Leveraging HTC for UK eScience with Very Large Condor pools: Demand for transforming untapped power into results.

Paul Wilson, Wolfgang Emmerich and John Brodholt
eMinerals Project, University College London, Gower Street, London WC1E 6BT, UK
paul | wolfgang | john @eminerals.org

We provide an insight into the demand from the UK eScience community for very large High Throughput Computing resources and provide an example of such a resource in current production use: the 930-node eMinerals Condor pool at UCL. We demonstrate the significant benefits this resource has provided to UK eScientists via quickly and easily realising results throughout a range of problem areas. We demonstrate the value added by the pool to UCL I.S infrastructure and provide a case for the expansion of very large Condor resources within the UK eScience Grid infrastructure. We provide examples of the technical and administrative difficulties faced when scaling up to institutional Condor pools, and propose the introduction of a UK Condor/HTC working group to co-ordinate the mid to long term UK eScience Condor development, deployment and support requirements, starting with the inaugural UK Condor Week in October 2004.

1. Introduction

UK eScience relies on its grid resources to compute data. HPCx, CSAR and other resources provide high-performance computing, integrated with two 64 dual-node clusters into the new UK eScience Production Grid. Complementing this is the L2G infrastructure and individual project or institutional resources. The value of these resources to eScientists cannot be underestimated. However, there is an increasing demand for resources which provide significantly greater numbers of CPUs, where the emphasis is not CPU power or intercommunication speed, but access to lots and lots of CPUs. This demand has emerged from eScientists who require PC-levels of CPU power, but could not, until recently, harness enough machines to satisfy the desired scope of their research.

2. eMinerals grid computing

The eMinerals project is a grid testbed project funded by the UK Natural Environment Research Council as part of the UK eScience programme. The main aim of the project is to develop a grid approach to tackling environmental issues from the molecular simulation perspective. The primary long-term aim is to be able to increase the scales of length, time and complexity in these simulations by using the possibilities of grid computing coupled with matching developments in simulation methodologies and coding. In so doing we hope that the simulation work will move away from idealised case studies towards being able to simulate cases that match what is found in the real natural environment. The project is bringing together, quite possibly for the first time, environmental and solid state scientists, simulation code developers, and computer scientists. The outcome we expect is a completely new way of carrying out environmental science, with a strong focus on sharing resources within the eMinerals virtual organisation.

A great deal of effort has gone into providing high-throughput computational resources for eMinerals project scientists. To date these have taken the shape of two Condor pools. The first is a Very Large 930+ node pool at UCL, made up of teaching PCs running Windows and now used for scientific production runs. The second pool consists of 20 nodes at the Department of Earth Sciences, Cambridge, running a mixture of architectures, and although production runs are submitted to this pool, it has acted as a useful testbed for development of a number of tools and middleware extensions to facilitate user access prior to their deployment on the UCL Pool. We demonstrate that extending the usefulness of existing, underutilised computing resources within academic institutions can provide powerful resources for the community, and that it can be done cheaply and efficiently.

Central to the eMinerals project is the development of a grid approach to tackling environmental issues from a molecular simulation perspective. Identified within this work is the need for problem-specific computational resources. Those areas requiring significant power and fast process intercommunication were suited to HPC and Beowulf resources. However, there was a gap within available UK resources: no provision for problem areas requiring the processing of hundreds or thousands of short low-power jobs, easily run in a few hours on standard desktop PCs. eMinerals scientists with access to thousands of PC CPU's could address problem areas previously rendered intractable due to the time taken (many months or years) to achieve complete data sets on existing computational resources.

3. A solution using Condor and pre-existing infrastructure

Project scientists need access to machines with a variety of computing power (depending on the size of the calculations) be it memory, CPU cycles or disk space. In several cases, the actual amount of power required for an individual job is relatively low- comparable to an average desktop PC - and parallelisation is not required. Scientists may also require many (thousands or more) runs of the same program, each with a different set of input parameters.

UCL has 930 pentium-3 Windows machines running 24 hours a day, in 31 networked cluster rooms distributed across its campus. All machines are centrally managed, network-booted with one of 8 Windows Terminal Server (WTS) thin-client images, facilitating expedient distributed software deployment and re-engineering. Additionally, the current thin client architecture requires very few local CPU cycles. Crucially, only 5% of the available CPU cycles have historically been used. Harnessing the remaining 95% of CPU cycles across the cluster network has the potential to create an ideal resource for scientists to compute the types of problems described previously.

In order to optimise a grid-based solution to our problem, we analysed the requirements and capabilities extracted from review of the applications, user-base and the local usage policies.

Tables 3.1, 3.2 and 3.3 summarise this information, and define our requirements.

3.1 The capabilities of the applications

- All are capable of using file-based inputs and outputs.
- All are capable of non-interactive 'batch' type execution.
- All are pre-compiled for a windows environment.

3.2 The requirements of the scientist user-base.

- The ability to run the applications in a grid environment without modification.
- The ability to use and access the system on demand.
- The ability to isolate results of each job.
- The ability to submit, monitor and stop jobs.
- The ability to submit jobs to resources lacking the application's software.
- The ability to be notified of any errors.
- The assurance that files, data and job execution is secure.
- The resource is robust and reliable.

3.3 The requirements of the UCL Information Systems Department

- The ability to remove the resource, either partly or wholly.
- Existing computer services and users must retain a higher priority.
- Grid resources must not conflict with existing computer services.
- Grid resources must comply with I.S security and access policies.

Condor was identified as a suitable candidate for meeting these criteria from an early stage, in fact at the start of the project in the summer of 2002. Indeed, there were a number of specific points that commended it:

- It is a free, open-source, heterogeneous resource *and* job management system.
- It is able to harness unused CPU cycles.
- It has a rich and extendable set of configuration options.
- It can run unmodified applications (though at the cost of losing some of the functionality one gains by linking to Condor's own libraries).
- It provides a single point of access.
- It supports a sophisticated suite of configurable security protocols.
- It is trivial to add new nodes to an existing pool.
- It is robust, reliable and designed to complement existing resources.
- It is a core middleware for the UK eScience program.
- It can interface with many toolkits- Globus, Oracle, Unicore to name a few.

This is not to say that Condor in itself provides a complete grid-based solution, but it certainly constitutes a good foundation for projects such as ours, namely distributed scientific computing. Although Condor does not adhere to any specific 'grid' protocol (not surprising since it pre-dates the grid concept), it is already well on it's way towards integration with web services, and potentially WS-RF.

A question that may be raised at this juncture is why Condor is not installed in more academic institutions around the UK? The primary reason is that provision of a cross-institution HTC resource is not trivial, involving multiple stakeholders, centralised and departmental computing resources, adaptation of stable information systems policy and infrastructure and acceptance of expected levels of reliability, robustness and security. We hope that our work will encourage other sites to consider deploying similar systems.

4. Case study: Success of the UCL Condor pool

The UCL Condor pool went into production mode on October 16th 2003 after 15 months of development, with a Linux central manager/submission node harnessing 930 PIII 1GHz Windows 2000 execute nodes created from pre-existing PCs in every cluster room and lecture theatre across the UCL campus. Its usefulness has greatly exceeded expectation:

- The continued provision of a high quality Information Systems-managed service to the priority front-end users of the UCL clusters: the student community.
- An expanding research user group of 16 users, from 6 sites in UK and EU.
- An average of 2 new users per month.
- 5 eScience projects active or testing on the pool
- Over 110 CPU-years (over 1300,000 hours) of calculations completed in 6 months.
- An expanding C, FORTRAN and JAVA code-base usable without any modification.
- Condor-G/DAGman/Globus/SRB remote job submission and data handling ability.

The usefulness of the pool is not restricted to eMinerals researchers. Other scientists with similar problem types are attracted to the possibilities stemming from the availability of 930 machines, a significant improvement on their previously limited ability to compute very large numbers of fast, low-power jobs. Thus far the pool has computed jobs ranging from financial derivatives research and climate modelling, through a range of molecular simulations to neurophysiological image processing. Extant resources are powerful, but do not exist in the numbers needed. The UCL Condor pool has started to fill this gap.

It should be obvious that the integration of a production-standard grid resource into the stable, established computing infrastructure of a large academic institution is not trivial. Initial communications between the eMinerals project and UCL Information systems were met with some not unreasonable caution and skepticism, and were only solved through extensive and detailed negotiations with all stakeholders.

Firstly, significant changes would be required to the existing set of WTS images to incorporate Condor, which would require adding many hours of development man-hours to the existing heavy workload of UCL Information Systems. Secondly, a tough testing regime would be necessary to convince cluster and network managers that Condor would not damage or impair the network. Thirdly, the question of priority was at the forefront of all initial negotiations between eMinerals and UCL EISD- namely that existing cluster development was aimed solely at a specific user group, the institutional student body. The addition of Condor would add a second user group, that of eMinerals and other research scientists. UCL

EISD required full assurance that the addition of Condor would not affect their priority student user-group. Finally, Condor would be deployed on the entire institutional network, embracing multiple domains and cluster owner-ship, multiple WTS image versions and multiple sub nets of the UCL network. Additionally, the Condor system would open up new security issues, and allow much wider access to UCL computing resources to persons outside UCL. Clearly, a user and system service agreement would have to be developed with the view that, if successful, the Condor system would eventually transfer control from eMinerals to UCL Information Systems and become an integral part of UCL computing infrastructure.

The success of the eMinerals Condor program at UCL was made possible through a long process of negotiation, compromise, testing, discovery and, above all, open communication between all stakeholders to bring political and policy issues out into the open. Fortunately the eMinerals project had a head-start. On paper the Condor proposal looked feasible and valuable to UCL, an institution already heavily involved in several leading eScience and grid projects. The eMinerals Condor proposal provided a structured method of developing a significant Condor resource which, for the first time in the UK, would provide true institution-wide Condor cluster deployment. Once proved, we demonstrated the potential for extending the resource to include UCL's 2000 additional WTS non-cluster PC's: creating an eMinerals-developed Condor pool capable of challenging the largest pools in the world.

4.1 UCL Deployment

The UCL pool has been deployed, and is administered, by one of the authors (PW). It now numbers 930 100% windows 2000 nodes, and is fully deployed across the UCL cluster network. Development of the resource evolved over the following timeline:

Months 1-2: Captured requirements, analysed and defined solution, designed system.

Month 3: Learned to use the solution and its technology. Tested on personal machines. Developed incremental testing program to simulate usage, damage and system overload.

Months 4-5: Initiated development process with institutional developers. Deployed for testing onto cluster software. Use limited to the project developer and one identified cluster. Monitored usage patterns of standard cluster users. Monitored network traffic.

Months 6-9: Increased user base to include one external user. Repeated final test program for two concurrent users. Evolved access protocols for external users. Repeated monitoring schedule. Initiated training of potential user base. Published on-line system documentation.

Months 10-13: Deployed onto second cluster on separate subnet and domain. Increased user group size. Repeated monitoring and testing. Maximum 50 nodes.

Months 14-15: Deployed onto four clusters on separate subnets and domains and to project-wide user group. Repeated monitoring and testing. Maximum 100 nodes. Took on system administration role. Developed metrics and system logging protocols.

Month 16: Deployed full production system. Opened system to academic community. Developed UCL Condor Service Level Agreement.

Present Day (Month 24): Preparing to transfer system to UCL Information System Department control as an official institutional grid resource, running through a UCL-managed dedicated server.

4.2 UCL Condor statistics

Diagrams 4.2.1 and 4.2.2 demonstrate the huge usage that the UCL condor pool received in its first 7 months of full production operation. It is noticeable that usage figures have remained fairly constant at around 25000 jobs per month. The figures for March 2004 are incomplete due to missing data; in fact there were over 25000 jobs completed that month as well. The figures for users are also fairly constant. The publicity of the pool's abilities was kept as low-key as possible while the ability of the Condor software to cope with such large numbers of jobs was evaluated- the steady month-by-month usage has been deliberate. One particular statistic is noteworthy- the pool is running at approximately 22% of its maximum capacity. This is A Good Thing; the machines are ours only whilst we maintain our record of NOT affecting their primary use- that of student teaching and coursework. Therefore, we are able to double our usage and still remain under 50% of maximum.

Demand for the pool is higher than the usage figures indicate. Since the figures were published, four new users emerged in May 2004 alone, one of whom has the real ability to double the workload immediately due to their complex workflows involving many hundreds of thousands of jobs. As publicity of the resource emerges, it is predicted that this will continue to increase. Furthermore, we are not including here the demand from our European colleagues. Already, the UCL Condor pool has been used for testing of climate modeling codes from researchers in Germany, who found out about it through word-of-mouth!

Diagram 4.2.1. Job statistics (per month) for the UCL Condor pool:

MONTH	Average job time	Total jobs per month	Total hours per month	Total users per month
Oct 2OO3	6.73	989	6651.20	5
Nov 2OO3	2.23	24391	54342.26	7
Dec 2OO3	15.26	23716	361781.10	5
Jan 2OO4	13.28	28252	375058.30	5
Feb 2OO4	9.73	19082	185563.60	7
Mar 2OO4**	4.93	2121	10453.23	7
Apr2OO4	2.34	24573	57515.68	5
TOTALS	n/a	**123124**	**1051365.37**	**n/a**
AVERAGES	**7.78**	**17589**	**150195.05**	**5.86**

**Add ~20,000 jobs to the March total. Abnormally low figure is due to loss of data.

Diagram 4.2.2. Average statistics for the UCL Condor pool.

av. jobs per month	**17589**
av. jobs per day	**578**
av. jobs per hour	**24**
av. Hours per month	**150195**
av. Hours per day	**4938**
av. Hours per hour	**206**

4.3 Technical issues relating to installing and administrating Very Large Condor pools.

Details below pertain to the UCL Condor pool, but are not unusual for any Very Large Condor resource.

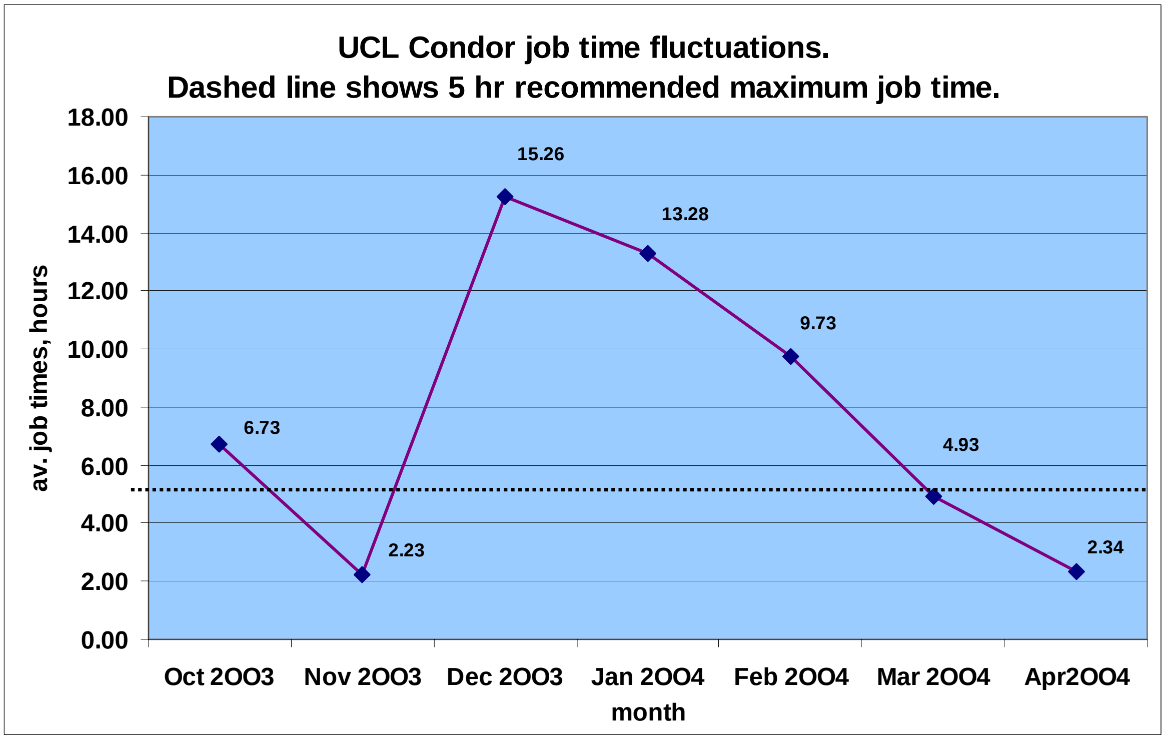
Initially, Condor did not install and work 'out-of-the-box'. This was a manifestation of the UCL Windows Terminal Service (WTS) image-build process, whereby a directory in the FAT file system would be scanned for filenames at a certain step in the process. During this time, the c:/Condor/bin directory was ruined- The file system formatting process would re-name to a junk name any file with a) more than 8 characters and b) who's first 8 characters matched those of any other files in the directory. As this corresponded to many of the condor executables, who's initial 8 characters were 'condor_s…', including the crucial condor_startd.exe and condor_starter.exe it was a critical fault.

Condor's built-in flexibility came to the rescue: by creating a separate and distinctly named c:/Condor/bin subdirectory for each mandatory executable for the remote execute machines (in this case, condor_startd, condor_starter, condor_master and condor_on/off executables), dictating these changes in the condor_config file, manually editing the Windows registry to effect the changes and creating a Windows service to automatically boot Condor on machine startup enabled us to remotely deploy Condor across the eight different WTS images required to populate the entire cluster network with up-to-date operating systems.

Other than this, there remains only one thorn in the side of UCL Condor: that of machine re-booting. Condor for Windows does not support checkpointing; once a machine 'dies', the job it is running dies also, re-joins the Condor queue and is re-submitted from scratch: all the data is lost. The UCL cluster network is unusual in that ~80% of all machines are re-booted every evening, ostensibly to empty the cluster rooms of students! Additionally, many machines are rebooted (due to lock-ups or student un-awareness) many times during an average term-time 24 hour period. As this was monitored, it became clear that a job with a run-time of less than 5 hours was most likely to complete first time, and anything over this time would inevitably involved at least one Condor re-start- The longer the job, the more restarts it would undergo before finally settling on a machine which remained undisturbed or the duration of the job.

Diagram 4.3.1 charts our efforts to educate users to ensure their jobs were as short as possible. We have now been achieving an average of less than 5 hours for over 3 months since the emergence of unacceptably long job submission times during November and December, and are investigating ways of lowering the frequency of machine shut-downs. It should be noted that this is a manifestation of the Condor environment at UCL, not a fault with Condor itself. Once Condor provides checkpointing for Windows this should become a thing of the past- but checkpointing itself introduces issues of significantly heavier network bandwidth and disk usage.

Diagram 4.3.1. Average monthly job run-time distribution.



Furthermore, with a single scheduler/master node, we have experienced very heavy loads on the scheduler. On two occasions this has resulted in Condor 'melt-down'- usually with over 3 or 4000 jobs in the queue- where Condor has to be re-installed from scratch. The Condor team are aware of the scheduler's weakness in controlling very large numbers of jobs, and debate continues over the relative merits of alternate job submission paradigms.

5. The demand for very large Condor Pools

The UCL Condor pool is only one resource, and, while there are many Condor pools on-line in the UK, none are close to the size of the UCL resource, nor are they all open to the whole eScience community. As awareness grows, the user group is steadily increasing (on average 2 new users per month at UCL alone) and the possibility for expanding the scale of simulations expands. For example: a recent request involved submitting 100 (initially) workflows involving 29 initial calculations with FORTRAN executable A, followed by 200 calculations for each of the initial 29 results with FORTRAN executables B1 – 4, the whole set taking 3 hours-300 hours in total, but 580,000 individual CPU-jobs! The UCL pool spends increasingly longer periods of time at full capacity with queues reaching into the thousands. This leads to the conclusion that the creation of similar resources is necessary to help spread the load and pre-empt the increases in demand and complexity. That this is achievable is proved by our UCL experience. Time taken to compute data is a necessary evil, particularly for those with large volumes of jobs. The ability to minimise this time is important- users of the UCL Condor pool have seen huge cuts in time taken to achieve results, in several cases down from years to months. Whilst individual jobs each take longer, the ability to submit them in their hundreds significantly decreases time taken to complete whole data sets.

6. Institutional benefits of Very Large Condor Pools.

The ability to add value to multi-million investment in I.S infrastructure is important. UCL has gained significant exposure through collaborating with eMinerals in provision of the first and largest institution-wide UK Condor pool, and provides a model example of collaboration between projects and I.S admin in provision of a world-class grid resource with production-level QOS. Cost has been limited to time and one FTE for two years, and the existing infrastructure now provides for high-quality science through harnessing, on average, ~22 % of it's previously wasted CPU power. Concerns regarding negative impact on network and I.S service quality have proved totally unfounded, and all aspects of the new service rigorously tested before public release. Even at 100% Condor utilisation, the additional load on the UCL infrastructure does not degrade the service to its primary users. Whilst the initial 15 month integration, development and testing stages incurred significant frustration and time costs for both I.S and eMinerals, the benefits are real: maintenance and administration are minimal, aided by Condor's ability to be installed and configured 'out of the box' with a minimal set of now trivial UCL-specific changes in a matter of minutes. In terms of basic infrastructure, a low estimate of £750,000 would be required to replicate the volume of work done in the same time period by purchasing a dedicated 750-node Linux cluster, not to mention time and energy costs of setting the whole thing up from scratch. One further important aspect of such large scale pools is that Both UCL and eMinerals gain from the extensive new collaboration between research communities active on the UCL Condor pool.

7. Towards a UK eScience Condor HTC infrastructure

Cambridge and Cardiff Universities are well under way with their own institution-wide Condor pool developments, with very different infrastructure and policies to UCL. Common to both is the availability of wasted power, the flexibility of Condor to adapt to heterogeneity of environment, the willingness of the Condor team at the University of Wisconsin-Madison to provide excellent support, and a user-base hungry for sufficient quantities of the right type of power to deliver un-bounded top-quality science. The Universities of Aston, Westminster, Southampton and Imperial also support well-developed Condor pools.

Beyond these programmes, the UK institutional infrastructure supports many tens of thousands of PC's which will continue to waste a significant quantity of their power unless it is harnessed into providing the missing level of UK eScience grid infrastructure. That this is not a trivial procedure should be made clear, and the need for such resources to be federated must be emphasised. The UK's position as the second largest Condor community in the world provides the motivation, expertise and Condor-related programs to develop a Federated Condor Service for the UK, providing accessible High-Throughput Computing for the academic community via a Grid Operation Centre Condor support network and a UCL/eMinerals-led UK Condor Working Group. The inaugural UK Condor Week, to be held at NeSC in October, and led by UCL, will be the starting point for these efforts along with the new UK Condor download mirror hosted at UCL. Concurrently, the continued lead of UK eScience research within Condor grid standardisation, Condor toolkit development and institutional-level Condor integration will ensure that there is the necessary expansion of technologies required for the continued success of very large Condor pools within the dynamic, heterogeneous grids of the future.

Acknowledgements

The eMinerals project is funded by NERC. The authors would like to thank Mr Andrew Dawson, Neil Christie and James Mitchell of the UCL Information Systems Department, and the Condor teams from the Universities of Aston, Cambridge, Cardiff, Southampton, Imperial and Westminster.

References

[1] D. Thain, T. Tannenbaum , M. Livny. *Condor and the Grid;* in F. Berman, A.J.G. Hey, G. Fox, editors. *GridComputing: Making the Global Infrastructure a Reality*, John Wiley, (2003)

[2] URL: http://www-unix.globus.org/toolkit/documentation.html#2.4

[3] I.Foster, C. Kesselman, J.Nick, S. Tueke, *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems integration.* Open Grid Services Infrastructure WG, Global Grid Forum. (2002)

[4] J. Frey, S. Graham, T. Maguire, D. Snelling, S. Tueke. *WS-Resource Framework and WS-Notification Technical Overview.* GlobusWORLD (2004)

[5] URL: http://www.andykhan.com/jexcelapi

[6] J.D. Gale, *Phil. Mag. B*, 73, (1996)

[7] J.D. Gale, *JCS Faraday Trans.* 93, (1997)

[8] W. Smith , T.R. Forester, *Journal of Molecular Graphics* 14 (3), (1996)
URL: http://www.cse.clrc.ac.uk/msi/software/DL_POLY

[9] 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, *CRYSTAL03, User's Manual*, University of Turin, Turin, Italy (2003);
URL http://www.crystal.unito.it/

[10 J. M. Soler, E. Artacho, J. D. Gale, A. García, J. Junquera, P. Ordejón and D.l Sánchez-Portal. *J. Phys.: Condens. Matter* 14 (2002)
URL http://www.uam.es/siesta

[11] X. Gonze, J.-M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G.-M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, Ph. Ghosez, J.-Y. Raty and D.C. Allan. *Computational Materials Science* 25 (2002)
The ABINIT code is a common project of the Université Catholique de Louvain, Corning Incorporated, and other contributors.
URL http://www.abinit.org

[12] http://www.cs.wisc.edu/condor/manual/v6.6/2_11DAGMan_Applications.html

[13] M. Litzkow, T. Tannenbaum, J. Basney and M. Livny, *University of Wisconsin-Madison Computer Sciences Technical Report #1346* (1997)

[14] G. W. Watson, E. T. Kelsey, N. H. de Leeuw, D. J. Harris, S. C. J. Parker. *Chem. Soc., Faraday Trans.* 92, 433. (1996)

[15] B. Allcock, J. Bester, J. Besnahan, A.L. Chervenak, I. Foster, C. Kesselman, S. meder, V. Nefedova, D. Quesnal and S. Tueke. *Parallel Computing Journal,* 28(5), (2002)

[16] URL: http:// www.ncsa.uiuc.edu/ssh

[17] J. Frey, T. Tannenbaum, I. Foster, M. Livny, S. Tueke, *Cluster Computing*, 5(3), (2002)

[18] C Chapman, P Wilson, W Emmerich, (University College, London) T. Tanenbaum, MFarrellee and M.Livny (University of Wisconsin). *NeSC Special Report.* (2004)

[19] P.Wilson et al. *Proc. 10th Global Grid Forum.* (2004)

[20] M Dove, P Wilson et al. *Proc. UK eScience All Hands 2004.* (2004)