Towards a GRID based Portal for an a priori Molecular Simulation of Chemical Reactivity

Osvaldo Gervasi[1], Antonio Laganà[2], and
Matteo Lobbiani[1]

[1] Department of Mathematics and Informatics, University of Perugia,
via Vanvitelli, 1, I-06123 Perugia, Italy
osvaldo@unipg.it
http://www.unipg.it/~osvaldo
[2] Department of Chemistry, University of Perugia,
via Elce di Sotto, 8, I-06123 Perugia, Italy
lag@unipg.it
http://www.chm.unipg.it/chimgen/mb/theo1/text/people/lag/lag.html

Abstract. The prototype of an Internet Portal devoted to the Simulation of Chemical reactivity has been implemented using an engine running in parallel. The application makes use of PVM, and it has been structured to be ported on a GRID environment using MPI.

1 Introduction

This paper illustrates the development and the implementation of a Problem Solving Environment (PSE)[1] for the Simulation of Chemical Reactive Processes. The application is based on an Internet Portal connected to a computer grid[2, 3], updating a set of visualization facilities and to monitor in real-time the evolution of the simulation.

As a prototype PSE we consider here an a priori Simulator of Molecular Beam Experiments, **SIMBEX**, for atom-diatom reactions[4]. Crossed Molecular Beams are a crucial experiment providing a stringent test for the understanding of molecular interactions and the rationalization of chemical processes[5]. Their a priori simulation is a high demanding computational procedure that for its progress relies on the advance in computing technologies. For this reason **SIMBEX** has been specifically designed for distributed computing platforms.

The rapid evolution of networking technologies is, in fact, making it feasible to run complex computational applications on platforms articulated as a geographically dispersed large clusters of heterogeneous computers ranging from versatile workstations to extremely powerful parallel machines (Computing Grid). This opens the perspective of carrying out realistic simulations of complex chemical systems by properly coordinating the various computational tasks distributed over the network. Such an approach challenges also the exploitation of a remote cooperative use of software, hardware and intellectual resources belonging to a cluster of various research Centers and Laboratories. On this ground Metalaboratories devoted to various complex computational applications in chemistry are

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2331, pp. 956–965, 2002.

being established in Europe[1] dealing with different targets in complex computational chemistry[8].

Among this, **SIMBEX** is an application aimed at designing a distributed version of the Crossed Molecular Beam Simulator prototype reported in the literature a few years ago [9, 10].

The basic articulation of the Web structure of **SIMBEX** consists of a client, a back-end and a middleware component. The client level consists of a Web browser connected to the Portal: the web pages drive the user to the selection of the application, to the collection of the input data and to the recollection and presentation of the results. The authentication of the user is taken care by the initial Web page. Then the user is offered a list of applications to run on the back-end system by activating the related hyperlinks. After the configuration, the simulation starts and the user can follow the quantities of interest in Virtual Monitors that are Java Applets downloaded from the Web server to the client. The back-end presently consists of a cluster of workstations distributed among the Department of Chemistry in Perugia (Italy), where crossed beam experiments are run, the Computer Center of the University of Perugia (Italy), that also shares with the cluster a section of its IBM SP2, and the Department of Physical Chemistry of the University of the Basque Country in Vitoria (Spain). An extension of the cluster to other Laboratories is under way.

The middleware layer consists of a static Web server (Apache), a Java Web Server (Tomcat) and a daemon called `SimGate`. The Web server deals with the client and with the Java server handling the requests of the users. `SimGate` is devoted to handle the communication with the applets, freeing the farmer of such task.

The paper is articulated as follows: In section 2 the architecture of the Internet portal is discussed. In section 3, the articulation of related computational procedures is analysed in order to single out models and templates for their distributed use. In section 4, the particular case of a prototype atom diatom reaction is discussed for illustrative purposes.

2 The Architecture of the Internet Portal

To allow the user to access the Problem Solving Environment a Java2 enabled Web browser is used. The problems related to the management of the distributed environment, and the communications between the various components are solved at the Server Web level.

The product has been developed using *Freesoftware* components. In particular use has been made of the Apache Web Server, the Tomcat Java Web Server, Java2, MySQL and PVM, powered by Linux RedHat.

[1] To incentive the gathering of research Laboratories having complementary expertises on common research projects, the European Community has launched within the COST (Collaboration in Science and Technology) in Chemistry[6] initiative the Action D23[7].

Access to the Portal requires an account. We have used MySQL to save the data of the users and to handle the authentication phase. Because of the multiuser environment, multiple requests to the web server are dealt using the parameter *SessionID.*

After the user has defined the configuration of the simulation, the Java Servlets start ABCtraj, the parallel application, as a Farmer process running on the Web server and the Worker programs running on the PVM distributed nodes. The communication between the Servlets and ABCtraj occurs trough a Unix Socket. The Servlets start up also a daemon called SimGate devoted to the communication between the Java Applets related to the Virtual Monitors and the server Web for the on-line update of the Virtual Monitors. The communications between SimGate and the applets occurs through a stateless protocol that makes use of a TCP/IP socket using a service port chosen between a set of reserved values. During the simulation the applet asks for updates to SimGate, that returns the values received by ABCtraj. Different contemporary simulations are easily handled by the multithread nature of Java.

The communication between the Farmer and SimGate has the following syntax:
UPD MON $[0-11]$ ID $SessionID$ DATA - $a_1, a_2, ..., a_{20}$
with a_i being floating point numbers. The $SessionID$ parameter allows SimGate to control that the flow originates from the right Farmer (the one associated to $SessionID$). SimGate answers with a success or an error message.

The protocol between SimGate and the applet is slightly different: the applet asks for new data, specifying its own $Monitor\ number$ and $SessionID$:
GET MON $[0-11]$ ID $SessionID$
to which SimGate answers sending the data or an error message. The amount of data exchanged between the server web and the applets related to the Virtual Monitors is estimated in 0.2K bytes per virtual Monitor for each update process.

3 The Parallel Application ABCtraj

As already mentioned in Section 2, the computational engine of the Problem Solving Environment is the ABCtraj parallel application, that allows the study of atom-diatom reactivity using a Quasiclassical trajectory approach[11].

ABCtraj has been structured as a task farm. The Farmer section of the code is sketched in Fig.1. The Worker portion of the code is sketched in Fig.2.

The Farmer receives the data from the Servlets, performs the initialization and calculates a seed for each trajectory to be computed. The seed will be used by the Worker to generate the string of pseudorandom numbers needed by each trajectory in a deterministic way.

The Farmer enrolls all Workers in the PVM[12] environment and sends to the Worker a bulk of common data to be used for the entire simulation. It also sends then to each Worker a trajectory number, the related random seed and waits for one of the Workers to complete its work unit. When a Worker has finished its

```
                         FARMER code
Receive from Servlets the input data via Unix Socket
Initialize the PVM environment, enrolling the worker program to $Workers$
Calculate a seed for each trajectory
Send initial data to all $Workers$

WHILE all_Workers_complete_work
    Waits for a $Worker$ to complete its work unit
    Send to the same $Worker$ a new work unit
    Update SimGate
END WHILE

Write out final histograms
Shutdown the PVM environment
Exit
```

Fig. 1. Scheme of the FARMER portion of the trajectory code for atom-diatom systems.

work unit, the Farmer receives the data and updates `SimGate` and sends to the Worker a new work-unit, until the last trajectory has been calculated.

The final section of the code carries out the remaining (non iterative) calculations relevant to the evaluation and the print out of rate coefficients, cross sections and product distributions for the different reaction channels.

After the Worker receives a trajectory number and the related random seed it starts the integration of the trajectory step by step to follow the evolution in time of positions and momenta. When the trajectory ends, the Worker sends the results to the Farmer and waits for a new work unit to be assigned. If no more trajectories have to be run ($trajectory\ number = 0$) statistics manipulations of the trajectory results are performed to evaluate reaction probabilities and cross sections, product energy and space distributions.

4 The atom diatom H + I Cl reaction case study

To compare with an already investigated system, we discuss here the case of the atom-diatom reaction $H+ICl \rightarrow HI+Cl, HCl+I$. This is a simple heavy heavy light system for which all parameters have been already given in ref.[11] where a simulation environment based on a Graphical user Interface (GUI) developed in X-Windows and Motif environments was discussed.

In Fig. 3 is shown the entry point of the Portal, from wich the researcher has two main possibilities: start a new simulation or analize the Virtual Monitors of a simulation already carried out in the past.

Before starting the simulation, the user must *login* into the PSE. As already mentioned this step is necessary to control who is using the PSE. However it is

```
                              WORKER code
Receive preliminar data from $Farmer$

Set initializations
Calculate auxiliary variables

WHILE not_last_trajectory
    Receive number of trajectory and related seed for
        random number generation from $Farmer$

    Generate the subset of pseudorandom numbers characterizing the trajectory
    Calculate the corresponding initial conditions
        LOOP on time
            IF (asymptote is not reached)
                THEN perform integration step
                ELSE exit time loop
            ENDIF
        END the time loop
    Calculate product properties
    Update statistical indexes
    Send to the $Farmer$ the trajectory results to update the Virtual Monitors
END WHILE

Leave PVM
Exit
```

Fig. 2. Scheme of the WORKER portion of the trajectory code for atom-diatom systems.

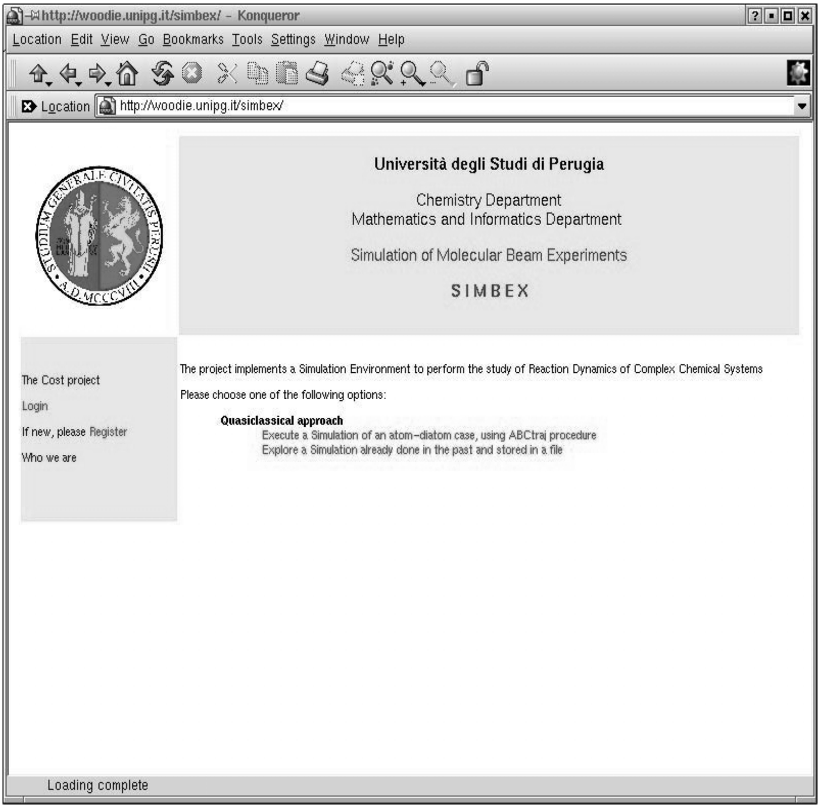
also necessary to allow the user to build a customized environment. From this page the user can select the type of application he wishes to run (presently, only `ABCtraj` is available). The user can also select the type of Database to be used. The default Database contains all Chemical Systems known by the Portal and available on the various sites of the grid, on Databanks of the Web and on the user's Database that contains the data and the systems already defined by the user. The Chemical System that will be used for the simulation is selected from a selection box built from the directories available on the Database chosen. After this selection, the researcher should choose one of the files listed in the directory to define the configuration of the simulation. In Fig. 4 is shown how to tune some parameters of the configuration. After the configuration phase, the application `ABCtraj` and the parallel environment are activated and the simulation starts.

The user is also enabled to access from the Web the Virtual Monitors he likes (at the moment only the Angular Distribution and the Opacity Function Monitors are activated) from the configurations he wants to study. When the hyperlink of a selected Virtual Monitor is accessed, a Java Applet is downloaded from the HTTP server to the researcher's client and the data of the simulation are shown and updated dinamically. In Fig. 5 an example of the $H+ICl \rightarrow HI+Cl$ Angular Distribution Virtual Monitor produced while the simulation was running is shown. The production of this or other Virtual Monitor at the experimental site supplies useful indications to the experimentalists on how modify measurement conditions.

5 Conclusions

In this paper we have discussed a prototype implementation of an Internet Portal for the distributed simulation of crossed beam processes. The system considered (the atom-diatom reaction $H + ICl \rightarrow HI + Cl, HCl + I$ for which a previous study has been made and the same parameters have been used) has made it possible to carry out a comparison with results obtained using an older version of the simulator. This implementation has shown that **SIMBEX** is a suitable test bed for grid approaches in which computing resources and complementary know how have to be gathered togheter. In particular the grid implementation of **SIMBEX** allows theorists to work on the application in variuos places and experimentalists to start their simulations at the site where the experiment is being carried out. The structure of the simulation is such that the calculations can be spread over a wide network of computers to run concurrently. This allows the simultaneous dealing of a large number of events and a real time clock of the simulation.

Extensions of the procedure to other phases of the simulation are in progress as well as more complex systems and a richer set of interfaces and monitors.


http://woodie.unipg.it/simbex/ - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location http://woodie.unipg.it/simbex/
The Cost project
Login
If new, please Register
Who we are
Loading complete

Università degli Studi di Perugia

Chemistry Department
Mathematics and Informatics Department

Simulation of Molecular Beam Experiments

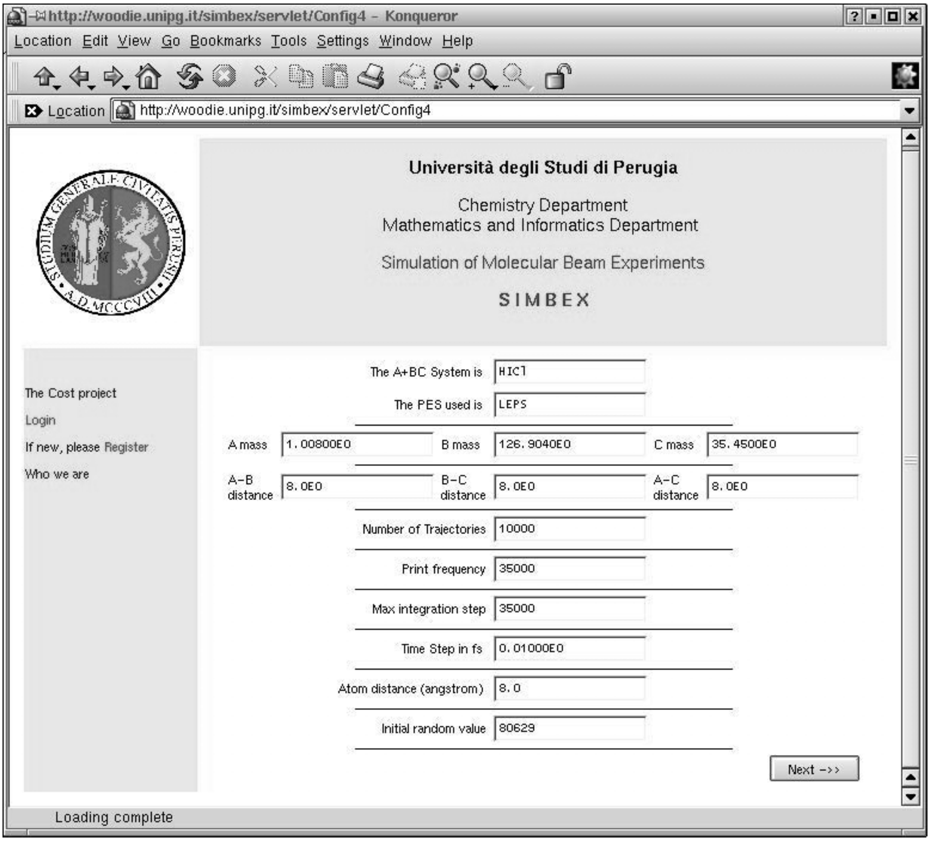
SIMBEX

The project implements a Simulation Environment to perform the study of Reaction Dynamics of Complex Chemical Systems

Please choose one of the following options:

Quasiclassical approach
Execute a Simulation of an atom–diatom case, using ABCtraj procedure
Explore a Simulation already done in the past and stored in a file

Fig. 3. The Portal entry point of **SIMBEX**


http://woodie.unipg.it/simbex/servlet/Config4 – Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location http://woodie.unipg.it/simbex/servlet/Config4
The Cost project
Login
If new, please Register
Who we are
Next –>>
Loading complete

Università degli Studi di Perugia

Chemistry Department
Mathematics and Informatics Department

Simulation of Molecular Beam Experiments

SIMBEX

The A+BC System is HICl

The PES used is LEPS

A mass	1.00800E0	B mass	126.9040E0	C mass	35.4500E0
A–B distance	8.0E0	B–C distance	8.0E0	A–C distance	8.0E0

Number of Trajectories 10000

Print frequency 35000

Max integration step 35000

Time Step in fs 0.01000E0

Atom distance (angstrom) 8.0

Initial random value 80629

Fig. 4. Definition of the configuration of the simulation, by tuning the parameters related to the Chemical System considered.

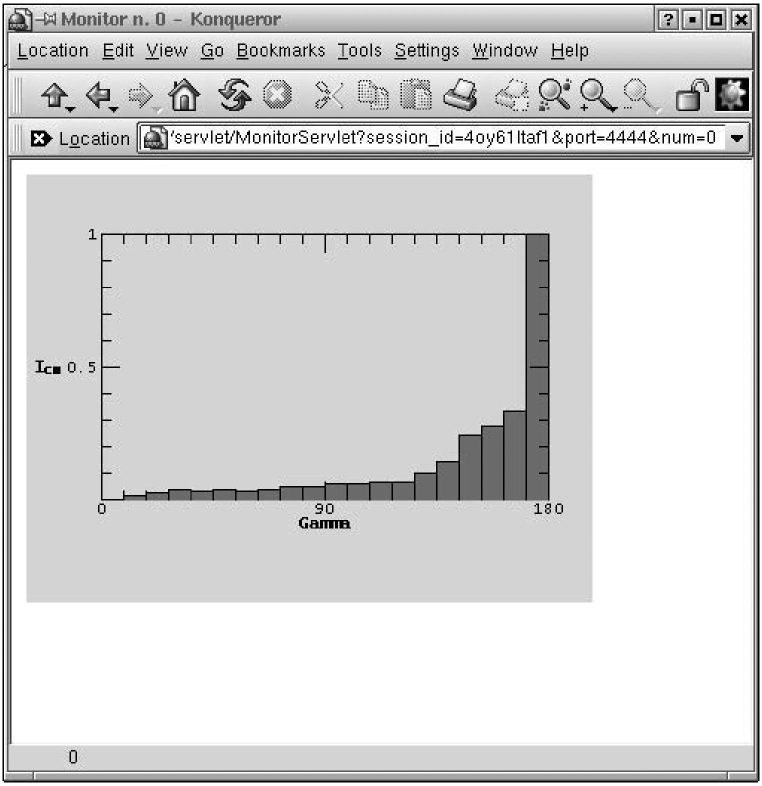

Monitor n. 0 – Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location
'servlet/MonitorServlet?session_id=4oy61ltaf1&port=4444&num=0
1
Icm 0.5
0
90
180
Gamma
0

Fig. 5. Example of Virtual Monitor of the Angular Distribution for $H + ICl \rightarrow HI + Cl$ reaction, taken while the simulation is running

6 Acknowledgments

This research has been financially supported by MIUR, ASI and CNR (Italy) and COST (European Union).

References

1. Gallopoulos, S., Houstis, E., Rice, J., Computer as Thinker/Doer: Problem-Solving Environments for Computational Science, IEEE Computational Science and Engineering, Summer (1994).
2. Foster, I. and Kesselman, C. Eds., The Grid: Blueprint for a Future Computing Infrastructure, Morgan Kaufmann Publishers, USA (1999).
3. Baker, M., Buyya, R., Laforenza, D., The Grid: International Efforts in Global Computing, SSGRR2000, L'Aquila, Italy, July (2000).
4. Gervasi, O., COST in Chemistry Action N.D23, Project 003/2001, *SIMBEX: a Metalaboratory for the a priori Simulation of Crossed Molecular Beam Experiments*
5. Casavecchia, P., Chemical Reaction Dynamics with Molecular Beams, Rep. Prog. Phys. **63** (2000), 355-414. J. Chem. Phys., **73** (1980), 2833-2850.
6. http://www.unil.ch/cost/chem
7. Laganà, A., METACHEM: Metalaboratories for cooperative innovative computational chemical applications, METACHEM workshop, Brussels, November (1999) (see also [4]).
8. Gavaghan, H., Nature, **406** (2000), 809-811.
9. Gervasi, O., Cicoria, D., Laganà, A., Baraglia, R., Pixel, **10** (1994) 19-26.
10. Gervasi, O, Cicoria, D., Laganà, A., Baraglia, R., Animation and parallel Computing for the Study of Elementary Reactions, Scientific Visualization '95, R. Scateni Ed., CRS4, Cagliari, Italy, World Scientific (1995), 69-78
11. Laganà, A., Gervasi, O., Baraglia, R., and Laforenza, D., From parallel to Distributed Computing for Reactive Scattering Calculations, Int. J. Quantum Chem.: Quantum Chem. Symp., **28** (1994), 85-102.
12. Beguelin, A., Dongarra, J., Geist, A., Manchek, R., and Sunderam, V., A user's guide to PVM Parallel Virtual Machine,Oak Ridge National Laboratory, Tennessee, 1992.
    Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V., PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing, MIT Press, Scientific and Engineering Computation, Janusz Kowalik, Editor, Massachusetts Institute of Technology, 1994.
    (http://www.netlib.org/pvm3/book/pvm-book.html);
    http://www.epm.ornl.gov/pvm/pvm_home.html;
    http://www.netlib.org/pvm3