Execution Management of Scientific Models on Computational Grids

Alexandre Vassallo[1], Cristiane Oliveira[1], Carla Osthoff [1],
Halisson Brito[2], Julia Strauch[3], Jano Souza[2,4]

[1] LNCC – National Laboratory for Scientific Computing
Av. Getúlio Vargas, 333, Quitandinha, ZIP Code: 25651-075, Petrópolis, RJ, Brazil.
{alex, cris, osthoff}@lncc.br
[2] COPPE/UFRJ – Systems Engineering and Computer Science Program
Federal University of Rio de Janeiro – PO Box 68511, ZIP Code: 21945-970, Rio de Janeiro, RJ, Brazil.
{hmbrito, jano}@cos.ufrj.br
[3] ENCE/IBGE – National School of Statistical Sciences, R. André Cavalcanti, 106 s. 401
ZIP Code: 20231-050, Rio de Janeiro, RJ, Brazil.
juliast@ibge.gov.br
[4] IM/UFRJ – Institute of Mathematics/Federal University of Rio de Janeiro
PO Box 68511, ZIP Code: 21945-970, Rio de Janeiro, RJ, Brazil.

Abstract. This paper presents ModRunner, a scientific models execution manager running on a Grid platform. ModRunner is part of MODENA environment. Besides model execution, MODENA also deals with knowledge management about scientific models. It also works as a models library allowing for cataloguing, searching, reusing and generating scientific models. ModRunner is a simple and effective Grid Computing access system that eases the management of the execution of independent tasks on the Grid. In this paper, we present ModRunner running over the Grid Computing middleware MyGrid. As a case study we have been using ModRunner to schedule, submit and manage tasks for the execution of Population Dynamics models.

1 Introduction

Model management has been the subject of scientists in several works, ranging from model creation and execution to result analysis and model feedback, as stated by [1].

In general, models can be described as simplified representations of reality, whose goal is to abstract the reality portion which matters to the solution of a problem. Besides, models contain relevant information on phenomena or processes with the advantage of hiding irrelevant details of real problems.

In scientific work, phenomena or processes are usually complex and unknown. So, models may be used to represent them, being essential parts of any scientific experiment. An experiment usually tries to verify (either positively or negatively) some hypothesis stated by a scientist and it may have an underlying model, or even a combination of models about the phenomenon it is intended to prove. So, models play an

important role both in research area and practical applications of many knowledge fields.

In this work we consider model execution as the steps of running "model instances", like programs or workflow definitions, in order to perform the simulation process. According to [2], the simulation process made with scientific experiments usually does the transformation of input data to produce data with added scientific value.

In order to support model execution, we present ModRunner, a simple and effective web tool to perform the execution on Grid platforms. It can be used to encapsulate tasks submission to the Grid, providing a management layer over that submission. It provides easy to use interfaces to perform management issues like capturing model parameters, obtaining remote input data, scheduling execution submissions, submitting a model to execution, keeping a history of each execution instance and storing result data.

ModRunner is part of MODENA [3], an environment for scientific model management on Computational Grid platform. This environment has been developed to support researchers of the Geoma Project (Thematic Network for Research in Environmental Modeling of the Amazon) [4], which aims at the development of models to evaluate and foresee sustainability scenarios under different kinds of human activities and public politics for the Amazon.

The goal of MODENA is to provide an infrastructure that allows geographically distributed research institutions to share data, metadata, models, knowledge, and workflow definitions, as well as to share model execution in high performance environments, through a uniform Grid Computing platform. MODENA environment has to provide, at the same time, client and data server features to 1) reduce data, information and knowledge acquisition cost, 2) avoid data duplication, 3) reduce data processing and selection time, 4) reduce environmental data analysis and execution time and 5) generate simulation models and environmental scenarios.

ModRunner has been developed to run either on Grid Workflow platforms, like Globus [5] or on Bag of Tasks (BoT) platforms, like MyGrid [6]. This paper presents ModRunner running on MyGrid, which is the part that is in its more advanced development stage.

We have been using population dynamics models to validate our proposals. Those models try to investigate mosquitoes population control to reduce the number of cases of malaria in the Amazon region.

This work is organized as follows: the second section presents the case study applied to population dynamics; the third discusses some related works in the technology employed here; the fourth describes the Grid Computing middleware architecture used in this work; the fifth section does a brief review of MODENA environment and presents the ModRunner task execution management system; and finally the sixth section presents final considerations and indications of future works.

2 The Population Dynamics Model Case Study

Malaria is a serious public health problem around the world, affecting 40% of the population of more than 100 countries [7]. In agreement with the World Health Organization, about 300 to 500 million of new cases and 1 million deaths happen each year. In Brazil, 99% of all cases occur in the Amazon region where about 500 thousand cases are reported every year.

In the last few years, scientists have been working to create genetically modified mosquitoes in order to encapsulate the malaria plasmodium. These mosquitoes should couple to wild mosquitoes, and introduce refractory genes into wild mosquito populations.

The substitution of wild mosquito populations for genetically modified ones aims at the reduction or elimination of disease transmission, since a vaccine for malaria has not been discovered yet. This attempt should, however, begin only after a rigorous study of control strategy viability and collateral effects had been introduced [8].

As part of the GEOMA Project, a recent work [8] aims to analyze a mathematical model composed of a system of ordinary differential equations, which represents the main characteristics of the population dynamics of *Anopheles darlingi* in areas of Brazilian Amazon. The model also takes into account the seasonal variation of the density of the mosquitoes population due to water level fluctuation.

This is an example of an application that may be shared and managed by geographically distributed researchers from distinct research fields.

3 Related works

Some related works are described in the literature. Allcock et al [9] argue that the service requirements involved in data transport over Grids to high-performance, distributed data-intensive applications are: i) secure, reliable and efficient data transfer; and ii) the ability to register, locate, and manage multiple copies of datasets. The authors also presented the design and implementation of GridFTP protocol which implements extensions to FTP that provide GSI security and parallel, striped, partial, and third-party transfers in Globus environment.

Karnik and Ribbens [10] presented an approach based on a data-centric framework that offers a high-level architecture for Grid Computing Environments based on layers with clear interfaces defined in three entities, as follows:

- Model: A model, according to [10], is a directed graph of specific executable pieces defining the control-flow and data-flow in a computation.
- Model Instance: A model instance is a model with all parameters specified.
- Simulation: A simulation is a model instance assigned to and run on a particular computational resource.

That architecture is composed of three tools: job submission, parameter sweep and simulation lookup. We highlight the parameter sweep that is composed of three subsystems: i) an XML Generator that produces an XML representation of a typical input file, identifying the various parameters in it; ii) a Parameter Sweep Definition tool that

allows the user interactively indicate parameters and ranges defining an experiment, and use the XML file to produce a parameterized input file; and iii) a Sweep-Engine. It is interesting because it distinguishes model from its representation, although this approach accepts some parameters that may not be specified until runtime.

Zang, Wu and Wang [11] presented grid workflow based on dynamic scheduling and performance evaluation implemented over standards as GCC and WFMC, which consists of user portal, resource management component, grid services management, performance management and grid workflow engine featured by dynamic scheduling.

W. Cirne et al. [12] present a MyGrid Molecular Dynamic Simulation applications platform. The platform provides an out-of-box solution to Grid users. However, the system is not integrated to any scientific model management database system.

The MODENA proposal, besides doing model management also does knowledge management, bringing a novelty and contributing with the dissemination of the knowledge about scientific models. Furthermore, it offers transparency to the user-researcher in the access to high performance environments based in MyGrid platform. So, one of the main advantages of ModRunner is that the end-user does not need to know low level details of grid configuration or submission to have their tasks executed in a grid environment. By using a simple web user interface, a scientist with no deep knowledge about grid details may submit his/her models to execution, track the execution evolution and get the results, gaining access to the benefits of using a grid environment.

4 Grid Computing Middleware Architecture

MyGrid Middleware [6] is a production-quality solution for users who want to execute Bag-of-Task (BoT) applications on computational grid today. BoT applications are parallel applications whose tasks are independent. MyGrid provides an important platform for users that do not want to pay the cost of the installation and deployment of a heavy grid software, like Globus Middleware [5].

MyGrid design goals are to be a simple, complete and encompassing Grid Computing Platform. By simple, it means that the system has to be as close as possible to an out-of-box solution. The idea is that when a user wants to run his/her application, the last thing he/she wants to be concerned is the grid details. Complete means that the system must cover the whole production cycle, from development to execution, passing through deployment and manipulation of input and output. Finally, the system is encompassing in the sense that all machines the user has access to can be used to run his/her BoT applications.

MyGrid middleware assumes that the user has a machine, which is called the home machine, which coordinates the execution of BoT applications through MyGrid. The user submits tasks that compose the application to the home machine, which is responsible for performing out the tasks in the user's grid. The home machine schedules tasks to run on grid machines.

MyGrid provides simple abstraction through which the user can easily deal with the grid, hiding away the nonessential details. It schedules the application over whatever

resources the user has access to, whether this access is some grid infrastructure, such as Globus or via simple remote login (such as ssh).

5 The System for Execution Management of Scientific Models

The model execution management is supported by a set of tools that perform other model management tasks, within the MODENA environment. MODENA (Model Development Environment) is composed of three layers (Fig. 1). The first layer is the management one, which provides several features for knowledge management about models and execution management of model instances; the second is the grid access layer; and the third comprises the MODENA knowledge storage layer, which includes data, metadata, models, knowledge and ontology bases.

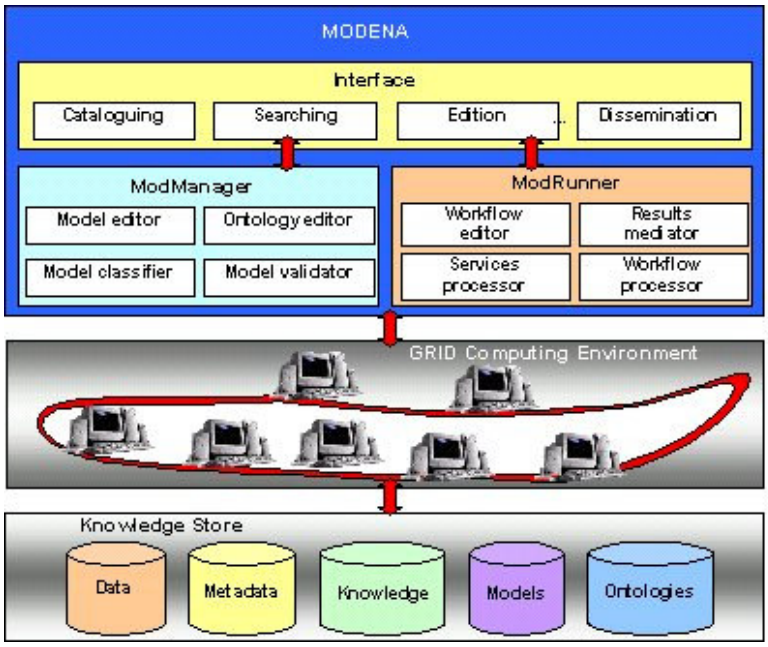

Fig. 1. The Architecture of MODENA

The fist layer of MODENA consists of two systems. The first one, named Mod-Manager [13], comprises a system for knowledge management on scientific models, responsible for activities like capturing, retrieving, generating and exchanging data, metadata and knowledge. Fig. 2 shows the ModManager screen responsible for model metadata registration. The screen portion presented corresponds to the registration of model parameters. The equations, workflow definitions, algorithms, programs, and default data, among many other model features, can also be registered.

It also enables model composition, which is the base for workflow composition that originates chained model execution.

The second system, which is the subject of this paper, is called ModRunner, a developing tool to perform computer simulations through the execution of instances of the models stored in the database. These instances may be constituted of workflow definitions that represent the steps for model execution. The workflow editor transforms model instances into steps to be executed by the services processor.

ModRunner provides an easy to use interface to perform tasks like capturing model parameters (e.g. files, numeric values, string data and command line parameters), obtaining remote input data, scheduling execution submissions, submitting a model to execution, keeping a history of each execution instance and storing result data. Output data are stored at the knowledge base, where model parameters and information about file submission are also kept. Furthermore, it lets the user to register qualitative data about the execution results.

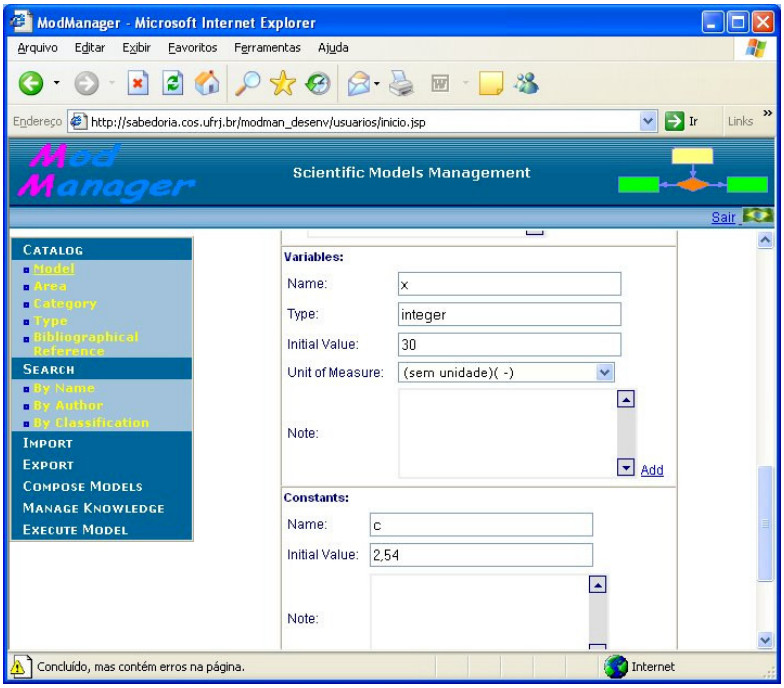

Fig. 2. Model metadata registration

The system works integrated to ModManager, meaning that the last registers and manages model metadata as the first captures model parameters, creates a model instance to execution, submits it to execution and stores the results. Both systems access the same database, as shown in Fig. 1.

As mentioned in section 1, ModRunner has been developed to run on different grid platforms, like Globus and MyGrid. However, this should be transparent to the user,

as he/she only wants to submit his/her tasks to the grid and obtain the results, with no knowledge about the grid infra-structure. The difference is greater to the system manager, who has to decide what grid platform he/she wants to provide access to.

A good point of this system is that it provides an interface that helps a user that is not an expert in Grid Computing to generate an execution task for each model to be submitted to the Grid.

Another feature of the system is that execution results, as well as execution histories, may be exported to other researchers, in formats like CSV, XML and KO (Knowledge Objects), just as ModManager does with any model metadata [12].

In order to submit a task, the user has to:

1) Get the target application executable file.
2) Save the executable file in MODENA database system.
3) Fill the application parameters fields.

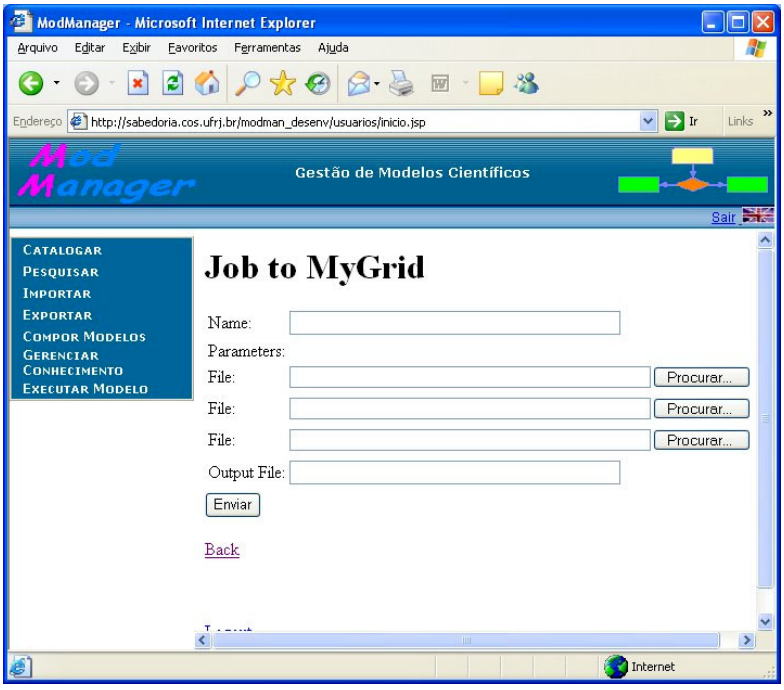

Fig. 3. Job Submission

Fig. 3 presents an example of a ModRunner task submission screen. In this case there are three parameter fields, all they of type 'file'. The name of the model and the quantity and type of the parameters have been retrieved from the system database. The last field is the name of the output file that the user may chose to save the results. If the field is left blank, the results will automatically be stored in the database.

After the model has been submitted, a MyGrid task execution command line is assembled and sent to the MyGrid Scheduler System to be executed. After the execu-

tion, MyGrid either sends back the results to the system or saves them into the file the user has chosen.

Job submission, in ModRunner, let the user either use different input data sets with the same model or use the same data set to different population dynamics models. Each simulation corresponds to different model instances running in the Grid Computing environment. So, ModRunner offers flexibility, usability, and extensibility to the model execution management, allowing the user change the parameters and submit jobs to the grid environment without any knowledge about it.

The models used in the case study had to perform numerical simulations in order to satisfy the following constraints:

- Assume that the genetic manipulation does not affect the environment fitness of mosquitoes;
- Consider transgenic heterozygous lines, thus the propagation of the malaria-refractory gene is stabilized at 56% in agreement with crossing rule;
- The population density of genetically modified mosquitoes maintains the same seasonal pattern as the population density of wild-type mosquitoes;
- Numerical simulation equations are adjusted to Novo Airão County (state of Amazon), the geographical area in study.

In order to find the closest parameters that represent the above constraints, the researcher had to test a large amount of parameters. Each parameter of job submission was stored in MODENA database for future analysis.

6 Final Considerations

The contribution of this work is to present a simple and effective system that encapsulates model execution submissions to the user, beyond providing a number of facilities for model execution management on grid platforms. It also works with ModManager, another module of MODENA, making a management cycle ranging from knowledge and metadata management to execution and data management.

The test with population dynamics model was effective once the scientists were able to do several simulations, changing the parameters and the kind of models until they find the best model that fitted their requirements, through a usable interface.

The tests also showed that the substitution of wild-type mosquitoes for genetically modified ones may take some years, depending, among other factors, on the amount of genetically modified mosquitoes introduced in the environment. Field observations should however be carried out for a sufficient large period of time to allow the detection of new variables or environmental modifications that initially were not taken into account in the mathematical model. With these new parameters, the model could be improved, tested and validated. Therefore, ModRunner has been considered a useful tool to help population dynamics researchers to manage and share their results with the scientific community.

As future works we aim the integration of the system with existing model libraries. We also aim the progress of the development of the workflow management features,

besides the implementation of grid services management and performance management.

Acknowledgement: The authors would like to acknowledge CNPq (Brazilian National Council for Research) and GEOMA Project for their funding and NACAD/UFRJ (High Performance Computing Center / Federal University of Rio de Janeiro), LNCC (National Laboratory for Scientific Computing) and IST/LNCC (Superior Institute of Technology).

References

1. Krishnan, R., Chari, K.: Model management: survey, future research directions and a bibliography. Interactive Transactions of OR/MS, (2000) 3 (1).
2. Cavalcanti, M.C. (et al): Sharing Scientific Models in Environmental Applications. Proceedings of the 2002 ACM symposium on Applied computing, Madrid, Spain (2002) 453-457.
3. Brito , H., J. Strauch, Souza, J., Osthoff, C.: Scientific Models Management in Computational Grids. 17th International Scientific and Statistical Database Management Conference. Santa Barbara, California (2005).
4. Geoma Project. http://www.geoma.lncc.br. (2006).
5. Globus Project http://www.globus.org. (2006).
6. MyGrid Project. http://www.ourgrid.org. (2006).
7. World Health Organization. http://www.who.int/en/. (2006).
8. Wyse, A. P., Bevilacqua, L., Rafikov, M.: Population Dynamics of *An. darlingi* in the Presence of Genetically Modified Mosquitoes with Refractoriness to Malaria. (2005).
9. Allcock, B., Bester, J., Bresnahan, J., Chervenak, A. L., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnel, D., Tueck, S.: Data Management and Transfer in High-Performance Computational Grid Environments. Mathematics and Computer Science Division. Argonne National Laboratory. http://www.globus.org/alliance/publications/papers/dataMgmt.pdf. (2004).
10. Karnik, A., Ribbens, C. J.: Data and Activity Representation for Grid Computing. Department of Computer Science, Blacksburg, VA. http://eprints.cs.vt.edu/archive/00000598/01/hpdc.pdf. (2002).
11. Zang, S., Wu, Y., Wang, W.: Grid Workflow based on Performance Evaluation. Department of Computing and Information Technology, Fudan University, Shanghai, China. http://166.111.202.9/chinagrid/download/GCC2003/pdf/347.pdf. (2003)
12. Cirne, W., Brasileiro, F., Paranhos, D., Costa, L., Santos-Neto, E., Osthoff, C.: Building a User-Level Grid for Bag-of-Tasks Applications. Book Chapter, High Performance Computing Paradigm and Infrastructure. Wiley Series on Parallel and Distributed Computing, Albert Y.Zomaya, Series Editor. (2005).
13. Brito, H., Strauch, J., Souza, J. ModManager: a Web-based system for Knowledge Management about Scientific Models (in Portuguese). IV Brazilian Congress of Knowledge Management (KMBrasil). São Paulo (2005).