Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Al-Qadisiyah
College of Computer Science and Information Technology

Cloud Resources Allocation Using Multi-Agent protocol based on user preferences

A Thesis
Submitted to the Council of the College of Computer Science and Information Technology at the University of Al-Qadisiyah in Partial Fulfilment of the Requirements for the Degree of Master in Computer Science.

By
Fouad Jowda Hussain

Supervised by
Dr. Muntasir Jaber Jawad Al-Asfoor

2021 **1443**

بسم الله الرحمن الرحيم

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ (١) خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ (٢) اقْرَأْ وَرَبُّكَ
الْأَكْرَمُ (٣) الَّذِي عَلَّمَ بِالْقَلَمِ (٤) عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ (٥)

صدق الله العظيم

سورة العلق الآية (١-٥)

Supervisor Certificate

*I certify that thesis entitled "***Cloud Resources Allocation Using Multi-Agent protocol based on user preferences***" is prepared and written under my supervision at the department of Computer Science / College of Computer Science and Information Technology / University of Al-Qadisiyah as a partial fulfilment of the requirements of the degree of Master in Computer Science.*

Signature:

Supervisor Name: Assist. Prof. Dr. Muntasir Jaber Jawad Al-Asfoor

Date: / 12 /2021

Head of Department Certificate

*In view of the available recommendations, I forward the thesis entitled "***Cloud Resources Allocation Using Multi-Agent protocol based on user preferences***" for debate by the examination committee.*

Signature:

Head Name: Dr. Qusay Omran Mosa

Head of the Department of Computer Science

Date: / 12 /2021

Certificate of the Examination Committee

*We, the undersigned, certify that (***Fouad Jowda Hussain***) candidate for the degree of Master in Computer Science, has presented this thesis entitled (***Cloud Resources Allocation Using Multi-Agent protocol based on user preferences***) for debate examination. The examination committee confirms that this thesis is accepted in form and content and displays a satisfactory knowledge in the field of study based on the candidate demonstration during the debate examination held on: ***25 / 11 / 2021***

Signature:
Name: Salam Al-augby
Title: Assist. Prof. Dr.
Date: / 12/2021
(Chairman)

Signature:
Name: Lamia AbedNoor Muhammed
Title: Assist. Prof. Dr.
Date: / 12/2021
(Member)

Signature:
Name: Ali Saeed Dayem Alfoudi
Title: Lecturer. Dr.
Date: / 12/2021
(Member)

Signature:
Name: Muntasir Jaber Jawad Al-Asfoor
Title: Assist. Prof. Dr.
Date: / 12/2021
(Supervisor and member)

Signature:
Name: Dhiah Eadan Al-shammary
Title: Assist. Prof. Dr.
Date: / 12/2021
(Dean of College of Computer Science and Information Technology)

List of publications

Publishing a conference :

F. Jowda and M. Al-Asfoor, “A Survey on Cloud Resources Allocation Using Multi-Agent System,” Expert Clouds Appl., pp. 139–151, 2022, doi: 10.1007/978-981-16-2126-0_13

Publish a local journal:

F.Jowda and M.Al-Asfoor , “ A MULTI-AGENT SYSTEM FRAMEWORK FOR CLOUD RESOURCES ALLOCATION, ” *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 13, no. 3 SE-Computer article, Sep. 2021, doi: 10.29304/jqcm.2021.13.3.847.

Dedication

First and foremost, I would like to dedicate this thesis to the people who sacrificed their lives for my beloved the Great Iraq to the martyrs of the army and the Popular Mobilization Forces

The light and meaning of my life ... my father and my mother.

My brothers, my sisters....

My lovely wife, and my loved children...

My friends who encouraged me.

And to everyone who helped & encouraged me.

Fouad J. Hussain

Acknowledgments

Always the greatest thanks to "Allah" the most glorious, merciful and compassionate.

I would like to express my gratitude to everyone who supported and encouraged me to complete this thesis.

First of all, I would like to express my sincere appreciation and deep gratitude to my supervisor Prof. Dr. Muntasir Jaber Jawad Al-Asfoor for his helpful suggestions, constant encouragement, patience, kindness and endless support throughout thesis and dissertation writing process.

I would like to thank the postgraduate teaching staff in the College of Computer Science and Information Technology for their help and support.

Finally, I would like to express my great appreciation to my family and friends for their constant encouragement.

Fouad J. Hussain

ABSTRACT

The cloud computing paradigm provides a shared pool of resources such as storage systems, CPU, and network bandwidth from one or more cloud providers for a limited of time at a variable or fixed price with different models delivered to the customers through the Internet via an on-demand dynamically-scalable form charged using a pay-per-use model.

The research problem faced by consumers is the exploitation of resources based on their needs at the lowest cost. In addition, the discovery of automated and intelligent cloud resources is a challenge that must also be addressed. Most of the time, the cloud user has trouble negotiating with the service provider because the negotiation interfaces are not available in most of the websites.

This thesis covers the stages of design, implementation, and evaluation of a Multi-agent-based resource allocation protocol for cloud computing, to achieve the objectives below:

1. Develop a framework for studying the performance of agent-based solutions for resource allocation challenges in a cloud computing setup.
2. Develop a resource management component that relies on agent techniques and artificial intelligence algorithms.

The implementation rate of the first model is 55% implementation rate for users' requirements. The second model involves negotiating with only one data center agent, where the rate of implementation of user requirements reached 67%. The third model includes a negotiation algorithm, in addition to the possibility of renegotiating with other data center agents in case the current negotiations fail to reach a solution and provide an offer commensurate with the user’s request. The rate of implementation of users’ requests reached 82%, which is the best proposed model.

The results confirm that the proposed protocol is efficient in implementing users' requirements because it includes automatic negotiation between agents based on the Contract Net Protocol (CNP) [1] protocol for interaction and coordination between agents. The results also confirm ensures customer satisfaction which consists in trying to use the service at the lowest cost.

[1] CNP is a protocol for sharing tasks in multi agent systems , introduced in (1980) by Reid G. Smith,It is used to distribute tasks between independent agents[81].

Table of contents

CHAPTER 3 - THE PROPOSED SYSTEM

CHAPTER 4 – EXPERIMENTS AND RESULTS EVALUATION

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Abbreviation	Description
ACC	Agent Communication Channel
ACL	Agent Communication Language
AI	Artificial Intelligence
AID	Agent-Identifier
Amazon EC2	Amazon Elastic Compute Cloud
Amazon S3	Amazon Simple Storage Service
AMS	Agent Management System
BA	Broker Agents
BDI	Beliefs, Desires, Intentions
CFP	Call For Proposal
CNP	Contract Net Protocol
CSP	Cloud Service Provider
CSV	Comma-Separated Values
CUA	Cloud Users Agent
DCA	Data-Center Agent
DFA	Directory Facilitator Agent
DoS	Denial of Service
FIPA	Foundation for Intelligent Physical Agents
GA	Global Agent
GAE	Google App Engine
GUI	Graphical User Interface
HA	Host Agent
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
IT	Information Technology
JADE	Java Agent Development Framework

JDK	Java Development Kit
KB	Knowledge Base
KVM	KERNEL-BASED VIRTUAL MACHINE
KQML	Knowledge Query and Manipulating Language
MA	Multi-Agent
MAS	Multi-Agent Systems
NIST	National Institute of Standards and Technology
PA	Platform Agent
PaaS	Platform as a Service
PM	Physical Machines
PS	Physical Server
QoS	Quality of Service
RMI	Remote method invocation
SaaS	Software as a Service
SLA	Service Level Agreement
TTL	Time-To-Live
UA	User Agents
UCA	User Cloudlet Agent
VM	Virtual Machine
VO	Virtual Organizations

Chapter 1 - Introduction

Chapter One
Introduction

1.1. Introduction

Cloud Computing is becoming a new paradigm for hosting and provision of resources via the Internet. In this model, the user pays the service providers based on their consumption. Similarly, when customers take out traditional public services such as water, electricity, and telephone, etc. This new model offers many advantages, such as rapid deployment, pay-as-you-go, cost reduction, ease of scaling, faster service delivery, network access, etc. Because of these different characteristics, Cloud Computing has become an interesting solution for businesses and researchers. It is also seen as a fully virtualized system, enabling compute, storage and use of software resources as well as servers as a single platform. Data management services are currently run in the user's local environment but are provided remotely by service providers the cloud [1].

Among the research areas that have paid particular attention to the problem of resource allocation are economics, operations research and informatics. Additionally, the resource allocation problem is relevant to a wide range of applications, such as e-commerce, supply chain[2], sensor networks[3], enterprise application integration. The task of resolving the resource allocation problem can be managed centrally or distributed. Centralized as in the case of combinatorial auctions where it is the auctioneer who issues the resource

[2] Supply-Chain defines as "A supply chain is an entire system of producing and delivering a product or service, from the very beginning stage of sourcing the raw materials to the final delivery of the product or service to end-users".[82]
[3] A sensor network is "a group of sensors where each sensor monitors data in a different location and sends that data to a central location for storage, viewing, and analysis" [83].

allocation decisions[2][3]. Either decentralized through automated trading, and it will become the role of market participants (agents) to contribute in resource allocation decisions. Automated trading is an area of research that involves artificial intelligence it has received significant attention in recent years [4][5] and its importance is widely recognized due to the fact that intelligent agents who negotiate with each other on behalf of human users are expected to produce more efficient negotiations [4][5] .

This chapter explains the general idea of using cloud computing, its importance and how to use it, as well as the importance of using resources in some scientific and research fields, and presents research-related work on the cloud computing resources using a multi-agent platform. It also presents the problem that was focused on in building the proposed system. In addition, it explains the purpose and contribution of the thesis. Finally, he presents the organization of the following chapters of the thesis.

1.2. Related work

Ralha et al.[6] this research presents MAS-Cloud, a multiagent system to dynamically monitor, predict and provide computational resources in cloud platforms. Deductive reasoning agents work cooperatively in a three-layer architecture to provide transparent horizontal elasticity of virtual machines in public cloud platforms (i.e., Google, Amazon EC2). MAS-Cloud was evaluated with a nondeterministic and CPU intensive application providing a challenging validation case. using inference rules for resource provisioning to automatically for the program operating on the cloud platform. Tested and validated using MASE-BDI factor-based simulation.

Alwadan [7] this proposed structure Cloud computing is designed to authorize access to large amount of computing resources. Multi-Agents System (MAS) technology introduces an ideal way for open and scalable systems that is varied dynamically. objective is to provide cloud computing solutions based on the design and development of software agents that can improve the use of cloud resources, also the system aims to monitor the jobs accomplishment and to provide the agent manger (broker) with the right feedbacks about the job process. the system is to track job output and provide the agent manager (broker) with the necessary input on the work process. The proposed architecture has been implemented and tested by the JADE (Java Agent Development Framework).

Mazrekaj et al.[8] suggest approach for Improving the performance and actual quality of the data center . It's based on the model utility function, based on the use of host CPUs to drive live migration activities. The decision to improve the utility function for Allocation of base resources provides A non-existent flexible resource sharing policy in rule-based policies and the threshold. HA is charged for continually monitoring the utilization of the host CPU and for determining if the host is in an overload or underload state, this information is moved on to GA who is it is also responsible for initiate acts for Local customization via Decide on the allocation of CPU capacity (CAP), virtual machines and Resolve disputes when the total of maximum values is greater than CPU capacity for all virtual machines. GA makes decisions on the global allocation of resources to improve VM placement to decrease violations of SLA and power usage.

W.Wang et al.[9] presented a resource allocation method based on multiagent (MA) to process the Cloud Service Providers (CSPs) problem of allocating appropriate VM resources to physical machines (PMs) to reduce energy consumption. By sending each PM a cooperative agent to assist the PM in resource management. There are two mechanisms 1) Auction-Based VM Allocation. 2) VM Consolidation based on negotiation. the local negotiation Mechanism of VM consolidation is designed for agents to exchange their dedicated virtual machines to save energy cost and address system dynamics.

F. De la Prieta & Corchado [10] this study proposes an architecture paradigm is called “+Cloud” (Multi-agent System Cloud) that based on intelligent agent “virtual organizations” (VOs). The major goal is to monitor and control a Cloud Computing framework, enabling it to respond to the needs at any given time automatically and dynamically. This study used the platform of cloud, published in BISITE Research Group's HPC environment that enables virtualization using the virtualization system Kernel-based Virtual Machine (KVM) and the Intel-VT technology.

Fareh et al [11] In this paper, we presented an agent-based approach for resource allocation in the cloud; the proposed approach is based on: a layered architecture to connect the cloud providers of different heterogeneous cloud with the cloud users. The multi-agent system is used to model the process of allocating of cloud resources. Several experiments show that autonomous agents make the clouds smarter in their interactions with users and more efficient in resources allocation. The only metric used in the validation phase is the cost of services or applications, while studying the performance of a system is usually measured by the speed of sending the response to a user. Thus, we will propose in future a new work to measure the elapsed time of each cloud user agent to meet the demand of the user. Moreover, we need to perform a study of the influence of directories of CUAs and BAs on the response time flow to satisfy a request of a user. used Simulations by CloudSim are carried out using the JADE platform.

Al-Ayyoub et al. [12] This study presented the dynamic resources provisioning and monitoring (DRPM) system, a multi-agent system to manage the cloud provider's resources while taking into account the customers' QoS requirements as determined by the SLA. The proposed DRPM system is evaluated using the CloudSim tool. The results show that the DRPM system allows the cloud provider to increases resource utilization and decreases power consumption while avoiding SLA violations.

Shyam & Manvi,[13] The proposed work fills the void of an efficient resource allocation scheme in cloud computing using user's cloudlet agent and provider's resource agent to (i) make sure that the number of VMs used is minimized, (ii) utilization of resources within the VM is maximized, (iii) the overall cost is minimized, and (iv) it adheres to the QoS guaranteed. The Best- Fit approach improves placement ratio, which brings benefits for users as well as providers. In addition, virtual machine migration caused by host overload is less likely to happen which improves the overall system performance. The possible improvements over the proposed techniques are: (1) implementing the proposed technique for dynamic resource allocation scenarios, and (2) embedding game theoretic approach among agents for resource bargaining in cloud computing environment.

Farahnakian, Liljeberg, et al.[14] In this paper, a three-level hierarchical architecture for VM management in a large-scale data center is proposed. This architecture uses multi agents to achieve the proposed objectives automatically. The objectives are to reduce the energy consumption and number of migrations in the data center while ensuring a high level of adherence to the SLA. Compared with the existing dynamic consolidation methods in CloudSim simulation, the proposed hierarchical VM consolidation model is able to reduce energy consumption, the number of migrations and SLA violations efficiently.

The table 1.1 below summarizes the process of integrating Cloud Computing with a multi-Agent system, as this process aims for several purposes as shown, in addition to the proposed systems, technologies and algorithms.

Table 1.1 Summary of studies

Year	Author	Aim	The Agents	parameters of evaluation
2019	Ralha et al [6]	dynamic monitoring, to allocate and distribute of computing resources, Reducing the time and cost of expense cloud services	VM Manager, Manager Agent, Monitoring Agent.	Quality of Service (QoS)
2018	Alwadan [7]	Improve the use of cloud resources, monitor the jobs accomplishment and to provide the agent manger (broker) with the right feedbacks about the job	Clients (Human Actor) , Agent Manager (broker), providers agent	Service Level Agreement (SLA)
2017	Mazrekaj et al [8]	Improving the Performance and actual quality of the data center	(Host Agent) local agent and (Global Agent) central controller	*VM SLA Violation, Energy Consumption, Number of VM Migrations, Energy and VM SLA Violations*
2016	W. Wang et al [9]	reduce energy consumption	agents (Swap contract only two agents and Cluster contract (> 2))	local VM migration
2016	De la Prieta & Corchado [10]	monitor and control a Cloud Computing framework, enabling it to respond to the needs at any given time automatically and dynamically.	local monitor, local manager, Global Manager every physical server, Service Monitor, Service Supervisor, SLA Broker, hardware supervisor, global supervisor and identity manager	Quality of Service (QoS)
2016	Fareh et al [11]	make the clouds smarter in their interactions with users and more efficient in resources allocation	agents (Cloud Users, broker, Coordinators, datacenter management, Cloud Providers)	cost of services or applications
2016	Al-Ayyoub et al [12]	increases the utilization of resources and reduces power usage while	agent (global utility, group a local utility)	QoS requirements as determined by the SLA

		preventing violations of SLA.		
2015	Shyam & Manvi [13]	work better in terms of allocation of VMs, time for job execution, expense, and usage of resources.	agents (Cloudlet, resource)	Quality of Service (QoS)
2014	Farahnakian, Liljeberg, et al [14]	minimize energy usage, effectively migrations and SLA violations	agents (cluster, Global, and local every PM)	*SLA Violations, Energy Consumption, Number of VM Migrations, Energy and SLA Violations*

1.3. Problem Statement

- One of the most common challenges that consumers are facing is choosing resources based on their needs at the lowest cost. Accordingly, it might be confusing for cloud customers who wish to utilize cloud services, and takes a long time to discover and provision Cloud resource. With an expanding market for Cloud resources as well as the increase in services providers, an automated, intelligent Cloud resources discovery is a challenge that needs to be addressed.
- Most of the time, the cloud user has the problem of contacting the service provider to negotiate via websites for the best offer because negotiation interfaces are not available in most websites.

1.4. Thesis Objectives

The aim of this study is to simplify the search for online service providers in cloud sites. It provides a platform where cloud users can enter criteria and receive best-bid to reduce workload for technology deployment. It also the process of allocating Cloud Computing resources easily to the consumer through the use of Multi Agent System (MAS) technology with cloud computing. The resulting system is efficient and negotiations can be conducted of price. The following are the research objectives that have been established to create the proposed platform:

1. Developing a framework to study the performance of agent-based solutions to resource allocation challenges in a Cloud Computing environment.
2. developing a resource management component that relies on agent techniques and artificial intelligence algorithms.

1.5. Thesis Contribution

A Multi-agent-based Cloud Computing system has been created, deployed, and used that can discover cloud providers, identify providers that match customer specifications, choose the best offers for cloud users, and conduct negotiations. The work presented in this thesis can serve the cloud computing markets by outlining the strategy that future companies may use to create their own cloud computing markets.

This thesis adds value to research by meeting the goals of the study as follows:

1. Maximize the resources of cloud providers by helping to meet the needs of cloud customers.
2. It helps cloud computing users to find the best offers from cloud service providers, which makes it possible to reduce the cost of resources.
3. By providing negotiating, it will enable users to renegotiate prices with service providers and get the services they need at the agreed price.

1.6. Thesis Organization

This thesis consists of five chapters.

Chapter 1: includes a general introduction to Cloud Computing and the use of Multi-Agent Systems (MAS), discusses the study conducted in both areas with the work carried out on the convergence of Cloud Computing and the multi-agent system, Problem Statement, identifies Thesis Objective, and finally contributions.

Chapter 2: provides an introduction of Cloud Computing, including definition, characteristics, types of cloud services, deployment models, and cloud concepts related to resource allocation. In addition, this chapter, provides an introduction to agent and multi-agent systems, the concept, definitions, types, Structure, Agent and object… etc.

Chapter 3: provides a full description of the system, including illustrations of the proposed model in addition to the algorithms used, as well as explains the simulation environment, and finally explains the proposed system settings for the system, which includes detailed information about each component in the simulation for the purpose of testing the system.

Chapter 4: the experimental findings derived from the suggested system implementation are presented.

Chapter 5: It presents the most important conclusions from this thesis as well as some recommendations for future work.

Chapter 2 – Theoretical Background

Chapter two

Theoretical Background

2.1 Introduction

This chapter is divided into two parts: The first part gives a detailed overview of the Cloud Computing approach. definitions of this concept and its essential characteristics. Then, how the cloud offers a wide choice of on-demand IT services and offers pay-as-you-go billing to users according to their needs. These services are presented in the form of software, platform, or infrastructure and are deployed in four models, Private Cloud, Public Cloud, Community Cloud, and Hybrid Cloud. With the advantages offered by the cloud, end users find that this technology is a good choice for using online services. Despite all these cloud-produced solutions, there are always limits. The most significant challenges that computing must face, including cloud technology, are to improve the quality of services provided to users.

The second part provides general details of the agent, including definitions and properties, as well as types of agents and the difference between the agent and the object, and then it moves to the topic of the multi-agent system, where it presents a general concept of it and the definitions of a multi-agent system and as well the agent environment and finally methods interaction in multi-agent systems.

2.2 Definition of cloud computing

The cloud is a Large virtual resource pool that is readily accessible and usable [15][16]. Depending on the principle of ”pay-as-you-go” [17]. The increasing importance of this model has contributed to the provision of a wide range of definitions [18][19] [20][21]. The most widely agreed term, a most valid from both a technical and a practical viewpoint, is provided by the National

Institute of Standards and Technology "NIST" [18]. In this definition, Mell et al Grance propose “that Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The cloud model is composed of five essential characteristics, three service models, and four deployment models”.

Vaquero et al.[22] defined cloud as “Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs”.

Plummer et al., [23] defined cloud as "a style of computing where massively scalable IT-related capabilities are provided as a service across the Internet to multiple external customers".

Marston et al. [24] definition of Cloud Computing is as : “It is an information technology service model where computing services (both hardware and software) are delivered on-demand to customers over a network in a self-service fashion, independent of device and location. The resources required to provide the requisite quality-of service levels are shared, dynamically scalable, rapidly provisioned, virtualized and released with minimal service provider interaction. Users pay for the service as an operating expense without incurring any significant initial capital expenditure, with the cloud services employing a metering system that divides the computing resource in appropriate blocks”.

2.3 Cloud Computing Characteristics

According to previous definitions, there are basic characteristics of Cloud Computing that have been expanded by researchers [18][25] [26]:

2.3.1 Free access on request

Cloud Computing allows customers to consume IT resources such as services (servers, storage, platform development, etc.), through networks, according to their needs, simply and flexibly, without the need for human interaction with provider services. Software, hardware, and cloud data can be automatically reconfigured, organized, and combined into a single image provided to the user [27].

2.3.2 Measured service

The Cloud provides transparency between the service provider and the service consumer by providing cloud customers with tools to control and monitor how their resources are used.

2.3.3 Shared resources

Cloud Computing is the new business model-based computing paradigm in which services and resources are aggregated and made available to consumers in a multi-tenant model. Usually, consumers have no knowledge or control over exactly where the supplied resources are stored [18].

2.3.4 Fast elasticity

The resources can be quickly increased or decreased depending on needs. In addition, these resources should be freed for other purposes. when they aren't needed anymore.

2.3.5 service accessible via a network

The services offered to users by the Cloud must be available on the network and accessible over standard mechanisms promoting the use of platforms heterogeneous, e.g. laptops, workstations, tablets, mobile phones [18].

2.3.6 Pay-as-you-go

Any Cloud offer includes a payment that is made each time you use a service. The provider can accurately count the consumption (in quantity and duration) of the various services (storage, CPU, bandwidth, etc.). This will allow him to bill the user according to his actual consumption. This billing method allows, first of all, to save commissioning costs, investment costs, and operating costs for companies [28].

2.3.7 Based Virtualization

A Cloud Computing system is a completely virtual system. Virtualization means the abstraction of the details of the virtual resources for high-level applications, as well as physical hardware. It supports Cloud Computing, because it offers the possibility of pooling computer resources from server clusters, and thus dynamically allocating virtual machines to applications on-demand [29].

2.3.8 Based Service Level agreement (SLA)

The SLA defines a contract between a service provider and a customer, such as delivery parameters, maintainability, availability levels, operations, performance, or other attributes of the service, such as billing, and even penalties in the event of a breach of contract [30]. With cloud services, a customer can negotiate the level of service he needs and he has to pay for it according to QoS guarantees [31][32].

2.3.9 Simplicity, flexibility, reliability, and fault tolerance

Cloud environments must guarantee a quality of service for users, such as flexibility, simplicity, reliability, and error tolerance. The allocation and use of cloud resources should be simple. These services are also intended to support heavy tasks such as small workloads in environments that benefit from the incorporated redundancy of the numerous servers that make it possible to have high availability and reliability [32][33].

2.3.10 Effective security

Safety, in all systems, plays a very important role, especially if sensitive data resides in the cloud. The loss or illegal access to data can lead to misunderstood effects, especially for data. For this reason, researchers and cloud providers consider this point by introducing the architecture of highly effective security, encryption, and authentication policies [34].

2.4 Types of cloud services

Cloud Computing allows users or companies to consume on-demand IT services. These services can be presented in several forms depending on the type of service that corresponds to the level of management responsibility layers of the standard IT environment, whether by users or by providers [18]. In the Cloud environment, lets us distinguish the type of service. According to NIST[18] and as illustrated in Figure 2.1 .

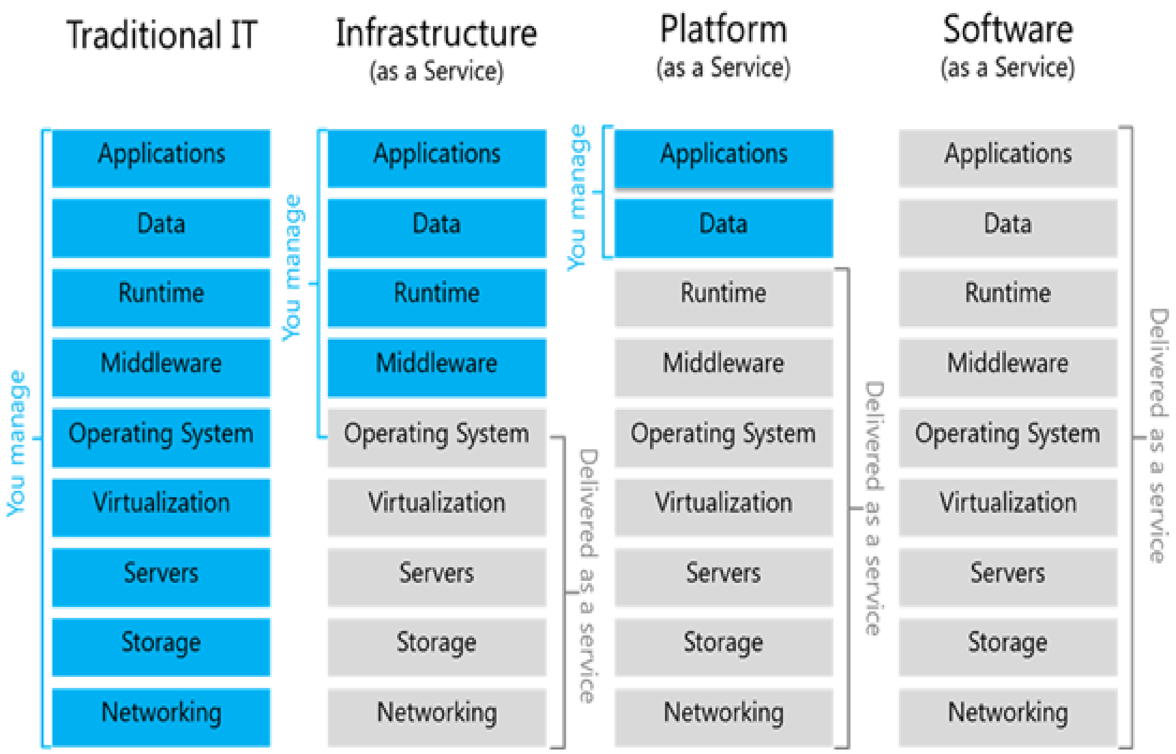

Figure 2.1: Types of Cloud Computing Service

2.4.1 Infrastructure as a Service (IaaS)

provides virtualized resources accessible through the Internet. the sense that these resources are an aggregation of several resources provided by several data centers which can be geographically distributed. As a result of virtualization technology, the user will have the impression that he owns a block of resources, despite the fact that these resources may be an aggregation of geographically distributed resources. Examples of this service are: "Amazon Elastic Compute Cloud" (Amazon EC2) and "Amazon Simple Storage Service" (Amazon S3) [18] [35].

2.4.2 Platform as a Service (PaaS)

It is a cloud model that fits fundamentally into development environments that provide platforms for the execution, deployment, and development of applications. In this model, the service provider provides everything that is required so that the user can develop their application. The user can control the software as well as the application development environment, and everything related to the underlying infrastructure is handled by the vendor. Typical examples of PaaS are Google App Engine (GAE), Elastic Beanstalk, Windows Azure, etc. [36][37] .

2.4.3 Software as a Service (SaaS)

SaaS applications are hosted on cloud servers and accessed through the Internet. Rather than requiring users to download and install a program on their device, Thanks to this model, the user does not have to worry about all the essential elements for the execution of the requested application: underlying infrastructure, operating system, etc. For example, mail servers (Gmail, Yahoo Email, etc.), Dropbox, Google Apps, Office 365, Facebook, Twitter, Google Documents, etc. at their datacenter [38][39].

Table 2.1: Comparison of cloud services [40]**.**

Service	Advantage	disadvantage
SaaS	-no installation -no more license -migration	-limited software -security -dependence on providers
PaaS	-no infrastructure required -no installation -heterogeneous environment	-limitation of languages -no customization in the configuration virtual machines
IaaS	-administration -personalization -flexibility of use	- security -need a system administrator

2.5 Cloud Computing Deployment Models

Cloud Computing models are differentiated by the use of resources. Resources may or may not be shared with the user or with a provider, with companies, or other types of users. NIST definition provides four deployment architectures describing approaches for providing consumer cloud services [18].

2.5.1 Private cloud

The private deployment model is intended for private companies that put all of the resources available exclusively and host them in these companies. This model consists of a collection of proprietary networks, frequently data centers located within the company, that support the control and management of these cloud resources. The main reasons for choosing this model are integration issues, data security concerns, and critical applications. whereas the provider is responsible for in a public cloud [41].

2.5.2 Public cloud

The public cloud represents the traditional cloud used by the majority of customers on the Internet. In this model, the consumer and the service provider are organizations that are different and resources are dynamically self-provisioned in an environment that is multi-tenant via applications or web services. access control and security of resources are ensured fully by the provider, which limits the freedom of customers in the operation of control and configuration [42][43].

2.5.3 Community cloud

The community cloud is used to share infrastructure among several independent organizations with similar interests. This community organization can also share the management tasks of these infrastructures, such as data security, application deployment, authentication, etc. [44].

2.5.4 Hybrid cloud

A Hybrid Cloud, as the name suggests, is formed when an organization develops a private cloud and wishes to operate public or community clouds in conjunction with its cloud for a specific purpose. companies can use the public cloud for less sensitive applications and the private cloud for critical and sensitive data and applications. So, hybrid clouds allow us to assemble the advantages of the other models [45]. Currently, the majority of cloud providers, like HP, VMware, and Amazon, provide hybrid cloud services.

2.6 Concepts related to resource allocation

Attributes to the customer a portion of the exploitable resources, including processors, storage space, bandwidth, and memory, and the pooled resources among all customers represent value point of the provider thus, the design and implementation of allocation policies in the cloud depends on the vendor's business strategy [46] .

2.6.1 Pooling

It is the practice of sharing the use of a set of resources by consumers (or any entities) having no link between them. The resources can be of various types, such as software or hardware. This practice is based on companies' desire to outsource their IT services to cloud infrastructures [46] .

2.6.2 Assignment and billing upon request

The concept of "pay-as-you-go" allows the consumer to pay only for what they use. So, the price is calculated using ratios like processors per hour, GB of disks per month, etc. This concept enables the cost of the use of computing services to be greatly reduced [47].

2.6.3 Scalability and elasticity

The concepts of scalability and elasticity are the match of the concept described above as "pay as you go". They give the user the impression of having permanent, unlimited resources. These resources can be easily and quickly added or withdrawn, sometimes automatically, to respond to user needs [47].

2.6.4 Virtualization

The concept of virtualization allows you to run one or maybe more logical machines on the same physical machine. Virtualization also makes it possible to reduce the waste of resources. A Cloud Computing system is a completely virtual system. There are different forms of virtualization which are: systems virtualization operating, storage virtualization, database virtualization, virtualization applications and hardware virtualization. Virtualization is the support of the Cloud Computing, because it offers the possibility of pooling computer resources to from server clusters [48].

2.7 Definition of an agent

The definition of a agent has aroused the interest of many researchers in many different disciplines, autonomic programs or agents can be visualized as a system that operates and/or reacts independently to external stimuli ,perceived from sensing itself or/and its environment [49] . There are several definitions of agent in the literature. They all look the same, but differ based on the application the agent is built on.

Yves Demazeau [50] defines agent as , “an agent is a real or virtual entity whose behavior is autonomous, evolving in an environment that it is capable of perceive and on which he is able to act and interact with other agents ”.

Jennings et al. [51] proposed the following definition: “An agent is a computer system, located in an environment, and which acts in an autonomous and flexible to achieve the objectives for which it was designed. "

Dayong Ye et al . [52] defined the agent as ‘‘an encapsulated computational system that is situated in some environment and this is capable of flexible, autonomous action in that environment in order to meet its design objective.’’

2.8 Types of agents

Architecture remains linked to the point of view of the designer, to the way of assembling the different parts of the agent so that the latter is capable of completing the assignments that have been entrusted to him. Agents are divided into categories based on their architectures and capabilities: cognitive, reactive, BDI, and mixed. Figure 2.2 illustrates the simplified architecture of an agent in terms of components and functions.

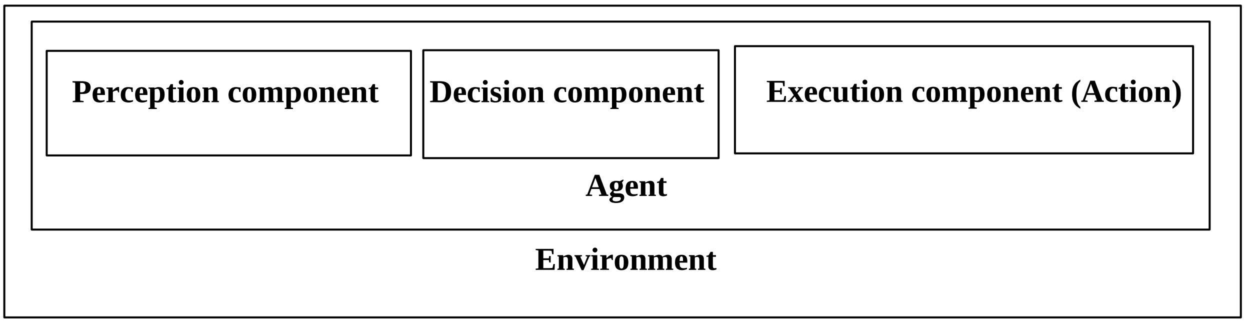

Figure 2.2: operation of an agent [50]

2.8.1 Cognitive agents

Cognitive agents, also referred to as "artificially intelligent agents", have cognitive abilities derived from the human model. They have an explicit representation of the environment in which they live and of other agents, and, most importantly, they represent themselves [53]. Cognitive agents are distinguished by the presence of a decision component, i.e., rationally developed logic. They, therefore, have a knowledge base containing different facts about their experience and management relationships with other agents and their environment. In addition, this interaction enables agents to communicate, collaborate, and take action. Therefore, they can make choices based on several factors, the information they have and plan their actions. They usually have explicit plans in place to help them achieve their objectives. They are structured in a society where there is real social organization. In this case, they can cooperate by coordinating their activities and can sometimes negotiate to resolve disputes.

2.8.2 Reactive agents

Agents with reactive capacities only have a protocol and a language of communication. They do not have an explicit representation of their environment and cannot make a decision based on their past actions. They can only respond to the law of motive/action. Indeed, as soon as they detect a change in their environment, they perform a pre-programmed action. They are always on the lookout for changes in their environment. Their quick and unthinking actions are similar to reflexes. Reactive agents are thus more interesting at the community level than at the individual level, and their members' ability to adapt and evolve emerges from their interactions [54]. These agents have limited reasoning skills, but their interactions allow the emergence of collective intelligence.

2.8.3 BDI agents (Beliefs, Desires, Intentions)

This agent belongs to a broader category of cognitive agents. Indeed, within the context of practical logic, which is thought based on rational states, the researchers developed the BDI architecture [55][56], an architecture built around practical reasoning. These agents are generally represented by a "mental state" having the following mental attitudes:

- **Beliefs:** An agent's perception of their environment.
- **Desires:** The various states to which the agent may wish to commit.
- **Intentions:** desires that the agent has committed to achieving.

2.8.4 Hybrid or mixed agents

From the beginning of the 1990s, it was realized that reactive systems and cognitive systems could not be suitable for solving all problems. Each of these two types of systems are suitable for certain types of problems. From then on, the researchers tried to combine the two ways to achieve hybrid architecture [57] [58] [59]. Hybrid agents combine the responsiveness of reactive agents as well as the reasoning of cognitive agents.

2.9 Structure of an agent

Ferber [53] proposes a general structure that an agent have, such as: Know-how, beliefs, control, expertise, and communication.

- **Know-how:** It is an interface that allows the declaration of knowledge and the skills of the agent. It also allows the selection of agents to be solicited for a given task.
- **Beliefs:** The agent's knowledge of the world (the representation of his environment, other agents, and itself).
- **Control:** Knowledge of control over an agent is represented by the goals, intentions, plans, and tasks it has.
- **Expertise:** This is the ability to solve problems. Uses both facts and heuristics to solve complex decision-making problems for example, The Expert System in AI can resolve many issues which generally would require a human expert. It is based on knowledge acquired from an expert.
- **Communication:** So that two people can communicate, they must speak the same language. In the case of agents, the same principle is essential, which is why the creation of a common language for all agents was needed to ensure good communication and coordination of actions.

2.10 Agent and object

Depending of the issue to be solved and processed, we can determine the approach to using agents or objects. These two concepts, despite intersecting at several points, are shared at the structure level (having an internal state, a set of modifications to this state, and a communication capacity) [60]. Several other points differed from the execution mechanism. Objects have no purpose or research satisfaction and the communication mechanism comes down to simple call methods. The main difference between the object and the agent comes down to the autonomy of the latter[60].

Indeed, A collection of services defines an entity (its methods) that cannot refuse any execution request if another object requests it. That the object does not control its behavior on the contrary, agents can receive messages that are not only requests for execution but information, plans, etc.. [61] Likewise, agents try to meet objectives that require more autonomy when interacting with others. Indeed, an agent can accept work requested by another, just as he can refuse it if he is too busy or does not know how to do it, or negotiate its execution if it is not in his interests. Another important distinction is the flexible behavior (responsiveness, positive activity, and sociability) of the agent who is completely absent from the object [62].

2.11 Definitions of Multi Agent System (MAS)

Each study uses its own definition, dependent on the specialty of the researcher. The objective is not to collect several definitions instead, to find a generic and accepted definition. There are several definitions of multi-agent systems in the literature, and these definitions should be examined to highlight the varying features of MAS and their essential properties.

A multi-agent system is one that consists of a number of agents, which interact with one another, typically by exchanging messages through some computer network infrastructure. In the most general case, the agents in a multi-agent system will be representing or acting on behalf of users or owners with very different goals and motivations. In order to successfully interact, these agents will thus require the ability to cooperate, coordinate, and negotiate with each other, in much the same way that we cooperate, coordinate, and negotiate with other people in our everyday lives [63].

2.12 Environments Classification

The environment refers to the agent's immediate surroundings. The agent collects data from the environment through sensors and outputs it via actuators. Numerous environments exist.

- **Fully vs Partially Observable:** When the agent's sensor can record or monitor the complete state of the environment at each point in time, it is said to be a fully observable environment. Conversely, if the agent's sensor can only record some information, it is said to be a partially observable environment [64].
- **Deterministic vs Stochastic:** Deterministic environments include elements in the agent's present state that completely define the agent's future state. The stochastic environment is, by definition, unpredictable, and incapable of being fully defined by the agent [64][63].
- **Competitive vs Collaborative:** When an agent competes with another agent to maximize the output, it is said to be in a competitive environment. When many agents collaborate to create the intended result, an agent is said to be in a collaborative environment [65].
- **Single-agent vs multi-agent:** A single-agent environment has just one agent. A multi-agent environment is one in which more than one agent exists[64][66].
- **Dynamic vs Static:** A dynamic environment is described as an environment in which the environment is constantly changing from one state to another. A static environment is the opposite of a dynamic environment, where nothing changes in the environment, it remains permanently fixed [64] [63].
- **Discrete vs Continuous:** A discrete environment is defined as one in which there is a limited number of percepts and actions that can be performed in the environment. This means that the environment in which the activities are done cannot be counted, and thus is not discrete, and, as a result, is referred to as continuous [64][63].

- **episodic vs Sequential**: Sequential environments in this type of environment, an agent needs memory of previous actions is required in order to determine the next best action. As for the Episodic environment, it is a series of one-time actions that are not stored, and only the current percept is required for action [64][66] .

2.13 Interaction in MAS

A dynamic association of two or more agents with a series of mutual behaviors is referred to as an interaction. It is through this interaction that the MAS is seen as a whole and not as a set of independent entities. For an agent, interacting with another is both the source of his strength and the source of his problems [53]. Interacting allows an agent to share information and services to meet its objectives while avoiding conflicts. An interaction, if it is started, must proceed correctly and also terminate correctly. It is for this reason that the interactions are structured according to typical diagrams called protocols.

Interaction protocols allow agents to exchange structured messages and control the exchange of these messages and thus facilitate their coordination. An interaction protocol specifies rules that must be respected by the agents during a conversation, and thus defines for each step the types of messages that can be sent.

There are generally five different type for interactions between agents: communication, cooperation, coordination, organization, and negotiation.

2.13.1 Communication

One of the most important aspects of the MAS is communication. It is a essential part of all interactions. It allows information sharing between agents. Agents can share knowledge and organize their tasks through conversations. The various interaction protocols are also implemented through communication. the methods of communication and correspondence between agents are direct or indirect.

2.13.1.1 Indirect communication

Agents do not communicate with each other directly through this type of correspondence, but the communication via Blackboard is that contacts share information using a shared memory that agents can access, each agent has access to information stored in shared memory. When agents interact with the environment, they leave tags or signals that other agents can detect. in this kind communicate there is no specific recipient.

As a result, they can, write messages, insert partial results of their calculations and get information. The blackboard is usually divided into several application-specific levels [59].

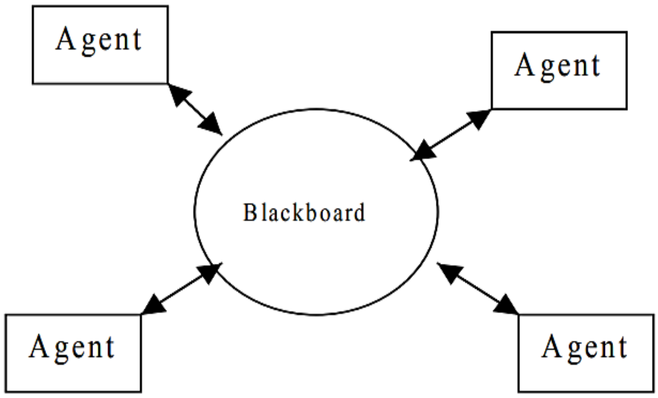

Figure 2.3: Indirect communication [67]

2.13.1.2 Direct communication

In this type of inter-agent communication, messages are oriented directly to the specified agent and are not shared with another agent as in the previous type.

It is specific to cognitive agents, because, the latter relatively to reactive agents, have knowledge about themselves and others.

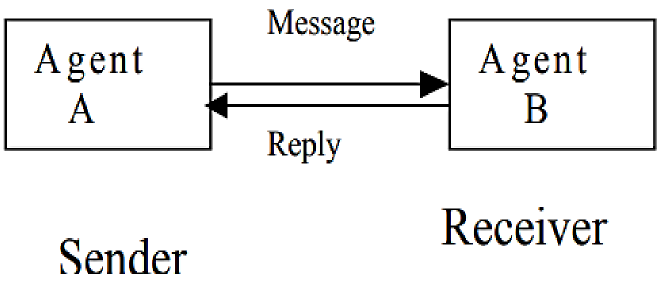

Figure 2.4: Direct communication [67]

Direct communication is based on three essential elements:

- **The communication language:** It allow to structure the messages that agents exchange [68]. The most widely used and most popular communication languages are "Knowledge Query and Manipulating Language" (KQML) [69] and the Agent Communication Language of "Foundation for Intelligent Physical Agents" (FIPA) (FIPA-ACL) [70][71]. They are based on the theory of language acts [72].
- **Ontology:** Is used to provide vocabulary and terms understandable by all agents. This semantics will be governed by rules and constraints that will make it possible to define a consensus on the meaning of the terms contained in the messages.
- **Communication mechanisms:** They make it possible to store, search, and address messages to agents [73] [74].

2.13.2 Cooperation

Cooperation is the general form of interaction. It is necessary when the agent cannot achieve his objectives without the help of other agents. It represents the typical interaction that relates to the manner of distributing work and, accordingly, the distribution of tasks among several agents. Cooperation can be static, so tasks are distributed through the design of a multi-agent system. It can be dynamic. In the latter case, tasks are distributed through the mechanisms of supply and demand. It may be carried out by a coordinating agent who centralizes offers and requests in a distributed manner.

Ferber in [53] defined cooperation as “We will say that several agents cooperate, or that they are in a cooperative situation, if one of these two conditions is verified: 1) Adding a new agent enables the group's performance to be increased differently. 2)Agents' action is to resolve or avoid current or potential or existing disputes." He then added, “It’s because they cooperate that agents can accomplish more. than the sum of their actions, but it is also because of their multitude that they must coordinate their actions and resolve conflicts”.

2.13.3 Coordination

When agents use common resources or solve problems that are not completely independent but related and complementary, system agents must perform in addition to their individual problem-solving tasks, additional tasks (called coordination tasks) that improve system operation

Jacques Ferber .[53] gives the following definition: "the coordination of actions in a Multi-Agent system is defined as all the tasks performed by the agents to carry out the other actions (effective actions) under the best conditions" .

The coordination between the agents of a system appears in two distinct forms, it serves on the one hand to avoid problems, and to improve the functioning of the system on the other hand, coordination is necessary to improve and maintain consistency in functioning of the system. The coordination tasks are not directly related to problem-solving, but allow the multi-agent system to operate in an efficient manner. This allows the system to solve the problem collectively, save time, avoid conflicts between agents, and reduce as much as possible interactions between agents, which increases system performance.

2.13.4 Organization

A MAS consists of a group of agents that operate in an environment in which they perceive and in which they act. Agents are engaged in a collective activity that requires them to interact and collaborate. This situation poses a problem for social organizations. Also, in a society, the word organization consists both of the action of structuring and of the result of this action, which is the model or the static structure.

Dignum, V et al .[75] An agent organization be defined as "a social entity composed of a specific number of members (agents) that accomplish several distinct tasks or functions and that are structured following some specific topology and communication interrelationships in order to achieve the main aim of the organization. Thus, agent organizations assume the existence of global common goals, outside the objectives of any individual agent, and they exist independently of agents".

2.13.5 Negotiation

Defined as any communication process that ends in a mutually accepted agreement, it is obvious that the CNP constitutes negotiation. As with the sealed-price second bid auction, the sealed-price second bid auction is a kind of negotiation that requires even less communication than the contract net. One of the responsibilities of a protocol designer is to develop a protocol that achieves the intended outcome while using the least amount of communication possible.

In the absence of negotiation, other solutions are possible but less efficient than negotiation, for example, the decision is made in an authoritarian fashion by an agent who has the authority to decide. The voting method can also be used to resolve conflict issues.

2.14 Summary

This chapter provides a brief description as an introduction to cloud computing and also introduces the definition, characteristics, and types of cloud services, and service deployment models. Additionally, it provides a proxy definition and details about the types of agents. It also explains the agent structure and the main differences between agent and object and provides the definition of MAS. clarification the environments in agent systems, explains the characteristics of the MAS, and finally provides details of the interaction methods in the MAS.

Chapter 3 - The proposed System

Chapter three

The proposed System

3.1 Introduction

This chapter provides a detailed description of the proposed design of a Cloud Computing system that includes the allocation of Cloud Computing resources using a multi-agent system. The proposed system applies to all types of cloud deployment, it also details the algorithms used, describes the environment used, and the environment settings for the purpose of testing.

3.2 The proposed System

System design is an important stage that requires consideration to achieve compatibility between the two systems, and it is one of the most important things that must be achieved in order for the system to function properly without errors.

The proposed system is a protocol that depends mainly on the Contract Net Protocol **(**CNP) in the process of communication between agents Communication is carried out directly by exchanging messages with each other through the Agent Communication Language (ACL) Foundation of Intelligent Physical Agents (FIPA). The environment of agents in this proposed protocol is (partially observable, deterministic, competitive, multifactorial, dynamic, discrete, and episodic) and the type of agents is an interactive agent.

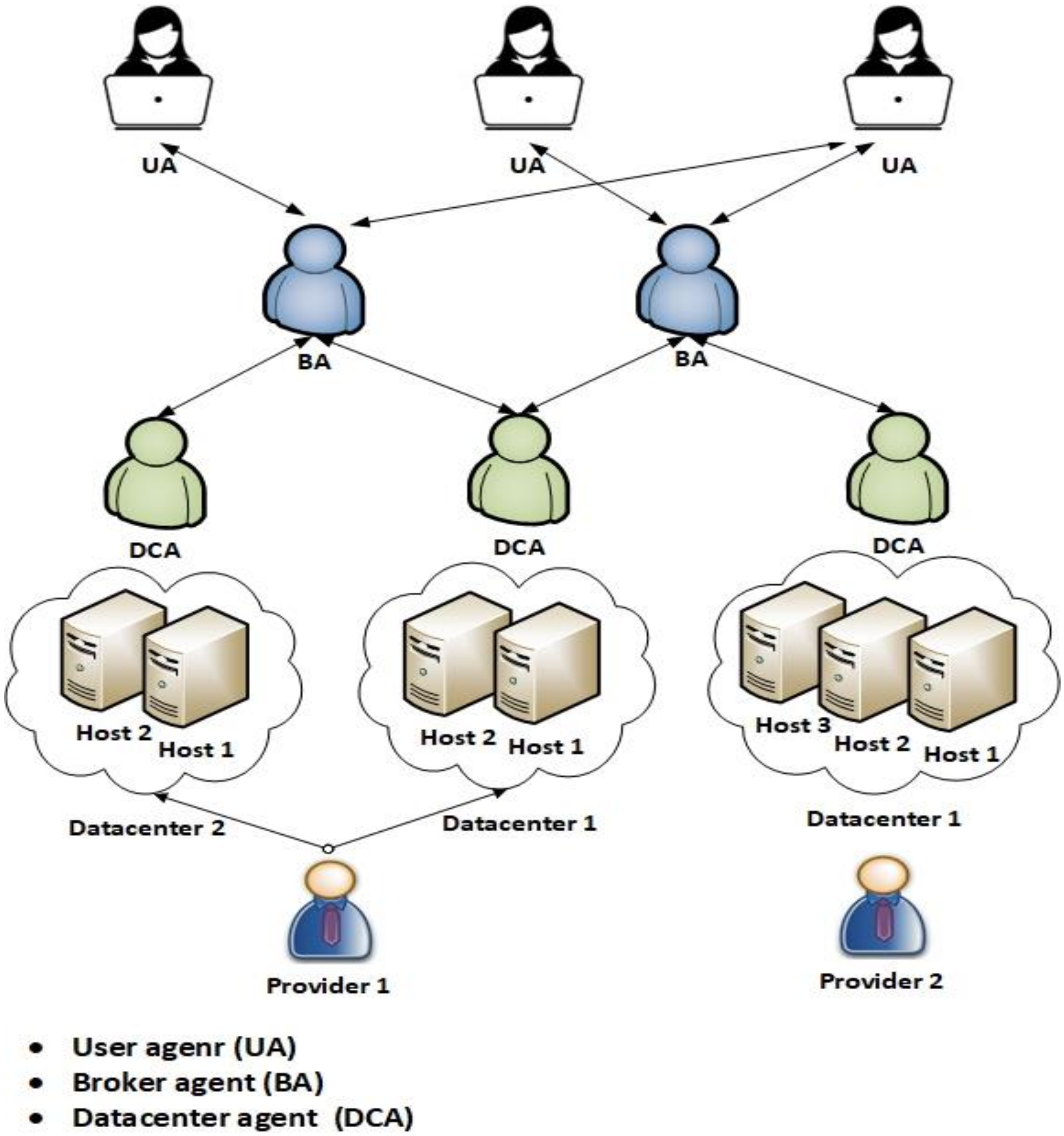

Figure 3.1: The proposed Cloud Computing systems architecture

The first model, which is shown in Figure 3.2 is the basic model, based on CNP. The protocol is implemented between the Broker agent and the data center agents. Sends a request from the user agent to the broker agent to allocate user resources (UR), the broker agent sends a call for proposals (CFP) to the data center agents and waits for a response, the agents start bidding, the latter submits offers as proposals. The Broker agent selects the best offers and then sends them to the

User agent to evaluate the offered offer by either accepting the offer if it is suitable for the consumer or rejecting the offer.

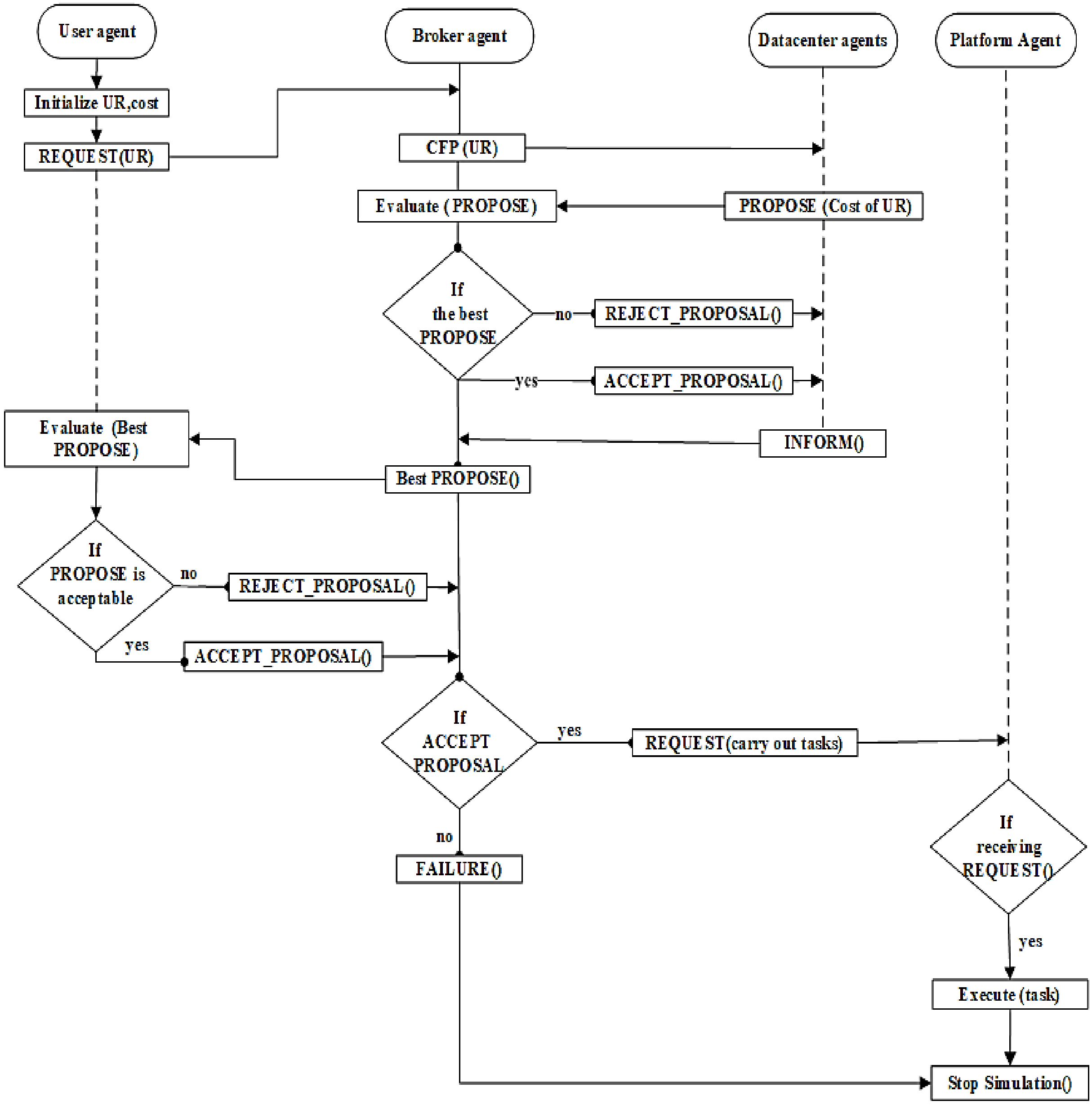

Figure 3.2 The first model sequence diagram

The second model, is somewhat similar to the previous model, except that there is a negotiation algorithm between agents, as shown in Figure 3.3, the negotiation process between the Data center agent and the user agent by the Broker agent to reach an agreement. After the user agent sends a request to the broker agent to allocate resources, the latter sends (CFP) to the data center agents, receives offers from them after a specified period and evaluates the offers, chooses the best among them and sends them to the user agent, who also evaluates the offer if accepted, the Allocate these resources at the agreed price. In the event of a refusal, the Broker agent asks the data center agent to initiate the negotiation process. If he agrees to negotiate, the offer is lowered Negotiations may continue more than once or the negotiation is refused and the process ends in failure.

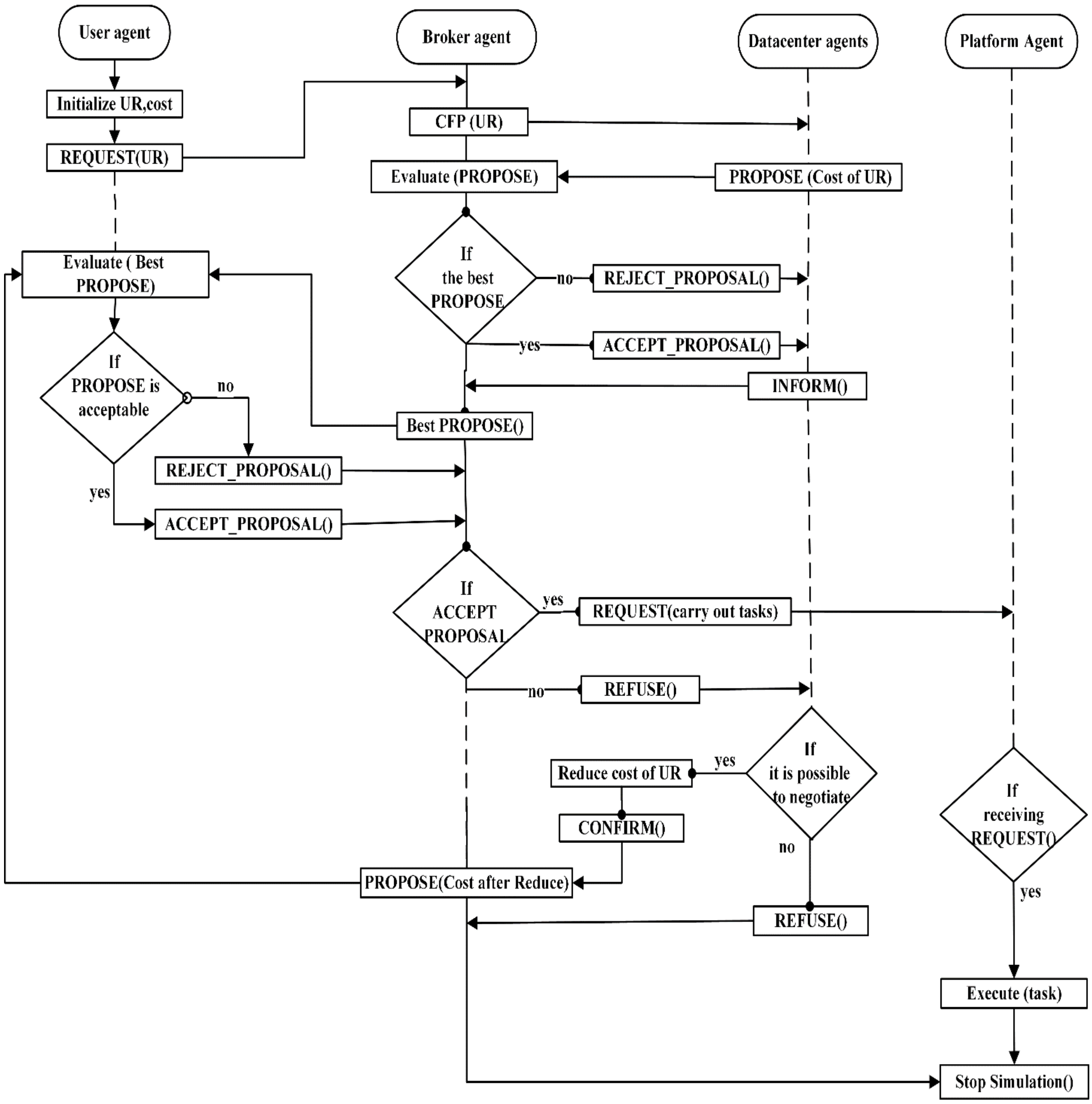

Figure 3.3 The second model sequence diagram

The third model is like to the second model, but the difference is that this model has greater negotiating capabilities than the previous model, and the ability to re-bid from data center agents in the event of failure of the negotiation with the data center agent. The same protocol used in the previous two models is used for communication between agents and bidding.

The whole organization of agents is visualized using the integrated interaction diagram, which displays the organizational structure as it is used in the problem domain. Figure 3.4 shows the agents and how they interact.

The broker is the heart behind the cloud infrastructure. It can communicate with agents. The broker acts as an agent manager in and of itself, coordinating communications between agents, monitoring running tasks, and keeping agents informed about the operation. In this model, the broker has moved from preserving the specifications of the cloud service providers to become an agent for processing results, in addition to choosing the best offer for the user.

The user agent performs the work according to algorithm 3.1 as it initializes the request message and sends it to the broker's agent. The broker agent extracts the requested resources from the request, searches for all the data center agents, it initiates the CNP by sending a request for bids to the data center agents and waits for a full set of responses within a specified time, if they are interested, the data center agent can make a proposal, or if they are not, they can send a rejection. This proposal contains all of the information needed for the broker agent to make a decision.

The broker’s agent performs the work according to algorithm 3.2 as it evaluates the offers received, making sure that the previously submitted offer will be excluded from the competition between offers. This process is done by checking the data of the service providers and their offers registered with the broker’s agent chooses the most appropriate proposal from the list and sends accept it to the corresponding agent then informed the other agents about its choice by sending them a reject and waits for a response from the data center agent whose performance is according to algorithm 3.3 When the contract is completed, the data center agent informs the broker agent using an inform message.

The offer is sent to the user agent through the "PROPOSE" message, who evaluates the offer and responds by accepting or rejecting the offer. A user agent sends an "ACCEPT _PROPOSAL" to the broker agent if the cost is acceptable. If the cost is higher than expected, send it "REJECT_ PROPOSAL".

If the broker's agent receives the "ACCEPT _PROPOSAL", a REQUEST (carry out tasks) is sent to the platform agent. In the event of receiving a "REJECT_ PROPOSAL", the data center agent fails to meet a Consumer Agent's request, the Broker Agent opens a negotiation with the resource data center agent. The terms of the agreement or negotiation process must be well understood by both parties (agents) to effectively finalize the agreement and all that. Willing to fulfill the conditions presented for the contract period in a simple net contract protocol, the parties are legally bound by the terms of the agreement. Other agents who bid on the contract may have previously taken on other commitments and are no longer willing to accept the offer from the agent who is no longer able to fulfill the contract. This means that it is expected that the entire procedure will need to send a new request.

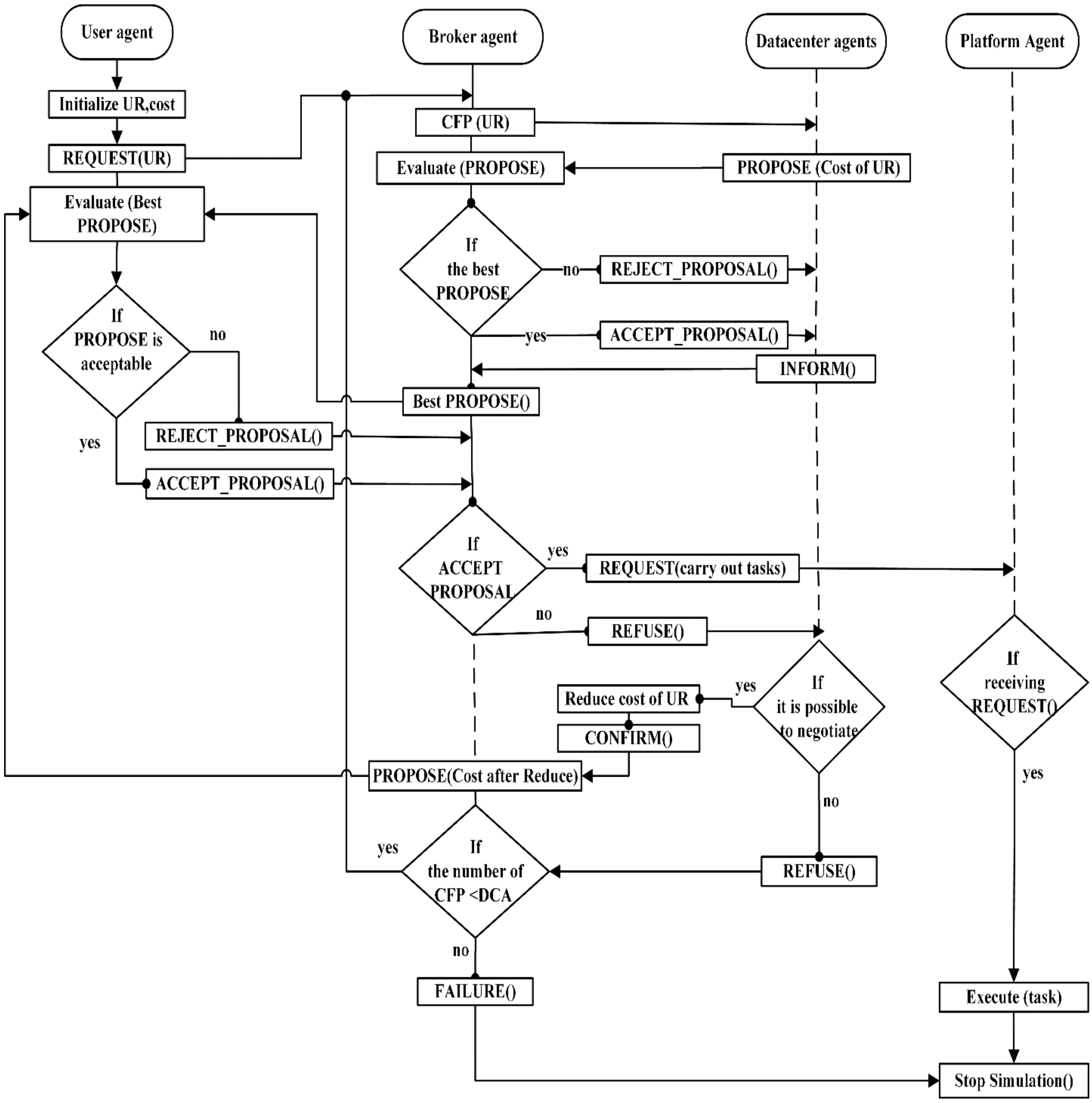

Figure 3.4 The third model sequence diagram

Table 3.1 as below shows some differences between the above models, Where the comparison includes the number of times requests are submitted by user agents, the number of times the request for proposals starts sent by send CFP from the broker agent to the data center agents, and finally the number of times of negotiation for each model

Table 3.1 comparison of models

Model	The number of requests	The maximum number of CFP	Maximum number of negotiations
Model-1	1	1	0
Model-2	1	1	Number of DCA negotiations
Model-3	1	Less or equal to the number of data center agents	Total number of DCA negotiations

The proposed framework consists of several agents, as shown in Figure 3.4, where the system is designed to automatically negotiate. The user does not need to know who the cloud service provider is and where the resources are located geographically. The consumer always gets the best offers available in Cloud Computing. There is no concern about this. The system is characterized by flexibility, and if the number of users increases, the number of agents increases automatically. The same is the case when the data centers increase or decrease, the agents will increase or decrease according to the situation that occurred.

1. User Agents (UA):

The user agent is responsible for receiving requests from users and sending them to the broker's agents through message passing. The results of the request are shown to the user via a user interface provided by the agent. To use the service, the User-Agent communicates with the Provider's Agent through the Broker's Agents.

User Agent responsibilities include:

- Receive consumer needs for the types of cloud resources required.
- Send service configuration requests to BA (Broker Agent).

• Evaluation whether the offer is appropriate for the user.

• Conduct negotiations if the offer is not suitable.

• the ability to Act on behalf Cloud users.

Algorithm 3.1 of the user agent below is used to perform the work and is as follows:

Algorithm 3.1: User-Agent (UA)
Input: Required resource specification, price, Required task
Output: ACCEPT_PROPOSAL, REJECT _ PROPOSA

Begin
1. Register UA in DFA agent // DFA is Directory Facilitator agent
2. initialize Price User, Resource User
3. AID[] searchResult = searchDF("BA") //AID (Agent-Identifier)
4. **While** (searchResult == null) **do**
4.1. Thread. Sleep (1000)
4.2. searchResult = searchDF("BA")
5. **End While**
6. **For** (AID BA: searchResult)
6.1. Add receiver (BA)
7. **End for**
8. Send REQUEST (UR) to BA // UR (User-requested resources)
9. **IF** receive PROPOSE (cost of UR) from BA then
9.1. **Evaluate (**Best PROPOSE**) //** A function to evaluate the offer received from BA
9.2. **IF** PROPOSE () is acceptable then
9.2.1. send “ACCEPT_PROPOSAL” to BA
9.3. **Else**

9.3.1. send “REJECT _ PROPOSAL”
9.3.2. goto step End
9.4. **End if**
10. **End if**
End

2. Broker Agents (BA):

It receives a request from the user agent, then sends a CFP (Call For Proposal) to the data center agents to start the bidding round. The data center agents receive offers from the data centers after a specified time, evaluate the offers received from the data centers, and select the best bid submitted, then send that bid to the user agent. He receives a response from him. By agreeing to the offer, the agent will be asked to carry out the tasks assigned to him or refuse the offer. In this case, the broker’s agent will conduct a negotiation process between the data center agent and the user agent, and the negotiations will be successful or unsuccessful depending on what is involved in the negotiation process between the agents. Algorithm 3.2 as show below is used to perform the functions of Broker agents create and provide virtual services for cloud clients. This is achieved by agent Responsibilities.

Broker Agent Responsibilities Include:

- Receive consumer requirements from UA.
- Find data center agents in the cloud.
- Send call for proposal to data center agent.
- Receive offers from data center agents.
- Matching offers and requirements
- Evaluate the offers received and choose the most suitable offers.

- Send the best offer to the user agent.
- Conducting the negotiation process between the two parties of the user agent and the agent of the service provider to reach an agreement between them.
- The BA acts as an intermediary between user agent and data center agent.

Algorithm 3.2: Broker-Agent (BA)
Input: Request (UR)
Output: send Request to PA for perform tasks / send failure to PA for stop

Begin
1. Register BA in DFA agent // DFA is Directory Facilitator agent
2. **IF** receive REQUEST (UR) from UA **then**
2.1. AID [] searchResult = searchDF("DCA") //DCA (Data-Center Agent)
2.2. **While** (searchResult== null) **do**
2.2.1. Thread. Sleep (1000)
2.2.2. searchResult = searchDF("DCA")
2.3. **End While**
2.4. **For** (AID DCA: searchResult)
2.4.1. Add receiver (DCA)
2.5. **End for**
2.6. send CFP (UR) to DCA
3. **End If**
4. **IF** handle Refuse from DCA **then**
4.1. Print “Agent Refuse”
5. **end If**
6. **IF** handle failure from DCA **then**
6.1. **IF** no replay **then**
6.1.1. Print “Reply does not exist”
6.2. **else**
6.2.1. Print “name Agent sender failed”
6.3. **End If**

7. **End If**
8. **IF** handle All Responses from DCA then
   8.1. **Evaluate** (PROPOSE)
   8.2. **Send** the state in the vector to DCAs
9. **End If**
10. **Function Evaluate (**PROPOSE**)**
   10.1. save All Responses in e as Enumeration
   10.2. **While** (e. has more elements ()) **do**
      10.2.1. **IF** (Resource matching (characteristics)==true)
      10.2.2. Calculate the cost of the VM depending on the bid by eq (1)
      10.2.3. **IF** PROPOSE **() is** acceptable **then**
         10.2.3.1. Save in Vector As "ACCEPT_PROPOSAL"
      10.2.4. **Else**
         10.2.4.1. Save in Vector As "RECJECT _ PROPOSAL"
      10.2.5. **End If**
      10.2.6. **End if**
   10.3. **End While**
   10.4. **Return Vector**
11.**End Function**
12.**IF** handle Inform from DCA **then**
   12.1. send the best PROPOSE() to UA
13.**End If**
14.**IF** receive "ACCEPT_PROPOSAL" from UA **then**
   14.1. send REQUEST() to PlataformAgent
15.**End If**
16.**IF** receive " REJECT_PROPOSAL " **then**
   16.1. send "REFUSE" to DC
17.**End If**

18.**IF** receive “CONFIRM” from DCA **then** // Receipt of a proposal after the negotiation process
    18.1. send PROPOSE (Cost after Reduce) to UA
    18.2. goto 8
19.**End If**
20.**IF** receive “REFUSE” from DCA **then**
    20.1. **IF** (the number of CFP <DCA) **then**
        20.1.1. goto 3
    20.2. **Else**
        20.2.1. send “FAILURE” to platform agent
    20.3. **End If**
21.**End If**
End

3. Data-Center Agents (DCA):

Each service provider owns one or more data centers. At each data center location, there is a data center agent who makes proposals on behalf of each data center and negotiates with the user agent to provide services, the algorithm for the work of the data center agent is algorithm 3.3, as shown below.

Data-Center agent responsibilities include:

- making offers to lease cloud resources to brokers.
- allocating/freeing cloud resources whenever transactions are agreed upon.
- Conducting negotiations with the user agent by offering discounts of offers at pre-determined rates for each negotiation process.
- ability to act on behalf cloud providers.

Algorithm 3.3: Data-Center Agent (DCA)
Input: CFP ( )
Output: Proposals to BA

Begin
1. Register DCA in DFA agent
2. initialize (number of time negotiation, discount percentage) // for each DCA
3. **IF** receive CFP(UR) from BA **then**
    3.1. Matching Protocol FIPA-CONTRACT-NET
    3.2. Prepare Response ()
    3.3. send PROPOSE **(**Cost of UR**)** to BA
4. **End if**
5. **IF** receive “ACCEPT PROPOSAL” from BA **then**
    5.1. send “INFORM” to BA
    5.2. Print “DCA Proposal accept “
6. **Else IF** receive “REJECT_PROPOSAL” from BA **then**
    6.1. Print “DC Proposal rejected “
7. **End if**
8. **IF** receive “REFUSE” from BA **then**
    8.1. **IF** it is possible to negotiate **then**
    8.2. Reduce cost of UR
    8.3. send “CONFIRM” and new price to BA
9. **Else**
    9.1. send “REFUSE” to BA
10.**End if**
11.**Else**
    11.1. Block ()
12.**End if**
End

4. Platform Agent (PA):

This agent has the primary task of implementing or canceling the implementation of resource allocation where the simulation process is resumed if the offer submitted by the service provider has been approved, and the execution is canceled if the process ends with the allocation of resources in failure after choosing the best offers and conducting negotiations with all service providers. The algorithm for the work of the PA is algorithm 3.4, as shown below.

Algorithm 3.4: Platform Agent (PA)
Input: Request or failure from BA
Output: Execution (task) or stop Simulation
Begin 1. Register PA in DFA agent 2. **IF** receive REQUEST(carry out tasks**)** form BA **then** 2.1. Execution (task) 3. **Else** 3.1. send “NOT_ UNDERSTOOD” to BA 4. **End if** 5. **IF** receive “FAILURE “from BA **then** 5.1. Stop_ Simulation () 6. **End if** **End**

3.3 Summary

In this chapter, we have presented the main phases and activities of our contributions, to know:

- A conceptual modeling of resource allocation in cloud computing.

- The proposal for an agent-based approach for the allocation of resources in the cloud computing.

In the context of the chapter, we presented the overall conception of the proposed approach which based on a multi-agent system, then we detailed the design of our system. We expressed our design approach, and more particularly the use case diagram to express the functionality of the system and the actors involved in its operation and sequence diagrams to express the behavior of the proposed system, in addition to a detailed explanation of the algorithms of the proposed model

In the next chapter, we look at the realization of our proposed system and this by implementing the approach presented in an environment involves the CloudSim simulator and the Jade platform.

Chapter 4 – Experiments and Results evaluation

Chapter Four

Experiments and Results evaluation

4.1 Introduction

The proposed models were compared by executing each model on the same used data and showing the results of each model. Due to the difficulty of implementing the system on actual cloud sites, the system has been implemented using two types of simulators, one is a cloud simulator (cloudsim) that provides an integrated cloud environment similar to the environment used by real computing, and the other is an Agent simulator JADE in order to implement and evaluate the proposed system. In this chapter, the results and performance analysis of the proposed system and models were evaluated, the use of requirements, resources, and tasks to test the proposed system are described in the previous chapter in Section 4.4.

The reported test results have implemented using Windows 10 pro with Java environment by using program Apache NetBeans IDE 11.3 in Intel(R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz and 16 GB RAM DDR4 and GPU card NVIDIA GeForce MX130, 2GB.

4.2 Types of messages in the proposed model

Agents communicate directly with each other by exchanging messages, as shown in Table 4.1, where there are basic types of messages. CFP messages are used for cooperation/service requests, PROPOSE messages are used for suggestions, CONFIRM messages are used for assertions, and INFORM messages are used to convey the outcome, either service or evaluation.

Table 4.1 Types of messages in the proposed system

Message	Acting models	Sender	Receiver	Function
Request	REQUEST (UA ,VM_description)	UA	BA	The UA asked the BA to provide a VM
	REQUEST(BA,service_description)	BA	PA	The BA requests the PA to perform to carry out the task
Propose	PROPOSE(DCA, offers)	DCA	BA	DCA send their offers to BA
	PROPOSE(BA, best offer)	BA	UA	The BA sends the best offer to the UA
	PROPOSE(BA, offer)	BA	UA	The BA sends the offer to the UA after negotiating with the DCA
Confirm	CONFIRM (DCA)	DCA	BA	Confirmation is sent to the BA if the DCA before negotiating the submitted offer
Inform	INFORM(DCA, result)	DCA	BA	DCA informs the BA that he is ready to provide the service
CFP	CFP(BA,service_description)	BA	DCA	BA asks DCA to start bidding
Accept proposal/ reject	ACCEPT PROPOSAL/ REJECT (UA)	UA	BA	Accept the offer submitted by the BA if the offer is suitable for the user or reject
	ACCEPT PROPOSAL/ REJECT (BA)	BA	DCA	BA accept the offer from DCA or not
	REJECT PROPOSAL (BA)	BA	DCA	A refusal is sent by the BA to the DCA if the UA rejects the best offer
Cancel	CANCEL (DCA)	DCA	BA	The DCA sends a refusal to negotiate to the BA if the

				provider service does not accept the negotiation
Agree	AGREE(PA)	PA	BA	He agrees to carry out the task
Not understood	NOT_UNDERSTOOD(DCA)	DCA	BA	If DCA does not understand what the specific purpose of the message is
REFUSE	REFUSE()	BA	DCA	The offer was rejected by UA
	REFUSE()	DCA	BA	Refusal to negotiate with the BA
FAILURE	FAILURE()	BA	PA	notification PA of operation failure for stop Simulation

4.3 Simulation Environment

The proposed system is primarily based on the JADE. is a platform for creating, controlling, and developing multi-agent systems written in the Java programming language, by F. Bellifemine, A. Poggy, G. Rimassa, and P. Turci for the company CSELT (Italy) in 1999 [73]. JADE complies with FIPA Standards 1997. The platform provides generic reception, identification, and communication services between agents. These services are:

- AMS (Agent Management System): it is in a way the heart of the platform FIPA form. It registers active agents, manages their identities, and keeps track of their states.
- The DF (Directory Facilitator): is a directory service allowing to identify user services on a platform.
- The ACC (Agent Communication Channel) is a specific agent responsible for check the messages between the different agents coming from FIPA platforms (or even non-FIPA) possibly distant. As a result, it offers a service reliable and precise for routing messages. In addition, it ensures interoperability between different platforms

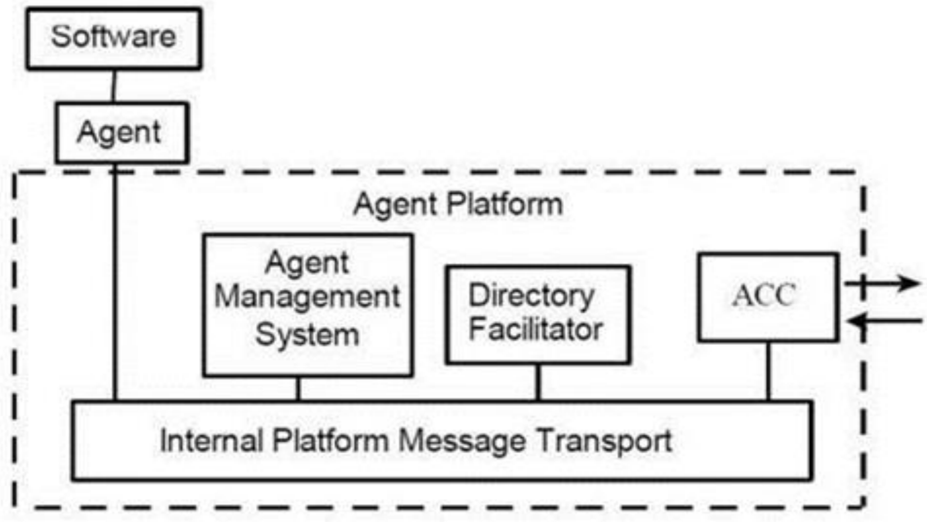

Figure 4.1 FIPA Reference model for Agent [73]

The multi-agent JADE platform is made up of several receptacles (containers) agents. Distribution of these receptacles through a computer network is permitted. Each agent receptacle is a multi-threaded, compound execution environment with an execution thread for each agent, in addition to the threads created at execution by the Remote Method Invocation **(**RMI) system to send messages. Only one receptacle is the main one. It is the one that contains the agents and the platform (AMS, ACC, and DF). The platform offers a graphical user interface (GUI) for management at a distance from agents. Communication between agents and the GUI and all the communication between this interface and the AMS is done by FIPA-ACL. Two tools graphics are available:

- The Dummy agent whose role is to inspect the exchanges of messages between agents, and allows to edit, write, send, receive and save messages in FIPA-ACL.
- Agent Sniffer gives a graphical interface to display Message's exchange between the different group's agents using a notation close to Unified Modeling Language (UML).

The simulation is performed in Figure 4.2 showing the appearance of the JADE platform.

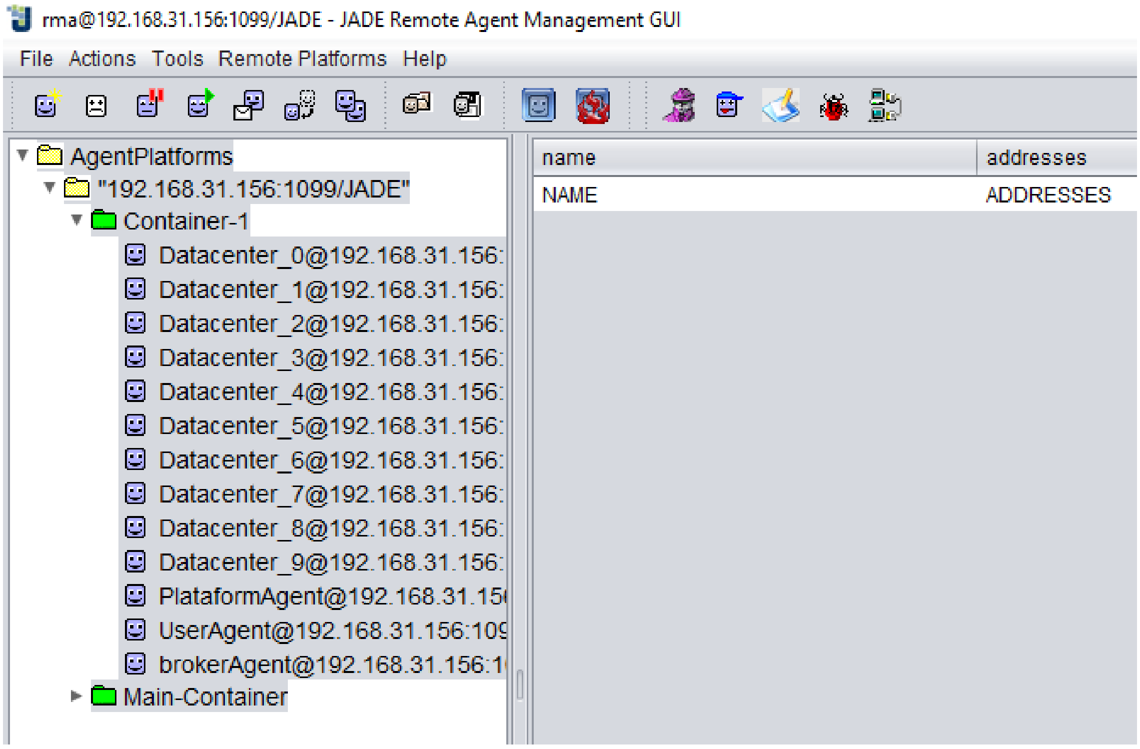

Figure 4.2 The Simulator Platform

4.4 Simulation Settings

The system consisting of several agents is simulated according to Table 4.2 below for each agent with a specific task. The system consists of four types of agents that are simulated, user agent, broker agent, data center agent, and platform agent.

Table 4.2 The number of agents

No	Name agent	Number agent
1	User agent	100
2	Broker agent	1
3	Data-Center agent	10
4	Platform agent	1

Assumptions and conditions for the scenario: First the use of two virtual machines is detailed in the Table 4.3 below.

Table 4.3 Hardware specifications for VM

Feature	VM
Storge	10 GB
RAM	512 MB
MIPS	250
Bandwidth	1000 MHz
Number of Pes	1
VMM	XEN

Second, using the number of hosts in the Table 4.4.

Table 4.4 Specifications of Hosts

Feature	Specifications of Host
Number of Pes	4
MIPS	10000
RAM	100 GB
Bandwidth	10000 MHz
Storage	10 TB

Third, the number of hosts (10 hosts) is determined; Equally divided into ten groups, each data center has one hosts. Fourth, the characteristics of the amount of VM required are the same and the cost required for each type of virtual machine is different because each service provider has its own price as in the Table 4.5 the data center information including the costs of CPU, Ram, Storge, and Bw determined by the service provider for each data center, In addition to the number of negotiation times and the percentage of reduction for each data center.

Table 4.5 Data Center Information

Name DC	Cost CPU (unit)	Cost ram (unit)	Cost storge (unit)	Cost Bw (unit)	Num times negotiate	Discount percentage
Datacenter_0	4	0.33	0.02	0.01	3	2
Datacenter_1	4	0.372	0.02	0.01	3	3
Datacenter_2	5	0.51	0.01	0.01	2	1
Datacenter_3	6	0.51	0.01	0.02	1	5
Datacenter_4	5	0.59	0.01	0.01	0	0
Datacenter_5	4	0.4	0.02	0.02	2	1
Datacenter_6	5	0.5	0.01	0.02	0	0
Datacenter_7	4	0.43	0.02	0.01	0	0
Datacenter_8	5	0.381	0.02	0.03	2	4
Datacenter_9	6	0.5	0.01	0.01	1	1

4.5 Software Requirements

The next subsection describes the software requirements which have been used to test and validate the suggested solutions.

4.5.1 NetBeans Program Version 11.3

NetBeans is an integrated development environment (IDE) for Java. It allows applications to be developed from a set of modular software components called modules, which run on Windows, macOS, Linux, and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript [76].

4.5.2 Java Development Kit (JDK) Version 8.0.2810

JDK is a development environment that includes useful tools for developing and testing programs written in the Java programming language and running on the Java platform [77].

4.5.3 JADE 4.5.0 software package

JADE is a software framework to make easier the development of agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems [78][79].

4.5.4 CloudSim 3.0.3 version

Cloudsim is one of the main cloud simulators, it is a framework that allows you to simulate an entire cloud-like infrastructure by writing a program using the objects provided by the programming interface provided by the library. It is a utility primarily intended for design research and assessment of the underlying architecture of IT service platforms at the request. It provides a model of the behavior of the Data Center as a whole[80].

4.6 Experimental Results

In this section, experimental results are presented. The test was conducted by 100 users and for each user there is a price that they pay for buying resources from the cloud. The user information is read through a CSV file to test all models. As for the resource cost prices that will be presented as offers to the user, each data center has its own prices that vary from one data center to another, as shown in Table 4.5. In addition to other information that is used in the test, such as the number of times to negotiate and percentages of reduction

When comparing the proposed protocol with the work presented by <> the results showed that the proposed protocol is better as it is efficient, includes CNP protocol and negotiation algorithm as shown in the table below.

The proposed model was evaluated based on two parameters include (the number of execution and failure cases users' requests, and the second factor is the number of messages exchanged in the system), which reflects the system's effectiveness and success in allocating computing resources using the multi-agent system.

4.6.1 The First parameter (User request) :

The system execution state is tested by determining the number of successes and failures. The test was repeated using more than one paradigm in each experiment.

The first model was used in the test represents the standard model of the agent using the Contract Net Protocol (CNP) interactive protocol between agents by exchanging messages. The simulation results have shown a system Success 55% (the possibility of executing user requests) in this model and a system failure 45% (system failure: is, the system's failure to implement the users' requests),

The second model we designed and implemented a negotiation algorithm between agents in order to obtain the best offers for users from the total offers submitted. As for the communication between agents, it is using CNP. The number of success cases rate was 67%, failures in this model was 33% of the total users, which is a better percentage than in the first model.

The third model is distinguished, with the possibility of re-requesting proposals from data center agents through the Broker agent in the event that no agreement was reached that satisfies the user agent and the data center agent (negotiation failed). The number of success cases rate 82%, The failure rate has decreased to 18%.

Figure 4.3 shows the full picture of the three models, including the proposed model, which shows the success cases for executing the requests and the failure cases classified according to each model through it we note that the success cases increased from the first model to the second and third model, which is the proposed model gradually. As for the failure, it decreased, moving from the first form to the third form.

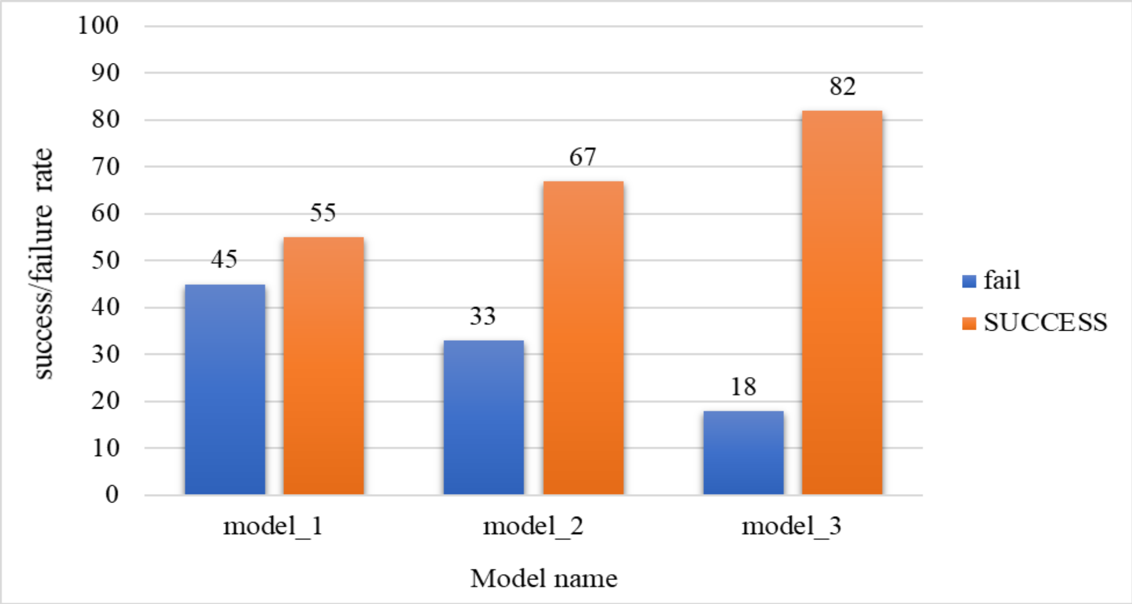

Figure 4.3: Success and failure rates (request)

4.3.2 The second parameter (messages exchanged)

The second test of the proposed system is through the number of messages exchanged in each model.

The simulation results of the first model showed that in the case of failure (the system's inability to implement the users' requests) we note that there are 36 messages exchanged, and in the case of successful execution of the requests (the possibility of executing user requests), the number of messages exchanged is 37 messages.

The second form, as mentioned previously, has a negotiation process in the system. The number of messages exchanged in case of success there are two numbers the first number is 37 messages because it is without a negotiation process, and the second number is 41 through a one-time negotiation with service provider. We notice an increase in the number of messages exchanged in the case of fails to achieve user requests after negotiating, the number of messages exchanged between agents in the system is 41.

The third model the number of messages in case of failure is 528 and the reason for this large number of messages is that the system negotiated with more than one data center agent to meet the user requirements and each agent had a number of authorized negotiations but they were to no avail. In the case of implementation succeed, the number of messages exchanged between agents 48 The model succeeded in being executed without negotiation. The number 52 indicates that negotiations took place only once, and the last number, 205, indicates more than one negotiation that took place between agents to meet the requirements successfully.

Show in Figure 4.4 a comparison between the exchange of messages for the success case between the three models, which shows that the proposed system contains more messages exchanged than the other models, that is, the largest possible number of user requests have been fulfilled and the increase is due to the system trying to negotiate with data center agents to implement requests users. As for the different numbers for the same model, they are according to the number of times of negotiation, as the figure shows that the success cases are three cases, the first without negotiation, the second negotiation once, and the third four times to negotiate, all of which are characterized by different colors.

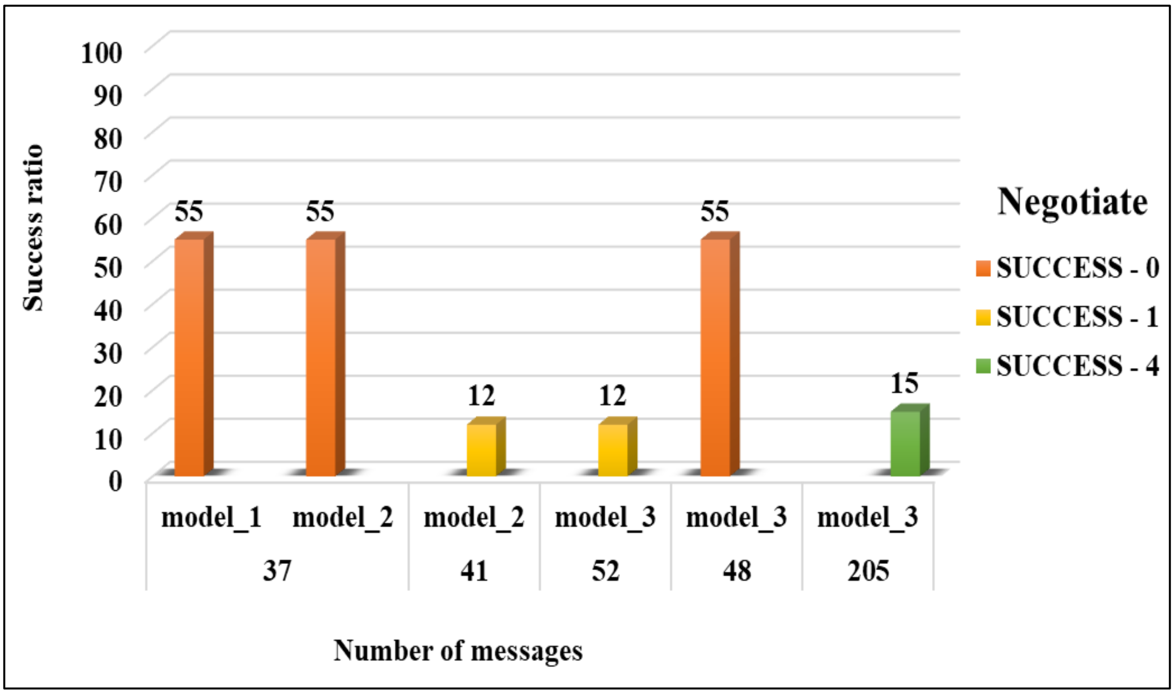

Figure 4.4: The number of messages in case of success

Figure 4.5 shows a comparison of failures between the three models, which shows that the proposed system includes the greatest number of messages exchanged between agents, amounting to 528 messages, and this large number is due to the failure of negotiations between the number of data center agents to more than once. The first model, the number was lower compared to the two models because it does not involve a negotiation process, so the number of messages is few. For the second model, because it has a negotiation algorithm, the number is more than 36 the first without negotiation and 41 with the negotiation process.

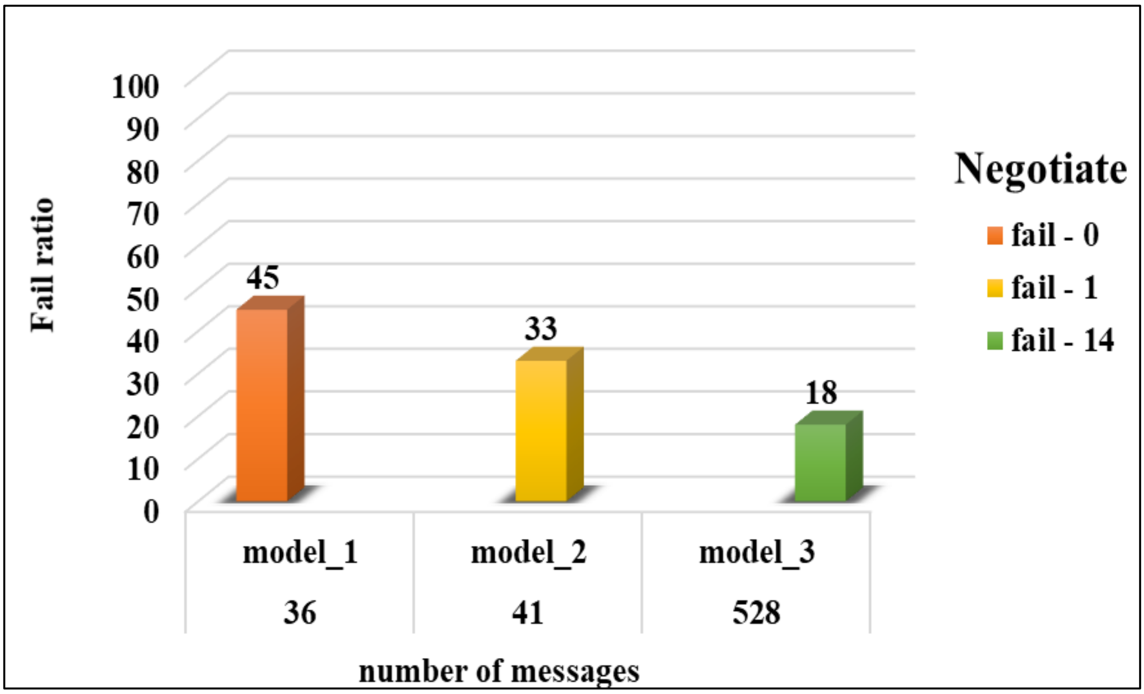

Figure 4.5: The number of messages in case of failure

4.4 Discussion of the proposed system

The proposed protocol is characterized by efficiency to implement the requirements of the users because it includes a negotiation automatically between agents by depending on the (CNP) protocol This will facilitate the work on the cloud service providers as well as the users Because it does not require direct communication between the user and the service provider to conduct negotiations to reach an agreement between the two parties.

The results showed that the proposed system is efficient in implementing the requirements of users by always getting the best offer from the total of offers, as well as because it includes a negotiation process on the offer submitted by the agents of the data centers and user agent through the use of the (CNP) protocol for interaction and cooperation between agents. On the other hand, we need to add the features below to the proposed protocol.

1. Dependence of resource allocation on geographical location and service quality factors.

2. Adding a trusted agent to the proposed system whose task is to evaluate and verify the reliability of service providers.

4.5 Summary

In this chapter to implement the set of ideas that characterizes the proposed approach by focusing on the implementation of multi-agent systems as well as the integration between different agents. Our architecture is implemented using a development environment JAVA which includes the CloudSim simulator to simulate the cloud computing environment and the Jade platform which allows you to visualize the interaction between all the agents implemented, and used of requirements, resources and tasks for testing as show in the previous chapter in Section 4.4 to compare and evaluate.

Chapter 5 – Conclusions and Future Work

Chapter Five

Conclusions and Future Work

5.1 Conclusions

Multi-Agents System (MAS) technology introduces an ideal way for open and scalable systems that is varied dynamically. In this paper a new design was introduced which is composed of MAS that being used by the cloud environment to help in choosing the best resources and to create negotiation technique between cloud providers and users to make use of the full potential of cloud computing.

The objective of this thesis is to propose an approach for Cloud Resources Allocation Using Multi-Agent System to meet the needs of cloud users by the choose to the best-offer resources offers by the cloud service providers which makes it possible to reduce the cost of the service or application.

The results showed that the proposed protocol is characterized by efficiency to implement the requirements of the users because it includes a negotiation automatically between agents by depending on the (CNP) protocol This will facilitate the work on the cloud service providers as well as the users Because it does not require direct communication between the user and the service provider to conduct negotiations to reach an agreement between the two parties, the results also confirm that the use of a multi-agent system for allocating resources in cloud computing ensures customer satisfaction, which is to reduce the cost of the resources being marketed to try to use the service at the lowest cost.

5.2 Future Work

This thesis constitutes a working basis from which, new activities of research can be initiated in order to improve the work presented. The perspectives that we can therefore move in the following directions:

1. Proposal of optimization methods to minimize the number of nodes physical, reduce in energy consumption etc. (at the data center management layer).
2. Dependence of resource allocation on factors of geographic location and QoS.
3. Avoiding application performance degradation, by tolerating the error and Commitment to the Service Level Agreement is another key factor to consider.
4. Adding a trusted agent to the proposed system whose task is to evaluate and verify the reliability.

References

References

[1] R. Tudoran, A. Costan, and G. Antoniu, "OverFlow: Multi-site aware big data management for scientific workflows on clouds," *IEEE Trans. cloud Comput.*, vol. 4, no. 1, pp. 76–89, 2015.

[2] A. Mochón and Y. Sáez, "Introduction to Combinatorial Auctions," in *Understanding Auctions*, Springer, 2015, pp. 87–103.

[3] L. Blumrosen and N. Nisan, "Informational limitations of ascending combinatorial auctions," *J. Econ. Theory*, vol. 145, no. 3, pp. 1203–1223, 2010.

[4] D. C. Parkes and M. P. Wellman, "Economic reasoning and artificial intelligence," *Science (80-.).*, vol. 349, no. 6245, pp. 267–272, 2015.

[5] L. Zhang and X. Chen, "An Agent-based Multi-attribute Sealed-bid Design for Bilateral Contract.," *J. Softw.*, vol. 4, no. 1, pp. 65–72, 2009.

[6] C. G. Ralha, A. H. D. Mendes, L. A. Laranjeira, A. P. F. Araújo, and A. C. M. A. Melo, "Multiagent system for dynamic resource provisioning in cloud computing platforms," *Future Generation Computer Systems*, vol. 94. pp. 80–96, 2019, doi: 10.1016/j.future.2018.09.050.

[7] T. Alwadan, "Cloud computing and multi-agent system: Monitoring and services," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 05, 2018.

[8] A. Mazrekaj, D. Minarolli, and B. Freisleben, "Distributed resource allocation in cloud computing using multi-agent systems," *Telfor J.*, vol. 9, no. 2, pp. 110–115, 2017, doi: 10.5937/telfor1702110m.

[9] W. Wang, Y. Jiang, and W. Wu, "Multiagent-based resource allocation for energy minimization in cloud computing systems," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 2, pp. 205–220, 2016.

[10] F. De la Prieta and J. M. Corchado, "Cloud computing and multiagent systems, a promising relationship," in *Intelligent Agents in Data-intensive Computing*, Springer, 2016, pp. 143–161.

[11] M. E. K. Fareh, O. Kazar, M. Femmam, and S. Bourekkache, "An agent-based approach for resource allocation in the cloud computing environment," *Proceeding 2015 9th Int.*

Conf. Telecommun. Syst. Serv. Appl. TSSA 2015, 2016, doi: 10.1109/TSSA.2015.7440447.

[12] M. Al-Ayyoub, M. Daraghmeh, Y. Jararweh, and Q. Althebyan, "Towards improving resource management in cloud systems using a multi-agent framework," *Int. J. Cloud Comput.*, vol. 5, no. 1–2, pp. 112–133, 2016.

[13] G. K. Shyam and S. S. Manvi, "Resource allocation in cloud computing using agents," in *2015 IEEE International Advance Computing Conference (IACC)*, 2015, pp. 458–463.

[14] F. Farahnakian, P. Liljeberg, T. Pahikkala, J. Plosila, and H. Tenhunen, "Hierarchical vm management architecture for cloud data centers," in *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, 2014, pp. 306–311.

[15] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[16] D. Dempsey and F. Kelliher, "Cloud computing: the emergence of the 5th utility," in *Industry Trends in Cloud Computing*, Springer, 2018, pp. 29–43.

[17] A. D. JoSEP, Ra. KAtz, A. KonWinSKi, L. E. E. Gunho, Dav. PAttERSon, and Ar. RABKin, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[18] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.

[19] C. Sethi and S. K. Pradhan, "A Comparison on Cloud Computing and Grid Computing," *Int. J.*, vol. 6, no. 2, 2016.

[20] F. F. Moghaddam, M. B. Rohani, M. Ahmadi, T. Khodadadi, and K. Madadipouya, "Cloud computing: Vision, architecture and Characteristics," in *2015 IEEE 6th Control and System Graduate Research Colloquium (ICSGRC)*, 2015, pp. 1–6.

[21] L. Wang *et al.*, "Cloud computing: a perspective study," *New Gener. Comput.*, vol. 28, no. 2, pp. 137–146, 2010.

[22] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition." ACM New York, NY, USA, 2008.

[23] D. Plummer, T. J. Bittman, T. Austin, D. C. Cearley, and D. M. Smith, "Cloud Computing: Defining and Describing an Emerging Phenomenon," no. June, 2009,

[Online]. Available: http://my.gartner.com/portal/server.pt?showOriginalFeature=y&open=512&objID=260&mode=2&PageID=3460702&id=697413&ref=.

[24] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing - The business perspective," *Decis. Support Syst.*, vol. 51, no. 1, pp. 176–189, 2011, doi: 10.1016/j.dss.2010.12.006.

[25] "Ibm, «Ibm cloud,» IBM, [En ligne] Available: http://www.ibm.com/cloud-computing/us/en/whatis-cloud-computing.html. [Accès le 07 octobre 2018].," 2018. .

[26] I. T. Gartner, "Gartner IT glossary," *Technol. Res.*, 2013.

[27] Y. Yang, K. Liu, J. Chen, X. Liu, D. Yuan, and H. Jin, "An algorithm in SwinDeW-C for scheduling transaction-intensive cost-constrained cloud workflows," in *2008 IEEE Fourth International Conference on eScience*, 2008, pp. 374–375.

[28] A. Fox *et al.*, "Above the clouds: A berkeley view of cloud computing," *Dept. Electr. Eng. Comput. Sci. Univ. California, Berkeley, Rep. UCB/EECS*, vol. 28, no. 13, p. 2009, 2009.

[29] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, 2011.

[30] P. Wieder, J. M. Butler, W. Theilmann, and R. Yahyapour, *Service level agreements for cloud computing*. Springer Science & Business Media, 2011.

[31] R. Buyya, S. K. Garg, and R. N. Calheiros, "SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions," in *2011 international conference on cloud and service computing*, 2011, pp. 1–10.

[32] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *2008 10th IEEE international conference on high performance computing and communications*, 2008, pp. 5–13.

[33] L. Wang, J. Tao, M. Kunze, A. C. Castellanos, D. Kramer, and W. Karl, "Scientific cloud computing: Early definition and experience," in *2008 10th ieee international conference on high performance computing and communications*, 2008, pp. 825–830.

[34] C. N. Höfer and G. Karagiannis, “Cloud computing services: taxonomy and comparison,” *J. Internet Serv. Appl.*, vol. 2, no. 2, pp. 81–94, 2011.

[35] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering cloud computing: foundations and applications programming*. Newnes, 2013.

[36] V. K. Reddy, B. T. Rao, and L. S. S. Reddy, “Research issues in cloud computing,” *Glob. J. Comput. Sci. Technol.*, 2011.

[37] S. Zhang, H. Yan, and X. Chen, “Research on key technologies of cloud computing,” *Phys. Procedia*, vol. 33, pp. 1791–1797, 2012.

[38] L.-J. Zhang, J. Zhang, J. Fiaidhi, and J. M. Chang, “Hot topics in cloud computing,” *IT Prof.*, vol. 12, no. 5, pp. 17–19, 2010.

[39] N. Sadashiv and S. M. D. Kumar, “Cluster, grid and cloud computing: A detailed comparison,” in *2011 6th International Conference on Computer Science & Education (ICCSE)*, 2011, pp. 477–482.

[40] T. M. L. Nussbaum, G. V. Kherbache, and M. Moussalih, “Cloud Computing.”

[41] V. J. R. Winkler, *Securing the Cloud: Cloud computer Security techniques and tactics*. Elsevier, 2011.

[42] L. Wu and C. Yang, “A solution of manufacturing resources sharing in cloud computing environment,” in *Cooperative design, visualization, and engineering*, Springer, 2010, pp. 247–252.

[43] D. E. Y. Sarna, *Implementing and developing cloud computing applications*. CRC Press, 2010.

[44] A. Lee-Post and R. Pakath, “Cloud computing: a comprehensive introduction,” in *Security, Trust, and Regulatory Aspects of Cloud Computing in Business Environments*, IGI Global, 2014, pp. 1–23.

[45] D. Rountree and I. Castrillo, *The basics of cloud computing: Understanding the fundamentals of cloud computing in theory and practice*. Newnes, 2013.

[46] A. Tchana, “Système d’Administration Autonome Adaptable : application au Cloud,” no. November 2011, 2014.

[47] O. Leclère and O. Leclère, "Mise en place d ' un site type Web 2 . 0 sur un Cloud To cite this version : HAL Id : dumas-00524318 par," 2010.

[48] S. Tamane, "A review on virtualization: A cloud technology," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 3, no. 7, pp. 4582–4585, 2015.

[49] A. Obied, M. A. Hajer, and A. H. Hasan, "Computerized situated agent as mediator in centralized computing market," *J. Adv. Res. Dyn. Control Syst.*, vol. 11, no. 2 Special Issue, pp. 344–350, 2019.

[50] Y. Demazeau and A. Costa, "Populations and organizations in open multi-agent systems," *... 1st Natl. Symp. ...*, no. June, pp. 1–13, 1996, [Online]. Available: http://www.rocha.c3.furg.br/arquivos/download/demazeau-costa-populations-organizations.pdf.

[51] A. Agents, M. Systems, and Q. Mary, "A Roadmap of Agent Research and Development," *Auton. Agent. Multi. Agent. Syst.*, vol. 38, no. 1.1, pp. 7–38, 1998.

[52] D. Ye, M. Zhang, and A. Vasilakos, "A Survey of Self-Organization Mechanisms in Multiagent Systems," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 3, pp. 441–461, 2017.

[53] F. Jacques, "Multi-agent systems. Towards collective intelligence," *InterEditions, Paris*, vol. 322, 1995.

[54] J. Ferber, "Reactive multi-agent systems: principles and applications," *Fundam. Distrib. Artif. Intell. Ed. by N. Jennings, Amsterdam North Holl.*, 1994.

[55] M. Bratman, *Intention, plans, and practical reason*, vol. 10. Harvard University Press Cambridge, MA, 1987.

[56] O. Boissier and Y. Demazeau, "ASIC: An architechture for social and individual control and its application to computer vision," in *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 1994, pp. 135–149.

[57] B. Chaib-draa, "Interaction between agents in routine, familiar and unfamiliar situation," *Int. J. Coop. Inf. Syst.*, vol. 5, no. 01, pp. 1–25, 1996.

[58] N. Shadbolt and G. Zu, "An architecture for dynamic, rational, mobile agents," in

Proceedings of the Intelligent Vehicles' 94 Symposium, 1994, pp. 515–519.

[59] I. Jarras and B. Chaib-Draa, "Overview of multiagent systems," Cirano, 2002.

[60] J. Odell, "Objects and agents: How do they differ?," *JOOP - J. Object-Oriented Program.*, vol. 13, no. 6, pp. 50–53, 2000.

[61] B. L. Wang, "Comparison of Objects and Agents," p. 4.

[62] M. Wooldridge, "Intelligent Agents," *Multiagent Syst.*, p. 1, 2013.

[63] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[64] "Agent Environment in AI - Javatpoint." https://www.javatpoint.com/agent-environment-in-ai (accessed Oct. 03, 2021).

[65] "6 Types of Artificial Intelligence Environments | by Jesus Rodriguez | Medium." https://jrodthoughts.medium.com/6-types-of-artificial-intelligence-environments-825e3c47d998 (accessed Oct. 02, 2021).

[66] "Artificial Intelligence/AI Agents and their Environments - Wikibooks, open books for an open world." https://en.wikibooks.org/wiki/Artificial_Intelligence/AI_Agents_and_their_Environments (accessed Oct. 03, 2021).

[67] M. Wooldridge and M. Wooldridge, *MultiAgent Systems*. .

[68] B. Chaib-draa and F. Dignum, "Trends in agent communication language," *Comput. Intell.*, vol. 18, no. 2, pp. 89–101, 2002.

[69] T. Finin, Y. Labrou, and J. Mayfield, "KQML as an agent communication language," 1995.

[70] T. F. Foundations, "Foundation for intelligent physical agents specifications," Technical report, The FIPA Consortium, 2003.

[71] G. K. Soon, C. K. On, P. Anthony, and A. R. Hamdan, "A review on agent communication language," in *Computational Science and Technology*, Springer, 2019, pp. 481–491.

[72] J. L. Austin, "How to do things with words Oxford University Press." Oxford, 1962.

[73] F. Bellifemine, G. Caire, T. Trucco, and G. Rimassa, "Jade programmer's guide," *Jade version*, no. C, pp. 1–49, 2010, [Online]. Available: http://banzai-deim.urv.net/~riano/teaching/jade.pdf.

[74] O. Gutknecht and J. Ferber, "Madkit: A generic multi-agent platform," in *Proceedings of the fourth international conference on Autonomous agents*, 2000, pp. 78–79.

[75] M. V Dignum, "A landscape of agent systems in the real world." UU WINFI Informatica en Informatiekunde, 2006.

[76] I. Kostaras, C. Drabo, J. Juneau, S. Reimers, M. Schröder, and G. Wielenga, "Porting an Application to the NetBeans Platform," in *Pro Apache NetBeans*, Springer, 2020, pp. 255–297.

[77] K. Arnold, J. Gosling, and D. Holmes, *The Java programming language*. Addison Wesley Professional, 2005.

[78] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE*, vol. 7. John Wiley & Sons, 2007.

[79] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi, "JADE—a java agent development framework," in *Multi-agent programming*, Springer, 2005, pp. 125–147.

[80] T. Goyal, A. Singh, and A. Agrawal, "Cloudsim: simulator for cloud computing infrastructure and modeling," *Procedia Eng.*, vol. 38, pp. 3566–3572, 2012.

[81] Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," *IEEE Trans. Comput.*, vol. C–29, no. 12, pp. 1104–1113, 1980, doi: 10.1109/TC.1980.1675516.

[82] "Supply Chain - Overview, Importance, and Examples." https://corporatefinanceinstitute.com/resources/knowledge/strategy/supply-chain/ (accessed Sep. 28, 2021).

[83] "Sensor networks (article) | Khan Academy." https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:computing-innovations/xcae6f4a7ff015e7d:monitoring-innovations/a/sensor-networks (accessed Sep. 28, 2021).

الخلاصة

يوفر نموذج الحوسبة السحابية مجموعة مشتركة من الموارد مثل أنظمة التخزين ووحدة المعالجة المركزية وعرض النطاق الترددي للشبكة من واحد أو أكثر من موفري السحابة لفترة زمنية محدودة بسعر متغير أو ثابت مع نماذج مختلفة يتم تسليمها للعملاء عبر الإنترنت عبر - نموذج طلب قابل للتطوير ديناميكيًا يتم احتسابه باستخدام نموذج الدفع لكل استخدام.

مشكلة البحث التي تواجه المستهلكون هي استغلال الموارد بناءً على احتياجاتهم بأقل تكلفة. بالإضافة إلى ذلك، فإن اكتشاف موارد السحابة الآلية والذكية يمثل تحديًا يجب معالجته أيضًا. في معظم الأحيان، يواجه مستخدم السحابة مشكلة في التفاوض مع مزود الخدمة لأن واجهات التفاوض غير متوفرة في معظم مواقع الويب.

تتناول هذه الرسالة مراحل تصميم وتنفيذ وتقييم بروتوكول تخصيص الموارد القائم على العوامل المتعددة للحوسبة السحابية، لتحقيق الأهداف التالية:

١. وضع إطار عمل لدراسة أداء الحلول المستندة إلى الوكيل لتحديات تخصيص الموارد في إعداد الحوسبة السحابية.

٢. تطوير مكون إدارة الموارد الذي يعتمد على تقنيات الوكيل وخوارزميات الذكاء الاصطناعي.

معدل تنفيذ النموذج الأول هو ٥٥٪ معدل التنفيذ لمتطلبات المستخدمين. أما النموذج الثاني فيتضمن التفاوض مع وكيل مركز بيانات واحد فقط، حيث بلغ معدل تنفيذ متطلبات المستخدم ٦٧٪. يتضمن النموذج الثالث خوارزمية تفاوض، بالإضافة إلى إمكانية إعادة التفاوض مع وكلاء مركز البيانات الآخرين في حالة فشل المفاوضات الحالية في الوصول إلى حل وتقديم عرض يتناسب مع طلب المستخدم بلغ معدل تنفيذ طلبات المستخدمين ٨٢٪ وهو أفضل نموذج مقترح.

تؤكد النتائج أن البروتوكول المقترح فعال في تنفيذ متطلبات المستخدمين لأنه يتضمن التفاوض التلقائي بين الوكلاء بناءً على بروتوكول شبكة العقد (CNP) للتفاعل والتنسيق بين الوكلاء تؤكد النتائج أيضًا على ضمان رضا العملاء الذي يتمثل في محاولة استخدام الخدمة بأقل تكلفة.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة القادسية
كلية علوم الحاسوب وتكنولوجيا المعلومات

تخصيص الموارد السحابية باستخدام بروتوكول متعدد الوكيل بالاعتماد على تفضيلات المستخدم

رسالة ماجستير

مقدمة إلى مجلس كلية علوم الحاسوب وتكنولوجيا المعلومات في جامعة القادسية كجزء من متطلبات نيل شهادة الماجستير في تخصص علوم الحاسبات

من قبل
فـــؤاد جـــوده حســين

تحت اشراف
الأستاذ المساعد الدكتور
منتصر جابر جواد العصفور

١٤٤٣ ٢٠٢١