Computing Molecular Energy Surfaces on a Grid

Loriano Storchi[1], Francesco Tarantelli[1,2], and Antonio Laganà[1]

[1]Department of Chemistry, University of Perugia,
via Elce di Sotto, 8, I-06123 Perugia, Italy
redo@thch.unipg.it, franc@thch.unipg.it, lag@unipg.it
[2]Istituto di Scienze e Tecnologie Molecolari, CNR,
via Elce di Sotto, 8, I-06123 Perugia, Italy
franc@thch.unipg.it

Abstract. SUPSIM is a web application, based on the Java Servlet technologies, to compute ab-initio potential energy surfaces of molecular systems for subsequent dynamical properties calculation. We describe the architecture of SUPSIM, its current implementation, and its possible future developments to tackle chemical systems of increasing complexity.

1 Introduction

As the complexity of a computational problem grows, it is crucial, in order to continue being able to solve it, to design a computational system flexible enough to grow comparably in complexity. Solutions can be worked out only by gathering together sufficient computing power and the appropriate know-how. This is, indeed, the case of the a-priori realistic simulation of a molecular process for which only the grid [1] can provide a computing platform suitable to make possible the needed cooperation among the various competences and laboratories involved.

The a-priori simulator proposed for the Molecular Science community, and the related Virtual Organization (VO) Compchem [2], is GEMS (Grid Enabled Molecular Simulator) that is derived from SIMBEX [3] (an a-priori simulation of crossed molecular beam experiments). The prototype version of GEMS has been implemented as a demo application [4] of the NA4 work-package of the EGEE project [5]. GEMS is articulated in four main steps (see Fig. 1): the calculation of the ab-initio values of the potential energy surface (PES), the fitting of the ab-initio points, the integration of the nuclei dynamics equations and the final statistical analysis and visualization of the results. The current version of GEMS (GEMS.0) assumes that the PES has already been generated by fitting a LEPS [6] functional form to the calculated ab-initio values, then GEMS.0 runs the necessary batches of trajectories by taking the value of the LEPS parameters from a library, and renders the properties of the process on some virtual monitors. To build the next, more general, version of GEMS two new computational modules have been created: SUPSIM and FITTING.

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3980, pp. 675–683, 2006.

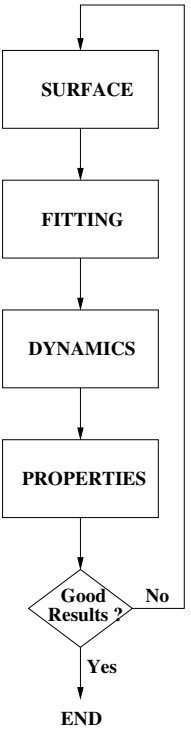

Fig. 1. A schematic diagram of the GEMS workflow. The first step (SURFACE) is the one to which SUPSIM corresponds. The other steps (FITTING, DYNAMICS and PROPERTIES) are devoted to the fitting of the potential energy values, the calculation of the dynamical properties and the statistical assembly of the observables quantities, respectively.

SUPSIM is the module devoted to the calculation of the potential energy surfaces using ab-initio methods. FITTING, instead, is the module that best fits the computed ab-initio points and stores the fitted values of the PES parameters (it can also build the related Fortran routine). The parameters (as well as the routine, conventionally called `Pesfunction`), are used for the subsequent dynamical calculations, carried out in the DYNAMICS module, for which either the Quasiclassical Trajectory or Quantum Mechanical approaches [7] are used. At the end, the last module PROPERTIES evaluates the observable properties of the system and of other relevant features of the chemical process by properly combining the calculated quantities.

The present paper is devoted to a description of SUPSIM. We illustrate the basic features of SUPSIM in section 2, give details of its design and implementation in section 3 and describe the communication frame in section 4. Further developments and the outlook are discussed in section 5.

2 What Is SUPSIM

SUPSIM is the module that will enable a GEMS user to submit the computation of a potential energy surface on a grid through a user friendly interface. At the present time SUPSIM has been deployed on the web with a test version reachable at the URL: `http://www.chemgrid.unipg.it:8080/`. Therefore, the present version of SUPSIM, that already displays some of the essential features of the complete product, is publicly available. Work is already in progress to insert SUPSIM in the GEMS workflow because its components have been already tailored for this purpose.

The PES calculation is articulated in three principal steps: parameters insertion, job submission, results retrieval. In the first step, the user is requested to specify the parameters of the calculation (such as, for example, the basis set and the theoretical method to be used, the electronic state parameters, the grid of nuclear positions where the electronic energy is to be computed, and so on). All the parameters can easily be specified via a simple HTML form. Then, the job for electronic energy computation can be submitted. Job submission can take place on an arbitrary grid. Once the necessary parameters are selected the user should choose a credential server, from where to retrieve a valid GLOBUS [8, 9] certificate, a suitable site where to run and a suitable quantum chemistry package to use. The present test version of SUPSIM runs on a local cluster of CHEMGRID (see section 5) and makes use of the ab-initio package MOLPRO [10]. The job is monitored using the standard GLOBUS [8] and OpenPBS [11] tools which notify the user about usage and the changes of state of the job.

At the end of the computation, all results can be collected from the hosts involved and inspected by the user to examine the key features of the computed surface, verify the suitability of the grid of points used, supplement, eliminate or recompute unconverged points, and generally take further action. At the end of this process, the computed PES values will be ready either to be used directly or to be fitted by the FITTING module. The computed PES values can also be graphically rendered or returned as an ASCII file.

3 SUPSIM Design and Implementation

SUPSIM is built using the Java Servlet technology [12] and runs in the Apache Tomcat Servlet container [13]. SUPSIM is actually divided in three different servlets:

- `SubCalc`. This servlet provides the user interface for job preparation and job submission. Actually this servlet builds the script used by the selected ab-initio program starting from the parameters specified by the user, and submits it for the computation.
- `ViewRes`. This servlet is devoted to monitor the progress of job execution and to retrieve, process and visualize the results.
- `CommErr`. This servlet is devoted to the management of the communication errors.

The first servlet, `SubCalc`, enables the user to specify the PES parameters and submit the job. In particular, the user specifies the parameters for the calculation using HTML forms. The present version of SUPSIM permits only the calculation of triatomic potential energy surfaces. The user must in general be able to select the basis set and the ab-initio method employed. At present, in order to provide a minimal qualitative description of the surface, use is made of the cc-pVDZ basis set [14] and of the valence CASSCF [14] level of theory. With a second HTML form it is possible to specify the parameters which determine the grid of nuclear geometries to be used for the calculation of the electronic energy. The nuclear positions can be specified either as internal coordinates or as bond-order [15] ones. Other parameters are concerned with the space/spin symmetry of the surface. This specification is related to the atoms constituting the system, so that, once the user has selected the molecular system, SUPSIM is able to build a list of all the possible symmetry and spin multiplicities. The user is then prompted for a list of selectable symmetries and spin multiplicities.

In general, the user should be able at this point (after obtaining a valid certificate) to select both the site or sites where to submit the calculation and the ab-initio programs to be used. He/she may be presented with a list of all the available ab-initio packages and prompted to select one. Otherwise, the grid interface will search the network for a list of sites offering the appropriate resource. At the moment the list of available sites and packages is not dynamic, but it is hard-written in an ASCII file.

After that, the servlet builds the script for the available quantum chemistry programs. `SubCalc` then generates a new thread that performs the job submission. As already mentioned, in the current version of SUPSIM the calculation is submitted on a local cluster of machines using the facilities offered by OpenPBS [11].

After submission, the user can obtain information on the status of the job by interrogating the second servlet, `ViewRes`. The first task of this servlet is, in fact, to retrieve the information on the job status. In the present implementation of SUPSIM this is achieved again by using OpenPBS services. The ViewRes servlet also provides the functionality to collect and display the results of the computation when this is completed. Two different display methods are available, a plain ASCII representation and a graphical rendering. If the user chooses the latter, the servlet builds and displays some images, presenting different views and sections of the surface along with the list of parameters used for the computation. All the images are built on the fly the first time that the user asks for the visualization. After this, the images are stored away to be retrieved on subsequent requests, until the computation is deleted. Similarly, the user can choose to visualize the PES values in ASCII format. The servlet then shows the contents of an ASCII file which contains some basic information on the calculation, such as the molecule, basis set and method used along with the calculated energies at each point of the chosen grid of geometries. Because all the data related to a single computation are stored in a separate directory, a user can retrieve, display and

use all the results of previous computations. In other words, a user has access to the history list of all her/his past computations, with the capability to manage it.

The third servlet, `CommErr`, is responsible for the proper handling of all communication errors. Communication is intended here as a transfer of information between SUPSIM and the other GEMS modules. When a PES calculation is completed, the data can be transmitted to subsequent GEMS modules, which in general reside somewhere else on the grid. This is in line with the fact that the overall application is designed to be an assembly of grid services implemented in various platforms, as is in the spirit of the Metalaboratories [16]. Because of the criticality of this function we devote the next section to a more detailed discussion of its implementation.

4 Communication

Communication between SUPSIM and GEMS takes place in two different phases. First, when it is required, GEMS asks SUPSIM to compute a certain set of PES values. This is achieved in GEMS by redirecting the user to the `SubCalc` URL. The second phase of the communication takes place in the reverse direction, from SUPSIM to GEMS. When the computation of the PES values is completed, SUPSIM transfers, upon user's request, the results to the FITTING module, and the user is then redirected to GEMS. More specifically (see Fig. 2), when the user logs into GEMS and indicates the chemical reaction to be simulated, it may happen that the necessary PES is already available in the user SUPSIM archive or somewhere else on the network. In this phase GEMS tries to locate the PES and, if it is available, it will simply retrieve it from the appropriate source. Otherwise, GEMS redirects the user to the `SubCalc` module of SUPSIM, to perform the electronic structure calculations. Some basic information on the computation, and in particular a unique identification number (ID), is transferred to SUPSIM. The ID number is stored by SUPSIM along with all other parameters of the computation and can be used, at any time, to identify the specific user

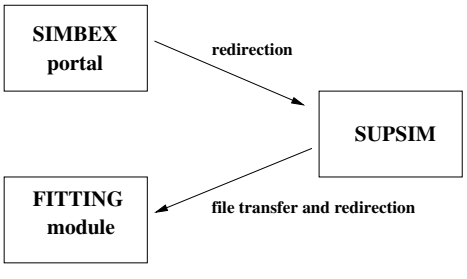

Fig. 2. A schematic representation of the GEMS communication process. The process takes place in two different phases: *a*) when it is required, GEMS asks SUPSIM to compute a certain set of PES values, *b*) when the computation is over, SUPSIM transfers the results to the FITTING module.

session. Once the job carrying out the calculation of the PES has been submitted, the information necessary to identify the user and the job for future use is recorded. The user can therefore logout and login again at some later time to resume work. In general, the user may also be e-mailed with information about the status of the job.

Once the PES calculation is finished and the user decides to continue with the dynamical simulation, SUPSIM transfers the PES values to GEMS. The file transfer takes place through a TCP communication and a simple protocol has been designed to allow this file transfer. The transaction can be summarized as follows:

- The client (SUPSIM) sends an *HELLO[id]* command, where *id* is the identification number related to the specific PES computation that `SubCalc` has received. The server checks if the *id* is valid and determines the user and the computation to which the files to be transmitted belong. If this phase is successful, the server send an *OK[id]*.
- After the first phase of communication is finished, the file transfer itself can be carried out. First of all the client sends a *FILE[nfile]* message to specify how many files there are to be transferred. If the server is able to receive them, an *OK[nfile]* message is sent back. Next, the server must be informed about the size of the incoming files. This is conventionally measured in units of transfer buffers, set at 4096 bytes. For each file to be transmitted, the client sends first the dimension with a *SET[dimfile]* message and waits for the return of an *OK[dimfile]* message from the server. Then the client starts sending the *dimfile* buffers of data one by one and, for each buffer it receives, the server responds with an *OK[n]* message, where n is the buffer counter.
- The TCP transaction is closed with an *ENDCOM[id]* message from the client followed by an *OKENDCOM[id]* from the server.

If the file transfer is successfully completed, the FITTING process can take control, otherwise the `CommErr` servlet will visualize the transaction error which has occurred. The use of a simple TCP connection rather then the GlobusFTP protocol [20] has historical reasons: originally, GEMS was built in a generic grid environment with no relationship with the Globus middleware. Switching to the GlobusFTP standard is a simple change that will be incorporated in the future versions of SUPSIM.

5 Developments and Future Work

In order to build an effectively grid-enabled version of SUPSIM, work will be undertaken to make it run as a testbed on CHEMGRID [17, 18] and on the EGEE production grid environment. This work will be articulated two steps. In the first step we shall enable `SubCalc` to submit the job using the GLOBUS environment [8] instead of the OpenPBS one. In the second step we shall modify `ViewRes` to enable it to retrieve job progress information and final results on the grid. We also envisage the possibility of having a more general and portable output format for the PES instead of a simple ASCII file.

Fig. 3. A map of the CHEMGRID sites

Depending on funding evolution SUPSIM will, as GEMS, develop along a double track. The first track is that of the production grid of EGEE. The second track is that of CHEMGRID, a project developed within GRID.IT [18]. CHEMGRID involves several Italian chemistry laboratories and is built using Globus toolkit 3.2 (see Fig. 3). Each site has, at present, a small Beowulf-type cluster [22] (see Tab. 1 for details), with a front-end which represents the public access point to the grid resources and two or more worker nodes. The request for a job is processed from the front end and is submitted to a worker node through the local resource management interface (OpenPBS, or LSF [19]).

Each site of the grid offers to the users the possibility of using various quantum chemistry packages. This will allow the SUPSIM users to choose a site and the ab-initio package best suited to his/her needs. The servlet will be shaped accordingly to build the script appropriate to the chosen ab-initio program and will submit the computation on the selected sites. At this point the other servlet, `ViewRes`, will be able to retrieve job processing information and the final results.

The retrieval of certificates will be implemented using a MyProxy server [21] (an on-line credential repository). A MyProxy server is already installed and usable for test and development on the front-end of the Perugia site, where also SUPSIM is currently installed.

Table 1. Characteristics of the Beowulf clusters involved in CHEMGRID

Site location	*CPU*	*Number of nodes*	*RAM*	*HD*
Perugia	Intel PIV HT 3.06 GHz	8	1024 MB	80 GB
Perugia	Intel PIV HT 2.08 GHz	1	1024 MB	80 GB
Bologna	AMD Athlon K7 1.2 GHz	4	512 MB	60 GB
Bologna	AMD Athlon K7 700 MHz	4	512 MB	26 GB
Padova	Intel PIV, 2.4 GHz	4	512 MB	40 GB
Napoli	AMD Athlon 1.0 GHz	4	256 MB	20 GB
Milano	Intel PIII Coppermine 1.0 GHz	8	1024 MB	16 GB
Bari	Intel Xeon HT 3.2 GHz	3	2048 MB	40 GB
ENEA	Intel PIV 1.8 GHz	4	512 MB	70 GB

Job submission and resource discovery/selection will be managed in the future by a grid service broker such as GridBus [23]. In such case it will be up to the grid broker to dynamically select a specific site or sites, starting from some basic information on the necessary resources, such as, for example the required ab-initio program, memory and disk space capabilities, machine architecture and so on.

An important point that will need attention is devising an appropriate and standardized format for the computed PES representation. This is an open problem that must be discussed in the general context of knowledge representation paradigms [24] and the specific representation of chemistry data. In recent years, the general problem of data representation and format has been widely discussed, especially with regard to the portability of data (see for example [25]). An appropriate solution for our purposes would be the adoption of an XML schema such as the CML [26], or QMCL [25], which is more suitable for quantum chemistry problems. Alternatively, the RDF [27, 28] format could be adopted. In this case, the PES could generally be made available and published on the WEB not only for a human, but also for an arbitrary software agent. It is clear that many of the ideas and tools exploited in the SUPSIM project can generally be used also for other complex quantum chemistry applications.

Acknowledgments

Financial support to this work has been provided by MIUR, ASI, CNR and COST in Chemistry (Action D23). Specific mention needs be made to the italian MIUR FIRB GRID.IT project (RBNE01KNFP) on High Performance Grid Platforms and Tools.

References

1. Foster, I. and Kesselman, C. Eds.: The Grid: Blueprint for a Future Computing Infrastructure, Morgan Kaufmann Publishers, USA (1999).
2. Gervasi, O., Dittamo, C., Laganà, A.: A Grid Molecular Simulator for e-Science, Lecture Notes in Computer Science, **3470**, (2005), 16-22.

3. Gervasi, O., Laganá, A.: SIMBEX: a portal for the a priori simulation of crossed beam experiments. Future Generation Computer System, **20**(**5**), (2004), 703-716.
4. The EGEE (Enabling Grids for E-science in Europe) project: `http://public.eu-egee.org/`
5. Laganà, A., Gervasi, O.: GEMS: an EGEE application migration report. The First EGEE Assessment workshop, Geneva, 9-11 February 2005.
6. Polanyi, J.C., Schreiber, J.L.: The Dynamics of Bimolecular Reactions, Physical Chemistry-An Advanced Treatise, Volume VIA, Kinetics Gas Reactions, Eds., H. Eyring, W. Jost and D. Henderson (Academic Press, New York, 1974), Ch. 6, p. 383.
7. Laganà, A., Riganelli, A.: Lecture Notes in Chemistry, **75**, (2000).
8. The Globus Project: `http://www.globus.org/`.
9. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A Security Architecture for Computational Grids. Proc. 5th ACM Conference on Computer and Communications Security Conference, pp. 83-92, 1998.
10. Werner, H.-J., P. J. Knowles, P. J.: MOLPRO, a package of *ab-initio* programs version 2002.6. (`http://www.molpro.net/`).
11. OpenPBS home page: `http://www.openpbs.org/`.
12. Sun Java Servlet home page: `http://java.sun.com/products/servlet/`.
13. Apache Jakarta Tomcat home page: `http://jakarta.apache.org/tomcat/`.
14. Helgaker, T., Jørgensen, P., Olsen J.: Molecular Electronic-Structure Theory, Wiley, June 2002.
15. Laganà, A., Ferraro, G., Garcia, E., Gervasi O., Ottavi, A.: Potential energy representations in the bond order space, Chem. Phys., **168**, (1992), 341.
16. Laganà, A.: Second Annual report of the activities of the workpackage 13 of GRID.IT (see ref. [18]) Annex: Structuring Chemistry Metalaboratories for Grid servicing.
17. The Chemgrid Project home page: `http://www.chemgrid.unipg.it/`.
18. GRID.IT: An Italian National Research Council Project on High Performance Grid Computing (RBNE01KNFP) (`http://www.grid.it`).
19. LSF (Load Sharing Facility): `http://www.platform.com/`.
20. GridFTP Protocol Specification (Global Grid Forum Recommendation GFD.20). W. Allcock, editor. March 2003.
21. Novotny, J., Tuecke, S., Welch, V.: An Online Credential Repository for the Grid: MyProxy. Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.
22. See e.g. `http://www.beowulf.org`.
23. Buyya, R., Venugopal, S.: The Gridbus Toolkit for Service Oriented Grid and Utility Computing: An Overview and Status Report, Proceedings of the First IEEE International Workshop on Grid Economics and Business Models (GECON 2004, April 23, 2004, Seoul, Korea), 19-36pp, ISBN 0-7803-8525-X, IEEE Press, New Jersey, USA. (`http://www.gridbus.org/`).
24. Goble, C. A., Roure, D. De: The Grid: an application of the semantic web, ACM SIGMOD Record, **31**, (2002), 65-70.
25. Rossi, E., Emerson, A., Evangelisti, S.: Common Data Format for Program Sharing and Integration, Lecture Notes in Computer Science, **2658**, (2003), 316-323.
26. Murray-Rust, P., Rzepa, H. S., Wright, M.: Development of Chemical Markup Language (CML) as a System for Handling Complex Chemical Content, New J. Chem, (2001), 618-634. (`http://www.xml-cml.org`).
27. Resource Description Framework: `http://www.w3.org/RDF/`.
28. The Semantic Web: `http://www.w3.org/2001/sw/`.