

# Uniform Access to the Distributed Resources for the Computational Chemistry Using UNICORE

Jarosław Pytliński, Lukasz Skorwider  
ICM Warsaw University  
Pawińskiego 5a, 02-106 Warsaw, Poland  
{pyciu, luksoft}@icm.edu.pl

Krzysztof Benedyczak, Michał Wroński, Piotr Bała  
Faculty of Mathematics and Computer Science  
Nicolaus Copernicus University  
Chopina 12/18, 87-100 Toruń, Poland,  
{golbi, wrona, bala}@mat.uni.torun.pl

Valentina Huber  
Research Center Jülich  
D-52425 Jülich, Germany  
v.huber@fz-juelich.de

**Abstract.** In this paper we describe deployment of quantum chemical and biomolecular applications on the grid. We have used UNICORE infrastructure as framework for development dedicated user interface to the number of existing computational chemistry codes as well as molecular biology databases. The user interface is integrated with the UNICORE client based on plugin mechanism which provides general grid functionality such as single login, job submission and control mechanism.

## 1 Introduction

Research in the area of computational chemistry requires access to the computer resources usually not available at the user workstation. This includes various types of resources such as computational servers, visualization workstations as well as number of databases. Current desktop computers provide significant CPU speed, disk space as well as visualization capabilities and therefore become important computational platforms. More often they are used for graphical representation and analysis of results both theoretical and experimental. However, nontrivial problems still require large memory, disk space and CPU time, available only at remote systems or sites. Also large and frequently updated databases are available remotely and cannot be hosted locally, also because of the license restrictions.

Distributed computational resources cannot be effectively utilized using typical unix tools based on the remote login. Users sharing external as well as departmental or even local computers have to solve number of practical problems.

They are faced with different site policies and practices such as different security, different user identification, different priorities or scheduling mechanisms and so on. Additionally, users adopt latest developments in the graphical user interfaces and simply do not want to use traditional, command line based tools. Number of applications exists but none of them can cover full range of functionality. In result user has to work with different programs dedicated to computations, visualization or database access. Each of them has specific user interface and requires dedicated and often non compatible file formats.

Currently various computational techniques become standard tools for a large number of users. One should note that increasing number of them have no long-standing experience in large scale computing and data mining as well as advanced visualization and expects intuitive tools which are easy to learn and use. Such tools cannot be just another application, but rather should allow for integration of existing tools and programs within one uniform framework. For example, user has to be able to use visualization software he is used to and should be able to replace it with new version or even another application.

Recent advances in computer technology, especially grid tools make them good candidate for development of uniform user interface to distributed resources [1]. Computational grids enable sharing a wide variety of geographically distributed resources and allow selection and aggregation of distributed resources across multiple organizations for solving large scale computational and data intensive problems in chemistry. This includes also access to various databases as well as unique experimental facilities. In order to attract users, grid interfaces must be easy to install, simple to use and able to work with different operating system and hardware platforms.

Most of the computational chemistry programs which provide graphical user interface run locally and access to the remote resources is limited. User can however use them for input preparation for applications running at the remote systems but this requires additional effort to transfer input and output files. On another hand, computational intensive applications in general have no dedicated user interfaces. Some expensive commercial graphical tools are available as well as academic software but even sophisticated tools such as GaussView [5], Tripos [6] or Cerius [7] are able to run user applications only on local workstations and do not take advantage of grid environment. These and many other applications can be used as aid for input preparation, but in most cases job must be prepared and submitted manually by the user.

The existing web technology can provide powerful user interface to computational resources and databases. It cannot however achieve high numerical efficiency required for most simulation problems. This leads us to solutions consisting on web interface for the user and traditional, usually batch type, numerical simulation engine.

This approach is presented by WebMo [8] which is web based submission system for quantum chemistry codes such as Gaussian [9], Gamess [10] and Mopac [11]. This tool is limited to the local batch systems and has no grid capabilities. The web submission to the geographically distributed systems is

possible within BioCore [12] which is web interface to the molecular dynamics code NAMD [13]. Currently this system is limited to the particular MD code and single visualization package (VMD). NPACI Gamess Portal presents analogous approach for quantum mechanical code.

Presented problems can be solved with the help of grid middleware which provides access to remote high performance resources. However most of the development goes to the design and production of general tools which can be used in different situations and provide simple access to remote resources. The existing solutions like globus [2], Legion [3], LSF [4] address most important issues. Unfortunately, these tools require an advanced computer skills for installation and usage. Moreover, in order to provide required functionality user application must be modified, sometimes significantly. This is not possible for commercial or legacy codes which cannot be easily adopted by the user. Most grid solutions have limited user interface not addressing application specific issues. The significant exception is UNICORE [14] middleware which provides uniform interface to the distributed computer resources of different type.

## 2 UNICORE

The details of the UNICORE can be found elsewhere [19] and we will summarize here its most important features.

UNICORE is uniform interface to the computer resources which allows users to prepare, submit and control application specific jobs and file transfers. Jobs to be run on the same system can be grouped in job-groups. The user specifies target system for job group as well as resource requirements for CPU time, number of CPUs amount of memory and disk space for job. Jobs can have complicated structure with various job subgroups which can be run at different systems. Subgroup jobs and file transfers can be ordered with user defined dependencies. This includes conditional execution and loops with control which can be used as environment variables in the user's jobs. The user input is mapped to the target system specific commands and options by the UNICORE infrastructure. Compared to other tools UNICORE has wider functionality, is more flexible and allows for much easier integration of the user interface with external applications.

The UNICORE architecture is based, like other grid middleware, on the three tier model. It consists of user, server and target system tier. The user tier consists of the graphical user interface, the UNICORE client, written as Java application. It offers the functions to prepare and control jobs and to set up and maintain the user's security environment. The UNICORE job can be build from multiple parts which can be executed asynchronously or dependently on different systems at different UNICORE sites. Within the UNICORE environment user has a comfortable way to use distributed computing resources without having to learn about site or system specifics.

In order to overcome this disadvantage the software is introduced to the UNICORE through the Incarnation DataBase (IDB) entries which describe details of the local installation. IDB allows for definition of the environment variables

for the script executed by the user. In this way we can define variables required for particular application or define path to the program.

The UNICORE infrastructure allows also for registration of the applications available on the target machine through `SOFTWARE_RESOURCE` entries in the IDB. The program name and program version together with shell commands required to set up execution environment are stored in the dedicated section of the IDB file. The UNICORE client has build-in capabilities to check software resource entries on the particular target systems and use it for job construction.

The UNICORE client provides also the plugin mechanism which become very attractive and efficient method for integration of applications with grid middleware. Plugin is written in Java and is loaded to the UNICORE client during start, or on request. Once it is available in the client, in addition to the standard futures such as preparation of the script job, user gets access to the menu which allows preparation application specific jobs. Dedicated plugins can be used for database access or postprocessing. Since plugins are written in Java, they can be easily integrated with external applications or existing Java applications or applets.

The UNICORE security is based on the Secure Socket Layer (SSL) protocol and the X.509 certificates. SSL uses public key cryptography to establish connections between client and server. Therefore each component of the UNICORE infrastructure has a public-private key pair with the public part known by the others. The keys have to be certified by a certification authority. The user's X.509 certificate is his identification and is maintained by the UNICORE client application in a encrypted data base. The client has to know about the CA which signs the user and gateway certificates. The SSL is used for the connection coming over insecure internet to the UNICORE gateway. The user is authenticated by the gateway when presenting certificate. The user certificate is translated to the user login under which all tasks are executed on the target system. The user can use number of certificates to access different remote resources. The authentication scheme supports also project concept and jobs can be executed using different projects with single login.

The UNICORE client provides users with tools for certificate management such as certificate signature request generation, keystroke editor or public key export.

### 3 Computational Chemistry Applications

The UNICORE software provides general framework for running user applications. This includes input preparation, job submission and control and finally postprocessing of the results. Advanced visualization and access to the databases and other distributed sources of information should also be considered here. The wide functionality and especially flexibility obtained with plugin concept makes UNICORE good candidate for general framework for access to the various, not only computational, resources. In the next section we will describe extensions which covers wide range of computational chemistry scenarios.

### 3.1 Application Specific Interface

In most cases input for the CPU intensive chemistry application has text form and is prepared with standard text editor. This approach requires significant experience from the user and any mistake in the file format results an error and extends time in which results will be obtained. The preparation of the input file requires knowledge of many keywords to describe atom coordinates, molecular symmetry, geometry optimization, electronic properties, etc.

This disadvantages can be removed by the application specific interface which assist users and generates parameters automatically to the text area, where it can be modified by experienced users by hand, e.g. they can add new keywords. It checks the input data and their interdependencies and prevents users from entering incorrect values while requesting to set mandatory parameters. Once generated input can be stored format to be reused and modified in the future.

Using UNICORE plugin concept such interfaces has been developed by us for the most popular CPU intensive quantum chemistry application such as Gaussian98, Car-Parrinello Molecular Dynamics calculations [20] and molecular dynamics code Amber [16].

In addition to the standard input preparation features plugins can provide user with additional functionality. For example Amber plugin contains additional help on keywords and parameters together with information on default values, Gaussian plugin estimates required CPU time.

More detailed description of the plugin functionality can be found in [17].

One should note that plugin development requires knowledge of Java programming as well as UNICORE Client interface, but once technology is learned the development is fast.

### 3.2 Job Submission

After successful generation of the input files user has to construct job and submit it to the target system. Traditionally, user develops shell scripts and adopts them to the target system. UNICORE can use well established model and job can be constructed using script tasks. We have developed templates for most popular biomolecular applications including Gaussian98, Gromos96 [15], Amber and Charmm [18]. The templates can be easily modified by the user in the part including application input, exactly as it is done for batch jobs. Significant difference comes from the fact using UNICORE middleware scripts can be run on any target system without modifications taking advantage of the environment variables defined in the IDB.

The job configuration includes information on the input files which has to be transferred from user workstation to the system where particular application will be executed as well as information on the output files which must be transferred back. Once this information is provided by the user all required file transfer will be performed automatically during job submission or retrieval of the results.

As described above, the UNICORE allows user for advanced workflow management. This includes creation subtasks which can be run on the different target

systems. For example the structure minimalization with Amber was run on one target system than results were transferred to another system where molecular dynamics simulations were performed.

While application specific input is prepared using dedicated plugin the proper job task is generated automatically. In this case user avoids script programming and job can be created even by non-experienced user. In addition, the plugin checks the availability of the software on sites and allows job submission only to sites, where the particular program is installed. Furthermore the site administrators can specify in the Incarnation DataBase several software versions with appropriated environments, e.g. different paths to binaries, and the plugin displays available versions in a choice box to be selected by a user.

As part of the job preparation, user specifies resources required by the job. This includes CPU time, required memory size and disk space. UNICORE client provides user with dedicated resource editor which displays information on the target system resources and allows for their adjustment to the particular job. Currently resources must be specified by the user, however the Gaussian plugin has build in estimation of the CPU time required for the job execution. This is based on the known algorithm scaling as well as on stored in the IDB target system performance and is calculated according to the particular input. Such estimate is of course not accurate and rather provides user with the upper limit of the CPU time.

Recent developments of the UNICORE middleware allows for easy discovery which target system has particular software installed. Dedicated plugin lists target systems either from the particular site, or from all sites user has access to.

### 3.3 Job Control

Once job is submitted user can check job status, monitor execution and retrieve output to the local workstation. All these functions can be performed from the UNICORE client with a single login during client startup. User can monitor job status and retrieve output to any computer connected to the network with the UNICORE client installed, in particular other than one used for job submission.

Using graphical interface provided by the UNICORE Client user can obtain information on the jobs status and is allowed to hold, resume or cancel executing job. With the dedicated plugin status of the job can be checked periodically. Since molecular dynamics applications require some specific job control mechanism we have developed plugin which display status of all users job known to the UNICORE Client. All available sites are checked and known jobs are listed and displayed in the table which can be differently ordered. User has also possibility to select particular job and get more details on it using standard UNICORE Client functionality.

Standard UNICORE monitoring mechanism provides only general information on the job or subjob status and lacks details such as estimation of the job progress or information on the intermediate results. This feature is required especially for long running molecular dynamics simulations.

We have removed this disadvantage by designing plugin which allows to select running job from the UNICORE Client and monitor its execution on the target system. The Plugin prepares service job which based on the provided UNICORE job number enters this job working directory on the target system and gets access to the generated files. Selected files can be retrieved to the users workstation as they exists in the working directory or can be filtered to transfer only most significant information. Plugin provides user with typical filters such as retrieval of the last lines of selected file, but more advanced filters can be written. The postprocessed file is transferred to the UNICORE client and can be displayed in the text window or used for visualization.

Designed plugins extends UNICORE client functionality and adopts it to the computational chemistry requirements for job control and monitoring. Especially these tools significantly help user to to handle number of long running jobs.

### 3.4 Analysis and Visualization

The important part of most computational chemistry applications is visualization and analysis of the results. In the past this task has been on the dedicated workstations, but recent development in computer hardware and architecture allows to use common workstations. Chemistry users adopt these changes and number of visualization packages has been developed and is used in both Windows an Unix environment. Since UNICORE provides user with easy to use, uniform interface to the resources we decided to use it also for visualization. This was performed by introducing graphics capabilities to the selected plugins. For example CPMD plugin which takes care on input preparation and retrieval of results is able to plot graphs using data transfered files during the calculation. The trajectory of the molecular structure can be displayed from the plugin used external java package JMol. The graphics can be updated on a user's request or a user can specify the timestep for the automatic update. This feature allow to monitor the calculation of long running jobs or to store the computing resources by aborting jobs with wrong behaviour.

Similar functionality is available in Gaussian and Amber plugins.

We have decided to use existing visualization packages because users are used to them and there is no need to develop just another visualization tool. As result we were able to integrate UNICORE plugins with such packages as JMol, JMV or RasMol/RasWin as well as any other application available on the Client workstation.

### 3.5 Database Access

The Librarian plugin allows for access to the remote databases from the UNICORE client. It allows user to prepare query to the PDB database [22], widely used crystallographic database and nucleotides database Entrez [23]. While plugin is started user can build database query using dedicated interface: simple, based on the keyword search, or advanced one with full functionality provided by the database. The query is submitted to the databases web interface and result

is returned to the UNICORE client. The search results can be browsed, visualized or saved for later processing. Since copies of both databases are available at different locations, the plugin checks automatically which mirror is currently available. For the user query the one which provides answer in the shortest time is used.

## 4 Conclusions

The UNICORE software was used to establish European computational grid - EUROGRID [24]. BioGRID is application oriented grid which adopt EUROGRID infrastructure to the specific area, namely molecular biology and quantum chemistry.

The UNICORE software was used as framework providing uniform access to the number of resources and applications important for computational chemistry and molecular biology. This includes seamless access to the computational resources, databases as well as framework for application specific interfaces development. Examples presented here demonstrates capabilities of the UNICORE middleware and shows directions for further development. Modular architecture based on the plugin concept and demonstrated visualization capabilities open number of possible applications in the different areas.

One should note, that UNICORE middleware can be used also together with other grid middleware, especially can be used as job preparation and submission tool for globus.

**Acknowledgements** This work is supported by European Commission under IST grant 20247. The software was used using EUROGRID facilities at ICM Warsaw University (Poland), Forschungszentrum Jülich (Germany), University of Manchester - CSAR (UK), IDRIS (France) and University of Bergen - Parallab (Norway).

## References

1. C. Kesselman I. Foster, editor. *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufman Publishers, USA, 1999.
2. I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *Int. J. Scientific Applications*, 11(2):115–128, 1997.
3. Legion. University of Virginia. Charlottesville. VA USA [Http://legion.virginia.edu](http://legion.virginia.edu).
4. LSF. Platform Computing. <http://www.platform.com>.
5. GaussView. Gaussian Inc., Pittsburgh PA. USA. 2000.
6. Tripos. Tripos Inc. St. Louis. USA.
7. Cerius<sup>2</sup>. Accelrys, San Diego. USA.
8. Webmo. <http://www.webmo.net>.
9. M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, J. A. Montgomery, Jr., R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam, A. D. Daniels, K. N. Kudin, M. C. Strain, O. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G. A. Petersson, P. Y. Ayala, Q. Cui,

- K. Morokuma, P. Salvador, J. J. Dannenberg, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. Cioslowski, J. V. Ortiz, A. G. Baboul, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, J. L. Andres, C. Gonzalez, M. Head-Gordon, E. S. Replogle, and J. A. Pople. Gaussian 98. 2001.
10. M. W. Schmidt, K. K. Baldridge, J. A. Boatz, M. S. Gordon, S. T. Elbert, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. J. Su, T. L. Windus, M. Dupuis, and J. A. Montgomery. General atomic and molecular electronic structure system. *J. Comput. Chem.*, 14:1347–1363, 1993.
  11. J. J. P. Stewart, L. P. Davis, and L. W. Burggraf. Semi-empirical calculations of molecular trajectories: method and applications to some simple molecular systems. *J. Comp. Chem.*, 8(8):117–23, 1987.
  12. M. Bhandarkar, G. Budescu, W. F. Humphrey, J. A. Izaguirre, S. Izrailev, L. V. Kal, D. Kosztin, F. Molnar, J. C. Phillips, and K. Schulten. Biocore: A laboratory for structural biology. In A. G. Bruzzone, A. Uchrmacher, and E. H. Page, editors, *Proceedings of the SCS International Conference on Web-Based Modeling and Simulation*. San Francisco, California, 1999.
  13. L. Kal, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten. NAMD2: Greater scalability for parallel molecular dynamics. *J. Comp. Phys.*, 151:283–312, 1999.
  14. UNICORE. Unicore Forum. <http://www.unicore.org>.
  15. W. Van Gunsteren and H. J. C. Berendsen. *GROMOS (Groningen Molecular Simulation Computer Program Package)*. Bimos, Laboratory of Physical Chemistry, ETH Zentrum, Zurich, 1996.
  16. P. Kollman. *AMBER (Assisted Model Building with Energy Refinement)*. University of California, San Francisco, USA, 2001.
  17. J. Pytliński, L. Skorwider, V. Huber, K. Bednarczyk and P. Bala. iUnicore- an uniform platform for chemistry on the grid, 2002
  18. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.*, 4:187–217, 1983.
  19. P. Bala, B. Lesyng and D. Erwin. Eurogrid-european computational grid testbed in press, 2002
  20. CPMD consortium. <http://www.cpmc.org>.
  21. V. Huber. Supporting Car-Parrinello Molecular Dynamics with Unicore. In *Computational Science ICCS - San Francisco, USA, Proceedings, Pt. 1*, volume 2073, pages 560–567. Springer Verlag, 2001.
  22. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 2000.
  23. <http://www.ncbi.nlm.nih.gov>.
  24. <http://www.eurogrid.org>.