

Cooperative modelling and design on the computing grid: data, flux and knowledge interoperability

Antonio Laganà · Elda Rossi · Stefano Evangelisti

Received: 31 October 2012 / Accepted: 1 April 2013 / Published online: 26 April 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract The fast interconnections of the presently available distributed platforms allow scientists to target highly complex problems by chaining software developed and maintained by experts of the relevant fields. A pillar of such cooperative endeavor in molecular and materials science and technologies is the so-called grid empowered molecular simulator that combines the expertise of molecular science theorists (electronic structure and nuclei dynamics) and experimentalists in order to build and validate ab initio models. This line has prompted an unprecedented level of data format standardization procedures, the bridging of high throughput and high performance platforms, the assemblage of ad hoc designed virtual experiments. In addition this approach has prompted the design and development of tools allowing the evaluation of the quality of the cooperative effort produced by the members of a given research community as well as its rewards to such effort through a credit economy is reported.

Keywords Cooperative · Data formats · D5cost · Grid computing · Grid economy · HPC · HTC · Molecular simulator · Q5cost

Introduction

The exploitation of advanced distributed power of the European Grid Infrastructure (EGI) [1] has allowed the design

and implementation of a prototype Grid empowered molecular simulator (GEMS) [2] based on the combination of collaborative interoperable service oriented computing. GEMS, in fact, builds up, out of the first principles of physics, accurate evaluations of physical observables. To this end GEMS is articulated into the following steps:

- Interaction: a step aimed either at performing ab initio calculations determining (at various levels of accuracy) of the electronic structure of the considered molecular systems within the Born-Oppenheimer approximation and at incorporating, if available, other existing ab initio and/or experimental information to the end of providing either a single point evaluation of the potential energy or an extended point-wise representation (including all the molecular geometries significant to the description of the evolution of the reactive process) of the related potential energy surface (PES);
- Fitting: a step aimed at performing, if required, either a global or a local interpolation of the available ab initio data using a suitable functional form and enforcing (whenever is possible) the reproduction of critical geometries and symmetries of the PES;
- Dynamics: a step aimed at integrating nuclei dynamics equations either on a single potential energy point (provided that is equipped with derivatives) or on the fitted PES;
- Observables: a step aimed at averaging over unobserved parameters and extending to larger scales (non atomistic) to the end of providing a realistic estimate of a measurable physical property.

In practice, however, the implementation of such scheme has issued unprecedented challenges in terms of software interoperability. These challenges and related solutions worked out by the authors for the Chemistry and Molecular & Materials Science and Technologies (CMMST) Virtual Organizations (VO) and Virtual Research Communities (VRC)

A. Laganà (✉)
Dipartimento di Chimica, Università di Perugia, Perugia, Italy
e-mail: lagana05@gmail.com

E. Rossi
CINECA, Casalecchio di Reno, Italy
e-mail: e.rossi@cineca.it

S. Evangelisti
Université Paul Sabatier, Toulouse, France
e-mail: stefano@irsamc.ups-tlse.fr

The first challenge is issued by the gathering in a cooperative fashion of complementary competences and the definition of standards in data formats aimed at enabling interoperability among quantum chemistry (QC) data and between QC and quantum dynamics (QD) ones (see “Quantum chemistry and quantum dynamics data formats”).

The practical adoption of some shared libraries and (at least de facto) standards and libraries is the second issued challenge. Efforts spent in this direction together with the analysis of the related performances are reported in “The Q5cost and D5cost data format standards, libraries and performances”.

The other two issued challenges, namely:

- a) bridging high throughput computing (HTC) and high performance computing (HPC) platforms and cleverly distributing highly coupled or simply parameter sweeping tasks on the most appropriate machine,
- b) defining appropriate metrics for evaluating quality of user (QoU) and quality of service (QoS) and building on them a suitable Grid economy enhancing cooperation are also discussed (see “Bridging HTC and HPC” and “A credit-to-quality based cooperative approach”).

Quantum chemistry and quantum dynamics data formats

As mentioned in the introduction, the first challenge issued by the assemblage of GEMS was the gathering of products of complementary competences and the adoption of standards in data formats aimed at enabling interoperability (like that of QC and QD). For QC and QD data sets, in fact, unlike for those of other chemistry areas, standard formats are unavailable and only quite recently QC researchers became interested in developing data format standards for the ab initio “wave-function based” methods [3]. For these methods the usual theoretical approach is the Born-Oppenheimer (BO) approximation [4] that leads to the solution of the eigenvalue problem:

$$H_e(\mathbf{r}; \mathbf{R})\Phi_e(\mathbf{r}; \mathbf{R}) = V(\mathbf{R})\Phi_e(\mathbf{r}; \mathbf{R}), \quad (1)$$

in which H_e is the electronic Hamiltonian, Φ_e is the electronic wavefunction and V is the potential energy of the system at a specific geometry of the nuclei while \mathbf{R} and \mathbf{r} are the position vectors of the nuclei and the electrons, respectively.

Generally speaking, these codes start from the definition of a molecular species in terms of the nature of the component atoms, the molecular geometry, and the atomic basis set. One- and two-electron integrals are computed. A preliminary Hartree-Fock (HF) or density functional theory (DFT) calculation is usually performed in order to produce molecular orbitals (MO) and to get a first approximation of

the energy. A post-HF treatment can follow to introduce electron correlation (static or dynamic). Eventually, general properties of the target molecule can be evaluated using the computed wavefunction.

Broadly speaking, there are two classes of QC codes:

- a) General codes, which are usually distributed under a license agreement, are stand-alone programs and do not need preliminary data objects. They can start from the definition of the molecular system and compute the wavefunction and different properties at a given level of theory. They are used outside the developers community, have a modular structure and bear well designed user interfaces.
- b) In-house developed codes which implement quite advanced or experimental post-HF treatments (configuration interaction, perturbation theory,...). These codes are in constant evolution and are mainly used by few laboratories (usually for development purpose). As a result, they are scarcely modular, their user interface is often primitive and seldom user-friendly. These type of codes are not stand-alone: they usually rely on general-purpose codes in order to pre compute input quantities (one- and two-electron molecular integrals, orbitals coefficients,...).

Again, broadly speaking, data can be divided into small data sets (SDs) and large data sets (LDs). SDs are used to define uniquely the molecular system, as well as the expectation values for computed properties (total energies, orbital energies, occupation numbers, basis-set overlaps, MOs, etc.). In principle, SDs are better stored in an easy to read and understand format and the amount of information associated with them is quite synthetic. In some cases, QC codes present SDs using extensible markup languages structures (as is the case of MOLPRO [5]). LDs, instead, are often huge sets of detailed data produced by the calculation and worth being stored for further use in the program. The preferred storage method is to use binary unformatted files because this is the most performing and effective way for writing and reading large amounts of data.

Several models are presently available for describing SDs. Our attention has therefore focused mainly on large binary data generated by QC general codes (like DALTON [6], MOLCAS [7], GAMESS-US [8, 9], COLUMBUS [10]) and their usability by QD programs. In GEMS the final user would eventually simply connect to a visual interface, choose the desired type of simulation and enter the needed input data. This is indeed the aim of the Q5cost [11, 12] and D5cost [13] data models and related libraries proposed as a result the COST action D37 activity [14, 15]. The Q5cost and D5cost data models are specifically designed for managing data of the QC and QD domains and for providing the scientific community with a de-facto standard for related computational packages. The prototype cases investigated

within the COMPCHEM activities are the atom diatom systems and the associated reactive processes $A + BC(v, j) \rightarrow AB(v', j') + C$ (in which A is an atom that collides with BC (a molecule in its quantum rovibrational state v, j) yielding a new molecule AB (in its v', j' rovibrational state) and an isolated atom C (or a new molecule AC (v'', j'') and an isolated atom B)) using quantum means. In GEMS the evolution of the atom diatom reaction can be followed using a quantum time dependent (TD) approach (like the program RWAVEPR [16]) that integrates the following differential equation:

$$i\hbar \frac{\partial \Psi(\mathbf{R}, t)}{\partial t} = [\hat{T}_R + V(\mathbf{R})] \Psi(\mathbf{R}, t), \quad (2)$$

where Ψ is the nuclear wavefunction, \hat{T}_R is the kinetic term of the Hamiltonian for the nuclei and t is the time variable. After a sufficiently long integration time, all reaction properties (such as the S matrix elements—whose square moduli are the detailed state to state reaction probabilities—cross sections, rate coefficients) can be worked out, once the calculated wavefunction has been compared (directly or indirectly) with that of the products in the asymptotic region. Alternatively, one can make use of the time independent (TI or stationary) formulation

$$E\Psi(R) = [T_R + V(R)] \Psi(R) \quad (3)$$

by eliminating the time dependence and formulating \mathbf{R} as an appropriate reaction coordinate connecting reactants to products, with E being the total energy of the system, using the ABC program [17].

Data to be passed for this purpose from QC to QD are the information on the potential energy values of all the relevant molecular geometries. This information is produced by the electronic structure ab initio packages and, if calculated on the proper grid and stored in a predetermined format, they can flow directly to the dynamics programs. The key difference between QC and QD schemes lie in the fact that the former refers to a fixed geometry scheme, producing one output result for each different molecular geometry while the latter considers all possible geometries yet only the electronic state belonging to the potential energy surface under investigation. As a consequence QD collects in one comprehensive file the information generated by multiple QC calculations. Further information passed from QC to QD are related to the set of adopted coordinates, the gradients, the Hessians and the non adiabatic coupling matrix.

The Q5cost and D5cost data format standards, libraries and performances

Q5cost and D5cost are data models suitable to represent most of the quantities used and produced by QC and QD

calculations, respectively. However, due to the continuous evolution of related codes and packages data types cannot be completely defined. As a result, data models need to be often modified and the same quantities may end up being represented in different ways. This has motivated the adoption of a general, flexible and easy to upgrade data scheme that may be different for SDs (mainly ASCII coded as is the case of atom labels, geometry, symmetry, basis sets) and LDs (usually binary as is the case of integrals and expansion coefficients).

Accordingly the strategy adopted consists in integrating and organizing data using a containment inspired logical hierarchical scheme. The dominant feature of both QC and QD data is its rapid scale up with the system size. To this end a technology assuring portability on computing platforms differing both in space (good compression algorithms) and time (good I/O performances). Moreover, the use of Fortran programming and of its bindings to other scientific languages, like C and Python, allows an easy management of information. Other important features are: a) a good level of descriptive information, b) an enhanced usability of data. This has led to the adoption (as base technology for data format) of HDF5 (<http://www.hdfgroup.org/HDF5/>) a highly flexible format [12, 18, 19] that allows an easy design of the file in an incremental way.

These hierarchical data models are articulated in containers (group in HDF5 terminology), metadata (attributes) and data (data sets) with the first (root) container for Q5cost containing the definition of the molecular system in terms of its structural data (chemical composition, spatial geometry and basis functions). The Q5cost domains are atomic orbitals (AO), that refers to the data defined on the AO basis, molecular orbitals (MO), that refers to the data defined on the MO basis, wave function (WF), that refers to the definition of the wave function, in terms of linear combination of determinants weighted by the proper coefficients. In the case of D5cost the structure is similar with the system (root) container again storing the information identifying the molecular system (number of atoms, nuclear charge etc.) with the exception of the values of the molecular geometry coordinates because the model allows to store a large number of geometries in a single file.

At present, the following domains have been included in the D5cost data model: interaction, dynamics and observables to which reference has been already made in the previous section. In particular, the interaction domain deals with the nuclear coordinates of a set of different nuclear geometries storing for each of them the Coulomb interaction as well as the potential energy values and related gradient and hessian matrices, the dynamics domain contains the information computed by the QD algorithms (S matrix elements, reaction probabilities) and the observables domain contains the value of relevant cross sections and reaction probabilities.

The Q5cost and D5cost libraries provide read and write access to files defined in accordance with the above mentioned data models and have extension .q5 and .d5 respectively. The two libraries are written in Fortran95 and are based on the HDF5 library. The Q5cost library is articulated into three sets routines: Q5core (a low level module designed to provide a wrapping facility for HDF5 routines), Q5error (routines needed for debugging and monitoring the behavior of the library and of the application code), Q5cost/D5cost (high-level application programming interface providing access to the Q5cost/D5cost format with a high level of abstraction). The D5cost library relies upon the same low level routines (Q5core and Q5error) even if the user level (D5cost itself) has been completely rewritten. The library routine names are chemically meaningful and closely recall their functions.

A case study considered for the application of Q5/D5cost data models and libraries to ab initio quantum investigations of reactive processes using GEMS is that of the hydrogen atom—hydrogen molecule system [20]. For such case study the related ab initio (AI) single geometry calculations were performed using the Dalton package [6] at the NEVPT2 [21, 22] level of theory with a cc-pV5Z basis set for 23 H₂ and 270 H₃ different geometries. Ab initio data were written into a set of q5 files thanks to the use of a specific Dalton/Q5cost wrapper and then collected into a single d5 file, by selecting the lowest potential energy value from each q5 file. Then an analytical PES was generated using the GFIT3C [23, 24] procedure, for which a D5cost/GFIT3C wrapper was used, using a polynomial fit of degree 7 for the long range two body term and a polynomial fit of degree 10 for the three body term (a r.m.s. of 0.19 kcal mol⁻¹ was obtained). Calculations of the S matrix elements were performed using the ABC time independent quantum scattering program [17] on a fine grid of scattering energies by concurrently distributing related jobs on the Grid (the scattering energy was varied in steps of 0.001 eV from 0.4 eV to 1.4 eV and all internal states with an energy ranging up to 2.4 eV and rotational quantum numbers ranging up to 50 were included in the expansion of the wavefunction). All calculations were performed at zero total angular momentum and diatomic parity +1 (only even rotational reactants states are considered). The outcomes of the calculations were then composed in quantities of experimental interest in the OBSERVABLES module, such as the cumulative reaction probabilities. At present significant efforts are being made to extend the quantum dynamics module of GEMS to four atom systems.

The most recent application of GEMS to the study of four atom systems was concerned with the diatom-diatom OH + CO reactive system to the end of reproducing the intensity of the CO₂ product beam (“virtual signal”) measured in a crossed molecular beam apparatus [25] (in this case, though,

dynamical calculations were performed using quasiclassical methods). A direct comparison of the virtual and the real signal has incontrovertibly singled out the deficiencies of the proposed PESs.

The memory space associated with the size of the QC ab initio potential energy values needed to carry out the best fit of the PES amounts usually to the order of giga bytes (GB) and some benchmark tests were carried out in order to verify if the Q5/D5cost library infrastructure imposes appreciable overhead on the I/O operations. All tests were performed on a single node of an IBM Linux cluster 1350 at CINECA (Intel Xeon Pentium IV, 3 GHz 512 Cache, equipped with a GPFS storage disk). The source code was compiled with the Intel FORTRAN compiler and linked with the HDF5 library version 1.8, with all the optional compression facilities being turned off. The test procedure is articulated as follows: (i) The code creates a Q5cost file with the proper internal structure, together with a conventional Fortran file with unformatted sequential access; (ii) a varying number of two-electron integrals are written in the Q5cost file, made of a one-dimensional array of real numbers (integral values) and a four-dimensional array of integer numbers (integral indices) using a chunk with varying size; (iii) the same number of two-electron integrals, with their indices, is then written in the conventional file, using a buffer of the same size of the previous chunk; (iv) finally the times spent for writing the Q5cost file and the conventional file are collected.

The amount of time spent for writing 15,000,064 integrals when using the Q5cost format resulted to be on the average more than 10 % lower than that spent when using the conventional format for the whole range of investigated size of the chunk/buffer (with the saving increasing with it and the total size of the integral file being 346 Mbytes for the former and 343 Mbytes for the latter). The fact that an increase in the chunk size (i.e., a limitation in the I/O access to the file) results in a significant reduction of the overall writing time, confirms the optimized approach of the HDF5 library in the management of large amount of data. It is worth pointing out here that no compression options was adopted for HDF5. When using the HDF5 compression option, the disk occupation reduces by a factor varying between 3 and 10 (depending on the system characteristics). At the same time timings reduce too owing to the fact the overhead associated to compression is more than counterbalanced by the reduction in the I/O volume.

Bridging HTC and HPC

Fundamental to the implementation of the above mentioned cooperative approach of the CMMST community and GEMS in particular, is the ability of redirecting the jobs to

the most appropriate HPC and HTC platforms. This is meant to enable the overcoming of the present highly unsatisfactory situation in which neither HPC nor HTC are completely fit alone to meet the requests of complex CMMST applications. After all, also on the resource providers side (and not only on the user one) there are good reasons for coordinating the use of HPC and HTC e-infrastructures to the end of interoperating large computational applications. This in fact allows an optimization of the usage of both HTC and HPC computing resources because it is not infrequent the case in which a user utilizes HPC platforms not as such but as a bunch of loosely coupled processors underutilizing their fast dedicated network. At the same time HTC users may utilize massively distributed HTC platforms to solve tightly coupled computational tasks ending up by wasting a large amount of time in transferring data on the net. A coordination of the two types of platforms to interoperate via a single workflow (or workflow of workflows) and properly manage the various components on the most appropriate hardware, would instead allow a clever composition of complex applications optimizing the use of the various computing resources and providing the users with the best level of performance.

We have already mentioned the case of a GEMS application to crossed beam virtual experiments. A similar procedure could be equally well applied to spectroscopy, spectrometry and other experiments. Related quantum detailed simulations may require node memories ranging from a few to many GBs to calculate single molecular geometry potential energy values. At the same time the number of geometries needed for a point-wise characterization of the whole PES may well be of the order of 10^N (where N is the number of atoms of the considered molecular system). Due to the fact that the accuracy required by this type of investigations may vary significantly one can cope with the related demand of computing resources either with the nodes offered by the usual grid platform or with those of supercomputers. Similarly, quantum dynamical calculations (both in their TI and TD versions) may end up using the usual grid platform or the supercomputer one depending on the energy of the system and on the structure of its PES.

We make here reference to the just mentioned case of the use of GEMS to calculate a PES in a point-wise fashion that may require (when needed) the use of supercomputer nodes from a highly distributed Grid calculation. This is what the sketch of Fig. 1 (taken from ref. [26]) is concerned with. In the figure a skeleton based on a HTC-HPC scheme (HTPC) that launches a large quantity of independent tasks on a HTC platform (circles) and individually passes the output as input to a HPC one (squares), is sketched. As shown by the figure, in the first upper layer of HTPC (that is of the HTC type) an emitter (upper triangle) generates the large number of independent events represented by circles. The

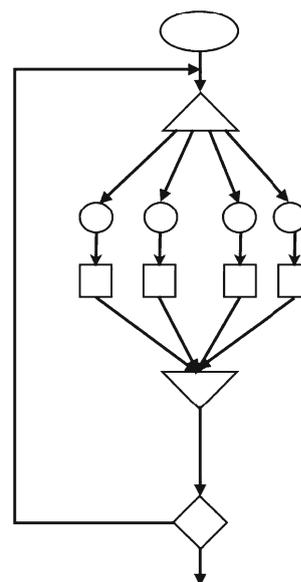


Fig. 1 HTPC skeleton: a HPC computation following a HTC one. An emitter (*upper triangle*) generates a (large) number of independent events (*arrows and circles*) each of which provides the input for a HPC highly coupled calculation (*central arrows and squares*) whose outcomes are returned (*lower arrows*) and gathered together by a collector (*lower (inverted) triangle*). In case the information collected is not yet converged, the sequence is further iterated a certain number of times (*lhs large arrow*)

output generated by elaboration carried out within each circle is then taken as the input for a HPC highly coupled calculation (square). The outcomes of the distributed HPC tasks are returned and gathered together by a collector (inverted lower triangle). In case the information collected is insufficient, the sequence is further iterated a certain number of times.

To implement the above mentioned use-case the following steps have been undertaken: allow access to the involved platforms by the users, integrate the HTC and the HPC workflows, integrate the application into the workflows, design and implement a bridging tool for the combined HTPC platform, develop and implement services on the HTPC platform and optimize the applications for the HTPC platform.

This was performed by implementing the above mentioned steps and de facto standards through two levels of intervention on the overall computing framework having different characteristics:

- a research community or a VO layer in which the case study is prepared, implemented and run.
- an infrastructure layer bridging EGI and CINECA through SSH allowing both access to computing (both HTC and HPC) via G-lite (lightweight middleware for Grid computing) authentication procedures and to storage resources usage managed by different middleware.

A credit-to-quality based cooperative approach

To the end of adopting a robust approach to cooperative work, use has been made of a service oriented architecture (SOA) framework that is well suited to handle today's complex and heterogeneous computing environments. The SOA is, in fact, an ideal approach to a cooperative development and management of software applications using a public network [26] with a service being an implementation of a well-defined functionality that can be consumed by clients in different applications or automatic processes. As a matter of fact, applications in SOA are built on services and SOA helps organize streamline processes by promoting loose coupling among software components to allow an easy and efficient reuse so as to leverage on past investments even if implemented using different technologies. This means that:

- software components have well-defined interfaces (the what) independent from their implementation (the how) and can be consumed by clients (called consumers) transparently with respect to the way the services will execute their requests;
- services are self-contained (perform predetermined tasks), loosely coupled (for independence) and can be dynamically discovered;
- composite services (e.g., workflows) can be assembled by gathering services.

Accordingly SOA-based applications are usually interoperable coarse-grained functionalities which, though differing from Web Services, are often based on them. In this paper we focus on some Grid services designed as a set of secure and collaborative Web services in our laboratory for the COMPCHEM VO [27].

The first of such services is built-in into GriF [28, 29] a Java-based SOA Grid Framework aimed at running on the Grid multi-purpose scientific applications. The basic goal of GriF is to provide the users with a high-level and user friendly tool allowing them to exploit the innovative features of Grid computing with no need for mastering the low-level Grid environment. This means that there is no need for the user to employ specific Grid operating system dependent commands to manage Grid operations (as, for example, running Grid and/or HPC jobs, checking their status and retrieving related results) when building applications of higher level of complexity (e.g., workflows). Accordingly, GriF makes HTC and HPC applications black-box like pushing the Grid computing to a higher level of transparency. This makes GriF a tool of extreme importance for enhancing the VO activities. Its utilization, in fact, leads to higher memory usage, reduced cpu and wall times consumption as well as to an optimized distribution of parallel (e.g., MPI and OpenMP) tasks over the various computing platforms made available.

The second service (also built-in into GriF) is meant to evaluate both the QoS (i.e., the quality achieved in Grid Service production) and QoU (i.e., the quality associated with the virtuosity of the user with respect to the platform and the community). Moreover, the intelligence built-in into GriF is used to offer to the Grid users new perspectives like those associated with selection [30] (as opposite to pure discovery) of the resources as well as service orchestration advanced features enhancing the possibility of establishing collaborative operational modalities in which users and providers collaborate.

This has paved the way to the development of a third service GCrS allowing to reward Grid users for the work done on behalf of the organizations they belong to. This is obtained by awarding QoU and QoS a certain amount of credits that can be redeemed via a preferential utilization of the resources (including financial ones) of a VO. Such development, in addition to leveraging on collaboration, also stimulates a certain extent of competition among the members of a community aimed at producing innovative Grid services and improving the existing ones. GriF is also a means to stimulate the VO members to further step up their membership level in the community and to contribute to the infrastructure development (Hardware Provider) by conferring to the VO some of their computing resources (passive) and taking care of their deployment (active) on the Grid. At later stages of the VO evolution a higher level of membership (stakeholder) is also foreseen for members highly committed to take care of the VO global management.

This defines a Grid economy model based on costs to be paid for the services utilized and credits earned in return for the effort spent on behalf of an organization. The study case considered refers to the COMPCHEM [31] VO and considers in a simplified way costs (as credits to be paid by users for the use of the Grid services offered by the community on the ground of their quantity and QoS) and credits (as credits to be awarded to users on the ground of their quantity and their QoU). It also assumes that Grid services are requested by the VO users (even by specifying only keywords and high-level capabilities rather than memory size, cpu/wall time and storage capacity) via GriF, any set of collaborative Web services of GriF running on the Grid by sharing a common distributed goal is a Grid service and QoUs refer to the collection and filtering of different implicit and explicit information provided by users. For QoS, we refer to the parameters usually characterizing Web services (like accessibility, integrity, performance, reliability, availability and security [32, 33]). Yet, particular emphasis is given also to the innovation parameter that meets one of the central missions of the COMPCHEM VO with the purpose of specifically rewarding active software providers. A functional formulation of the innovation has been tentatively given in terms of the following variables:

- Age (A) as number of days elapsed starting from the publishing date of the Grid service considered;
- Consolidation (C) as number indicating how many previous consolidated Grid services (starting from the input of the former up to the output of the latter) have been wrapped into a new (workflow-enabled) one;
- Diffusion (D) as the ratio between the number of times that the Grid service has been used and the number of VO users in a considered time interval. D implicitly comprises also the friendliness of use;
- Efficiency (E) as the ratio between the number of results produced by the Grid service and the number of VO users in a considered time interval;
- Production (P) as the ratio between a value (ranging between 0 and 1) indicating the level of new functions (e.g., with respect also to already available Grid services recognized as standards by the scientific community of the specific domain of application considered) offered by the Grid service (that is based on both the number and kind of them) and their related developing costs in terms of time and money also expressed by a value ranging between 0 and 1;
- Social (S) as the weighed sum of three values (each of them ranging between 0 and 1) representing, respectively, the level of ethics (e.g., the promotion of universal human values like peace), of fairness (e.g., the promotion of universal availability like open source software) and of social impact (e.g., the promotion of universal welfare) introduced by the Grid service.
- Green (G) as the amount of natural harmlessness including energy saving (in Watt), eco-compatibility of materials (energy saved for their materials recycling) and space saving (in rack unit, or RU) of the Grid computing systems of a hardware provider (the Grid site).

As to QoU, in order to single out research and development activities of new applications, it has been parameterized in terms of the frequency of code modification and compilation. Such activity is typical of the so called active users (AU)s which develop on the Grid platform new applications (or new versions of an existing application) for later massive use by themselves or for collaborative usage [32, 33].

Acknowledgments The activity described in this paper was carried on within the D37 COST Action, Grid Computing in Chemistry [8], the first action in the CMST domain aimed at introducing new information technology tools into the scientific environment, fostering distributed and grid based architectures for computational scientists.

References

1. European Grid Infrastructure (EGI). <http://www.egi.eu/>
2. Costantini A, Gervasi O, Manuali C, Fagnas Lago N, Rampino S, Laganà A (2010) *J Grid Comput* 8(4):571–586
3. Rossi E, Emerson A, Evangelisti S (2002) *J Comput Meth Sci Eng* 2:417
4. McQuarrie DA, Phys Chem (1997) A molecular approach. University Science, Sausalito
5. Werner HJ, Knowles PJ, Knizia G, Manby FR, Schütz M, Celani P, Korona T, Lindh R, Mitrushenkov A, Rauhut G, Shamasundar KR, Adler TB, Amos RD, Bernhardtsson A, Berning A, Cooper DL, Deegan MJO, Dobbyn AJ, Eckert F, Goll E, Hampel C, Hesselmann A, Hetzer G, Hrenar T, Jansen G, Köppl C, Liu Y, Lloyd AW, Mata RA, May AJ, McNicholas SJ, Meyer W, Mura ME, Nicklaß A, O'Neill DP, Palmieri P, Pflüger K, Pitzer R, Reiher M, Shiozaki T, Stoll H, Stone AJ, Tarroni R, Thorsteinsson T, Wang M, Wolf A, MOLPRO. <http://www.molpro.net/>
6. Helgaker T, Aa Jensen HJ, Jørgensen P, Olsen J, Ruud K, Ågren H, Andersen T, Bak KL, Bakken V, Christiansen O, Dahle P, Dalskov EK, Enevoldsen T, Heiberg H, Hetttema H, Jonsson D, Kirpekar S, Kobayashi R, Koch H, Mikkelsen KV, Norman, Packer MJ, Saue T, Taylor PR, Vahtras O (1997) Dalton: An electronic structure program, Release 1.0
7. MOLCAS. <http://www.molcas.org/>
8. Schmidt MW, Baldrige KK, Boatz JA, Elbert ST, Gordon MS, Jensen JH, Koseki S, Matsunaga N, Nguyen KA, Su S, Windus TL, Dupuis M, Montgomery JA (1993) *J Comput Chem* 14:1347–1363
9. Gordon MS, Schmidt MW (2005) GAMESS-US. In: Dykstra CE, Frenking G, Kim KS, Scuseria GE (eds) Theory and applications of computational chemistry: the first forty years. Elsevier, Amsterdam, pp 1167–1189, <http://www.msg.ameslab.gov/gamess/>
10. COLUMBUS. <http://www.univie.ac.at/columbus/>
11. Rossi E, Emerson A, Evangelisti E (2003) *Lect Notes Comput Sci* 316–323
12. Scemama A, Monari A, Angeli C, Borini S, Evangelisti S, Rossi E (2008) *Lect Notes Comput Sci* 5072:1094–1107
13. Rampino S, Monari A, Rossi E, Evangelisti S, Laganà A (2011) *Chem Phys* 398:192–198
14. Rossi E, Bendazzoli G, Evangelisti S, Mihally K, Baldrige K, Ruud K, Renzo C, Valeire V, Jose SM (2011) In: Laganà A, Luethi HP (eds) COST action D37 final scientific report. Dimensione Grafica Editrice, Spello, p 64
15. Laganà A, Balint-Kurti G, Garcia E, Launay JM, Lendvay G, Manthe U, Farantos S, Nyman G (2011) In: Laganà A, Luethi HP (eds) COST action D37 final scientific report. Dimensione Grafica Editrice, Spello, p 27
16. Skouteris D, Pacifici L, Laganà A (2004) *Mol Phys* 102(21–22):2237–2248
17. Skouteris D, Castillo JF, Manolopoulos DE (2000) *Comput Phys Commun* 133:128
18. Borini S, Monari A, Rossi E, Tajti A, Angeli C, Bendazzoli GL, Cimraglia R, Emerson A, Evangelisti S, Maynau D, Sanchez-Marin J, Szalay PG (2007) *J Chem Inf Model* 47:1271
19. Angeli C, Bendazzoli GL, Borini S, Cimraglia R, Emerson A, Evangelisti S, Maynau D, Monari A, Rossi E, Sanchez-Marin J, Szalay PG, Tajti A (2007) *Int J Quantum Chem* 107:2082–2091
20. Rampino S, Monari A, Evangelisti S, Rossi E, Ruud K, Laganà A (2010) A priori modeling of chemical reactions on a grid-based virtual laboratory. In: Proc Cracow 09 Grid Workshop, Krakow, p 164–171
21. Angeli C, Cimraglia R, Evangelisti S, Leininger T, Malrieu JP (2001) *J Chem Phys* 114:1025
22. Angeli C, Pastore M, Cimraglia R (2007) *Theor Chem Accounts* 117:743
23. Aguado A, Tablero C, Paniagua M (1998) *Comput Phys Commun* 108:259
24. Aguado A, Paniagua M (1992) *J Chem Phys* 96:1265

25. Laganà A, Garcia E, Paladini A, Casavecchia P, Balucani N (2012) *Farad Disc Chem Soc* 157:415–436
26. Laganà A, Manuali C, Costantini A, Rossi E, Carpenè M, Ghiselli A, Cecchi M. HIPEG: a project for a high performance grid bridging HTC and HPC in scientific computing. <https://www3.compchem.unipg.it/ojs/index.php/ojs/article/view/8>
27. Manuali C (2011) A grid knowledge management system aimed at virtual research communities sustainability based on quality evaluation. Dissertation, Department of Mathematics and Informatics, University of Perugia (IT). http://www.unipg.it/carlo/PhD_Thesis.pdf, Accessed 27 Jan 2012
28. Manuali C, Rampino S, Laganà A (2010) GriF: a grid framework for a web service approach to reactive scattering. *Comput Phys Commun* 181:1179–1185
29. Manuali C, Laganà A (2011) GRIF: a new collaborative framework for a web service approach to grid empowered calculations. *Futur Gener Comput Syst* 27(3):315–318. doi:[10.1016/j.future.2010.08.006](https://doi.org/10.1016/j.future.2010.08.006)
30. Karta K. An Investigation on Personalized Collaborative Filtering for Web Service (2005) Honours Programme of the School of Computer Science and Software Engineering, University of Western Australia
31. Laganà A, Riganelli A, Gervasi O (2006) On the structuring of the computational chemistry virtual organization COMPChem. *Lect Notes Comput Sci* 3980:665–674
32. Manuali C, Laganà A (2011) A grid credit system empowering virtual research communities sustainability. *Lect Notes Comput Sci* 6784:397–411
33. Mani A, Nagarajan A (2002) Understanding quality of service for web services. Improving the performance of your Web services. <http://www.ibm.com/developerworks/webservices/library/ws-quality.html>. Cited 02 Dec 2011