Evaluating mixed HTC/cloud approaches for parameter sweep applications in systems biology

Ivan Merelli[1], Ettore Mosca[1], Daniele Cesini[2], Elisabetta Ronchieri[2], and Luciano Milanesi[1]

[1] Istituto Tecnologie Biomediche, Consiglio Nazionale Ricerche,
Via F.lli Cervi 93, 20090, Segrate (MI), Italy
[2] Istituto Nazionale di Fisica Nucleare,
Viale Berti Pichat 6/2, 40127 Bologna, Italy

Abstract. In biology, dynamical models are used to understand and predict the behavior of biochemical systems composed of numerous species and reactions. Model parameters represent aspects of the studied system and, therefore, modifications of model parameters correspond to perturbations of the real system. Several methods of model analysis are based on model perturbation by means of its parameter values. Thus, the execution of parameter sweep applications (PSAs) over large parameter spaces becomes very CPU intensive and time consuming.
In this paper, we describe how to perform PSA employing a stochastic simulator of chemical kinetics to explore the behavior of a biological model. We created a distributed computing model that enables the user to create mixed cloud/HTC (High Throughput Computing) workflows by combining the powerful distributed systems and the flexibility of the cloud. As a case study, we investigated the impact of the parameters of a bacterial chemotaxis model by considering four different PSAs in which we reproduced the behavior of the system under various conditions by using our stochastic simulator. We accomplished this study by evaluating some existing cloud/HTC approaches used to address biology problems and reliable to deal with PSAs of systems biology models and to handle the simulations required in sensitivity analysis and parameter estimation.

Keywords: HTC/cloud, parameter sweep applications, dynamical systems

1 Introduction

Recent experimental investigations at the single-cell level [1] have highlighted the presence of noise in the cellular processes. Hence, standard modeling approaches (for example based on ordinary differential equations) inadequately capture the effects of biological random processes, as those that bring the system to different states starting from the same initial conditions (such as lysis or lysogeny in phage-infected bacteria [2]). On the other hand, many algorithms that perform stochastic simulations of biochemical reaction systems have proved their intrinsic suitability for reproducing the dynamics of many cellular processes [3].

Therefore mechanistic mathematical models are precious for representing real biological systems and resembling the actual observations; they are usually composed of large numbers of components, which interact through many biochemical processes. To analyze these systems, many parameters of each model need to be considered to determine crucial factors by executing several computer simulations and gathering statistical properties of the dynamics.

There are several techniques devoted to the analysis of model dynamics. More precisely, steady state analysis concerns the identification of points in the space of reachable states where some properties of the system remain unchanged over time (e.g., where the behavior of the system is constant over time); bifurcation analysis studies the qualitative variation of the steady states (e.g., transition from oscillating to non oscillating regimens) as a consequence of the variation of the parameters; sensitivity analysis relates the uncertainty of the input of a model (i.e., variations in parameters and initial conditions) to its output (namely, the resulting behavior); and parameter sweep application (PSA) explores the parameters space of a system by means of independent experiments. PSA represents one of the straightforward methods, and a distributed architecture is suitable for its execution.

In this paper, we present a comparison in the implementation of a PSA, which consists in performing a large numbers of stochastic simulations by using the Sτ-DPP [4] simulator - a stochastic simulator based on the τ-leaping method [5]. As a case study, we executed a number of PSAs to explore the parameters space of a bacterial chemotaxis model, which describes cells that move according to chemical gradients. We used this particular system regarding the relatively large number of chemical reactions and molecular species involved, and the average time required to perform a single simulation of its dynamics makes it an appropriate test case. We analyze technical solutions available to the scientific community that can be exploited to realize a mixed HTC (High Throughput Computing)/Cloud approach for our application. Finally we compare the performance obtained with a pure grid computing model to the one obtained using a preliminary implementation of the mixed model. The results provide a useful starting point for future implementations of other, more complex, analysis methods for stochastic biological models, which involve a large number of simulations; some examples are evolutionary and particle swarm algorithms for parameter estimation, Morris and Sobol's sensitivity analysis methods [6], [7].

The paper is organized as follows: Section 2 recalls some basic notions of stochastic modeling and simulation of biochemical systems; Section 3 introduces the PSA in the context of stochastic simulations of biochemical pathways; Section 4 describes considered HTC/cloud approaches; Section 5 discusses the platforms used for performing the PSA and the strategy adopted for distributing the computation; Section 6 presents the case study and the approach to analyze the results; Section 7 discusses the performance on the different platforms; Section 8 talks about future activities; Section 9 draws some conclusions about the tests performed.

2 Stochastic modeling and simulation of biochemical systems

A mathematical model of a biological system is usually developed over two steps: the identification of the set of molecular species involved and the set of reactions, which describe the interactions among species; the selection of a proper set of parameters (such as molecular quantities and reaction rates) to fit the system behavior with experimental data.

Since noise and discreteness play an important role in cellular processes, stochastic modeling is a suitable method for the study of these systems. The Stochastic Simulation Algorithm (SSA) [8] is the most used algorithms for the description of the dynamics of a biochemical system. SSA deals with well-mixed single volume systems, the conditions of which (such as pressure and temperature) are kept constant during the evolution. This procedure provides the exact behavior of a system - i.e., it is proved to be equivalent to the Chemical Master Equation [8]. The stochastic formulation of a biochemical system is given by means of a set of chemical species along with their quantities to denote the current state, a set of chemical reactions to specify how the chemical species interact and evolve, and a set of stochastic constants associated with the reactions. Each constant summaries the chemical and physical properties of the corresponding reaction; it is used in the computation of the reaction application probability along with a combinatorial function.

The main drawback of SSA is the computational time needed to perform a simulation. The algorithm time complexity is proportional to the number of species and the number of reactions; hence, many real problems can be inadequately solved by using this technique. To speed up stochastic simulations, a procedure called τ-leaping has been introduced [5]. Here, instead of describing the dynamics of the system by tracing the occurrence of every single reaction, a time increment is computed and many reactions are selected (according to a specified distribution) and executed in parallel. Consequently, the obtained behavior of the chemical system is inaccurate in contrast with SSA. In this paper, we refer to the τ-leaping version presented by Gillespie in 2006 [9], a more efficient multi-volume stochastic algorithm that stands at the basis of the Sτ-DPP to simulate biochemical systems [4].

3 Parameter sweep application for the simulations of stochastic models

A PSA executes an application multiple times with a unique set of input parameters per computation. This type of application contains several independent jobs operating on different data sets to explore a wide range of scenarios and parameters. It is executed by processing N independent instances on M parallel or distributed computational resources, where N is typically much larger than M. However, this high-throughput parametric computing model is simple, yet powerful enough to formulate distributed applications ranging in many different areas.

In the framework of stochastic simulations, the application executed during the PSA is a simulation performed by using the Sτ-DPP algorithm. We can obtain the parameterizations $P = (p_1, ..., p_s)$ by varying the species, the reactions, the values of the initial species amounts, the values of the constants defined in the model, and the set of the stochastic simulator parameters (such as the length of the simulation interval and the frequency according to which the system's state has to be saved) that directly affects its performance (e.g., the time and memory space needed for its execution). The definition of the set of PSA parameterizations depends on the specific applications and on the data type of the involved parameters. Usually, the parameterizations use the Cartesian product of the parameters and sample values from the space defined by their ranges of variation. When the number of parameters is high, their values can be sampled by using quasi-random series [10] - also called low discrepancy series - that aim at uniformly cover the space with few samples (i.e., with a lower number of points compared to classic uniform distributions). The output of each PSA is composed of the set of results generated by all the executions of the selected application, each one with a different parameterization. In this work, we considered the output of a PSA as the set of stochastic simulation results $D = (d_1, ..., d_s)$, where $d_i = (x_0^i(0), ..., x^i(t))$ with $i = 1, ..., s$ is the single output of a simulation obtained by using the Sτ-DPP algorithm.

In the field of systems biology, this kind of application raises issues related to the required computational resources, since the algorithms are computationally expensive in the case of stochastic models, and due to stochasticity, more than one stochastic simulation is usually needed to characterize the systems dynamics. A possible solution to cope with this computationally intensive problem is to exploit a distributed approach such as grid computing. Grid is an ideal platform for handling high throughput applications, which are characterized by independent and sequential jobs that can be individually scheduled on many different computing resources. Therefore grid computing suitably compute large number of independent simulations that the PSAs of biochemical models require. The advantage of using a grid approach for large computational challenges relies on the high-end scalability of this technology. If grid jobs are completely independent, as in this case, then the theoretical scalability of the system is linear. This is untrue for real computations because of the time needed for scheduling jobs, for transferring data and for resubmitting failed jobs [11], [12], [13].

4 Available HTC/cloud solutions

To combine the power of the grid infrastructure and the flexibility of the cloud computing, new systems that can instantiate VMs on top of the infrastructure resources are under development. In the field of grid and cloud integration we can observe various approaches performed at the infrastructure level. One is the grid of federated cloud that develops and manages multiple cloud stacks to address communities needs. In Europe this approach is followed by various research projects, such as the CERN openlab [14] and the European Grid Infrastructure (EGI) [15] projects: the former is determined in developing a federated hybrid cloud built on OpenStack provider; the latter through the federation cloud working group [16] aims at creating a federation of various cloud providers that run different software stacks, which need to interoperate publishing resource discovery information and accounting/monitoring data to the central EGI services. Another approach is the grid over cloud that requires the creation of grid sites on top of a cloud infrastructure through landscape deployment of virtual resources [17]. Projects that adopt this solution are, for example, StratusLab [18] that provides grid services by using the StratusLab IaaS system as well as the CERN openlab that exploits the OpenStack solutions to run grid services. The cloud over grid is another interesting solution that allows existing grid infrastructures to access cloud services minimizing the changes to be applied at site level. The Worker Node on Demand Service (WNoDeS) [19] and the CLEVER middleware [20] are examples of framework that use this approach. The WNoDeS framework realizes a cloud-over-grid paradigm able to instantiate Virtual Machines (VMs) from pre-defined images through a grid interface and through a pure cloud interface accessible via command line or via a dedicated portal. CLEVER provides a simplified access to private/hybrid clouds.

There are other approaches that provide the user with an abstraction to grid and cloud solutions by using the same API, such as the SAGA BigJob framework [21]. It is based on a general implementation of pilot jobs to be usable over several heterogeneous distributed infrastructures. Pilot jobs have been used to utilize resources effectively, to reduce the net wait time of a collection of tasks, to facilitate bulk or high-throughput simulations, and to implement application-specific scheduling decisions and policy decisions. BigJob address a wide range of scientific applications for different communities (also connected with biology) by using cloud and traditional grid/cluster resources [22].

5 Implementations for PSA over distributed platforms

Performing a distributed computation requires identifying a suitable strategy for creating jobs to define the granularity of the computation. As a matter of fact, the computation of long jobs on the grid may cause significant data loss in case of system failure or data transfer problems. On the other hand, the execution of a large number of short jobs raises the total latency time in the batch queues, affecting the global performance of the system. Middle and long jobs are the most suitable to exploit grid computing, because they represent a good trade-off between grid latency and failure problems [23]). To characterize the best granularity for the particular application presented in this paper, we performed different PSAs by varying the number of jobs and the number of simulations per job, and by altering the computation time with different strategies of parameter selection and - as a direct consequence - the size of the output files. Considering the definition of PSA given in Section 3, the set of calculated dynamics $D = (d_1, ..., d_s)$ constitutes the output.

In this case, users must use Storage Elements (SEs)-the entry point for distributed disks or tapes-to archive the numerical results before downloading them to the user interface. This approach is essential when the complete numerical outcomes need to be retrieved for further investigations of the system's dynamics. However, preliminary analysis of grid performance suggested that the output data size has a significant effect on both the computation overhead and the success rate of computations [13], [24].

Moreover, in the grid platform many errors arise from the misconfiguration of the environment in which the jobs land. Misconfigurations can affect both the grid middleware and the user software or libraries installed on the site, with the same effect of aborting the jobs which force a job resubmission and increase the total execution time to complete the production [13].

These kind of problems can be easily faced changing the computing paradigm to a cloud based approach that allows the users to run the applications on their own images. VMs instantiated trough a cloud interface can survive to the jobs end, allowing for an easier implementation of the data management of the jobs output and reducing the error causes.

5.1 Testbed description

In this work, EGI as well as the IGI (Italian Grid Infrastructure) [25] were used as distributed infrastructures. IGI is composed by 55 distributed sites spread all over the country and offers a total of about 32000 CPU cores and 40PB of storage capacity split between disk and tape. During 2012, more than 1100 users have used the IGI resources for a total of 260 millions CPU hours. The EGI and IGI resources are provided to Virtual Organizations (VOs): the shares for each VO depend on agreements between the VO and the site managers. Most of the resources are however shared by multiple VOs allowing an efficient use of the sites, but creating concurrency in the resources exploitation, in particular the CPU cycles. In this study, two VOs were used: BIOMED (an international VO) and GRIDIT (an Italian national VO). The access to the resources was done in an opportunistic and concurrent way, i.e. with unreserved shares. The sites we exploited were all equipped with the gLite middleware [26]. Furthermore, we used the WNoDeS framework as one of the cloud over grid approaches considering the experiences done at INFN with various communities [27]: astro-particle physics with the Auger experiment to access their scientific applications at INFN Tier1; life science with WeNMR virtual research community; and biology with macro molecular applications [28].

For the PSA use case, we developed a test to compare the pure grid performance with those obtainable with mixed HTC/cloud approach. In the first case, the EGI grid infrastructure has been used with few requirements on the sites that were selected by the grid systems, while in the second case, we used a single IGI site where we were able to instantiate virtual machines on top of the grid resources. The scale of the test is small if compared to the one of a real-life

PSA production because of the limited number of cloud enabled sites in IGI, but it gives anyway a good indication that the mixed HTC/cloud approach can be successfully exploited to achieve better performance in terms of abort ratio in a cost effective manner.

6 Bacterial chemotaxis: a case study

According to our purpose of testing the HTC/cloud infrastructure for the analysis of biochemical systems, we considered a model describing the bacterial chemotaxis [13]. As a matter of fact, the relatively large number of chemical reactions and molecular species, and the average time required to perform a single stochastic simulation are all factors that make the bacterial chemotaxis model a suitable test case to prove the effectiveness of the grid infrastructure. More precisely, the model we considered is composed of 40 species participating in 59 reactions. The model dynamics correctly reproduces the behavior of the real system: for example, the quantity of the CheY protein (the system's response regulator) correctly settles back to a pre-stimulus level after treatments with different ligand concentrations, and shows an appropriate response according to the state of the receptor.

Table 1. Computational requirements for the PSA_i of the test

Req.	PSA_1	PSA_2	PSA_3	PSA_4
$JobNumber$	60	60	100	100
$\frac{JobDuration}{TotalPSADuration}$	$\frac{45}{45}\frac{mins}{hours}$	$\frac{90}{90}\frac{mins}{hours}$	$\frac{230}{383}\frac{mins}{hours}$	$\frac{30}{50}\frac{mins}{hours}$
$\frac{SingleJobOutputSize}{TotalOutputSize}$	$\frac{70}{4}\frac{MB}{GB}$	$\frac{25}{1.5}\frac{MB}{GB}$	$\frac{188}{19}\frac{MB}{GB}$	$\frac{12}{1.2}\frac{MB}{GB}$

We performed four $PSAs$ (in the following called PSA_1, PSA_2, PSA_3 and PSA_4) by creating a number of parameterizations in which we modified the stochastic constant values of the bacterial chemotaxis model to explore the n-dimensional space of values in the proximity of a reference point. In doing so, the values of the n constants let the model correctly reproduce the behavior of the real system.

With the aim of experimenting the grid using different settings, we modified the number of jobs and the simulations per job across the four $PSAs$. Therefore, the PSA differ from each other both for the number of simulations per job and the number of grid jobs. We selected these settings to obtain an expected CPU time between 0.5 and 3.5 hours, which is the typical time of middle jobs for the grid infrastructure. The expected CPU time was estimated by computing a single job on an Intel Xeon 2.5GHz, 10GB RAM.

For what concerns the output produced, it is in the order of 4GB for PSA_1, 1.5GB for PSA_2, 19GB for PSA_3 and 1.2GB for PSA_4. Table 1 summaries the computational requirements for the 4 PSA_i.

7 Performance Discussion

For our test the pure grid implementation has a success rate of 78%, which leads to roughly 75% of results correctly retrieved (close to the EGI standards [29]). The expected computational time was about 24 days on a single CPU while the grid computation took only 30 hours, which corresponds to Crunching Factor of 19. This value is lower than the peak number of processors used concurrently, which reached 81 CPUs, spread over 12 sites, and this is due to errors and resubmissions of aborted jobs. In this implementation the distribution efficiency - defined as the ratio between the overall crunching factor and the maximum number of concurrently running CPUs - is around 20%, which is close to the value reported by Lee [11]. To reduce the size of data transfer in the pure grid implementation the function f for the analysis of the system dynamics is computed directly on the grid resources, avoiding the raw output download. So doing, the output was reduced to less than 1 MB for each job (320 MB for all the PSA_i, i=1..4) compared to the 25GB of the raw output size.

In the mixed HTC/cloud implementation, we used a single site located at the INFN-CNAF [30] computing center. We prepared two VM images: the $type_1$ containing the software to execute the simulation and the $type_2$ to act as output data repository and to perform the function f calculation. Two data repositories ($type_2$ images) were instantiated using the WNoDeS pure cloud interface: one repository to host the data for the PSA_1 and PSA_2 and the other to host data for PSA_3 and PSA_4. This splitting was done to avoid overload of the repositories. The two repositories were up during the entire simulation execution allowing an easy retrieval of the whole raw output data set (in the order of 30GB) at the end of the computation with standard Linux commands. The job submissions were performed using standard grid submission tools. The jobs, once landed into the site, instantiated the VM with images of $type_1$ and performed their computation. Input parameters were passed to each jobs using grid tools as done in the pure grid implementation. Transfers to the data repository of raw output were executed with Linux commands. A maximum of 21 concurrent CPUs were used to perform the computation that was completed in 35 hours, leading to a crunching factor of 16, which is lower than the value obtained using the pure grid approach, but the distribution efficiency was close to 76%, 4 times higher than the other implementation. The total number of jobs needed to complete the production was 232 leading to a 94% of total job efficiency: 10 of the 12 aborted jobs encountered grid errors before reaching the site, while 2 were aborted during the execution for causes that are not yet understood.

In summary, we can conclude that for our 4-PSA test the duration of the computation is comparable between the two implementations but the mixed HTC/cloud approach allows a strong reduction of errors and resubmission, in particular, in the data management phase, thus increasing the overall production efficiency. However the size of our test was very limited if compared to a real life use case, that can easily reach 10 times the size of the problem discussed in this paper; this was due to the availability of virtualization systems on IGI resources that forced us to use just one site that cannot be sufficient to maintain reasonable execution times for scaled up use cases. To use more sites we need to more EGI/IGI sites to offer both cloud and grid accesses to their resources, however we showed that mixed HTC/cloud computing models can drastically increase the efficiency of the computation in PSA applications that, being embarrassingly parallel, are suitable to be run on an HTC environment.

8 Future Works

Future works heavily depends on the evolution and availability of the tools that are able to provide cloud access to the resources currently used by our virtual organization in grid sites. This is out of our control and depends on the infrastructure providers. Depending on the cloud systems that will be available, a couple of aspects of our computing model can be improved and further investigated: the interaction between the grid worker nodes and the data repositories (VM instantiated with image of $type_2$) - this interaction could be performed easily exploiting services provided by the cloud infrastructure if available, i.e. a virtual disk space that can be mounted by both types of instances; the marketplace and related policies to store the VM images needed by the computation. With one site this problem was faced using a single local marketplace obtained from the agreement with just one resource provider, but if more actors come into play this aspect can become a difficult task having to deal with different site policies.

Another possible evolution for this computing model is to integrate the resources offered by the so-called EGI federated cloud infrastructure [16], which is the way EGI is trying to offer cloud services with a grid of federated clouds approach. In the near future this option will be offered as a production services to the same virtual organizations supported in the grid environment. Moreover, the resources of the EGI federated cloud are, at least at moment of writing, completely disjoint with the grid sites in most of the sites, making it difficult to combine the HTC and cloud paradigms.

9 Conclusions

Since the large number of simulations often required in the field of stochastic modeling of biochemical systems demands a significant computational work, we carried out a study to compare two different computing models on distributed platforms and verify if they can be a useful solution for such task. To obtain a meaningful test we decided to take into account a stochastic model of bacterial chemotaxis, characterized by a size (number of molecular species and biochemical processes) and a computational cost in agreement with other models of the same kind in the field of systems biology. We organized the test as a set of PSAs, in which we simulated the model modifying both the parameters of the model and of the stochastic simulator.

Concerning the EGI performances, we found that some formerly known problems of the EGI (e.g. high failure rate, high latency time on the cluster queues, and last job syndrome) can prevent from obtaining optimal performances when these issues are quantified using realistic use cases rather than employing dummy jobs. Moreover, the granularity appeared as an element to carefully consider. In fact, to obtain the optimal granularity during a PSA we should take into account that the expected computational cost (time and space) of a simulation depends on the parameterization and, therefore, the number of simulations to include in a single job to obtain the desired computation time varies according to the parameterizations.

To improve efficiency of the computation we are trying to move towards a cloud based computing model, but to exploit the great amount of resources still available in a cost effective way to scientific communities in the European distributed infrastructures we are evaluating systems that are able to instantiate Virtual Machines on top of the grid resources. We built a mixed HTC/cloud computing model and run a 4-PSA test case to evaluate the new computing model performance. We managed to obtain a distribution efficiency 4 times greater than the pure grid computing model maintaining a similar total execution time. This is possible given the reduced number of errors and consequent job resubmissions. We could not scale up the simulation to test a more realistic problem dimension due to the limited number of sites providing both grid and cloud access to their resources, however our study encourages the use of mixed/HTC solutions to distribute large scale simulations and analysis of stochastic models concerning biochemical systems.

Acknowledgments

This work has been supported by the Italian Ministry of Education and Research through the Flagship (PB05) InterOmics , HIRMA (RBAP11YS7K) projects and the European MIMOmics (305280) project. The research leading to the results presented in this paper has been possible thanks to the grid resources and services provided by the European grid Infrastructure (EGI) and the Italian grid Infrastructure (IGI).

References

1. Elowitz, M., Levine, A., Siggia, E., Swain, P.: Stochastic gene expression in a single cell. Science **297**(5584) (2002) 1183 1186
2. Arkin, A., Ross, J., McAdams, H.: Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells. Genetics **149**(4) (1998) 1633 1648
3. Turner, T., Schnell, S., Burrage, K.: Stochastic approaches for modelling in vivo reactions. Computational Biology and Chemistry **28**(3) (July 2004) 165 178

4. Mosca, E., Cazzaniga, P., Pescini, D., Mauri, G., Milanesi, L.: Modelling spatial heterogeneity and macromolecular crowding with membrane systems. In Gheorghe, M., Hinze, T., Paun, G., Rozenberg, G., Salomaa, A., eds.: Membrane Computing. Volume 6501 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 285 304
5. Gillespie, D.: Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. **1** (2001) 1716 1733
6. Morris, M.: Factorial sampling plans for preliminary computational experiments. Technometrics **33**(2) (1991) 161 174
7. Sobol', I.: Sensitivity analysis for nonlinear mathematical models. Math Mod Comput Exp **1** (1993) 407 414
8. Gillespie, D.: Exact stochastic simulation of coupled chemical reactions. J Phys Chem **81** (1977) 2340 2361
9. Cao, Y., Gillespie, D.T., Petzold, L.R.: Efficient step size selection for the tau-leaping simulation method. J Chem Phys **124**(4) (2006)
10. Niederreiter, H.: Random number generation and quasi-monte carlo methods. In: Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1992)
11. Lee, H.C., Salzemann, J., Jacq, N., Chen, H.Y., Li-Yung Ho, I.M., Milanesi, L., Breton, V., Lin, S., Wu, Y.T.: Grid-enabled high-throughput in silico screening against influenza a neuraminidase. IEEE Transaction Nanobioscience **5**(4) (2006) 288 295
12. Merelli, I., Morra, G., Milanesi, L.: Evaluation of a grid based molecular dynamics approach for polypeptide simulations. IEEE Transaction Nanobioscience **6**(3) (2007) 229 234
13. Mosca, E., Cazzaniga, P., Merelli, I., Pescini, D., Mauri, G., Milanesi, L.: Stochastic simulations on a grid framework for parameter sweep applications in biological models. In: International Workshop on High Performance Computational Systems Biology, IEEE Computer Society (2009) 33 42
14. Murray, A.C.: Cern, rackspace collaborate on hybrid cloud (July 2013)
15. EGI: Egi site. http://www.egi.eu
16. EGI: Egi federated clouds task force. https://wiki.egi.eu/wiki/Fedcloud-tf:FederatedcloudsTaskForce
17. Barone, G.B., Bifulco, R., Boccia, V., Bottalico, D., Canonico, R., Carracciuolo, L.: Gaas: Customized grids in the clouds. In Caragiannis, I., Alexander, M., Badia, R.M., Cannataro, M., Costan, A., Danelutto, M., Desprez, F., Krammer, B., Sahuquillo, J., Scott, S.L., Weidendorfer, J., eds.: Euro-Par 2012: Parallel Processing Workshops. Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 577 586
18. Konstantinou, I., Floros, E., Koziris, N.: Public vs private cloud usage costs: the stratuslab case. In: the 2nd International Workshop on Cloud Computing Platforms (CloudCP). Volume 10. (2012) 260 282
19. Ronchieri, E., Donvito, G., Veronesi, P., Salomoni, D., Italiano, A., Torre, G.D., Andreotti, D., Paoloni, A.: Resource provisioning through cloud and grid interfaces by means of the standard cream ce and the wnodes cloud solution. Proceedings of Science **PoS(EGICF12-EMITC2)124** (2012)
20. Tusa, F., Paone, M., Villari, M., Puliafito, A.: Clever: A cloud cross-computing platform leveraging grid resources. In: IEEE Internatonal Conference on Utility and Cloud Computing. (2011) 390 396
21. Luckow, A., Lacinski, L., Jha., S.: Saga bigjob: An extensible and interoperable pilot-job abstraction for distributed applications and systems. In: The 10th

IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. (2010)

22. Jha, S., Katz, D.S., Luckow, A., Merzky, A., Stamou, K.: 13. In: UNDERSTANDING SCIENTIFIC APPLICATIONS FOR CLOUD ENVIRONMENTS. Cloud Computing: Principles and Paradigms (2011) 345 371
23. C, G.R., RTA, O., C, L.: Grid scheduling for interactive analysis. Stud Health Technol Inform **120** (2003) 25 33
24. Merelli, I., Pescini, D., Mosca, E., Cazzaniga, P., Maj, C., Mauri, G., Milanesi, L.: Grid computing for sensitivity analysis of stochastic biological models. In Malyshkin, V., ed.: Parallel Computing Technologies. Volume 6873 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 62 73
25. IGI: Welcome to igi, italian grid infrastructure. https://www.italiangrid.it
26. Ferrari, T., Gaido, L.: Resources and services of the egee production infrastructure. Journal of Grid Computing **9** (2011) 119 133
27. Ronchieri, E., Verlato, M., Salomoni, D., Torre, G.D., Italiano, A., Ciaschini, V., Andreotti, D., Pra, S.D., Touw, W.G., Vriend, G., Vuister, G.W.: Accessing scientific applications through the wnodes cloud virtualization framework. Proceedings of Science **PoS(ISGC 2013)029** (2013)
28. Ronchieri, E., Cesini, D., DAgostinoy, D., Ciaschini, V., Torre, G.D., Cozziz, P., Salomoni, D., Clematisy, A., Milanesiz, L., Merelli, I.: The wnodes cloud virtualization framework: a macromolecular surface analysis application case study elisabetta. In: Parallel, Distributed, and Network-Based Processing (PDP), will be published (12 14 February 2014)
29. Jacq, N., Salzemann, J., Jacq, F., Legr?, Y., Medernach, E., Montagnat, J., Maa?, A., Reichstadt, M., Schwichtenberg, H., Sridhar, M., Kasam, V., Zimmermann, M., Hofmann, M., Breton, V.: Grid-enabled virtual screening against malaria. Journal of Grid Computing **6**(1) (March 2008) 29 43
30. INFN: Infn cnaf. https://www.cnaf.infn.it/en