IL NUOVO CIMENTO Vol. 28 C, N. 2 Marzo-Aprile 2005
DOI 10.1393/ncc/i2005-10184-3

# Modeling and computational issues for air/water quality problems: A grid computing approach(*)

G. Giunta, R. Montella, P. Mariani and A. Riccio
*Dipartimento di Scienze Applicate, Università di Napoli "Parthenope" - Napoli, Italy*

(ricevuto l'8 Febbraio 2005; approvato il 28 Luglio 2005; pubblicato online il 23 Settembre 2005)

**Summary.** — In this paper we report some results on the design of a grid computing application in the field of environmental sciences. The case study concerns the integration of several independent computational models (weather, air and water quality and sea wave and currents) over a grid computing infrastructure. The aim of the project is the development of an efficient, high-performance and general-purpose distributed laboratory for environmental modeling. Expected end users may be either researchers in computational environmental sciences, whom a standard interface to access existing and distributed resources (database, models, visualization tools, parallel computers and virtual organization's collaborative community facilities) is provided, or ordinary citizens who can access the results of operational runs of the integrated system for obtaining short-scale forecasts using a web-based interface. Presently, the application continuously supplies real-time forecasts and scenario analysis for both weather and air quality and it is interactively serviceable via a dedicated web portal.

PACS `89.20.Ff` – Computer science and technology.
PACS `01.30.Cc` – Conference proceedings.

## 1. – Introduction

Grid computing has recently emerged as a key technology in computational sciences. Grid computing supports the sharing and coordinated use of multiple resources within the so-called Virtual Organizations, that means the creation, using several geographically distributed and independent components, of integrated virtual computational systems providing a desired application layer [1].

Many international projects, such as the Enabling Grids for E-science in Europe (EGEE) [2] and the TeraGrid project [3] (the United States multi-year effort to build and deploy the world's largest and fastest distributed infrastructure for open scientific

---

(*) Paper presented at CAPI 2004, 8° Workshop sul calcolo ad alte prestazioni in Italia, Milan, November 24-25, 2004.

research) are presently tackling the implementation of grid computing infrastructures for high-performance distributed data intensive applications.

Grid computing is expected to play a fundamental role in many fields of environmental sciences and policy, namely climate simulation, meteorology and oceanography, air/water quality, land management, real-time response to natural or man-made disasters [4]. The need of high performance computing in the environmental modeling fields has already been emphasized by the Japan's Earth Simulator project [5]. There is an emerging agreement among environmental scientists about the vision that the unprecedented power in computing and data accessibility and exchange, promised by a powerful computing infrastructure, will make effective the holistic understanding of the environment, *i.e.* the quantitative and integrated study of the complex, multiscale, nonlinear, strongly interrelated environmental phenomena as a whole.

This paper shows the results of a simple system integration approach in designing a grid computing application concerning the interoperability and the integration of complex numerical models for weather and air/water quality analysis and forecasting. The goal is an efficient multi-coupled modeling engine to perform either analysis of past environmental scenes or short-time forecasts.

The grid application has been implemented using the grid services technology (Globus Toolkit 3.x) based on OGSA and following the Open Grid Service Infrastructure. We are ready to port our work under the upcoming web service based Globus Toolkit 4 [6].

In sect. **2** we provide a brief description of the numerical models which our grid application is based on. Section **3** surveys the grid application design. In sect. **4** we describe the implementation of the grid application. Section **5** shows a simple but effective grid application example. Section **6** contains the conclusions.

## 2. – The numerical models of the grid application

The purpose of our grid application is to build and deploy a computational system that, through a standard and efficient framework, allows the coupling of existing, independent, distributed numerical models and databases, and is able to produce suitable weather and air/water quality analysis and forecasts and to distribute them across the Internet. The numerical models included in this first version of our grid application are: MM5 [7], PNAM [8], STdEM [9], POM [10] and WW3 [11].

MM5 is a numerical weather prediction system suitable for the study of a broad spectrum of theoretical and real-time applications, including both predictive and four-dimensional data assimilation of synoptic (monsoons, hurricanes, cyclones) and mesoscale (convective systems, land-sea breezes, mountain-valley circulations, urban heat island effects) weather phenomena. It includes multiple-nest capabilities, non hydrostatic dynamics and four-dimensional assimilation capabilities. We used the parallel distributed memory version (MMP) of the model.

PNAM and STdEM are a couple of models for the simulation of air quality phenomena on the meso and urban scale. STdEM is a GIS-based emission model, that is a model able to estimate the atmospheric emissions from several natural and anthropogenic sources (road and non-road traffic, biogenic, industrial combustion). PNAM is a 3D air quality model designed for parallel distributed memory computers. Starting from a predefined emission database and atmospheric dispersion pattern, PNAM solves an initial-boundary value problem for a system of three-dimensional advection-diffusion-reaction equations. These equations arise from an Eulerian description of the transport and the photochemical transformation of air pollutants, including emission and deposition phenomena. The reaction system is based on the LCC/SAPRC kinetic mechanism [12], regulating 107 gas-phase chemical reactions among 42 chemical species. Emission and deposition terms are included in the equations as ground-level boundary conditions; emissions are provided at fixed time intervals, while deposition terms are computed using the so-called three-resistance approach. Its degree of sophistication is comparable to others second-generation air quality systems [13]. The computational approach used to solve the initial-boundary value problem is based on a symmetric time-splitting technique that decouples advection and horizontal diffusion from vertical diffusion and chemistry. Moreover, it has been specifically developed for MIMD-distributed memory parallel machines. The parallel implementation is based on grid partitioning and dynamic load balancing techniques. It is written in Fortran 90 and is based on the parallel RSL [14] and FLIC [15] libraries to implement domain decomposition, data communication and dynamic load balancing. It has been routinely applied to the simulation of photosmog episodes over the Campania region (Southern Italy) and the Naples urban area [16, 17].

POM is a 3D primitive equations model for the description of a wide range of oceanic processes including estuarine and shelf circulation, open ocean studies and general circulation simulations. The model has been used to forecast the circulation in the Mediterranean sea and for water quality and ecological studies [18]. Using the same source approach and the RSL and FLIC libraries, we developed a parallel and nested implementation of the POM with the aim to achieve higher performance and better accuracy in our simulations [19].

WW3, WaveWatch III, is a third-generation wave model developed at NOAA/NCEP after the WAM wave model, as a further development of WaveWatch I (Delft University of Technology) and WaveWatch II (NASA, Goddard Space Flight Centre). The governing equations simulate variations in time and space of mean water depth and current; wave growth and decay produced by the surface wind, dissipation, and the bottom friction effects. The solution of the governing equations is based on a third order accurate numerical scheme. The breaking waves physics are not modeled, hence the model is only applicable outside of the surf zone and on large scales. Outputs from the model include significant wave height fields with associated wave directions and periods, spectral information about wave energy at the different wavelengths.

Coupling between models is achieved by forcing PNAM, POM and WW3 with the MM5 forecast fields (see fig. 1), however different models may be used, provided the interface between them.

## 3. – Grid application design

Several interesting services can be offered, by exploiting a coordinated and integrated use of those computational models, *i.e.* a) high-resolution and real-time weather forecasting; b) analysis of mesoscale weather phenomena; c) real-time air quality forecasting; d) assessment of the effectiveness of air pollutant emission abatement processes; e) real-time water quality forecasting; f) assessment of wave activity on coastal erosion. A typical coordinated application consists in the use of global meteorological analysis from a meteorological centre (ECMWF [20] or NCEP [21]); these global data are used to drive a mesoscale meteorological/oceanographic model, *i.e.* MM5 or POM, which in turn can drive an air quality model, namely PNAM, or a wave model, like WW3. Alternatively, the meteorological model can provide a high-resolution database to drive both the oceanographic and the air quality models. In this context, grid computing may represent an interesting opportunity towards the efficient and integrated use of these models over

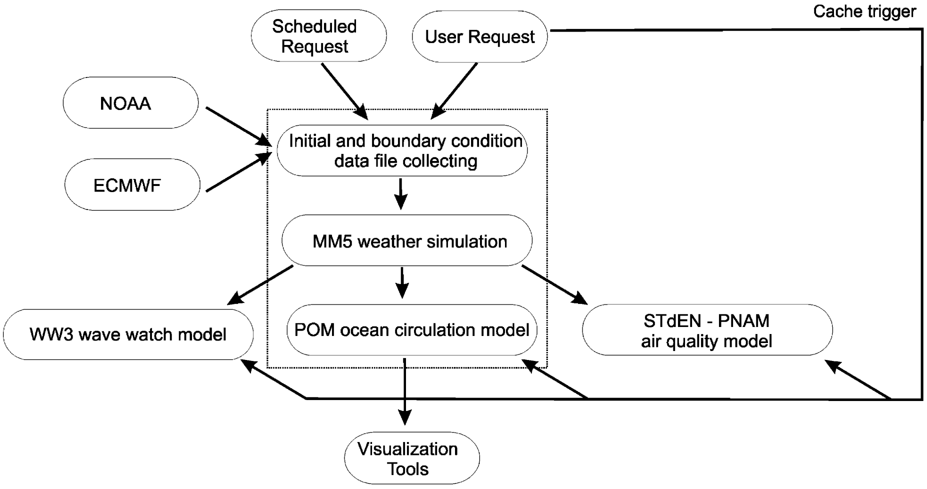


Fig. 1. – Building blocks of the coordinated use of our environmental models.

a geographically distributed computational environment (fig. 1).

Our prototype grid infrastructure has been developed from custom hardware resources. Two Linux clusters, two Linux workstations and one Windows server for web publication have been used. Following the Teragrid approach, we reduced the hardware lack of homogeneity installing the same software suites, operating systems and environmental software models and utilities over all computational elements.

The application models are written in Fortran77/90 (MM5, PNAM, POM, WW3) or C++ (STdEM). The High performance Portland Group (PGI) [22] and Intel Compiler (IC) suites [23] were installed. The Message Passing Interface (MPICH) [24] has been used on both workstations and parallel machines.

The Globus Toolkit 3.2 (GT3) has been used as the grid middleware software. Host and user certificates are needed and provided by a Certificate Authority (CA). A minimal version of this kind of software (SimpleCA) is included in the GT3 distribution and is suitable for research and testing purposes. The SimpleCA has a main role in the grid infrastructure because of the importance to store in a safe place a copy of all emitted certificates.

The GridFTP service was installed to provide an efficient file transfer mechanism over our grid infrastructure. Also, the Grid Resource Access Management (GRAM) was installed for job submitting purposes.

A grid application is a complex software system and the use of advanced software design techniques is needed. Particular attention to the analysis of system requirements and to the design of software architecture should be paid. The rules of the Rational Unified Process (RUP) approach [25] have been applied, using the UML software design techniques [26].

The RUP approach suggests relying on the "use case" diagrams for the formal definition of all functional requirements. We defined two main use cases: the operational behavior, when the application run is fired by the availability of new ECMWF/NCEP data, and the on-demand behavior, when the user asks for a particular data and model configuration. Each use case may correspond to a grid service.

For the grid application operational mode we devised the following use cases (we refer only to the MM5-PNAM coupling, since MM5-POM and MM5-WW3 coupling are similar):

1. the application subscripts as notification sink to the service which has in the charge the ECMWF/NCEP data download;
2. the service notifies about data availability sending the data URL in a push notification fashion;
3. the application receives the notification;
4. the application creates an instance of the MM5 service, subscripts as notification sink for computation progress information and performs all stage in operations;
5. MM5 notifies to the application when it finished;
6. the application receives the notification about the MM5 termination and creates an instance of the publishing service, submitting the related job;
7. the application creates an instance of the PNAM service, subscripts as notification sink for progress information and runs the air quality model;
8. the PNAM service notifies when it finished;
9. the application receives the notification, creates an instance of the publishing service and submits the related job, creating PNAM outputs.

Use cases could be more or less detailed with regards to design abstraction level; for example, in the use case number 4, the MM5 service invocation could be detailed with each invocation of MM5 pre- and post-processing modules. The on-demand grid application mode use case is pretty similar to the previous one, except steps 1 to 3:

1. the user selects simulation parameters, date and time steps and model coupling configuration;
2. the application creates an instance of the job satisfying the user requests;
3. the service notifies about data availability, *e.g.*, initial values and boundary conditions used by the MM5 model, as soon as possible. If requested data are not available on the local cache, a download operation from ECMWF/NCEP is performed using access credentials.

Job submission is performed through the MasterManagedJobFactoryService and other GRAM features based over the Resource Specification Language or, alternatively, through a custom grid service implementing service data element, notification and advanced job monitoring features.

Each use case is encapsulated into a runnable Java class and then wrap converted into a grid service. Each grid service exposes one or more remote methods; the client can invoke any one of these methods using the GT3 client stub classes so that the grid service is seen by the client as a local software component.

We think that such “componentization approach” is the best and the only viable one, at least in the case the software to be grid enabled is a sort of legacy type.

Our grid application uses all transient grid services implemented with the Factory Design Pattern characteristics of GT3. Each service exposes its features in a standard and well documented way. Remote methods, service data elements and notifications were strategically planned in the design step, in order to offer a full interoperable connection system among computational components.

*MeteoDataProviderService*: implements all needed functionalities to retrieve initial data and boundary conditions for the MM5 model. This service manages a local data cache and the remote data download on demand. Presently, the service downloads data only from NCEP; in a future implementation the service will work with ECMWF data too. The service is a notification source for data availability.

*TerrainService*: this service is the MM5 TERRAIN module grid wrap dedicated to the production of spatial terrestrial data, *e.g.*, terrain elevation, landuse and sea/land map data. The grid application implementation creates a new set of dynamically placed nested domains used by the RegridService.

*RegridService*: this service encapsulates the behavior of both MM5 modules called Pregrid and Regridder. The service performs all pre-processing stuff related with space and temporal interpolation of NCEP/ECMWF global data in order to provide the initial and boundary conditions for the mesoscale weather simulations. Since an invocation of Regrid for each nested domain is needed and there is no communication between domains, this service can run in parallel for each nested domain.

*InterpfService*: as the RegridService, this service exposes the Interpf MM5 module features in a grid service fashion. Its invocation is needed to prepare regridded data to be used by the MM5 model, and eventually performing a data assimilation preprocessing work. As in the RegridService case, an invocation of Interpf is needed for each nested domains with no communication between domains, so that each Regrid-Interpf pair for each nested domain can be run in parallel with overall performance improvement;

*MM5Service*: this service encapsulates the MM5 model in both sequential and parallel version. The user does not take care about the MM5 parallel or sequential version, because the choice is automatically performed by the resource broker. The service is a notification source for the computational progress status.

*STdEMService, PNAMService, POMService, WW3Service*: these grid services implement their wrap models in the same way as the MM5Service using service data element and notification sources. This kind of approach allows to expand the number of environmental models available on the grid simply implementing a new wrap.

*GrADSService*: The multiplatform Grid Analysis and Display visualization System (GrADS [27]) was grid enabled to achieve the best performance in model results rendering, developing a simple grid wrap over an off-the-shelf application.

## 4. – The grid application implementation

Once all services and their interfaces are well defined, a grid application can be formally defined as a set of tools implemented with the aim of solving a numerical problem through a computational grid infrastructure [28].

Our grid application uses a complex network workflow representing job submission in both serial and parallel approaches. We developed a job flow scheduler (JFS) in order to provide a simple way to build computational experiments over the grid. The user can build his experiments using a custom Job Flow Description Language (JFDL) based on a XML evolution of the Condor Job Description Language [29]. The JFDL integrates both Globus Toolkit Gram RSL job specifications and job relationships descriptions. For example, in a very simple grid application, performing initial conditions data download, weather forecast and ocean circulation forecast, the JFDL file appears as follows:

```
<jfdl:jfs project="verySimpleWeatherForecastApplication">
  <!- Job Definitions -->
  <jfdl:jobs>
    <jfdl:job name="downloadConditions"
         target="dgric.uniparthenope.it"
         rsl="downloadConditions.xml"/>

    <jfdl:job name="runMM5"
         target="dgbeobi.uniparthenope.it"
         rsl="runMM5.xml"/>

    <jfdl:job name="runPOM"
         target="dgbeobe.uniparthenope.it"
         rsl="runPOM.xml"/>
   </jfdl:jobs>

   <!-Job nodes and relationship definitions-->
   <jfdl:nodes>
     <jfdl:node job="downloadConditions">
       <jfdl:next>runMM5</jfdl:next>
     </jfdl:node>
     <jfdl:node job="runMM5">
       <jfdl:prev>downloadConditions</jfdl:prev>
       <jfdl:next>runPOM</jfdl:next>
     </jfdl:node>
     <jfdl:node job="runPOM">
       <jfdl:prev>runMM5</jfdl:prev>
     </jfdl:node>
   </jfdl:nodes>
</jfdl:jfs>
```

The JFDL file is divided into two parts. In the first part, tagged by the ⟨`jfdl:jobs`⟩ element, each job is identified by a symbolic name. The job definition contains the name or the IP address of the computational element; in this example it is statically assigned, but it may be also dynamically referenced by a resource broker (RB). A late binding resource broker mechanism was also developed using a custom implementation of the Globus MasterManagedJobService. The job specification is performed using the Globus Gram RSL description file.

The ⟨`jfdl:nodes`⟩ element describes the job flow graph. Each node is defined using the ⟨`jfdl:node`⟩ element. The relationships between nodes are described by the ⟨`jfdl:prev`⟩ and ⟨`jfdl:next`⟩ elements: each job contained in the ⟨`jfdl:prev`⟩ element must be successfully submitted and completed before the current job is submitted. Moreover, each job in the ⟨`jfdl:next`⟩ element is submitted in a non-blocking way after the current job is successfully completed.

The actual grid application JFDL file is more complex than the simple one reported, but shares the same kind of approach. Other loosely coupled environmental models can interact with the application workflow in a similar fashion.

The grid application also provides for a complete job monitoring and management tool based on the Globus Toolkit Gram batch execute feature. This makes straightforward a grid portal environment integration based on MyProxy temporary credential delegation [30]. The job manager utilize a service handler (GSH) to keep track of running jobs and invoking a low-level api.

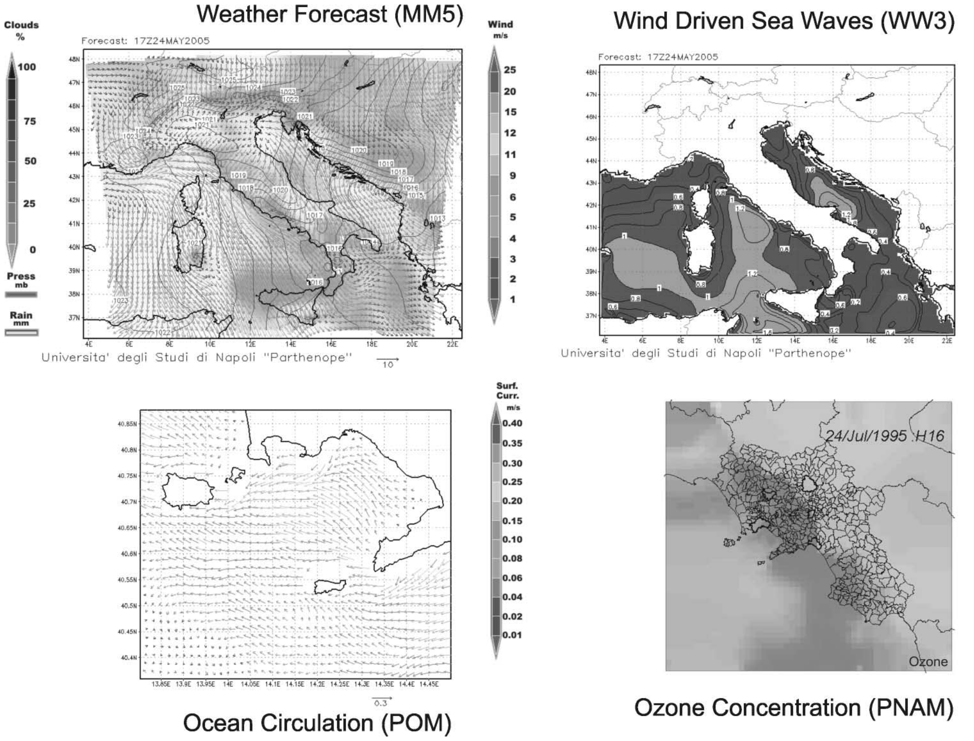
## 5. – An example of grid application job

The grid application implements the grid enabled version of the MM5, POM, WW3, STdEM, PNAM and GrADS models. The model code is considered as legacy so the black box paradigm is used. Details about the running environment, such as the scratch directory creation and deletion, staging operation, parameters management and standard streams, are in charge of the grid service developer.

The grid application job is an example of operational system producing short time weather forecasts using the NCEP global model data to initialize the MM5 model. Using a nested grid system the computational resolution is dynamically downscaled from a cell size of 81 km over the Mediterranean Europe, down to 3 km over the Naples' Bay. MM5 output forces the air quality model (via the wind speed, pressure, relative humidity, precipitated water and turbulence parameters), and the POM model (via the wind stress and radiation fluxes). The STdEM, using EMEP (`http://www.emep.int`) and local emission data, supplies emissions of atmospheric pollutants to the air quality model. Lateral boundary conditions and initial conditions for the POM model are obtained by spatial and temporal interpolation of monthly climatologically dataset for the Mediterranean Sea (MEDAR/MEDATLAS 2002).

The MM5 model can be seen as a grid service producer, while all other models are grid service consumers. Grid service consuming is based on the use of stub classes performing any kind of access and invocation to a local object marshaled on the corresponding remote service. This kind of approach hides from the programmer all the complexity related with the SOAP messages exchange requests and responses between the server and the client. The grid application named ModelClient uses all grid services throw the factory paradigm so multiple service instances could be used independently.

All grid components are based on "off the shelf" computing hardware, based mainly on Beowulf PC clusters running the Linux operating system. Two parallel machines are dedicated to model execution: the first one is composed by 8+1 Pentium IV at 1.5 GHz, each with 256MB of RAM menory and 40 GB hard disk connected by a Fast Ethernet switch connection; the second one is composed by 8+1 Pentium IV at 3.0 GHz HT, 512MB of RAM and 160 GB hard disk connected by a Giga Ethernet switch. The first Beowulf PC cluster is used to run the parallel nested POM and STdEM/PNAM models, while the second one is used to run the MM5 model and the WW3 model. Because WW3, POM and STdEM running can be done in a parallel fashion, the total simulation time is given by the computing time of each application job flow sequential branch. Currently, the MM5 run takes about 30 minutes to complete and three days ahead forecasts are available at about 8:00 LST every day, depending on the availability of NCEP data. A comparable amount of time is needed by the POM, PNAM and WW3 models. Figure 2 shows an example of real-time weather circulation results published on the web (`http://www.dsa.uniparthenope.it`). Presently, data are updated with a



Weather Forecast (MM5)

Wind Driven Sea Waves (WW3)

Ocean Circulation (POM)

Ozone Concentration (PNAM)

Fig. 2. – An example of results published on the web by the grid application.

daily frequency. A systematic comparison between forecasts and model output, for model verification purposes, is currently under way.

## 6. – Conclusions

The grid computing approach applied to advanced environmental modeling will be widely diffused and pervasive in the near future. We developed a grid computing application dedicated to air and water quality and to other environmental issues which is based on an experimental grid infrastructure and provides operational and on demand features. The application supplies a continuous service which can produce forecasts and scenario analysis for both weather and air quality and is interactively serviceable via a dedicated web portal.

All adopted design solutions make straightforward the expansion of the computational and storage elements of the grid infrastructure, since the grid enabled software can easily handle the installation of new nodes.

## REFERENCES

[1] Foster I., Kesselman C. and Tuecke S., *Int. J. High Performance Computing Appl.*, **15** (2001) 200.

[2] *The EGEE Project: Enabling Grids for E-science in Europe* `http://public.eu-egee.org/`

[3] *The TeraGrid Project, National Science Foundation* `http://www.teragrid.org/`

[4] Johnston W., in *The Grid: Blueprint for a New Computing Infrastructure*, edited by Foster I. and Kesselman C. (Morgan Kaufmann) 1999, pp. 75-103.

[5] *Earth Simulator Center - Earth Simulator.* `http://www.es.jamstec.go.jp/esc/eng/`

[6] *The Globus Toolkit, The Globus Alliance*, `http://www.globus.org/`

[7] Grell G., Dudhia J. and STauffer D., NCAR/TN-398+STR, NCAR Technical Report (1995).

[8] Barone G., D'Ambra P., Di Serafino D., Giunta G., Riccio A. and Murli A., *J. Environ. Manag. Health*, **10** (2000) 209.

[9] Montella R. and Riccio A., *Ingegneria Ambientale*, **11** (2002) 605. In Italian.

[10] Blumberg A. F. and Mellor G. L., in *Three-Dimensional Coastal Ocean Models*, edited by Heaps N. S. (American Geophysical Union, Washington, DC) 1987.

[11] Tolman H. L., NOAA/NWS/NCEP/OMB 166, NOAA Technical Note (1996).

[12] Lurmann F.W., Carter W. P. L. and Coyner L. A., Tech. Rep. EPA/600/3-87/014, U.S. Environmental Protection Agency and Statewide Air Pollution Research Center (1987).

[13] Peters L. K., Berkowitz C. M., G. R., Carmichael R. C., Easter G., Fairweather S. J., Ghan J. M., Hales L. R., Leung W. R., Pennell F. A., Potra R. D., Saylor and Tsang T.T., *Atmos. Environ.*, **29** (1995) 189.

[14] Michalakes J., ANL/MCS-P663-0597 Technical Report (1997).

[15] Michalakes J., ANL/MCS-TM-223 Technical Report (1997).

[16] Barone G., D'Ambra P., Di Serafino D., Giunta G., Riccio A. and Murli A., in *Large Scale Computations in Air Pollution Modelling* edited by Z. Zlatev *et al.* (Kluwer Academic Press) 1999, pp. 39-52.

[17] Barone G., D'Ambra P., Di Serafino D., Giunta G., Riccio A. and Murli A., *Environ. Modeling & Softare*, **15** (2000) 503.

[18] Zavatarelli M. and Mellor G. L., *J. Phys. Oceanogr.*, **25** (1995) 1384.

[19] Bortoluzzi A., Falcione V., Mariani P. and Riccio A., IMFA/TR 6/04 Technical Report. In Italian.

[20] European Centre for Medium-Range Weather Forecasts, `http://www.ecmwf.int/`

[21] National Centers for Environmental Prediction, `http://www.ncep.noaa.gov/`

[22] High Performance Compilers PGI, Portland Group, `http://www.pgroup.com/`

[23] Intel Compilers Suite, Intel Inc., `http://www.intel.com/software/products/compilers/`

[24] The Message Passing Interface (MPI) standard, `http://www-unix.mcs.anl.gov/mpi/`

[25] KRUCHTEN P.: *The Rational Unified Process: An Introduction* (Addison-Wesley) 2000.

[26] PRESSMAN R. S.: *Software Engineering Principles* (McGraw-Hill) 2000.

[27] *Grid Analysis and Display System (GrADS)*, Institute of Global Environment and Society - IGES, `http://grads.iges.org/grads/grads.html`

[28] Ferriera L., Thakore A., Brown M., Lucchese F., Ruobo H., Lin L., Manesco P., Mausolf J., Momtaheni N., Subbian K. and Hernandez O., *Grid Services Programming and Application Enablement* (IBM Redbooks) 2004.

[29] The Condor Project, University of Wisconsin, `http://www.cs.wisc.edu/condor`

[30] MyProxy Online Credential Repository, Grid Computing at NCSA, University of Illinois, `http://grid.ncsa.uiuc.edu/myproxy/`