ENHANCED CLOUD SERVICE BROKER SELECTION POLICY BASED ON DIFFERENTIAL EVOLUTION ALGORITHM

YOUSEF KHADER KHATTAR SANJALAWE

UNIVERSITI SAINS MALAYSIA

2020

ENHANCED CLOUD SERVICE BROKER SELECTION POLICY BASED ON DIFFERENTIAL EVOLUTION ALGORITHM

by

YOUSEF KHADER KHATTAR SANJALAWE

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

October 2020

ACKNOWLEDGEMENT

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

{نَرْفَعُ دَرَجَاتٍ مِّن نَّشَاء وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ -76 - }   {سورة يوسف}

First of all, I would like to thank “Allah”, our creator, for giving me his blessings and the opportunity to complete my Ph.D.

Now, I would like to share my happiness by acknowledging some of the numerous people who have helped me directly or indirectly in shaping my academic career over the past years. I would like to express my deepest gratitude to my learned supervisors Dr. Mohammed F. R Anbar (main supervisor), and Prof. Rosni Abdullah (Co-supervisor), for their encouragement, innovative suggestions, and invaluable help during the entire period of my Ph. D at National Advanced IPv6 Centre of Excellence (NAv6), which is a high-profile institution. I also wish to thank my research committee members for providing insightful and constructive comments.

Most importantly, I would like to thank my Family. Their support, encouragement, patience, and unwavering love are undeniably the foundation upon my success. Their tolerance of my occasional discourteous moods is a testament to her unyielding devotion and love. I thank all of them, for their faith in me and for allowing me to be as ambitious as I wished. It was under their watchful eye that I gained the drive and ability to tackle challenges head-on. Last but not the least; I would like to dedicate this thesis to my father’s soul, as he always provides unconditional support and guidance as needed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

AEUR	Average of Executed Users Requests
AvgPT	Average Processing Time
AvgRT	Average Response Time
AvgWT	Average Waiting Time
CC	Cloud Computing
CDCP	Closest Data Center Policy
CPU	Central Processing Unit
CSB	Cloud Service Broker
CSPs	Cloud Service Providers
CURS	Current User Request Size
DC	Data Center
DCAV	Data Center Availability
DCEFF	Data Center Efficiency
DCPP	Data Center Processing Power
DCUT	Data Center Utilization Percentage
DDCV	Donor Data Center Vector
DE	Differential Evolution
DRCSB	Dynamically Reconfigurable Routing with Load Balancing
DTC	Data Transfer Cost
DV	Dimensional Vectors
EC2	Elastic Compute Cloud
EDoS	Economic Denial of Sustainability
ExpPT	Expected Processing Time
ExpRT	Expected Response Time
ExpTotC	Expected Total Cost
GA	Genetic Algorithm
GUI	Graphical User Interface
HPLC	Highest Performance and Lowest Cost
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
IdT	Idle Time Percentage

IT	Information Technology
JAR	Java Archive
JDK	Java Development Kit
LB	Load Balancing
MIPS	Million Instructions per Second
MOF	Multi-Objective Function
NoVM	Number of Virtual Machine
NxUR	Next User Request
OF	Objective Function
ORTP	Optimized Response Time Policy
OS	Operating System
OSBRP	Optimized Service Broker Routing Policy
PaaS	Platform as a Service
PN	Population Numbers
PR	Processing Ratio
PT	Processing Time
PUT	Processor Utilization Percentage
QoS	Quality of Service
QoS	Quality-of-Service
RAM	Random Access Memory
Reg	Region
RMI	Remote Method Invocation
RR	Round Robin
RT	Response Time
SaaS	Service as a Service
SLAs	Service Level Agreements
TDCV	Trial Data Center Vector
Thr	Throughput
TotC	Total Cost
TV	Trial Vector
UB	User Base
UR	User Request
VM	Virtual Machine
VMC	Virtual Machine Cost

VMM	Virtual Machine Monitor
VPN	Virtual Private Network

POLISI PEMILIHAN BROKER PERKHIDMATAN AWAN YANG DIPERTINGKATKAN BERDASARKAN ALGORITMA EVOLUSI KEBEZAAN

ABSTRAK

Pengkomputeran awan (CC) merupakan satu teknologi yang sedang berkembang pesat dalam sektor Teknologi Maklumat dan akan menjadi satu bidang penyelidikan yang penting. CC membenarkan penyedia perkhidmatan awan untuk menyediakan perkhidmatan awan yang pelbagai kepada pengguna yang berbeza. Penyedia perkhidmatan tersebut bersedia memenuhi permintaan pengguna dengan tawaran kos terendah dan masa respons yang minimum, yang sangat bergantung pada pusat data (DC) yang bertanggung jawab untuk melaksanakan permintaan mereka. Pemilihan DC dalam lingkungan CC adalah tanggungjawab broker perkhidmatan awan (CSB) yang beroperasi sesuai dengan polisi penghalaan terbina dalam. Pemilihan DC yang silap boleh menyebabkan cerutan dan kesesakan dalam seni bina broker perkhidmatan, mengakibatkan ketirisan sumber dari segi masa dan kos. Walaupun telah wujud banyak polisi Broker Perkhidmatan Awan, namun masih terdapat keperluan kepada satu polisi yang cekap bagi memastikan tahap QoS yang tinggi dari segi jumlah kos, masa tindak balas, dan masa pemprosesan. Oleh itu, tesis ini berisrat untuk mencadangkan satu polisi Broker Perkhidmatan Awan yang cekap, yang dikenali sebagai Polisi Broker Perkhidmatan Awan dengan Kos Terendah dan Prestasi Tertinggi (HPLC), untuk memilih pusat data yang paling sesuai berdasarkan algoritma metaheuristik. Polisi HPLC ini menyesuaikan algoritma Evolusi Kebezaan sebagai teknik pengoptimuman untuk mencari Pusat Data yang optimum untuk melaksanakan permintaan pengguna. Oleh kerana keberkesanannya dalam menyelesaikan masalah sebenar dalam jangka masa yang munasabah, polisi HPLC mengadaptasi algoritma metaheuristik (iaitu Evolusi Kebezaan) sebagai teknik pengoptimuman untuk mencari pusat data yang optimum untuk melaksanakan permintaan pengguna. Polisi HPLC terdiri daripada empat peringkat utama, iaitu: (i) Perumusan asas pengguna, (ii) Pemilihan broker perkhidmatan awan (CSB) berasaskan peraturan, (iii) pengiraan parameter pusat data, (iv) pengoptimuman parameter pusat data. Polisi yang dicadangkan telah dinilai menggunakan enam senario simulasi, di mana setiap senario merangkumi kes-kes yang berbeza daripada pengagihan pusat data dan pangkalan pengguna di antara rantau-rantau dalam persekitaran Pengkomputeran Awan. Di samping itu, keberkesanan polisi HPLC yang dicadangkan dibandingkan dengan polisi-polisi Broker Perkhidmatan Awan yang sedia ada. Keputusan membuktikan bahawa dasar HPLC yang dicadangkan mengatasi polisi-polisi CSB sedia ada yang lain. Dari segi purata masa pemprosesan, polisi HPLC yang polisi pusat data terdekat (CDCP) sebanyak 48.1%, Broker Perkhidmatan Awan Boleh-Susun Semula Dinamik (DRCSB) sebanyak 84.8%, Polisi Masa Tindakbalas Teroptimum (ORTP) sebanyak 47.2%, dan Polisi Penghalaan Broker Perkhidmatan Teroptimum (OSBRP) 37%. Selain itu, dari segi purata masa tindak balas pemprosesan, polisi HPLC yang dicadangkan mengatasi CDCP, DRCSB, ORTP, dan OSBRP sebanyak 50.5%, 54.1%, 50.4% dan 40%, masing-masing. Selanjutnya, dari segi purata jumlah kos, polisi HPLC yang dicadangkan mengatasi CDCP, DRCSB, ORTP, dan OSBRP masing-masing sebanyak 63.4%, 91.7%, 63.4% dan 56.1%.

ENHANCED CLOUD SERVICE BROKER SELECTION POLICY BASED ON DIFFERENTIAL EVOLUTION ALGORITHM

ABSTRACT

Cloud computing (CC) is speedily an emerging technology in the Information Technology sector and is becoming a pivotal research area. CC permits cloud service providers to provide different types of cloud services to different users. Cloud users are willing to meet their requests demands at the lowest cost and minimum processing time and response time, which is highly dependent on the data center (DC) that is responsible for executing the requests. The selection of a DC in the CC environment is the responsibility of the Cloud Service Broker (CSB) that operates in accordance with a built-in routing policy. Improper DC selection might cause a bottleneck and congestion in the service brokerage architecture, resulting in resource leakage in terms of time and cost. Despite that there are many Cloud Service Broker policies proposed in the literature, still, there is a need to propose an efficient policy to ensure a high level of quality-of-service in terms of total cost, response time and processing time. Therefore, this thesis intends to propose an efficient CSB policy, called Highest Performance and Lowest Cost Cloud (HPLC) Service Broker Policy, to select the most appropriate DCs based on a metaheuristic algorithm. Due to its effectiveness in solving real problems in a reasonable amount of time, HPLC policy adapts a metaheuristic algorithm (i.e. Differential Evolution) as an optimization technique to find the optimal data center to execute the user requests. HPLC policy consists of four main stages, namely: (i) User base formulation, (ii) Condition-based Cloud Service Broker selection, (iii) Data centers parameters calculation, (iv) Data centers parameters optimization. The proposed policy was evaluated using six simulation scenarios, where each scenario covers different cases of the distribution of data centers and user bases among the regions in the Cloud Computing environment. In addition, the effectiveness of the proposed HPLC policy was compared with well-known existing Cloud Service Broker policies. Results prove that the proposed HPLC policy outperforms other existing CSB policies. In terms of average processing time, the proposed HPLC policy enhances Closest Data Center policy (CDCP) by 48.1%, Dynamically Reconfigurable Cloud Service Broker (DRCSB) by 84.8%, Optimized Response Time policy (ORTP) by 47.2%, and Optimized Service Broker Routing Policy (OSBRP) by 37%. Moreover, in terms of average response time, the proposed HPLC policy enhances CDCP, DRCSB, ORTP, and OSPRB by 50.5%, 54.1%, 50.4%, and 40%, respectively. Furthermore, in terms of average total cost, the proposed HPLC policy enhances CDCP, DRCSB, ORTP, and OSBRP by 63.4%, 91.7%, 63.4%, and 56.1%, respectively.

CHAPTER 1

INTRODUCTION

1.1 Overview

During the last 30 years, computer science has demonstrated itself as a fundamental part of all industries. The rapid development of computer science started in the 80s, starting with the personal computing industry (i.e. Mac), up to the mid-90s with the revolution of the Internet. Mainly, users have used the Internet to perform regular tasks such as shopping, sending and receiving emails, and fast information access (Bera *et al*., 2015). Internet technologies have improved and advanced to supply users with more than the regular known services such as social networks, Cloud Computing (CC), and Internet of Things, which provide Internet users with an easy and instant means for sharing knowledge, and for allowing them to use resources and services from any place at any time, while services provided by CC is one of the most commonly used services by Internet users nowadays.

CC holds several computing resources, which commonly comprises a huge number of connected devices (i.e. computers) connected together by a network. CC is considered a synonym of computing, which is distributed resources over a network, and it indicates the capability to operate applications via several linked computers simultaneously (Nandwani *et al*., 2015).

In fact, using available computing resources is the core advantage provided by CC, as the main objective of CC is to provide huge pool of connected resources and share it between different users dynamically using virtualization techniques. However, the way of how the Cloud resources have managed is invisible to users who use or access theses resources including: shared storage capabilities, processing power, and applications.

CC technology is considered one of the leading technologies for the time being. It provides businesses and users with remote software services and computing resources. Instead of having local software/hardware services and resources, CC allows users to make use of these services and resources from the Cloud Service Providers (CSPs) companies, such as Microsoft, Amazon, Google, etc. (Shawish & Salama, 2014). CC attracts several leading vendors in the Information Technology (IT) sector, such as Amazon, Microsoft, Google, IBM, etc., to invest more in this emerging technology to provide the Cloud services. Figure 1.1 illustrates the top five Cloud vendors by revenue (in Billion U.S. Dollars) in 2019 (Cloudwars, 2019).

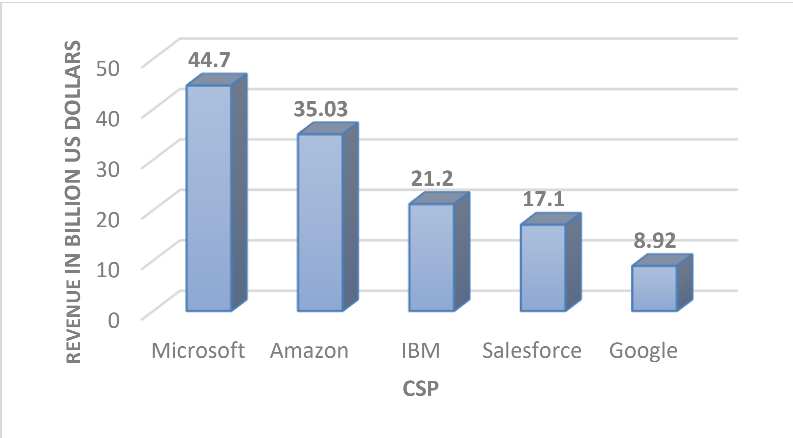

Figure 1.1 The Top Leading Cloud Vendors (Cloudwars, 2019)

Several CSPs, such as Microsoft, Amazon, Microsoft and Google, provide multiple geo-distributed Data Centers (DCs) distributed around the world, in order to ensure providing Cloud services with high level of Quality-of-Service (QoS). Figure 1.2 shows an example of geo-distributed Cloud DCs provided by Amazon Elastic Compute Cloud (EC2).

Figure 1.2 Geo -distribution of DCs by Amazon EC2 (Zhou et al., 2017)

Therefore, the optimal DC that provides the service with high level of QoS should be selected. Indeed, the selection of the best DC that provides Cloud service is the responsibility of the Cloud Service Broker (CSB), which in turn uses selection policy to determine which DC is optimal one to provide that service.

As a result, the QoS requirements depend on DC that is selected by the CSB through its selection policy to execute user request, while user request is a service request that initiated form user who looks for Cloud applications hosted by CSP, for example, a user looks for using Dropbox to store some private content (i.e. SaaS request).

1.2 Background

This section provides an overview of CC technology, the CSB, the Cloud DC, and the QoS requirements related to the Cloud environment.

1.2.1 Cloud Computing

The concept of CC come up in 1960s by John McCarthy, as he said "*computation may someday be organized as a public utility*" (Wheeler & Waggener, 2009). In 1966, Douglas Parkhill investigated the features of CC were investigated first time in his book “*The challenge of the Computer Utility*” (Tajammul & Parveen, 2020). Initially, history of “Cloud” as a term was originated from telecommunications world, as telecom organizations started providing Virtual Private Network (VPN) services along with comparable QoS at a lowest price. Before invention of VPN, dedicated point-to-point data circuits were used with wastage of network bandwidth, but by using VPN services, network traffics can be switched to balance utilization of overall network. However, nowadays, CC extends this to cover servers and network infrastructure. As different vendors in industries have moved into CC and implemented it. For instance Amazon played a vital role and launched the Amazon Web Service (AWS) in 2006.

With the turn of the 21st century, CC technology began to be used widely used. As shown in Figure 1.3, there are many new computing paradigms have been emerged with the emergence of technological advances such as multi-core processors and networked computing environments, to edge closer toward achieving the grand vision of computing. These new computing paradigms are cluster computing, Grid computing, P2P computing, service computing, market oriented computing, and most recently Cloud computing (Padhy & Patra, 2012).

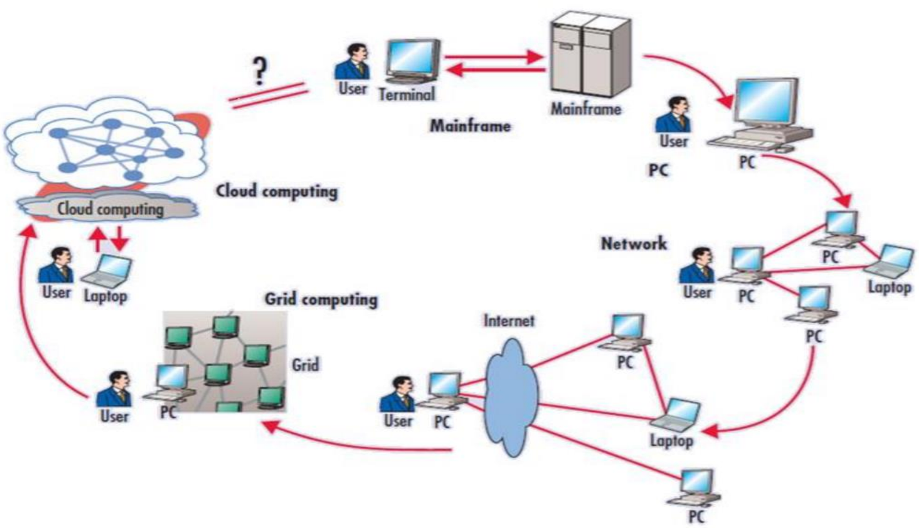

Figure 1.3 Evolution of CC (Padhy & Patra, 2012)

As aforementioned, CC technology is considered modern trend in the IT sector that takes the available computing resources and data away from desktop and portable devices into huge data repositories, which called data centers (Boru *et al*., 2015). It works with storage services, software, computation devices, and data access without the need from the user to know where the physical location of these resources is, and what are the configurations of the system required to provide such services (Shen *et al*., 2014). Usually, it relates to network-based services provided by real server hardware (Nandwani *et al*., 2015).

From high level of abstraction, there are four main actors in the Cloud environment interact with each other in different manners to satisfy the main goal of CC, these actors are shown below in Figure 1.4. First, a cloud consumer who is a user who can use/access the services provided by a CSP. Second a CSP, which is an organization that owns the services and provides these services to the interested users. Third, a CSB is an intermediate entity between cloud consumer and CSP, in other words, it manages the interactive relationship between cloud consumer and CSP by offering services either by an extended form or offering the service directly from the CSP to the consumer. Last, a Cloud carrier is also an intermediary actor, which is responsible for transporting of CC services from CSPs to Cloud consumers, either via a CSB or directly to the user using the network connection (Vyomtech, 2013). The role of CSB and Cloud carrier is almost similar, but the main difference is that the CSB can modify the service before delivering it to the Cloud consumers while the carrier cannot.

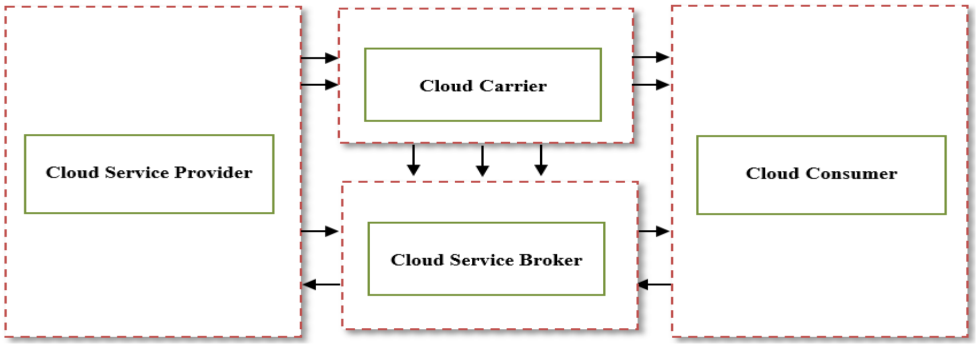

Figure 1.4 Interactions between CC's Actors (Vyomtech, 2013)

1.2.2 Cloud Service Broker (CSB)

A CSB routes the users requests to the most appropriate DCs located at different regions around the world (Chen *et al.*, 2016; Khurana & Bawa, 2017). Then, after the selection of the most appropriate DCs, these requests are executed by specific Virtual Machines (VMs) selected by the load balancer which is located in the selected DC (Xu & Li, 2012). After that, the response is redirected back to the user who is looking to get a service with high level of QoS. Since users satisfaction is measured by level of QoS that depends on the selected DC and its capabilities to provide services efficiently. Figure 1.5 depicts the process of the selection of DC through CSB policy.

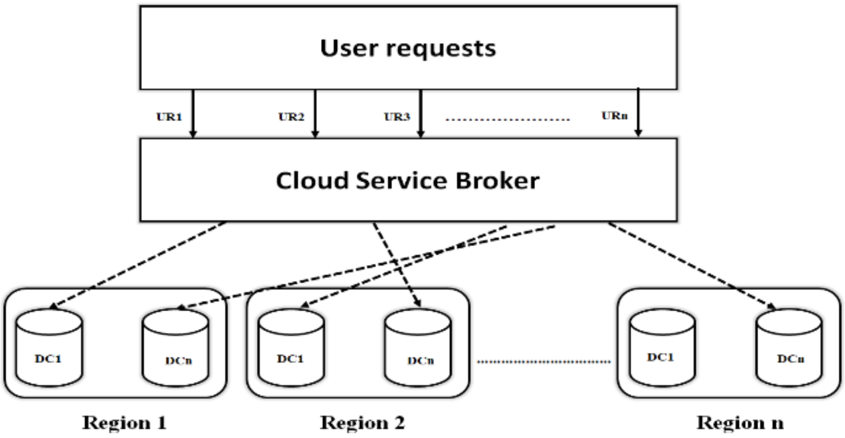

Figure 1.5 Cloud DC Selection Process

As examples, CloudSwitch is one of the common CSBs which was established in 2008 by Amazon EC2 (Elijorde & Lee, 2015). RightScale is also another well-known CSB that provides management platforms to facilitate the deployment of applications across different cloud platforms (Bernstein, 2014). For the time being, Dell and VMWare are interested in cloud services brokerage, and they have collaborated to provide brokerage infrastructure (Taylor, *et al*., 2010).

Actually, CSB can be categorized into three main categorizes, namely (Gartner, 2009; Liu *et al*., 2011; Chen, *et al*., 2016): (i) intermediator CSB which enhances the capabilities of existing service before delivering it to the user, (ii) aggregator CSB which combines services from more than one CSPs to provide a new value-added service, (iii) arbitrator CSB which combines services dynamically from more than one CSP based on user requirements. Despite CSB is many types, the main goal of CSB still to select the best DC to execute the user request.

1.2.3 Cloud Data Center (DC)

The data center is "*a set of servers, storage and network devices, power systems, and cooling systems*" (Abts & Felderman, 2012). DC contains several components such as processing capabilities, power supply, storage, security devices, etc., which are essentially required to achieve different functionalities. Traditional DC is a physical server where all servers are located in, but the Cloud DC is a set of shared computing resources with high level of QoS at a lower total cost of ownership (Qi *et al*., 2014). Table 1.1 shows the differences between traditional DC and Cloud DC. However, the main difference between traditional DC and Cloud DC is virtualization that provides enormous scalability, virtualized computing resources, and on-demand utility computing.

Table 1.1 Comparison between Traditional DC and Cloud DC (Qi et al., 2014; Makhija, 2019)

	Traditional DC	**Cloud DC**
Servers	Co-located Dependent Failure	Integrated Fault-Tolerant
Resources	Partitioned Performance Interrelated	Unified Performance Isolated
Management	Separated Manual	Centralized Full Control With automation
Scheduling	Plan ahead Overprovisioning	Flexible Scalable
Renting	Per Physical Machines	Per Logical Usage
Services and Applications	Fixes on designed Servers. Large number of small sized applications.	Runs across all VMS. Smaller number of very large applications.
Workload Installation	Complex	Simple
Hardware environment	Mixed	Homogenous
Management tools	Multiple	Standardized
Ownership	Servers and software are owned by users, but vendors own infrastructure	Everything is owned by CSP

In fact, DC virtualization is a precious opportunity for IT. It saves the cost to a remarkable extent through efficient sharing of the available servers, storage and network capabilities, which are translated into lowest cost of purchasing and operating. At DC virtualization level, more applications and services have provided compatible manner, and fast implementation with high level of QoS. Now, with the prospect of CC technology, DC virtualization is used as a springboard to access Cloud services provided by third party CSP to build private CC platform. However, as shown below in Figure 1.6, the Cloud DC evolution is started with standardized hardware application silos, then virtualization technology is started at the beginning of the 20^{th} Century. Now, virtual DCs and Cloud DCs are available to be used along with different types of CC (i.e. private, public, or hybrid).

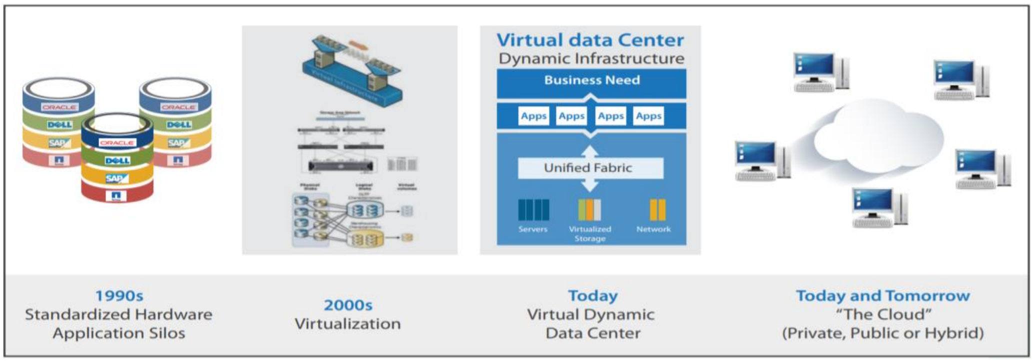

Figure 1.6 DC Evolution (Jagadeesh, n.d.)

1.2.4 Quality of Service in Cloud Computing

Due to the increasing demand of services provided by CC, the QoS becomes an increasingly important concern for the Cloud users, since there are many open issues should be resolved especially those related to performance, cost and availability (Jelassi *et al*., 2017; Odun-Ayo *et al*., 2018). QoS as a term can be defined as "*the level of performance, reliability, availability offered by an application and by the platform or infrastructure that hosts it*" (Akpan & Vadhanam, 2015).

However, the method for providing the Cloud services must be beneficial for both the Cloud users and the CSPs, since the CSP should use the best hardware and software configurations efficiently to provide a Cloud service with a desired level of QoS for users while preserving efficient resource utilization simultaneously (Barba-Jimenez *et al*., 2016). Hence, QoS presents the non-functional requirements of the Cloud services. It covers five main requirements including service/execution cost, execution time, reliability, availability, and reputation (Gupta *et al*., 2015). Besides, service cost (i.e. total cost) is the required cost by the virtual machine to execute user request plus cost required to transfer service or data to users (Khurana & Bawa, 2017).

Execution time indicates processing time and response time, where response time is "*the total time between receiving user request asking for specific service to the time of delivering the service to that user*", but, the processing time is "*the time* *required by the server (i.e. DC) to execute the user request*" (Singh, 2014). Cloud reliability relates to "*how the cloud is available to offer its services even if some of its components fail*" (Kaur & Kumar, 2015). On the other hand, availability can be defined as "*to the uptime of a system, a network of systems, hardware and software that collectively provide a service during its usage*" (Ahuja & Mani, 2012), where availability of Cloud DC is considered in this thesis as a composite value of delay and bandwidth.

Whilst reputation is defined as "*is the aggregated opinion of a community towards that entity, which is typically represented by a comprehensive score reflecting the overall opinion, or a small number of scores on several major aspects of performance*" (Huang & Nicol, 2013).

As aforementioned, Cloud users always seek to select the Cloud service from the CSPs with the high level of QoS (Karakus & Durresi, 2017). Consequently, services provided by CC have attracted businesses, but commercial offerings should deliver the expected level of QoS to users. If the provided Cloud services do not meet user requirements, those users can look up for alternative CSPs. The ability to determine and achieve the QoS is an important issue for Cloud users and CSPs alike (Karakus & Durresi, 2017). QoS provides a guarantee of service availability and high level of performance and offer level of proof that the Cloud users requirements have precisely provided (Batista *et al*., 2017). Hence, QoS is directly associated with Cloud users and services provided by CSPs, which ensures efficient resource utilization using CSB policies and load balancers.

1.3 Research Motivation

For the time being, the need of Cloud DC is rapidly growing due to its importance of delivering IT services easily and providing processing and storage capabilities, and networking to the users who look for services from different locations around the world. Besides, the increased need for business agility and cost saving with high level of QoS has led to the rise and growth of cloud DCs usage over the traditional ones. Figure 1.7 shows the increased distribution of workload and computing instances for Cloud DCs over traditional DC from 2016 to 2021 (Cisco, 2018a).

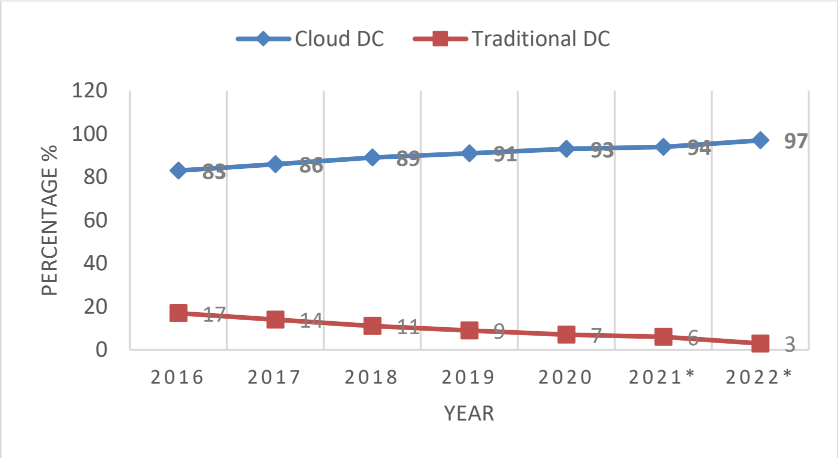

Figure 1.7 DC Workload and Compute Instance Distribution (Cisco, 2018a)

In addition, services provided by CC have increased speedily in terms of numbers and scales across different application areas. A recent report presented by Cisco (2018b) shows that the amount of DCs traffic was 7.7 Zettabyte in 2017, and it reached 9.8 Zettabyte in 2018, whereas DCs traffic is expected to reach 11.9 Zettabyte and 14.1 in 2019 and 2020 respectively, as shown below in Figure 1.8.

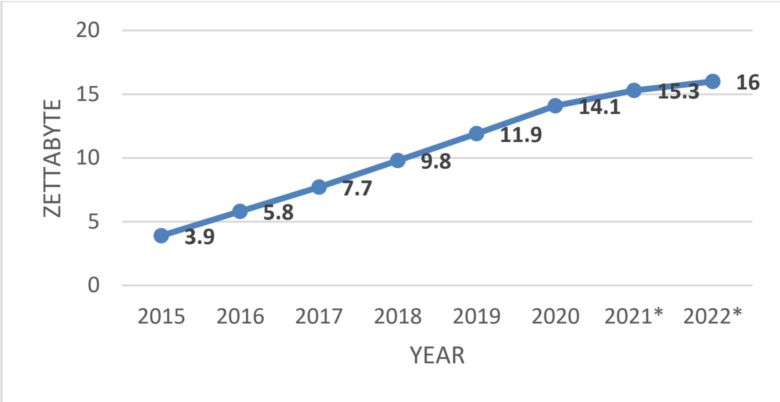

Figure 1.8 DCs Traffic from 2015-2020 (Cisco, 2018b)

With increasing volume of Cloud services generated and data volume stored across geo-distributed DCs, how to route the users requests in a manner that ensures efficient resource utilization and high level of QoS requirements becomes an emerging topic. Since the location of DC has a direct influence on the level of QoS requirements provided by the Cloud environment, a proper DC must be selected to execute users requests with high level of QoS. Consequently, an appropriate DCs selection provides efficient resources usage, and it reduces the processing and response time, providing scalability and averting deadlock (Naha & Othman, 2016). Besides, the increasing demand of the Cloud services makes the Cloud users aware of having services with high level of QoS requirements, which however, raises a new challenge of how to select the best DC from a huge set of DCs distributed among different regions around the world, in an efficient manner based on the users predefined needs.

1.4 Research Problem

Decision-makers and cloud users would like to get high benefits from the advantages of the CC such as low-cost infrastructure, low maintenance, and high availability (Rjoub *et al.*, 2020). However, there are many considerations when they decide to move from traditional computing to the CC. These considerations might be related to security-related concerns, reliability, performance, and execution cost.

Enhancing the overall performance and the execution cost is urgent and a vital element for CC success, which raises cloud reliability and availability. The performance of the CC environment is extremely dependent on resource availability, for instance, network latency, bandwidth, throughput, data transfer rate, and so forth (Bulla & Rao, 2019; Tuli *et al*., 2019). Network latency, throughput, and bandwidth are very important factors in the success of CC. All the available computational resources deployed on the CC need two-ways data transfer. Besides, network bandwidth and throughput become more important when big data are involved in the CC environment, where high bandwidth and throughput are required.

Also, this is completely true when talking about big enterprises and individuals as well. In the near past, cloud storage with 1 Gigabyte was enough for storing back up of mobile data, but nowadays cloud storage with 10 Gigabyte is too small to store in mobile pictures; which imposes additional network activities. Consequently, transferring a large volume of data back and forth over a slow communication network might cause serious deficiency concerns, because of the time required to transfer data (Deng *et al*., 2019).

On the other hand, the computational time (i.e. processing and response time) is also very important in fulfilling the requirements of users and CSPs as well, since it reflects efficient resource utilization (i.e. DCs utilization) and users' satisfaction (Bhandari & Gupta, 2019; Kumar & Kumar, 2019). Also, CSPs look for the same goals as the users in terms of efficient DCs utilization. Users would like to get the required cloud services with a minimum cost and with a reasonable response and processing time. CSB policies play a crucial role in the selection of the most appropriate DC to execute users requests (Al-Tarawneh & Al-Mousa, 2019). The CSB is considered the first point of contact for cloud users whenever they request tasks from the CC. After the CSB receives the user request, it decides to route the request to the most appropriate DC.

As aforementioned, the cloud DC is a logical set of multiple physical servers or processing units (i.e. servers). Besides, a potential bottleneck in the CC might cause additional serious performance degradations. Because of the strategic location of the CSB and its role in the CC, it may be a potential bottleneck by the selection of improper DCs for users requests. The CSB must be able to select the appropriate DC that achieves the users requirements.

Indeed, proposing an efficient CSB policy grabs the researchers attention to address the issues the previous issues and concerns. The existing CSB policies might be commonly categorized based on users QoS needs into: (i) CSB policy based on enhancing response and processing time, (ii) CSB policy based on enhancing DC availability, and (iii) CSB policy based on enhancing total cost.

The CSB policies based on enhancing processing and response time such as (Jaikar & Noh, 2015; Radi, 2015; Mahalle *et al.*, 2015; Naha & Othman, 2016; Dubey & Jain, 2016; Arya & Dave, 2017; Sheikhani *et al.*, 2017; Islam & Waheed, 2017; Manasrah & Gupta, 2019) face several challenges because a majority of them ignore DC availability and efficiency parameters, and DC selection assumptions do not take the size of incoming user requests into consideration (i.e. the DC selection assumption was built based on the previously executed users requests); thus inaccurate DC selection might occur. Therefore, there are still gaps in these considerations because the overall performance of the CC environment might be negatively impacted.

Meanwhile, the proposed CSB policies based on enhancing availability such as (Sharma, 2014; Kapgate, 2014a; Lee *et al.*, 2014; Naha & Othman, 2016; Manasrah *et al.*, 2017) increase the availability of DCs and slightly enhance processing time and response time. In spite of their advantages, such policies do not consider the dynamic nature of the size of users requests (i.e. different users requests may have different sizes). Besides, the size of the incoming users requests was not taken into consideration when building the DC selection assumptions, and the total cost is not as efficient as required.

Moreover, several CSB policies have been proposed to decrease the total cost borne by DCs to execute users requests such as (Sharma *et al*., 2013; Cardellini *et al*., 2013; Kapgate, 2014b; Kumar & Parthiban, 2016; Benlalia *et al*., 2019). In fact, these policies provide considerable results in terms of the total cost, but the processing and response time require further improvements, and the number of executed users requests are not always as required (i.e. throughput is relatively low) due to the high response and processing time consumed by DCs to execute users requests. Furthermore, these CSB policies have concentrated only on enhancing the performance in terms of total cost and gaps still exist in these CSB policies in terms of ignoring important parameters when the selection of DCs, which will be considered in this research.

Besides, most of the CSB policies proposed in the literature are based on a single (simple or complex) optimization objective, while no one has used a multi-objective optimization in the DC selection process. Therefore, a multi-objective might enhance DC selection since it has more than one objective to be achieved simultaneously. In other words, a multi-objective optimization does not focus on one objective at one time. For instance, some users look for the services with the lowest cost, while others look for services with a high level of QoS. This is acceptable but it will not be efficient in the future, especially with the expanding number of users requests and connected Cloud DCs. Consequently, implementing a multi-objective problem-based DC selection technique is highly required, and the use of CSB policies based on metaheuristic to provide an efficient DC selection is also required. Metaheuristic is used due to its efficiency in solving real problems in a reasonable amount of time, simplicity, robustness, and fastness.

To summarize, this research pertains to the improper DC selection in the CC environment due to the following reasons:

- Processing time and response time and/or a high total cost still requires more enhancements during the execution of user requests.
- To ensure user requets execution adaptively, the size of incoming users request should be considered when building the DC selection assumptions; thus, inaccurate estimation of the processing time is avoided.
- The selection of the most appropriate DC efficiently needs more important DCs parameters to be considered during the decision of DC selection.
- There is a lack of using a multi-objective optimization in terms of Cloud users QoS needs, rather few CSB policies only uses a single (simple or complex) optimization objective.

1.5 Research Objectives

The main goal of this thesis is to propose a multicriteria decision-based CSB policy to efficiently route users requests in Cloud Computing environment. In specific, the following objectives are formulated to achieve the main goal of this thesis:

1) To propose set of parameters to be contributed in the selection of the most appropriate DC to execute users requests efficiently.

2) To propose a multi-objective function that ensures providing high level of Cloud users QoS requirements by considering all of the previously proposed DCs parameters simultaneously.
3) To optimize the proposed multi-objective function using a metaheuristic algorithm.
4) To propose a hybrid-based CSB framework.
5) To evaluate the performance of the proposed policy in terms of its processing time, response time and total cost.

1.6 Research Scope

This thesis is to propose a CSB policy to select the most appropriate DC to execute the incoming users requests efficiently. The proposed CSB policy is limited to the selection of the DCs on simulation environment with different simulation scenarios adapted from Wickremasinghe *et al*. (2010), Manasrah *et al*., (2017), Kofahi *et al*., (2019), and Al-Tarawneh & Al-Mousa (2019).

In this thesis, some of real features of the Cloud environment and Cloud DC such as service level agreement, fault tolerance mechanism, security mechanism, and cooling techniques, etc., are out of scope. Besides, real cloud users satisfaction is out of scope since it cannot be measured in simulation environment. In addition, because of it is very complex and time-consuming to perform repeatable experiments in large-scale environments such as the CC environment, the CloudAnalyst simulator, which reflects the real CC environment, will be used to conduct simulation in this thesis to guarantee the repeatability of experiments. Also, dynamic arrival nature of users requests is also out of scope. However, Table 1.2 summarizes the research scope and limitations.

Table 1.2 Research Scope and Limitations

No	Item	Scope of Research
1.	**Domain**	Cloud Computing Environment
2.	**Policy Level**	Cloud Service Broker Policy
3.	**Policy Goal**	Appropriate Data Center selection
4.	**Fitness Function**	Multi-objective function (maximizing DC efficiency and availability, and minimizing expected total cost and expected processing time)
5.	**Simulated Dataset**	Different simulation scenarios
6.	**Evaluation Metrics**	Average response time, average processing time, and average total cost
7.	**Testing environment**	Simulator

1.7 Research Contributions

The key role of CSB was discussed in the previous sections. Previous studies have proposed various CSB policies to ensure efficient routing of users requests to the most appropriate Cloud DC. However, these policies are generally suffering from some issues due to their designs as explained in detail in Section 1.4. This thesis contributes to the literature on service provisioning and CSB polices by highlighting the significance of using multiobjective optimization in the selection of the Cloud DC to execute users requets. There is a unique contribution of this thesis in that it incorporates more than one QoS (i.e. users interests) simultaneously when executing users requets. Besides, this thesis looks at human level at resource allocation by providing an adaptive resource allocation policy. The contributions of this thesis are summarized as follows:

- Incorporating a set of QoS requirements simultaneously during DC selection in CC environment.
- Highlighting the crucial role of the specifications of the Cloud infrastructure in delivering SaaS for users efficiently.

- Introducing a multicriteria decision-based policy to ensure the selection of the most appropriate DC selection in CC environment.
- Proposing a metaheuristics-based CSB policy.
- Proposing a hybrid-based CSB framework.

Mapping among research challenges, objectives and contributions are shown below in Table 1.3.

Table 1.3 Mapping among Research Gaps, Objectives and Contributions

Problem Gap(s)	Research Objective(s)	Research Contribution(s)
Executing the users requests with high computation time.	• Objective # 1 • Objective # 2 • Objective # 5	• Contribution # 1 • Contribution # 2
Important DC parameters such as (DC efficiency and DC processing power) are not contributing in the selection of DCs efficiently.	• Objective # 1 • Objective # 2	• Contribution # 1 • Contribution # 2 • Contribution # 3
Size of incoming user request is not considered to build accurate assumption for DC selection.	• Objective # 1 • Objective # 2 • Objective # 4	• Contribution # 1 • Contribution # 2 • Contribution # 5
Lacking of adoption of CSB policy based on a multi-objective optimization in terms of Cloud users QoS needs.	• Objective # 2 • Objective # 3	• Contribution # 3 • Contribution # 4

1.8 Research Steps

This thesis proposes a new CSB Policy based on a metaheuristic algorithm to select the most appropriate DC with minimum processing and response time, and lowest total cost. Figure 1.9 summarizes the steps followed by this thesis to satisfy the objectives. These steps include the followings:

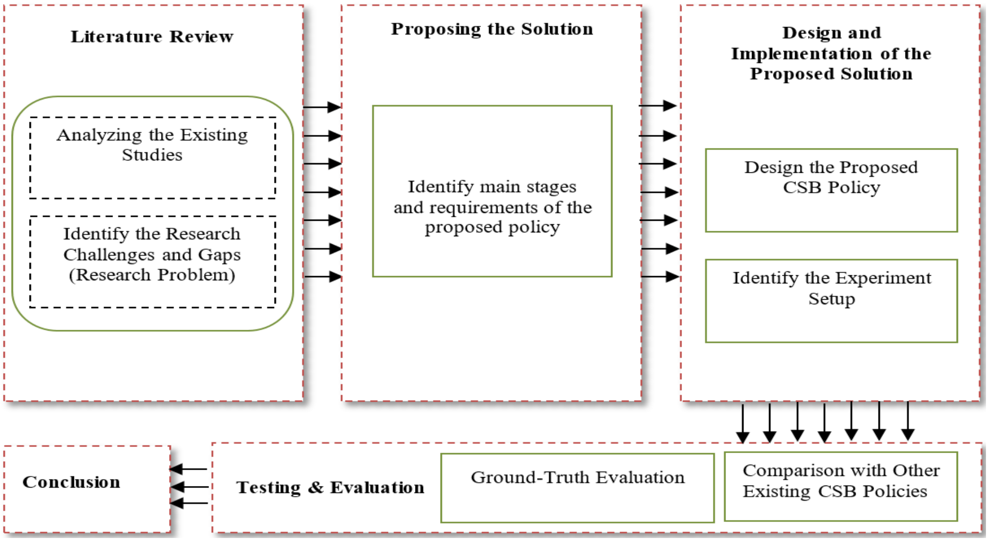

Figure 1.9 Research Steps

The first step is exploring and analyzing the literature review. It reviews the background of Cloud Computing, its simulators, and metaheuristic algorithms used in CC environment. Besides, it provides analyzes the existing CSB policies and highlights the limitations and gaps of these polices. Then, the research problem is identified based on these limitations and gaps.

The second step is proposing the solution. In this step, the proposed solution and its stage have identified to achieve the research objectives. The details and requirements of each stage have discussed.

The third step is design and implementation. In this step, detailed design of the proposed CSB policy is presented. In details, description of the cloud simulator (i.e. CloudAnalyst), the simulation configurations, and the design of each stage of the proposed policy, simulation scenarios, and evaluation metrics are presented.

The fourth step is testing and evaluation. In this stage, the proposed solution is evaluated based on evaluation strategy dived into ground-truth evaluation and comparison with other existing solutions based on predefined evaluation metrics.

The fifth step is the conclusion. In this step, the present research contributions and limitations, research future works and recommendations have presented.

1.9 Thesis Organization

This thesis comprises of six main chapters as follows:

Chapter 2 presents the research issues tackled in this thesis from a high-level point of view. In details, this chapter presents background, basic concepts and analysis of prior research works used related to this thesis, including an introduction to the Cloud technology, load balancing algorithm used in the Cloud environment, the Cloud simulators, metaheuristic algorithms used in the Cloud field, cloud service broker and anabasis of existing CSB policies.

Chapter 3 presents a detailed research methodology. It shows how the methodology phases will be integrated with each other to achieve the research objectives.

Chapter 4 discusses the design, simulation and implementation environment used for the proposed policy. This chapter presents the design and implementation of each phase of the proposed policy in details. Besides, simulation scenarios, and evaluation metrics used in this thesis are presented in details.

Chapter 5 analyzes the experiments and their findings. Also, it presents a comprehensive analysis of simulation results obtained using the proposed CSB policy. Furthermore, the performance of the proposed policy is evaluated in comparison with existing CSB policies in this chapter.

Chapter 6 concludes the research and presents future research directions.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter presents a literature review related to the proposed study. It is organized as follows: Section 2.2 provides research background, which is divided into four main subsections: First, Subsection 2.2.1 provides introduction to CC, second, Subsection 2.2.2 presents some of widespread load balancing polices used in the Cloud environment. Third, Subsection 2.2.3 introduces a comprehensive review of the Cloud Simulators, and last, Subsection 2.2.4 reviews the most commonly used metaheuristic algorithms in the Cloud environment

In addition, Section 2.3 provides review of related works conducted in the prior literature. Summary of this chapter is presented in Section 2.4.

2.2 Background

Cloud Computing is a rich research environment due to the fact that CC has many challenges need to be addressed to enhance the performance of the overall performance and QoS provided by the Cloud environment (Lakshminarayanan *et al.*, 2013). Basically, researchers, organizations and educational institutions used DC, a large group of connected servers devices, and distributed storage devices to build testing environment for CC (Greenberg *et al.*, 2011). However, different challenges may be resulted, such as no scaling, no resource virtualization capabilities, finite available computing resources, and high cost (Puthal *et al.*, 2015).

The regular rules in CC are not always apparent during the execution of the Cloud applications because these applications are not like grid applications. As the Grid applications require an intensive computation that requires intensive workload on the available computation resources such as using Map Reduce in the Cloud environment is input/output intensive distributed software. However, the following subsections discuss an introduction of CC, load balancing policies, cloud simulators, and metaheuristic algorithms used in the Cloud environment

2.2.1 Introduction to Cloud Computing

The term of Cloud Computing is popular for its Pay-Per-Usage basis. In order to catch up with the significant advancements and the widespread use of the Internet, the IT industry is produced an emerging paradigm named "*Cloud Computing*", which is a buzzword technology in both academia and industry nowadays. The widespread definition of the term is the one that is established by National Institute of Standards and Technology, which states that "*Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*" (Badger *et al.*, 2012). CC was initially coined in 1997 by Dr. Chellapa, in Texas University, in a talk on "*New Computing Paradigm*", CC is developed through different versions, including Grid Computing, Utility Computing, and application service provider, etc. (Kerridge, n.d.).

According to (Villegas *et al*., 2012; Rashmi & Basu, 2016), there are three main service models of the Cloud, namely: (i) Infrastructure as a Service (IaaS) which aims to make it easier for users to receive services from powerful and updated computer infrastructures via the Internet (Wu *et al*., 2015), (ii) Software as a Service (SaaS) which is a sort of mode of software that is administered by the Internet, where users can select a Web-based application without enduringly buying the whole software package of that model, and (iii) Platform as a Service (PaaS) which provides platforms allowing users to promote and operate the services without the need of constructing or maintaining the infrastructure on their personal devices (Kushwaha & Gupta, 2015). Figure 2.1 summarizes the Cloud service models, and it provides examples of how the Cloud clients can access this model via Web browser, Mobile applications, Terminal Emulator, etc. (Kansal & Chana, 2012).

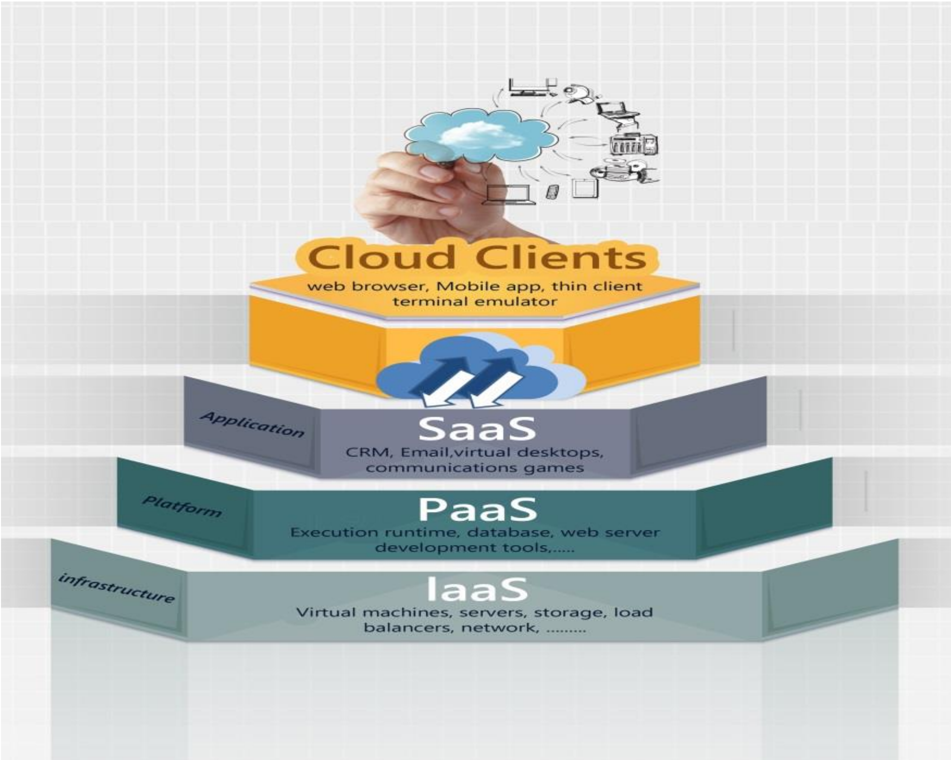

Figure 2.1 Service Models of CC

Indeed, similar to well-defined IT environment, CC was developed for providing remote access for IT resources, and for offering scalable resources. However, the "*Cloud*" is a new brand of the Internet technology based on decentralized IT resources (Sobhanayak *et al.*, 2015). The services have accessed in complicated business process managed by Service Level Agreement (SLA) (Jain *et al.*, 2017; Wagle, 2015). In fact, SLA might be violated because of dynamic elements, hardware, and software malfunctions and workloads. During the interaction between regular