



System Multi Agents for Automatic Negotiation of SLA in Cloud Computing

Zineb Bakraouy^(✉), Amine Baina, and Mostafa Bellafkih

STRS Laboratory, National Institute of Posts and Telecommunications,
2, Avenue Allal El Fassi, Madinat Al Irfane, Rabat, Morocco
Zineb.bakraouy@gmail.com, {baina,bellafkih}@inpt.ac.ma

Abstract. Over the past few years, more and more researchers have been involved in research on both agent technology and cloud computing. Efforts have mainly been made to reduce the border between the two technologies. The combination of agents and cloud computing is based on the challenges faced by both communities and the need to develop more intelligent systems. The technology of agents whose goal is to deal with complex systems has revealed opportunities for improving the Cloud Computing Domain. This article present our contribution that is devoted to the integration of SMA in Cloud Computing. The role and functionality of each which constitute our system is detailed. We realized an SMA-based Cloud Computing system for the classification and the automatic negotiation of SLAs.

Keywords: Agent · SMA · SLA · Negotiation · QOS · Availability · Web services
Service broker

1 Introduction

Cloud computing has a promising technology that facilitates the execution of scientific and commercial applications. It provides flexible and scalable services, at the request of users, via a pay-as-you-go model. Typically, it can provide three types of services: SaaS (Software as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service) and three deployment models: public cloud, private cloud, hybrid cloud. At the same time, multi-agent systems (SMAs) represent a new concept in distributed applications. SMAs are based on multiple agents interacting with each other to solve problems using a decentralized approach where several agents contribute to the solution by cooperating with each other. The client/server model is the most used model for construction of applications. Therefore, the disadvantage of this model has is increasing the traffic on the network and needs a permanent connection. Moreover the large and strong demand of services in cloud provokes and causes many collisions in the network. For this, we propose a framework that uses system multi agents, in order to improve availability of services with automatic negotiation based SMA. The use of SMAs has a number of advantages:

- **Dynamic system:** SMAs inherit the benefits of AI in terms of symbolic (knowledge) processing. However, contrary to the traditional approaches of Artificial Intelligence that simulate, to a certain extent, the capacities of human behavior, SMAs allow to model a set of interacting agents [1].
- **Large number of agents:** a large number of agents are at the heart of the problem in this type of modeling, contrary to the theory of games where rarely more than three actors are represented.
- **Flexibility of the IT tool:** to modify the behavior of agents, to add or remove possible actions, to extend the information available to all agents, unlike models traditionally used in economics.
- **Distributed problem solving:** it is possible to decompose a problem into sub-parts to solve each independently to arrive at a stable solution [2].

In this paper we provide an overview of the SMA approach adopted in the QoS Negotiation platform to expose negotiation functionalities to Web services. SLA-based negotiation is a crucial support to handle the widely-ranging requirements that characterize Web services. The paper is organized as follows: Sect. 2 overviews the Cloud Computing concept, in Sect. 3 we present the System Multi Agents used in our architecture, in Sect. 4 we discuss a collaboration of SMA and Cloud computing from an analysis of existing Literature, in Sect. 5 is dedicated to present our solution with results and discussion, eventually in Sect. 6 we draw our conclusions.

2 Cloud Computing

2.1 Definition

There are many definitions of the term Cloud Computing (CC) and there is little consensus on a single and universal definition. This multitude of definitions reflects the diversity and technological richness of Cloud Computing. In what follows, we cite some of the most relevant. According to [7], based on a close-up view of the Grid computing grids [3, 4], Cloud Computing [6] is mainly based on the paradigm of distributed computing [5] on a large scale to ensure an on-demand service accessible through the Internet. A second definition, proposed in [8, 9] and which is more abstract, defines cloud computing by using the computing resources (hardware and software) that are offered as a service through a network (typically the Internet). A third definition, developed by a working group of the European Commission [10], considers Cloud Computing as an elastic performance environment for resources involving multiple actors to offer a service with a certain level of quality of service. This definition has been extended in [11] taking into account the perspectives of the different players in the Cloud Computing ecosystem (supplier, developer, user). However, the definition proposed by the National Institute of Standards and Technology (NIST) in [12], defines Cloud Computing as a model that allows access via a network in a simple and on-demand way to a set of shared and configurable computing resources. These IT resources can be allocated and released

quickly with minimal management effort or interaction with service providers. In addition, NIST states that Cloud Computing is composed of five essential features, three service models and four deployment models. These elements are listed below.

2.2 Service Models of Cloud Computing

Behind the term Cloud Computing mainstream hides an economic and technological trend that concerns all levels involved in services between a provider and a customer. It is also an application (the supplier runs the application for the client and returns the results), as well as the hardware (the supplier offers its rental equipment as well as a software infrastructure to exploit it). The current taxonomy declined in more precise ways cloud computing into three main categories [13]: Software as a Service (SaaS): Provides to the consumer the capability to use the supplier's applications running on a cloud infrastructure. So the applications are accessible from various client devices through either a thin client interface, like a program interface, or a web browser. The consumer does not control or manage the underlying layers of cloud infrastructure including such as network, servers, operating systems, storage. Platform as a Service (PaaS). Provides to the client the capability to deploy into the cloud infrastructure consumer-created using libraries, programming languages, tools, and services supported by the provider. The consumer does not control or manage the underlying layers of cloud infrastructure including operating systems, network, servers, or storage, but has possibly configuration settings for the application-hosting environment and control over the deployed applications. Infrastructure as a Service (IaaS): Provides to the client the capability to provision processing, networks, storage, and other computing resources where the consumer is able to run and deploy arbitrary software, which can include operating systems and applications. The customer has control over operating systems, storage, and deployed applications; but he has possibly limited control of select networking components.

2.3 Deployment Models of Cloud Computing

Generally, there are three main types of deployment models for cloud computing: private cloud, public cloud, hybrid cloud, and Community Cloud [14]. The Private Cloud: The infrastructure of a private cloud is only used by a single client. It can be managed by this customer or by a service provider and can be located at the premises of the client company or at the service provider, if applicable. Using a private cloud ensures, for example, that the allocated hardware resources will never be shared by two different clients. The Public Cloud: The infrastructure of a public cloud is publicly accessible or for a large industrial group. Its owner is a company that sells computers as a service. The Hybrid Cloud: The infrastructure of a hybrid cloud is a composition of two or three of the cloud types mentioned above. The different clouds that comprise it remain independent entities in their own right, but are linked by standards or proprietary technologies that allow the portability of applications deployed on different clouds. A typical hybrid cloud usage is the load balancing across multiple clouds during peak utilization rates. The Community Cloud, is used by many organizations with common needs. It can

host a very specialized business application that is common to several entities, who decide to federate their efforts by building a cloud to host and manage it.

3 SMA: System Multi Agents

3.1 Agent

An agent is a computer program that is located in an environment and that has autonomous behaviors (action) enabling it to achieve, in this environment, the objectives that were set during its conception [15]. A software agent is an autonomous entity capable of communicating, having a partial knowledge of its surroundings and private behavior, as well as its own capacity for execution. An agent acts on behalf of a third party (another agent, a user) that he represents without being necessarily connected to it, he reacts and interacts with other agents. The agent is capable of [16]: To act in an environment and to communicate directly with other agents, Is driven by a set of trends, and possibly to reproduce, also to own own resources and to perceive its environment, and to possess skills and to offer services, Behaviors tending to meet its objectives, moreover to perceive its environment in a limited way, having only one partial representation of this environment. Agent Features, An agent must be [17]:

- Autonomous: the agent is able to act without the influence or intervention of a human or agent and controls his own actions as well as his internal state;
- Proactive: the agent must exhibit opportunistic behavior;
- Social: the agent must be able to interact with other agents, especially when the situation requires it;
- Cooperation: able to coordinate with other agents to achieve the common objective;
- Mobility: the agent can be mobile, able to move to another environment;
- Rationality: the agent is able to act according to his internal objectives and his knowledge;
- Learning: the agent is able to evolve and learn; as a function of this learning, he is able to change his behavior.

3.2 System Multi Agent: SMA

The agent is the main component of multi-agent systems. According to Ferber [17] a multi-agent system is a system composed of the following elements:

- An environment E , that is to say a space generally having a metric.
- A set of objects O . These objects are located, that is to say that for any object it is possible, at a given moment, to associate a position in E . These objects are passive; That is, they can be perceived, created, destroyed and modified by agents.
- A set A of agents, which are particular objects, which represent the active entities of the system.
- A set of relations R that unite objects (and therefore agents) between them.
- A set of operations Op allowing the agents of A to perceive, produce, consume, transform and manipulate O objects.

4 Multi-agent System for Cloud Services Management

Several studies have established a mathematical model for modeling resource provisioning requirements in cloud computing, and these studies are based on the theory of queues mostly. The researchers proposed numerous multi-agent approaches in a Cloud context to overcome the limitations of the latter such as resource allocation and security threats. For example, in article [18], the authors proposed a combination of Agent technology with the Cloud to arrive at a method for calculating the resource management model. It reinforces the theory that the use of agents makes it possible to effectively achieve the management of resources in the Cloud. In Mansura's article [19], the integration of SMA in a Cloud context can allow high performance for complex systems and intelligent applications, proving that a reliable and scalable infrastructure can be Application on a large scale. SMAs have also been used in the cloud to provide an access management service. Indeed, in paper [20] emphasize that traditional systems are not sufficiently effective to support the functionality of access control in the cloud mainly due to the high scalability of the environment of cloud. They used a multi-agent system to define the accessibility and functionality of their model in order to improve the access control system. Other uses of multi-agent systems to provide Cloud-based security services have been proposed as for article [21] where the authors used SMA architecture to ensure privacy and availability for a collaborative storage service Hosted in the cloud, and also in a disaster management service presented in the paper. [22] The authors used a workflow model to help and maintain rescue and reorganization of disaster activities. Other work has addressed the concept of formal description of SMA.

Their studies aim to evaluate the integrated functionalities and to present the formal specifications of multi-agent systems. Typically the article [23] proposes a new collaborative formalism in SMA between agents combined with the Ferber model, characterized by the possibility of self-evaluation in the application of collaborative work. Their results show that the proposed formalism more effectively manages agent communication for better production.

5 SMAANQOS Framework (System Multi Agent for Automatic Negotiation of Quality of Services)

5.1 Motivation

The rise of embedded information systems in hosted mode has shown a marked decline since the beginning of the year 2010. This trend is explained by the arrival of an outsourced offering, notably in the market for business solutions on demand, driven by both the diversity and richness of the Cloud Computing operators' offerings and the customers' financial, technological and operational flexibility. In an economic approach, the Cloud is emerging as a promising commercial offer that can satisfy both applicants and digital service providers. Thus, SMEs are required to contract some of the services offered by Cloud Computing operators in order to obtain operational guarantees, in addition to the usual services between customers and suppliers. In a customer/operator

relational approach, the delicate concretization of a cloud service depends strongly on the service model targeted, ranging from the provision of software to that of infrastructures via the platform (Saas, PaaS, IaaS) and Model of deployment privileged (Public, Private, Hybrid, Community). So this conclusions cited above we decide to build a new framework of negotiation of QOS based on SMA in order to improve availability of services.

5.2 SMAANQOS Framework

SMAANQOS framework is based on JADE (Java Agent DEvelopment Framework) that is a multi-agent platform developed in Java by CSELT (Gruppo Telecom Research Group, Italy), which aims to build multi-agent systems and implement applications that comply with the FIPA standard (Platform Foundation for Intelligent Physical Agents, 1997). JADE consists of two basic components: a FIPA-compatible agent platform and a software package for the development of Java agents. Jade is a middleware that facilitates the development of multi-agent systems (SMA) (Fig. 1).

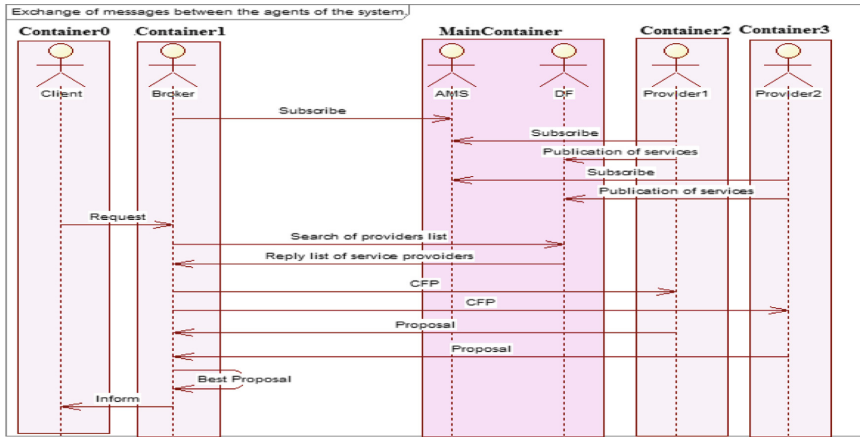


Fig. 1. SMAANQOS diagram of interactions

JADE contains: A runtime environment: the environment where agents can live. This runtime environment must be enabled in order to launch the agents, also a class library that developers use to write their agents and a suite of graphical tools: which facilitate the management and supervision of the agent platform. Each instance of the JADE is called a container, and can contain multiple agents. A set of containers is a platform. Each platform must contain a special container called the main container and all other containers register with the container as soon as they are launched. In our platform, Multi agents are compatible with FIPA, it includes three agents that are automatically created and activated when the platform is activated (Fig. 2):

- Agent Management System (AMS): is the agent who exercises supervisory control over the access and use of the platform; it is responsible for authenticating resident agents and checking records.
- Agent Communication Channel (ACC): is the agent that provides the route for the basic interactions between agents in the platform.
- Directory Facilitator (DF): is the agent that provides a yellow pages service to the multi-agent platform.
- Remote Management Agent (RMA): The RMA allows controlling the lifecycle of the platform and all its components.



Fig. 2. FIPA reference model of an agent [23].

Our framework is implemented using the JAVA language, under the environment Eclipse development. For the multi-agent system we used the JADE platform.

The choice of JAVA language was motivated by the following reasons:

- The agents developed under the JADE platform are entirely written in Java.
- Java ensures total independence of applications from the environment execution: any Java-enabled machine is capable of to run a program without any adaptation (neither recompilation nor parameterization of environmental variables).
- JAVA has a huge library of ready-to-use objects, which fully implement the implementation procedure.

Our framework runs as below:

- Step 1: The agents broker and agents providers subscribes by AMS.
- Step 2: Then the providers agents register their services.
- Step 3: Every second the broker contacts the DF in order to have information about the publication of the providers agents
- Step 4: When a client send a request for specific service, the broker contact the DF to have a list of providers of the service requested
- Step 5: After getting list of providers, the broker send CFP (call for proposal) to each one.
- Step 6: The providers responds to a CFP by sending their proposal.
- Step 7: The broker make a choice of the best proposal by calculate the best ratio availability, price.
- Step 8: At the end the broker inform client about the provider which has a best ratio.
- Step 9: if the client send another request for the same service while DF had not another entry for the same service the broker responds client with the same message if not the broker send another time CFP for each provider and calculate the best proposal.

N.B: If the broker will be not reachable or killed, the client switches automatically to another broker from the list.

Inside JADE container the message exchanged in JADE architecture components can be traced by a special JADE agent called SNIFFER Agent. (See Fig. 3) presents SNIFFER GUI.

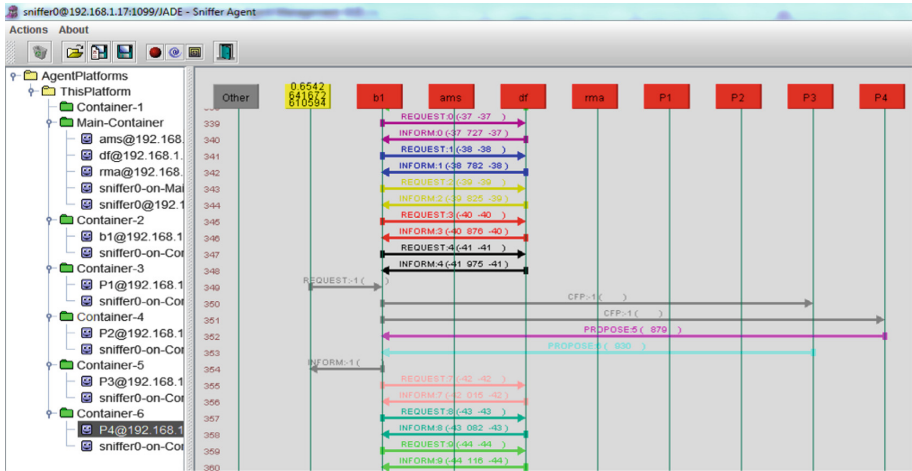


Fig. 3. Exchange of messages between the agents of the system.

An agent who wishes to publish a service must provide a description that includes: Its AID identifier, list of languages and ontologies that other agents must use to communicate with them, List of published services (Fig. 4).

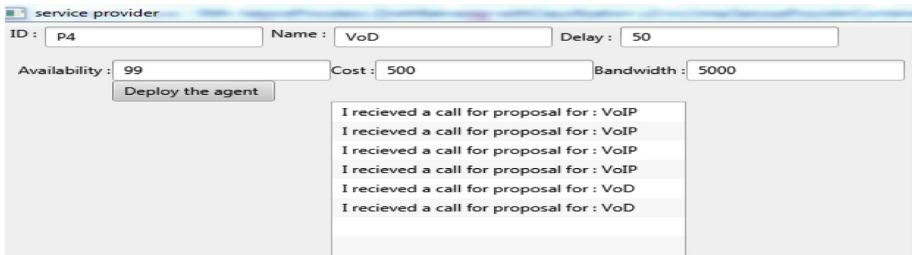


Fig. 4. Container of provider agent.

The Class diagram of Broker agent is modeled as below (Fig. 5):

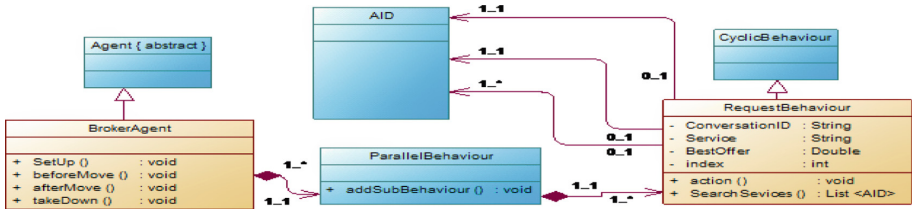


Fig. 5. Class diagram broker agent.

For each published service, indicate: Type of service, the name of the service, and the languages and ontologies to be used to operate this service. The Class diagram of Provider agent is modeled as below (Fig. 6):

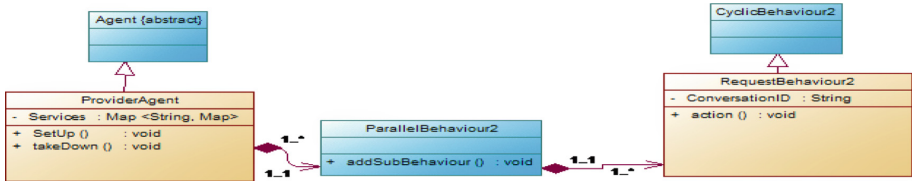


Fig. 6. Class diagram provider agent.

6 Conclusion and Future Work

The Cloud Computing has appeared as a new computing concept which has a purpose to provide reliable, customized, dynamic computing environments towards better IT infrastructure availability and quality of service without much financial burden. Therefore, the large and strong demand of services in cloud provokes and causes many collisions in the network. In order to relieve network and improve availability of services by automatic negotiation we were able developed a new framework based system multi agent technology called SMAANQOS used in a cloud computing as it simulates above. This article can be considered as a proof of concept of the SLA negotiation using SMA in a micro cloud computing. As the SMA developed is flexible and scalable, it can be extended to appropriate cases for the cloud computing environment and impose new algorithms for the automatic negotiation of Quality of Service. As a Future work, we will compare the performance of our proposal Framework to other existing solutions for negotiating access to cloud services.

References

1. Chaib-Draa, B., Jarras, I., Moulin, B.: Systèmes multi-agents: principes généraux et applications, pp. 1030–1044. Edition Hermès (2001)

2. Le Bars, M.: Un Simulateur Multi-Agent pour l'Aide à la Décision d'un Collectif: Application à la Gestion d'une ressource Limitée Agroenvironnementale. Diss. Université Paris Dauphine-Paris IX (2003)
3. Rajaei, H., Wappelhorst, J.: Clouds & grids: a network and simulation perspective. In: Proceedings of the 14th Communications and Networking Symposium, CNS 2011, pp. 143–150. Society for Computer Simulation International (2011)
4. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud computing and grid computing 360-degree compared. In: Grid Computing Environments Workshop, vol. abs/0901.0, pp. 1–10. IEEE (2008)
5. Höfer, C.N., Karagiannis, G.: Cloud computing services: taxonomy and comparison. *J. Internet Serv. Appl.* **2**(2), 81–94 (2011)
6. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.* **25**(6), 599–616 (2009)
7. Kushwaha, D.S., Maurya, A.: Cloud computing-a tool for future. *Int. J. Math. Comput. Res.* **1**(1), 09–14 (2013)
8. Ahmed, M., Sina, A., Chowdhury, R., Ahmed, M., Rafee, M.H.: An advanced survey on cloud computing and state-of-the-art research issues. *IJCSI Int. J. Comput. Sci. Issues* **9**(1), 201–207 (2012)
9. Schubert, L., Jeffery, K., Neidecker-Lutz, B.: The future of cloud computing - opportunities for european cloud computing beyond 2010. Technical report, European Commission (2010)
10. Schubert, L., Jeffery, K.: Advances in clouds: research in future cloud computing. Technical report, European Commission (2012)
11. Mell, M., Grance, T.: The NIST definition of cloud computing. Technical report 6, National Institute of Standards and Technology (2011)
12. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. *Commun. ACM* **53**(4), 50–58 (2010)
13. The NIST Definition of Cloud Computing, September 2011. <http://dx.doi.org/10.6028/NIST.SP.800-145>. Accessed 20 Oct 2017
14. Wooldridge, M.: An Introduction to MAS (2009)
15. Ferber, J.: Les systèmes multi-agents: vers une intelligence collective. Inter Editions (1995)
16. Briot, J.P., Yves, D.: Principes et architecture des SMA, Hermès (2001)
17. Wu, W., Zhang, X., Zheng, Y., Liang, H.: Agent-based layered cloud resource management model. In: 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII), vol. 2, pp. 70–74 (2013)
18. Talia, D.: Clouds meet agents: toward intelligent cloud services. *IEEE Internet Comput.* **16**(2), 78–81 (2012)
19. Habiba, M., Islam, M.R., Ali, A.S.: Access control management for cloud. In: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 485–492 (2013)
20. Amir, A.M., Rodziah, A., Rusli, A., Masrah, A.M.: Security framework of cloud data storage based on multi agent system architecture-a pilot study. In: International Conference on Information Retrieval and Knowledge Management (CAMP), pp. 54–59 (2012)
21. Mansura, H., Shamim, A.: MAS workflow model and scheduling algorithm for disaster management system. In: 2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM), pp. 164–173 (2012). 1-2-3-5-13-14-15

22. Khezami, N., Otmane, S., Mallem, M.: A new formal model of collaboration by multi-agent systems. In: International Conference Integration of Knowledge Intensive Multi-Agent Systems (KIMAS 2005), pp. 32–37 (2005)
23. Foundation for Intelligent Physical Agents. Specifications (1997). <http://www.fipa.org>. Accessed 20 Oct 2017