

Air Pollution Modeling in the CrossGrid Project*

J. Carlos Mouriño, María J. Martín, Patricia González, and Ramón Doallo

Computer Architecture Group. Department of Electronics and Systems
University of A Coruña, Spain
jmourino@mail2.udc.es

Abstract. The CrossGrid project develops, implements and exploits new Grid components for interactive compute and data intensive applications like simulation and visualization for surgical procedures, flooding crisis team decision support systems, distributed data analysis in high-energy physics, and air pollution combined with weather forecasting. We present in this paper the integration of the air pollution application with other components of the project and show the benefits of using a Grid platform for running this application.

1 Introduction

Grid technologies have been under development since the late 1990s, extending cluster and distributed computing concepts to allow organizations to share and combine resources. The purpose of this work is to show the usefulness of the Grid in air pollution modelling. STEM-II, an Eulerian air quality model, is used to simulate the environment of As Pontes Power Plant in A Coruña (Spain). The STEM-II program, forced by meteorological data generated by an atmospheric model, simulates gaseous and aqueous concentrations fields of different pollutant species, reaction rates, amount of deposited species and ionic concentrations. The code is computationally intensive, thus, Grid computing should be potentially applied to achieve a reasonable response time.

The CrossGrid Project [1] offers us a Grid environment oriented towards compute and data-intensive applications that need interaction with an external user. Our model needs the interaction of an expert in order to make decisions about modifications in the industrial process. For this reason, besides the parallelization of the model, a Graphical User Interface (GUI) was developed with the possibility of testing the results and modifying parameters in real time.

The structure of this paper is as follows. Section 2 presents a brief overview of the CrossGrid Project. In Section 3, we briefly describe the air pollution model STEM-II. The GUI designed to support the interactivity needed for the application and its functionality is also described. In Section 4 the interactions with other CrossGrid components are shown. Finally, in Section 5, we briefly comment the future work.

* This work is supported by the CrossGrid European Project (Ref: IST-2001-32243)

2 The CrossGrid Project

The project is developed with the purpose of supporting applications that require high computational resources and a great amount of data, with the characteristic that real time interaction with users is needed. Examples of these applications are: interactive simulation and visualization for surgical procedures [2], flooding crisis team decision support systems [3], distributed data analysis in High-Energy Physics (HEP) [4], and air pollution [5] combined with weather forecasting [6].

For the efficient development of this kind of applications into the Grid, new tools for verification of parallel source code, performance prediction, performance evaluation and monitoring are needed and are also developed into the project. All these applications and tools require new components into the Grid for application-performance monitoring, efficient distributed data access, and specific resource management. The final users should run the applications on the Grid in an easy and transparent way, without the need of knowledge on technical details or on how the Grid is implemented and how it works. Thus, the project develops user-friendly portals and personalized environments to access the applications and developed tools.

The project also offers a testbed architecture to test, validate and execute tools, applications and the Grid services developed. Testbed sites are distributed among ten different institutions in nine European countries. All this individual elements, together with others achievements from other Grid projects such as DataGrid and EuroGrid, are integrated in a new Grid environment.

3 The STEM-II Air Quality Model

Coal-fired electrical power plants constitute one of the most significant sources of air pollution, and for that reason, its study is one of the keys in pollution control. The aim of our work is to control the emissions produced by the Endesa Power Plant sited at As Pontes (A Coruña, Spain). This power plant generates 1400 MW of electrical energy by means of coal combustion. A coal mixture made of a local lignite, with a high content in sulphur, and other foreign coals, of greater calorific power and free of sulphur, are consumed. One of the objectives of the technicians of the power station is to know the optimal mixture so that the maximum yield is obtained fulfilling, at the same time, the norm on emissions and quality of the air.

In order to carry out the complete characterization on a regional scale of the dispersion of atmospheric pollution, it is necessary to simulate different pollutant emissions, transport by advection, convection and turbulence, with dynamic meteorological conditions; chemical transformation; and pollutant removal by deposition. STEM-II (Sulphur Transport Eulerian Model 2) is an Eulerian air quality model which simulates transport, chemical transformations, emissions and deposition processes in an integrated framework [7]. The model is computationally intensive because the governing equations are nonlinear, highly coupled and stiff. The speedup to be achieved in the simulation process is really important in order to save time when making decisions about modifications in the

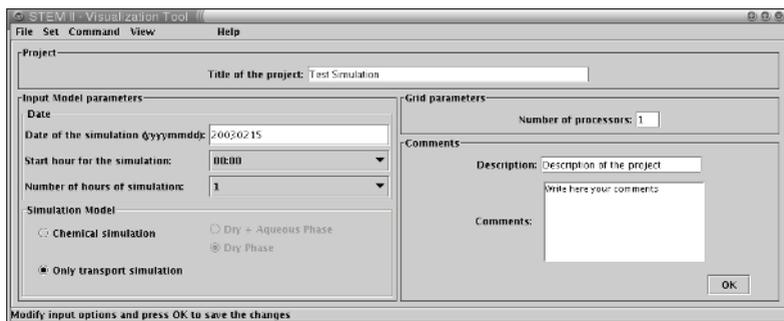


Fig. 1. Input parameters window

industrial process to fulfill the European regulations referring to the emission of pollutants. As with other computationally intensive problems, the ability to fully utilize these models remains severely limited by today's computer technology. Grid computing will be applied to achieve a reasonable response time.

3.1 Supporting STEM-II Interactivity in a Grid Platform

With the aim of helping the experts to make decisions, we have developed a user-friendly interactive visual tool which allows to set the input/output parameters, and watch the concentrations of pollutants graphically. This interface is interactive, that is, the technicians can vary parameters in real time and follow graphically the impact of these changes.

One of the features we had in mind to develop the graphical user interface (GUI) was the portability of the final tool. The Java Foundation Classes (JFC) are a set of Java class libraries provided as part of J2SE to support building graphics user interface and graphics functionality for Java technology-based client applications. Apart from these features, JFC is also used by other developers in the CrossGrid project. So the use of Java technology to develop the Air Pollution Visualization Tool will simplify its future integration with other tools of the project.

Another important objective to be considered was the abstraction of the graphical interface. This objective aims to present the Grid execution as transparent as possible for the user. To achieve it the Visualization Tool consists of three main components:

- User Interface, in charge of building the user environment: windows, panels, buttons, messages, etc.
- Data processing module, in charge of the image processing from the original data files. A module to process and convert the data in these matrices into image files is also provided in the Visualization Tool.
- Interface with the Grid platform, includes the construction of the RSL files and the job submission to the Grid platform via Globus. This is the most complex part of the tool. It involves relations both with the User Interface and with the data processing module.

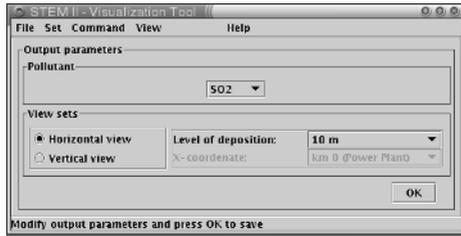


Fig. 2. Output parameters window

An initial window of the tool including the logo of the project and a menu bar with several options (**File**, **Set**, **Command**, **View** and **Help**) is first presented to the user. In the **File** item the user can choose the typical options: start a new project, load a saved project, store the actual project or finish the application.

The input and output parameters of the model can be set in the **Set** item (see Figure 1). This parameters are the date, the start hour and the number of hours of the simulation. Besides, some flags for the execution of an only transport simulation, or a simulation with chemical reactions in dry, or dry plus aqueous phases can be set.

In the **View** item we can see the specific graphical representation chosen in the output parameters window (see Figure 2). In this window the pollutant to visualize (SO₂, O₃, SO₄-) can be set, as well as the type of view (horizontal or vertical), and the level or coordinate of representation in function of the view.

From the **Command** item the program can be submitted to the Grid. A graphical window with the selected output will be generated. Figure 3 shows an example with an horizontal view. The **type** box shows the pollutant that is being represented while the **level** box presents the altitude of the representation. The concentrations are represented with a background map for a better understanding of the results. The location of the power plant is also indicated. Moreover, a scale with a color code is shown on the right part of the image. The image is refreshed on real time when new concentration data are available. We can change interactively any output parameter, and the new representation will be shown.

4 The Air Pollution Application into the CrossGrid Framework

The STEM-II model requires as input data meteorological data such as temperature, wind fields, precipitation, etc. This meteorological data are provided by the COAMPS [6] meteorological prediction model maintained in the Interdisciplinary Centre for Mathematical and Computational Modelling (ICM) at Warsaw. This application generates data in the format and resolution needed. The CrossGrid project provides mechanisms to share these data. The file management policy is based on replication. Replica catalogues are used for cataloging and locating replicated files in the distributed CrossGrid environment.

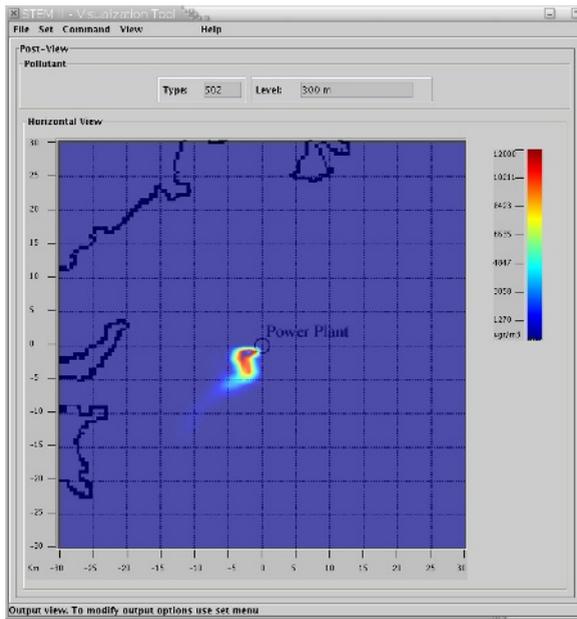


Fig. 3. Horizontal view of SO₂ concentration at 300 m

Previously to the Grid-enable version of the code we have developed a MPI parallel version of the program. Details about this parallelization and experimental results in different platforms can be obtained from previous works [5]. Additionally, an experimental Grid built in our laboratories has also been used for testing the parallel code [8].

The CrossGrid project offers the possibility of testing the MPI implementation using MARMOT [9], an MPI verification tool developed at the High Performance Computing Center Stuttgart (HLRS). The objective of MARMOT is to verify the correctness of parallel, distributed Grid applications using the MPI paradigm. Thanks to this tool some “holes” were detected in our communications, which introduced unnecessary overhead in the parallel program. We packed non-consecutive data in each communication using complex datatypes. These datatypes have useful data and “holes” that must be skipped in the communications. MARMOT warned us that these “holes” were communicated as empty data. Now, that “holes” have been eliminated, changing the datatypes and increasing the number of communications, and the performance of the communications has been improved.

Besides, there are another tools in the project that propose a set of performance metrics to describe concisely the performance capacity of Grid applications. One of them is GridBench [10], developed in the High Performance Computing Laboratory (HPCL) of Dept. of Computer Science at University of Cyprus. It develops and implements benchmarks that are representative of typical Grid workloads. Such benchmarks are used for estimating the values of

performance metrics for different Grid configurations, identifying important factors that affect end-to-end application performance, and providing application developers with initial estimates of expected application performance. Another task in the project develops an on-line tool, PPC [11], that allow application developers to measure, evaluate and visualize the performance of Grid applications with respect to data transfer, synchronization and I/O delay as well as CPU, network and storage use.

The MPI parallel code was transported to the Grid environment fairly easily by introducing appropriate components from the Globus bag of services [12]. Information about available Grid resources is essential for job scheduling. The Globus toolkit uses the MDS (Monitoring and Discovering Service) to publish static and dynamic information about existing resources. This implementation is based on the LDAP protocol. The Workload Management System (WMS) manages the Grid resources guarantying that jobs are executed in the “best” resources. The WMS consist of: the Resource Broker (RB) [13], that is the responsible of matching job requests, written in JDL, with the available resources; the Job Submission Service (JSS), that performs the job submission, using the Globus GRAM service, to the remote Computing Element found by the RB; the User Interface (UI), which is the component between the user and the RB and allows the user to obtain information about the jobs, transfer data files, find resources, cancel jobs, etc; and the Logging and Bookkeeping (LB), that keeps information about the executions.

Some results have been obtained submitting directly the program to specific nodes of the CrossGrid testbed (See next Section 4.1). Currently, we are working in the integration of our application into the portal and the migration desktop. Both elements are developed with the support of the roaming access subtask, that allows the users to access the Grid from different locations and different platforms. This is mainly developed by the Poznan Center of Supercomputing (PSNC) in Poland, and Datamat, Italy.

The portal is developed, by Algosystems, with the Jetspeed [14] technology, using Java and XML. A portal makes network resources available to end-users and the user can access the portal via a web browser. Each application must create its own portlet with the fields for the parameters needed to submit the application, in an XML specification. The migration desktop, however, is a personalized graphical desktop developed using Java technologies by the PSNC. The applications must provide specific plugins in order to be submitted to the Grid. From these plugins the parameters of the application and other general parameters as the number of processors, the data files, etc can be specified. The application developers can also provide output plugins to visualize the results in a graphic and interactive manner.

4.1 Preliminary Results in the CrossGrid Testbed

We have obtained some experimental results executing the air pollution application in the CrossGrid Testbed. Table 1 shows the results of executing a simulation of one hour among nodes that belong to different sites and nodes that belong to

the same site of the testbed. Sites used to build Table 2(a) are placed at Univ. Valencia (Spain), Univ. Warsaw, Institute of Nuclear Physics (Poland) and Al-gosystems (Greece) while executions for Table 2(b) were carried out on nodes of the same site at Cyfronet (Poland). In this last case the obtained results are much better due to the lower communication cost. This should be taken into account in the design of the scheduler tool of the CrossGrid project.

Table 1. Times and speedup of 1 hour of real time (60 iterations)

proces.	tot. time	com. time	speedup
1	2133	20	-
2	5107	485	0.42
3	20212	13126	0.11
4	37874	24359	0.06

(a) executed in different sites

proces.	tot. time	com. time	speedup
1	4952	11	-
2	3316	73	1.49
3	2480	98	2.00
4	1865	112	2.65
5	2535	122	1.95
6	2466	125	2.01
7	2198	132	2.25
8	1304	136	3.80

(b) on nodes of the same site

It can be observed that the speedup does not grow linearly. This is because different iterations of the simulated mesh have different computational weight. We have observed experimentally that the most time-consuming iterations are due to presence of rain in that part of the simulated space. We are working on finding some different data distributions to increase the performance. Besides, the project offers tools for application monitoring, helping developers to detect bottlenecks and optimize the code. We plane to use some of these tools to improve the parallel performance.

5 Future Work

At this moment the parallel Resource Broker is in a testing stage. For this reason we are working with two versions of the program, one sequential for the automatic submission (through the RB using JDL) and a parallel version that currently is used for manual submission (via a RSL script and Globus). When the resource broker will be fully available we will submit our parallel version through the RB, and we will analyze its performance.

Currently, we are working on the integration of the air pollution application and its GUI into the portal and migration desktop, emphasizing in the interactivity feature. We also intend to improve the performance of the parallel STEM-II model using the tools that the CrossGrid project offers for performance prediction and monitoring. Moreover, we are working on a better data distribution to achieve large efficiency in the testbed.

References

1. Crossgrid home page. <http://www.crossgrid.org/>.
2. R.G. Belleman and P.M.A. Sloot. The Design of Dynamic Exploration Environments for Computational Steering Simulations. In *Proceedings of the SGI Users' Conference 2000*, volume 1, pages 57–74. Academic Computer Centre CYFRONET AGH, Krakov, Poland, October 2000.
3. L. Hluchy, O. Habala, B. Simo, J. Astalos, V.D. Tran, and M. Dobrucky. Problem Solving Environment for Flood Forecasting. In *1st International NAISO Symposium on Information Technologies in Environmental Engineering (ITEE'2003)*, Gdansk, Poland, June 2003.
4. S. Bethke, M. Calvetti, H.F. Hoffman, D. Jacobs, M. Kasemann, and D. Linglin. Report of the Steering Group of the LHC Computing Review. In *CERN/LHCC/2001-004, CERN/RRB-D 2001-3*, February 2001.
5. María J. Martín, David E. Singh, J. Carlos Mouriño, Francisco F. Rivera, Ramón Doallo, and Javier D. Bruguera. High Performance Air Pollution Modeling for a Power Plant Environment. *Parallel Computing*, 29 (11–12):1763–1790, 2003.
6. M. Niezgodka and B. Jakubiak. Numerical Weather Prediction System: Scientific and Operational Aspects. In *Proceedings of the III Symposium on Military meteorology*, pages 191–197, 1998.
7. G.R. Carmichael, L.K. Peters, and R.D. Saylor. The STEM-II Regional Scale Acid Deposition and Photochemical Oxidant Model - I. An Overview of Model Development and Applications. *Atmospheric Environment*, 25A(10):2077–2090, 1991.
8. José C. Mouriño, María J. Martín, Patricia González, Marcos Boullón, José C. Cabaleiro, Tomás F. Pena, Francisco F. Rivera, and Ramón Doallo. A Grid-enable Air Quality Simulation. In *First European Across Grids Conference*, 2003.
9. Bettina Krammer, Katrin Bidmon, Matthias S. Müller, and Michael M. Resch. MARMOT: An MPI Analysis and Checking Tool. In *Parallel Computing 2003*, Dresden, Germany, September 2003.
10. G. Tsoloupas and M. Dikaiakos. GridBench: A Tool for Benchmarking Grids. In *4th International Workshop on Grid Computing (Grid2003)*, Phoenix, Arizona, 2003. Accepted.
11. V. Blanco, P. González, J.C. Cabaleiro, D. Heras, T.F. Pena, J.J. Pombo, and F.F. Rivera. Visualizing the Performance Prediction of Parallel Iterative Solvers. *Future Generation of Computer Systems*, 19:721–733, 2003.
12. I. Foster and C. Kesselman. GLOBUS: a Metacomputing Infrastructure Toolkit. *International Journal Supercomputing Applications*, pages 115–128, 1997.
13. Elisa Heymann, Miquel A. Senar, Emilio Luque, and Miron Livny. Adaptive Scheduling for Master-Worker Applications on the Computational Grid. In *Lecture Notes in Computer Science*, editor, *Proceedings of the Grid Computing - GRID 2000: First IEEE/ACM International Workshop*, volume 1971, page 214, Bangalore, India, December 2000.
14. D. Engh, S. Smallen, J. Gieraltowski, L. Fang, R. Gardner, D. Gannon, and R. Bramley. GRAPPA: Grid Access Portal for Physics Applications. June 26 2003. Talk from the 2003 Computing in High Energy and Nuclear Physics (CHEP03), La Jolla, Ca, USA, March 2003.