BioGRID - An European grid for molecular biology

Piotr Bała, Jarosław Pytliński, Łukasz Skorwider,
Faculty of Mathematics and Computer Science, N. Copernicus University
Chopina 12/18, 87-100 Toruń, Poland,
Mirosław Nazaruk
ICM Warsaw University, Pawińskiego 5a, 02-106 Warsaw, Poland
Victor Alessandrini, Denis Girou, Gilles Grasseau
IDRIS, Batiment 506, F-91403 Orsay cedex, France
Dietmar Erwin, Daniel Mallmann
Research Center Juelich, D-52425 Juelich, Germany
Jon MacLaren, John Brooke
Manchester Computing, University of Manchester Manchester M13 9PL, UK
Jan-Frode Myklebust
Parallab, University of Bergen, N-5020 Bergen, Norway.

Recent advances in computer technology, especially grid tools makes them good candidate for development of user interfaces to computing programs and 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 science.

User point of view is most important principle in the development of the Unicore [2] software which was used to establish European computational grid - EUROGRID [3]. BioGRID is application oriented grid which adpots EUROGRID infrastructure to the specific area, namely molecular biology and quantum chemistry [4]. Research in these areas requires computer resources usually not available at the user workstation. Large problems need resources on variety of systems at different locations. Users are faced with different site polices and practices such as different security, different user identification and so on. Distributed computational resources cannot be effectively utilized using typical unix tools based on the remote login. In such situation easy and uniform access to the remote computer systems becomes crucial for users.

Most of the quantum chemical and molecular biology applications has no dedicated user interface, however some graphical interface exist. Some expansive commercial graphical tool are available as well as academic software but even sofisticated tools such as GaussView, Tripos or Cerius are able to run users applications only on local workstations and does not take advantage of grid environment. This and number of similar applications can be used as aid for input preparation, but in most cases job must be prepared and submitted manually by the user.

The another approach is presented by WebMo [5] which is web based submition systems for quantum chemistry codes such as Gaussian [6], Gamess [7] and Mopac [8]. This tool is limited to the local batch systems and has no grid capa-

bilities. The web submission to the geographically distributed systems is possible within BioCore [9] which is web interface to the molecular dynamics code NAMD [10]. 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.

Unicore is uniform interface to the computer resources which allows user 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 CPU's amount of memory and disk space for job. Job can 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 dependency. The user input is mapped to the target system specific commands and options by the Unicore infrastructure. Compare to the tools described above BioGRID, using Unicore as main grid tool, has wider functionality, is more flexible and allows for much easier integration of the User interface with external quantum chemistry or biomolecular applications.

The Unicore architecture is based, as 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 - 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.

Within the Unicore environment user has a comfortable way to use distributed computing resources without having to learn for site or system specifics.

The graphical user interface offers functions to maintain security, to prepare Unicore jobs, submit and monitor them. The Unicore job can be build from multiple parts which cane be executed asynchronously or dependently on different systems at different Unicore sites. Basic element are:

- script task, to submit job scripts;
- transfer task to specify data transfer between different jobs groups;
- job groups, to build subjobs for other systems.

All elements can be add and edited with the graphical interface in the Unicore client. User can define target system for each job group as well as dependences between job groups and file transfers. Each job group must have defined resources it requires such as number of processors or nodes, memory, CPU time and disk space. Target system is checked if it can fulfill this requirements. This tasks is performed during process of preparing job, before it is submitted.

The Unicore architecture is based, as 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 - 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 client generates from the user's input an Abstract Job Object (AJO) which is sent to the other components of the Unicore infrastructure. The AJO is a key component which compromises the Unicore protocol between

user interface and Network Job Supervisor (NJS) together with the abstract job specification generated from user input. The Gateway is the first part of the Unicore Server tier. It takes care of user authentication, secure communication between client and server. Gateway provides client with information on resources available at site. It also talks to the Network Job Supervisor at the site to send jobs and data, status requests and control commands for further processing and to receive data to make it available to the user.

NJS controls one Virtual Site (Vsite) which is a single system or cluster which shares the same userids and file space. It maps the Unicore user id to the local user id at the target system, extracts and translates the jobs contained in AJO into real batch jobs sends job-groups to be executed at other sites to the corresponding gateway, provides local resource information to the gateway and takes care of the file transfer. The transfers are done by Unicore transparently to the user.

The NJS translates the abstract job definition into batch jobs for the destinate systems and sends them to the target system interface (TSI) on the system.

The security is based on the Secure Socket Layer (SSL) protocol and the X509 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 (CA). By default certificates signed by unknown authority are not accepted. The user's X509 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 user is authenticated by the gateway when presenting certificate which is also part of Abstract Job Object.

The SSL is used for the connection coming over insecure internet to the Unicore gateway which communicates with the NJS by sockets.

Within Unicore infrastructure the user program can be run as script submitted to the target system in the way analogous to the batch execution mode. The advantage is single login and transparent file transfer but submitted job is similar to the one used with the traditional queuing system.

Compare to other grid tools and applications this solution is easy to install, both at the server and client side and provides user with simple and intuitive interface. Unicore client provides general mechanisms for user authentication and job preparation, submission and control. It is also used as framework for development application specific interfaces.

The biomolecular 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. This entries must be customized for each target system and can be used by any user.

The Unicore execution model allows to include shell command in the IDB entries, as it is done in the second line of example. This allows to prepare any

environment for execution shell. Knowledge of defined variables is required for preparing Unicore job, but allows to run the same script without modifications on any target machine.

The Unicore infrastructure allows also for more sofisticated registration of the applications available on the target machine through 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 built in capabilities to check software resources entries on the particular target systems and user gets informed if the chosen program is not available. This mechanism however cannot be used with the simple user script jobs and is accessible only when user job is prepared with the dedicated plugin to the Unicore client.

Taking advantage of the environment variables defined in the IDB at target systems we have developed Unicore Abstract Job Objects (AJO) for most popular biomolecular applications including Gaussian98, Gromos96 [11], Amber [12] and Charmm [13]. The AJOs can be easily modified by the user in the part including application input, exactly as it is done for the batch jobs. However prepared AJOs can be run without any modifications using Unicore middleware on any target system.

Unicore script job still requires knowledge of the input files for the biomolecular applications. In most cases it is performed with standard text editor and requires significant experience from the user. Any mistake in the file format results with an error and extends time in which results will be obtained.

We have used Unicore as framework for development dedicated user interface to the biomolecular applications. 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 for preparation application specific job.

Once input is ready user can specify target system and resources required for jobs using standard Unicore client facilities. For example 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 the 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 submition.

Currently plugin's are availiable for most popular quantum mechniacal and molecular dynamics codes: Gaussian98 and Amber 6.0.

User can set up type of *ab initio* calculations and add options specific to the particular type of simulations. Plugin allows only for options which do not conflict with the chosen simulation type. The theory level, basiss set, total charge and multiplicity can be set. All options are presented with the defaults which correspond to the simple calculations.

Amber plugin allows for preparing input for MD simulations in various modes: constant energy, constant temperature or constant pressure. In each

mode user is able to specify most important parameters of the simulations such as time step, total simulation time, initial time and others

The gathered experience will be used for development of plugins for other bimolecular applications. Unicore client provides basic grid functionality which makes development of the new application specific interfaces easy. This will result in increasing number of applications accessible in the grid environment.

Acknowledgements This work is supported by European Commission under IST grant 20247. The software was used using EUROGRID facilities at ICM Warsaw University (Poland), Forschungcentrum Juelich (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. Unicore. Pallas. Germany http://www.unicore.org
3. EURGRID. http://www.eurogrid.org
4. BioGRID. http://biogrid.icm.edu.pl
5. Webmo. http://www.webmo.net.
6. 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. ab initio molecular orbital package.
7. 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. ab initio molecular orbital package.
8. 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.
9. Biocore. http://www.biocore.net.
10. 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.
11. W. Van Gunsteren and H. J. C. Berendsen. *GROMOS (Groningen Molecular Simulation Computer Program Package)*. Biomos, Laboratory of Physical Chemistry, ETH Zentrum, Zurich, 1996.
12. P. Kollman. *AMBER (Assisted Model Building with Energy Refinement)*. University of California, San Francisco, USA, 2001.

13. 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.