

HIGH PERFORMANCE AIR QUALITY SIMULATION IN THE EUROPEAN CROSSGRID PROJECT

J. Carlos MOURIÑO

*CESGA Supercomputing Center
Santiago de Compostela, Spain
e-mail: jmourino@cesga.es*

María J. MARTÍN, Patricia GONZÁLEZ, Ramón DOALLO

*Computer Architecture Group
Department of Electronics and Systems
University of A Coruña, Spain*

Manuscript received 11 October 2004

Communicated by Ladislav Hluchý

Abstract. This paper focuses on one of the applications involved into the CrossGrid project, the STEM-II air pollution model used to simulate the environment of As Pontes Power Plant in A Coruña (Spain). The CrossGrid project offers us a Grid environment oriented towards computation- and data-intensive applications that need interaction with an external user. The air pollution model needs the interaction of an expert in order to make decisions about modifications in the industrial process to fulfill the European standard on emissions and air quality. The benefits of using different CrossGrid components for running the application on a Grid infrastructure are shown in this paper, and some preliminary results on the CrossGrid testbed are displayed.

Keywords: Air Quality models, STEM-II, multicomputers, Grid, MPI, CrossGrid Project

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. Grids consist of high-end computers, servers, workstations, storage systems, and databases that work in tandem across private and public networks. The major distinguishing feature of Grids is that they are configured from components that are owned and/or managed by independent individual organizations.

The CrossGrid project [1] implements and exploits new Grid components for interactive computation- and data-intensive applications. The purpose of this work is to show the usefulness of these components 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 concentration fields of different pollutant species, reaction rates, amount of deposited species and ionic concentrations. The code is computation intensive; thus, Grid computing should be potentially applied to achieve a reasonable response time.

The structure of this paper is as follows. Section 2 presents a concise overview of the CrossGrid Project. In Section 3, we briefly describe the air pollution model STEM-II. In Section 4 the interactions with other CrossGrid components are shown, including the Migrating Desktop and the Performance Prediction Tool. Some preliminary results are shown in Section 5. Finally, in Section 6, we make some brief comments about future work. The main conclusions of the work are given in Section 7.

2 THE CROSSGRID PROJECT

The CrossGrid project involves institutions that belong to eleven European countries (see Figure 1). 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].

The first application develops a Grid-based prototype system for pre-treatment planning in vascular interventional and surgical procedures through real-time interactive simulation of vascular structure and flow. The second application is a Grid-based support system for flood prevention and protection; this is a numerical flood modeling that requires a physical model for flood representations. The third application develops a final user application for physic analysis running in a distributed Grid environment using large distributed databases for High Energy Physics. The last application is weather forecast and air pollution modelling. This application includes a meteorological model, a wave model, a data mining system for meteorological reanalysis, and an air pollution model.

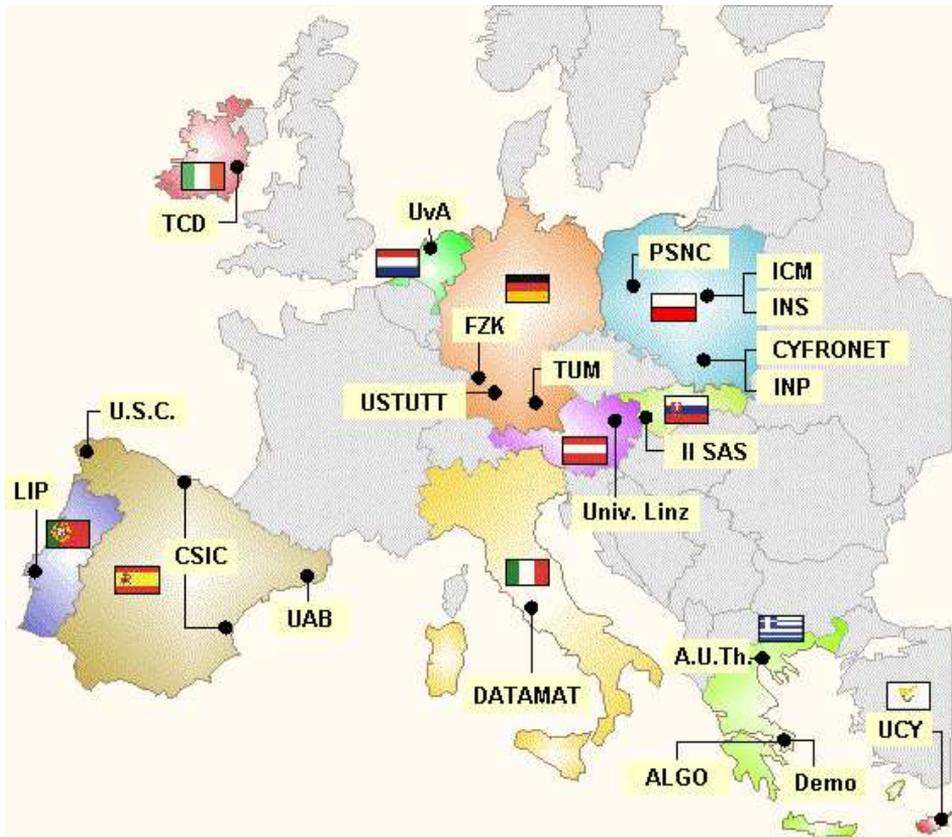


Fig. 1. Countries involved in the CrossGrid project

For the efficient development of these kinds 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 be introduced into the Grid for application-performance monitoring, efficient distributed data access, and specific resource management. The final users should be able to run the applications on the Grid in an easy and transparent way, without the need for knowledge of technical details or of how the Grid is implemented, nor of 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 these individual elements, taken together with achievements from other Grid projects such as DataGrid and EuroGrid, are integrated in a new Grid environment.

3 THE STEM-II AIR QUALITY MODEL

Due to increasing sources of air pollutants, the development of tools to control and prevent the accumulation of pollution has become a high priority. 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 to pollution control.

Specifically, we are interested in the control of the pollutants' emissions produced by the Endesa Power Plant sited at As Pontes (A Coruña, Spain). This power plant generates 1 400 MW of electrical energy by means of coal combustion. A coal mixture made of local lignite, with a high content in sulphur, and other foreign coals, of greater calorific power and sulphur-free, are consumed. One of the objectives of the technicians of the power station is to ascertain the optimal mixture so that the maximum yield is obtained, fulfilling, at the same time, the standards on emissions and air quality.

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. To carry out these simulations, the STEM-II model has been employed [7].

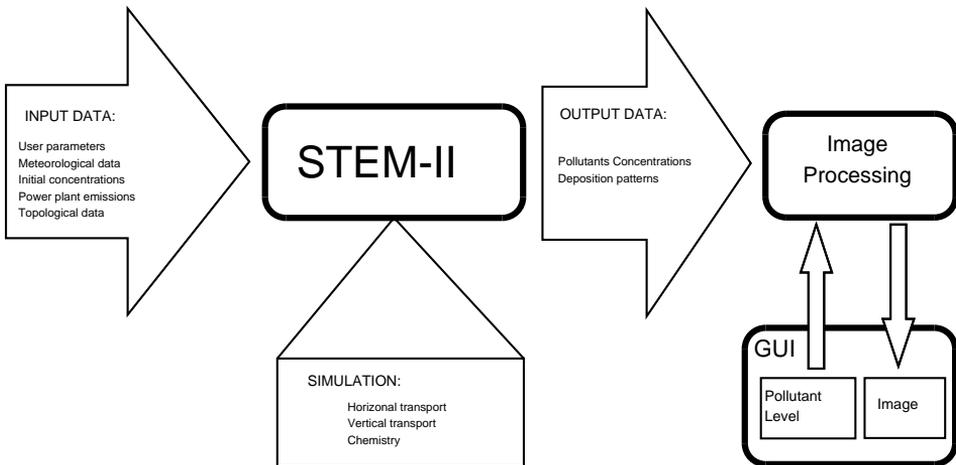


Fig. 2. Simplified block diagram of the STEM-II model

STEM-II (Sulphur Transport Eulerian Model) is an Eulerian air quality model which simulates transport, chemical transformations, emissions and deposition pro-

cesses in an integrated framework. The basic structure of the STEM-II model is shown in Figure 2. The model accepts as input data the parameters introduced by the user: the date of the simulation, the start hour, and the number of hours of simulation; the meteorological conditions in that moment in the power plant environment: wind, temperature, precipitation rates, etc.; the initial pollutant concentrations in the simulation space; the emissions of the power plant and the topological data. The simulation module of the model consists of two main components: the horizontal transport and the **vertlq** module, that includes the vertical transport of pollutants and the chemistry reactions, the most time-consuming routine. As outputs, the program generates binary files with the concentrations of all the species in every point of the simulated mesh and the amount deposited. Additionally, we have developed a graphical module where the output data files are processed and converted into graphical images taking into account the pollutant and the level selected by the user. This module is described in detail in Section 4.1 below.

The pseudocode of the STEM-II model is shown in Figure 3. The program consists of four main nested loops, a temporal loop (*loop_t*) and a loop for each dimension of the simulated space (*loop_x*, *loop_y* and *loop_z*). The temporal loop specifies the duration of the simulation process and the time simulated in each iteration. Usually, meteorological prediction data are needed to evaluate the pollutants behavior of the next day, therefore we will need to simulate at least 24 real hours. In our experiments each iteration of the temporal loop corresponds to a minute of real time. Beside this, there exists a fifth loop corresponding to the reaction time of the aqueous phase. The basic modules and routines of the program are shown in lower case and bold letters in the figure. The main modules in the program are: horizontal transport module (**hor** module), **vertlq** module and I/O module. Regarding the I/O module, before starting the temporal loop the time independent data are read. These data are: longitude and latitude of the domain, topography of the environment and initial concentration of the pollutants. In each iteration of the temporal loop the time dependent data are read. These data are: air density, relative humidity, temperature in each point of the grid, the components of wind or velocity of deposition. Finally, the output of the program provides the pollutant concentrations in all the points of the grid and the deposition patterns.

The simulation used by the STEM-II model is extremely expensive, both in terms of amount of resources consumed and of execution time. As an example, a typical simulation of pollutants dispersion for 1 day requires approximately 31 hours of CPU in an Origin 2000 system with R10000 processor at 250 MHz. The speedup to be achieved in the simulation process is really important in order to save time when making decisions about modifications in the 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.

```

input1: Read topological data and initial concentrations
DO loop_t (temporal loop)
  input2: Read meteorological data

  hor: Horizontal transport
      -- Begin vertlq Module (Parallel Section) --

  DO loop_y
    DO loop_x
      IF ( $\exists$  gas_phase y  $\exists$  aqueous_phase) THEN
        asmm : cloud parameters model (includes loop_z)
        vertcl: vertical transport for gas species (includes loop_z)
        DO loop_trfl
          vertcl: vertical transport for aqueous species (includes loop_z)
          rxn: chemical reaction (includes loop_z)
        END DO
      ELSE
        vertcl: vertical transport for gas species (includes loop_z)
        rxn: chemical reaction (includes loop_z)
      END IF
    END DO
  END DO

      -- End vertlq Module --

  output: Compute totals accumulated in the time and print result files
END DO

```

Fig. 3. Pseudocode of the STEM-II program

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 is 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 this data. The file management policy is based on replication. Replica catalogues are used for cataloging and locating replicated files in the distributed CrossGrid environment.

Previously to the Grid-enabled version of the code a MPI [8] parallel version of the program was developed. Details about this parallelization and experimental

results in different platforms can be obtained from previous works [9],[5]. Additionally, an experimental Grid built in our laboratories has also been used for testing the parallel code [10].

The CrossGrid project offers the possibility of testing the MPI implementation using MARMOT [11], an MPI verification tool developed at the High Performance Computing Center Stuttgart (HLRS). The objective of MARMOT is to verify the correctness of parallel applications using the MPI paradigm. The basis of this development is the MPI profiling interface, which allows a detailed analysis of the MPI application at runtime. Thanks to this tool some “holes” were detected in our communications, which introduced unnecessary overhead in the parallel program. Non-consecutive data was packed in each communication using complex datatypes. These datatypes had useful data and “holes” that must be skipped in the communications. MARMOT discovered that these “holes” were communicated as empty data. Thus, those “holes” have been eliminated, changing the datatypes and increasing the number of communications, and the performance in communications has been improved.

Beside these, there are other 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 [12], developed in the High Performance Computing Laboratory (HPCL) of Department 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 on-line tools 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. This software is called Performance Prediction Tool (PPC) [13] and it was developed in the Department of Electronics and Computer Science at the University of Santiago de Compostela. The tool supports any level of granularity, from the Grid level down to the process level.

The MPI parallel code was ported to the Grid environment fairly easily by introducing appropriate components from the Globus bag of services [14]. 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. The Workload Management System (WMS) manages the Grid resources guaranteeing that jobs are executed in the “best” resources. The WMS consists of: the Resource Broker (RB) [15], responsible for 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, found resources, cancelled

jobs, etc.; and the Logging and Bookkeeping (LB), that keeps information about the executions.

Some results have been obtained submitting the program directly to specific nodes of the CrossGrid testbed (see Section 6). Currently, we have fully integrated our application into the Migrating Desktop and into the portal. 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 in Italy.

The portal is developed by Algosystems, with the Jetspeed [16] 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.

4.1 The Graphical User Interface of the STEM-II Model

With the aim of helping experts make decisions, other than generating the binary files with the pollutant concentration values in each point of the simulated mesh, we have developed a user-friendly interactive visual tool which allows them to set the input/output parameters, and observe 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 when we developed the graphical user interface (GUI) was the portability of the final tool. The Java Foundation Classes (JFC) [17] are a set of Java class libraries provided as part of J2SE to support building graphics user interface (GUI) and graphics functionality for Java technology-based client applications. JFC includes an extensive set of technologies that enable developers to create a rich, interactive user experience for client applications that can run on multiple platforms.

Apart from these features, JFC is also used by other developers in the CrossGrid project, in particular by the Portal and Migration Desktop teams. So the use of Java technology to develop the Air Pollution Visualization Tool would 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 this 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 image processing from the original data files. The results of simulation using the STEM-II code are stored as text files, in the form of multidimensional matrices whose coefficients represent the different

pollutants concentration values in the mesh. 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 (getting input and output parameters from the user to build the RSL file), and with the data processing module (providing it with updated data files from the execution at hand in the Grid platform).

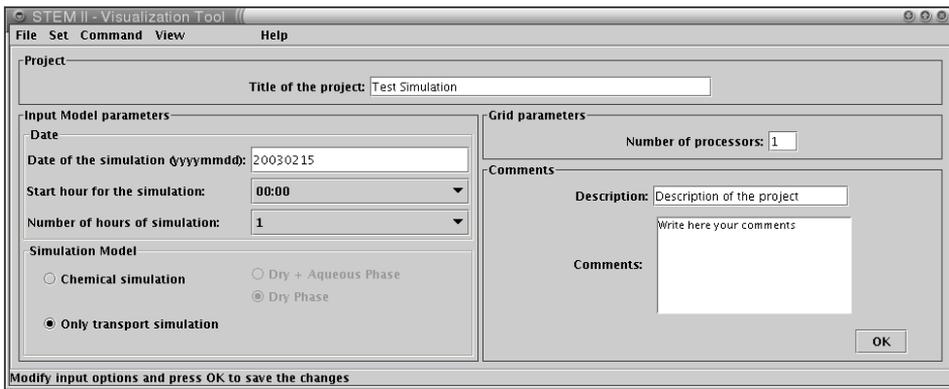


Fig. 4. Input parameters window

The input and output parameters of the model can be set in a specific window, to allow users to interact with the application, both at the beginning (before the submission) and during the execution. In the input parameters window (see Figure 4) the parameters of the application, such as the date, the start hour and the number of hours of the simulation are set. Beside these, since the application has been parallelized, the number of processors to be used also needs to be set.

The visualization tool represents in a 2-D graph the concentration of a given pollutant in a level of high, as shown in Figure 5. 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 in real time when new concentration data are available.

The main achievement of this GUI is its interactivity. The program runs into the Grid, and the output files are generated in any node of the Grid. When new data are generated the files are copied locally to the machine where the GUI is being executed and the output image is refreshed. The image processing module generates the images from the binary files accepting as parameters the pollutant, the type of representation and the level of this representation. These parameters can be set

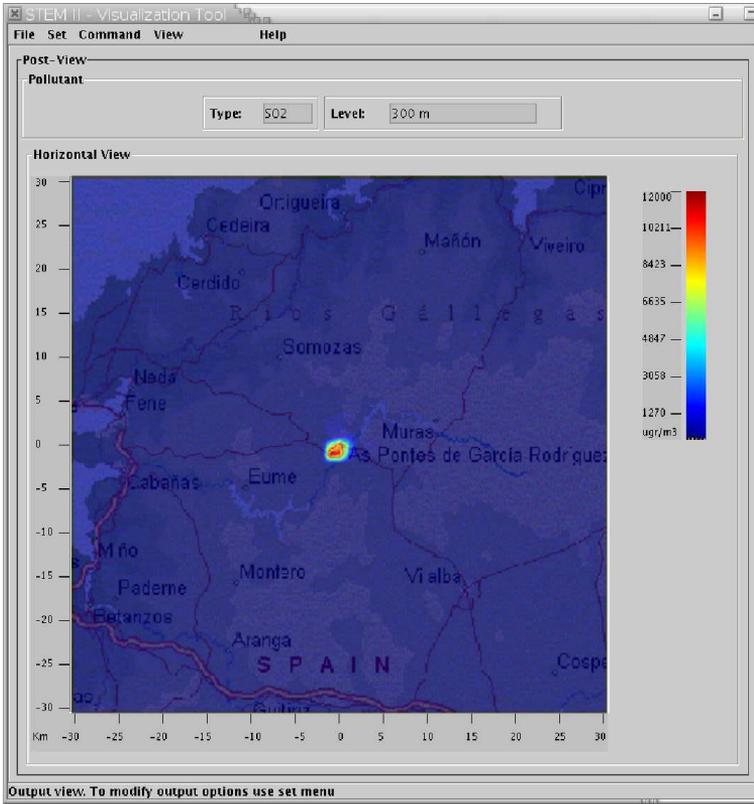


Fig. 5. Horizontal view of SO2 concentration at 300 m

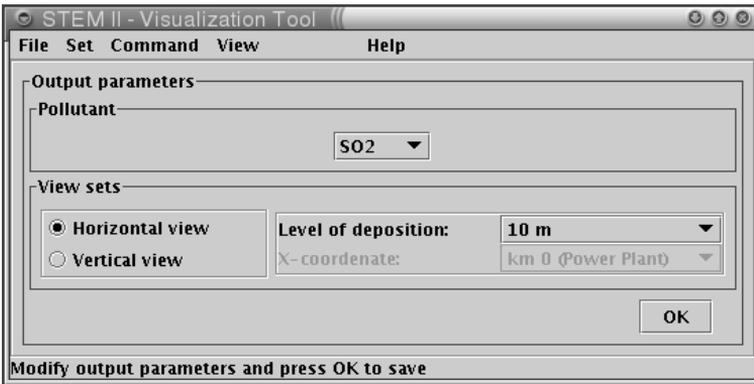


Fig. 6. Output parameters window

in the output parameters window (Figure 6). The user can vary these parameters while the program is running and the new representation will be presented.

With the help of this tool, experts can see, in real time and in a graphical mode, the concentration of the different pollutants, and make decisions about the coal combustion.

4.2 Integration of the STEM-II Model into the Migrating Desktop

The Graphical User Interface has been completely integrated into the Migrating Desktop. The Migrating Desktop (MD) [18] is a personalized graphical desktop developed using Java technologies by the PSNC, which allows working with Grid resources, hiding the complexity of the Grid middleware and the real location of the resources. MD provides access to the applications from different locations and different platforms, as well as interactivity properties. It is just a front end to Remote Access Server (RAS), which intermediates between different Grid middleware and applications. The RAS offers a well-defined set of web-services that can be used as an interface for accessing HPC systems and services (based on various technologies) in a common, standardized way. It is possible to add tools, applications and to support visualisation of different formats using that mechanism.

Grid applications available for users are set in a user-friendly way in Job Submission Wizard in the MD. This wizard simplifies the process of specifying parameters, suggesting user defaults or recently used parameters. The applications must provide specific plugins in order to be submitted to the Grid. One of the goals of MD is to provide the containers, frameworks for plug-ins written by application developers. The MD offers a module called Tool Plug-in (TPlug) [19] that was designed in order to standardize the integration and provide easy access to the resources. It is possible to add tools, applications and to support visualisation of different formats using that mechanism. The TPlug can be used for visualization of Grid applications output and for the integration of a Java application into the MD.

As described in Section 4.1, the GUI of the STEM-II model has been developed in Java, thus it was easy to convert the frames of the Graphical User Interface into valid plugins for the MD. The input parameters window was included as the first tab of the Job Submission wizard (see Figure 7), allowing the user to set the specific parameters of the application. The TPlug module has been used as protocol of integration in order to pass these parameters to the MD submission service.

Beside the input plugin, the application developers must also provide the output plugin if they want to visualize the results. We have adapted the visualization window of the application GUI by using the TPlug provided by the Migrating Desktop in order to access the output file with pollutants concentrations. The Migrating Desktop allows to specify the output files and their refresh time. We will use the ASTEP file with a refresh of 20 seconds as output (see Figure 8). This file will be used in the output plugin to build the graphical image. As this file is refreshed while the program is running, the output images will be updated. The data files are converted into images thanks to the image processing module that is included

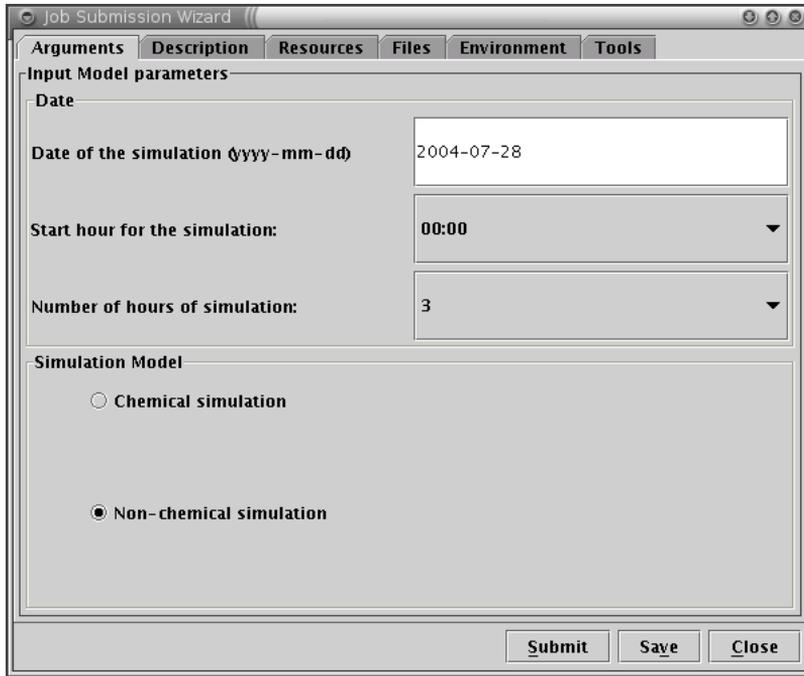


Fig. 7. Input parameters tab

into the output plugin. The appearance of the output is exactly the same as shown in the GUI (see Figure 9). This plugin is also interactive, so the user can vary the pollutant or the level in the visualization plugin, using the drop-down lists in the parameters boxes.

4.3 Improving the Performance of Our Parallelization

The Performance Prediction Tool (PPC) [13], developed also into the CrossGrid project, was used to analyze the performance of the air pollution application. In particular, we have focused on the module that consumes most of the runtime, the `vertlq` module (Figure 3). It was analyzed, and an analytical model for characterizing its performance was developed. This module basically consists of an intensive computational part that is executed in parallel and just one reduction operation that involves communications. Both parts are uncoupled, so the model adds both contributions.

The computational cost is measured in FLOPs (FLoating-point OPerations). Due to the complexity structure of `vertlq`, the number of FLOPs is not constant. The execution flow depends on the meteorological conditions and on the point of the mesh considered. A set of parameters can be selected as responsible for these

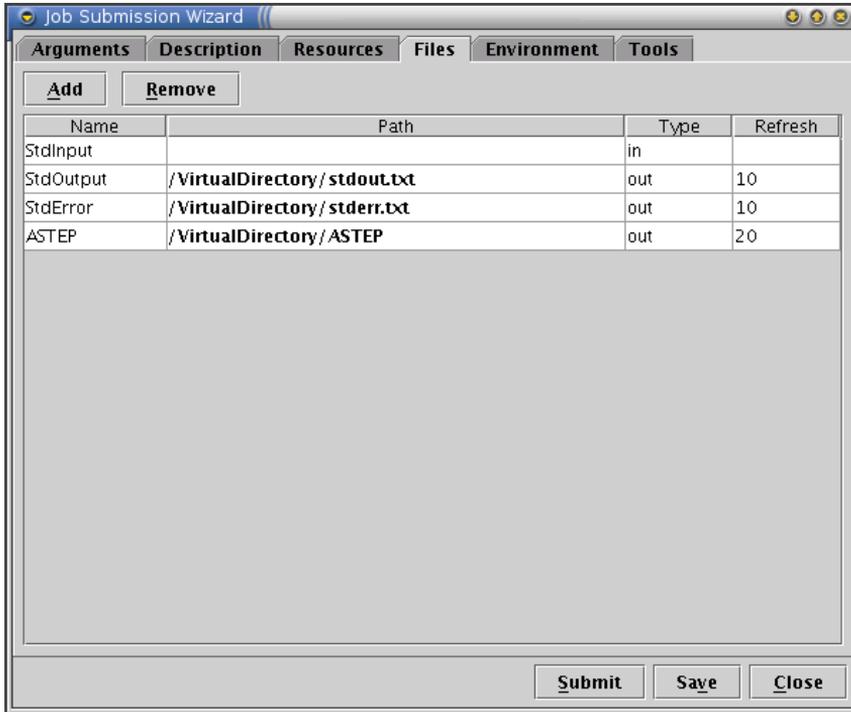


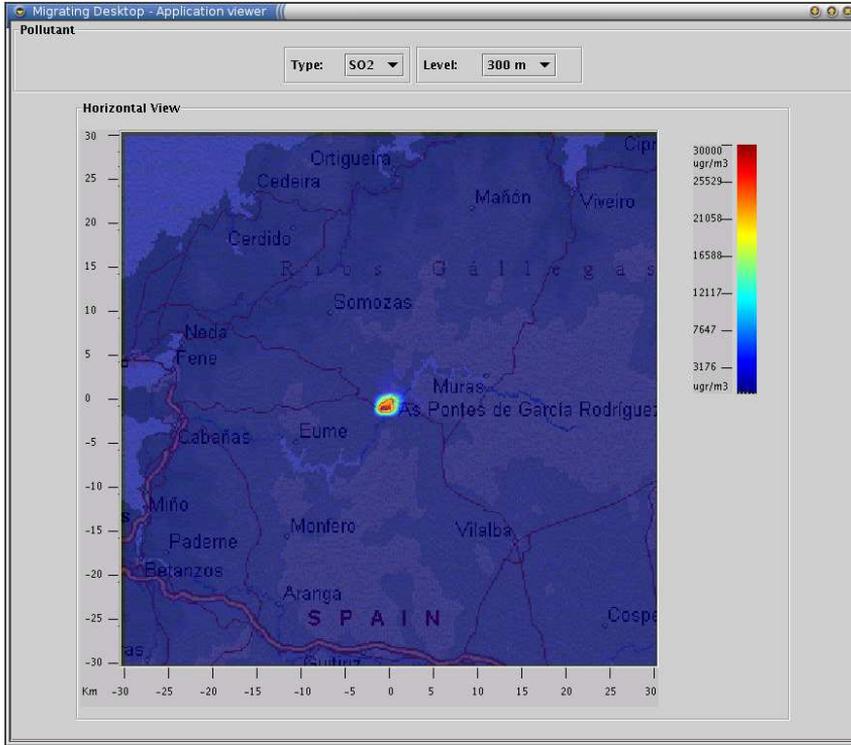
Fig. 8. Job files tab

differences in the execution time. These parameters will determine the entrance or not in the main conditional branches of the code and the number of iterations in the most time-consuming loops. After an exhaustive analysis, the model has concluded that 12 parameters are enough to make a prediction about the behavior of the program flow. These parameters can be determined exclusively from the meteorological input data and they are directly related to the water concentration.

Using this estimation of the number of FLOPs, we can deal with the problem of load balance in the `vertlq` module, without executing the routine, by only analyzing meteorological data. We can know in advance the best data distribution every time that new meteorological data are read.

5 RESULTS IN THE CROSSGRID TESTBED

We have obtained some experimental results executing the air pollution application in the CrossGrid Testbed. Tables 1 and 2 show, respectively, the results of running one hour of simulation among nodes that belong to different sites and nodes that belong to the same site of the testbed. The sites used to build Table 1 are placed at University of Valencia (Spain), University of Warsaw (Poland), Institute of Nuclear

Fig. 9. SO_2 pollution at 300 m high

Physics (Poland) and Algosystems (Greece) while executions for Table 2 were carried out in nodes of the same site at Cyfronet (Poland). In this last case the results obtained are better due to the lower communication cost. This should be taken into account in the design of the scheduler tool of the CrossGrid project.

processors	total time	comunication time	speedup
1	2 133	20	–
2	5 107	485	0.42
3	20 212	13 126	0.11
4	37 874	24 359	0.06

Table 1. Times (in seconds) and speedup of 1 hour of real time (60 iterations) executed in different sites

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 tried several dynamic and static data distributions in the parallelization of the application. Static methods involve deciding in advance the mapping of iterations

processors	total time	comunication time	speedup
1	4 952	11	–
2	3 316	73	1.49
3	2 480	98	2.00
4	1 865	112	2.65
5	2 535	122	1.95
6	2 466	125	2.01
7	2 198	132	2.25
8	1 304	136	3.80

Table 2. Times (in seconds) and speedup of 1 hour of real time (60 iterations) executed in nodes of the same site

to processors and leaving the balance unchanged during the whole execution. Static block and cyclic distributions do not guarantee an optimum load balance as the computational load of each iteration depends of the meteorological conditions. A better solution is to use a dynamic strategy. Dynamic load balancing involves changing the allocation of iterations to the processors during the execution of the simulation. Three dynamic distributions have been tried: queue of iterations, distribution by time and distribution by prediction. In the queue of iterations, the iterations are processed upon request. The root processor will send the iterations individually to the clients nodes when they have finished the previously assigned iteration. In the distribution by time, the time consumed by the first iteration when new data are read is measured, and the data is redistributed taking this time into account. Using the *Performance Prediction Tool* (PPC) [13] several parameters have been found to control the computational load of the program. With this information the FLOPs consumed by each iteration can be predicted, and the iterations can be well distributed from the first iteration. It is the distribution by prediction.

processor number	0	1	2	3	factor
block	1 148	777	1 607	1 813	0.74
cyclic	1 458	1 340	1 338	5 340	0.92
it. queue	1 532	1 448	1 512	5 660	0.91
by time	1 337	1 336	1 374	5 361	0.97
prediction	1 321	1 355	1 365	5 321	0.97

Table 3. Load balance for different data distributions

In Table 3 the different data distributions are compared. We have measured the execution time per processor for 60 iterations of the program over four processors. The measurements have been made in the `xgid.icm.edu.pl` CrossGrid Testbed node. The last column is the balance factor

$$factor = \frac{\sum_{i=1}^4 t_i}{4 \times \max(t_i)}$$

where t_i is the time consumed in the processor i . The closer this value is to the unit, the better balance is obtained. It can be observed that the block distribution gets the worst results and the distribution based on the PPC Tool (prediction in Table 3) gets the same load balance as the distribution by time. The distribution by prediction has less communication cost and a better total execution time; thus, we can conclude that the prediction tool successfully chooses the best data distribution.

6 FUTURE WORK

In a Grid platform, made up of a large number of nodes, the probability of fault in any part of the system is high. Thus, it would be advisable to endow the application with some kind of fault tolerance mechanism. Currently we are studying fault tolerance options for the air quality model. Our main idea is to periodically save the state of the application to a checkpointing file and to restart the application from the last checkpoint in case of fault.

7 CONCLUSIONS

The air pollution application has been integrated into the CrossGrid project, taking advantage of the integration with the middleware and other tasks of the project. With the use of MARMOT we have improved the process communications and with the help of the Performance Prediction Tool a better load balance has been achieved, proving the usefulness of other tasks of the project in the parallel implementation of applications into the Grid. Thanks to the graphical user interface for the STEM-II, experts can observe in real time, and graphically, the concentration of the different pollutants in order to make decisions in the industrial process. The integration of the air pollution interface with the Migrating Desktop has obtained one of the main goals of the project, a user-friendly and interactive Grid application.

Acknowledgement

This work was supported by the Crossgrid European Project (Ref: IST-2001-32243), the Ministry of Education and Science of Spain and FEDER funds of the European Union (Project TIN-2004-07797-C02) and by the Galician Government (Project PGIDT04TIC105004PR).

REFERENCES

- [1] Crossgrid home page. <http://www.crossgrid.org/>.
- [2] BELLEMAN, R. G.—SLOOT, P. M. A.: The Design of Dynamic Exploration Environments for Computational Steering Simulations. In Proceedings of the SGI Users' Conference 2000, Vol. 1, pp. 57–74. Academic Computer Centre CYFRONET AGH, 2000.

- [3] HLUCHÝ, L.—HABALA, O.—ŠIMO, B.—ASTALOŠ, J.—TRAN, V. D.—DOBRUCKÝ, M.: Problem Solving Environment for Flood Forecasting. In 1st International NAISO Symposium on Information Technologies in Environmental Engineering (ITEE 2003), 2003.
- [4] BETHKE, S.—CALVETTI, M.—HOFFMAN, H. F.—JACOBS, D.—KASEMANN, M.—LINGLIN, D.: Report of the Steering Group of the LHC Computing Review. In CERN/LHCC/2001-004, CERN/RRB-D 2001-3, 2001.
- [5] MARTÍN, M. J.—SINGH, D. E.—MOURIÑO, J. C.—RIVERA, F. F.—DOALLO, R.—BRUGUERA, J. D.: High Performance Air Pollution Modeling for a Power Plant Environment. *Parallel Computing*, Vol. 29, 2003, Nos. 11–12, pp. 1763–1790.
- [6] NIEZGODKA, M.—JAKUBIAK, B.: Numerical Weather Prediction System: Scientific and Operational Aspects. In *Proceedings of the III Symposium on Military Meteorology*, pp. 191–197, 1998.
- [7] CARMICHAEL, G. R.—PETERS, L. K.—SAYLOR, R. D.: The STEM-II Regional Scale Acid Deposition and Photochemical Oxidant Model – I. An Overview of Model Development and Applications. *Atmospheric Environment*, Vol. 25A, 1991, No. 10, pp. 2077–2090.
- [8] PACHECO, P. S.: *Parallel Programming with MPI*. Morgan Kaufman Publishers, Inc., 1997.
- [9] MARTÍN, M. J.—PARADA, M.—DOALLO, R.: High Performance Air Pollution Simulation Using OpenMP. *The Journal of Supercomputing*, Vol. 28, 2004, pp. 311–321.
- [10] MOURIÑO, J. C.—MARTÍN, M. J.—GONZÁLEZ, P.—BOULLÓN, M.—CABALEIRO, J. C.—PENA, T. F.—RIVERA, F. F.—DOALLO, R.: A Grid-enable Air Quality Simulation. In *Proceedings of the First European Across Grids Conference*. LNCS 2970, pp. 155–162, Springer-Verlag, 2004.
- [11] KRAMMER, B.—MÜLLER, M. S.—RESCH, M. M.: MPI Application Development Using the Analysis Tool MARMOT. In *Proceedings of the 4th International Conference in Computational Science (ICCS 2004)*, LNCS 3038, Vol. 3, 2004, pp. 464–471, Springer Verlag.
- [12] TSOLOUPAS, G.—DIKAIAKOS, M.: GridBench: A Tool for Benchmarking Grids. In *4th International Workshop on Grid Computing (Grid2003)*, pp. 60–67, 2003.
- [13] BLANCO, V.—GONZÁLEZ, P.—CABALEIRO, J. C.—HERAS, D.—PENA, T. F.—POMBO, J. J.—RIVERA, F. F.: Visualizing the Performance Prediction of Parallel Iterative Solvers. *Future Generation of Computer Systems*, Vol. 19, 2003, pp. 721–733.
- [14] FOSTER, I.—KESSELMAN, C.: GLOBUS: a Metacomputing Infrastructure Toolkit. *International Journal Supercomputing Applications*, pp. 115–128, 1997.
- [15] HEYMANN, MIQUEL E.—SENAR, A.—LUQUE, E.—LIVNY, M.: Adaptive Scheduling for Master-Worker Applications on the Computational Grid. In *Proceedings of the First IEEE/ACM International Workshop on Grid Computing (GRID 2000)*. LNCS 1971, pp. 214–227, Springer Verlag, 2000.
- [16] ENGH, D.—SMALLEN, S.—GIERALTOWSKI, J.—FANG, L.—GARDNER, R.—GANNON, D.—BRAMLEY, R.: GRAPPA: Grid Access Portal for Physics Applications. In *Proceedings of the 2003 Computing in High Energy and Nuclear Physics (CHEP03)*, 2003.

- [17] Java Classes: Now and Future. White Paper, last updated april 2003. <http://java.sun.com/products/jfc/whitepaper.html>.
- [18] KUPCZYK, M.—LICHWALA, R.—MEYER, N.—PALAK, B.—PLOCIENNIK, M.—STROINSKI, M.—WOLNIEWICZ, P.: The Migrating Desktop as a GUI Framework for the “Applications on Demand” Concept. In Proceedings of the 4th International Conference in Computational Science (ICCS 2004), LNCS 3036, Vol. 1, 2004, pp. 91–98, Springer Verlag.
- [19] Tool plug-in implementation. <http://ras.man.poznan.pl/crossgrid/plugins/tool/toolplugin.html>.



J. Carlos MOURIÑO received the Master Degree in Computer Engineering from the A Coruña University, Spain, in 2000, and the Ph.D. in Computer Architecture in 2006. Nowadays he is working at CESGA. His research interests include parallel computing, prediction and improvement of performance for applications, grid computing and fault-tolerance for message passing applications.



María J. MARTÍN is an Associate Professor of computer engineering at the University of A Coruña, Spain. She earned the B.Sc. (1993), M.Sc. (1994) and Ph.D. (1999) degrees in physics from the University of Santiago de Compostela, Spain. Her major research interests include parallel algorithms and applications, parallelizing compilers, grid computing and fault-tolerance for message passing applications.



Patricia GONZÁLEZ received the B.Sc. (1996), M.Sc. (1996) and Ph.D. (2001) degrees in physics from the University of Santiago de Compostela, Spain. Currently she is an Assistant Professor in the Department of Electronics and Systems at the University of A Coruña. Her research interests include parallel algorithms and numerical methods, prediction and improvement of performance for irregular codes, grid computing and fault-tolerance for message passing applications.



Ramón DOALLO received the B. Sc. and M. Sc. degrees in Physics from the University of Santiago de Compostela in 1987, and the Ph. D. in Physics in 1992. In 1990 he joined the faculty at the University of A Coruña, where he became full professor in 1999. He has extensively published in the areas of computer architecture, and parallel and distributed computing. Professor Doallo is a member of the IEEE Computer Society and ACM.