

Integrating Heterogeneous Grid Computing Environments

Alessandra C. G. Nascimento¹, Fábio M. Costa², Marcio N. Miranda³

¹Faculdade de Goiás (FAGO)
Rua 67-A 216 – 74063-321 – Goiânia, GO – Brazil

²Instituto de Informática – Universidade Federal de Goiás (UFG)
Caixa Postal 131 – 74001-970 – Goiânia, GO – Brazil

³Instituto de Computação – Universidade Federal de Alagoas (UFAL)
BR-104 Norte km 97 – 57072-970 – Maceió, AL – Brazil

alessandragarcia2004@yahoo.com, fmc@inf.ufg.br, marcio@tci.ufal.br

Abstract. *Grid computing provides integration of different environments, creating a unified system, where organizations might share, manage and access different resources regardless where they are localized. However, the lack of integration among different grid computing systems does not provide total exploitation of resources. This work investigates and develops an interface for integration of two heterogeneous grid computing environments: Globus and InteGrade. Such integration increases the computational power, transparently. Users will be able to submit jobs through both grid environments and applications will be executed into Globus, or into InteGrade or into both systems.*

1. Introduction

There are many workstations in different institutions where researchers can work. Although convenient for its users, this kind of distributed work is less efficient than a supercomputer, belonging to one institution, which is constantly busy, because these personal workstations stay inactive in the periods where the user is absent. But the high cost of a supercomputer becomes this solution undesirable. Grid computing [Foster 2001] appear to integrate these idle resources and are becoming widely used. They had left of being a simple method of aggregation of widely distributed resources in wide scale, with focus only in the community of high performance computing (high performance grids), to be an infrastructure of integration of systems guided to services (services grid). Several tools have been designed to implement grid computing application platforms. We can mention the Globus [Globus 2006], the InteGrade [Integrate 2006], the

Condor [Condor 2005] and the OurGrid [Ourgrid 2005], among others.

Although the conventional technologies that compose the current Internet allow the communication and exchange of information between computers, they do not allow an integrated approach for shared and coordinated use of resources belonging to different institutions in the accomplishment of more complex computations. In this scenario, it is common to find a large amount of idle resources coexisting with an increasing demand for computational capacity. The demand of high performance applications (such as those related to the areas of the health, genetic and physics of high energy) could be supplied through the exploitation of these idle resources. The main objective of grid computing is to supply an infrastructure to exploit these idle resources. Grid Computing presents concepts of virtual computational resources [Foster 1999], that is a basic requirement for computing on demand (use of available idle computational resources) and also it allows the creation of virtual organizations for implementation of thematic projects based on distributed environments.

To supply the lack of integration between the computational resources of several institutions and to integrate tools and systems of existing grid computing environments in research centers and university, this work proposes a way for the integration of heterogeneous grid computing environments, which combines environments that work with OGSA architecture (Open Grid Services Architecture) [OGSA 2005] and environments that work with CORBA technology [OMG 2003]. We desire to perform applications on a heterogeneous grid computing environment in a transparent way, without worry about changes

in the source code and allocation of computational resources

Section 2 briefly describes some related works. Section 3 presents the grid environments that are used as the case study for this work. Section 4 details the integration architecture. In section 5 the obtained results are presented. Final Remarks are drawn on section 6.

2. Related Works

Several research groups have worked hard in order to integrate different kinds of grid computing environments. [Foster et al. 2002]. This increased computing address collaboration, data sharing and interaction between nodes that involve distributed resources.

The job submission, monitoring and control using Condor-G and Globus [Mausolf 2005] show how we can use Condor job submission components with Globus to provide a more comprehensive solution. Condor is a batch queuing system and job scheduler for managing compute-intensive jobs, developed at the University of Wisconsin at Madison. We can use the strengths of both technologies to provide a good interface for job submission, monitoring, and control in a heterogeneous compute environment that may span administrative domains. The Java CoG project [Verma 2001] has done some similar adaptation between different (OGSA and pre-OGSA) versions of Globus. Another work is the GIN consortium (Grid Interoperability Now) being led by various people at the Global Grid Forum (now called the "Open Grid Forum") [GIN 2006].

The D-Grid integration project [DGRID 2005] of the D-Grid consortium is a platform that creates substantial synergies for distributed and new scientific collaborations on top of Internet based Services. To reach this goal, a general middleware and network infrastructure allows the provision and demanding of Grid resources (i.e. computing power, data, information, programs, etc.) and its dynamic allocation. This infrastructure will build the basis for new community specific applications and will enable new inter disciplinary collaborations to solve scientific problems. On top of this infrastructure new kind of "virtual organizations" will be created and will benefit from the general availability of services, resources and infrastructures.

3. Grid Computing Environments

One of the main characteristics of a grid is to deal with the heterogeneous equipments that compose it. The grids must be capable of communicate with each other. This integration is a big challenge if we consider that different grids are controlled by different and independent organizations, which do not share the same administration, the same politics or the same tools.

3.1 Globus Toolkit

The Globus Toolkit [Foster 2001] is a set of tools and libraries of software that give support to the grid architecture and applications. It is a project developed by Argonne National Laboratory (ANL) and by the University of Southern California. The main services belonging to the infrastructure are: Globus Resource Allocation Manager (GRAM), Monitoring and Discovery Service (MDS), Reliable File Transfer (RFT) and Grid Security Infrastructure (GSI). The Globus Toolkit 4 (GT4) [Globus 2006] is based on the concept of Grid Services, an expansion of the concept of Web Services. The main concepts and standards in which are based the GT4 are: OGSA, OGSII (Open Grid Software Infrastructure) [OGSI 2005] and Web Services.

3.2 InteGrade

The InteGrade [Finger 2003] is a middleware developed by University of São Paulo that allows the implementation of grids over not dedicated computational resources. The main characteristics of the InteGrade include: objected-oriented architecture, the use of CORBA as communication infrastructure and support to non-trivial parallel applications. Afterwards, it collects and uses machine standards in order to improve the scheduling. The basic structural unit of InteGrade is the cluster. A cluster is a set of machines grouped according to a specified criterion, where each one is called node. The dedicated node is a reserved machine. The shared node is the one belonging to a user who shares its idle resources with the grid. Finally, the cluster manager is the node where the modules responsible for collecting information and scheduling are executed.

4. Integration Architecture

The developed methodology uses, as case study, computational grids environments already implanted and simulations in real environments. With this purpose, the tools Globus and InteGrade are used. The most important services in the integration of the Globus and InteGrade environments are: security, remote submission of tasks and management of available information and resources in the grid. These are essential services to keep a grid working. Figure 1 presents the integration architecture between the Globus and InteGrade systems. The user submits the application to the grid in a transparent way without worrying about which type of system will control and manage the available resources for this execution. The user will be able to submit the applications throughout Globus, using the command "globus-run" or throughout the InteGrade using the ASCT tool. The applications scheduler will decide how to distribute the application in the grid according to the parameters given by the user. The Globus and the InteGrade do not exchange information because they have different architectures and protocols. To solve this problem, an adapter has been created in order to work like a proxy, to transfer the applications from one side of the grid to the other one. The information about the available resources in the grid can be obtained using the MDS component of the Globus.

The Globus and InteGrade integration is divided in 2 parts, according to figure 2. In the first part, the Globus user submits tasks to be processed by the InteGrade grid. The Globus considers the InteGrade grid as a local scheduler, where the GRM receives requisitions for processing tasks in its LRM nodes. The Globus GRAM has a set of modules that implement interfaces for the local schedulers of computational resources. These interfaces are responsible for the management of submission of tasks to the scheduler. The GRAM has interface for several types of local schedulers that belongs to other grids. This flexibility to accept new local schedulers, using the set of existing interface modules, allows construction of an interface for the InteGrade GRM.

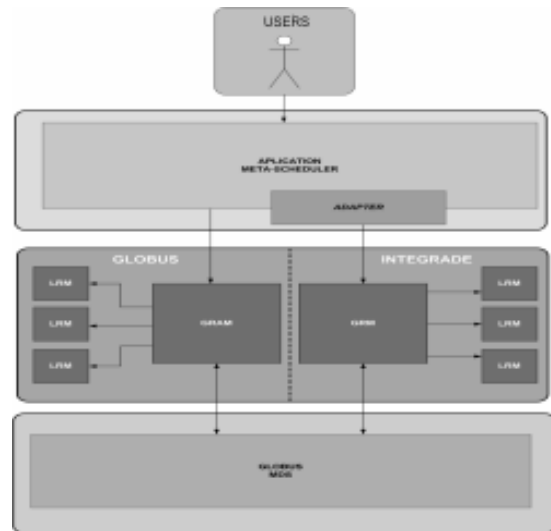


Figure 1 – Integration architecture

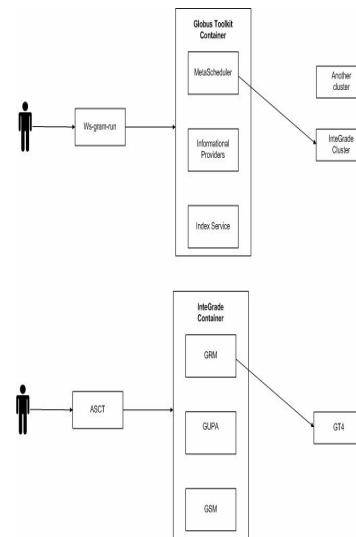


Figure 2 - Integration Models

Figure 3 shows the InteGrade architecture as an external scheduler of the Globus grid. Two new components has been inserted in the Globus Toolkit architecture in order to make possible to the InteGrade to process tasks sent by Globus: the InteGrade adapter, a module written in Perl language, that uses the interface of the Globus for local schedulers as reference, and the SGE which allows the monitoring of tasks sent from InteGrade to the Globus.

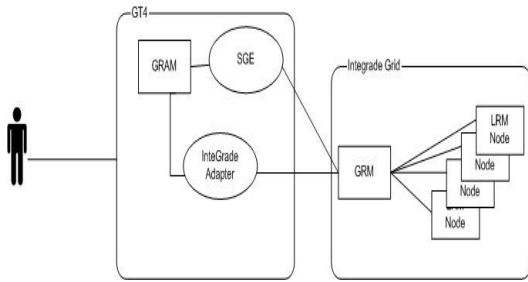


Figure 3 - InteGrade like external scheduler

Figure 4 illustrates how an InteGrade user submits tasks to be processed in Globus (GT4). For the scheduling of available computational resources, the InteGrade GRM does not consider only its LRM nodes because the Globus will be seen as an InteGrade LRM node, capable to receive tasks to be executed. With this purpose, a new component in InteGrade is created, called LRM Globus Node, which works like a proxy. Its function is to translate InteGrade information in Globus information. This new node also supplies information about the GT4 to the GRM. This kind of integration between the two systems avoids changes in the architecture or in the codes of its components. Moreover, a version of the ASCT was created in order to provide a command line interface to supply the integration with Perl module.

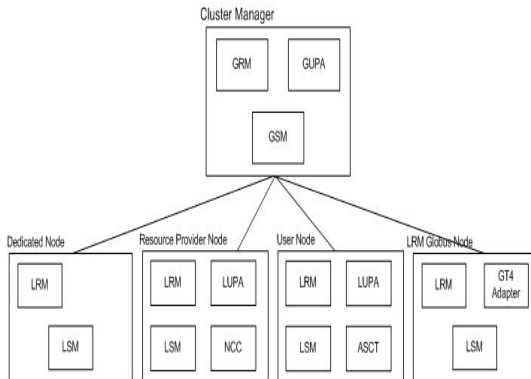


Figure 4 - Globus like InteGrade component

5. Results

The numerical results presented were obtained as follows: a script in Perl was developed in the Globus side and Java module was developed in the InteGrade side. A performance analysis is performed, using the execution time of the fibonacci(45) as performance measure on up to 4

machines. Figures 5 and 6 illustrate the obtained results.

5.1 Globus and InteGrade Analysis

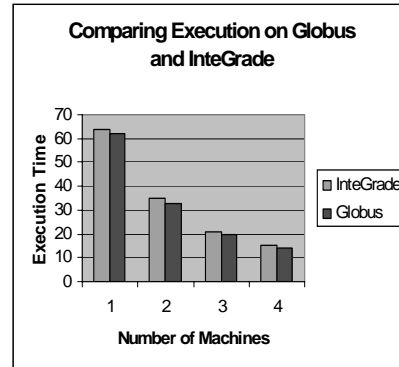


Figure 5: Globus and InteGrade performances

The InteGrade performance is a little worse than the Globus performance because it needs to register the application in the application repository and each LRM has to search the application bytecode whenever this information is necessary. On the other hand, the Globus entirely transfers all the application code to each remote host that it will execute the task.

5.2 Analysis of the Integrated Environment

Figure 6 shows the execution time of the same application using the integrated system, that is, submitting an application to InteGrade from the Globus or to the Globus from the InteGrade. The execution of the application using the integrated environment does not modify significantly the execution times. It does not matter if the application is executed from Globus or from the InteGrade. There is a higher delay comparing to the execution in each environment separately because it is necessary to wait the GRAM, from the Globus side, and the ASCT, from the InteGrade side, answer to the availability request so that they can execute the tasks and this generates an overhead. This happens due to relative simplicity of the application used as example. However, as soon as the applications become more complex and need more amount of resources, the influence of the overhead tends to diminish and the performance of the integrated system tends to increase.

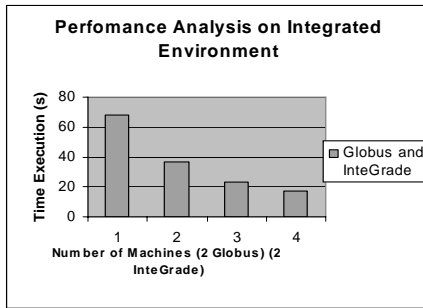


Figure 6 - Integrated environment performance

6. Final Remarks

The main goal of this work was to develop an interface to integrate heterogeneous grids, using as case study the Globus and InteGrade environments. The architecture of integration is defined and is implemented as specific modules, essential to the well-functioning of the two systems, allowing the integration between them. Among the main motivations of this work, we can mention the need to create and to develop a high performance parallel programming environment that could be executed in a simple, efficient, and transparent way, in any set of machines. The obtained results from the implementation of the proposed architecture of section 4 show the viability of the integration between the Globus and InteGrade tools and the diverse advantages of this integration, as the following: the mapping of a new component in InteGrade (LRM-GT4) makes possible to create a new module for the InteGrade, which is able to understand and to interact with the requirements of the Globus; the possibility of InteGrade to use the components MDS and GSI through the interaction with the Globus allows it to use new features to spread and to locate new computational resources; the integration and the interchange of knowledge between researchers of different research centers that eventually have different platforms. As an extension of this work, some points can be cited to be more deeply explored as, for example, the load balancing, whose algorithm can be modified in order to make it more efficient. Other extensions can include the access remote to files and to perform tests with more complex applications. Web Services can be used to implement future integrations.

References

- The Globus Alliance. Available in: <http://www.globus.org> Accessed in: February/2006
- InteGrade project. Available in: <http://incubadora.fapesp.br/projects/integrate> > Accessed in: February/2006
- Foster, I., Kesselman, C., Tueke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputing Applications, 15 (3), 200-222, 2001.
- Foster, I. (ed.), Kesselman C (ed.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers: San Francisco, 1999.
- Open Grid System Architecture. Available in: <http://www.globus.org/ogsa> Accessed in: August/2005
- OMG. The Common Object Request Broker: Architecture and Specification, rev. 3.0, Object Management Group, Needham-MA, USA, 2003.
- Open Grid System Infrastructure. Available in: http://www.globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf >.
- Condor Project. 2005 Available in: <http://www.cs.wisc.edu/condor/>. Accessed in: August/2005.
- S. Verma, J. Gawor, G. v. Laszewski, and M. Parashar, A CORBA Commodity Grid Kit, presented at 2nd International Workshop on Grid Computing in conjunction with Supercomputing 2001 (SC2001), Denver, Colorado, 2001.
- GIN - Grid Interoperability Now. Available in <http://wiki.nesc.ac.uk/read/gin-jobs>>. September 2006.
- Ourgrid Project. 2005. Available in: <http://www.ourgrid.org>>. Accessed in: August/2005.
- Finger, M., Goldchleger, A., Kon, F., Goldman, A., Bezerra, G. C. Object-Oriented Grid Middleware Leveraging Idle Computing Power of Desktop Machines. Journal of Concurrency and Computation: Practice and Experience, 2003.