Earth Sci Inform (2009) 2:75–82
DOI 10.1007/s12145-008-0018-z

METHODOLOGY ARTICLE

Execution management in the GRID, for sensitivity studies of global climate simulations

V. Fernández-Quiruelas · J. Fernández · C. Baeza · A. S. Cofiño · J. M. Gutiérrez

Received: 17 September 2008 / Accepted: 18 December 2008 / Published online: 17 January 2009

Abstract Recent trends in climate modeling find in GRID computing a powerful way to achieve results by sharing geographically distributed computing and storage resources. In particular, ensemble prediction experiments are based on the generation of multiple model simulations to explore, statistically, the existing uncertainties in weather and climate forecast. In this paper, we present a GRID application consisting of a state-of-the-art climate model. The main goal of the application is to provide a tool that can be used by a climate researcher to run ensemble-based predictions on the GRID for sensitivity studies. One of the main duties of this tool is the management of a workflow involving long-term jobs and data management in a user-friendly way. In this paper we show that, due to weaknesses of current GRID middleware, this management is complex task. Those weaknesses made necessary the development of a robust workflow adapted to the requirements of the climate application. As an illustrative scientific challenge, the application is applied to study the El Niño phenomenon, by simulating an El Niño year with different forcing conditions and analyzing the precipitation response over south-American countries subject to flooding risk.

Communicated by: H. A. Babaie

V. Fernández-Quiruelas · J. Fernández · A. S. Cofiño (✉)
Department of Applied Mathematics and Computer Sciences,
Universidad de Cantabria,
Santander, Spain
e-mail: antonio.cofino@unican.es

C. Baeza
Center for Mathematical Modeling, Universidad de Chile,
Santiago, Chile

J. M. Gutiérrez
Instituto de Física de Cantabria, CSIC-UC,
Santander, Spain

Keywords CAM model · Climate models · El Niño phenomenon · GRID computing · Workflow management

Introduction

The EU-funded project EELA (E-Infrastructure shared between Europe and Latin America) aims at bringing the e-Infrastructures of Latin American countries to the level of those established in Europe, identifying and promoting a sustainable framework for e-Science (http://www.eu-eela.org). The present paper describes the new developments achieved as a result of porting a climate application to the GRID under the EELA framework with the goal of analyzing el Niño phenomenon, which is a key factor for Latin-American (LA) climate prediction. El Niño has a special interest due to its direct effect in the Pacific coast of South America and, in particular, in Peru and Chile (EELA LA partners).

For this reason, the climate applications in EELA were designed around this phenomenon with the main objective of developing a simulation and analysis tool especially useful for LA partners.

GRID technologies emerged in the 1990s as a way to share computer resources and other scientific equipment across geographically distributed locations in a user-transparent way (Foster and Kesselman 1999). By sharing computer resources it is meant not only to share their storage capacity, but also the computer power, which would be used to run applications. The user transparency relies on what is referred to as "middleware", a software layer between the applications and the GRID infrastructure.

A number of research and commercial projects have developed different middleware solutions and applications

(e.g. the EGEE project (www.eu-egee.org) is the reference in GRID development in Europe). New applications ported to the GRID demand new services which are not always available in the existing middleware.

In this paper, we present a new paradigmatic example on the area of climate simulation which demands solutions in terms of, e.g., job duration and workflow management.

We selected a Global Circulation Model as the first application to be ported to the GRID, since any further simulation or analysis step would require a global simulation as starting point. Particular features of the GCM posed specific problems for the GRID, such as experiments lasting beyond proxy certificates lifetime, control of jobs, etc. Using the existing middleware solutions we created a new application developing extra middleware to run the GCM in the GRID with a specific workflow, solving most of the problems encountered.

The paper is organized as follows. A brief overview of the climate models is given next, followed by a summary of the specific benefits obtained from the GRID and also the requirements posed on the GRID. Then, the GRID-CAM application is described introducing the existing middleware solutions used and the new developments performed to achieve the deployment of the climate model on the GRID. Finally, some scientific results achieved with GRID-CAM are presented.

Climate modeling

Dynamical climate models are mathematical models that numerically solve the nonlinear equations governing the atmosphere on a global lattice with horizontal resolutions ranging from 50 to 300 km, depending on the application. These models require a set of initial conditions (values of climate variables—wind, pressure, temperature, etc,—on a lattice of points at the starting time) to propagate the solution forward in time.

In order to analyze the atmospheric part of the global climate system, we selected the CAM model (Community Atmosphere Model), which is the latest in a series of global atmosphere models developed at NCAR for the weather and climate research communities (Collins et al. 2004). CAM can be run for short (hours, days) or long periods of time (decades, centuries) to investigate the present, past or future climate variability. A great percentage of the total variability (70–90%) of global climate is obtained when an atmospheric model is forced with the oceanic fluxes; say the sea surface temperature and ice cover. As a first step, we ported the CAM model, which is an atmosphere-only model and requires lower boundary conditions at the surface (sea surface temperature). This model enables a wide range of experiments.

The port of a fully coupled model (such as the CCSM model which comprises CAM as the atmospheric part) is left as a natural future work.

The model can be run either in parallel (using MPI) or as a single process. The single-process version has been deployed and run in the EELA testbed with T42 resolution: 128 (longitude)×64 (latitude) and 27 vertical levels, i.e. 221184 points per time step. The model produces 32 3D and 56 2D variables over the lattice. The simulation of a year takes approximately 48 hours of wall clock time in a 3 GHz Intel Pentium D processor, while 100 years would take around 7 months of wall clock time. The model produces 197 MB per time step, i.e. more than 720 GB per century. These figures increase when running the model at a more state-of-the-art resolution. T42 was state-of-the-art one decade ago and is only in use for very long simulations (centuries), but we used it for the development process.

The application we designed aims to perform sensitivity experiments by running an ensemble of CAM simulations with perturbed the sea surface temperatures as boundary conditions.

Benefits of GRID

Climate models are complicated computer programs which require large amounts of CPU power. Most of them are parallelized. However, the GRID cannot make the most of this kind of parallelism, since the latency across geographically distributed computers would render the program completely inefficient. Apart from computer parallelism, climate science is recently making use of a large number of simulations, referred to as an "ensemble", of the same phenomenon in order to assess the uncertainty inherent to the simulation (Hagedorn et al. 2005; Palmer 2002). Ensembles of simulations with varying parameters are also used for sensitivity experiments and many other applications. Each simulation in an ensemble is independent of the others and can be run asynchronously. This kind of parametric jobs is well suited for the GRID, since each simulation can be carried out in different nodes and the results are made available as a uniform data set in the Logical File Catalogue (LFC), ready to be analyzed.

Unlike volunteer computing projects, such as *climateprediction.net* (Allen 1999) where the GCM needs to be simplified and most of the results thrown away to avoid the overloading of the volunteer hosts, the GRID allows running a full state-of-the-art model and store the regular output information.

Requirements and workflow management

Nowadays, it is uncommon the use of GRID computing to run long-term jobs, due to the high rate of job failure and

the CPU-time limitations for the jobs on the local management system (typically only jobs lasting less than 48 h are allowed). These problems become critical for long simulations such as those performed with climate models and other Earth Science applications. Thus, unlike many other applications ported to GRID, earth science applications need to make use of advanced techniques in workflow management. In particular, the climate application described in this paper has the following requirements:

- Failure aware: Due to the nature of GRID there are several reasons which may cause job failures in the testbed, including heterogeneity of resources, CPU-time limited queues, etc.
- Checkpointing for restart: The complexity of the climate model runs may require jobs to be restarted in a different working node due, for instance, to the excessive duration of the job. Unlike other applications, it is not affordable to throw away an interrupted simulation.
- Monitoring: Since the climate simulations last for a long time, we need to know what is happening with the simulation once it has been sent to the testbed: whether the model is running or not, which time step is being calculated, which files have been uploaded to Storage Elements, which one is the last restarting point, etc.
- Data and Metadata storage: The goal of our application is the generation of output information that can be easily accessed by users, so data and metadata should be stored in an appropriate form.

The above requirements made necessary the development of a goal-oriented workflow manager in order to run the experiments with a minimum of human intervention. Therefore, we developed the GRID-CAM application which is a "GRID workflow management layer for CAM simulation".

The requirements of the climate model described in this section are only an example of the needs of many other Earth Science applications.

The GRID-CAM application

In this section we briefly introduce and define the different components involved in a typical climate simulation on the GRID. We define an *experiment* as an ensemble of simulations (parametric jobs) designed to answer some scientific question (a single execution is the simplest experiment); each of the ensemble executions is called a *realization* and requires a set of input data to run the model in the prescribed simulation period (typically several years). A particular type of experiment is that related to climate sensitivity studies. In this case, the different sets of input data are obtained from a single one including certain user-defined perturbations to form the ensemble (perturbed initial or boundary conditions, perturbed parameters, perturbed radiative forcing, etc.).

The lowest level component of our application is a *job*. This component matches with a standard GRID job and cannot be related one to one with a realization since realizations cannot be guaranteed to finish in a single job. In general, a realization requires several jobs to be completed, each one restarted from the previous one. As the job is running, the model generates information (files and metadata) that has to be available from every other component of the GRID: restart files (for failure recovering), current simulation time step, number of restarts, job id (for monitoring purposes), statistical information, output data, etc. Hereinafter, all the data and metadata generated by the models will be referred to as output information.

Figure 1 shows a scheme illustrating the relationships between the main components of the application.

Therefore, climate simulation on the GRID requires the management of a complex workflow formed by experiments composed of realizations split across jobs. This workflow is not trivially managed by the currently available GRID middleware, so a new layer is necessary for a proper execution of climate simulations.

Grid middleware used

gLite middleware

The gLite middleware is an integrated set of components designed to enable resource sharing in GRID (Burke et al. 2007).The core components of the gLite architecture are the following:

- User Interface (UI): It is the access point to the GRID.
- Computer Element (CE): A set of computing resources localized at a site (i.e. a cluster, a computing farm).
- Worker node (WN): The cluster nodes where the jobs are run.
- Storage Element (SE): Separate service dedicated to store files.

The Logical file catalog (LFC) is a GRID secure catalog containing logical to physical file mappings. The primary function of the LFC is to provide central registration of data files distributed amongst the various Storage Elements. On the other hand, AMGA (Koblitz et al. 2007) is the gLite Metadata Catalogue, and we use it just as a classical GRID-enabled database where we store all the data and metadata information required by the application and the user.

We also used GridWay (http://www.gridway.org/), which is a GRID meta-scheduler which provides a scheduling

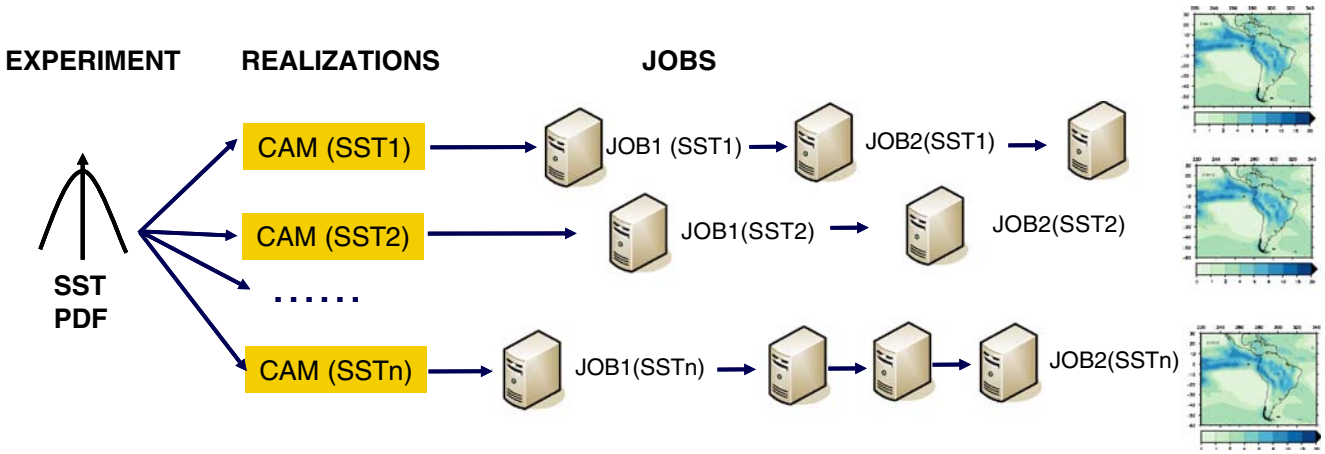

Fig. 1 GRID-CAM Application main components. In order to run an experiment (ensemble of simulations) we will create several realizations that will be simulated by several jobs in cascade

framework similar to that found on local Resource Management systems, supporting resource accounting, fault detection and recovery and the definition of state-of-the-art scheduling policies.

In addition to these existing middleware products, some GRID developments were necessary in order to deploy the climate application and to develop the appropriate workflow elements. These new components are described in the following sections.

The Grid Enabling Layer (GEL)

Climate models are mature applications with thousands of lines of code (usually Fortran). We introduced small modifications to the code to perform system calls to specific applications which are in charge of interacting with the GRID on behalf of the climate model. To this aim, we developed a new software layer, referred to as GRID Enabling Layer (GEL), which provides the model with the ability to interact with the GRID. The slightly modified source code of the model plus its GEL conform a fully featured GRID application. Since climate models are developed by external institutions, this approach is the best suited to keep up with the most recent updates with the least effort: only the small modifications to interact with the GEL need to be introduced at key points of any new release.

The GEL provides the following capabilities:

- Realization monitoring: Since our simulations last for a long time, we need to know their status once they have been sent to the testbed: If the model is preparing the WN or running, which step of time is calculating, which files has uploaded to SE-LFC, which is the last restart point, etc. This is analyzed in detail in the next section.
- Restart management: Each time CAM finishes simulating a time-step, the GEL uploads the restart files to the nearest SE and registers them in the LFC. It also publishes the restart field associated to this experiment in the AMGA database. This way, if the job fails and the realization is rescheduled to another WN, it will continue calculating from this time-step on.
- Data and Metadata management: In order to store all the output and restart information generated by the model, we need that the events and files are permanently registered in a place accessible from any component of the GRID (AMGA and LFC-SE).

The above issues were solved by introducing Fortran system calls at 4 specific points of the CAM source code. These calls execute the GEL scripts which carry out the previously mentioned tasks. The slight modifications to the Fortran source code also allow us use fast-development programming languages for the GEL scripts (we used a mixture of shell, Perl and Python).

Workflow management layer (WML)

As we pointed out in "Requirements and workflow management", there are several reasons which may break the flow of the job. In fact, during the first stages of the porting of the application, we did not account for all these possible errors. The result was that very few of our simulations finished.

To better understand the WML, we describe next the steps a GRID job goes through in our system since a user sends it to the scheduler until it is finished.

When a user submits a job from the UI, the scheduler sends it to a CE. Once there, the job takes the pending status while the CE finds a WN for the job in the local site.

If everything goes fine, the job starts its execution in the WN and finishes with the SUC final state (see Fig. 2).

Unfortunately, there are several situations where the job does not reach this status. In order to create a robust workflow, we identified the failure points that were not managed by the gLite middleware and we designed a workflow able to overcome the following situations:

- Due to misconfigurations in local sites, the CE is publishing free WNs when, in fact, there are not. Then, the job can be in pending status in the CE indefinitely. To overcome this, if a job remains for a long time (currently set to more than 5 min) in pending status in the CE it passes to an error state (SUSP) and the job will be rescheduled.
- When a job is running in the WN, it can crash as normal executions crash in a cluster (Core dumps, power failures, …). In these cases, the job is finished with the ERROR state and rescheduled to another CE.
- Even if the job is in active state in the CE (it is supposed to be running in the WN), sometimes the WN is not executing it. If after being in active state in the CE for more than 5 min, the job has not been started, it will be passed to the MID.ERR state and rescheduled.
- When the WNs are not well configured (e.g. cannot access the LFC, SE or AMGA), the job is passed to MID.ERR state and rescheduled to a different site.
- Some clusters do not distribute the load in a proper way among the WNs. When a job detects that the CPU assigned to it is less than a given threshold (currently set to 30%), it is passed to the PERF state and is rescheduled.
- The local queue kills our job when it has reached the maximum allowed time in the cluster (usually 48 hours). In this situation, the job takes the WALL (local queue walltime) status and is rescheduled.

All these final status (SUC, SUSP, ERR, MID.ERR, PERF, WALL) are stored in the AMGA database for each job for further analysis.

After analyzing several job managers, we found that GridWay meta-scheduler was the one that best fulfilled our requirements, since GridWay detects job failures for all of the problems mentioned before, and it is able to reschedule the failed jobs to another cluster. Moreover, once the re-scheduled job starts to run in the WN, a component of the WML developed queries the AMGA database to find the latest restart files for this realization in order to continue the simulation started by the previous job. We have also adopted an additional monitoring feature provided by GridWay: while the job is running in the WN, a monitor script (running also in the WN) checks the status of the job. This monitor can copy the output and error files of our job to the UI with a given frequency. In this way, from the UI we can accurately determine any failure in each of our realizations. When an ensemble of simulations is sent to the GRID, each realization of the ensemble is converted to a GridWay job that is sent to the scheduler. When GridWay receives the jobs, it searches the WN better suited to our application needs and chooses the best among them. To do so, it uses a powerful scheduling policy that takes into account the user requested requirements (memory, CPU, etc.) and a heuristic scheduling based on the jobs sent in the past. For instance, if all jobs sent to a CE failed, GridWay will not try to send jobs to that site for a long period.

Finally, in order to manage the issue of the expiration of the proxy, which affects every job lasting longer than 48 h, we used the *myproxy* credential management system as a provisional solution.

Monitoring with AMGA

The AMGA database has two different tasks in the application. On one hand, it is used to store the information generated by the experiments executed in the GRID. On the other hand, it is used for monitoring purposes, storing all the status information about each of the jobs as metadata information. The tables and relationships used by GRID-CAM are shown in the Fig. 3.

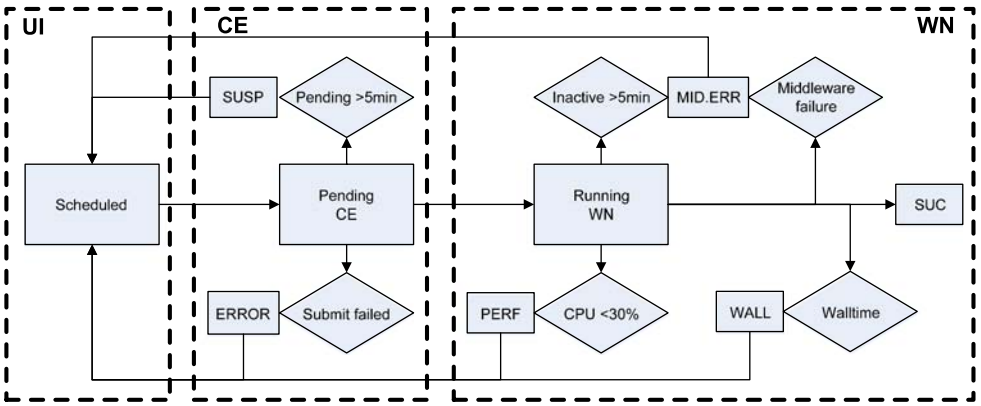

Fig. 2 Flow chart with conditions (*diamonds*) for possible states (*squares*) and state-transitions (*arrows*) of a job. The elements are grouped by GRID component (*dashed squares*)

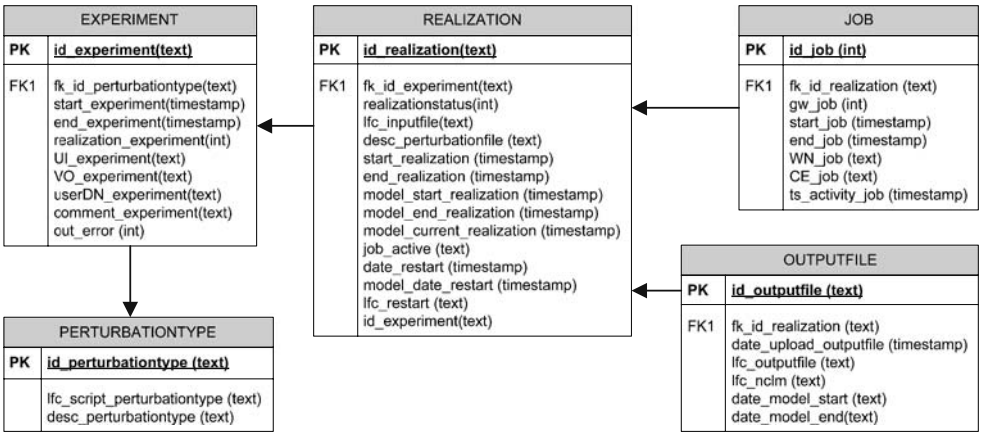

Fig. 3 Structure of the AMGA database used to store metadata and status information for the GRID-CAM application

Some of them are also relevant to the workflow, as described below:

- Experiment: When preparing the experiment, this table is filled with the type of perturbation used (multiplicative, random, etc), the number of realizations and a description and simulation dates of start and end.
- Realization: Each realization can be executed in many different nodes. This table keeps track of current time step, restart files, id of the current job executing the realization, etc.
- Job: This table is used to keep track of the different jobs used in an experiment. It stores the timing information, the WN and the realization it contributed to. Most of this information is stored for statistical purposes.
- Outputfile: Each realization generates a number of files as it runs. This table stores metadata and access information for the files stored in the catalog. This speeds up the data discovery process.

Scientific experiment: design and results

In our current status, the application allows us to run a variety of scientific experiments using CAM. As a first example, we ran an experiment of 50 CAM realizations with boundary conditions (sea surface temperature) as observed during the period from January 1997 until March 1998 (15 months). This period includes the strongest El Niño event observed to date. We investigated the sensitivity of the precipitation over south-American countries to modifications of the sea surface temperature (SST) by means of an ensemble of CAM realizations.

To create the ensemble, we perturbed the observed SST by adding a given spatial pattern scaled by a random number. The spatial pattern used to perturb the SST is shown in Fig. 4. It is the SST anomaly of the last two strongest El Niño events with respect to the climatological mean. For the 1997 event, this anomaly reached a maximum value of 2.5 K. We normalized this anomalous pattern by dividing by this maximum value. Then, we selected random scale factors in the range from −2.5 (normal conditions, since the perturbation is opposite and of similar intensity to the anomaly generated by El Niño) to 2.5 (double anomaly than in the 1997 event). A zero value, mean a forcing similar to that observed during 1997 year.

The initial conditions for the experiment correspond to a previous CAM run which started from climatological conditions on 1st January 1990, but was forced by the observed SST for one decade. The atmospheric and soil conditions on the 1st January 1997 from this run were used as initial conditions for the perturbed SST experiment.

The 10-year simulation showed a slight but clear precipitation response to El Niño conditions. However, the perturbed runs from our Grid experiment did not clearly

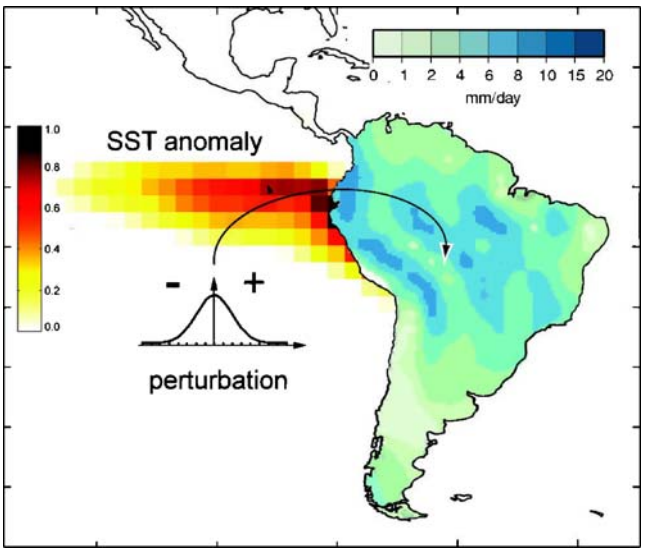

Fig. 4 Scheme of the sensitivity experiment. The normalized pattern over the sea is scaled with a random number and added as a perturbation to the observed SST. The resulting SST is used as lower boundary condition for a CAM realization, giving rise to a precipitation distribution over land

respond to higher than observed SST anomalies (Fig. 5a). Figure 5 represents the mean precipitation for the period October through January, when the most intense anomalies are registered. When the precipitation at a single sensitive point is analyzed against the scale of the perturbation applied (Fig. 5a), the response to increased El Niño conditions is clear until a value 0 of the perturbation scale (Niño'97 conditions) but seems to remain stable to stronger perturbations. These can also be seen in Fig. 5b and c where the whole South American distribution of precipitation is presented for a Niño'97-like realization and a stronger (double) anomaly realization. These are preliminary results which still need to be analyzed in a more systematic manner, using a larger number of simulations for statistical significance, and probably require the use of a regional model to generate more reasonably the precipitation response over mountainous areas.

Technical results

The scientific results showed on the previous section were obtained using the EELA testbed during 1 week. The testbed was composed by seven sites distributed in Europe and Latin America making a total of 150 CPUs available.

After one week, 41 realizations concluded successfully, nine were still running. In order to complete all the experiments 510 independent jobs were necessary. In the next table we can see the output status (listed in a previous section) of the 510 independent jobs submitted to complete the experiment:

- 143 SUC and WALL
- 158 ERR
- 87 SUSP
- 41 PERF
- 81 MID.ERR

The big amount of ERR status (error) was caused by the misconfiguration of two sites where the all the jobs failed systematically. The amount of SUSP status is also higher than expected because during the experiment some sites didn't accept jobs because there were a lot of jobs waiting in their queues.

Another source of errors, mainly MID.ERR, is because some of the middleware services are overseas (Europe and Latin America). After testing some other testbeds the previous results can be considered as an example of the performance of the GRID.

Conclusions and future work

We presented the successful port of an atmospheric Global Climate Model to the GRID by using existing middleware solutions plus newly developed tools (Grid Enabling Layer and Workflow Management Layer) to account for specific requirements posed by this application. In doing so, several

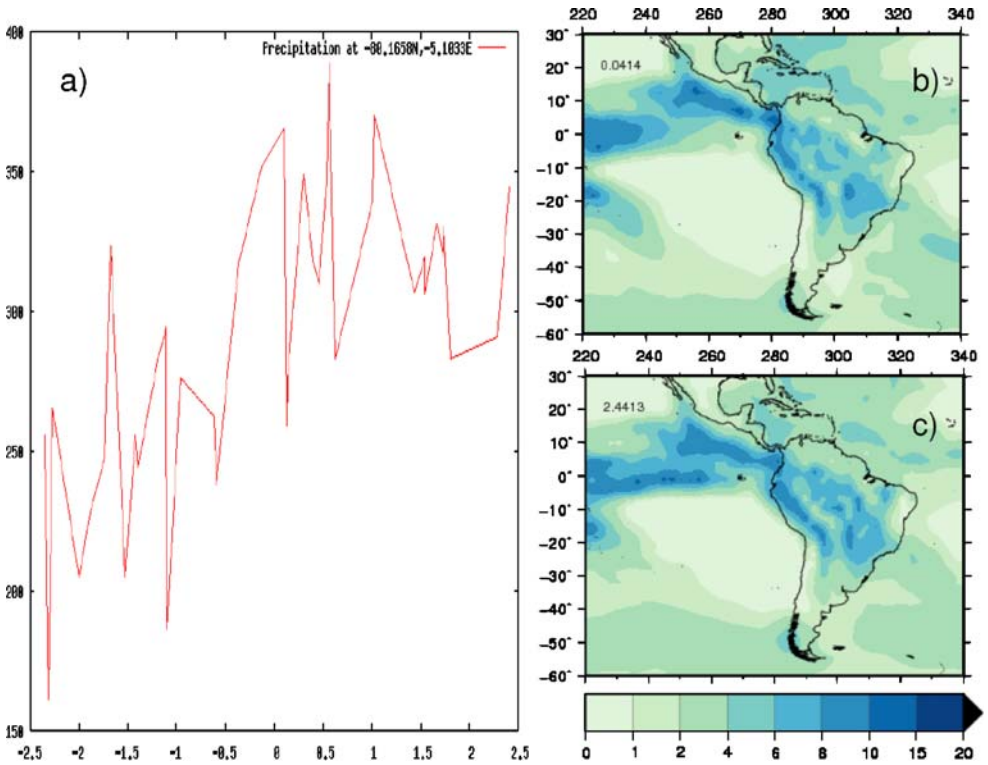

Fig. 5 **a** Accumulated precipitation interpolated to the location of Piura vs. the perturbation scale applied. **b**, **c** Mean precipitation (mm/day) for the period October 1997 through January 1998. The random scale factor used is shown on the *upper left corner*

weaknesses of current middleware were identified. We showed how current GRID technology is immature and not completely well suited for the Earth Science community. Specifically, for climate models, it has several weaknesses such as a high failure rate and time simulation limits, which require the creation of new middleware able to handle these problems.

In spite of these weaknesses, there are some fields in earth science, including recent trends in ensemble forecasting or any other involving independent simulations, which could greatly benefit from the GRID.

In our case, the achievements made thanks to the GRID-enabled application could not have been reached with the means usually at out hand. Unlike local clusters or supercomputers, optimized for MPI and OpenMP jobs, the GRID is perfectly suited for independent simulations. In our case, it would have been impossible accessing a cluster with the number of CPUs available in the GRID. An even more important topic is the storage capacity; climate models manage huge files that need a lot of storage capacity. Thus, even though many jobs failed and needed to be rescheduled, the benefits from accessing a GRID composed of a large number of processors and disk space outweigh the drawbacks. We completed around 50 simulated years in 1 week.

The next steps in our work are porting the regional WRF model to the GRID and create a new application combining CAM and WRF. The idea is that the users specify a region to study and then run a cascade of CAM and WRF simulations for this region. In order to do this, improvements to the workflow will be necessary to manage the dependencies between the CAM and WRF jobs.

As a final remark, although the effort of developing an application in GRID is very high, the results are worth. Once the initial effort is done, many other applications with similar requirements can be much easily ported. This successful experience is promising and we will continue working in this line.

Acknowledgements This work has been partially funded by the EELA project under the 6th Framework Program of the European Commission (contract no. 026409) and the Spanish Ministry of Education and Science through the Juan de la Cierva program.

References

Allen M (1999) Do it yourself climate prediction. Nature 401:642

Burke S, Campana S, Delgado Peris A, Donno F, Méndez Lorenzo P, Santinelli R, Sciabà A (2007) gLite 3.0 users guide. https://edms.cern.ch/file/722398/1.1/gLite-3-UserGuide.pdf

Collins WD, Rasch PJ et al (2004) Description of the NCAR Comunity Atmospheric Model (CAM 3.0). Technical Report NCAR/TN-464+STR, National Center for Atmospheric Research. http://www.ccsm.ucar.edu/models/atm-cam/docs/description/description.pdf

Foster I, Kesselman C (1999) The grid. Blueprint for a new computing infrastructure. Kaufmann, San Francisco

Hagedorn R, Doblas-Reyes FJ, Palmer T (2005) The rationale behind the success of multi-model ensembles in seasonal forecasting—I. Basic concept. Tellus A 57:219–233

Koblitz B, Santos N, Pose V (2007) The AMGA metadata service. J Grid Comput 6:61–76

Palmer TN (2002) The economic value of ensemble forecasts as a tool for risk assessment: from days to decades. Quart J Royal Meteor Soc 128:747–774