A Multiagent Grid Metascheduler

JOSE AGUILAR[1], RODOLFO SUMOZA[2]
[1]CEMISID, Facultad de Ingeniería, Universidad de los Andes,
Mérida, VENEZUELA
[2]Departamento de Computación y T.I., Universidad Simón Bolívar
Caracas, VENEZUELA
aguilar@ula.ve, sumoza@usb.ve

Abstract: - This work proposes a Metascheduler for GRID platforms based on the Interaction Protocols of the Multiagent Systems. These protocols use the paradigm of economic models to define the coordination mechanisms in agent communities. Specifically, in this work we use auction and tender economical models. We propose an adaptive Metascheduler for GRID platforms using these ideas. According to the number of available resources one of these models is used to coordinate the assignment of resources.

Key-Words: - GRID, Metascheduler, Multiagents Systems, Auction, Tender, Assignment problem.

1 Introduction

The GRID intends to satisfy computational environment's necessities that the traditional systems have not been able to do, as it is to share the great capacity of calculate and storage, dispersed geographically [1, 2, 16, 17, 18, 19]. This area presents a great quantity of topics that they even need to be broadly researched, such as that of the Metascheduling. This work studies the topic of the global scheduling problem, or metascheduling, in GRIDs platforms. The metascheduler should decide if new applications introduced in the platform will be accepted, and it should guarantee the quality of the service to offer for each application considering the availability of resources, maximizing the performance of the platform, optimizing the throughput and execution time, among other aspects [3, 4, 14, 16, 18]. We propose a protocol based on Multiagent Systems (MAS) that execute the processes of a global scheduler. The mean idea is to take advantage from the MAS, such as its capacity to solve problems in a distributed and autonomic way. Particularly, we consider the use of the MAS interaction protocols used in coordination tasks, to adapt them to the GRIDs platforms. These protocols have been well defined for the FIPA [5].

One of the first proposals that combines the theory of agents with the scheduling in GRIDs, was presented in [4]. The idea of this work is to assign scheduling tasks to the client. For it, each client possesses one agent, which has the responsibility of managing the user's necessities, entering to the market of resources when they receive a request from a local application. In [6] are evaluated metaschedulers' architectures and task assignment politicals for intensive calculation, in which they modeled and simulated a metascheduler for a GRID constituted by several clusters interconnected by a WAN. Each cluster internally is interconnected through a LAN, and its metascheduler consists on a group of agents, which have the task of verifying if the resource is or not available. If the resource is available; the agent informs the central coordinator that sends it a task that is adapted to it. In [8, 16] they develop a long term scheduler using economic models like auctions, from the point of view of the competition among users. They describe the characteristics of the process of planning from two levels: the long term schedulers, and the local or short term schedulers. In [7], in addition to the economic models to solve the problem related with the long term scheduling in the GRID, they simulate it using real workloads. [7, 8, 16] don't use the theory of agents in the proposal. In [3] they study the application of the economic models in the GRID. In that work several economic models are described which can be used for the handling of resources and the scheduling to long and short terms. They don't use the MAS theory. There are other proposals to solve the metascheduling problem using other approaches different to the economic models and the theory of agents. In [1, 13, 14, 17, 18, 19] several of these alternatives are described and evaluated.

We propose an adaptive Metascheduler for GRID platforms using the auction and tender economical models. According to the number of available resources one of these models is used for our scheduler to the assignment of the grid resources.

2 Proposed Protocol: The Metascheduler

2.1 Elements of our System for the Resource Assignment in a Grid Platform

In the figure 1 can be appreciated the participants and their interactions, in a GRID platform, to the assignment of resources/services.

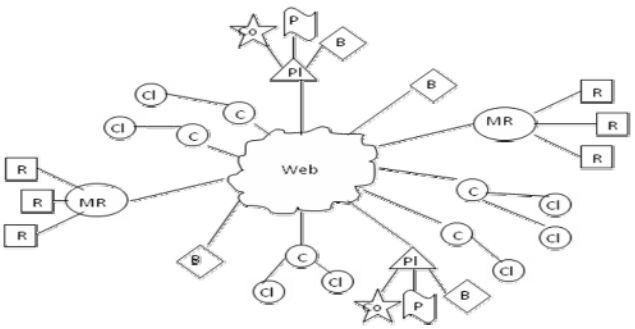

Fig. 1. The typical participants and their interactions, in a negotiation by resources in a GRID platforms

Where:

R: Owner of Resources and their offered resources, which are the suppliers of services. These resources can be in different geographical places in the network, and they can establish groups or associations to satisfy a requirement in a certain moment.

MR: Resources or Services Manager. it is the virtual representation of the resources. They determine the terms under which will be carried out the services proposals. This participant is the services supplier in the GRID.

Cl: Clients or Users, they can be devices, applications, other resources, etc., which are interested in using a certain resource.

C: The brokers are virtual participants and represent the clients or users in the GRID.

Pl: Metascheduler is a virtual participant that controls or coordinates the long term process of planning. It has to mediate between the brokers and the resources managers. It is the regulator entity of the interaction between the clients and the resources, and it is the responsible of the global planning. It is composed by several elements, such as the searchers, the portals, and the resource assignment mechanisms.

B: Resources Searchers. They are applications that are part of the metaschedulers. Their role is looking for the available resources in the GRID that fulfill the requirements given by the clients. They can be integral part of the metaschedulers, or can be independent.

Co: Coordinator, they are responsible of carrying out the selection tasks and contracting.

P: Portals, it is the interface among users and resources, and the input door to the GRID.

2.2 Metascheduling Process Characterizations

- When a client requests a service, it makes it through one of the portals that are distributed geographically. When the user interacts with a portal, one agent's instance "broker" dialogues with the user or client to collect the information about the resources that client needs. With this information, the scheduler assigns the task to the searcher (or to several searchers).
- The scheduler interacts with the searchers, with the Resources Manager agents, and with the Brokers.
- The Resource Managers publish the resources offered by the owners. A Resources Manager can work for several resources' owners.
- The searchers have the responsibility to find resources that fulfill the requirements of the users.
- The users or clients are represented by brokers; these brokers try to get the best resource for their client. One broker can give supports to several clients.
- When resource owners want to offer their services, they should make it through their own Resources Manager, or with a shared MR.
- Each client's request can have associate one or several processes. Each process is represented like a sequence of tasks, where each one of them requires different types of resources. The client can be an application, final users, etc.
- The resources' final selection process, including contracting and assignment, is carried out by the coordinators.
- The Co is the agent that coordinates the negotiations among the participants.
- In our MAS, the MM, B, C and Co are the agents.

The scheduler has two schemas, and together they generate a hybrid schema, which represents the final approach proposed in this work.

- The first is from the point of view of the users or clients, denominated in this work as the client-schema, where the Resources Managers through the scheduler tries to satisfy the requirements of the clients. This process is represented by the structure of bids/tender (economic model), where several salespersons and a buyer exist.

The clients' interests are considered; the main objective is that these can acquire the resources with the best benefits and to the best price.

- The other schema is from the point of view of the owners of the resources, denominated resource-schema, where the idea is that the client proposes the best offer, according to the salesperson's specifications. This process is similar to the auctions (another economic model), where several interested buyers and a single salesperson exist. In this case, the interests that are considered are those of the owners of resources, and the main objective is to be able to sell the resources or services to the client that offers the best conditions.

2.3 Hybrid Schema

In the two previous schemas, some of the probable situations that can present in the moment to establish a process of "negotiation" about some grid service or product are the following:

- one where several bidders of products or services can converge vs. a single buyer or client,
- and the inverse scenario where several clients exist for a single resource.

These are typical scenarios on GRID Platforms. Understanding that the GRID is a dynamic system (the clients and services/resources suppliers are changing permanently), it is preferable to think in a hybrid schema, where we combine the client and resource schemas (see figure 2)

The macro-algorithm is:

A. At the moment in that a client generates a request of a resource
   - To invoke the service of the portal, allocated in the corresponding scheduler server in the GRID. Since several types of clients exist: final users, applications, etc., the service of portal will interact in different ways as it is client's type.
   - For reasons of security, once connected with the portal, it should be executed an identification and authentication process.
   - With respect to search and assignment of resources, we need to introduce the listing of requirements.
   - The scheduler server activates a broker that is the application (agent) for the client's requirements. The broker sends the listing of requirements to the server.

B. The metascheduling process:
   - Given the list of requirements sent by the broker, the metascheduler active the general searcher.
   - The searcher begins the search process.
   - The searcher generates a list of resources. This list contains a list of resources for each type.
   - Once has the list of resources, the selection process is activated through the use of the client schema (bid), only if the number of resources or owners of resources are bigger than that of the interested clients. This protocol interacts with the RM to select a resource for type:
     - In Pl a profile is generated (bid sheet) with the user's requirements. This profile is the pattern that is used to look for the best resource among the list.
     - Because is a bid process, the Managers of services in the initial search offers their best proposal to try to win the client's approval.
   - For the cases where just one salesperson exists, or there are several clients concurrently requesting a resource in particular, the negotiation process is the auctions. This consists on making an offer for the service to the selected supplier. The supplier will choose, among the offers that it has, to who offered its service.
   - Once concluded the selection of all the necessary resources to fulfill the client or owner's requirements, it is carried out the contracting with all them.
   - After carrying out this contracting, the authorization is given to the client so that it uses the resources. In general, the resources are managed using theirs local planning mechanisms.

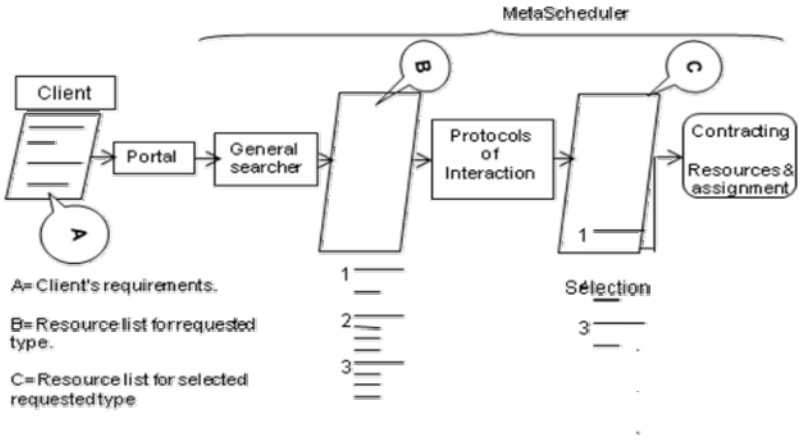

Fig. 2. Hybrid schema of the Metascheduling

3 Agent's Interaction

AUML is used [9] to represent the MAS proposed in this work, which is an extension of UML (Unified Modeling Language) that incorporates the representation of agents. The Protocol, activities, and collaboration diagrams were used to describe our approach. In the protocol diagrams are represented the agents in squares or boxes, and the lines constitute the interactions among them. The sequential or chronological order is specified by the numeration that accompanies to each message. When we need to specify the internal activities of an agent, the activities and collaboration diagrams are used. The Figure 3 shows the interaction UML diagram of our protocol. The semantics of the messages exchanged is based in [9]. Next we describe some of the messages (the rest is in [10]).

**(request-1*
:sender (agent-identifier :name broker)
:receiver (set (agent-identifier :name coordinator))
:content
"buy-service (resource Y, offers of the broker X) "
: language C)
Description: With this message the broker sends the coordinator its requirement.

**(cfp-1*
: sender (agent-identifier: name Manager)
: receiver (Sep (agent-identifier: name coordinator))
: content
"((action (agent-identifier: name coordinator)
(to sell resource Y))
(for Y (price Y> C) "
: language C)
Description: For the cases where a bigger number of clients exists with respect to the Managers number, we use the auction (resource schema)).

(propose-1
:sender (agent-identifier :name broker)
:receiver (set (agent-identifier :name coordinator))
:content
"(action (interacción (comprar recurso Y, oferta a Y)))
:in-reply-to cfp-1 o cfp-3
:language C)
Description: This message sends a broker to the coordinator, previously established an auction, to make its proposal to the manager.

(propose-2
:sender (agent-identifier :name manager)
:receiver (set (agent-identifier :name coordinator))
:content
"(action (interacción (comprar recurso Y, oferta a Y)))
:in-reply-to cfp-2
:language C)
Description: This message send a coordinator to the broker, previously established the licitation process, to make its proposal

(inform-1
:sender (agent-identifier :name manager)
:receiver (set (agent-identifier :name coordinator))
:content
"venta-recurso (Y)"
:language C)
Description: For the case of a licitation, the manager sends to the coordinator its availability to satisfy the requirement of the broker.

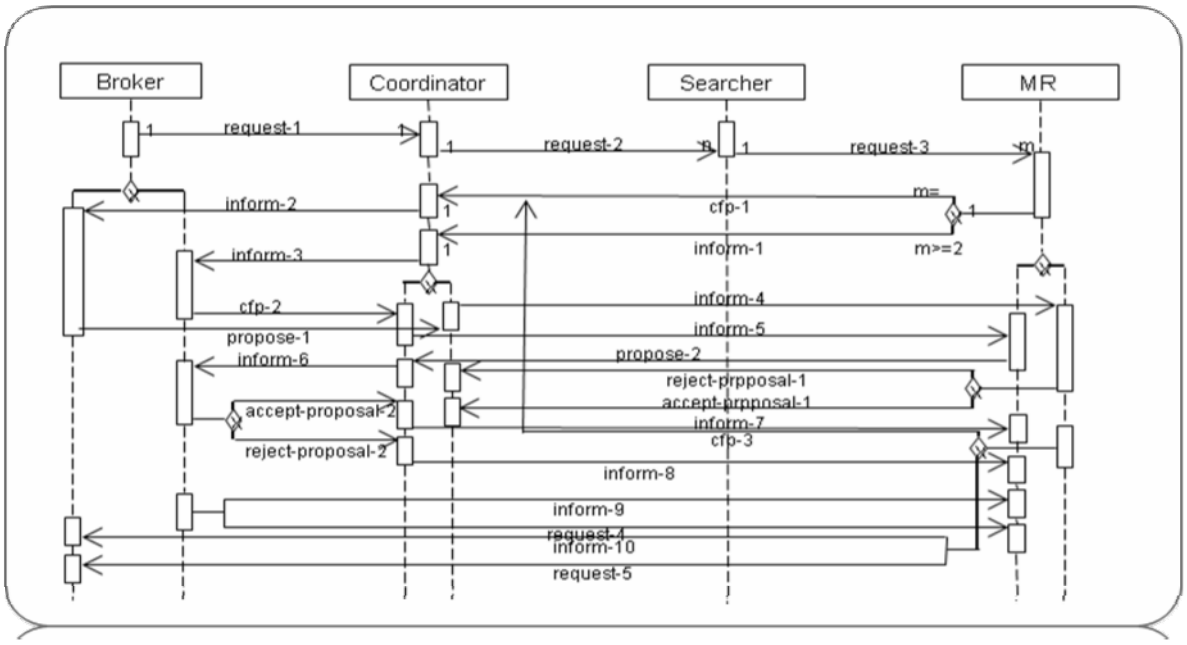

Fig. 3. The Interaction UML diagram of our proposal

The Figure 4 shows the collaborative and Activities UML diagrams of our proposal. In these diagrams we can see the adaptive capabilities of our system. According to the number of grid resources available, it uses an auction or a tender protocol for the resource assignment problem. The arrows indicate the sense of the interactions (additionally, it has the list of messages associated to each interaction). The numeration indicates the chronological or sequential order of the messages.

The auction cases are indicated with a "s", for example, the message 3s:cfp-1 indicates that for the case of auction, in the step 3 the message cfp-1 was sent from the Manager to the coordinator. The cases of bid are indicated with "l." Also, it is important to highlight that the brokers have two different roles, the change between roles depend on the schema used: auctions or bids (That is similar for the Managers). The diagram of activities is defined for the coordinator agent, pointed out by the serial horizontal lines. In this diagram the coordinator's activities are shown since it receives the request of the brokers, until establishing a winner, either for an auction or for a bid. The explanation or semantics of the messages is the same one that the one used for the interaction diagram of the hybrid schema.

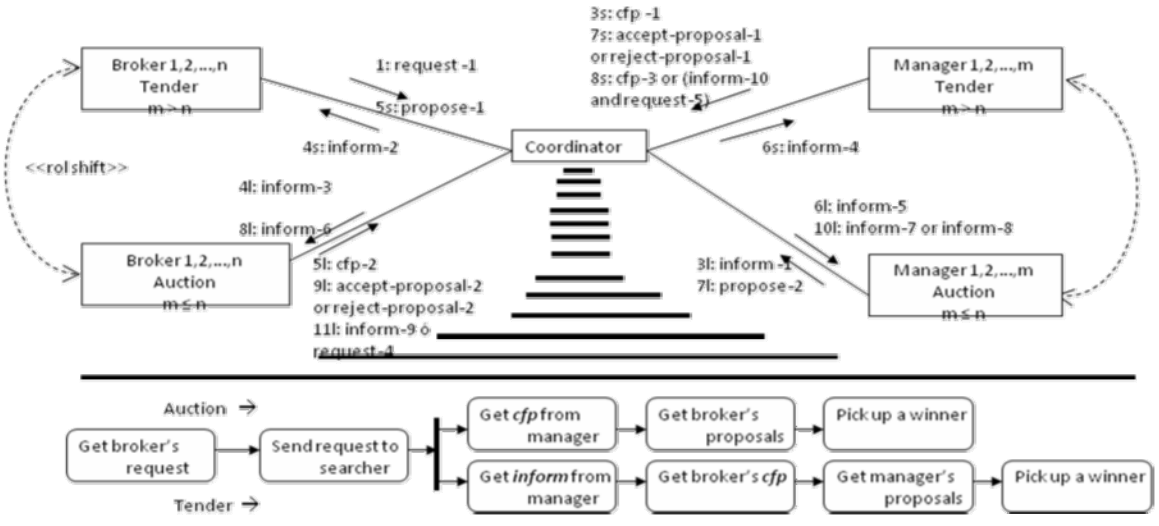

Fig. 4. The collaboration and Activity UML diagrams of our proposal

4 Experiments

4.1 Modeling and Simulation of the protocol in SimGrid

The tool for simulation was the SimGrid [6, 7], which allows to simulate distributed algorithms of scheduling. Particularly the used module was MetaSimGrid or MSG, designed to carry out simulations of general propose. Our architecture of agents establishes a group of agents (broker, coordinator and Manager), each one with functionalities well specified. For our simulation, we will consider several broker agents, a single coordinator, and several Manager agents.

For the function related to the broker, the necessary instructions were programmed for the creation of the initial messages that includes the data required to establish the auction or bid processes. For the simplification, each type of service or product is represented by a natural number, beginning from 1, until the quantity of types of resources that can request a broker. Each broker can generate and to send several requests for several types of resources. To carry out each request it should generate and to send a task that allows it to make such a requirement to the coordinator, each one of these tasks will have associates the data corresponding to the bandwidth that consumes that task, and the computation capacity that requires to be processed. The input parameters are: price of the initial offer, maximum and minimum number of requests, quantity in bytes that each task can consume at the level of communication, quantity in flops that each task can consume at the level of computation. Additionally, the broker agent has a group of instructions dedicated to receive the messages sent by the coordinator. For the generation of request, we have used a Poisson distribution, calculating the times between each request according to an Exponential Distribution.

At the level of the coordinator agent's, four specific blocks were programmed: the first one is to the reception of the messages sent by the brokers, the second one is for the shipment of requirements to the Managers and the reception of the answers to these shipping from the Managers, the third block is the heart of the scheduler since it decides if we will use the auction or bid/tender protocol, and depending on each case, it determines the winners in each process type, according to the data sent by the brokers and the Managers. The fourth block is dedicated to send the assignments results to the broker and Manager agents. Inside the coordinator, one data structure is used (a list of lists) that contains the information about all the requests made by the brokers, according to its type, and the Managers that responded informing that they are willing to provide that product or to give that service (see figure 5). The input parameters of the coordinator agent's are: maximal number of brokers and Managers to work, names of the brokers and Manager, quantity in bytes that each task can consume at the level of communication, quantity in flops that each task can consume at the level of computation.

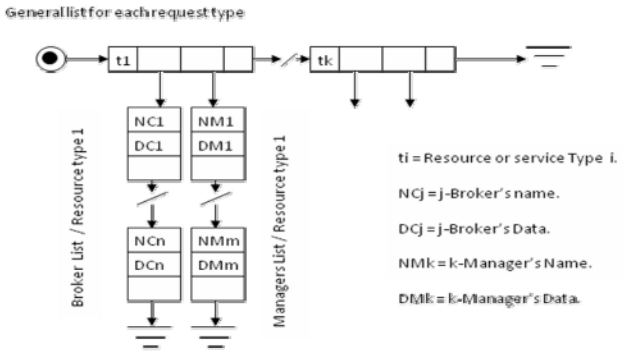

Fig. 5. Data Structure of the Coordinator Agent

With regard to the Manager agent, this is divided in two blocks, the first one is to the reception of the requests sent by the coordinator, the second one is used to send the information related to the resource type that the Manager offers, the initial cost, and the minimum cost. That is the necessary information to establish the auction or bid/tender processes. The input parameters are: number of resources or service to offer, the broker's name with which will be negotiated. The initial and minimum costs of the resources are generated randomly (uniform distribution from 0 to the number of resources introduced as parameter).

4.2 Experiments

For this work, we have modeled the Reacciun 2 interconnection network, the Venezuelan Academic network that interconnects the different Venezuelan public universities. The reason to use this network model is because on this WAN has been installed a GRID, denominated the Venezuelan GRID [11]. In this network, for its initial phase, eight Universities and Research centers were interconnected: ULA, UCV, UC, USB, IVIC, UPEL, UDO and UCLA. We are supposed "D" to be a group of domains or local networks (the university campus of Reacciun 2) present in the platform. Each domain is connected to a global network WAN through a switch, where bwtd

is the bandwidth of the global connection and latend represents the global latency for that connection.

With respect to the modeling of the applications in the GRID, we used a model proposed in [6]. In this work, the global workload corresponds to the tasks introduced in the system that are dedicated to the negotiations of the metascheduler. A task mk possesses the following attributes: tk, longk, prock, clientek, where tk is the task's input time to the global system, longk is the CPU requirement (computation units, for each time unit), tipok is the resource type that the client requests, and clientek is the organization or client introduced the task. According to [8], the methods to generate workload are: randomly, derived from trace of real systems, or using stochastic methods. We use the stochastic methods in this work; specifically, the generation of tasks was carried out using an exponential distribution to determine the times among each generation. The average used for the exponential distribution was 10 time units. This way, applications were generated from each one of the domains (the 8 universities and research centers). On the other hand, in the first experiment we generate 100 tasks by each domain and we carried out 30 simulations with each pattern. In the second experiment a random number of tasks in a giving interval by each domain were generated and we carried out 10 simulations by pattern.

The processes associated to the metascheduling were only considered, which conclude when is carried out the assignment of the resources. Remember that in our simulation each domain represents one organization in Reacciun 2. Each domain will offer resources in the GRID Venezuela, that means we will have a manager (RM) for each domain, and also, as the requirements of resources will be generated also from these domains, they will also have a broker. The requirements of the tasks generated by the brokers were considered as the metrics to evaluate our approaches. Two states are defined for a broker's task mk: execution state and concluded state. If one task is in execution that implies that it has been generated by the broker and it has been received by the coordinator, but it waits for the answer of the coordinator. In the case of having given an answer for the coordinator, the task is considered as concluded. The beginning of each state is defined as: ek for the beginning time and fk for the finalization time. The waiting time is defined like ek-tk, and the execution time is fk-ek. The waiting time average is:

$$Wt = \frac{1}{\text{cardinality}(T)} \sum_{k \in T} ek - tk \qquad (1)$$

The execution time average is:

$$Et = \frac{1}{\text{cardinality}(T)} \sum_{k \in T} fk - ek \qquad (2)$$

We define as the execution global time as the quantity of time used to culminate the metascheduling process, in every period of executed time.

$$t - global = \max_{k \in T}(fk) - \min_{k \in T}(tk) \qquad (3)$$

The network in SimGrid is modeled by three elements: (a) the individual connections that are characterized by their bandwidth (bytes per seconds), and for their latency (seconds), (b) the routes connections between two hosts or nodes, and (c) each host or node characterized by its computation capacity (flops). The values that are assigned to these parameters configure the network scenarios in each simulation. The type of requested resources is generated randomly among the pre-established range of resources types, for the tests, 6 types of resources were used, and the offered prices for each request were also generated randomly. For the Managers, their resource types, the initial prices and the minimum prices of them, were generated randomly, too. For the simulation, we know the lists of brokers and Managers previously.

4.3 Results

4.3.1 Experiments for 100 tasks by domain

Initially, our protocol was compared with another protocol with very basic characteristics: a centralized metascheduling implemented in SimGrid with FIFO assignment policy. In this way, like for our protocol, for the FIFO metascheduler were generated 30 simulations. The time in our approach includes the computation of the economic models of selection and assignment of resources. The metrics considered have like objective know the response times of both protocols to be able to compare them, The results are observed in the table 1.

Table 1. Comparison between the our metascheduler and the FIFO approach

	Our APPROACH		FIFO	
	Wt (sec)	Et (sec)	Wt (sec)	Et (sec)
Aveg.	467.0579933	603.8980131	472.461643	617.1684525

There are not a big difference among the results of both protocols (they differ between 1% and 2%). That is due to that the execution times of the approaches are very similar, since they are composed by control tasks that require very little computation power. But our protocol is better than the FIFO, since it offers a group of additional advantages (selection of resources or users' based on economic models) that allow improving the quality of the scheduling, without additional execution times. This way, from the point of view of the scheduling, the selection of resources and of users in the FIFO metascheduler doesn't allow choosing the best participants in the GRID. In the FIFO approach, the 30% of the occasion was chosen the economically appropriate participants (better offers) in each one of the simulations; in comparison to the 100% obtained with our protocol. All this is achieved with an execution time similar to FIFO approach.

In order to understand this, in table 3 is shown the resources selection for each approach. In particular, we have chosen at random two simulations, and we can observe that in the FIFO approach the resources with very high amounts in their prices (for tenders), or users with very low offers (for the auctions), were chosen.

Table 2. Comparative selection among models

Metascheduler	Economic Model	Selected Resource or User	Selected Price or offer
Our approach	Tender/Bid	1	200
FIFO		3	900
Our approach	Auction	1	700
FIFO		4	10

The FIFO metascheduler does not use the characteristics of the participants in the GRID, it only uses like selection policy the order of the arrivals. Our proposal requires that the participant characteristics are known, which gives a greater degree of trustworthiness. All this is obtained with similar response times.

Also, our approach is a decentralized metascheduler, in which several coordinators can be used if we have a big volume of requirements. For example, when 100.000 tasks approximately exist in a real GRID, the best form of using our protocol would be to define a group of coordinators to manage this number of tasks. The FIFO protocol, centralized, doesn't have this type of versatility; since it is not scalable (it has a centralized queue for the tasks).

4.3.2 Experiments for a random number of tasks by domain from the interval [500, 1000]

In the second experiment we increase the workload by domain (see table 3); additionally, we compare our metascheduler with which traditionally is used to manage Grid platforms, the Globus toolkit v4.0 [2, 14, 15] (see table 4).

Table 3. Comparison for tasks generated randomly by domain from the interval [500-1000]

	Our APPROACH		**FIFO**	
Task/domain	**Wt**	**Et**	**Wt**	**Et**
602	2341.3	3811.3	6812.5	3901.9
781	3131.1	4422.6	9341.0	4031.2
931	3666.2	5812.1	9113.5	5341.6
887	3341.1	4931.6	8932.9	4679.3
522	2331.6	3941.6	6510.2	3971.9
651	2562.8	3698.3	4712.6	4062.4
971	3701.1	5141.6	9291.3	5271.6
821	3311.4	4028.9	9831.2	4074.0

Table 3 shows clearly that the behavior of our approach is more efficient when the number of tasks is great. This is the scalability problem that we have spoken previously; in the measurement that increases the tasks, Wt is very long for the FIFO metascheduler. Even, our algorithm could reduce Wt by the possibility of distributing the task of coordination between several agents. With respect to the time to process the requests of the tasks (to make the allocation of the resources), our approach, in spite of being complex in calculation and to use a single coordinator, gives times similar to the FIFO approach (see Et columns).

A last test is carried out to compare our metascheduler with the schemes that uses the Globus toolkit v4.0 [2, 14, 15]. Globus allows an insurance and transparent access to the resources through a service, called GRAM (Grid Resource Allocation and Management). GRAM allows the management of resources within an organization, or dispersed geographically in several sites, using for it some type of scheduler. Between the schedulers, it habitually uses Condor, LRM, SGE, LSF, and PBS. We will compare our proposal with two of them:

- By defect, GRAM uses a scheduler that tries to execute the tasks immediately, called “fork scheduler". This is made trying to execute the

tasks locally, if they do not require special software or they do not have a specific requirement. Otherwise, it sends them to remote nodes to initiate a negotiation process. GRAM has a set of policies that handle this process of negotiation (see [15] for details).

- Previously, we have been said that GRAM can use other schedulers. A classic is Condor [15]. Condor implements a policy of scheduling by levels. A first protocol is responsible to compare the requirements of resources with the offers. A second protocol is responsible to initiate and to maintain the resource allocation. Other protocols for the data transfer and management of the fault tolerance exist; they are not of interest in this work. Returning to the description of the first protocol, Condor initiates the resource allocation periodically comparing the requirements of the resources with the offered resources. When a coincidence is found, the protocol warns to who made the request and to who made the offer, so that to start the second protocol. The first protocol has two policies in the owner of resource, one to indicate when a user can begin to use a resource, and other to indicate when it can de-allocate a resource to a user, both depend on several factors: use of CPU, day, etc.

Table 4 shows the results obtained with these metaschedulers.

Table 4. Comparison between our proposal and some schedulers used by Globus Toolkit V4.0

	Our APPROACH		CONDOR		fork scheduler	
Tasks	Wt	Et	Wt	Et	Wt	Wt
602	2341.3	3811.3	2212.5	3901.9	1811,1	3687,8
729	3012.7	3922.2	3122.1	4091.5	2634,1	3986,5
522	2331.6	3941.6	2410.2	3871.9	1898,2	3623,8
971	3701.1	5141.6	3391.3	5271.6	3110,2	5476,9
821	3311.4	4028.9	3031.2	4074.0	2881,0	4234,3

We have obtained interesting results concerning Wt, in spite of using a single coordinator. Although it is certain that the tasks wait more time in our approach, it is very near to the other approaches. Concerning Et, the times are similar, since the three metaschedulers have negotiation processes. With respect to this, the negotiation processes have implicit a criterion of optimization for the allocation of the resources of the Grid in each approach, that will improve later the execution of the tasks. Of the three approaches, Condor and our proposal deals to optimize the resource allocation on all the Grid platform, whereas the "fork scheduler" only does when there are special requirements that cannot be covered by the local nodes, which can generate important unbalances in the workload.

Two aspects to evaluate in our proposal are the cost of implantation on a tool as Globus, as well as the runtimes average of the tasks once made the allocations of the resources.

5 Conclusion

A protocol was designed and implemented for the Metascheduling in GRID platforms, based on the interaction protocols of the MAS defined by the FIPA. Our protocol is a hybrid of two schemas based on human economic models: auction and bid/tender.

The proposed protocol was compared with another protocol based on a FIFO policy. The execution time differences among them are not significant since both protocols require very similar computation capacities. Our metascheduler is better since offer a great quantity of additional advantages with regard to the FIFO protocol. The FIFO metascheduler doesn't consider the characteristics of the users neither of the owners of resources to make the planning, it only considers the arrival order. Also, our protocol is scalable because is a decentralized approach (MAS), that is not the case of the FIFO metascheduler. We can see that when we compare their results when the numbers of tasks by domain are great (more than 500 tasks).

Finally, we have compared our proposal with the schemes of scheduling used by Globus Toolkit. The obtained results are very promising because we obtain acceptable results concerning the wait times of the tasks like for the allocations of the resources.

There are aspects to value in our proposal, such as the what is the cost of the implementation of our proposal on tools of management of Grid platforms, what is the number ideal of agents of each type to use, what is the average time of execution of the tasks once assigned the resources them, what is the degree of autonomy and self-organizing of our proposal, etc.

References:

[1] Aguilar J., Leiss E. *Introducción a la Computación Paralela*, CDCHT-ULA, Gráficas Quintero, 2005.

[2] Foster I., Kesselman C. y Tuecke S. Globus Toolkit Version 4: Software for Service-Oriented Systems. *Lecture Note on Computer Sciences* (*IFIP International Conference on Network and Parallel Computing*), Vol. 3779, 2006, pp. 2-13.

[3] Buyya R., Abramson D. y Giddy J. A Case for Economy Grid Architecture for Service Oriented Grid Computing. *10th IEEE International Heterogeneous Computing Workshop in conjunction with IPDPS*, 2001.

[4] Chien C., Chang P. y Soo V. Market-Oriented Multiple Resource Scheduling in Grid Computing Environments. *19th International Conference on Advanced Information Networking and Applications.* Vol.1, 2005, pp. 867-872.

[5] FIPA: http://www.fipa.org.

[6] Caron E., Garonne V. y Tsaregorodtsev A. Evaluation of Meta-scheduler Architectures and Task Assignment Policies for High Throughput Computing. *Technical Repo*rt, Inria. 2006.

[7] Ernemann C., Hamscher V., Yahyapour R.: Economic Scheduling in Grid Computing. *In Job Scheduling Strategies for Parallel Processing, Edinburgh*, Scotland, 2002.

[8] Bubendorfer K. Improving Resource Utilisation in Market Oriented Grid Management and Scheduling. ACM International Conference Proceeding Series. Vol. 167. (2006)

[9] Odell J., Van-Dyke H., Bauer B. Representing Agent Interaction Protocols in UML. *Agent-Oriented Software Engineering* (Paolo Ciancarini and Michael Wooldridge eds.), Springer-Verlag, 2001, pp. 121–140.

[10] Aguilar J., Sumoza R.: La Metaplanificación y la GRID, *Technical Report No 4-2008*, CEMISID-ULA, 2008.

[11] REACCIUN2: http://www.reacciun2.edu.ve..

[12] SIMGRID: http://simgrid.gforge.inria.fr.

[13] Varela C., Ciancarini P., Taura K. Worldwide Computing: Adaptive Middleware and Programming Technology for Dynamic Grid Environments. *Scientific Programming Journal Special Issue on Dynamic Grids and Worldwide Computing*. Vol. 13. 2005, pp.29-37.

[14] Huedo , E. Montero, R. Llorente, I. A modular meta-scheduling architecture for interfacing with pre-WS and WS Grid resource management services, *Future Generation Computing Systems*, Vol. 23, No. 2, 2007, pp. 252-261.

[15] Feller M., Foster I., Martin S. "GT4 GRAM: A Functionality and Performance Study", In *TERAGRID 2007 Conference*, 2007.

[16] Caramia, M. Giordani, S. Resource Allocation in Grid Computing: An Economic Model, *WSEAS Transactions on Computer Research*, Vol. 3, No. 1, 2008, pp. 19-27.

[17] Zhang, W. Huang, X. Zou, F. Ma, F. O-Match: Semantic Resource Discovery in Grid Environments, *WSEAS Transactions on Information Science & Applications*, Vol. 4, No. 6, 2007, *pp. 1175-1179.*

[18] Sanchez, A. Yuste, S. Munoz, J. Garcia, S. Maqueira, J. Bruque, S. A Dynamic-Balanced Scheduler for Genetic Algorithms for Grid Computing, *WSEAS Transactions on Computers*, Vol. 8, No. 1, 2009, pp. 11-20,

[19] Fang, Y. Gao, K. Wang, X. Grid Resource Discovery Based on Semantic, *WSEAS Transactions on Systems*, Vol. 7, No. 4, 2008, pp. 277-287.