A CBR Approach to Allocate Computational Resources Within a Cloud Platform

Fernando De la Prieta, Javier Bajo and Juan M. Corchado

Abstract Cloud Computing paradigm continues growing very quickly. The underlying computational infrastructure has to cope with this increase on the demand and the high number of end-users. To do so, platforms usually use mathematical models to allocate the computational resource among the offered services to the end-user. Although these mathematical models are valid and they are widely extended, they can be improved by means of use intelligent techniques. Thus, this study proposes an innovative approach based on an agent-based system that integrated a case-based reasoning system. This system is able to dynamically allocate resources over a Cloud Computing platform.

1 Introduction

The technology industry and the scientific community have taken great strides in recent years toward implementing the Cloud Computing (CC) technological paradigm. This has resulted in a rapid growth of both private and public platforms [12, 17, 25, 28] aimed to provide innovative solutions that can resolve the current needs of the CC paradigm.

The marketing model used in the CC paradigm is innovative, as it is based on a pay-as-you-go concept [2], in which users must negotiate and previously establish a Service Level Agreement (SLA) in order to access services [1]. Once this contract

F. De la Prieta (✉) · J.M. Corchado
Department of Computer Science and Automation Control,
University of Salamanca, Plaza de la Merced s/n, 37008 Salamanca, Spain
e-mail: fer@usal.es

J. Bajo
Department of Artificial Intelligence, Technical, University of Madrid,
Bloque 2, Despacho 2101, Campus Montegancedo,
Boadilla del Monte, Madrid 28660, Spain
e-mail: jbajo@fi.upm.es

P. Novais et al. (eds.), *Intelligent Distributed Computing IX*,
Studies in Computational Intelligence 616
DOI 10.1007/978-3-319-25017-5_7

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. In this regard, novelty is determined by the innovative 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 [16, 26, 31]. These new possibilities at a technological level lead to the birth of a new concept, elasticity [9], which is based on the just-in-time production method [13].

Existing research in the state of the art is based on methods that use centralized algorithms based on mathematical and heuristic models [15, 19, 30], 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 models derived from Artificial Intelligence (AI), since fron an internal point of view, a CC is characterized by its massive distribution, heterogeneity, and high level of uncertainty, which is precisely where the application AI holds great potential. The inclusion of proactive, self-adaptation and learning capabilities, among others, is key for the evolution of these elastic management algorithms for computational resources. As a result, agents and multiagent systems [29] (MAS) were selected among all the available AI techniques because of their distributed nature and ability to work in environments such as CC systems, whose characteristics would clearly identify them as open systems.

Using this MAS-based approach, the framework of this study proposes a dynamic and self-adapting model for the distribution of computational resources in a CC environment. This model is based on the learning capabilities provided by a case-based reasoning (CBR) [10] system, an approach which has not previously been used in this type of distributed environment. These reasoning systems develop a reasoning model similar to that of humans, using past experiences to solve a specific problem.

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

2 Resource Allocation in Cloud Computing Platforms

In a CC environment, the hardware infrastructure is virtualized [7, 8], 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 their needs to be a work 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 [5]:

- **QoS-aware based, or market oriented** [5]. 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 [18, 23, 27].
- **Energy-aware based** [5]. 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 [3, 15, 19].

In light of these studies in the current state of the art, it is necessary to propose a model for the distribution of computational resources which would take energy consumption into account. The aim is to reduce the energy consumption required to satisfy the SLA that have been established with the platform users. The present study follows a completely different approach based on optimization techniques and AI, which allows for the distribution of resources by following a distributed and scalable model, thus allowing the system to learn over an extended period of time.

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. 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 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 [4, 6, 14, 20–22, 24].

3 Proposed Architecture

Taking into account the needs and shortcomings detected in the review of the state of the art, this study proposes a new model of allocating resources based on a CBR approach and guided by a multiagent architecture especially designed for the management of CC environments. This section will describe the key components that allow extending the operation of the elastic algorithms for the distribution of resources proposed within this work.

To begin, we would like to note that since the proposed MAS is a distributed system by nature, each of the agents that work in the distribution of resources can be located throughout the entire CC environment. That is, the CC system is monitored and controlled in a distributed manner. 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 dual objective of complying with the established SLA agreements and reducing energy consumption.

Figure 1 presents an overview of the agent-based architecture The following agents are directly related to the monitoring and control of the hardware:

- **Local Monitor**. In charge of gathering data related to the state of the local resources for each physical server, including the physical machine as well as the different virtual machine it hosts.
- **Local Manager**. In charge of controlling the computational resources of the physical machine. In other words, responsible for initiating or turning off virtual machines according to the previously configured service templates.
- **Global Manager**. The primary agent in charge of decision making with regard to the distribution of computational resources. In order to perform this task, the agent uses a CBR-BDI model, which will be explained in detail in the following section. As a means of support for making decisions regarding the distribution of resources, this agent uses a partial knowledge base (provided by the Local Monitor) and the ability to modify local resources in each machine (provided by the Local Manager).

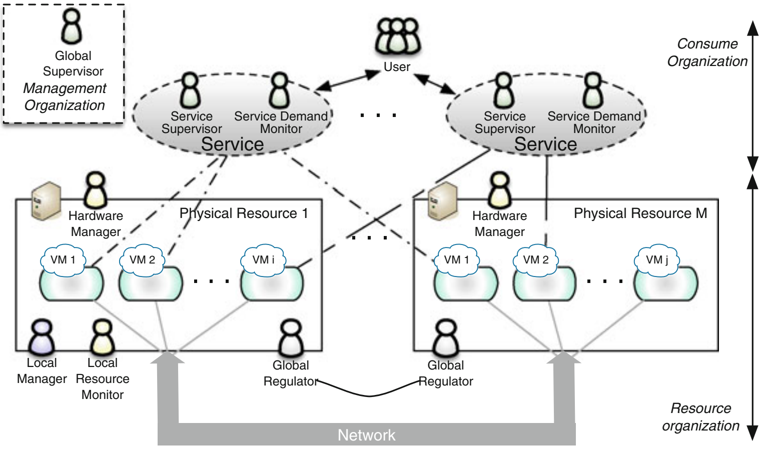

Fig. 1 Agent-based architecture for a Cloud Computing Platform

The redistribution of resources at a macro level is performed by the *Global Manager* agents, which have greater authority than the *Local Manager* agent and can inform them of the need to start up a new machine with a specific service and specific characteristics. At the end of this process, a new virtual machine (*VM*), with specific characteristics of Memory and virtual cpus will be instantiated in order to meet the current demand.

The *Global Manager* is a highly specialized agent that implements a CBR-BDI [10, 11] deliberative architecture. As a result, the reasoning process in each physical node is based on past experience gained from storing similar cases. The case memory is central to the entire CC system; the system's global knowledge can be shared by each of its members, in this case the *Global Manager* agent. Given that this memory can grow exponentially as a maintenance strategy, a high-speed schema-less database is used to provide fast access to the stored data, based on MongoDB.[1]

The Global Manager initiates the process by defining the concept of case $C = \{P, S(P), E\}$ where:

- P corresponds to the problem description, which has a matrix-matched representation associated to the instantiation of the use of resources.
- $S(P)$ is associated with the solution to the problem: $S(P) = \{M, vcpu\}$ in terms of memory and *vcp*u.
- The efficiency (E) is measured from two perspectives:

  - the degree of efficiency of the proposed solution within the physical server where the virtual machine has been instantiated. This degree of efficiency is

[1] http://www.mongodb.org.

proposed by the *Local Monitor* agent according to the usage rates of the processor and the allocated memory.
- The degree of efficiency from the point of view of the service. The degree of efficiency measures the number of additional nodes required by the service.

The CBR (*Case-Based Reasoning*) process is initiated and retrieves similar cases from the case memory. The most similar cases are selected according to the following steps: (i) Select the cases from the physical machines with similar characteristics; (ii) a vector is configured for each case that contains the same number of virtual machines that are in the case; and (iii) the cases selected from this subset are those that previously used the same service that is now requesting resources, and during a period of time similar to the current case.

A solution to the problem, which is based on the retrieved cases, will be prepared during the reuse phase:

- If the case base does not contain a previous similar case, the solution to the problem will be associated to the minimum resources determined at the level where the service is instantiated.
- If, on the other hand, similar cases are retrieved, the solution to the problem will be the closest case multiplied by the case efficiency:
- If the values assigned to the previous solution are greater than the values assumed by the machine, due to the fact that there are not many resources available, the result of the case will be the maximum amount of resources available in the machine.

Once the solution to the case has been calculated, the new node will be instantiated and its use evaluated from a micro and macro perspective, thus providing the value of efficiency for the solution. Finally, during the final state of the proposed CBR cycle, the case and its corresponding efficiency will be stored for future use.

4 Evaluation

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 the BISITE research group,[2] 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.

During the experiment, 10 threads that query to specific methods of the service (*GetSize* and *GetFolderContent*) are launched every three seconds, to a maximum

[2] http://bisite.usal.es.

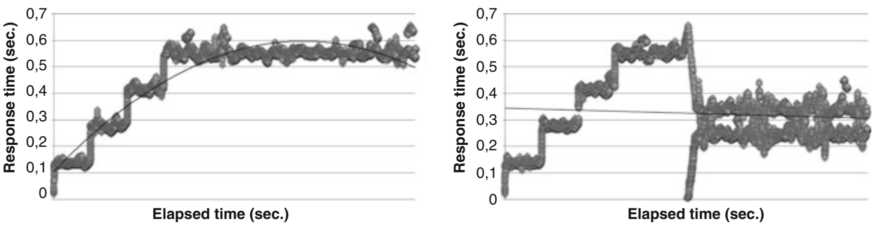

Fig. 2 Experiment 1: readjustment of the infrastructure resources for method (*left* no adaptation; *right* adaptation)

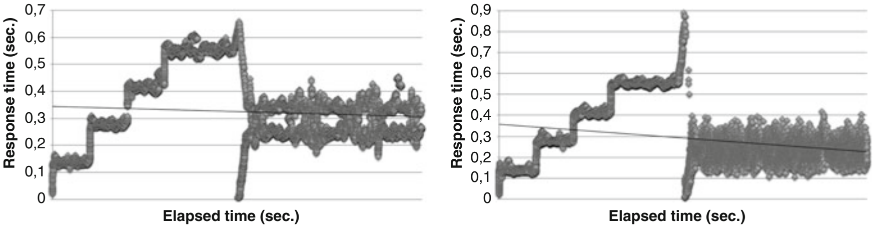

Fig. 3 Experiment 1: readjustment of the infrastructure resources for method (*left* experiment 1; *right* informed adaptation)

of 40 threads. The process starts once the agent-based architecture detects a decrease in performance, at which time it directly executes the adaptation process. The *Global Manager* agent for each of the physical machines that host the service nodes. We should recall that the *Global Manager* agent is a specialized agent that uses a CBR-BDI reasoning process [10] in charge of the distribution of resources at a macro level. Once they receive the initial alert, these agents resend the alert message to the remaining *Global Manager* agents in the CC system.

Each individual Global Manager hosted by each physical machine carry out in parallel the process described in the previous section. Thus *n* solutions are proposed. The agent-based architecture reactively selects the node that offers the most resources at the virtual machine level.

The results in terms of QoS can be seen in Fig. 2, which also show an increase in the quality level after the adaptation has been completed.

The case study was repeated numerous times, which made it possible to store a good number of past experiences in the case memory. However, as presented in Fig. 3 when there are many cases in the memory and a number of past experiences similar to the current problem, the adaptation results are actually better because the QoS level is lower.

5 Conclusions

This study initially set forth to be one of the first MAS approaches 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 with a clearly integrative character. A series of algorithms for the distribution of computational resources in a CC environment were developed, evaluated and validated. Its biggest innovation centers on the system's dynamic ability to automatically adapt according to demand and learn from previous experiences.

This new model has demonstrated that a control and monitoring system in a CC environment can be designed with MAS. The inherently distributed nature of MAS makes it possible to implement elastic algorithms for services by following a distributed strategy. The distribution of responsibilities within the scope of this type of algorithm makes it possible not only to make decisions where the problems actually arise, but to distribute the computing capability required to reach a solution among different instances of the CC environment.

This approach also ensures independence of the decision-making process in software layers where the various actions are executed. 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. 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 within the MAS.

Finally, This approach can maximize the degree of efficiency of the proposed solutions with regard to previous solutions, which in turn progressively improves the response and the system's capability since it is capable of learning. Moreover, this learning ability is important in an uncertain environment such as the CC system. If the context or environment of the CC platform changes at any given time, the adaptation model will evolve in turn, adapting the proposed solutions in order to maximize the efficiency of the given solution.

Acknowledgments This work has been supported by the MICINN project TIN2012-36586-C03-03.

References

1. Alhamad, M.; Dillon, T.S., Chang, E.: Conceptual SLA framework for cloud computing (2010)
2. Armbrust, M., et al.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
3. Beloglazov, A. Abawajy, J. Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener. Comput. Syst. **28**(5), 755–768 (2012)

4. Braubach, L., Jander, K. Pokahr, A.: A middleware for managing non-functional requirements in cloud PaaS. In: IEEE International Conference on Cloud and Autonomic Computing (ICCAC), pp. 83–92 (2014)
5. Buyya, R., Beloglazov, A., Abawajy, J.: Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. Preprint arXiv:1006.0308 (2010)
6. Cao, B.-Q., Li, B. Xia, Q.-M.: A service-oriented QoS-assured and multi-agent cloud computing architecture. In: Cloud Computing, pp. 644–649. Springer, Berlin (2009)
7. Che, J., et al.: A synthetical performance evaluation of OpenVZ, Xen and KVM. In: IEEE 2010 Asia-Pacific Services Computing Conference, pp. 587–594. IEEE (2010)
8. Chen, W., et al.: A novel hardware assisted full virtualization technique. In: The 9th International Conference for Young Computer Scientists, pp. 1292–1297. IEEE (2008)
9. Chiu, D.: Elasticity in the cloud. Crossroads **16**(3), 3–4 (2010)
10. Corchado, J.M., et al.: Replanning mechanism for deliberative agents in dynamic changing environments. Comput. Intell. **24**(2), 77–107 (2008)
11. Corchado, J.M., Laza, R.: Constructing deliberative agents with case-based reasoning technology. Int. J. Intell. Syst. **18**(12), 1227–1241 (2003)
12. Fisher, P., Pant, R., Edberg, J.: Cloud Computing: Assessing Azure, Amazon EC2, Google App Engine and Hadoop for it Decision Making and Developer Career Growth. Apress, New York (2010)
13. Hutchins, D.:Just in time. Gower Publishing Ltd., London (1999)
14. Kang, J. Sim, K.M.: Cloudle: an ontology-enhanced cloud service search engine. In: Web Information Systems Engineering–WISE 2010 Workshops. Springer, Berlin, Heidelberg, pp. 416–427 (2011)
15. Kusic, D., et al.: Power and performance management of virtualized computing environments via lookahead control. Cluster Comput. **12**(1), 1–15 (2009)
16. Liu, F., et al.: NIST Cloud Computing Reference Architecture, vol. 500, p. 292. NIST Special Publication (2011)
17. Luo, J.-Z., et al.: Cloud computing: architecture and key technologies. J. China Inst. Commun. **32**(7), 3–21 (2011)
18. van Nguyen H., Dang Tran, F. Menaud, J.-M.: 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 (2009)
19. Raghavendra, R., et al.: No power struggles: coordinated multi-level power management for the data center. In: ACM SIGARCH Computer Architecture News, pp. 48–59. ACM, (2008)
20. Sim, K.M.: Agent-based cloud computing. IEEE Trans. Serv. Comput. **5**(4), 564–577 (2012)
21. Talia, D.: Cloud computing and software agents: towards cloud intelligent services. In: WOA 2011, pp. 2–6
22. Talia, D.: Clouds meet agents: toward intelligent cloud services. IEEE Internet Comput. 16(2), 78–81 (2012)
23. Van Hien N., Tran, F.D., Menaud, J.-M.: SLA-aware virtual resource management for cloud infrastructures. In: Ninth IEEE International Conference on Computer and Information Technology, CIT'09, pp. 357–362. IEEE (2009)
24. Venticinque, S., et al.: A cloud agency for SLA negotiation and management. In:Euro-Par 2010 Parallel Processing Workshops, pp. 587–594. Springer, Berlin, Heidelberg (2011)
25. Von Laszewski, G., et al.: Comparison of multiple cloud frameworks. In: 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), pp. 734–741. IEEE (2012)
26. Wang, L., et al.: Cloud computing: a perspective study. New Gener. Comput. **28**(2), 137–146 (2010)
27. Wei, G., et al.: A game-theoretic method of fair resource allocation for cloud computing services. J. Supercomput. **54**(2), 252–269 (2010)
28. Wen, X., et al.: Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In: 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 2457–2461. IEEE (2012)

29. Wooldridge, M., Jennings., N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. 10, (02), 115–152 (1995)
30. You, X., et al.: RAS-M: resource allocation strategy based on market mechanism in cloud computing. In: Fourth ChinaGrid Annual Conference, ChinaGrid'09, pp. 256–263. IEEE (2009)
31. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. **1**(1), 7–18 (2010)