

2016 IEEE 4th International Conference on Future Internet of Things and Cloud

# Mohamed Shaheen Elgamel1

Arab Academy for Science, Technology & Maritime Transport (AASTMT)

Alexandria, Egypt [mohamed.shaheen@aast.edu](mailto:mohamed.shaheen@aast.edu)

*.*

1. INTRODUCTION

The main advantage of cloud computing is its capability to offer services that are characterized with high performance and rapid elasticity property besides, a large amount of storage that satisfy the needs of huge number of users. Cloud computing enlarged the arena of distributed computing systems by providing advanced Internet services that complement functionalities of distributed computing provided by the Web, Grid computing and peer-to-peer networks [1]. Multi agent systems (MAS) represent a distributed computing model, but MAS has a different view that is centralized on the cooperation among smart, intelligent and independent agents interacting with each other. One key feature of software agents is the intelligence that can be embodied into them according to some collective artificial intelligence approaches which need cooperation among several agents that can run on a parallel or a distributed computer to achieve the needed high performance for solving large complex problems keeping execution time low [2]. Accordingly, cloud computing can offer a very powerful, reliable, predictable and scalable computing infrastructure for the execution of Multi- agent systems (MAS). Thus implementing complex agent-based applications, especially when modelling, simulation and real- time running of complex systems must be provided. On the other hand, software agents can be used as basic components for implementing intelligence in Cloud computing systems making them more adaptive, flexible, and autonomic in resource management, service provisioning and in running large-scale

1 This publication was made possible by Information Technology Industry Development Agency (ITIDA), Egypt, Grant# PRP2014.R15.11, "Cloud-

285

978-1-5090-4052-0/16 $31.00 © 2016 IEEE DOI 10.1109/FiCloud.2016.48

285

978-1-5090-4052-0/16 $31.00 © 2016 IEEE DOI 10.1109/FiCloud.2016.48

# 

applications. This paper presents some research directions for using intelligent software agents to improve cloud performance and functionality. The paper proceeds as follows: Section II introduces Cloud computing concepts. Section III discusses multi-agent systems. Section IV presents how intelligent software agents are used to improve the performance and functionality of Clouds. Section V introduces some Performance Measurements that are used to evaluate the agent.

Section VI presents research issues and challenges which show problems definitions and list of open challenge area that readers can work on. Finally, the last section concludes the paper and introduces future directions.

1. CLOUD COMPUTING
2. *Definition*

There are many different definitions of cloud computing. More than 20 definitions were discussed and compared in [3]. The National Institute of Standards and Technology (NIST) has given a complete reference definition that covers and summarizes the main concept of cloud computing. NIST defined cloud computing as: *Cloud computing is a model for enabling 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.* [4].

1. *Essential Characteristics*

NIST has listed 6 essential characteristics for cloud, and whenever one of this is not there, it could not be named a cloud [4]. (1) On demand self-services, (2) Broad network

access, (3) Resource pooling, (4) Rapid elasticity, (5) Measured service and (6) Multi tenacity.

1. *Service Models*

There are three main services that have come to define cloud computing and how end users can access their resources and services according to NIST as illustrated in Fig. 1.

* 1. *Infrastructure as a service (IaaS):* IaaS provides infrastructure components to clients. Components may include virtual machines, storage, networks, firewalls, load balancers, and so on. With IaaS, clients have direct access to the lowest- level software in the stack that is, to the operating system on

based 3D Teleradiology for Rural Areas" from Nov. 2014 Jan. 2016. The statements made herein are solely the responsibility of the authors.

## 



978-1-5090-4052-0/16 $31.00 © 2016 IEEE DOI 10.1109/FiCloud.2016.48



285

virtual machines, or to the management dashboard of a firewall or load balancer.

* 1. *Platform as a service (PaaS):* PaaS delivers a pre-built application platform to the client, clients do not need to spend time building underlying infrastructure for their applications. On the backend, PaaS automatically scales and provisions required infrastructure components depending on application requirements. Typically, PaaS solutions provide an API that includes a set of functions for programmatic platform management and solution development. Google App Engine is a popular PaaS provider.
  2. *Software as a service (SaaS*): SaaS provides ready online software solutions. The SaaS software provider has complete control of application software. SaaS application examples include online mail, project-management systems, customer relationship management CRMs, and social media platforms.

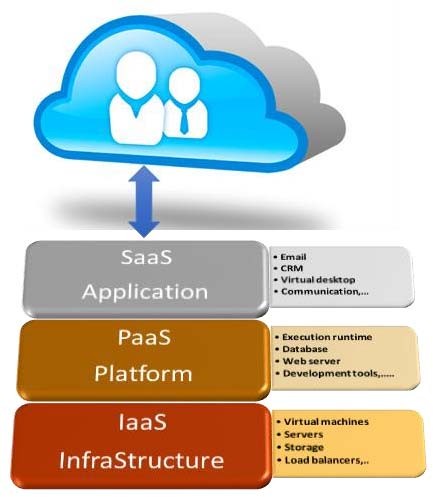


Fig. 1. A cloud service Models.

1. *Deployment Models*

There are four deployment models of cloud. (1) Public Cloud: owned and operated by a cloud provider, (2) Community Cloud: owned and operated by organizations or by a cloud service provider, (3) Private Cloud: managed by the organization or a third party, (4) Hybrid Cloud: Is a combination of different methods of resource pooling

1. AGENT-BASED COMPUTING

Agent was first used by *Yoav Shoham* within his Artificial Intelligence studies, in 1990. His agents are specific to his own paradigm as they have just one method, with a single parameter. An agent is any entity whose state is viewed as consisting of mental components (e.g., beliefs, capabilities, choices, and commitments) [6][7]. An agent is a computer system that is situated in some environment, and is capable of autonomous action in this environment in order to meet its design objectives [8].

1. *Intelligent Agents*

[8] define an intelligent agent characteristics as:

(1) Reactivity: Intelligent Agents perceive their environment and respond to changes that occur in it in a timely fashion as shown in Fig. 2, (2) Pro-activeness: Intelligent Agents are able

to exhibit goal-directed behavior by taking the initiative. (3) Social ability: Intelligent Agents interact with other agents via some kind of agent-communication language (negotiation, cooperation) to satisfy their design objectives. (4) Autonomous: Agents work without the direct intervention of humans, and have some kind of control over their actions and internal state [9].

Fig. 2. A simple agent in its environment. (Modification of figure taken from [5])

1. *Intelligent Agents vs Smart Agents*
   1. describe intelligent agents in terms of a space defined by the three dimensions of agency, intelligence, and mobility as shown in Fig. 3:

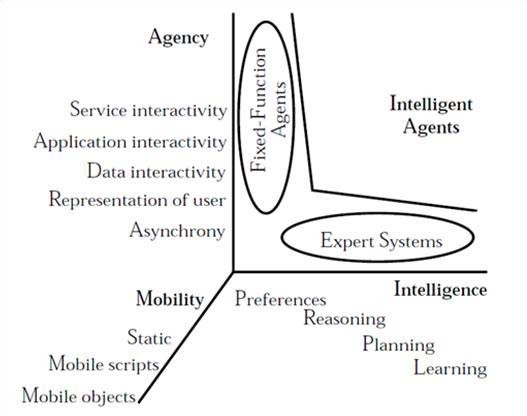


Fig. 3. Scope of intelligent agents [10].

* 1. proposes a typology of agents that identifies other dimensions of classification. Agents may thus be classified according to primary attributes, such as autonomy, cooperation, and learning. From these characteristics, there are four agent types: collaborative, learning, interface, and smart. Smart agents can combine all idle and primary attributes as shown in Fig. 4.

1. *Multi agent system*

A multi agent system is one that consists of a number of agents, which interact with one-another. In the most general case, agents will be acting on behalf of users with different goals and motivations. They will require the ability to cooperate, coordinate, and negotiate with each other as much as people do.

The characteristics of MASs are defined as follows [12]:

(1) Each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint, (2) There

is no global control system, (3) Data are decentralized and (4) Computation is asynchronous.

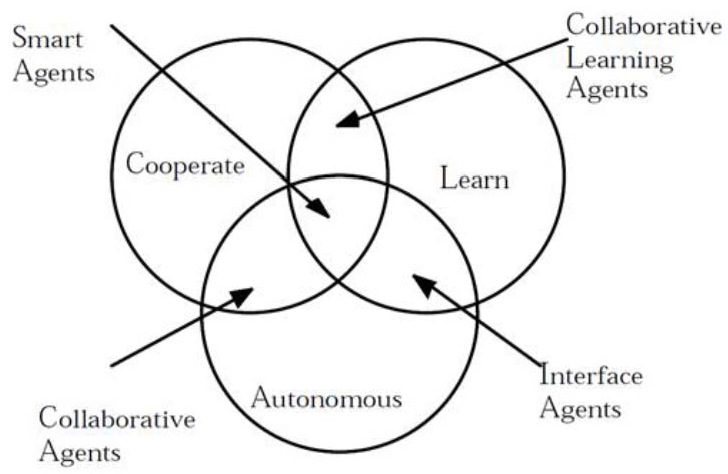


Fig. 4 [11].

1. *MAS architecture*

A basic structure of a MAS can be broadly classified into the following architectures [13]. (1) Centralized: collection of simple homogeneous, non-communicative agents that are managed by a single control center in a master-slave relationship, (2) Distributed: collection of communicative agents managed by a single layer control structure. Each local agent is responsible for and has knowledge of its own part in the network, indeed no single agent has complete knowledge of the whole domain, and (3) Hierarchical: some agents having authority over the actions of other agents.

1. *Development platforms*

MAS has four stages for platform development process [14]. Analysis (application domain and problem), Design (solution architectures), Development and Deployment.

Building a MAS requires an agent development environment that supports at least some stages of the MAS conceptual design process. Researchers used many platforms to develop MAS that survived and compared in [15]. The most common tools used are JADE and Jadex. JADE (Java Agent Development Framework) is a Java framework for the development of distributed multi agent applications [16]. JADE implemented an agent middleware that has multi services to make agent creation easy with graphical tools for debugging and testing. Jadex is also a software framework for the creation of agents. Jadex has the main functions of JADE and is considered as an extension of JADE, but the main goal of Jadex is its follow the Belief-Desire- Intention (BDI) model. The framework is realized as a rational agent layer that sits on top of a middleware agent infrastructure such as JADE [15]. There are more platforms used to implement intelligent agents [17]. Platforms and languages

domain are introduced and in addition, features, advantages and

disadvantages are summarized in [17].

1. AGENT-BASED CLOUD COMPUTING

A cloud service life cycle (CSLC) consists of: service requirements, service discovery, service negotiation, service composition, and service consumption [18] as shown in Fig. 5. A new discipline, called agent-based Cloud computing must be set for providing agent-based solutions founded on the design

and development of software agents for improving Cloud resources, service management, discovery, service-level agreements SLA negotiation, and service composition. Autonomous agents can make Clouds smarter in the interaction with users and more efficient in allocating processing and storage to applications [2].

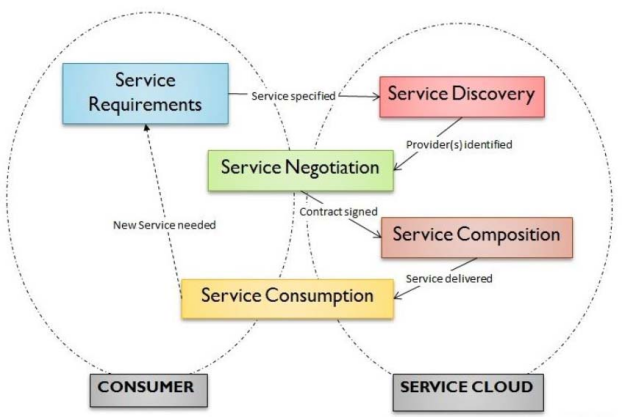


Fig. 5. A cloud service life cycle

There is no formal taxonomy for a research area that focuses on cloud using MAS. This paper will discuss and represent the interaction between agent and cloud computing throw the cloud service life cycle as shown in Fig. 6. It analyzes how an agent can manage and help the cloud in each service. It also reviews some research activities in different CSLC phases based on the proposed classification as shown in Table 1. It introduces the functionality of each phase and shows different researches in this phase.

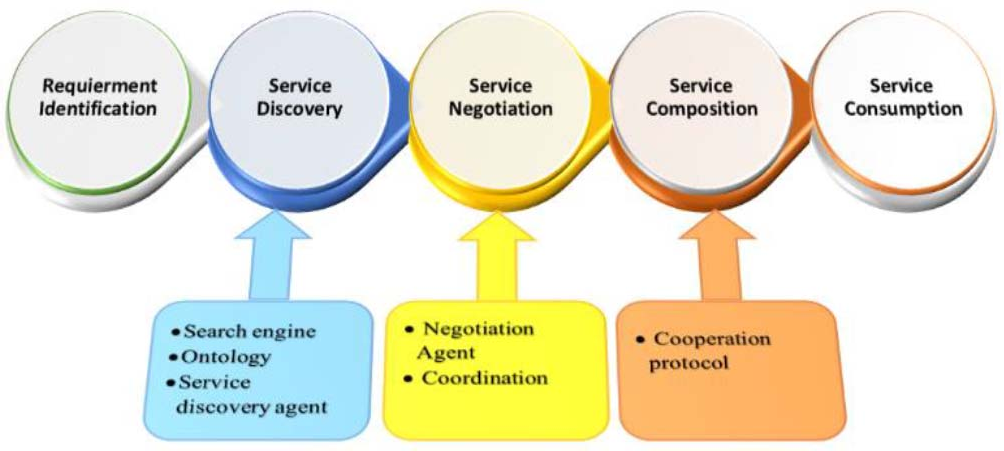


Fig. 6. Agent in Cloud Service Life cycle

1. *Service requirements*

Consumers provide details of the functional requirements (functions and tasks that a service should provide), technical requirements (e.g., hardware and operating systems), and budgetary requirements (acceptable service cost) [19].

1. *Service discovery*

The challenge in the service discovery phase is to run a query

database by  budgetary requirements [19].

TABLE 1 RECENT RESEARCH BASED ON THE CSLC.

|  |  |  |  |
| --- | --- | --- | --- |
| year | Agent-based Cloud Service Life Cycle | | |
| *Service Discovery* | *Service Negotiation* | *Service Composition* |
| 2010 | [31] |  | [42], [44] |
| 2012 | [27], [29] | [36] | [43], [45] |
| 2013 | [30] | [33] | [46] |
| 2014 | [28] | [34], [35], [39], [40] |  |
| 2015 |  | [37], [38], [41] | [47] |
| 2016 | [32] |  |  |

[27] Proposed a system that used multi-agent and ontology, it developed a prototype for service discovery in cloud environment. The system can help customers (Users) discovering appropriate service according to user demand. The system consists of three layers. The Application Layer represents the User Agent (UA) that has the ability to enable users to make a query in natural language and get a reply. Also the UA can interact with other agents in the system, Context Layer include major agents of the system that have Reasoning Agent (RA) and Discovering Agent (DA), and a domain knowledge repository. The main contribution of this study is the use of the basic power of semantic that makes the search technique more easy to use. Recall and Precision are calculated to evaluate the accuracy of the system. It shows that the system can effectively discover the user desired cloud services automatically. [28] Also introduced SMARTSPACE platform that is based on the multi-agent based distributed semantic service discovery. Therefore in SMARTSPACE, services and

used to avoid such issues. Every

service input and output parameters are defined using a concept from a global domain ontology serving as a base ontology. Similarly, query input and output parameters are assumed to be mapped to concepts from the same global ontology. This work also introduced a distributed algorithm called SmartDiscover that consists of three components SmartCluster, SmartDirect and SmartMap. SmartCluster is responsible for efficient clustering of service descriptions into a semantic taxonomic cluster (STC) space. SmartDirect and SmartMap algorithms are responsible for satisfying a user query by exploiting the structural and semantic properties of the STC space to efficiently discover the relevant service agents. [29] Presented the same approaches like previous articles but in different way using the new strong agent property called mobile agent, in fact mobile agent is an entity which can migrate from a platform to another in order to carry out one or more specific task. Also mobile agent has the ability of synchronism, autonomy and auto-adaptability. In this article the author proposed a web services discovery system which consists of three parts. Web Interface Part, Cloud Computing Part and Mobile Agents Part. The Mobile agent part include some agent with different tasks. Interface Agent (Ia) is used to transmit the client request to the region A and display selected web services to client via the web interface, Researcher agent (Ra) is used to do a syntactic research based on keywords. The last agent is Transport agent (Ta) that is used to transport the client request from the region A to the region B, and on the other hand, has to transport the web services detected by the researcher agents to the region B, and then has to transport each Web service selected by the region B to region A. The main advantage

of using the mobile agent is in case of fault tolerance in any region (A, B) the mobile agents decide (autonomy and auto- adaptability) to keep the client's request until the fault is set. Unlike the previous study [27],[28] that used a semantic technique and shared ontology to satisfy a user query [29] designed an algorithm called comparison and filtering to solve the problem. [30] Proposed a support system for the discovery and selection of public cloud infrastructure services to construct a hybrid cloud computing environment using ontologies and agents. The main idea of the study is a support system to search for appropriate public cloud services and monitor the hybrid cloud condition. That system monitors the network utilization of a public cloud via an agent and store such information in ontology and allow the selection of public cloud services with due consideration to the current state, structure, and contents of the service. Also collect the dynamic information of the in-house private cloud using agents and store them in the ontology database. [31] Proposed an agent-based scheme to discover comprehensive service, select and allocate resources for supporting the cloud applications in wireless platforms with respect to efficiency and fairness of resource utilization. In addition, it presents an optimization resource selection strategy with selection and allocation of network resources based on the agent-based scheme in enabling Quality of Service (QoS) in wireless cloud environment. The goal of that work is to realize a service-oriented system for supporting and maintaining the



using agent based scheme architecture that consists of different properties, wireless AN resources, user wireless terminal (WT) equipped with a wireless user agent (WUA), and agent-based server (ABS) that acts as a gateway entity and to discover the requested cloud service from the application server providers and selects and allocates the best resources to support and maintain s applications. [32] Presents four-stage, agent- based Cloud service discovery protocol. Additionally, two Cloud ontologies (CO-1 and CO-2) are designed to semantically define the relationship among Cloud services. Whereas CO-1 contains only Cloud concepts, CO-2 contains a set of Cloud concepts, individuals of those concepts, and the relationship among those individuals. The similarity among Cloud services is determined by concept, object property, and data type property similarity reasoning. In addition, two kinds of recommendation approaches (R1 and R2) based on attribute value prediction are presented. R1 is based on the maximum and R2 on the average similarity between the provided and the requested requirements. The results show that system achieved the best performance in finding the appropriate Cloud services with CO-2 and R2.



1. *Service negotiation*

Consists of message exchanges between consumers and brokers, and between brokers and providers for establishments of service-level agreements. The challenge in cloud service negotiation is to establish SLAs between consumers and brokers, and between brokers and service providers. Many researchers presented different ways for using agent based negotiation system [33][34][35][36][37][38][39][40][41].

[33] Present (EMan) a multi-agent elasticity management solution that enables the Application Service Providers (ASP) to

respond to the conflict between satisfying the ASP business goals and the end-user preferences. Implementing a multi-agent approach allows the use of automated negotiation as a conflict resolution technique to achieve useful trade-offs and win-win settlements between the negotiating parties. That work uses negotiation protocol that allows agents to make offers and counter-offers. This process continues until an agreement is reached, or the negotiation deadline of one of the parties is reached. Agents in the EMan architecture combine two types of negotiation strategies: trade-offs and concessions. [34] Also designs a multi-agent based negotiation model under the circumstances of cloud computing. The negotiation framework contains three components: multi-agent system, application interface layer and cloud resource market. Multi-agent system is the platform for the service trade, agents represent resource user, resource supplier and intermediary. Application interface connects the multi-agent system to cloud resource market. Cloud resource market contains all the resources used for service trade. Multi agent system consists of negotiation framework three agents, Resource User Agent, Resource Supply agent and Intermediary Agent. Resource user and resource supply agents catch information over intermediary agent, and negotiate with rival agents. Resource supply agents hold the resource in the cloud computing market. Intermediary Agent interacts with cloud resource market through application interface layer to get market information. The negotiation model was applied to the Resource Allocation of CloudSim and achieved cloud resource allocation in the simulation experiments. The results show that the negotiation model efficiently applied to cloud resources could get higher effectiveness. [35] Raises a different view concerning semantic negotiation protocol. Instead of client and service providers the agent that has intelligent behavior can

perform SLA negotiation, but as human beings, they need to

converge on each common meaning of the terms used in Communications and services to be effective Discussions on the SLA. This protocol allows the parties of an SLA negotiation to explain the semantic of the used terms and to transform the used technical terms in a shared form which is understandable. The used negotiation protocol consists of a number of standard ACL messages between different agents (customers, providers), global database EDB and the local databases UDB, represented



and understanding capability coefficients. The main purpose of this protocol is to solve the misunderstanding problems that occur when having to negotiate SLA with different providers.

[36] Introduced automated SLA negotiation for cloud resources between a consumer and a number of providers. The consumer enters into English auction, while providers are equipped with a sort of intelligence, agents should be in place for all parties during the negotiation, test whether such a condition is possible, answering questions concerning which intelligence is better. The main agent in this study is Q-learning agent, which gives efficient results in the case of dynamic and random environments according to the fact that it does not need a pre-defined model of the environment for the success of the learning process. The same concept as used in [37], that introduced agent-level negotiations to manage the QoS of a

cloud-based distributed data service. Agent dynamically learns about the system behavior and the challenger agent goals, as part of an effective negotiation strategy. The learning and negotiation, occurring hand-in-hand, purport to move the system towards an optimal behavior. The system is based on two agents, system-level service provider and application-level users (ALU). ALU learns from past decision errors and adjusts the QoS needs and offers to converge to a stable solution. This agent used the same negotiation protocol introduced by [19], also the same approach of learning capability as Q-learning [36] was applied. Different approaches are presented in [38], where most of the work focused on the areas of SLA negotiation, resource management and cost optimization in the cloud. In that work, the authors proposed a multi-agent system based approach for monitoring SLA violations while using MDE (Model Driven Engineering) to express the SLA contract requirements. MASs multi agent system consists of a set of intelligent and reactive agents to monitor the SLA violation. There are six agents. Provider/Consumer agents used for communication between service providers and service consumers, SLA establishment agent holds the information about the SLA established between the signatories, Monitoring agent monitors the SLA parameters, and detects violations, Penalty agent is a reactive agent equipped with an efficient reacting model, the last agent is Database agent that is used to retrieve or to insert information in the knowledge database. The main contribution is to solve the trade-off between the advantage of the supplier and the customer satisfaction.

In [39] a framework was proposed for achieving automated

SLA negotiation between providers and consumers for cloud computing resources using state-of-the-art automated negotiation algorithms/agents. A negotiation protocol called

is based on agent communication (Agent-to-Agent) over a series of rounds. There are seven agents introduced in that paper that are responsible for following different kinds of strategies to perform each action in the negotiation protocol. The agents are informally termed. (1) (introduced in [21] that has a low computational complexity, this enables it to generate bids very rapidly.), (2) *Tit-for-Tat*, -for-



d in [22]

aiming for the Nash point of the negotiation scenario. (3)

*Hardliner*, introduced in [23], has a main task which is the

-it-or-leave-



This strategy makes a bid of maximum utility for itself and never concedes, (4) *IAMhaggler*, this agent consists of three main parts, the first part predicts the concession of the opponent by using a Gaussian process regression technique, this technique is introduced and discussed in [24]. The second part sets the concession rate in such a way that it optimizes the expected utility given that prediction. The third part generates a multi- issue offer according to the concession rate, (5) *Gahboninho*, This agent strategy is that when the opponent attempts to propose a realistic and compromising offer, there is no need to compromise, (6) *AgentFSEGA*, This agent follows a Bayesian learning strategy introduced in [25] that adapted to handle with time limits and (7) *WiseH-T*, The Bidding Opponent Acceptance

(BOA) framework [26] is used to form the agent Wise H-T. The BOA negotiation agent architecture allows researchers to re-use existing components from other BOA agents. [40] like [39] presented a robust negotiation strategy which can be employed in service-based markets to dynamically negotiate near optimal and fair SLAs. That study not only introduced a new model but also used the ready system that enables improvement of difficulties in the design process of general automated negotiators termed GENIUS, a General Environment for Negotiation with Intelligent multipurpose Usage Simulation, introduced and implemented in [20] that actually supports the design of general automated negotiators, and even enables the



their generality. It also acts as real life negotiations. The main purpose of [40] is to design a negotiation strategy named

-for-

[39] which produces concessions if the opponent is perceived to be doing the same and vice versa. In [41] an agent-based negotiation model for knowledge exchange among service architectures was introduced. But this study differs from previous ones introduced a different view on agents' Society, Community, and agent Clusters that can be applied to software agents to improve the communication between users and cloud services. It also introduced a notion of Concerns that can represent the cloud service itself, and the fact that a concern can also be perceived as a sub-concern from less abstract level of the society tree, and so on until a sub-concern is linked to a member of a society that ends at that level of abstraction. Negotiation issues proposed in this study aim to define the communication between two agents in the same *service-centric community* that all agents are aware of the community's abstract concern / provided service. The agents are working on their negotiation, and may agree on its realization. Depending on the sets of *requests*, each negotiating agent is looking for *requests* to achieve. A *union* between two agents reflects the completeness of a unique community's sub-concern, an agreement between two agents can be achieved under different conditions. There is a central agent that called *a head- agent*. This *head-agent* is the managing authority of a community. So, the link is between an agent's possible situations of approval with its *requests*. Also putting together the sets of *requests* of an agent and the *tradeoffs.* It is ready to offer and connect them with different situations that may occur successful negotiations.

1. *Service composition*

Broker combines a set of services from multiple providers, and delivers the combined service as a single virtualized service to a consumer. The challenge in cloud service composition is to dynamically put together a set of services provided by multiple service providers to form a single unified service to be delivered to a consumer [19].

[42] offered the market-based model on cloud computing environment. It allows buyers to order an arbitrary composition of services from different service providers. That article introduced two mechanisms allocation and pricing, it also for developed a simulation environment based on agent software to evaluate the performance of the system. Market-based model

consists of two inter-dependent mechanism: allocation mechanism and pricing mechanism. Every mechanism is represented as an agent; the first one is an agent bidding that message and allocation mechanism, the second one is pricing mechanism (k-Pricing), it depends on double-sided combinatorial auctions. The basic idea is to distribute the welfare generated by the allocation mechanism between buyers and service providers according to a factor k [0, 1]. The simulation environment based on agent simulation development, consists of buyer/service provider agents and a market exchange. [43] Introduced a BDI model and multi-agent oriented approach to decouple autonomous management functionality with Cloud APPlication CAPP, and simplify the management policy configuration process. BDI model aims to optimize performance and resources utility efficiency improvement. CAPP Autonomic Management System Model



(CE) layer. In [44] self-organizing agent-based service composition framework is presented based on agent cooperative technique. Also a test bed that evaluated and demonstrated the advantages of self-organizing agents in Cloud service composition was implemented. [45] Introduced middleware based approaches for web services composition for end to end scenario including atomic services registration, composition and execution. The framework uses distributed algorithms for composition and execution. The framework provides services for registration and deregistration, Service composition, QoS optimization and Services execution. The framework consists of a set of Service Agents that coordinate with each other by executing a distributed algorithm to plan a composition workflow, QoS Optimizer: optimizes the services selection for the workflow based on QoS information, Executor Service: Takes care of executing the composite service, the last is Publisher: takes care of registering and deregistering of services by Service Providers. The advantage of this study is the avoidance of the single point of failure. Different agents work in parallel, so the throughput is increased. While one agent is busy serving one user, another agent can serve another user, thus enabling the application response to multiple users at the same time. [46] focus on demonstrating the effectiveness of adopting agent-based techniques for Cloud service composition by showing the desirable property that agents can autonomously and successfully deal with changing service requirements through self-organization and collaboration. [47] building information retrieval system (IR) for specific field (music field in that paper) using related ontology to generate mobile agent moving around cloud nodes.

1. *Service consumption*

Service consumption is the end of CSLC, which all phases have completed perfectly and based upon which the service will be delivered to the consumer.

1. PERFORMANCE MEASUREMENTS

This section aims to list some performance measurements that are used to evaluate agents in each CSLC phase.

1. *Service discovery:*

The well-known evaluation approach of information retrieval to evaluate the correctness of the system is Precision and Recall [48].

*Precision = # (relevant items retrieved) / # (retrieved items)*

*= P (relevant | retrieved).*

*Recall = # (relevant items retrieved) / # (relevant items)*

*= R (retrieved | relevant).*

The Evaluation of the Discovery Service Agent (DSA) can be performed using three different approaches [28]:

* 1. By Measuring the runtime efficiency and message overhead of the DSA algorithm.
  2. By measuring the efficiency of the SDA algorithms for service retrieval, i.e., by measuring the variation of the query response time for a different number of clustered services in the system.
  3. By measuring the accuracy of SDA and comparing it with the widely accepted benchmark service discovery algorithm. The accuracy was tested in terms of precision and recall.

1. *Service negotiation:*

There are two criteria to evaluate each agent in a negotiation protocol. (Performance and Fairness) [39]. Performance is the sum of the all utilities the agent has while negotiating with other agents, while the Fairness takes care of utility of the opponent in addition agent utilities. The Performance and Fairness are measured between 0 and 1.

1. *Service composition:*

Performance can be measured by [44]: (1) Number of successful service compositions, (2) Number of messages, these were used as the baseline for evaluating the capabilities of agents and the efficiency of the service composition method*.*

1. RESEARCH ISSUES AND CHALLENGES

This section lists some of the open challenges for new research area related to agent-based cloud computing that light the road to researchers who are interested to work in this topic. *The state of the art of agent-based cloud computing* is design and development of software agents for bolstering cloud service discovery, service negotiation, and service composition [19]. First approaches aim to know what are the problems or problems definitions in each service.

*Service discovery*, the problem of building a search engine for cloud services is more complex because one needs to search for services that satisfy three types of requirements (functional, technical and budgetary).

*Service negotiation*, the problem of devising a complex negotiation mechanism for cloud commerce is much more complex because a complex cloud negotiation mechanism specifies parallel negotiation activities among three types of participants (consumers, brokers, and providers). In multiple interrelated markets

*Service composition*, the problem of making various



service capabilities.

*List of open challenges:*

The scalability of services is a challenging aspect of distributed application processing in cloud computing. Automatic methods for matching SLAs are essential. Negotiation of other issues such as QoS and time slot negotiation.

The effect of the increase and decrease of the deadline to the negotiation outcome is investigated.

Robust negotiation strategy which can discover low- conflict domains suitable for fault tolerant services and include metrics like selfishness and altruism.

Deploying the agent-based architecture in a semantic web service framework.

Design search engines that decrease query response time and the number of message exchanges.

Designing a framework in real-time to evaluate the scalability of the agent-based cloud service composition approach.

1. CONCLUSION AND FUTURE WORK

This work shows how the collaboration and interaction between clouds and agents can benefit each other. Agents can run on the parallel distribute nature of the cloud and the cloud can be managed by the intelligent multi-agent running on them. It explores different research areas of the intelligent Cloud services and the high-performance multi-agent systems on Clouds. This study introduces a blue print taxonomy to the used agent in cloud computing and how multi-agent systems are integrated in every phase in the cloud computing life cycle, agent based (discovery, negotiation and composition) services. Different negotiation protocols are introduced in different modules based on intelligent agents, coordination and cooperation agents to manage and enhance cloud computing. It explained how multi agent can give different solutions to solve the service-level agreements problems that aim to satisfy all customers and provide , time, reliability and availability). Also test beds models designed by multi agent to manage and resource allocation in cloud computing. In summary agent-based cloud computing introduce: Intelligent and flexible Cloud services, Autonomous and pro-active services and Autonomic Clouds self-management. In the future we will cover how cloud computing architecture can help multi agent systems. We will show how complex agent-based applications or large- scale simulations based on MASs often require high- performance computing systems and large data storage devices. So, cloud infrastructures can offer an ideal platform to run MAS- based simulations and applications.

REFERENCES

1. Dinesh R, Ashwin R, "Multi-Agent based Cloud Services," in International Conference on E-Governance & Cloud Computing



1. D. Talia, " Clouds Meet Agents: Toward Intelligent Cloud Services,"

participants are allowed to breach contracts by paying penalty fees.

*Internet Computing, IEEE,* vol. 16, no. 2, 2012.

Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, Maik Lindner, "A break in the clouds: towards a cloud definition, " *ACM SIGCOMM Computer Communication Review*, v.39 Number 1, January 2009.

1. "NIST," 22 December 2015. [Online]. Available: [www.nist.gov/itl/cloud.](http://www.nist.gov/itl/cloud)
2. S. Russell and P. Norvig, Artificial Intelligence A Modern Approach (SECOND EDITION), Prentice Hall Series in Artificial Intelligence, 2003.
3. Y. Shoham, " Agent-oriented programming," Technical Report STAN- CS-1335-90, Computer Science Department, Stanford University, Stanford, CA 94305, 1990.
4. Y. Shoham, "Agent-oriented Programming," Artificial Intelligence, vol. 60, no. 1, pp. 51-92, March 1993.
5. M. J. Wooldridge, Ed., Introduction to multiagent systems, 2nd edition, Wiley, 2009.
6. M. Wooldridge. "Agent-based computing." *Interoperable Communication Networks.* Vol l, Number 1, pp. 71-97. January 1998
7. Gilbert, D.; Aparicio, M.; Atkinson, B.; Brady, S.; Ciccarino, J.; Grosof,



1995. IBM Intelligent Agent Strategy, IBM Corporation.

1. Nwana, H. S. 1996. Software Agents: An Overview. Knowledge Engineering Review, 11(3): 205-244.
2. Sycara, K.P. (1998) Multiagent Systems, Artificial Intelligence, summer 1998, pp. 79-92.
3. A. Kantamneni, L. Browna, G. Parker and W. Weaver, "Survey of multi- agent systems for microgrid control," *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 192-203, 2015.
4. P. M. Ricordel and Y. Demazeau, "From analysis to deployment: a Multi-Agent Platform Survey," *Engineering Societies in the Agents World. Springer,* pp. 93-105, 2000.
5. R. H. Bordini, L. Braubach and M. Dastani, "A Survey of Programming Languages and Platforms for Multi-Agent Systems," *Informatica,* vol. 30, pp. 33-44, 2006.
6. S. Prakash and A. Singh, "A SURVEY OF IMPLEMENTATION OF DISTRIBUTED MULTI AGENT SYSTEM USING THE JADE TECHNOLOGY," *International Journal of Advanced Technology & Engineering Research (IJATER),* pp. 71-75, 2014.
7. K. Kravari and N. Bassiliades, "A Survey of Agent Platforms," *Journal of Artificial Societies and Social Simulation,* vol. 18, no. 11, 2015
8. K.P. Joshi, T. Finin, and Y. Yesha, "Integrated Lifecycle of IT Services

*Proc. Third Int'l Conf. Virtual Computing*

*Initiative (ICVCI '09)*, 2009.

1. K. M. Sim, "Agent-Based Cloud computing," *IEEE Transactions on services computing,* vol. 5, no. 4, October-December 2012.
2. Lin Raz, S. Kraus, T. Baarslag, D. Tykhonov, K. Hindriks, and C. M.

*GENIUS: An Integrated Environment for Supporting the Design of Generic Automated Negotiators*

2008.

1. *HardHeaded*



Complex Automated Negotiations: Theories, Models, and Software Competitions, pp. 223 227, Springer-Verlag, 2013.

1. [22]



Strategy for Real-

*Complex Automated*

*Negotiations: Theories, Models, and Software Competitions, pp. 223 227, Springer-Verlag*, 2013.

1. T. Baarslag, K. Fujita, E. H. Gerding, K. Hindriks, T. Ito, N. R. Jennings,



Practical Negotiating Agents: Results and Analysis of the 2011



*the MIT Press*, 2006.

-constrained SLA negotiation

*in competitive computational grids, Future Generation*

1. [24]
2. [25]

*Computer Systems*, 28 (8), P1303-1315, 2012.

1. T. Baarslag, K. Hindriks, M. Hendrikx, A. Dirkzwager, C. Jonker, egotiating agents to explore the space of negotiation *In: Proceedings of the 5th International Workshop on Agent-*

*based Complex Automated Negotiations*, ACAN, 2012.

1. Y.-S. Chang, T.-Y. Juang, C.-H. Chang and J.-S. Yen, "Integrating Intelligent Agent and Ontology for Services Discovery on Cloud Environment," in IEEE International Conference on Systems, Man, and Cybernetics, 2012.
2. S. Dasgupta, A. Aroor, F. Shen and Y. Lee, "SMARTSPACE: Multiagent Based Distributed Platform for Semantic Service Discovery,"

IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, vol. 44, no. 7, 2014.

1. S. Hamza, B. Aïcha-Nabila, K. Okba and A. Youssef, "A Cloud computing approach based on mobile agents for web services discovery," in *Innovative Computing Technology (INTECH), 2012 Second International Conference*, 2012.
2. T. Uchibayashi and B. Apduhan, "A Framework of an Agent-based Support System for IaaS Service Discovery," in 13th International Conference on Computational Science and Its Applications, 2013.
3. Y. Zhou and Y. Yang, "An Agent-based Scheme for Supporting Service and Resource Management in Wireless Cloud," in Ninth International Conference on Grid and Cloud Computing, 2010.
4. J. Kang and K. M. Sim, "Ontology-enhanced agent-based cloud service discovery," Int. J. of Cloud Computing, vol. 5, no. 1/2, pp. 144 - 171, 2016.
5. A. Najjar, X. Serpaggi, C. Gravier and O. Boissier, "Multi-Agent Negotiation for User-Centric Elasticity Management in the Cloud," in *IEEE/ACM 6th International Conference on Utility and Cloud Computing*, 2013.
6. J. Chen, X. Han and G. Jiang, "A Negotiation Model Based on Multi- agent System under Cloud Computing," in *The Ninth International Multi-Conference on Computing in the Global Information Technology (ICCGI)*, 2014.
7. F. Messina, G. Pappalardo, D. Rosaci and C. Santoro, "An Agent based Negotiation Protocol for Cloud Service Level Agreements," in *IEEE 23rd International WETICE Conference*, 2014.
8. Morar, G.A.; Ilea, A.; Butoi, A.; Silaghi, G.C., "Agent-based cloud resources negotiation," Intelligent Computer Communication and Processing (ICCP), 2012 IEEE International Conference on , vol., no., pp.297-300, Aug. 30 2012-Sept. 1 2012
9. Ravindran, K., "Agent-based QoS negotiation in data-centric clouds," in Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on , vol., no., pp.331-334, 5-7 Oct. 2015
10. A. Maarouf, M. El Hamlaoui, A. Marzouk and A. Haqiq, "Combining Multi-agent systems and MDE approach for Monitoring SLA violations in the Cloud Computing," 2015.
11. F. Alsrheed, A. El Rhalibi, M.

*Multimedia Computing and Systems (ICMCS), 2014 International Conference*, 2014.

1. [40]



Negotiation of Service Level Agreements for Cloud-based Services

*IEEE International Conference on*

*Services Computing*, 2014.

1. -based Negotiation

in *Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2015 2nd IEEE International Conference*, 2015.

1. Y.-M. CHEN and H.-M. YEH, "AUTONOMOUS ADAPTIVE AGENTS FOR MARKET-BASED RESOURCE ALLOCATION OF CLOUD COMPUTING," in *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao*, 2010.
2. L. Xu, G. Tan, X. Zhang and J. Zhou, "A BDI Agent-Based Approach for Cloud Application Autonomic Management," in *IEEE 4th International Conference on Cloud Computing Technology and Science*, 2012.
3. J. O. Gutierrez-Garcia and K.-M. Sim, "Self-Organizing Agents for Service Composition in Cloud Computing," in *2nd IEEE International Conference on Cloud Computing Technology and Science*, 2010.
4. Y. Marathe and R. Ingle, "A FRAMEWORK FOR WEB SERVICES COMPOSITION," in Communication and Computing (ARTCom2012),Fourth International Conference on Advances in Recent Technologies in, 2012.
5. J.O. Gutierrez- -based Cloud Service

. 436-464, 2013

1. Y. M. Zaw, "Multi-Agent Based Cloud Service Composition using Contract Net Protocol for Information Retrieval Purpose," Journal of Engineering and Technology, vol. 5, no. 1, 2015.
2. Y. S. Chang, H.- itecture

*Journal of Supercomputing, Journal of Supercomputing*, 60(3), 2012: 338-359.

Agent-Based Cloud Computing: A Survey

2016 IEEE 4th International Conference on Future Internet of Things and Cloud

2016 IEEE 4th International Conference on Future Internet of Things and Cloud

# Mohamed Galal Hafez

Arab Academy for Science, Technology & Maritime Transport (AASTMT)

Alexandria, Egypt [mohamedgalalhafezrady@gmail.com](mailto:mohamedgalalhafezrady@gmail.com)

# Mohamed Shaheen Elgamel1

Arab Academy for Science, Technology & Maritime Transport (AASTMT)

Alexandria, Egypt [mohamed.shaheen@aast.edu](mailto:mohamed.shaheen@aast.edu)