A Grid execution model for Computational Chemistry Applications using the GC3Pie framework and AppPot

Alessandro Costantini[*]
INFN-CNAF — National Institute of Nuclear Physics, Bologna, Italy.
IGI — Italian Grid Infrastructure, Italy.
Department of Mathematics and Informatics, University of Perugia, Italy.
E-mail: alessandro.costantini@cnaf.infn.it

Riccardo Murri
Grid Computing Competence Centre, University of Zurich, Switzerland.
E-mail: riccardo.murri@gmail.com

Sergio Maffioletti
Grid Computing Competence Centre, University of Zurich, Switzerland.
E-mail: sergio.maffioletti@gc3.uzh.ch

Sergio Rampino
Department of Chemistry, University of Perugia, Italy.
E-mail: ser_ram@dyn.unipg.it

Antonio Laganà
Department of Chemistry, University of Perugia, Italy.
E-mail: lag@unipg.it

Porting and running of computational chemistry applications on distributed systems have been performed for a set of quantum mechanics scattering programs within a collaboration between the Grid Computing Competence Centre (Zurich), the Computational Dynamics and Kinetics Group (Perugia) and the Italian Grid Infrastructure (Bologna). For this purpose the high throughput execution framework GC3Pie and the AppPot cloud/grid virtual machines have been used to implement a grid based workflow dealing with the central blocks of an *ab initio* Molecular simulator. These blocks carry out both the fitting of the potential energy values computed using ab initio methods and the evaluation of some atom diatom quantum reactive scattering properties. Related performances are discussed and compared with other approaches.

EGI Community Forum 2012 / EMI Second Technical Conference,
26-30 March, 2012
Munich, Germany

[*]Speaker.

1. Introduction

Significant efforts are being paid by some members of the COMPCHEM Virtual Organization (VO) [1] to implement its layered structure (users, software providers, grid deployers, stakeholders) both for their active and passive roles. To this end COMPCHEM has developed (and continues to develop) tools to provide its user of the first layer with instruments for an optimized usage by the user him/herself or by other users. Moreover the VO is developing tools to encourage its members of the second layer to implement, validate and use software for an effective Service Oriented Approach (SOA) approach [2]. In this spirit, the high availability of computer power on Grid platforms is being exploited for the implementation of distributed complex computational chemistry suites of codes. This is, indeed, the case of the user friendly Grid empowered versions of the molecular simulators SIMBEX [3] and GEMS [4] developed by some members of the community in the recent past having in mind the needs of crossed molecular beam studies. More recently this has prompted also the development of appropriate frameworks (like GriF [5]) allowing to define event-related dependencies between complementary applications (which may involve different computer environments) in a workflow-like fashion. These are the activities that some COMPCHEM members are carrying out in cooperation with the COST Action CM1002 called COnvergent Distributed Environment for Computational Spectroscopy (CODECS) [6] to design and develop a collaborative grid empowered simulator for Molecular Spectroscopy. A similar effort is being paid by other members of COMPCHEM within the activities of the COST Action CM901 called Detailed Chemical Kinetic Models for Cleaner Combustion [7] to design and develop a collaborative grid empowered simulator for Cleaner Combustion and in a cooperative endeavour with WeNMR [8] to design a molecular simulator for NMR studies.

A fundamental support to SOA collaborative activities is given by workflows [9] which facilitate the composition and coordination of different programs in a structured and flexible execution scheme.

The work presented here is the result of a collaboration among the Grid Computing Competence Center, University of Zurich (GC3) [10], the Computational Dynamics and Kinetics (CDK) research group of the University of Perugia (UNIPG) [11], the Italian Grid Initiative (IGI) [12] and the already mentioned COMPCHEM (VO) of the European Grid Initiative (EGI) [13] aimed at developing high throughput workflows by taking some complex computational chemistry problems as use cases. Such workflow developing effort has made use of the high throughput execution framework GC3Pie [14] and the AppPot [15] cloud/grid virtual machines (both developed by the GC3 group) and was addressed to the implementation of a routine suited to perform the global fitting of the potential energy surfaces for triatomic systems and a quantum mechanical atom-diatom reactive scattering program both belonging to the set of computational applications in which the central block of GEMS, the already mentioned Grid Empowered Molecular Simulator of COMPCHEM, is articulated.

In this paper we discuss the potentialities of this approach by focusing on the compiling procedure of some blocks of the present version of GEMS (which deal with a least mean square deviation problem and the integration of a set of coupled second order differential equations) and on the collecting, in an automatic way, distributed cycles for the considered simulation using several grid computing resources simultaneously. The specific purpose of the work reported here is twofold. a) the creation of user friendly grid applications beneficial for the computational chemistry community; b) the development of tools applicable by other groups to port parameter study style applications onto production grids.

Accordingly, in section 2 details of the computational chemistry programs considered as use case are provided; in section 3 attempts from previous and related work are recapitulated; in section 4 the articulation of the developed Grid based workflow and related tools are illustrated; in section 5 the implementation of the considered case study and the obtained results and performances are analyzed. in section 6 the implementation of an advanced application of the workflow to a heavier three different atom system is discussed. Our conclusions are summarized in section 7.

2. The outline of the considered computational applications

GEMS [4] has been designed as a workflow and has been rationalized in a sequence of three computational sections:

- INTERACTION is the first section of GEMS and carries out the theoretical step of the calculations which determine the electronic structure of the system.
- DYNAMICS is the second section of GEMS and carries out the theoretical step of the calculations which determine the dynamics of the nuclei of the system.
- OBSERVABLES is the final section of GEMS and carries out the necessary statistical and model treatments of the outcomes of the theoretical calculations to provide an a priory estimate of the measured properties of the system.

In order to decompose the computational problem, each section has been partitioned in several blocks. In particular, FITTING is the block of INTERCATION devoted to the representation of the ad hoc (or by purpose generated or taken from the literature) *ab initio* potential energy values by means of a suitable global functional form. Such global analytical representation, usually called Potential Energy Surface (PES), can be generated in different ways [16]. In the presently available version of GEMS the FITTING block is specialized in atom diatom systems and produces a PES fortran routine based on the formulation of the three atom (A, B, C) potential as a many-body expansion [17] in the internuclear distances as follows:

$$\begin{aligned} V_{ABC}(r_{AB}, r_{AC}, r_{BC}) = & V_A^{(1)} + V_B^{(1)} + V_C^{(1)} \\ & + V_{AB}^{(2)}(r_{AB}) + V_{AC}^{(2)}(r_{AC}) + V_{BC}^{(2)}(r_{BC}) \\ & + V_{ABC}^{(3)}(r_{AB}, r_{AC}, r_{BC}). \end{aligned} \tag{2.1}$$

This means that the potential is written as a sum of three monoatomic terms ($V_A^{(1)}$, $V_B^{(1)}$, $V_B^{(1)}$), three diatomic terms ($V_{AB}^{(2)}$, $V_{AC}^{(2)}$, $V_{BC}^{(2)}$), plus one triatomic term ($V_{ABC}^{(3)}$). The one-body terms are the energies of the separated atoms in their corresponding electronic state (three constant values). The two-body terms correspond to the diatomic IJ (with I and J being one of the three A, B, C atoms) potential energy curves (which include the nuclear repulsion) and are usually expressed as low order polynomials either of the inverse or of the damped IJ internuclear distances r_{IJ}. As this is the case of the GFIT3C routine used in the present work [18], they can also be expressed as a low order polynomial of the bond order variables ($n = exp[-\beta(r_{IJ} - r_{IJe})]$), in which r_{IJe} is the diatomic equilibrium distance, or of mixed variables like $\xi_{IJ} = r_{IJ}exp(-\beta_{IJ}r_{IJ})$ in which ξ_{IJ} is optimized to represent the long-range term of the two-body potential. The three-body term $V^{(3)}_{ABC}$ corresponds to the residual interaction due exclusively to three atoms forces. The advantage of this development is that, if we impose certain restrictions on the internuclear distances, the correct asymptotic limits are obtained. Its value is easily obtained by subtracting the one and two body terms to the (if necessary adjusted to reproduce the experimental ergicity of the process or spectroscopic information on stable intermediates) calculated *ab initio* values. The three body terms is also formulated as a polynomial in the above mentioned variables (lets call them basis functions $b_k(\mathbf{r})$ with $\mathbf{r}$ being the coordinates adopted to describe the potential) and a_k the related coefficients.

To the end of calculating the values of the a_k coefficients of the adopted basis best fitting the (adjusted) *ab initio* values the following equation

$$V_{ABC}(\mathbf{r}) = \mathbf{a}^T\mathbf{b}(\mathbf{r}) = \sum_{k=1}^{K} a_k b_k(\mathbf{r}) \tag{2.2}$$

(where the superscript T means transpose and K is the number of basis functions) has to be solved.

Among the central blocks of GEMS, DYNAMICS is the one calculates the exact quantum three-dimensional state-to-state partial reactive probability $P^J_{vjv'j'}(E)$, of the atom-diatom reaction

$$\mathrm{A} + \mathrm{BC}(v, j) \rightarrow \mathrm{AB}(v\prime, j\prime) + C \tag{2.3}$$

in which the reactant diatomic molecule BC is in its v and j vibrational and rotational states (unprimed quantities) while the corresponding product states are v' and j' (primed quantities). The $P^J_{vjv'j'}(E)$ values calculated at a given value of the total energy E and of the total angular momentum quantum number J are then combined to estimate some observable quantities of the considered process. For this purpose $P^J_{vjv'j'}(E)$ is formulated in terms of the square modulus of the corresponding detailed $\mathbf{S}$ matrix elements $S^J_{vjKv'j'K'}(E)$ as follows:

$$P^J_{vjv'j'}(E) = \frac{1}{(2K_{\max}+1)} \sum_{K=-K_{\max}}^{K_{\max}} \sum_{K'=-K'_{\max}}^{K'_{\max}} \sum_{p=0}^{1} \left| S^{Jp}_{vjKv'j'K'}(E) \right|^2 \tag{2.4}$$

where $K_{\max} = \min(j, J)$, K is the helicity quantum number (the discrete body-fixed projection of the total angular momentum $\mathbf{J}$) and p is the total parity.

Detailed $\mathbf{S}$ matrix elements can be calculated by integrating the Time Independent (TI or stationary) Schrödinger equation. In the TI approach adopted by us for the present work (by using the atom-diatom quantum reactive scattering program called ABC [19]) the $S^J_{vjKv'j'K'}(E)$ values are calculated using a hyperspherical formalism in which the hyperradius $\rho = \sqrt{R^2 + r^2} = \sqrt{R'^2 + r'^2}$ is taken as reaction coordinate the wavefunctions is expanded locally (at fixed ρ values called sectors) in terms of the Delves basis functions $B^{JM}_{vjK}(\alpha, \theta)$ (α and θ are the Delves hyperangles). The matrix of the coefficients of the expansion $\mathbf{g}(\rho)$ is then propagated from the strong interaction region ($\rho = 0$) to the asymptotes (large ρ values) by integrating the follwing differential equations

$$\frac{\mathrm{d}^2\mathbf{g}}{\mathrm{d}\rho^2} = \mathbf{O}^{-1}\mathbf{U}\mathbf{g} \tag{2.5}$$

in which

$$O_{vjK}^{v'j'K'} = \langle B_{vjK}^{JM} | B_{v'j'K'}^{JM} \rangle \tag{2.6}$$

and **U** the potential and kinetic energy coupling matrix

$$U_{vjK}^{v'j'K'} = \langle B_{vjK}^{JM} | \frac{2\mu}{\hbar^2}(\bar{H} - E) - \frac{1}{4\rho^2} | B_{v'j'K'}^{JM} \rangle \ \ , \tag{2.7}$$

while μ is the reduced mass of the triatom and $\bar{H}$ the assemblage of all the terms of the full Hamiltonian which do not contain derivatives with respect to ρ. The **S** matrix elements are then evaluated by mapping the value of the wavefunction calculated at the asymptotes onto the product states.

3. A first attempt to build a workflow

A first attempt to build a suitable workflow for the central blocks of GEMS was confined to the TI calculations carried out by the quantum mechanical atom-diatom reactive scattering program called ABC [19] (see Ref. [20]) using the P-GRADE Grid Portal [21, 22].

P-GRADE allows to define a parameter study application structure in a graphical environment and, based on this description, it generates the scripts and commands which actually carry out the execution on the distributed grid platform.

Accordingly, the central component of the workflow (labelled as SEQ in Fig. 1) is made of a bash script in which the following steps are performed:

(A) download of the ABC executable (already compiled on the User Interface machine of the EGI Grid) from a COMPCHEM server;

(B) configuration of the enviroment variables;

(C) execution of the ABC executable.

The smaller boxes attached to the various components represent the input and output files which are used and produced by the application. During Grid execution the P-GRADE workflow manager is responsible for preparing these files for the Fortran program and transferring them to the EGI Computing Element. This makes the executable know nothing about the Grid and no modification is required.

The first component of the workflow is the Automatic Generator (see the A_GEN box of Fig. 1). The automatic Generator is a special job type used to generate input text file variations for the related job. Using the parameter definition window of the Automatic Generator, the user is able to define and set the values of the variable used for the bench concurrent simulation.

The second component of the workflow is the already mentioned SEQ.

The third component of the workflow is a Collector job (see the COLL box of Fig. 1) which is again a special job type in P-GRADE. A Collector in a P-GRADE workflow is responsible for

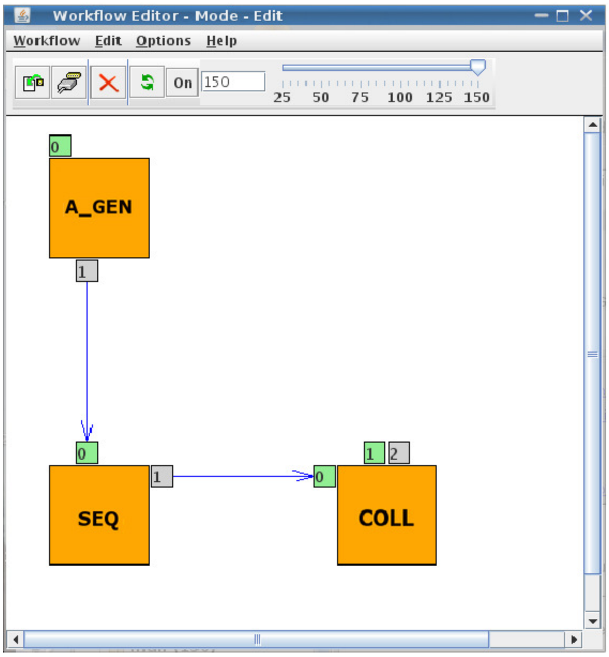

Figure 1: Sketch of the P-GRADE workflow components developed for ABC

collecting the results of the parameter study workflow, analyzing them and creating a typical user friendly filtered result (to be considered as the final result of the simulation). In the present case the Collector simply collects the results from the ABC jobs and compresses the files into a single archive file that can be downloaded by the user through the Portal web interface. The purpose of this step is, in fact, to make the results of the computations accessible by the end users.

Unfortunately the software dependency of the computational codes from specific libraries and compilers, seldom available in the most common Linux distributions over the grid, can not be easily handled with P-GRADE with the result that the execution of such codes turns out to be strictly dependent from the machines where the calculation is performed. The above mentioned limitation and the needs to define event-related dependencies between different applications prompted us to investigate other solutions as it is the case of the GC3Pie high throughput execution framework. In GC3Pie the entire execution schema is assembled at runtime and steps can be added and removed dynamically as the program progresses, adapting to the outcome of individual computations making the programmatic creation of workflows more user friendliness.

4. The presently adopted Grid-based workflow model and actors

In our work, the Python-based GC3Pie framework has been modelled to the needs of the two previously quoted GFIT3C and ABC computational codes in order to build a flexible workflow.

In a this use case, users provide the *ab initio* points of the potential energy surface to be fitted by the GFIT3C application. The fortran routine and the related data produced in this way (fitting file) is then merged into the source code of the ABC application and compiled to produce an ABC binary that will be executed in a parameter study fashion.

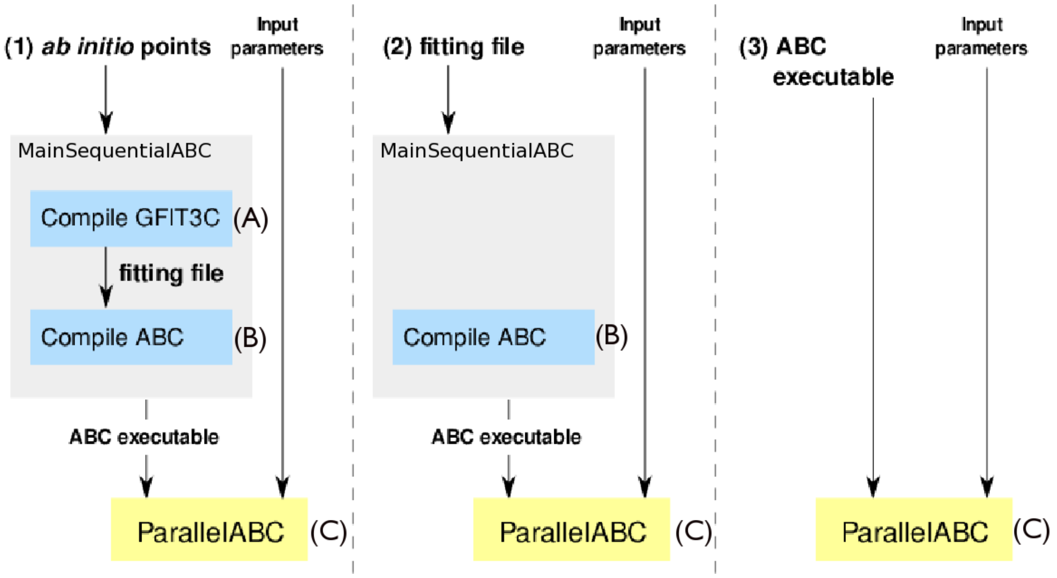

Figure 2: Sketch of the developed workflow showing its components and the connections between them.

Making use of a simple syntax, GC3Pie allows to easily define a large number of applications executing them simultaneously on a large scale distributed computing infrastructure. The steps involved in the workflow are:

(A) The *ab initio* points of the PES are passed as input to GFIT3C which carry out a Fortran file *(fitting file)* containing the values of the PES for the considered system.

(B) The fitting file is then merged into the ABC source code, which is compiled into an executable binary.

(C) The resulting ABC binary is distributed on the grid and executed for computations.

On this basis we considered in our work three different use cases:

(1) The user provides the *ab initio* points of the PES; this executes steps (A), (B), and (C).

(2) The user provides the *fitting file* (produced by an earlier run of GFIT3C); only steps (B) and (C) need to be executed.

(3) The user provides his/her version of the ABC executable and wants to distribute the calculations in a parameter study fashion: only step (C) is executed.

Out of the three different use cases, which a graphical illustration is provided in Figure 2, the first one is the most complex and complete. For its porting, in fact, different factors needed to be taken into account like the dependence from libraries and compilers used.

Due to the fact that the execution time of both steps (A) and (B) is negligible compared with the run time of the produced ABC binary in step (C), the latter is the dominant contribution to the overall execution duration. Depending on the input parameters, in fact, one fixed energy execution of ABC can take more than 10 hours on a single-core desktop computer. Moreover, the ABC code is not run on a single set of parameters only: in step (C) (and, in general, in any parameter-sweep study) the ABC program must be executed several times, consuming a large amount of CPU time. This justifies the use of an "embarassingly parallel" execution scheme in which many instances of ABC run independently and results are collected at the end.

This usage pattern fits well computational Grids. However, some issues arise because of the way the steps of the workflow interact (the standard ABC code is compiled correctly only by the G95 Fortran compiler [23] which is not natively available in any of the most common Linux distributions). Because of that step (B) seldom can be executed on the computational nodes normally available on grid infrastructures. A traditional approach to this problem would require users to negotiate the installation of the dependent software on a proper amount of resources. A simpler solution to this comes from a Virtual Machine (VM)-based approach: workflow steps are run within a VM that provides a controlled and uniform execution environment, including all software dependencies. This is what the AppPot [15] (a self-contained software system comprising a standard Linux VM image and a set of auxiliary programs that can be deployed by just copying a few files) guarantees. Accordingly, AppPot provides a way to run commands inside the VM, possibly in a non-interactive fashion, and for copying files in and out of the VM filesystem. Based on the User Mode Linux (UML) [24, 25] virtualization technology, AppPot provides therefore a way to run a Linux VM inside a Linux host using only code that runs in "user space" without requiring "root" privileges. As a result, Linux-based VM can run as grid jobs and implement generic application deployment on a computational grid.

The workflow engine adopted by us is the GC3Pie [14] high-throughput framework. Based on Python programming language, GC3Pie is a library of Python classes for running large job campaigns on different batch-oriented execution environments, including ARC-based computational grids. GC3Pie provides also facilities for implementing command-line driver scripts, in the form of Python object classes whose behavior can be customized by overriding specified object methods. At the heart of the GC3Pie model is a generic *Application* object, which provides a high-level description of a computational job: list of input/output files, what command to run, resource requirements and limits, etc. GC3Pie translates this information into the job description format needed by the actual execution back-end selected, e.g., xRSL for ARC-based Grids, or a submission script for direct execution on a batch-queuing system. *Application* objects can be adapted to provide behavior customized to a specific use case.

Following the GC3Pie development model, dedicated GC3Pie Applications classes have been developed to describe the execution logic of the workflow. Among them the *ABC_Workflow* class represents the top level logic of the workflow. In fact, it conditionally enables the source compiling stage (controlled by *MainSequentialABC*) or the ABC execution stage (performed by *ParallelABC*) for every input file and parameter combination provided by the user. The input file definition, in fact, defines the sequence of tasks to be executed by the workflow (see Table 1 for details).

If a compiling stage is required, the *MainSequentialABC* creates an instance to the class named *Gfit3C_ABC_uml_Application*. The class is able to build-up a pre defined AppPot VM in any of the execution nodes present in the Grid infrastructure and control steps (A) and (B) by compiling the source code of both GFIT3C and ABC or only ABC (depending on the provided parameters). At

Table 1: Input file definition syntax and attribute helper implemented in the workflow.

Inputs file definitions		
Attribute	File	
- -g3c	g3cfile	G3C input file
- -dim	dimensions	Surface file
- -exec	abc_exec	ABC binary file

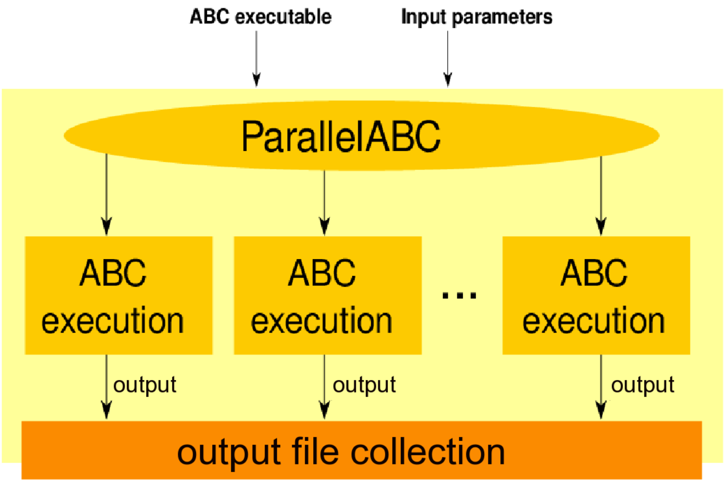

Figure 3: Illustration of the ParallelABC Application class developed to distribute the ABC code into a grid environment.

the end of this stage, the statically compiled ABC executable and the related log files are transferred back to the user account in the client machine for a control check.

Whereas a single Application class is able to control the compiling stage, the management of single and multiple ABC instances are demanded to different classes called *ABC_Application* and *ParallelABC*, respectively. The first class is able to submit and monitor the ABC execution as a single job in a grid environment while the second manages the submission of multiple instances of the *ABC_Application* depending on the number of input files provided by the user (see Figure 3 for graphical illustration of the details).

Also in this case the results carried out by each computation, together with the file logs, are saved on the client machine. Attention has been paid to the need that different output files must be stored in different directories to facilitate checks and further utilization by the final user.

5. Tests, results and performances

The porting of the application and the carrying out of the tests aimed at developing the workflow for the proposed case studies were initially performed on the Swiss National Grid infrastruc-

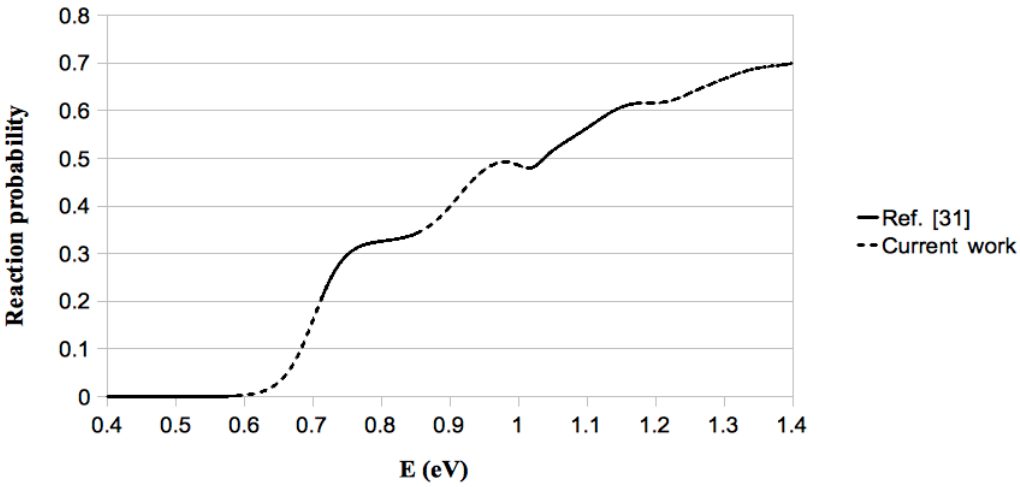

Figure 4: Reactive probabilities as a function of total energy calculated on the present study. The results shown a good agreement with those obtained by Rampino et al. due to the adoption of the same set of *ab initio* points.

ture (SMSCG) [26] on which the GC3Pie framework is available and a test account was created on the User Interface (UI) machine at GC3 supporting ARC [27] middleware.

As a first test of the accuracy of the computational machinery and of the validity of the developed workflow, we executed on the SMSCG infrastructure the complete workflow for the calculation of the TI quantum probabilities of the atom diatom reaction

$$H_A + H_B H_C(v, j) \rightarrow H_A H_B(v', j') + H_C \tag{5.1}$$

to reproduce the data reported by Rampino et al. in Ref. [28]. For this reason we started from pints of Rampino ed al. and the first step of the workflow was devoted to run the GFIT3C routine to fit them. From the fit a r.m.s. of 0.19 kcal/mol was obtained when using a polynomial of degree 7 for the two body terms and a polynomial of degree 10 for the three body one.

In the second step of the workflow the PES Fortran routine obtained from the GFIT3C fit was incorporated into the ABC code. Then dynamical calculations were performed by varying the scattering energy from 0.4 eV to 1.4 eV at zero total angular momentum and diatomic parity +1.

The calculated $v = 0$, $j = 0$ state specific reaction probability is plotted in Fig. 4 (dotted line) as a function of the total energy. Such results agree perfectly well (to emphasize the agreemetn we plot alternatively either values) with those of Ref. [28]. This provides a solid validation of the proposed workflow approach.

Due to the fact that the execution time of both the run of GFIT3C code and the compiling stages are negligible compared with that of the run of the ABC code itself, the latter is the dominant component of the overall execution duration.

On the reference single Intel Xeon machine with 3.4 GHz CPU and 2 GB memory, the run takes from 3 to 5 hours for the adopted set of parameter values. On the contrary when using the Grid, all jobs ends up by spending a considerable time in the job queue before being executed leading to an average execution time about twice as long as that of corresponding dedicated local machine.

Going into more detail, in our validation tests we submitted 1000 jobs in the above mentioned energy interval in order to have a regular grid of results of step 10^{-3} eV.

To avoid the overflow of the queues in the Computing Element machines available in the grid environment where the jobs are submitted, the launch of the jobs were planned so as to have 100 jobs permanently running on the grid environment (function implemented in GC3Pie framework).

Using this approach, after 32 hours we were able to collect the results coming from 712 job submissions for a grand total of about 2300 CPU-hours. The average run time of successful jobs was thus of about 3 CPU-hours.

A simulation of an ideal submission of 100 jobs constantly running with no queue wait time was found to take about 26 hours to complete. According to our simulations, the grid execution model added less than 25% overhead to this realistic use case.

6. An advanced application

Following the success of the validation process of the procedure, we decided to extend its application to the heavier three different atom systems

$$Li + HF(v, j) \to LF(v', j') + H. \tag{6.1}$$

To this end a potential energy surface (PLC) based on high level *ab initio* values [29] was adopted having in mind the rationalization of the outcomes of highly detailed crossed beam experiments and a comparison with our previous theoretical investigations [29–43]. Such PES is slightly endoergic and shows a small barrier to reaction. Yet the endoergic nature of the reactive process turns into an exoergic one when including the corresponding zero point energy, allowing the reaction to occur (in principle) with no help of collision energy [31, 32, 35].

Quantum calculations of the excitation function performed on the PLC PES reasonably well compare with related measurements apart from the low energy portion of the excitation function that, in our case, increase significantly as energy lowers [31, 32].

To better assess the validity of the PES and in particular to check their accuracy in the reaction barrier region (that is typically the PES feature governing the low energy behaviour of the reactive cross section) of great importance is the recent experimental work performed at collision energy values ranging from 82 meV to 376 meV [44, 45]. More recently Loesch and coworkers [46] published data of variable angle crossed beam experiments in which collision energy was lowered down to 25 meV and provided clear evidence for a rising $\sigma(E_{\mathrm{tr}})$ as E_{tr} decreases below 0.1 eV (in qualitative agreement with the already mentioned calculations [31, 32] performed on PLC). However, data published in the mentioned papers do not allow a quantitative comparison with experimental results because insufficiently resolved in the corresponding energy region.

As to the TI computational campaign, by making intensive use of the automatic procedure of the developed workflow, the **S** matrix was calculated for a grid of total energy values ranging from 0.25 eV to 0.45 eV (in steps of 0.0001 eV for energy values up to 0.27 eV and 0.001 eV in the energy interval ranging from 0.27 eV to 0.45 eV). The wavefunction was expanded into diatomic

Table 2: Typical input parameters adopted for the present TI calculations.

Total angular momentum quantum number	0
Triatomic parity eigenvalue	+1
Diatomic parity eigenvalue	0
Maximum internal energy in any channel (eV)	1.25
Maximum rotational quantum number of any channel	200
Helicity truncation parameter	0
Maximum hyperradius (bohr)	24.0
Number of log derivative propagation sectors	150
Initial scattering energy (eV)	0.25
Scattering energy increment (eV)	0.001
Total number of scattering energies	10
Maximum value of v for which output is required	0
Maximum value of j for which output is required	2

basis functions with internal energy smaller than 1.25 eV and propagated through 225 hyperradius sectors up to 24.0 bohr.

Parameters used for the J-shifting approximation in Eq. **??** are $B = \hbar^2/2I_B = \hbar^2/2\mu_{\mathrm{LiF}}r^2 = 3.59 \cdot 10^{-3}$ kcal/mol (the rigid diatomic rotor constant of LiF was computed at the saddle point ($r = 1.62$)) with μ_{LiF} being the LiF reduced mass. Properties of the transition state of the Li + HF system were derived by calculating the minimum energy path at a value of the collision angle corresponding to that of the transition state. Transition state bond lengths were found to be 1.62, 1.31, and 1.73 for the LiF, HF, and LiH pairs, respectively.

The calculated $v = 0,\ j = 0$ probabilities (Ref. [47]) show that there is no collision energy threshold to reaction and that the reactive probability is constantly 1 in a small interval of near vanishing energies (for a range of 0.005 eV). Narrow resonance peaks are found in the low energy region (0.005 eV - 0.075 eV) while a smoother trend characterizes the higher energy region.

The cross values of $\sigma(E_{\mathrm{tr}})$ calculated out of the obtained detailed reactive probabilities are plotted in Fig. 5. Contrary to previous results, they lead to a perfect agreement between theoretical and experimental quantities.

7. Conclusions

The key feature of the present paper is the discussion of the porting and running of some computational chemistry applications on distributed systems by exploiting the innovative features of the workflow developed by the Grid Computing Competence Centre (Zurich), together with the libraries of molecular dynamics codes developed for them in the present work. The use case computational application considered is the implementation of the workflow for the central blocks of the GEMS simulator developed by the Computational Dynamics and Kinetics Group (Perugia) as part of its engagement in the Italian Grid Infrastructure (Bologna) activities. The workflow has been first implemented to reproduce the results of a previous quantum study of the reactive behaviour

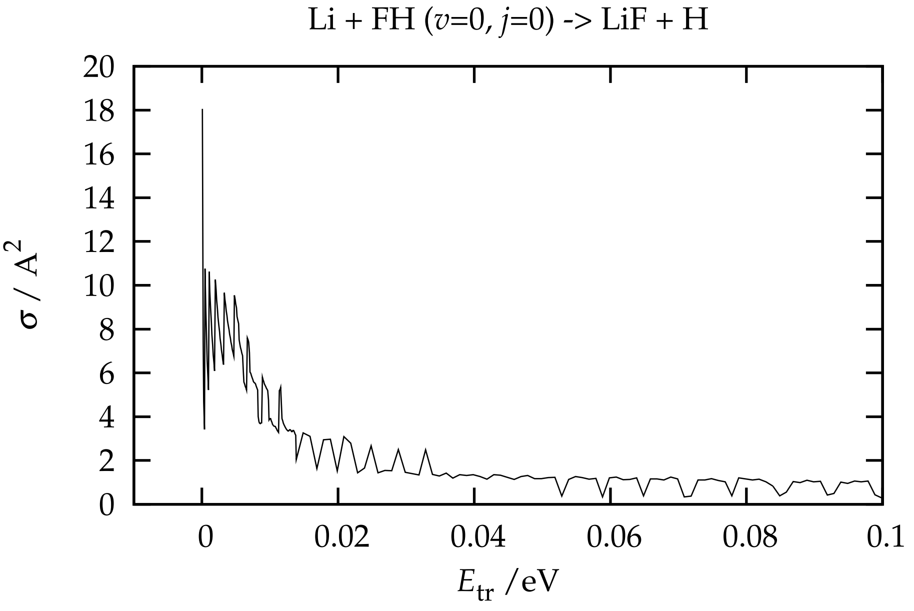

Figure 5: Plot of the *J*-shifting estimate of the cross section as a function of translation energy.

of the H + H_2 system. In this case the used high throughput execution framework GC3Pie and the AppPot cloud/grid virtual machines allowed us to define event-related dependencies between different applications and execute them simultaneously on a large scale distributed computing infrastructure leading to a perfect agreement with previously published results. The implemented case study demonstrates the validity of this approach by following different routes depending on the availability of outcomes of previous investigations. Such a feature of GC3Pie paves the way to further work on better embodying workflows in molecular simulators.

8. Aknowledgements

The authors acknowledge the financial support of the COST CMST Action CM1002 CODECS, through the activities of Working Group 3; the EGI-Inspire contract 261323; the MIUR PRIN 2008 contract 2008KJX4SN_003; the ESA ESTEC contract 21790/08/NL/HE; the Phys4entry FP7/2007-2013 contract 242311. INSTM, Fondazione Cassa di Risparmio of Perugia and ARPA Umbria are also thanked for support. This work was also supported by the AAA/SWITCH funded Swiss Multi Science Computing Grid project (http://www.smscg.ch) with computational infrastructure and resources.

References

[1] A. Laganà, A. Riganelli, and O. Gervasi, “On the Structuring of the Computational Chemistry Virtual Organization COMPCHEM.,” *Lecture Notes in Computer Science*, vol. 3980, pp. 665–674, 2006. http://www.eu-egee.org/compchem.

[2] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005.

[3] O. Gervasi and A. Laganà, “SIMBEX: a Portal for the a priori simulation of crossed beam experiments.,” *Future Generation Computer Systems*, vol. 20, pp. 703–715, 2004.

[4] O. Gervasi, S. Crocchianti, L. Pacifici, D. Skouteris, and A. Laganà, “Towards the Grid design of the Dynamics engine of a molecular simulator.,” *Lecture Series in Computer and Computational Science*, vol. 7, pp. 1425–1428, 2006.

[5] C. Manuali, A. Laganà, and S. Rampino, “GriF: A Grid framework for a Web Service approach to reactive scattering,” *Computer Physics Communications*, vol. 181, pp. 1179–1185, 2010.

[6] “CODECS (COnvergent Distributed Environment for Computational Spectroscopy).” http://www.cost.esf.org/domains_actions/cmst/Actions/CM1002.

[7] “Detailed Chemical Kinetic Models for Cleaner Combustion.” http://www.ensic.inpl-nancy.fr/cost/.

[8] “A worldwide e-Infrastructure for NMR and structural biology.” http://www.wenmr.eu/.

[9] S. Gogouvitis, K. Konstanteli, G. Waldschmidt, S.and Kousiouris, G. Katsaros, A. Menychtas, D. Kyriazis, and T. Varvarigou, “Workflow management for soft real-time interactive applications in virtualized environments,” *Future Generation Computer Systems*, vol. 28, no. 1, pp. 193–209, 2012.

[10] “GC3 (Grid Computing Competence Center.” http://www.gc3.uzh.ch/.

[11] “Computational Dynamics and Kinetics (CDK) research group, University of Perugia (UNIPG).” http://www.chm.unipg.it/gruppi?q=node/48.

[12] “IGI (Italian Grid Infrastructure).” http://www.italiangrid.it/.

[13] “European Grid Initiative.” http://www.egi.eu.

[14] “GC3Pie website.” http://code.google.com/p/gc3pie/.

[15] “AppPot website.” http://code.google.com/p/apppot/.

[16] C. G. Schatz, “Fitting potential energy surfaces,” *Lecture notes in Chemistry*, vol. 75, pp. 15–32, 2000.

[17] N. J. Murrell, S. Carter, S. C. Farantos, P. Huxley, and A. J. C. Varandas, *Molecular Potential Energy Functions*. New York, USA: John Wiley & Sons, 1984.

[18] A. Aguado, C. Tablero, and M. Paniagua, “Global fit of ab initio potential energy surfaces I. Triatomic systems,” *Computer Physics Communications*, vol. 108, no. 2–3, pp. 259 – 266, 1998.

[19] D. Skouteris, J. F. Castillo, and D. E. Manolopulos, “Abc: a quantum reactive scattering program.,” *Comp. Phys. Comm.*, vol. 133, pp. 128–135, 2000.

[20] D. Skouteris, A. Costantini, A. Laganà, G. Sipos, Á. Balasko, and P. Kacsuk, “Implementation of the ABC Quantum Mechanical Reactive Scattering Program on the EGEE Grid Platform,” *Lecture Notes in Computer Science*, vol. 5072, pp. 1108–1120, 2008.

[21] Z. Farkas and P. Kacsuk, “P-GRADE Portal: A generic workflow system to support user communities,” *Future Generation Computer Systems*, vol. 27, pp. 454–465, May 2011.

[22] P. Kacsuk and G. Sipos, “Multi-Grid, Multi-User Workflows in the P-GRADE Portal,” *Journal of Grid Computing*, vol. 3, pp. 221–238, 2005.

[23] “The G95 Project.” http://www.g95.org/.

[24] J. Dike, *User Mode Linux*. Prentice Hall, April 2006.

[25] “UML website.” http://user-mode-linux.sourceforge.net.

[26] "SMSCG website." http://www.smscg.ch.

[27] "ARC middleware website." http://www.nordugrid.org/arc/.

[28] S. Rampino, A. Monari, E. Rossi, S. Evangelisti, and A. Laganà, "A priori modeling of chemical reactions on computational grid platforms: Workflows and data models," *Chemical Physics*, 2011, doi: 10.1016/j.chemphys.2011.04.028.

[29] G. A. Parker, A. Laganà, S. Crocchianti, and R. T. Pack, "A Detailed three dimensional quantum study of the Li + FH reaction," *J. Chem. Phys.*, vol. 102, pp. 1238–1250, 1995.

[30] A. Laganà, A. Ochoa de Aspuru, A. Aguilar, X. Gimenez, and J. M. Lucas, "Threshold effects and reaction barrier in the Li + FH reaction and its isotopic variants," *J. Chem Phys*, vol. 99, pp. 11696–11700, 1995.

[31] V. Piermarini, S. Crocchianti, and A. Laganà *J. Comp. Method in Sciences and Eng.*, vol. 2, pp. 361–367, 2002.

[32] A. Laganà, S. Crocchianti, and V. Piermarini, "Towards a full dimensional exact quantum calculation of the Li + HF reactive cross section," *Lect. Notes Comp. Science*, vol. 3044, pp. 422–431, 2004.

[33] A. Laganà, E. Garcia, and O. Gervasi, "Improved infinite order sudden cross sections for the Li + HF reaction," *J. Chem. Phys.*, vol. 89, pp. 7238–7241, 1988.

[34] M. Baer, E. Garcia, A. Laganà, and O. Gervasi, "An approximate three dimensional quantum mechanical study of the Li + HF $\rightarrow$ LiF + H reaction," *Chem. Phys. Lett.*, vol. 158, pp. 362–368, 1989.

[35] A. Laganà, X. Gimenez, E. Garcia, and O. Gervasi, "Parallel calculations of approximate 3D quantum cross sections: the Li + HF reaction," *Chem. Phys. Lett.*, vol. 176, pp. 280–286, 1991.

[36] M. Baer, H. Loesch, H. J. Werner, and I. Last, "Integral and differential cross sections for the Li + HF $\rightarrow$ LiF + H process. A comparison between j_z-quantum mechanical and experimental results," *Chem. Phys. Lett.*, vol. 219, p. 372, 1994.

[37] M. Baer, I. Last, and H. Loesch, "Three-dimensional quantum mechanical study of the Li + HF $\rightarrow$ LiF + H process: Calculation of integral and differential cross sections," *J. Chem. Phys.*, vol. 101, p. 9648, 1994.

[38] G. G. Balint-Kurti, F. Gogtas, S. Mort, A. Offer, A. Laganà, and O. Gervasi, "A comparison of time-independent and time-dependent quantum reactive scattering Li + HF $\rightarrow$ LiF + H model calculations," *J. Chem. Phys.*, vol. 99, pp. 9567–9584, 1993.

[39] F. Gogtas, G. G. Balint-Kurti, and A. Offer, "Quantum Mechanical Three-Dimensional Wavepacket Study of the Li + HF $\rightarrow$ LiF + H Reaction," *J. Chem. Phys.*, vol. 104, p. 7927, 1996.

[40] G. Parker, R. Pack, and A. Laganà, "Accurate 3D quantum reactive probabilities of Li + FH,," *Chem. Phys. Lett.*, vol. 202, pp. 75–81, 1993.

[41] A. Laganà, G. Parker, and R. Pack, "Li + FH Reactive cross sections from J=0 accurate quantum reactivity," *J. Chem. Phys.*, vol. 99, pp. 2269–2270, 1993.

[42] A. Laganà, A. Bolloni, and S. Crocchianti, "Quantum isotopic effects and reaction mechanisms: the Li + HF reaction," *Phys. Chem. Chem. Phys.*, vol. 2, pp. 535–540, 2000.

[43] A. Laganà, A. Bolloni, S. Crocchianti, and G. Parker, "On the effect of increasing the total angular momentum on Li + HF reactivity," *Chem. Phys. Lett.*, vol. 324, pp. 466–474, 2000.

[44] O. Hobel, R. Bobbenkamp, A. Paladini, A. Russo, and H. Loesch, "," *Chem. Phys. Lett.*, vol. 6, p. 2168, 2004.

[45] O. Hobel, M. Menendez, and H. Loesch, “The translational energy dependence of the integral reaction cross section for Li + HF($\nu = 0$) → LiF + H,” *Phys. Chem. Chem. Phys.*, vol. 3, p. 3633, 2001.

[46] R. Bobbenkamp, H. Loesch, M. Mudrich, and F. Stienkemeier, “The excitation function for Li + HF → LiF + H at collision energies below 80 meV ,” *J. Chem. Phys.*, vol. 135, pp. 204306–204313, 2011.

[47] S. Rampino, L. Pacifici, and A. Laganà, “Li + FH → LiF + H: a reaction born to behave as barrierless,” *J. Chem. Phys.*, vol. (submitted), 2012.