] and uses the powerful VTK visualisation toolkit [12]. As both of these technologies are available for free and cross-platform, the CCP1 GUI will run on most platforms in use today, and the use of an object-oriented scripting language like Python, means that development time is relatively quick.

The use of VTK provides the CCP1 GUI with a highly powerful graphics toolkit for visualising molecules, as well as scalar and vector data – an essential for examining the complex results of (particularly) quantum chemical calculations.

5.2 The CCP1 GUI and GROWL/ Globus

As a first step towards enabling grid job submission for the CCP1 GUI, extensive use was made of the GROWL scripts for copying the files to and from the remote machine, submitting the job and polling for status.

Subsequently, the CCP1 GUI's job submission library has been extended to incorporate most of the functionality provided by the GROWL scripts, removing GROWL as a dependency, and permitting the CCP1 GUI to take advantage of any underlying client-side installation of Globus (although the GROWL installation is still the recommended way to install the client and certificates, and handle the conversion of Grid certificates).

With the current interface to the Globus client tools, the CCP1 GUI presents an interface to the user whereby the only aspects of the job they need concern themselves with are:

- which remote resource to run on;
- where the binary lives and where the calculation should run;
- whether any environment variables etc. need to be set on the remote machine in order to enable the job to run;
- in some cases it may be necessary to specify the jobmanager on the remote resource.

Once this has been set, the CCP1 GUI will then stage data to and from the resource, submit and monitor the job, and copy the results back to the users' local machine without any additional intervention from the user.

In practice, it was found that even remembering the location and environment required on each machine was excessively onerous and a barrier to effective work. The CCP1 GUI was therefore modified to use an internal database that stores the data for a particular code on a remote resource, so that this data need only be input once.

The Globus interface to the CCP1 GUI is currently being used by chemists studying novel Hydrogen storage materials as described further below.

5.3 The CCP1 GUI and Nordugrid

One of the codes that the CCP1 GUI provides an interface to is the Dalton quantum chemistry program [13]. As Dalton originates from Scandinavia, Dalton was already installed on many of the Nordugrid machines, which are required to be accessed through the Nordugrid ARC client. The CCP1 GUI jobmanager was therefore extended to work with the Nordugrid ARC software.

In many ways this was much easier than interfacing directly to the Globus toolkit. Although ARC does provide a set of command-line tools, the client software itself is written in C++ and automatic wrapping with SWIG [14] provides a Python language interface to the code. This permits much finer access to the underlying functionality than can be accessed through a command-line tool, and also facilitates error checking at each stage of the calculation.

The ARC interface to the CCP1 GUI is somewhat richer than the Globus one, because of the additional functionality presented by the extended RSL (named xRSL) used in ARC. So, for example, although a user need not select a particular cluster to run a job on (as the

resource broker will determine one), it is possible to specify requirements for a job, such as memory and runtime and then provide ARC with a list of machines so that the resource-broker will select the most suitable one.

The Nordugrid interface to the CCP1 GUI is currently being tested by developers of the Dalton code for production use.

6 Science carried out on the NW-GRID using Different Middleware Solutions

6.1 Modelling polymers for hydrogen storage

Abbie Trewin, George R. Darling and Andrew I. Cooper.

The widespread use of hydrogen as a fuel is limited by the lack of a convenient method of H₂ storage [15]. Molecular H₂ storage by physisorption is appealing because it is reversible, cyclable, and sorbents exist which are tolerant to minor impurities such as water. Many porous sorbents have been investigated including carbon [16], metal-organic frameworks (MOFs) [17], and organic polymers [18]. Two significant drawbacks to H₂ physisorption, however, are the low temperatures required and the associated system weight implications. To date, most H₂ physisorption experiments have been conducted at 77.3 K because of the low average isosteric heat of sorption (4-7 kJ/mol) of H₂ with most materials [19]. Hence, there is a need to consider new H₂ sorbents with substantially higher average isosteric heats which might store H₂ at more practicable temperatures.

Molecular H₂ can interact with a porous substrate via weak dispersive interactions, electrostatic interactions, orbital interactions, or in non-classical dihydrogen complexes [20]. Dispersive forces dominate in substrates such as carbon [16], and porous organic polymers [18]. Electrostatic interactions between H₂ and charged sites are typically stronger; for example, a recent report refers to a maximum isosteric heat range of sorption (at low H₂ coverage) of 10.1 kJ/mol for exposed Mn²⁺ coordination sites in a microporous MOF [21]. Lochan and Head-Gordon [20] have calculated much higher H₂ binding affinities in the broad range 10-350 kJ/mol for charged metals (e.g., Li⁺, Al³⁺), ligands (e.g., SO₄²⁻, F⁻), and metal complexes (e.g., M[(CO)_x]^{y+}). Other studies have calculated the interaction energies for F⁻-(H₂)_n (n = 1-8) anion complexes using quantum chemical calculations at the MP2/ aug-cc-pVTZ level [22]. It is, however, impossible to achieve isolated "bare" ions such as Al³⁺ or F⁻ in real physical materials [20].

Our broad aim is to design and evaluate synthetically-viable high-energy "binding sites" for hydrogen physisorption which can be incorporated into real materials.

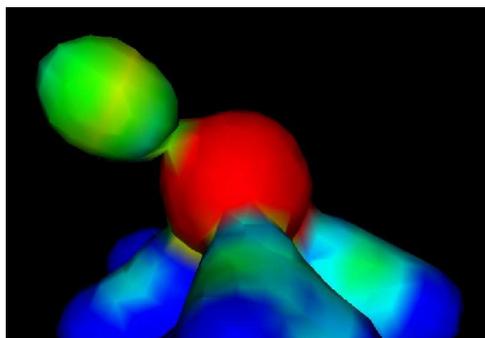


Figure 2: The interaction between an F anion and its surrounding framework structure visualised with the CCP1 GUI

Ab initio calculations employing the GAMESS-UK code are used to investigate the binding interactions between molecular H_2 and a range of molecules. The calculation of binding energies requires several separate calculations at high computational expense.

The computing facilities made available using the NW-GRID have enabled us to calculate systems which would otherwise be prohibitively expensive. For example, we are able to investigate the effects of the surrounding structural environment as well as the binding site locality. This is an important factor as the binding interaction forces are generally weak and easily influenced by the materials framework structure.

The use of the CCP1 GUI provided a simple environment to generate multiple jobs easily and efficiently, and then visualise the resulting data (see Figure 2). As the CCP1 GUI also handled all aspects of running the calculation on the NW-GRID, the calculations were able to be run by users with no knowledge of the underlying technologies being used to access the Grid resources.

6.2 *Ab initio* studies of aluminium fluorides

C.L. Bailey, S. Mukhopadhyay, A. Wander, B.G. Searle and N.M. Harrison.

Aluminium fluorides (AlF_3) have great potential for use as catalysts in many Cl/F exchange reactions. Experimentally, large differences in catalytic reactivity are observed between the different phases of AlF_3 . A series of detailed first principles simulations has started to unravel the complex structures of these surfaces which in turn provides clues to their different chemical properties. Little is known experimentally about the detailed atomic scale structure of these surfaces and consequently these calculations provide the only model that explains this catalytic activity.

These results allow us to explain the different reactivity of aluminium fluoride phases in terms of the substantial differences in the accessibility of the reactive, co-ordinatively unsaturated aluminium sites at their respective surfaces. For crystalline α - AlF_3 we have shown that the Al atoms are effectively covered by flu-

orine atoms [24, 25] leading to a surface that displays no Lewis acidity or catalytic activity. Conversely, we have shown that β - AlF_3 , which shows moderate Lewis acidity and catalytic activity, contains co-ordinatively unsaturated five fold aluminium ions at the surface [24, 26]. On the basis of these results we can postulate that the highly reactive, amorphous “high-surface area” AlF_3 may display acidic sites similar to those found on the β - AlF_3 surface.

It is known that the surfaces of AlF_3 strongly hydrolyse water and consequently it is likely that the surfaces of real operating catalysts will contain significant quantities of hydroxyl groups. To quantify this, a series of calculations have been performed allowing us to predict the surface chemistry and structure of these materials as a function of environmental conditions – namely the partial pressure of water and hydrogen fluoride the surface is exposed to. The surface stability of the (0001) surface of α - AlF_3 is shown in Figure 3 [27]. These predictions are currently awaiting experimental verification although they appear to explain all known experimental data.

This project requires a large amount of processor time. We typically run geometry optimisation calculations in parallel on up to 32 processors. A single calculation typically takes around 60 hours. We have been able to use NW-GRID to run a significant proportion of these calculations. For example, the surface structure of α - AlF_3 as a function of HF and H_2O partial pressure, shown in Figure 3, involved over 150 different geometry optimisations using hybrid exchange density functional theory as implemented in the CRYSTAL [28] code. We have made an extensive use of the graphical user interface DLV [29] within this study. DLV is analogous to the CCP1 GUI described earlier, and provides a powerful interfaces to many of the CCP3 codes such as CRYSTAL and DL-EXCURVE.

DLV has been used to both set up our calculations and visualise the resultant structures and properties of the systems. The latest development version of DLV can also be used to remotely submit jobs to external clusters such as NW-GRID, using the GROWL interface.

The ongoing development of different middleware has led to a project to extend the capability of DLV to allow further integration with NW-GRID. We are currently considering extending DLV to enable multiple jobs to be quickly and easily submitted, with the results written back to a local machine for analysis. We also look forward to the opportunities arising from developments around co-scheduling and job-migration to allow jobs to be remotely submitted to NW-grid to be run on whichever cluster has resources available. Loosely coupled jobs could even be run over clusters over multiple sites.

7 Conclusions

The idea of Grid computing is becoming increasingly popular and is starting to impact on scientists in a growing range of different disciplines. However, Grid comput-

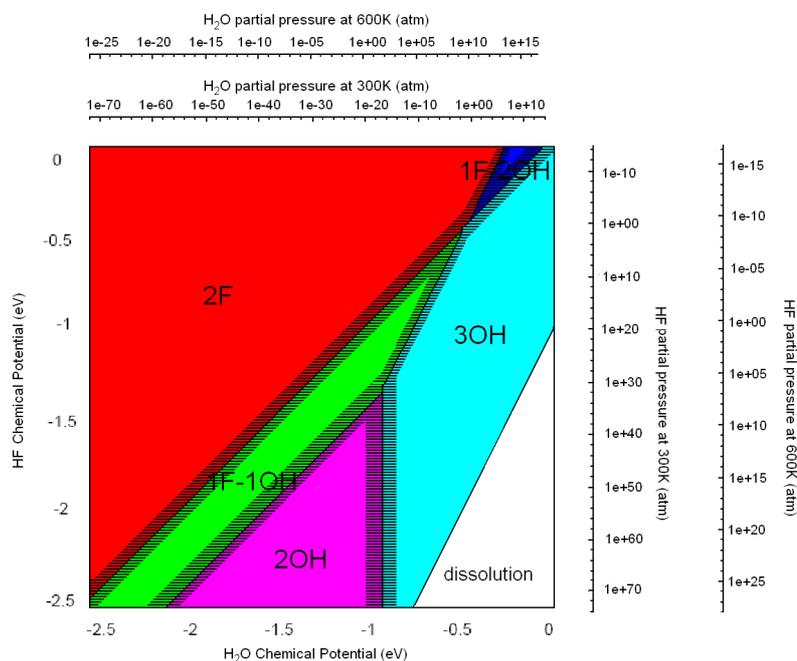


Figure 3: Phase diagram for $\alpha\text{-AlF}_3(0001)$. The coloured regions represent the most stable surface terminations that will exist as a function of gaseous HF and H_2O partial pressures.

ing is still very much in its infancy in terms of “ease of use” and there is no general consensus on the best way to approach it. Middleware solutions such as Globus and the Nordugrid ARC software are becoming more mature and effective, but their uptake still presents a significant hurdle to mainstream scientists interested in taking advantage of the resources becoming available on computational Grids.

Solutions such as the AHE and GROWL ease this problem somewhat by buffering the users from some of the underlying complexity inherent in Grid middleware. These tools are in active use and are now helping scientists do “real” science as demonstrated in this paper, but generally in the context where the scientists are already very close to the providers of the Grids or the software developers who are then able to provide the support and training needed to familiarise the users with this new technology. Although this works for a small number of people, it is not a viable scenario if the vision of Grid computing is to be realised more widely.

The work that has been done with the CCP1 GUI demonstrates one approach that can help scientists with little or no interest in the underlying Grid technology make effective use of resources such as the NW-GRID. By building application-specific client tools on top of the supplied Grid toolkits, and thus presenting a familiar interface, users are more comfortable running on these resources, as is demonstrated by some of the work pre-

sented in this paper.

8 Acknowledgments

Some of the work described in this paper was undertaken at the STFC (formerly CCLRC) e-Science Centre, Daresbury Laboratory. Development of GROWL was supported by JISC. NW-GRID is funded by NWDA, the North West Development Agency.

We would like to thank the EU for the support of part of this work through the 6th Framework Programme (FUNFLUOS, contract No. NMP3-CT-2004-5005575).

References

- [1] NW-GRID: the North West Grid.
<http://www.nw-grid.ac.uk>
- [2] The Globus Alliance.
<http://www.globus.org>
- [3] Nordugrid.
<http://www.nordugrid.org>
- [4] R.A. van Engelen *gSOAP: C/C++ Web AServices and Clients*
<http://www.cs.fsu.edu/~engelen/soap.html>

- [5] Jonathan Chin and Peter Coveney, *Towards tractable toolkits for the Grid: a plea for lightweight, usable middleware* (June 2004) www.realitygrid.org/lgpaper21.pdf
- [6] P.V. Coveney, R.S. Saksena, S.J. Zasada, M. McKeown and S. Pickles *The Application Hosting Environment: Lightweight Middleware for Grid-Based Computational Science* Computational Physics Communications (2007) in press
- [7] The Reality Grid project. <http://www.realitygrid.org/AHE/index.shtml>
- [8] Open Middleware Infrastructure Institute. <http://www.omii.ac.uk>
- [9] The GROWL Project. <http://www.growl.org.uk/>
- [10] The CCP1GUI Project. <http://www.cse.scitech.ac.uk/qcg/ccp1gui/index.shtml>
- [11] Python Programming Language. <http://www.python.org>
- [12] The Visualisation Toolkit. <http://www.vtk.org>
- [13] The Dalton Quantum Chemistry Program. <http://www.daltonprogram.org>
- [14] Simplified Wrapper and Interface Generator. <http://www.swig.org>
- [15] L. Schlapbach and A. Züttel, *Nature* 2001, 414, 353.
- [16] A.C. Dillon and M.J. Heben, *Appl. Phys. A, Mater. Sci. Proc.* 2001, 72, 133. Z. Yang, Y. Xia and R. Mokaya, *J. Am. Chem. Soc.* 2007, 129, 1673.
- [17] See for example: J.L.C. Rowsell and O.M. Yaghi, *Angew. Chem., Int. Ed.* 2005, 44, 4670 and references therein. N.L. Rosi, J. Eckert, M. Eddaoudi, D.T. Vodak, J. Kim, M. O'Keeffe and O.M. Yaghi, *Science* 2003, 300, 1127. X.B. Zhao, B. Xiao, A.J. Fletcher, K.M. Thomas, D. Bradshaw and M. J. Rosseinsky, *Science* 2004, 306, 1012. G. Férey, C. Mellot-Draznieks, C. Serre, F. Millange, J. Dutour, S. Surlé and I. Margiolaki, *Science* 2005, 309, 2040.
- [18] N.B. McKeown, B Gahnem, K.J. Msayib, P.M. Budd, C.E. Tattershall, K. Mahmood, S. Tan, D. Book, H.W. Langmi, A. Walton, *Angew. Chem., Int. Ed.* 2006, 45, 1804-1807; (b) J. Y. Lee, C. D. Wood, D. Bradshaw, M.J. Rosseinsky, A.I. Cooper, *Chem. Commun.* 2006, 2670-2672; © J. Germain, J. Hradil, J.M.J. Fréchet, F. Svec, *Chem. Mater.* 2006, 18, 4430-4435.
- [19] S.K. Bhatia, A. L. Myers, *Langmuir* 2006, 22, 1688-1700.
- [20] R.C. Lochan, M. Head-Gordon, *Phys. Chem. Chem. Phys.* 2006, 8, 1357-1370.
- [21] M. Dincă, A. Dailly, Y. Liu, C.M. Brown, D.A. Neumann, J.R. Long, *J. Am. Chem. Soc.* 2006, 128, 16876-16883.
- [22] B. Nyulasi, A. Kovács, *Chem. Phys. Lett.* 2006, 426, 26-29.
- [23] K.O. Christe, H.D.B. Jenkins, *J. Am. Chem. Soc.* 2003, 125, 9457-9461; (b) T. Borrmann, E. Lork, M. Mews, W.D. Stohrer, *J. Fluor. Chem.* 2004, 125, 903-916.
- [24] A Wander, C.L. Bailey, S. Mukhopadhyay, B.G. Searle, N.M. Harrison *J. Mat. Chem.* 16 (2006) 1906-1910
- [25] A. Wander, B.G. Searle, C.L. Bailey and N.M. Harrison *J. Phys. Chem. B* 109 (2005) 22935-22938
- [26] A. Wander, C.L. Bailey, B.G. Searle, S. Mukhopadhyay and N.M. Harrison *Phys. Chem. Chem. Phys.* 7, (2005) 3989-3993
- [27] S. Mukhopadhyay, C.L. Bailey, A. Wanderr, B.G. Searle, S.L.M. Schroeder, R. Lindsay and N.M. Harrison, accepted in *Surface Science* (2007)
- [28] V.R. Saunders, R. Dovesi, C. Roetti, R. Orlando, C.M. Zicovich-Wilson, N. M. Harrison, K. Doll, B. Civalleri, I.J. Bush, Ph. D'Arco and M. Llunell, *CRYSTAL 2003 Users Manual*, University of Torino, Torino, 2004.
- [29] B.G. Searle, *Comp. Phys. Comm.* 137 (2001) 25