CrossMark

ORIGINAL RESEARCH

MAS-based self-adaptive architecture for controlling and monitoring Cloud platforms

Fernando De la Prieta[1] · Javier Bajo[2] · Sara Rodríguez[1] · Juan M. Corchado[1]

Abstract The theory of agents and multiagent systems can provide a new model for managing Cloud Computing 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. This article presents an architecture to model the control and monitoring system of a Cloud Computing platform by using a Virtual Organization of intelligent agents that self-adapt and reorganize according to the needs of the surrounding environment.

Keywords Multiagent system · Cloud computing · Architecture · Distributed systems

1 Introduction

The industry and the academia have taken great strides in recent years toward developing the Cloud Computing (CC)

✉ Fernando De la Prieta
fer@usal.es

Javier Bajo
jbajo@fi.upm.es

Sara Rodríguez
srg@usal.es

Juan M. Corchado
corchado@usal.es

[1] Department of Computer Science and Automation Control, University of Salamanca, Plaza de la Merced s/n, 37008 Salamanca, Spain

[2] Department of Artificial Intelligence, Technical University of Madrid, Bloque 2, Despacho 2101, Campus Montegancedo, Boadilla del Monte, 28660 Madrid, Spain

technological paradigm. The marketing model used in the CC paradigm is innovative, as it is based on a pay-as-you-go concept (Armbrust et al. 2010), in which users must negotiate and previously establish a Service Level Agreement (SLA) in order to access services (Alhamad et al. 2010). 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. This has resulted in a rapid growth of both private and public platforms (Fisher et al. 2010; Luo et al. 2011; Wen et al. 2012; Von Laszewski et al. 2012) aimed to provide innovative solutions that can resolve the current needs of the CC paradigm.

At technological level, the originality is determined by the spectrum of underlying technology (virtualization, service farms, web services, etc.), which have recently reached the point of allowing the services to be offered with the same level of quality, regardless of existing user demand (Liu et al. 2011; Wang et al. 2010; Zhang et al. 2010). These new possibilities at this level lead to the birth of a new concept, elasticity (Chiu 2010), which is based on the just-in-time production method (Hutchins 1999).

Existing research about elasticity, in the state of the art, is based on methods that use centralized algorithms based on mathematical and heuristic models (You et al. 2009; Chandwadkar and Kharat 2014; Mann 2015), neither of which can ensure the efficiency of the system, or even its availability, in the event of a system failure.

Given these shortcomings, it is necessary to study new techniques that allow for the evolution of existing models with regard to elasticity of services. This study proposes the use of agent and multiagent systems (MAS) (Wooldridge and Jennings 1995), since from an internal point of view, a CC is characterized by its massive distribution, heterogeneity, and uncertainty. The inclusion of proactive,

self-adaptation and learning capabilities, among others, is key for the evolution of these elastic management algorithms for computational resources. Using this MAS-based approach, the framework of this study proposes a dynamic and self-adapting architecture for the distribution of computational resources in a CC environment.

This work is organized as follows: the following section provides a description of the context of and related approaches, Sect. 3 proposes the architecture based on multiagent systems, while the evaluation and discussion of these systems are presented in Sect. 4. Finally, the last section presents the conclusions of the research.

2 State of the art

In a CC environment, the hardware infrastructure is virtualized (Che et al. 2010; Chen et al. 2008), which means that there is an abstraction layer between the real hardware infrastructure and the computing nodes. Each of the services is actually deployed in the computing nodes of this abstraction layer (referred to as virtual machines). In turn, the services are generally distributed among various computational nodes, which is why it is needed to include a load balance system that can distribute the requests among the various computational nodes attending the services.

The use of virtualization greatly simplifies the management of computational resources at the infrastructure level, making is possible to dynamically create or eliminate virtual machines on demand or even migrate a virtual machine from one physical server to another in execution time, without needing to stop or pause the machine. Therefore, and given the capabilities offered by virtualization technology, the problem, while complex, is actually simple in itself, since it is only based on the efficient redistribution of physical (real) resources among the different computational (virtual) nodes.

In current literature, the distribution of resources is viewed from two points of view (Buyya et al. 2010):

- *QoS-aware based, or market oriented* (Buyya et al. 2010) This first group is associated with a client-oriented distribution of resources model which attempts to minimize computational risks in order to distribute the computational resources according to the SLA reached, and following the pay-per-use economic model. According to this model, the management techniques for the computational resources aim to adhere to these agreements at all time, thus providing the quality of service that was requested and consequently expected by the end user. The state of the art includes studies in line with this approach by means of mathematical models (Nguyen Van et al. 2009a, b; Wei et al. 2010).

- *Energy-aware based* (Buyya et al. 2010) In this second approach, the distribution of resources takes place by taking into account both the pre-established SLA and the energy consumption, which assumes compliance with both. There are fewer studies in the state of the art with this approach as compared to the first, although they are more novel. This includes a variety of techniques are also based on mathematical models (Beloglazov et al. 2012; Kusic et al. 2009; Raghavendra et al. 2008).

In light of these studies in the current state of the art, it is necessary to propose a new architecture for distributing of computational resources which would take energy consumption into account. The present study follows a completely different approach based on optimization techniques and Artificial Intelligence (AI), which allows for the distribution of resources by following a distributed and scalable model, thus allowing the system to include different types of AI algorithms by means of the usage of organizational MAS.

A MAS framework based on Virtual Organizations (VO) has been selected to deal with these obstacles. MAS can be used to create a much more efficient, scalable and adaptable design for the CC environment than what is currently available. 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 low, but 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 platform* is becoming a common concept, mentioned by various authors in recent years (Talia 2011, 2012; Kang et al. 2010; Sim 2012; Venticinque et al. 2011; Braubach et al. 2014; Cao et al. 2009).

3 Architecture model

Taking into account the needs and shortcomings detected in the review of the state of the art, this article 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, 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, such as CC platform, 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, a MAS based on virtual organizations allows 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 (Rodriguez González 2010). 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 article is called +Cloud (*Multiagent System Cloud*) and is based on 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 of 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 (Heras et al. 2012; De la Prieta et al. 2013).
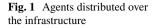
In order to model an architecture such as that proposed in this study, it is necessary to have advanced design methodologies. The GORMAS (*Guidelines for Organization-based MutiAgent Systems*) (Argente et al. 2011)

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.
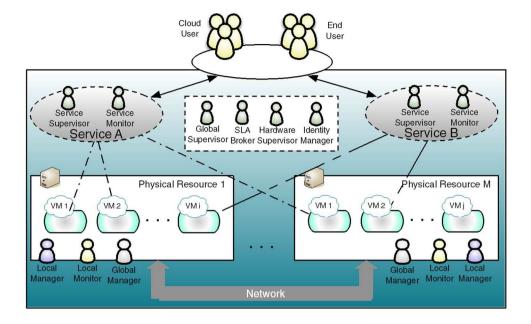
3.1 Formalizing the architecture

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 is 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 (Buyya et al. 2010). The scope of this article 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. Figure 1 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.

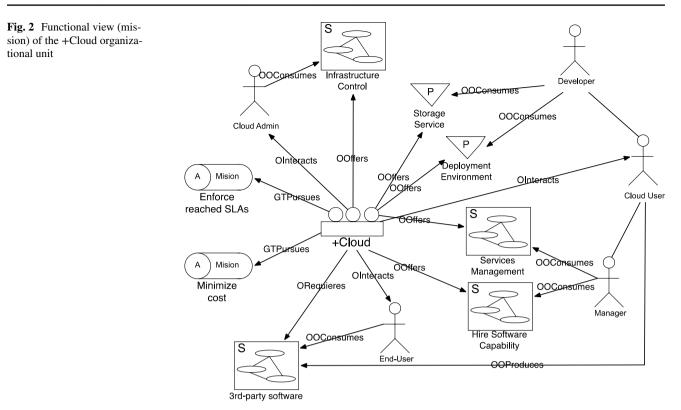**Fig. 1** Agents distributed over the infrastructure

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 of 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 PRs 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 (An et al. 2010; Alhamad et al. 2010; Sim 2010, 2012; Venticinque et al. 2011).

Finally, 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 (Liu et al. 2011); 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, intuitive, and contains a high level of abstraction (Agüero et al. 2009). 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 Fig. 2. This view presents the products and services offered by the system, the global objectives to pursue (mission and justification) and the affected interest groups.

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

Fig. 2 Functional view (mission) of the +Cloud organizational unit

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.

According to GORMAS methodology, it is necessary to specify four organizational dimensions: departmentalization, specialization, coordination and normalization. Based on this analysis of the organizational dimensions, the departmentalization and coordinating mechanisms within the system are analyzed. The coalition organizational pattern is selected as the most appropriate to model the system, since the various entities that compose a structure of this type will form groups according to functional similarities, and will coordinate among themselves to offer the most complex and elaborate functionality; this description is perfectly suited to the coordination model pursued in +Cloud. Using this pattern, the main organizational unit (+Cloud) is departmentalized in the following specialized organizational (sub) units.

- *SLA Negotiation,* in charge of grouping tasks to establish agreements and negotiations with the external *Cloud User* role.

- *Service Management* is assigned with the monitoring and supervision tasks for any products offered, and for overseeing the agreed-upon quality agreements for those products.
- *Infrastructure Management* controls the underlying hardware infrastructure, i.e., the system's computational resources (real or virtual) by monitoring the resources.

Finally, the last noteworthy product from the architecture's design will be described within the environment in which the proposed organizational MAS is located. The environment is dynamic, complex, uncertain and hostile. These characteristics define the type of environment in which MAS, especially those based on VO, are the most effective. Figure 3 provides a complete view of the model of the environment; the different system roles interact with the environment to achieve their individual objectives through the provided ports, whether to simply read or modify:

- Two applications that use interfaces to operate with roles external to the system. In other words, the application that can monitor the underlying infrastructure of the CC environment, especially designed for the *Cloud Admin* role and the web desktop, which has specific applications for the *Cloud User* role. The end users access applications directly from third parties that are deployed in the system.
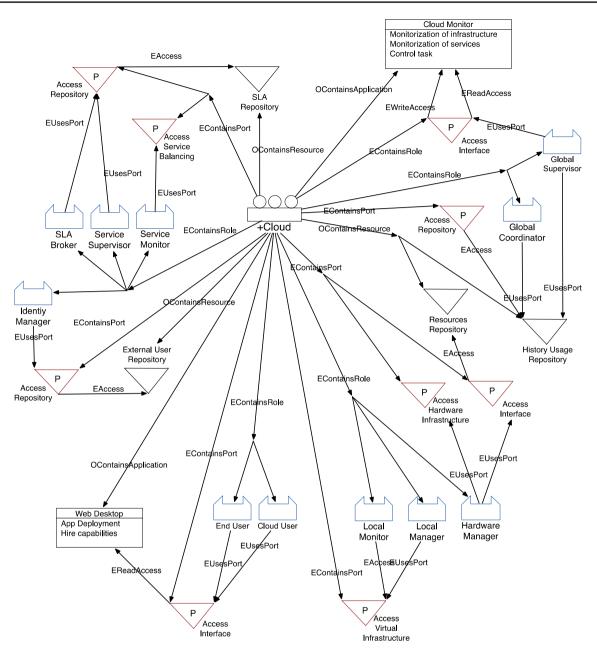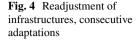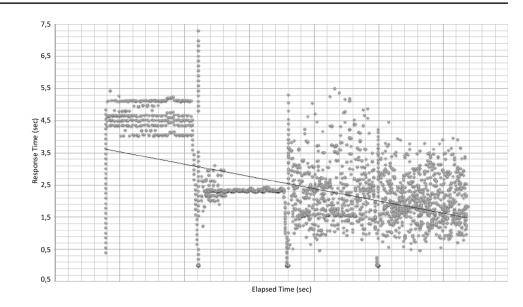
Fig. 3 Model of the environment. Access to the ports in the environment

- Three repositories associated with the information that should be persistent, the user´s storage, the agreements established with the users, the real hardware available, and the user's history.
- The figure also shows the different ports to access each of the resources for the environment. We can see the access ports to the resources that have been previously identified (repositories and interfaces), as well as the three ports that can reconfigure the underlying infrastructure (control of the hardware environment, virtual environment, and balancing systems).

In conclusion, the design of the architecture based on the GORMAS methodology can provide a set of artefacts that can specify the system's design. This process has made it possible to identify, from an external level, the mission, services and products provided by the system. Additionally, from an internal level, it has also been possible to identify three organizational units, their corresponding agents, means of interacting with the environment (communication interfaces with the users and access ports to the environment), as well as the interaction and negotiation model used. On the basis of this architecture, each agent

Fig. 4 Readjustment of infrastructures, consecutive adaptations

4 Evaluation and discussion

can include different reasoning approach and algorithms that participate in the society to carry out the distribution of computational resources. These algorithms fall in line with the philosophy of MAS and are based on the autonomy of the components, such as distributed decision making. With this model it is possible not only to optimize the resource distribution process, but also to improve system availability. This resource distribution model is innovative in and of itself; since current models are centralized and do not, therefore, consider the system's dynamic self-adaptation in response to changes produced in the environment.

4 Evaluation and discussion

The evaluation and validation of the model for this study will be done through a CC platform developed within the scope of the research carried out by BISITE research group[1] and will include different computational services at the hardware and software level. This CC platform 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.

In order to evaluate the proposed MAS 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.

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 experiments were looking for various consecutive executions in the adaptation process so that a single execution of the adaptation algorithm could not satisfy the demand in the services. These tests, from the point of view of the adaptation model, were positive, functioning perfectly within the limits of the case study, as presented in the graph in Fig. 4, after 2 or 3 readaptation processes, depending on the method evaluated. This dispersion was not considered to be due to the proposed resource distribution algorithms, but to the overload of the demand for resources in the lower levels that contain persistence data (database and distributed storage system).

The previous experiments provided an empirical evaluation in a case study of a multiagent architecture and the adaptation models proposed within the platform of this study. The experiments conducted have made it possible to validate various aspects as explained in detail below.

In the current state of the art, the execution of the assignment algorithms is a complex task that requires a great deal of computational time and power (Goudarzi and Pedram 2011). In contrast, 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. In terms of the specific adaptation algorithms, the proposed solution uses optimization

---

techniques, which have been previously used (Kusic et al. 2009), but never following a distributed approach.

The other big difference in this study with regard to other approaches in the current state of the art is the minimum unit of distribution. With this approach it is possible to distinguish the micro and the macro level in the distribution of infrastructure resources, which makes it possible to solve the problem of demand without needing to instantiate virtual machines, which is in itself an energy efficient solution and maximizes the use of computational resources.

5 Conclusions

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 study proposed a new architectural model based on a MAS VO with a clearly integrative character. 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. These characteristics 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, distributed file systems, etc.).

References

Agüero J, Rebollo M, Carrascosa C, Julián V (2009) 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

Alhamad M, Dillon T, Chang E (2010) Conceptual SLA framework for cloud computing. In Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on IEEE (pp. 606–610)

An B, Lesser V, Irwin D, Zink M (2010) 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

Argente E, Botti V, Julian V (2011) GORMAS: An organizational-oriented methodological guideline for open MAS. In Agent-Oriented Software Engineering X (pp. 32–47). Springer, Berlin Heidelberg

Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A et al (2010) A view of cloud computing. Commun ACM 53(4):50–58

Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener Comput Syst 28(5):755–768

Braubach L, Jander K, Pokahr A (2014). A middleware for managing non-functional requirements in cloud paas. In Cloud and Autonomic Computing (ICCAC), 2014 International Conference on IEEE (pp. 83–92)

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

Cao BQ, Li B, Xia QM (2009) A service-oriented QoS-assured and multi-agent cloud computing architecture. In Cloud Computing. Springer, Berlin Heidelberg (pp. 644–649)

Chandwadkar R, Kharat MU (2014) Review on Agent based Cloud Computing, IJCA Proceedings on Innovations and Trends in Computer and Communication Engineering—ITCCE Number 3

Che J, Yu Y, Shi C, Lin W (2010) A synthetical performance evaluation of openvz, xen and kvm. In Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific IEEE (pp. 587–594)

Chen W, Lu H, Shen L, Wang Z, Xiao N, Chen D (2008) A novel hardware assisted full virtualization technique. In Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for IEEE (pp. 1292–1297)

Chiu D (2010) Elasticity in the cloud. ACM Crossroads 16(3):3–4

De la Prieta F, Rodríguez S, Bajo J, Corchado JM (2013) A multiagent system for resource distribution into a Cloud Computing environment. In International Conference on Practical Applications of Agents and Multi-Agent Systems (pp. 37–48). Springer, Berlin Heidelberg

Fisher P, Pant R, Edberg J (2010) Cloud computing: assessing Azure, Amazon Ec2, google App Engine and Hadoop for it decision making and developer career growth. Apress

Goudarzi H, Pedram M (2011) Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems. In Cloud Computing (CLOUD), 2011 IEEE International Conference on IEEE (pp. 324–331)

Heras S, De la Prieta F, Julian V, Rodríguez S, Botti V, Bajo J, Corchado JM (2012) Agreement technologies and their use in cloud computing environments. Prog Artif Intell 1(4):277–290

Hutchins D (1999). Just in time. Gower Publishing, Ltd

Kang J, Sim KM (2010) Cloudle: an ontology-enhanced cloud service search engine. In Web Information Systems Engineering—WISE 2010 Workshops (pp. 416–427). Springer, Berlin Heidelberg

Kusic D, Kephart JO, Hanson JE, Kandasamy N, Jiang G (2009) Power and performance management of virtualized computing environments via lookahead control. Cluster Comput 12(1):1–15

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

Luo JZ, Jin JH, Song AB, Dong F (2011) Cloud computing: architecture and key technologies. J China Inst Commun 32(7):3–21

Mann ZÁ (2015) Allocation of virtual machines in cloud data centers—A survey of problem models and optimization algorithms. ACM Computing Surveys (CSUR), 48(1):11

Needham RM (1993) Denial of service. In Proceedings of the 1st ACM Conference on Computer and Communications Security (pp. 151–153). ACM

Nguyen Van H, Dang Tran F, Menaud JM (2009a) Autonomic virtual resource management for service hosting platforms. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing (pp. 1–8). IEEE Computer Society

Nguyen Van H, Dang Tran F, Menaud JM (2009b) SLA-aware virtual resource management for cloud infrastructures. In Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on (vol 1, pp. 357–362). IEEE.

Raghavendra R, Ranganathan P, Talwar V, Wang Z, Zhu X (2008) 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.

Rodríguez González S (2010) Modelo adaptativo para Organizaciones virtuales de agentes.

Sim KM (2010) Towards complex negotiation for cloud economy. In advances in Grid and Pervasive Computing (pp. 395–406). Springer, Berlin Heidelberg

Sim KM (2012) Agent-based cloud computing. Services Computing, IEEE Transactions on, 5(4):564–577

Son S, Sim KM (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

Talia D (2011) Cloud Computing and Software Agents: towards Cloud intelligent services. In WOA (vol 11, pp. 2–6)

Talia D (2012) Clouds meet agents: toward intelligent cloud services. IEEE Internet Computing, 2:78–81

Venticinque S, Aversa R, Di Martino B, Rak M, Petcu D (2011) A cloud agency for SLA negotiation and management. In Euro-Par 2010 Parallel Processing Workshops (pp. 587–594). Springer, Berlin Heidelberg

Von Laszewski G, Diaz J, Wang F, Fox GC (2012) Comparison of multiple cloud frameworks. In: 2012 IEEE 5th International conference on cloud computing (CLOUD). IEEE, pp 734–741

Wang L, Von Laszewski G, Younge A, He X, Kunze M, Tao J, Fu C (2010) Cloud computing: a perspective study. New Gener Comput 28(2):137–146

Wei G, Vasilakos AV, Zheng Y, Xiong N (2010) A game-theoretic method of fair resource allocation for cloud computing services. J Supercomput 54(2):252–269

Wen X, Gu G, Li Q, Gao Y, Zhang X (2012) 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

Wooldridge M, Jennings NR (1995) Intelligent agents: theory and practice. Knowl Eng Rev 10(2):115–152

You X, Xu X, Wan J, Yu D (2009) RAS-M: resource allocation strategy based on market mechanism in cloud computing. In 2009 Fourth China Grid Annual Conference (pp. 256–263). IEEE

Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. J Internet Serv Appl 1(1):7–18