Grids of Agents for Computer and Telecommunication Network Management

Marcos Dias de Assunção, Carlos Becker Westphall
Network and Management Laboratory
Federal University of Santa Catarina
Florianópolis, SC, 88049-970, PO Box 476, Brazil
{assuncao, westphal}@lrg.ufsc.br

Fernando Luiz Koch
Intelligent Systems Group
University of Utrecht
The Netherlands
fkoch@acm.org

Abstract

The centralized system approach for computer and telecommunication network management has been presenting scalability problems along with the growth in the number and diversity of equipments being managed. Moreover, the increase in complexity of the services being offered through the networks also contributes for adding extra workload to the management station. The amount of data that must be handled and processed by only one administration point can lead us into a situation where we won't have enough processing and storage power to carry out an efficient job. In this work we present an alternative approach by creating a highly distribute computing environment through the use of grids of autonomous agent to analyze large amounts of data, which reduce the processing costs by optimizing the load distribution and resources utilization.

1. Introduction

Grid computing has emerged as an initiative that provides the resource sharing forming a large scale distributed system and making possible the resolution of complex scientific and commercial problems [1]. By distributing the workload of its applications, a user can take advantage of a computational and storage capacity that, in a traditional environment, would be financially impracticable.

This resource sharing and aggregation type differs from currently available technologies on the Internet, in the way this integration takes place. Its aim is to provide a cheap, efficient, easy, pervasive and large-scale access to resources. The basic differences between grid computing and a traditional distributed system are the great heterogeneity of resources, the dynamic environment characteristics and the high latency of the networks that interconnects them. A grid is an aggregation form that can occur in several ways, including through the cooperation and negotiation mechanisms provided by agents.

In the traditional workflow of a network management system we have a situation in which data is collected from network devices using some management protocol; the collected data is analyzed and; finally, it is transformed into high-level management information. Later on, managers make their analyses and take actions to correct weak points of the system [2]. Rule-based and inference systems could be used to analyze this data, extract necessary information and identify eventual problems. However, the consolidation of these data into management information is an intensive task and it consumes a great processing power. As the environment grows, the efficiency of a centralized system decreases and the cost of hardware increases.

We present an alternative architecture, based on agent grids, distributing, from the management, the workload of those intensive tasks. We will present some results from our work on grids and its possible use in network management.

This work is divided into 5 sections: In section 2 we present grid computing, giving a brief overview and, agent-based grids and their possible use in computer network management; in section 3 we present our proposal for an agent-grid architecture [3]; we compare our architecture with other approaches in section 4; finally, in section 5 we present our conclusions and future works to be achieved in this line of research.

2. Grid Computing and Agent Grids

In the computational literature, a grid has appeared as a hardware or software infrastructure capable of grouping components, providing new functionalities, starting from existing components [4].

This research area has shown results in scientific works that require intensive computation and a computing power that cannot be achieved in a traditional environment. Most of these applications can be decomposed into lesser activities, are not sensible to the high latency that can exist in the communication among the elements of the grid, and their sub-tasks do not require the exchange of a large number of synchronization messages. Also exist works that attempt to provide middleware to facilitate grid computing [5].

In [6] an organization is shown based on three levels: computational grid [1],[7], information grid and knowledge grid. When we find the term "agent grid" related in some work, we perceive that two distinct situations exist: first, we have a grid of agents as a system to connect heterogeneous components [8] and in the second one we perceive a grid as a kind of middleware that provides interoperability among different agent platforms [9].

It is evident that the concepts involved in agent technologies and multi-agents systems can be used for the grid construction and, as a way of making grid-computing possible. It is possible to provide large-scale resource sharing and integration. Multi-agents systems, through cooperation and negotiation concepts, can also aid in the load balance. Besides these means, agents are tolerable to the latency that can exist in communication in systems of this load.

In our work we use concepts of the works presented above in the elaboration of our own architecture for the agent grid construction, directed to computer network management. It is based on small agents and uses the AgentLight [10], following the specifications of FIPA [11] in its conception.

2.1. Agent Grids in Network Management

As computer and telecommunications networks grow, the complexity of their management and the information flow in management systems also grow. As described in [2], a traditional network management system presents a workflow similar to the one of Figure 1. In this model, first, the data is collected from managed network devices through a management protocol. After that, the information is analyzed and finally it is condensed, thus creating real management information. Later on, based on this information, the network manager can carry out his tasks and correct weak points in the system.

In a network environment formed by a large amount of equipment, the volume of collected data that have to be analyzed is proportionally big. The task of transforming these data into information and reports that demonstrate the network operation service conditions or that indicates eventual problems could become an intensive task, demanding a great processing power of stations in the network management.

Many approaches based on Artificial Intelligence concepts had been proposed, with aim of increasing network management efficiency. Some systems have their functioning based on production rules and inference. Although they assist in the management, these applications also grow in the direction of holding a large number of rules to identify and deal with problems, thus increasing the management system complexity, decreasing its performance and causing a great rise in the hardware cost of management stations. In such model, the management station carries out the role of information collector.

A negative factor in this scene is the non-existence of parallelism in the analysis. This situation can be improved if we have a large number of agents, possibly located in different stations, analyzing information simultaneously.

Taking computer network management as a grid application, where there is a vast amount of data that needs

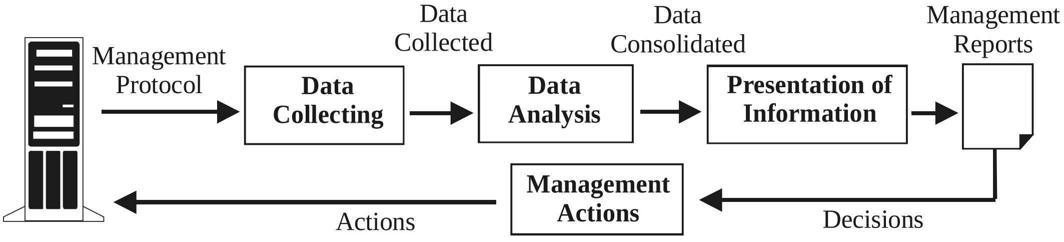

Figure 1: Network management workflow.

to be transformed into management information, we find a scene where agent grids could be applied. We can have a large quantity of analysis rules that we can keep and use in such a system. It is possible to effect a distribution and load balancing of the analysis of collected data based on the resource availability and capacity of the grid resources. If the system requires a greater processing capacity, we need only to add it to the grid. In this way, we can have significant profits and a considerable reduction of hardware.

The agents of the grid can learn new rules. With this, we increase knowledge concerning management. Many agents learning new rules can produce a greater synergy in the system.

3. Agent-based Grid Architecture for Network Management

In this section we present our agent grid architecture to computer network management. Considering the traditional management workflow, presented in the second section, we identified some distinct tasks: data collection from network devices; the classification of this data; data analysis to transform it into management information; and information presentation. In a first view of the architecture, we present the agent grids that compose the management grid and, after that, we present details on each point of the architecture. Figure 2 shows a sketch of this architecture.

3.1. Collector Agent Grid (CG)

The role of the collector agent grid is to carry out the data-collecting task from network devices of the managed network, through management protocols or in some other way. We will call "interface" this agent ability of collecting data through a management protocol. A collecting agent can have an SNMP (Simple Network Management Protocol) interface or to use a command line utility, for instance.

The information extracted from network devices could have quite heterogeneous formats and therefore it is necessary to create a common representation for these data. This representation can be made using XML and ontologies [11],[12]. In this case, we guarantee that the agent grid, which will receive it, could correctly interpret this information.

A collector agent has a knowledge base with rules that allow collecting data using the requested protocol. Those agents can have one or more goals that consist of extracting managed object values from one or more pieces of equipment in the network between time intervals. The collector grid can contain agents that execute some local information analyses. The information collected by the collector grid, already in a standard format, is sent to the classifier grid, through any existing protocol – such as SMTP or HTTP.

3.2. Classifier Agent Grid (CLG)

In a relatively large network, we can have many data collectors. Consequently, we will have data arriving from many types of equipment. The classification grid carries out the task of classifying and storing this information in a more organized and easy-to-retrieve form. A file containing collected data sent by one grid could contain collected values from many managed objects in heterogeneous equipment. It is important to organize this information before carrying through any analysis. From the description above, it is clear that the classifier grid performs parsing, classification, indexing and storing data tasks. The aims of grid classifier agents consist of classifying and storing this information correctly, separating it in a way that facilitates its distribution and analysis (data-clustering). After that, it sends a FIPA ACL message [11] to the processing grid.

3.3. Processor Agent Grid (PG)

Grid of processing or analysis agents is the most important part of the architecture, and it is where we could find a big number of development challenges. It is responsible for transformation of collected data into management reports. The main problems that can arise are related to division of analysis activities, resource management, and load balancing and fault tolerance. The result of analysis is management information, which forms management reports, and alerts that will be sent to the user by the interface grid.

This grid is composed of containers of agents, which are distributed among many computers. In this way, we have a dynamic environment where new resources can be added to or removed from the grid. If the grid needs to increase its processing power, we can do this by adding new resources.

The processor grid can carry through a series of analyses on multiple levels. Some examples of these analyses, and how they occur are described as follows:

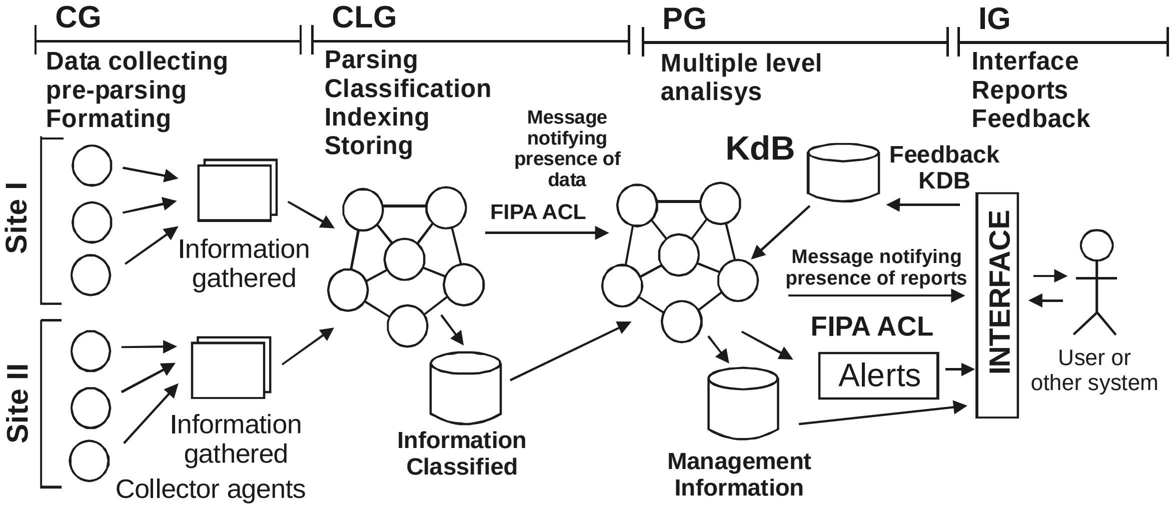

Figure 2: General architecture of the system.

- When receiving a message from the classifier grid, indicating the presence of data, the processor grid makes the containers and agents, based on their rules, look for problems in this data. This first analysis does not consider the information previously stored in the database and it does not look for any relationship among the facts.
- The agents belonging to this grid can consolidate data, extracting information previously stored. They can look for problems and group data, so as to generate some information that will be used for assembling management reports.

On the third analysis level, the agents of the grid can seek relationships among the facts on the database, having a higher-level vision of the information. In this way, they can identify problems that arose through the crossing of information from a whole complex of equipment and not just isolated data.

In this context, the main difficulty is providing a division of the analysis activities in the grid. The division of distributed application datasets has been studied for several years, and in network management, this division can be somewhat complicated, and has to be treated in such a way that there are no losses of meaning in the information. This is precisely the role of the classifier grid - it prepares the information so that the analysis tasks can be easily distributed and the processing grid root co-ordinates this distribution, functioning as a broker in the system.

The analysis grid root receives a message from the classifier grid, indicating that there is data to be analyzed and that this analysis needs to be distributed among the containers of the grid. To do this, there must be a register of containers that compose this processing grid and their abilities, so that this distribution can be carried out. In Figure 3 we have an abstract view of the division of analysis activities.

3.4. Interface Agent Grid (IG)

The grid of interface agents is the communication channel between the grid and the network manager. It is also a way of receiving feedback from the user and supplying it to the system. The interface is both an input and output channel to the system. Through an interface the user can receive the processed information from the processing grid and the alert messages. It is possible to program new reports and to interact with the knowledge base, defining new rules and goals, and to modify existing goals.

This interface must be flexible and multi-protocol, allowing several ways of interaction according to the user's necessity - for example, HTML pages, email, chat, XML/HTTP - and the return from the user also can be given in any of these ways. This interface, besides being the communication channel with the user, can provide an input for another system, or this system serve as a feedback channel for this system, from which can learn new management rules and to transmit them to the grid. The interface grid can learn new rules through the analysis of user's preferences and necessities, in this way customizing the system.

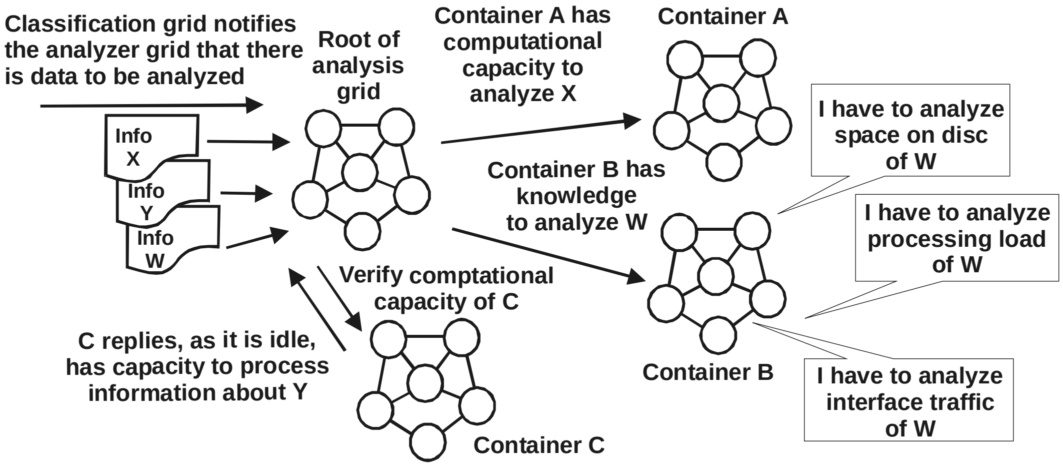
Classification grid notifies the analyzer grid that there is data to be analyzed
Root of analysis grid
Container A has computational capacity to analyze X
Container A
Info X
Info Y
Info W
I have to analyze space on disc of W
Container B has knowledge to analyze W
I have to analyze processing load of W
Verify comptational capacity of C
C replies, as it is idle, has capacity to process information about Y
Container B
I have to analyze interface traffic of W
Container C

Figure 3: An example of analysis tasks division in the grid.

3.5. Load Balancing in Agent Grid

One advantage of using agent grids in the collected data analysis is the intelligent distribution of the workload for the analysis in grid. When receiving a large amount of data, the grid can carry out the distribution between containers, based in the following principles:

- Distributing it among containers with knowledge to process it;
- Utilizing resources that have computational capacity to process it and;
- Using resources that are idle.

To provide this distribution service, first the grid must know the resources that compose the grid at a given moment and their capabilities. This information is stored in the grid root using directory service. When a new container is added to grid, it can register itself and inform the capacity of the resource in which it is running. The ontology that is used when the container informs this profile is defined by FIPA [11]. This interaction can be seen in Figure 4, where container 1 is added to grid: it informs the profile of the resource in which it is located, and the services that it is capable of providing. The root registers this information in the directory D1. When it is necessary to attribute some activity that involves workload distribution, the root uses this information to select the resources capable of processing it.

When the processing grid receives a message, informing it that there is data to be processed, besides giving this static information, it could request the current profile of the resources or negotiate [13] with containers concerning the possibility of sending information to be processed by them. In this way it can use negotiation protocols established by FIPA [11].

In the same way, the data consolidation and the search for relations between facts can be distributed among several containers in the grid. The analysis grid can select resources capable of performing these tasks and delegate the processing tasks. The root could use the directory information to select capable containers.

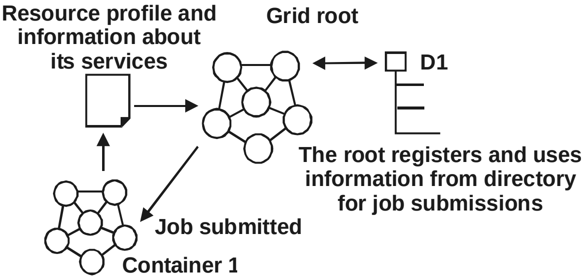
Resource profile and information about its services
Grid root
D1
The root registers and uses information from directory for job submissions
Job submitted
Container 1

Figure 4: When a container is added to the grid, it will inform the profile of the resource on which it is running.

4. Architecture Advantages

We can evidence some advantages of architecture using agent grids in comparison with the architecture without grids. In the management without agent grids we have a similar scene to the presented one in Figure 5.

In this model, using agents, we would have collectors extracting data from several kinds of equipment, a classifier storing these data in a repository and an analyzer agent analyzing this information. We also have an interface with the user provided by interface agents.

Each network has a similar structure and there's no relation among different sites. There is no integration in this information; and no high level analysis can be carried out. Moreover there is no kind of workload distribution. The only possible evolution of this system would be the integration of knowledge bases.

Another disadvantage is that it does not scale well. The only way to increase the power of information analysis is to increase the power of hardware that maintains the system. We have an expensive and unnecessary addition and can add to this the fact that these resources can be allocated and are not being used even when there is no information to be analyzed.

In the architecture we are proposing we would have a more flexible and scalable structure, as it was described in Figure 2.

With agent grid, we merely have a common flow that data must follow. The main advantages are related to the distribution of workload processing and the use of idle network resources. We can also have a number of further advantages:

- Extensibility - The architecture can be expanded on any of its levels. New containers or agents can be added; and new goals can be attributed to the existing agents. In the processing grid, we can add new containers to different kinds of equipment, according to the processing and analysis need we wish to meet. A container with several agents can be added to the grid to carry out a more specific type of analysis or to identify a particular type of problem.
- Analysis time reduction - data analysis time can be reduced, because we will have several containers simultaneously analyzing data. In this way a large number of rules and problems can be verified in a shorter period of time. We can affirm this, based on the distribution of the activities and the possibility of aggregating resources in the grid. In a traditional management application, the only way to speed up the processing of this information would be to increase the hardware capacity of the management station.
- A large number of analysis rules - the system can hold a large number of rules. The agents can carry out multiple-level analyses, by crossing data using a number of rules that would not be retained and verified in traditional ways.
- Intelligent load balancing - another advantage of intelligent agents is that they have the knowledge as to how to distribute the processing load, through the discovery of available resources or the utilization of cooperation and negotiation concepts.
- A greater processing power and its better distribution - we have a greater processing power resulting from resource aggregation. We also have a more cost-effective use of resources, preventing a situation where some are idle while others are overloaded.
- Shared knowledge - in a system where there is management of several networks, shared knowledge is an important advantage.

In spite of its advantages, we can perceive that the utilization of agent grids appears to be most attractive when the volume of information to be analyzed on the network is relatively large. In less busy environments, traditional approaches or those based on multi-agent systems, still prove to be more cost-effective.

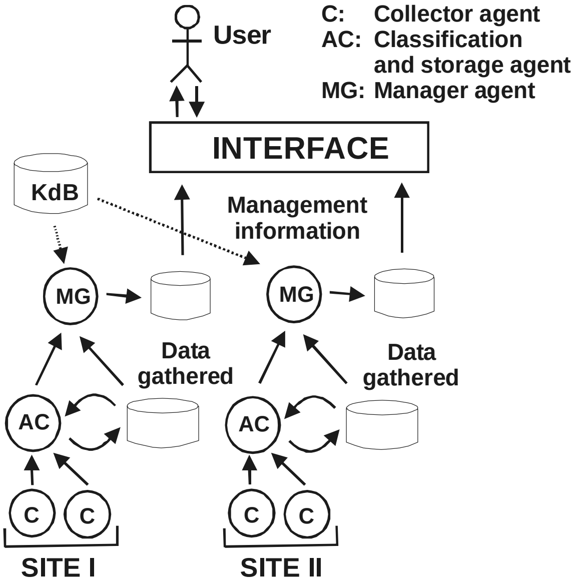

Figure 5: Architecture without agent grids.

There is also a point where the utilization of agent grids is no longer advantageous. As previously mentioned, the division of data and analysis activities may bring about a serious loss of meaning in the information obtained.

4.1. Advantages in grids of agents applications

We can take as an example a network management system whose goal is to monitor the network stations' performance by collecting data such as processor usage memory availability, available disk space and the list of processes being executed at probing time. After collecting the data, a parsing module extracts the relevant information passing it to the analysis module, which runs the inferences rules on top of the collected values.

For this example, in order to evaluate the management system performance we will take in consideration three measurable values: communication network, processor and disk utilization. For each management activity we will attribute a relative value as described on Table 1.

Tasks	CPU	Network	Disc
Request A	10	5	
Request B	10	10	
Request C	10	15	
Parse A	15		
Parse B	15		
Parse C	15		
Storing	10		10
Inference A	20		5
Inference B	20		5
Inference C	20		5
Inference AxBxC	40		8

Table 1: Relative times of management tasks

Considering a scenario where we have 10 requests of each type, as for example, after some simulations we reach the results presented on Figure 6 that shows the information about the resources utilization for each management model.

In Figure 6a, the model implementing centralized management will present higher network utilization as the data transmitted between the resource and manager station is in raw format, being parsed by the manager itself. Moreover, as there is only one host involved in all activities, its processor becomes the bottleneck.

The second model, whose results are presented at Figure 6b, is the one that implements a multi-agent system. Here, there are more than one data collector hosts, which also carry out parsing tasks where unnecessary information is removed before the data is transmitted to the manager host. Moreover, in this model the collectors have the knowledge about how and when to collect certain value, releasing the manager from making these requests. These features lead to reduction in communication traffic but keep a centralized data analysis structure, which, again, is the system bottleneck.

Finally, in the model where a grid of agents is applied, besides the distribution of collection and storage tasks, we can also implement a distributed information analysis. For this example, we have 3 collectors hosts running the information collection, one host for storage and the two hosts running the inference tasks.

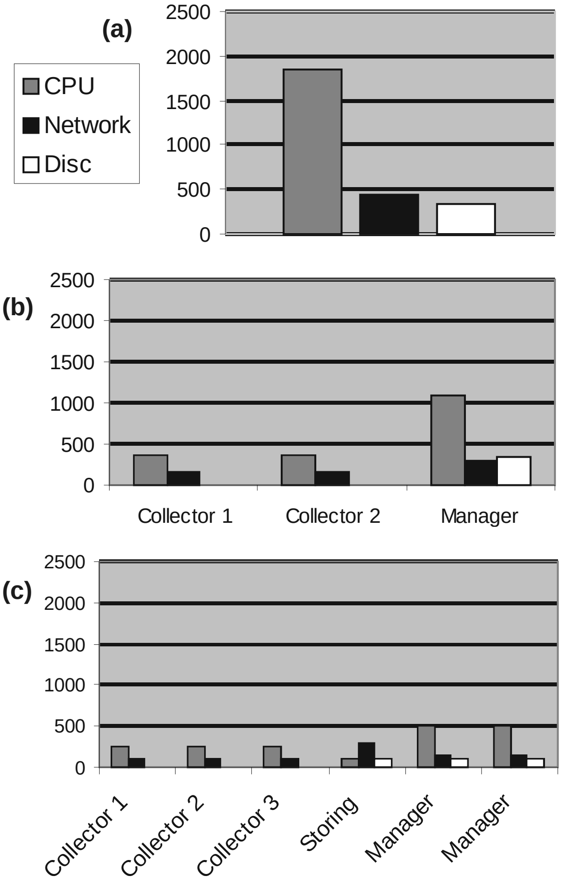

Figure 6: Compared performances from three architectures: (a) centralized management; (b) multi-agent with 2 collectors; (c) grid of agents.

As we can see from Figure 6c, the highly distributed processing implemented through the agent grid architecture led to extensive work load balancing thus improving resource utilization and allowing higher scalability.

5. Conclusion and Future Works

In this work we have presented an agent grid architecture applied to computer network management. We have also shown the concepts involved in grid computing and the advantages of its utilization in the processing and analysis of management data.

Since the concept of agent grids is relatively new, its utilization in network management must be increasingly more exploited, and this is first work in the area. We consider it important to investigate the possibilities of its utilization and propose an agent grid architecture for

Network Management, validating this approach theoretically and through an implementation of a prototype that is currently being developed. We will enhance the activities carried out up to now, and will proceed with the following work:

- Developing better prototypes, showing the advantages of utilizing agent grids by performing more effective measurements.
- Determining more clearly the point at which the utilization of an agent grid becomes more advantageous and the point where it ceases to be.
- Conducting studies on load balancing on the processing grid, also seeking efficient means for the division of analysis tasks. To make further measurements of the processing capacity achieved with a processing grid and its advantages compared with traditional techniques.
- Investigating further the utilization of mobile agents in data analysis and in load balancing. Agent mobility allows for a migration of analysis activities attributed to them, improving the utilization of resources.
- Improving the efficacy of forms of storage, replication, indexing and recuperation of management data by agent grids.

We have shown the advantages of applying agent grids in network management, reducing costs and providing a more efficient processing and analysis through the architecture developed. There is still a great deal to be done and we believe that the continuity and enhancement of this work deserve special attention from the whole scientific community in this area, since it involves a new and promising approach.

References

[1] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan-Kaufmann, Orlando (1999).

[2] F. L. Koch and C. B. Westphall, *Decentralized Network Management using Distributed Artificial Intelligence*, Journal of Network and Systems Management, vol 9, Plenum Publishing Corporation, USA (2001) p. 291-313.

[3] M. D. Assunção, C. B. Westphall, F. L. Koch, *Agent Grid Architecture Applied to Computer and Telecommunication Network Management*, 21th Brazilian Symposium on Computer Networks, Natal (2003), in Portuguese, (to appear).

[4] O. F. Rana and L. Moreau, *Issues in Building Agent-Based Computational Grids*, 3rd Workshop Of The UK Special Interest Group On Multi-Agent Systems, Oxford (2000).

[5] I. Foster and C. Kesselman, *Globus: A Metacomputing Infrastructure Toolkit*, Intl J. Supercomputer Applications, 11(2) (1997) p. 115-128.

[6] K. G. Jeffery, *Knowledge, Information and Data, A briefing to the Office of Science and Technology*, `www.itd.clrc.ac.uk/Publications/1433/KnowledgeInformationData20000124.htm`, (2000).

[7] R. Buyya, *Economic-based Distributed Resource Management and Scheduling for Grid Computing,* PhD Thesis, School of Computer Science and Software Engineering, Monash University, Melbourne, Australia (2002).

[8] O. F. Rana, and D. W. Walker, *The Agent Grid: Agent based resource Integration in Problem Solving Environments*, 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation, Lausanne (2000).

[9] N. J. E. Wijngaards, B. J. Overeinder, M. van Steen, F. M. T. Brazier, *Supporting Internet-Scale Multi-Agent Systems*, Data and Knowledge Engineering, 41 (2-3) (2002) p. 229-245.

[10] F. L. Koch and J-J. Meyer, *AgentLight: developing logic-based autonomous agents for small devices*, I Workshop of Thesis and Dissertations in Artificial Intelligence, Brazil, (2002).

[11] FIPA, *The Foundation for Intelligent Physical Agents,* `http://www.fipa.org.`

[12] T. R. Gruber, *Toward principles for design of ontologies used for knowledge sharing,* Technical Report KSL-93-94, Knowledge Systems Laboratory, Stanford University, (1993).

[13] W. Shen, Y. LI, H. Ghenniwa, C. Wang, *Adaptive Negotiation for Agent-Based Grid Computing*, Proceedings of AAMAS2002, Workshop on Agentcities: Challenges in Open Agent Environments, Bologna, (2002) p. 32-36.