MIDDLEWARE GRID MODEL RESOURCES PLANNING
BASED ON JAVA PORTABLE PLATFORM

PhD. Student Eng. Radu Niculiță
Politechnical University Bucharest, Faculty of Automatic Control and Computers

Abstract – This article presents developing a new model of Grid middleware resides is the huge unused computing power consisting of all PC computers connected to the Internet. The experience revealed that the design of such a Grid middleware model will raise numerous difficulties and will have to take into account the specific characteristics of such a special and highly distributed Grid system.

We suggest in our model an innovative method of planning, namely we do not mark the tasks to be scheduled and send it to the producers, but mark producers to take certain references from the communication medium.

Key words – Grid systems, middleware, resource broker

1. Introduction

The starting point in developing a new model of Grid middleware resides is the huge unused computing power consisting of all PC computers connected to the Internet.

Indeed, if supercomputer systems use less than half of the processing capacity then, the use of the computing capacity of hundreds of millions of today existing PC computers in the world is much reduced in the specific case of Grid system, including the multitude of PC connected to the Internet, reaching by the thesis’s author estimates, even levels of 5 ... 10%.

To use this large reserve of latent processing capacity, a middleware is needed than needs to fulfill two main requirements: to be independent of the platform, the operating system and any libraries installed on PCs and to be simple to install.

The need to fulfills the second main requirement of the middleware, namely to be simple to install, derives from the fact that unlike academic Grid systems, where there are network administrators who install and configure Grid system so that it can run a middleware, our Grid system contains a large number of desktop PCs connected to the Internet which do no benefit from a similar feature.

The solving idea and solution represents the main objective of the doctoral thesis and resides in designing and developing a middleware Grid model using a Java portable medium by using virtual Java type machines.

The author’s approach during the doctoral period was to use documentary and theoretical studies and analysis as well as creating applications and experimenting. The experience revealed that the design of such a Grid middleware model will raise numerous difficulties and will have to take into account the specific characteristics of such a special and highly distributed Grid system.

The most important characteristics that have led to difficulties in the process of designing and developing the new middleware model and which required finding appropriate solutions are:

- Adaptive character, non-uniform and non-predictable nature of the network and of all the resources requiring the middleware to be designed in such a way that the resulting model would be able to adjust to the dynamic behaviour of available resources;

- Not all applications can be parallelized, some are inherent sequential, which requires the need for a general middleware and a feasible Grid programming model to make targets achievable and easy to be implemented;

- Size and distribution of the Internet and its nature lead to the necessity of providing solid security throughout the Grid system;

- Any Grid system is very heterogeneity in nature due to the characteristics of available system resources, configurations and capabilities, of different networking characteristics with various types of connection, processing speed and different local levels of security; this requires the designed middleware to be able to choose the best out of the available resources during execution of an application and to ensure the necessary overall system performance.

2. Brief presentation of the developed middleware model

There are two main paradigms in the world of parallel and distributed systems, which are used as inputs for modelling of such systems:

- Master-slave paradigm - it consists of a series of master programs that controls the overall
functioning of the system and applications and more slave applications which are making calculations that are transmitted by the master program; this model is also known as the task farming model;

- Peer-to-peer paradigm - in this model there is no central entity to control or have a central vision of the system; the model consists of a number of independent tasks working together to achieve the same result.

Using a hybrid paradigm is a less explored approach. Such an approach is closer to the master-slave model through the existence of the central elements which allow slave programs to create their own sub-slave programs. This enables parallel development of more complex algorithms like the ones in the “divide et impera” class. The relations to the peer-to-peer model are explored from this perspective.

The middleware Java based Grid model that we propose has the following basic components, as shown in the diagram in Figure 1:

- Consists of a Grid architecture based on Java middleware, starting from a hybrid paradigm, close to peer-to-peer model in terms of the task planning and actual calculation;
- It is based on an existing widespread library, allowing the exchange of objects between systems; a new library is added that permits objects and classes serialization in an agnostic way, while also ensuring the needed mechanisms to send them to another computer; a blocking communication medium is used for communication purposes solely to ensure blocking communication, while object loading and saving is developed from scratch;
- Our model is based on a three-tier architecture, consisting of consumers, producers and a central control element, the Resource Broker; this system aims to develop a scalable model which is at the same time modular, portable and powerful;
- Further development by Grid applications programmers will be based on templates to be developed with the new middleware.

The proposed architecture is based on the following three levels (see Figure 2):
- Consumers who send applications to the system;
- Resource Broker, which deals with resource management and with scheduling, with a total central view of the system;
- Producers, performing the code execution for sent to the system.
The connection between the components is ensured through the existing network infrastructure, either in a LAN environment (if the system is running on a cluster) or via Internet (if the system is running on a large geographical environment). A simplified functional diagram of the communication within the system is shown in Figure 3.

Figure 2 Three level architecture scheme of the proposed model.

Figure 3 Simplified functional scheme of the communication within the system
3. Centralized planning model proposed for the resource broker

A principle diagram of the classical approach to scheduling the execution of a task is shown in Figure 4.

![Figure 4 Principle scheme of on how to plan the execution of a task in scheduling classic way.](image)

The planning of a task execution inside the developed middleware model is presented in the principle diagram in Figure 5.

![Figure 5 Principle scheme for planning a task execution according to the middleware developed model.](image)
BCM – Blocking communication medium
We suggest in our model an innovative method of planning, namely we do not mark the tasks to be scheduled and send it to the producers, but mark producers to take certain references from the communication medium. This translates into designating producers to run a task and the purpose of a task can be changed by delegating to another producer to run that task, without changing the references in the blocking communication system.

This approach also allows grouping of producers, but this time dynamically; another consequence is that it can change the task a producer will run without influencing the tasks already planned.

To mark a producer to run certain tasks, we are simply sending the producer a number of items to use when it takes objects from the communication medium; this can be seen illustrated in the simplified operating scheme of the developed model for centralized planning reproduced in Figure 6.

When a producer starts, it will first publish its capabilities. A list with all the binary codes it knows how to execute is included. Each such ability is actually a collection of three things: a language for which the producer has runtime support, running equipment processor and operating system. After publishing the capabilities, the resource broker creates a list of templates for each producer. It will then send the list back to producer. Soon as it receives the first templates, the producer is ready to work and start running tasks. If at a later time Resource Broker decides to change the templates for a producer in order to assign it to run other tasks, all it has to do is to create a new list of templates in the form of messages and send them to that producer. Each producer has a special thread waiting to receive new templates and when an appropriate message is received, it automatically updates its running templates.

4. Conclusions
- The scheduling module is the main component of the resource broker. Its task is to choose producers that can and should run any new task sent in the system and the choose task producers which will start generating tasks for a new application.
- Scheduler design allows having several different implementations for the underlying planning algorithms. However, a single

---

**Figure 6** Simplified operating scheme of the developed model for centralized planning.

The Romanian Review Precision Mechanics, Optics & Mechatronics, 2009 (19), No. 36
implementation can be used at a certain time. To implement a new scheduler, our system programmer needs only to extend Scheduler class. This class consists of two methods and a running thread. The methods are calls used to plan a new application/task, call that returns a line containing the IP address or DNS name of the computer that has been delegated to run that application/task that is given in the parameter when the method is called. The scheduler to be used in the system is chosen at boot time and starting point of the Resource Broker.

- To cover in an effective way the authorization, accounting and security aspects of the system, system that goes far beyond the organizational boundaries and reaching the whole Internet, the model has provided and designed a centralized secure scheduler at the resource broker’s level;
- The developed scheduling model is designed to meet the requirements established in the PhD thesis: to accept, with a minimum of restrictions, to perform general types of activities; the access to the resources needs to be performed via a reliable mechanism; the scheduling model should be capable of dynamically update its algorithms and should implement all the update activities in a transparent way;
- The developed model is putting forward an innovative method of planning which does not mark the tasks that are placed in the communication medium but it is marking the producers to take certain references from the communication medium; this method provides a number of enhancements for the performance of the scheduling activities which are presented in the thesis.

5. References