

Cloud computing and multiagent systems, a promising relationship

Fernando de la Prieta and Juan Manuel Corchado

Department of Computer Science and Automation Control, BISITE Research Group. Edificio I+D+I, C/ Espejos s/n, 37007, Salamanca (Spain)

{fer, corchado}@usal.es

Abstract The technology industry is presently making great strides in the development of the paradigm of Cloud Computing. As a result, the number of both closed and open source platforms has been rapidly increasing. They all have a similar architecture. A cloud computing platform can be considered as an open system because it is dynamic, heterogeneous and unpredictable. Theoretically, this kind of environment is the most suitable for agent technology. Under the frame of this chapter we explain a new agent-based proposed architecture to control a cloud computing platform that satisfies the hypothesis.

1. Introduction

Historically speaking, Cloud Computing (CC) was first introduced as a term by Professor R.K. Chellappa [14], who suggested that the computational model of the future would be much more closely tied to economic interests than to the limitations imposed by technology. Although almost a decade has passed, this notion could be considered too utopic given that it was the continual advances in technology that drove market and business models. Today, it seems clear that, despite the many obstacles yet to be overcome, technologically speaking, the rate of technological innovation is determined by the macroeconomic interests imposed by the large technological companies, as the CC paradigm has shown since its birth in 2007 [32]. These economic interests have led to the concept of CC being as closely tied to a business plan as to research.

The marketing model used in the CC paradigm is also innovative, as it is based on a *pay-as-you-go* concept [5], just as with any other traditional public service (light, water, gas, etc.). In this sense, CC follows in line with the model proposed by the abstract *Utility computing* paradigm [45] in which users must negotiate and previously establish a *Service Level Agreement* (SLA) in order to access services [2]. Once this contract for computing goods has been established, both the users (through regular payments) and the CC system (by maintaining the service) are obligated to follow through with their agreement.

New possibilities at a technological level lead to the birth of a new concept, elasticity [16]. This concept is based on the *just-in-time* production method [24], which references the manner in which the services (computational) and the resources they require are produced. Thus, the services produced within the framework of CC only receive the amount of resources they need to maintain a uniform level of quality while immediately responding to demand [60] [57].

Elastic algorithms should not only provide an elastic model (increasing and decreasing), but should do so following a model that can also reduce operational costs in the CC environment. However, existing research in the state of the art are based on methods that use centralized algorithms based on mathematical and heuristic models [64] [40] [28], neither of which can ensure the efficiency of the system or, even less so, its availability in the event of a system failure. Therefore, this type of algorithms should evolve toward a model in which different agents are represented within an uncertain environment, which forces them to interact and share information with their equals. This would allow the algorithms managing the resources to be distributed throughout the system, which would therefore facilitate their implementation regardless of the size of the data center.

In this context, the theory of agents and Multiagent Systems (MAS) [62] can provide a new model for managing CC systems based on the distribution of responsibilities, flexibility and autonomy. Managing the functions of the nucleus of a CC system through an agent-based model allows the resulting platforms to be much more efficient, scalable and adaptable than they currently are. However, joining both computational models (MAS and CC) is a great challenge, given the difference between the two models. However, since the CC system is considered an open system and the application of MAS in open systems is a recognized challenge in which there has already been a notable rate of success [65] [43] [61] [12] [42] the union of the two models is an achievable challenge within the framework of this research study. MAS that have been designed through organizational models are considered especially effective within the framework of this project. This is due to the fact that they can already provide advanced and innovative solutions [19] that make it possible to exploit differentiating elements when it comes to providing flexibility, capability and response speed as part of a strategy defined by the provision of customer satisfaction [47]. Finally, the use of MAS enables continuing research in techniques, tools and methodologies that allow incorporating intelligent characteristics, such as autonomy or experiential learning, on a CC platform.

The scope of the present study is based on the premise that since technology is continually offering new possibilities, it is necessary to design new resource distribution models that incorporate these new characteristics. The proposed model is based on agent technology and is able to distribute computational resources throughout the entire CC environment and allowing the distribution of its complexity and associated computational costs. The development of a monitoring and management system for a CC environment that follows a MAS-based design model differs from traditional models that control this type of platform, which

tend to have a centralized decision-making process. This model allows the decision-making process to be carried out right where the information is gathered, on the base that provides local knowledge, which has made it possible to design agile control processes based on uncertain information, prior knowledge, and the interaction among similar agents. This new approach in monitoring and, in particular, controlling the CC system, makes it possible to incorporate the new characteristics, as previously mentioned, that virtualization has to offer.

This document is organized as follows: the following section presents an introduction about CC, then section 3 introduce the reference architecture proposed by NIST and the characterization of a CC environment. To end with the review of the state of the art, Section 4 presents the relationship between CC and MAS. Section 5 presents our proposed architecture to monitor and control a CC based on MAS and, finally, last section present the evaluation, discussion and conclusions.

2. Cloud Computing

The rapid growth of the paradigm CC is to a large extent tied to the more sophisticated developments that have been reached by related technologies. These technologies have contributed from different points of view (hardware, software and business models), as shown in Table 1.

Table 1. Cloud Computing and related technologies

Previous technology	Hard.	Soft.	Buss.	Influence
Grid Computing	X			High value add to the end user services
Utility Computing			X	Origin of the business model for the CC paradigm, based on the provision of computational services on demand.
Autonomic Computing		X		Define the basis of self-monitoring and automated control of the computational environment
Virtualization	X			Underlying software environment that allows the rapid provision of services
High Availability	X	X		Grouping data centers in clusters and high availability techniques
Service Oriented Architecture		X		Model the provision of computational services at the software/hardware level.

The increasing importance of this paradigm has led to a vast number and variety of definitions [35] [54] [58] [21]. The most generally accepted definition, and in our opinion the most precise from both a technical and functional point of view, is proposed by NIST (*National Institute of Standards and Technology*) [35]. In this definition, Mell *et Grance* propose that *Cloud computing is a model for enabling*

ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

According to NIST, the services must contain the following characteristics [35]:

- Services on demand, meaning that services, regardless of their type, must be provided automatically and without human interaction according to user demand.
- Availability of services through the Internet, meaning that clients should access the services through the Internet and providers, as a result, must use this medium to provide their services.
- Availability of resources, meaning the provider must be able to offer services independently of their demand, using physical or virtual hardware resources assigned dynamically to each resource and reassigned according to demand. In this respect, there are authors such as [66] [11] who speak directly of high availability services, technology closely related to high availability computing.
- Elasticity, meaning that the different resources should be provided elastically and even automatically according to demand.

Together with the definition, NIST proposes four deployment models and three types of services (*Capabilities*). The types of services (capabilities), which can be any type of computational resource, are presented first:

- *SaaS. Software as a Service.* These services are available when the provider offers its applications to the consumer. The applications are executed directly on the cloud infrastructure. Although this model includes advantages such as ubiquity and the use of light clients, it also entails a set of weaknesses directly related with the fact that the consumer loses control of the infrastructure.
- *PaaS. Platform as a Service.* The provider provides the tools that the users need to create their own applications. These services include programming, libraries, tools, etc.
- *IaaS. Infrastructure as a Service.* The type of capability provided to the consumer is a type of hardware, such as processing, storage, network, etc.

In line with the definition proposed by NIST, the last step required to characterize the CC systems consists of identifying the four proposed deployment models. First is the *Private Cloud* model, which is used when the CC infrastructure is used by a single organization, which can in turn include different consumers. Second is the *Public Cloud*, which is associated with infrastructures for open use by the general public. Based on these two models (public and private), various authors propose the existence of a hybrid models named *Virtual Private Cloud* [27] [66]. In which private CC are built over the infrastructure services of the public platforms.

As a result of this hybrid model, there are multi-holdings [9][66] when various CC environments share the same public infrastructure.

As for the other two models, the third is named *Community cloud* and is available when the CC infrastructure is used by a specific group of consumers or organizations that share a common interest. Finally the *Hybrid Cloud* model permits the combination of any of the other three models presented. In this case, we can talk about the interoperability between platforms [38], which are generally complex since each CC platform is closed.

A CC system is undoubtedly complex and is part of an open environment composed of different technologies, users and economic interests that give way to a new computational model that has revolutionized the way in which services are offered through the internet. For all of these aspects to coordinate and work together to achieve common objectives, bearing in mind any existing technical limitations, complicated architectures have been developed.

2.1. Cloud Computing Architecture

NIST also proposes a reference architecture [31], shown in **Fig. 1**. The details of the architecture, focusing on the main roles in the paradigm, are presented below:

- **Cloud Provider.** This role is in charge of (i) *Coordination of Services* (IaaS, PaaS, or SaaS), which are provided to third parties and therefore require an underlying infrastructure. This role must also (ii) *facilitate the management of the services* offered through a support layer for marketing and business. Finally it is responsible for (iii) *security* and (iv) *privacy*.
- **Cloud Auditor.** This agent is capable of monitoring the service and ensures that the agreed-upon requirements are being met.
- **Cloud Broker.** This agent acts as an intermediary between consumers and providers. It aims to seek and provides the services best suited for the consumer's objectives, regardless of the provider.
- **Cloud Carrier.** Provides connectivity between provider and consumer
- **Cloud Consumer.** Who is the end user of the services provided by the CC platforms

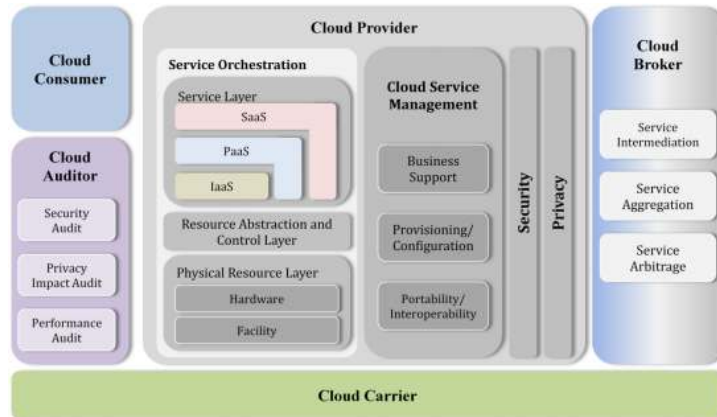


Fig. 1.- Reference Cloud Architecture for NIST[Liu *et al.*, 2011]

2.2. Characterization of a Cloud Computing environment

Given the complexity of the environment, as well as the different artificial and human components involved in this context, it is necessary to define how the services will be offered at a technical level. For this reason, and following the CC model as presented in Figure 2, each software service for the platform, at the PaaS or SaaS level, can be deployed simultaneously on various virtual machines (nodes or *workers*).

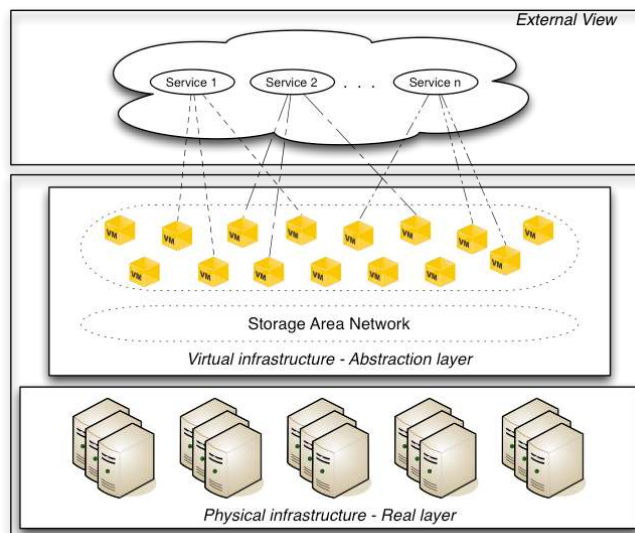


Fig. 2.- Deployment model in Cloud Computing

The intrinsic characteristics of the service are therefore determined by the template of the virtual machine that offers the service. The template will make it possible to instantiate and modify, in execution time, the resources associated with a specific service. Consequently, every template for a virtualized node associated with any service k (VM_t^k), will be described through a set of properties: identifier, minimum assignable memory, minimum number of assignable CPUs, type (hardware or software service) and state, which determine whether it is balanceable:

$$VM_t^k = \{ID^k, M_{min}, vcpu_{min}, type, state\}$$

Each physical server will host a set of virtual machines. This will allow every physical server in the system to have at all times an associated matrix with information regarding the physical machine as well as the different virtual machines that it houses at any given time. This matrix is an instantiation of each physical server that has been executed and can be used to determine which services will receive resources and the amount of resources to be lent.

$$exec^e = \left\{ \begin{array}{ccc} & PR^e & \\ VM_1 & \dots & VM_m \end{array} \right\}$$

The user will, in turn, receive a set of computational services of any kind, including software, platform and infrastructure, through a Service Level Agreement (SLA). A service agreement will be formalized for any given user j ($ServA^j$) as the combined set of user agreements established for each service i on an individual basis (SLA_i^j) [20] [39].

$$ServA^j = \bigcup_{i=0}^{i=m} SLA_i^j$$

Using this simple expression, the goal for achieving an adequate model of the context consists of measuring of the quality of services offered. Various related studies can be found in the current state of the art [2][22]. Within the scope of this study, the metrics that are directly dependent on the underlying computational resources will be used by the service, and the response time for a request will be selected. In other words, for a service k with a set of methods (r_i^k) which make up the service API, the quality of each request that forms the service API is determined by the following expression for the size of the response (s_i^k) and the corresponding transmission time (t_i^k):

$$QoS^k = \{\overline{r_1^k} \dots \overline{r_i^k} \dots \overline{r_n^k}\} \text{ where } \overline{r_i^k} = \frac{1}{m} \sum_{i=1}^m s_i^k / t_i^k$$

Using this very characteristic deployment model, the control system in a CC environment should vary the computational resources assigned to each service according to the demand that exists at any given time, making it possible to maintain the QoS levels in each of those services. In this regard, the greatest advantage of virtualization is that the assignment of resources at any instance of execution can be reconfigured dynamically, which makes it possible to elastically modify the amount of resources associated with each service in execution time. In terms of requests for a specific service, the demand is balanced among the different virtual machines that are associated to the service. Additionally, the weight of each virtual node on the scale can vary dynamically in execution time. Therefore, the elasticity is based on modifying the (virtual) resources that have been assigned to each service dynamically according to demand.

3. Cloud Computing and Multiagent systems

As noted above, the CC computational paradigm has grown strongly in recent years; its development has led to the advancement of a large number of platforms, both public and private. According to Gartner¹, its wide acceptance within the business world [36], as well as its easy and rapid integration with traditional technology architectures [18] has led to the rapid development thus far. Likewise, the pay-as-you-go marketing model [5] similar to traditional utility products has also been a key factor in its rapid evolution.

The definition provided by NIST [35] stands out among the others since it not only defines the paradigm, but also its characteristics, services and deployment models as well. Nevertheless, in spite of the quality and breadth of this definition, more than a few others [34] have noted that the definition is not sufficiently ambitious. There is a vast number of technological innovations that have emerged within the scope of this paradigm [33] [7] that are nevertheless not a part of existing platforms. The majority simply focuses their efforts on providing hardware infrastructure services through the use of the underlying virtual technology, without considering the capabilities of the higher levels such as platform and software.

Given these weaknesses and vulnerabilities, as well as the limitations of CC platforms, and in line with the hypotheses proposed in this study, a MAS framework based on VO has been selected to deal with these obstacles. Although one may initially consider these two distributed systems (MAS and CC) to be incompatible, a detailed analysis demonstrates that they are in fact not only complementary, but share considerable synergy between them. First of all, CC environments can cover the computational needs for persistence of information and the computing potential that MAS require for different applications such as data mining, management of complex services, etc. Additionally MAS can be used to create a

¹ <http://www.gartner.com/technology/home.jsp>

much more efficient, scalable and adaptable design for the CC environment than what is currently available. Finally, the use of MAS in the framework of the design for CC systems provides this paradigm with new characteristics such as learning or intelligence, which makes it possible to develop much more advanced computational environments in all aspects (intelligent services, interoperability among platforms, efficient distribution of resources, etc.).

The number of studies that can be found on the state of the art relating CC with agent technology is actually quite low. However, this tendency is changing and it is becoming increasingly common to find studies and applications focused on this field. Despite the limited number of studies on the matter, **Agent-based Cloud computing**, or the **Agent-based Cloud platform**, is becoming a common concept, mentioned by various authors in recent years [51][52][25][26][48][49][56].

The present study proposes a new classification from the point of view of CC, based on the reference architecture proposed by NIST [31] and the different responsibilities of each of the roles that participate in the CC paradigm as identified in the architecture: *Provider, Consumer, Broker, Carrier and Auditor*. The role of Cloud Carrier is not included in this classification since it is the agent responsible for providing the transportation of information and does not provide any possible functionality between MAS and CC. Figure 3 presents a general overview of the synergy between both paradigms

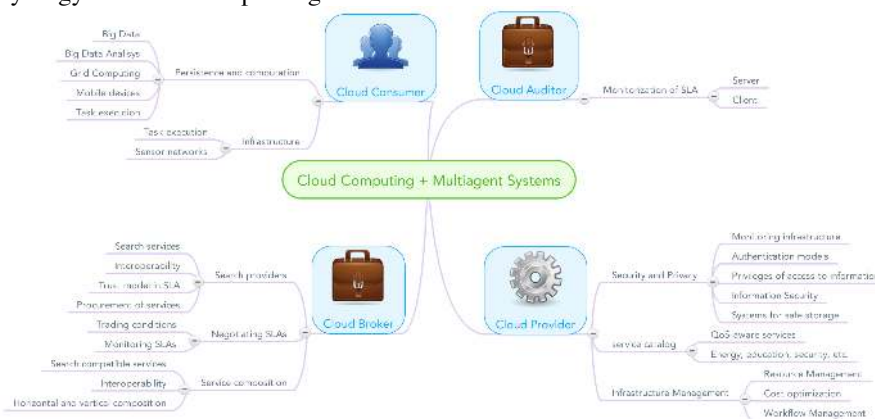


Fig. 3.- Cloud Computing and multiagent systems

As shown in **Fig. 3**, when a MAS takes the role of Cloud Consumer, the CC environment offers high performance technology that can be used by the MAS [52]. These services allow and facilitate the application of MAS in a wide variety of complex applications due to the ability to expand the reasoning and knowledge model of traditional MAS. The temporary restrictions are reduced and different high performance storage models are available [46] [17] [15] [29].

When MAS takes the role of *Cloud Auditor*, or *Cloud Broker*, it serves as a third party (intermediary) that can intervene in the existing business relationship between user and consumers. This ability is widely applied primarily in the *Cloud*

Broker role, significantly highlighting the search features offered by MAS [48], the selection of services [2], and the automated and simultaneous negotiation of agreements with different providers [6] [56]. An incipient, rapid growth is seen in the use of MAS when arranging cloud services (*Cloud manufacturing*) [30].

Finally, when a MAS takes the role of *Cloud Provider*, its main contribution is related to the security and privacy of data, due to the ability of the agents to monitor, provide reasoning skills and respond proactively to the changes in the environment [55] [53]. However, despite the different related studies that offer CC type services [63] [13], it is only possible to find promising and incipient studies in the state of the art related to the quality of services and provision or computational resources [59] [37].

The latter group, which includes the most interesting studies for the scope of this study, includes those in which MAS is applied to existing problems or weaknesses in the CC system. These studies are based on the experience from related or preceding studies and have a direct application to CC environments. However, it is also possible to observe that these studies solve partial or specific problems, but do not address the issue of controlling a CC environment or dealing with existing challenges comprehensively (management of infrastructure, SLA agreements, orchestration of services, etc.). As the current state of the art does not include references in which social models are applied in a CC context, the present study will pioneer the application of MAS in the development of a CC platform.

4. A Cloud computing architecture based on Multiagent System

Taking into account the needs and shortcomings detected in the review of the state of the art, this study proposes a new model of a multiagent architecture based on VO and especially designed for the management of CC environments. Prior to formalizing the proposed architecture, it is necessary to formalize the context and the environment in which the proposed architecture will be executed. Given the complexity associated with a CC environment, as well as the different artificial and human components involved in this context, it is necessary to define how the services will be offered at a technical level. For this reason, and following the CC model as introduced previously, each software service for the platform, at the PaaS or SaaS level, can be deployed simultaneously on various virtual machines (nodes or *workers*). This ability makes it possible to elastically configure the resources assigned to each service. In terms of requests for a specific service, the demand is balanced among the different virtual machines that are associated to the service. Additionally, the weight of each virtual node on the scale can vary dynamically in execution time. Therefore, the elasticity is based on modifying the (virtual) resources that have been assigned to each service dynamically according to demand.

The design of a monitoring and control system in a technology environment, requires the use of AI techniques to be able to incorporate the tasks that allow the dynamic adaptation to the changes and alterations in the demand of the services offered. The dynamic adaptation to changes that occurs in the environment requires learning capabilities, distributed representation of knowledge, and advanced reasoning models. In this sense, the MAS based on virtual organizations allow the incorporation of theories, models, mechanisms, methods and tools that facilitate the development of systems with reorganization capabilities and those that can adapt automatically to future changes in their environment [20]. Furthermore, this design model permits the external agents to perform services within the organization, which facilitates the incorporation of new functionalities that are not directly developed by the system.

The architecture proposed within the scope of this study is called +Cloud (*Multiagent System Cloud*) and is based on virtual organizations (VO) of intelligent agents, which in turn allows for the provision of new solutions required by CC platforms for the components to adapt, change, enter and exit. The main objective of +Cloud is the monitoring and ability to control a CC environment, allowing it to automatically and dynamically adapt to the needs at any given time. +Cloud gathers data from the entire CC environment, including the underlying infrastructure as well as the demand for the services it provides. This distributed monitoring model makes it possible to instantly adapt existing resources to the CC environment according to demand for each service, which in turn meets the double objective of complying with the established SLA agreements and reducing energy consumption. One of the most innovative aspects of +Cloud is the design of agents with advanced reasoning capabilities for the distribution of resources, as will be explained in a future section.

The proposed architecture is based on organizational aspects and, as such, it is necessary to identify the organizational structure to be used. To do so, the first step involves identifying the components of the architecture, which permits establishing the interaction model based on an analysis of the needs of the potential system users. Based on this analysis, it was possible to deduce the roles of the users and components that participate in the system and the way they will exchange information.

The development of a monitoring and management system for a CC environment that follows a MAS-based design model differs from traditional models that control this type of platform, which tends to have a centralized decision-making process [10]. The scope of this study follows an alternative model based on the theory of agents and MAS in which the responsibilities, primarily monitoring and decision-making, are distributed among the platform components. This model allows the decision-making process to be carried out right where the information is gathered, on the base that provides local knowledge, which has made it possible to design agile control processes based on uncertain information, prior knowledge, and the interaction among similar agents. To a certain extent, this unique feature may lead to a situation in which, while the system adapts to demand by following

the principle of elasticity of CC systems, some of the agents enter and exit the system according to the life cycle of the physical components where they are located. **Fig. 4.** shows how each one of the agents/roles that participate in the organization is located throughout the entire computational environment.

In following the indicated distribution model, each physical server in the CC environment contains an agent in charge of monitoring (*Local Monitor*) and another responsible for the local level (*Local Manager*). Between the two they have the authority to completely control the physical server (PR) where they are located, which in turn implies a distribution of resources in the virtual machine. However, when the resources must be distributed, which involves the assignment or removal of nodes for a particular service, another specialized agent (*Global Manager*), which is also located in each one of the physical nodes of the infrastructure, is in charge of making these types of decision, which involves more than one physical node on the CC platform.

Following a similar model, each service offered to the users is associated with two agents, one for monitoring (*Service Monitor*) and the other for control (*Service Supervisor*), both of which are in charge of ensuring compliance with the previously established SLA agreement. They are physically located in the node that balances the work-load among the different worker nodes, which permits them to have precise information available to make the correct decisions at their level. In this sense, the tasks for this level are related to the workload balance among the different nodes, error detection and, most importantly, monitoring the quality parameters for the service.

There are also other agents with very different tasks located in the entry point of the CC system. First, two control agents, the first of which is in charge of controlling the hardware infrastructure (*Hardware Supervisor*), its state, and the starting or stopping of the physical servers according to demand. A supervisor agent is the global controller (*Global Supervisor*) that ensures that the remaining components and agents function correctly and in accordance with their specification. Finally, there is also an agent in charge of establishing service agreements with the platform users (*SLA Broker*), which can negotiate the QoS level of services according to user needs and the state of the system at any given moment. It should be noted that this aspect of the CC paradigm extends beyond the scope of this research project and is considered part of future work to be carried out. Nevertheless, the state of the art includes a great variety of techniques and algorithms, some of them based on MAS [3] [2] [50] [48] [49] [56].

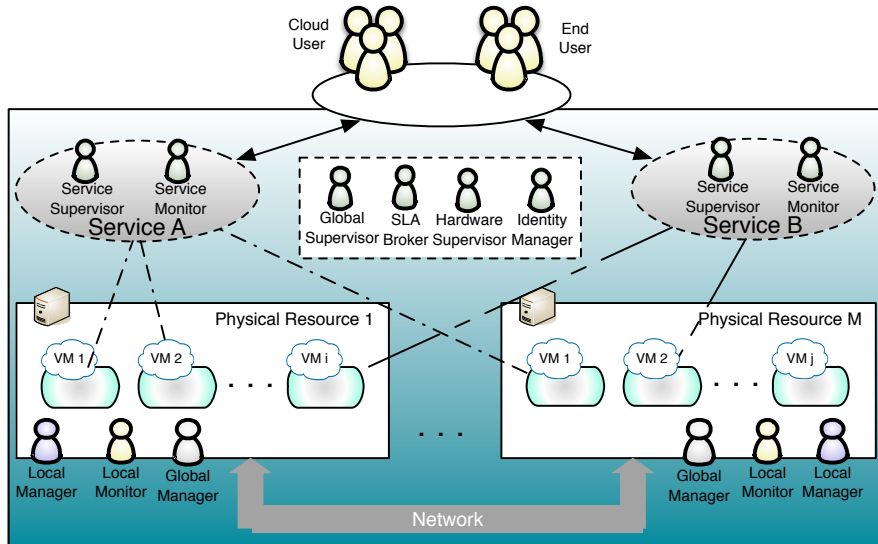


Fig. 4.- Agents distributed over the infrastructure

The system also includes an intelligent agent linked to the human users with the aim of simplifying the user's interaction with the system. The agents that are linked to external (human) entities are the *Cloud User* and the *End User*. The *Cloud User* agent is linked to the *Cloud Consumer* role according to the architecture proposed by NIST [31]; in other words, it consumes the services and products provided by the CC system, which in this case are persistence and deployment for web applications. The *End User* agent is the end user of the applications deployed by third parties in the CC system. Additionally, we have considered the existence of another agent, called *Identity Manager*, which is linked to the entity in charge of managing the entry and exit of users and their affiliation with agents within the system.

Given this identification of agents and the roles that participate in the system, it is possible to design an organization that is unified and intuitive, and contains a high level of abstraction [1]. In order to model an architecture such as that proposed in this study, it is necessary to have advanced design methodologies. The GORMAS (*Guidelines for Organization-based MultiAgent Systems*) [4] methodology is used in the present study. It is based on six meta models (agent, activity, interaction, environment, organization and norms), which make it possible to describe any MAS organization from four points of view: structural, functional, social and dynamic. The following sections describe the proposed architecture.

In line with the guidelines indicated in the GORMAS methodological guide, one of the first tasks is to instantiate the functional view (*mission*) of the organizational model, which is shown in Figure 5. This view presents the products and ser-

services offered by the system, the global objectives to pursue (mission and justification) and the affected interest groups.

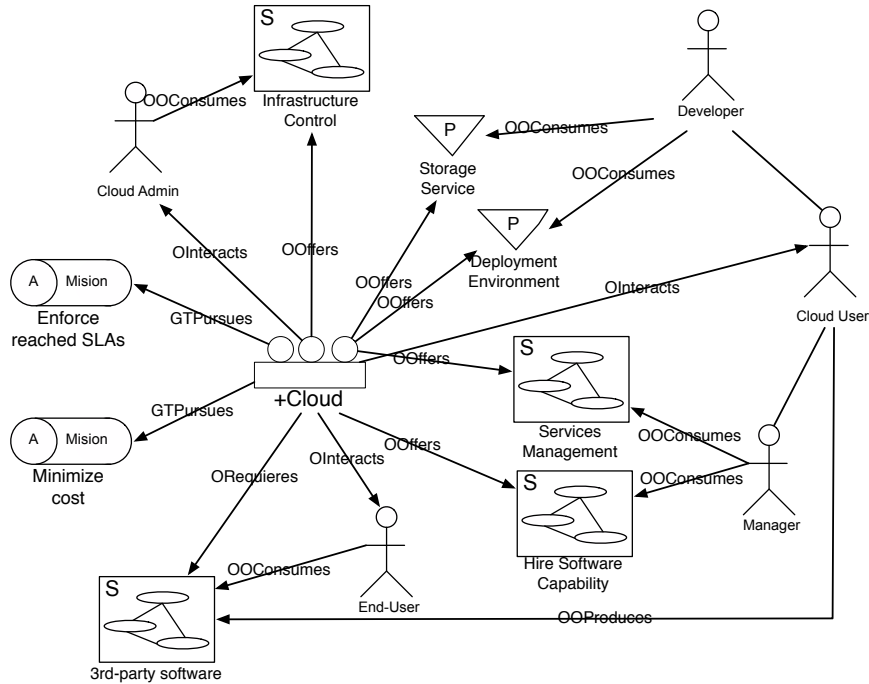


Fig. 5.- Functional view (mission) of the +Cloud organizational unit

Thus, the first mission and reason for the existence of an organization will be to comply with the service agreements agreed upon with the *Cloud User* role, while minimizing the costs associated with this mission. The diagram indicates the types of users that use the system (*Cloud Admin*, *Cloud User* and *End User*) and the products that are offered (storage and deployment of software). In order to facilitate the interaction of the platform, the following intrinsic services are also offered: software management, software hiring, and infrastructure control. Among the services offered, it should be noted that the platform also offers as a service those applications that can be deployed in the system by third parties (*Cloud User*); in other words, these types of applications are required by the platform to justify the need to offer storage and infrastructure products. However, given that a CC platform is a simple means (and not an end), these applications are also services that are offered to the *End User*.

5. Evaluation, discussions and Conclusions

The evaluation of the proposed dynamic distribution model is a thorough task and requires a hardware and software environment specially adapted to its needs. The evaluation and validation of the model for this study will be done through a CC platform developed within the scope of the research done by the BISITE research group,² and will include different computational services at the hardware and software level. From the beginning, this platform was conceived to integrate the proposed MAS +Cloud. As a result, the ability to reorganize and to adapt agent behavior are necessary functionalities for the platform to operate correctly.

In order to evaluate the proposed multiagent architecture, a series of experiments were conducted with the aim of simulating the behavior of an organization and its members in a real adaptation case. The results obtained from these experiments have made it possible to empirically evaluate whether the dynamic system responds according to its specification, dynamically adapting according to the state of the environment and the demand for services. Upon verifying the proper functioning of the organization in the simulation, the next step was to evaluate the behavior of the reasoning models that enable the dynamic adaptation of the organizational MAS. This was done through the distribution of the infrastructure resources in the CC platform among the different services offered in response to user demand.

This case study used the +Cloud platform, which was deployed in the HPC environment of the BISITE research group and composed of 15 latest generation machines that support virtualization in the hardware with the use of Intel-VT technology and the KVM virtualization system.

The case study is based on a simulated *Denial of Service* (DoS) attack [Needham, 1993] using methods that expose the platform for persistence of files. The *GetSize* method is a complex function that uses recursion to calculate the sum of the size of the files contained in a directory. In contrast, the *GetFolderContent* method is a much more simple function that only returns the identifiers for the files or directories contained in the path provided as a parameter.

The experiments performed use the same initial state as represented in Figure 6, where the file storage service is deployed in different nodes (VM1 and VM2), each one hosted by a different physical machine (VM1 and VM2 respectively). This starting point, in addition to being didactic and easy to understand, reflects the typical deployment of any service in a CC environment. Likewise, Figure 6 also shows the main agents that intervene in the readaptation process in the case study.

With regard to the distribution of the initial resources, the file storage service is deployed in different nodes (VM1 and VM2), each one hosted by a different physical machine (VM1 and VM2 respectively). As a result of this deployment, the

² <http://bisite.usal.es>

service has a high availability (it is deployed in two servers), but it is at the same time located in physical machines with different computational loads, which is what occurs in a real environment, since both physical machines host other virtual machines that correspond to other services from the CC platform. In other words, the physical server PR1 has many available and unassigned resources, while PR2 has no available resources and the machines it hosts have a high computational load.

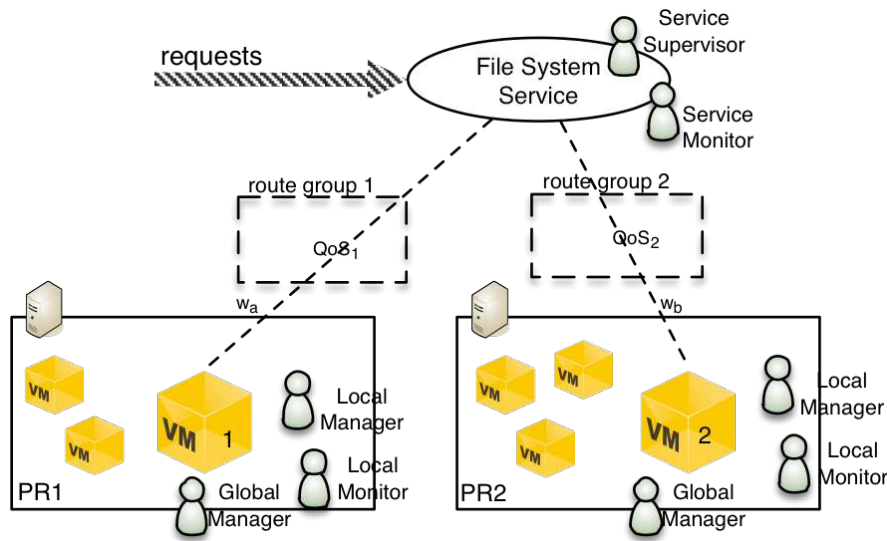


Fig. 6.- Initial state of the evaluation case study

The result of the experiments is presented shows that once the system detects that the QoS level in the service has decreased, that is, when the average response time is greater than the threshold (2.5 seconds, in this case), it automatically initiates an adaptation process for the infrastructure at virtual and real level. Once the auto-adapt process is complete and the value of the weights have been adjusted, we can see that the response time for the service returns to a value less than the acceptable QoS levels (less than 2.5 seconds).

The results of this study initially set forth to be one of the first MAS approaches, or more specifically a VO-based MAS approach, to fall within the framework of control and monitoring systems in a CC environment. The proposed architecture model is appropriate for the problem we need to solve. This new model has demonstrated that a control and monitoring system in a CC environment can be designed with artificial societies. This approach ensures independence of the decision-making process in software layers where the various actions are executed. This characteristic is particularly important in a CC environment because, as shown in the first phases of this research, current platforms exhibit a high dependency on the technological environment (virtualization tools, load balancers, dis-

tributed file systems, etc.). In the case of the +Cloud architecture, which uses ports to communicate with the environment, this dependency is limited to implementing the port itself; in other words, the communication interface with the environment. There is no doubt that a change in the capabilities offered by the underlying technology will also require changes to be made in the proposed reasoning models, as with any approach with a traditional design. However, in these cases the organizational models also offer an appropriate response to this challenge. Given the definitions of roles at a high level, if the technology proposes new capabilities, the adaptation in the proposed architecture will consist of modifying the individual or individuals that perform specific tasks or have a role in the organization.

An empirical comparison of the proposed model with other existing approaches in the state of the art is not possible, since it is difficult to recreate the computational environments and/or simulation in which they were evaluated. However, it is possible to perform a theoretical comparison of the approach proposed, in this regard the proposed model follows a distributed approach to solve the problem, which is completely different from other studies in the state of the art [41][8]. This approach, which has been demonstrated to be valid for the distribution of computational resources in this type of CC environment, presents certain advantages with respect to availability since there is not just one component in charge of the distribution of resources; instead, it is the system (society) itself that reorganizes through the individual adaptation of its components (agents). Furthermore, the proposed model simplifies the search for an appropriate solution to the problem because (i) it distributes the computational needs among different nodes; (ii) there are fewer values to consider since each node need only consider the data for its own resources; and finally (iii) each node can autonomously apply a partial solution to the problem, thus eliminating the need to coordinate at the global level of the platform.

Acknowledgments This work is supported by the Spanish government (MICINN) and European FEDER funds, project iHAS: Intelligent Social Computing for Human-Agent Societies (TIN2012-36586-C03-03)

References

1. Agüero, J., Rebollo, M., Carrascosa, C., & Julián, V. (2009, January). Agent design using model driven development. In 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009) (pp. 60-69). Springer Berlin Heidelberg.
2. Alhamad, M., Dillon, T., & Chang, E. (2010, September). Sla-based trust model for cloud computing. In Network-Based Information Systems (NBIS), 2010 13th International Conference on (pp. 321-324). IEEE.
3. An, B., Lesser, V., Irwin, D., & Zink, M. (2010, May). Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-

- Volume 1 (pp. 981-988). International Foundation for Autonomous Agents and Multiagent Systems.
4. Argente Villaplana, E. (2008). Gormas: Guías para el desarrollo de sistemas multiagente abiertos basados en organizaciones.
 5. Armbrust, Michael, Fox, Armando, Griffith, Rean, Joseph, Anthony D., Katz, Randy, Konwinski, Andy, Lee, Gunho, Patterson, David, Ariel, Rabkin, Stoica, Ion, Matei Zaharia. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50-58.
 6. Aversa, R., Di Martino, B., Rak, M., & Venticinque, S. (2010, February). Cloud agency: A mobile agent based cloud system. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on* (pp. 132-137). IEEE.
 7. Azodolmolky, S., Wieder, P., & Yahyapour, R. (2013). Cloud computing networking: challenges and opportunities for innovations. *Communications Magazine, IEEE*, 51(7).
 8. Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5), 755-768.
 9. Bezemer, C. P., Zaidman, A., Platzbeecker, B., Hurkmans, T., & Hart, A. (2010, September). Enabling multi-tenancy: An industrial experience report. In *Software Maintenance (ICSM), 2010 IEEE International Conference on* (pp. 1-8). IEEE.
 10. Buyya, R., Beloglazov, A., & Abawajy, J. (2010). Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. arXiv preprint arXiv:1006.0308.
 11. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6), 599-616.
 12. Capera, D., Georgé, J.-P., Gleizes, M.-P., & Glize, P. (2003). Emergence of organisations, emergence of functions. *Symposium on Adaptive Agents and Multi-Agent Systems*, (págs. 103-108).
 13. Chao, P., & Sun, H. M. (2013). Multi-agent-Based Cloud Utilization for the IT Office-Aid Asset Distribution Chain: An Empirical Case Study. *Information Sciences*.
 14. Chellappa, R. (1997, October). Intermediaries in cloud-computing: A new computing paradigm. In *INFORMS Annual Meeting*, Dallas.
 15. Chen, D., Wang, L., Wu, X., Chen, J., Khan, S. U., Kołodziej, J., Tiana M., Huang F. & Liu, W. (2013). Hybrid modelling and simulation of huge crowd over a hierarchical Grid architecture. *Future Generation Computer Systems*, 29(5), 1309-1317.
 16. Chiu, D. (2010). Elasticity in the cloud. *ACM Crossroads*, 16(3), 3-4.
 17. Decraene, J., Cheng, Y. Y., Low, M. Y. H., Zhou, S., Cai, W., & Choo, C. S. (2010, August). Evolving agent-based simulations in the clouds. In *Advanced Computational Intelligence (IWACI), 2010 Third International Workshop on* (pp. 244-249). IEEE
 18. Desisto, R.P. (2013, November). The Top Three Impacts of Cloud Computing on Sales and Business Applications. In: <https://www.gartner.com/doc/2622822>
 19. Dignum, V. (2004) A model for organizational interaction: based on agents, founded in logic, PhD. Thesis,
 20. Emeakaroha, V. C., Brandic, I., Maurer, M., & Dustdar, S. (2010, June). Low level Metrics to High level SLAs-LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments. *InHPCS* (pp. 48-54).
 21. Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08* (pp. 1-10). Ieee.
 22. Goiri, Í., Julià, F., Fitó, J. O., Macías, M., & Guitart, J. (2010). Resource-level QoS metric for CPU-based guarantees in cloud providers. In *Economics of Grids, Clouds, Systems, and Services* (pp. 34-47). Springer Berlin Heidelberg.

23. Gutierrez-Garcia, J. O., & Sim, K. M. (2010, November). Self-organizing agents for service composition in cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on* (pp. 59-66). IEEE.
24. Hutchins, D. (1999). *Just in time*. Gower Publishing, Ltd..
25. Kang, J., & Sim, K. M. (2011, January). Cloudle: An ontology-enhanced cloud service search engine. In *Web Information Systems Engineering–WISE 2010 Workshops* (pp. 416-427). Springer Berlin Heidelberg.
26. Kang, J., & Sim, K. M. (2011, June). Ontology and search engine for cloud computing system. In *System Science and Engineering (ICSSE), 2011 International Conference on* (pp. 276-281). IEEE.
27. Krauthem, F. J. (2009, June). Private virtual infrastructure for cloud computing. In *Proceedings of the 2009 conference on Hot topics in cloud computing* (pp. 5-5). USENIX Association.
28. Kusic, D., Kephart, J. O., Hanson, J. E., Kandasamy, N., & Jiang, G. (2009). Power and performance management of virtualized computing environments via lookahead control. *Cluster computing*, 12(1), 1-15.
29. Leitao, P., Inden, U., & Rückemann, C. P. (2013, November). Parallelising Multi-agent Systems for High Performance Computing. In *INFOCOMP 2013, The Third International Conference on Advanced Communications and Computation* (pp. 1-6).
30. Li, B. H., Zhang, L., Ren, L., Chai, X. D., Tao, F., Luo, Y. L., Wang Y., Yin C., Huang G. & Zhao, X. (2011). Further discussion on cloud manufacturing. *Computer Integrated Manufacturing Systems*, 17(3), 449-457.
31. Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST cloud computing reference architecture. NIST special publication, 500, 292
32. Lohr, S.. Google and ibm join in cloud computing research. In: *New York Times*, October 2007. RI: http://www.nytimes.com/2007/10/08/technology/08cloud.html?_r=0
33. Low, C., Chen, Y., & Wu, M. (2011). Understanding the determinants of cloud computing adoption. *Industrial management & data systems*, 111(7), 1006-1023.
34. McKendrick, J. (2012, March). NIST definition of cloud computing doesn't go far enough. In: ZDNET. <http://www.zdnet.com/blog/service-oriented/nist-definition-of-cloud-computing-doesnt-go-far-enough/8634>
35. Mell, P., & Grance, T. (2011). The NIST definition of cloud computing (draft). NIST special publication, 800(145), 7.
36. Mitchell Smith, D., Plummer, D.C., Bittman, T.J., Bova, T., Basso, M., Lheureux, B.J., & Prentice, B (2012, December). Predicts 2013: Cloud Computing Becomes an Integral Part of IT. In: <https://www.gartner.com/doc/2263916/predicts--cloud-computing-integral>
37. Núñez, A., Andrés, C., & Merayo, M. G. (2012). MAScloud: a framework based on multi-agent systems for optimizing cost in cloud computing. In *Computational Collective Intelligence. Technologies and Applications* (pp. 436-445). Springer Berlin Heidelberg.
38. Ortiz, S. (2011). The problem with cloud-computing standardization. *Computer*, 44(7), 13-16.
39. Patel, P., Ranabahu, A. H., & Sheth, A. P. (2009). Service level agreement in cloud computing.
40. Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z., & Zhu, X. (2008, March). No power struggles: Coordinated multi-level power management for the data center. In *ACM SIGARCH Computer Architecture News* (Vol. 36, No. 1, pp. 48-59). ACM.
41. Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z., & Zhu, X. (2008, March). No power struggles: Coordinated multi-level power management for the data center. In *ACM SIGARCH Computer Architecture News* (Vol. 36, No. 1, pp. 48-59). ACM.
42. Razavi, R., Perrot, J., & Guelfi, N. (2005). Adaptive modeling: an approach and a method for implementing adaptive agents. *Lecture Notes in Artificial Intelligence* , 3446, 136-148.
43. Reitbauer, A., Battino, A., Karageorgos, A., Mehandjiev, P., Valckenaers, P., & Saint-Germain, B. (2004). The MaBE middleware: extending multi-agent systems to enable

- open business collaboration. International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services - BASYS04
44. Rodríguez González, S. (2010). Modelo adaptativo para Organizaciones virtuales de agentes.
 45. Ross, J. W., & Westerman, G. (2004). Preparing for utility computing: The role of IT architecture and relationship management. *IBM systems journal*, 43(1), 5-19.
 46. Russom, P. (2011). Big data analytics. TDWI Best Practices Report, Fourth Quarter.
 47. Schertler, W. (1998). Virtual Enterprises in Tourism: Folklore and Facts—Conceptual Challenges for Academic Research—. In *Information and Communication Technologies in Tourism 1998* (pp. 278-288). Springer Vienna.
 48. Sim, K. M. (2012). Agent-based Cloud Computing. *IEEE TRANSACTIONS ON SERVICES COMPUTING*.
 49. Sim, K. M. (2012). Complex and concurrent negotiations for multiple interrelated e-markets.
 50. Son, S., & Sim, K. M. (2012). A Price-and-Time-Slot-Negotiation Mechanism for Cloud Service Reservations. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(3), 713-728.
 51. Talia, D. (2011). Cloud Computing and Software Agents: Towards Cloud Intelligent Services. In *WOA* (pp. 2-6).
 52. Talia, D. (2012). Clouds meet agents: Toward intelligent cloud services. *Internet Computing, IEEE*, 16(2), 78-81.
 53. Talib, A. M., Atan, R., Abdullah, R., & Murad, A. (2012, March). Security framework of cloud data storage based on Multi Agent system architecture-A pilot study. In *Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on* (pp. 54-59). IEEE.
 54. Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1), 50-55.
 55. Venkataramana, K., & Padmavathamma, M. (2012). Agent Based approach for Authentication in Cloud. *IRACST-International Journal of Computer Science and Information Technology & Security*, 2(3), 598-603.
 56. Venticinque, S., Aversa, R., Di Martino, B., Rak, M., & Petcu, D. (2011, January). A cloud agency for SLA negotiation and management. In *Euro-Par 2010 Parallel Processing Workshops* (pp. 587-594). Springer Berlin Heidelberg.
 57. von Laszewski, G., Diaz, J., Wang, F., & Fox, G. C. (2012, June). Comparison of multiple cloud frameworks. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on* (pp. 734-741). IEEE
 58. Wang, L., Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J., & Fu, C. (2010). Cloud Computing: a Perspective Study. *New Generation Computing*, 28(2), 137-146.
 59. Wei, Y., & Blake, M. B. (2013, March). Adaptive Service Workflow Configuration and Agent-based Virtual Resource Management in the Cloud. In *Proceedings of IEEE International Conference on Cloud Engineering*, San Francisco, CA.
 60. Wen, X., Gu, G., Li, Q., Gao, Y., & Zhang, X. (2012, May). Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on* (pp. 2457-2461). IEEE.
 61. Weyns, D., Schelfhout, K., Holvoet, T., & Glorieux, O. (2004). Role based model for adaptive agents. Forth Symposium on Adaptive Agents and Multiagent Systems at the AISB04 Convention
 62. Wooldridge M. y Jennings N. R. (1995). *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review, vol. 10(2) pp. 115-152, 1995.
 63. Yang, S. Y. (2012). A novel cloud information agent system with Web service techniques: Example of an energy-saving multi-agent system. *Expert Systems with Applications*.

64. You, X., Xu, X., Wan, J., & Yu, D. (2009, August). Ras-m: Resource allocation strategy based on market mechanism in cloud computing. In ChinaGrid Annual Conference, 2009. ChinaGrid'09. Fourth (pp. 256-263). IEEE.
65. Zambonelli, F., Gleizes, M.-P., Mamei, M., & Tolksdorf, R. (2004). Spray computers: frontiers of self-organisation for pervasive computing. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises - WETICE04 (págs. 397-402). IEEE Computer Society
66. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1), 7-18.