

MUAT: An Environment for Accounting and Characterization of the Use of Computational Grids¹

Glauco Antonio Ludwig, Luciano Paschoal Gaspar, Gerson Geraldo Cavalheiro

Programa Interdisciplinar de Pós-Graduação em Computação Aplicada (PIPCA)
Universidade do Vale do Rio dos Sinos (UNISINOS)
Av. Unisinos 950 – 93022-000 – São Leopoldo – RS - Brasil
glauco1@turing.unisinos.br, {paschoal, gersonc}@unisinos.br

Abstract. The use of computational grids, which allow sharing distributed resources to achieve high processing power, has been spreading and acquiring importance lately (mostly in business environments). When the grid is employed in an inter-institutional, decentralized manner, it becomes necessary to characterize and account the use of the infrastructure to identify the most used resources, the number of services executed, who contributes with more resources, and so on. The management tools available are limited to monitor the status of environment resources, such as CPU load and memory usage, neglecting statistical and historical data about the execution of applications on the grid. To bridge this gap, the paper presents MUAT (MyGrid/OurGrid Usage Accounting Tool), an environment that aims to evaluate the use of computational grids infrastructures based on MyGrid/OurGrid solutions.

1 Introduction

Nowadays the use of computational grids is increasing gradually and consistently. The concept of a virtual supercomputer, built from the sharing of distributed resources has become very important in academic environments as well as in business environments. The distributed model used by the grid infrastructures present some motivations: (a) reduces the need for equipments upgrade, using geographically distributed computers; (b) enables a gain of performance when solving problems that require a high computational cost; (c) reduces the improper use of resources, taking advantages of computers processing cycles [1]; (d) provides a transparent use of resources, considering them as one powerful computer [2]; (e) offers access only to specific resources of a node, i.e., having access to one single node doesn't imply access to all the resources of this node [2].

In order to be able to use computational grids in production to execute more critical applications, it is necessary to have mechanisms for their management [3]. When used in large scale, involving many institutions and participants, it is necessary to characterize and account the use of the infrastructure aiming to understand its usage and allow the accompaniment of its evolution. Some reasons for this include:

¹ This work was developed in cooperation with HP Brazil P&D.

- obtain detailed information about the applications executed at the grid (i.e. origin, where they were executed, execution time, resources consumed and users that started their execution);
- identify the execution of malicious applications (an application executing for a very long time could point out the intention of only consuming grid resources);
- guarantee the correct scheduling and allocation of tasks at the grid, denying or allowing users to execute new applications based on their usage history;
- identify stations that are part of the grid and that are not contributing in a productive way (a certain machine receives tasks to compute and ends up by failing to process them very often);
- follow the evolution of the use of infrastructures (allowing the recognition of patterns and usage tendencies).

There are proposals such as [4], [5] and [6] that address grid management; nevertheless they concentrate mostly in describing architectures for monitoring the use of resources in grid environments. The existing tools are not concerned about accounting and characterizing the use of the infrastructure, being limited in relation to the kind of information they offer such as CPU load, memory usage, and storage capacity.

This work presents MyGrid/OurGrid Usage Accounting Tool (MUAT), an environment that aims to evaluate the use of computational grid infrastructures based on MyGrid/OurGrid solutions. Information that account and characterize the use of the grid are presented such as: (a) detailed information about the applications executed; (b) evolution of the use of the infrastructures; (c) identification of stations that are not contributing in a productive way. Even though a software architecture is also being proposed, special attention is given to the information presented and on how it can be useful to the grid administrator.

This paper is organized as follows: Section 2 describes the conceptual structure related to MyGrid/OurGrid. Section 3 approaches the proposal of an architecture and its validation by a tool. The paper is finalized in section 4 where the summary is presented.

2 MyGrid/OurGrid Grid Infrastructure

MyGrid and OurGrid [7] are computational grids solutions that are being developed by the Federal University of Campina Grande in cooperation with Hewlett-Packard (HP). The main objective of this work is the creation of an environment to execute Bag-of-Tasks (BoT) applications. Many research institutions are adopting the MyGrid/OurGrid technology. An example of its crescent popularity is the Pauá community, which is the first Brazilian grid environment geographically dispersed involving various educational institutions as well as HP Brazil.

The grid infrastructure can be described by the two software levels that are part of it (MyGrid and OurGrid). While MyGrid is characterized by being a single user grid, OurGrid is characterized by enabling the exchange of resources between different

administrative domains. These domains form a virtual community, thus generating a peer-to-peer network to share resources. MyGrid can execute independently from an OurGrid solution, or also be coupled to an OurGrid peer becoming the user front-end to a community.

There are three basic components that form the MyGrid/OurGrid grid infrastructure: GuMs, GuMPs and MyGrid. The GuMs are the stations that form the grid, being responsible for sharing their resources to compute tasks. The GuMPs are services that allow the configuration of which machines can be used as GuMs and are also responsible for dynamically providing GuMs when required by a MyGrid (a GuMP can be part of an OurGrid community or not). The last component is MyGrid, which provides the necessary support to describe and execute grid applications. During the execution of an application, MyGrid obtains GuMs, according to the demand, of the GuMPs to which it is connected. It is MyGrid's responsibility to schedule the execution of tasks and accomplish files transfer and collection to/from GuMs.

In MyGrid/OurGrid the applications executing are called jobs. Considering that these applications are of the BoT type, so each job is then composed of a set of independent tasks. A replica is an attempt to execute a task in one of the machines forming the grid. Multiple replicas of the same task can be created in two situations: (a) when the machine where it is being processed fails; and (b) to try to reduce the computing time of the job. MyGrid/OurGrid represents the execution of a job/task/replica according to a standard format. This standard helps the user to identify what is being processed by the grid. For example, when identifying replica 3.2.1, the first element (3) informs the job, the second (2) represents the task and, the third (1), the replica identification.

3 MUAT Environment (MyGrid/OurGrid Usage Accounting Tool)

This section describes the architecture used to obtain the relevant data to the accounting and characterization of the use of the grid and the main functionalities of the tool developed.

3.1 Architecture

The architecture developed employs the analysis of log files to characterize the use of the grid. MUAT was implemented in Linux, using Java programming language, HSQLDB database and JFreeChart library (for graphics generation).

Figure 1 illustrates the environment architecture. The collector module is responsible for collecting log files along the MyGrids existing in the grid (1) and providing them to the parser module. The parser module is responsible for processing the logs (2), extracting from each entry of the file the relevant attributes and inserting them in the database (3). The grid administrator can interact with the tool using a Graphical User Interface (GUI). Once a graphic is chosen, the GUI consults the database and the results are processed and presented accordingly (4).

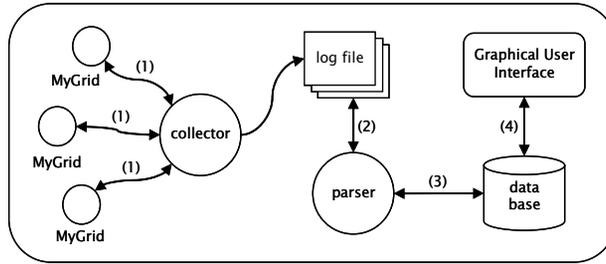


Fig. 1. Components of MUAT architecture and their relationships.

3.2 Main Functionalities

In MUAT all the information is available to the user through graphic views. Figure 2 illustrates the tool interface, which is organized to provide two types of information: Execution Duration and Execution Statistics. In Execution Duration it is possible to obtain information related to the execution times, such as for example:

- comparison of the execution time of various jobs that executed in the grid;
- time necessary to compute the tasks of a certain job;
- time consumed by certain machines processing tasks.

In Execution Statistics it is possible to find:

- number of jobs and tasks that executed in a certain time period (day, month or year interval);
- number of replicas that executed in certain machines of the grid (and the state of these replicas);
- number of machines available to compute tasks in the grid in a certain time period (day, month or year interval).

After choosing the type of information to view, the user chooses at the Display options panel some more detailed options, such as a time interval (if this is the case). By clicking at the Display button, the graphic is generated and presented at the Result panel or opened in a new window (in case the choice Open in a new window is selected at the Display options panel). This option can be quite useful to generate various graphics and compare them.

The graph inside the Result panel in Figure 2 illustrates the execution time of the job 3 tasks. It is possible to notice that the job was composed of 11 tasks. It can be seen also that the final state of all these tasks was Finished (represented by 'F') and that their execution time, in minutes, was around 2.233 to 6.333. This graphic contributes for the characterization of the use of the grid, providing details about the applications executed (in this case, the execution time and the state of the tasks of a certain job). It can be deduced also that the tasks were executed in machines with heterogeneous configurations, or even in machines under different CPU loads. Because it is an application organized in tasks with similar computational cost, their execution times were not expected to be so different. Furthermore, it is possible to

notice that the grid is possibly working correctly, since there are no tasks that have failed (what could indicate that some machines in the grid were not contributing in a productive way).

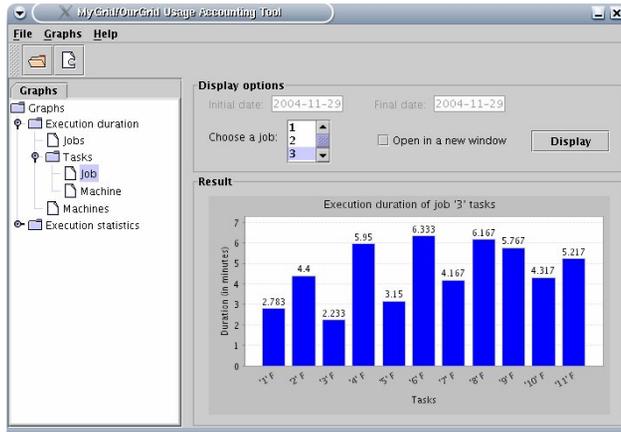


Fig. 2. MUAT graphical interface.

In Figure 3 it is possible to identify the number of jobs that executed in the grid in a certain time interval. After deciding which interval to use (day, month or year) the number of jobs finalized in this period is summed and the results are presented. As one can see it is represented the number of jobs that have finalized their execution between January and November 2004. It is possible to observe that from the month of August the grid has started to be used to execute an increasingly higher number of jobs. The type of result generated by this graphic, besides accounting information related to the applications that executed in the grid, helps on the identification of patterns and tendencies of its use. Having this information, an administrator can, for example, determine more precisely the need to upgrade the infrastructure.

Figure 4 illustrates the number of replicas executed in certain machines that are part of the grid. In this figure, one can view how many replicas have finished their execution successfully, how many have failed and how many were aborted or cancelled. By carefully analyzing this graphic, the MUAT user can notice the presence of two machines that are giving a very small contribution to the grid. It is possible to observe that Gum 10.16.165.44 has finished 14 replicas with the state Finished and 7 with the state Failed, which can indicate, for example, that this machine is frequently losing its connection to the network. Station 10.16.165.47 has concluded 12 replicas successfully and 17 with the state Aborted, leading to the conclusion that this machine is very slow and another replica of the same task is concluded first with state Finished. An alternative would be to remove these stations from the available GuMs relation since they can be degrading the performance of the grid and of the applications executed. This kind of information could be forwarded to the scheduler that would be able to guide its decisions taking this feedback into consideration.

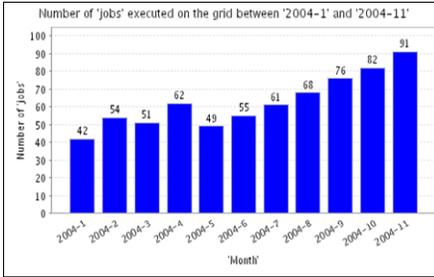


Fig. 3. Number of jobs executed in a certain time interval.

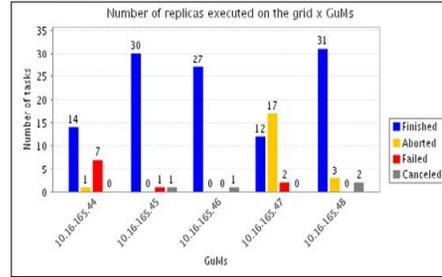


Fig. 4. Number of replicas executed x GuMs used

4 Summary

Nowadays the use of computational grids is consolidated in the academic environment. In business environments, its employment is more recent. To be able to use these grids in production, it is necessary to have automated mechanisms that allow them to be efficiently managed. This paper has presented the ideas of an environment whose objectives are to account and characterize the use of computational grids.

With this work it was possible to realize that the MyGrid/OurGrid grid infrastructures provides a valuable set of information for an application management schema. Hence MUAT was created, an environment that helps grid users and administrators to have a better understanding about the use and evolution of the grid, facilitating a task that would be very difficult or unlikely to be accomplished manually. The current version of MUAT is being provided to MyGrid/OurGrid administrators.

References

1. Foster, I., Kesselman, K. (1998) "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, 1st edition.
2. Nemeth, Z., Sunderam, V. (2002) "A formal framework for defining grid systems", In: Second IEEE/ACM International Symposium on Cluster Computing and the Grid, p. 202-211.
3. Cirne, W. (2003a) "Grids Computacionais: Arquiteturas, Tecnologias e Aplicações", In: Anais da Terceira Escola Regional de Alto Desempenho, p. 103-134.
4. Lee, D., et al. (2003) "VisPerf: Monitoring Tool for Grid Computing" In: International Conference on Computational Science, p. 233-243.
5. Baker, M. and Smith, Garry. (2003) "GridRM: An Extensible Resource Monitoring System", In: IEEE International Conference on Cluster Computing, p. 207-215.
6. Dinda, P. et al. (2001) "The Architecture of the Remos System", In: 10th IEEE International Symposium on High Performance Distributed Computing, p. 383-394.
7. OurGrid Project Home Page (2004a). <http://www.ourgrid.org>.