Experiences on a Large Scale Grid Deployment with a Computationally Intensive Biomedical Application *

J.M. Alonso, V. Hernández, G. Moltó

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera S/N, 46022 Valencia, Spain
{jmalonso,vhernand,gmolto}@dsic.upv.es

Abstract

With the recent advent of Grid Computing technologies, resource-starved applications can greatly benefit from the power that the Grid aims to deliver. In this paper, we focus on the application of Grid Computing to a biomedical computationally intensive application that simulates the electrical activity on cardiac tissues. A user level tool, called GMarte, was developed to simplify the usage of Grid facilities for the execution of scientific applications on distributed deployments. A cardiac case study has been executed, via GMarte, on the largest distributed Grid deployment available in Europe (the EGEE testbed) in order to assess the benefits of Grid Computing. The usage of a Grid infrastructure has dramatically reduced the time required to execute the case study, thus increasing research productivity.

1. Introduction

In this paper, we describe the usage of Grid Computing technologies, by using the GMarte middleware [3], to support the execution of parametric studies of a computationally intensive biomedical parallel application that simulates the electrical activity in cardiac tissues [1]. GMarte is a client-side middleware that abstracts the process of remote task execution on Grid deployments based on the industrial standard Globus Toolkit [4]. It was developed on top of the Java CoG Kit in order to simplify the usage of Globus. GMarte offers a high level Application Programming Interface (API), from the Java language, that enables the user to employ resource discovery functionality as well as fault-tolerant metascheduling capabilities for the allocation of tasks to computational resources.

The GMarte middleware can currently interact with either resources under the Globus Toolkit 2, as well as with Computing Elements in a LCG[1] deployment, thus allowing to span executions in the EGEE[2] testbed, one of the largest computational deployments devoted to Grid Computing in science. Although we could employ the client tools of the LCG middleware to perform the executions, we have preferred to use GMarte because this enables us to modify the scheduling policy for task allocation. This middleware only needs to be installed in the client machine.

*The authors would like to thank the financial support received from the Spanish Ministry of Science and Technology to develop the project GRID-IT (TIC2003-01318) and the Comission of the European Communities, Directorate-General Information Society, Research Infrastructures, for the EGEE project (IST-2003-508833). This work has been partially supported by the Structural Funds of the European Regional Development Fund (ERDF).

[1]LCG - LHC Computing Grid Project. http://lcg.web.cern.ch.

[2]EGEE - Enabling Grids for E-sciencE. http://eu-egee.org.

1063-7125/05 $20.00 © 2005 **IEEE**

Table 1. Initial state of the resources and distribution of the simulations in the testbed, for each machine. The number in parentheses indicates the number of nodes involved in the execution

Machine	Country	Available Nodes	Number of Simulations
GRyCAP	Spain	8	2 (7 p.), 1 (2 p.)
CNAF-INFN	Italy	7	7 (1 p.)
IN2P3-1	France	42	2 (1 p.)
IN2P3-2	France	28	8 (1 p.)
BA-INFN	Italy	32	1 (1 p.)
SINICA	Taiwan	65	-

2. Assessing the Efectivity of the Grid: A Case Study Simulation Approach

Cardiac studies are computationally and data intensive tasks that require the execution of a large number of simulations for a single case study. For example, to assess the effects of new medicines in the electrical behaviour of the heart, it is required to vary the drug concentration, over an interval, to measure how the action potential propagation is altered. This gives place to multiple independent simulations that can be executed on a Grid infrastructure.

To investigate the efectiveness of Grid Computing, we have considered a case study that analyses the influence, in the action potential propagation, of different degrees of ischemic conditions (a pathology that can lead to ventricular fibrillation) that take place from 0 to 10 minutes from the onset of a myocardial ischemia in a tissue. This case study was originally executed on a restricted Grid deployment, composed of machines from our research group and from another university at Spain, with a naive Globus-based Grid prototype. A fully detailed description of the case study, which requires the execution of 21 parametric simulations, can be found in a related work [2]. The simulation time has been reduced from 80 ms., in the original case study, to 20 ms.

The resources considered for execution are those available in the *Biomed* Virtual Organisation of the EGEE testbed (a set of resources belonging to different sites that carry out or support biomedical research). In addition a Globus-based MPI-enabled cluster of PCs, belonging to our research group (GRyCAP), has been employed. Currently, the submission of parallelized tasks with the MPI standard is not supported in the EGEE testbed. Therefore, even though GMarte fully supports the execution of parallel applications, on multiprocessor or clusters of PCs, the executions on machines belonging to the EGEE testbed will always be sequential, that is, with just one processor.

Table 1 indicates the available computational nodes of the resources at the initial stage of the scheduling, as well as a summary of the task allocation process in the Grid deployment. Two of the simulations, initially allocated to IN2P3-2, resulted in a premature failed execution, but the implemented policy in GMarte re-scheduled them as long as the remaining tasks had already been allocated. Both failed executions were assigned to GRyCAP, one employing 7 processors and the other using 2 processors.

It can be seen that most of the simulations were assigned to the resources in the geographical neighbourhood. In fact, notice that although the machine in Taiwan had the largest number of available computing nodes during the case study execution, the scheduler never selected it. Indeed, the cost of all the data transfer from the local machine (in Spain) to Taiwan was computed to be up to 7 times more expensive than to one of the resources located at Italy. Therefore, the implemented resource selector followed a task allocation policy that considers resource proximity.

The total execution time of the case study, that is, since the scheduling process started until all the simulation data results were available on the local machine, was 102 minutes. A sequential execution of the case study, one simulation after another on a Pentium IV at 2 GHz machine (an average platform on the EGEE testbed, to make the comparison as fair as possible), required a total of 1569 minutes. Using a cluster of such PCs, performing executions with 8 processors, lasted for a total 221 minutes. In fact, if there were more resources in the tesbed than tasks to be executed, all simulations would be performed simultaneously. Therefore, the only overhead would reside in the scheduling decisions and the data transfer both for sending the application and the input data files, as well as for retrieving the simulation output data. In our case, those two failed executions, that were re-scheduled, have also contributed to a slight increase in the global execution time.

Anyway, we have achieved a speedup of 15.38 when executing 21 simulations in a Grid Computing infrastructure. This implies to reduce the total execution time from more than one day to less than two hours. This dramatic reduction of the simulation time enables to increase the research productivity as much more results can be obtained in less time. Therefore, Grid Computing allows biomedical experts and scientists in general to face new challenges that require such amount of computing power that could overcome the resources of a single organisation.

3. Conclusions

In this paper, we have briefly described the application of Grid Computing technologies to a computationally intensive biomedical application that simulates the electrical activity of cardiac tissues. The parameter sweep tasks that arise from the cardiac case studies are very suitable for execution under a Grid deployment. The GMarte middleware provides an object oriented, high level view of Grid Computing, what enables to simplify the usage of Grid facilities for the transparent execution of applications on Globus-based Grids.

To assess the effectiveness of Grid Computing, a cardiac case study has been run on one of the largest testbeds available to the scientific community. We have shown that the power that a distributed infrastructure delivers allows to cope with resource-starved cardiac case studies, by the simultaneous execution of different simulations on the computational resources of the Grid.

Acknowledgements

The authors would like to thank the Instituto Nazionale di Fisica Nucleare (INFN), the Institute National de Physique Nucléaire et de Physique des Particles (IN2P3) and the Academia Sinica (SINICA) for providing the computational resources, in the framework of the EGEE project, on which the executions were performed.

References

[1] J. M. Alonso, J. M. Ferrero (Jr.), V. Hernández, G. Moltó, M. Monserrat, and J. Saiz. Simulation of Action Potential Propagation on Cardiac Tissues: An Efficient and Scalable Parallel Approach. *Parallel Computing: Software Technology, Algorithms, Architectures and Applications, included in series: Advances in Parallel Computing*, pages 339–346, 2004.

[2] J. M. Alonso, J. M. Ferrero (Jr.), V. Hernández, G. Moltó, M. Monserrat, and J. Saiz. Tree Dimensional Cardiac Electrical Activity on Cluster and Grid Platforms. In *6th International Meeting on High Performance Computing for Computational Science, VECPAR'04*, pages 485–498, 2004.

[3] J. M. Alonso, V. Hernández, and G. Moltó. Enabling High Level Access to Grid Computing Services. *ERCIM News. Special: Grids: The Next Generation*, 59:42–43, 2004.

[4] I. Foster and C. Kesselman. Globus: A Metacomputing Infrastructure Toolkit. *Intl. J. Supercomputer Applications*, 11(2):115–128, 1997.

1063-7125/05 $20.00 © 2005 **IEEE**