ENHANCE MODEL FOR SELECTING CLOUD SERVICE PROVIDER USING DECISION MAKING ALGORITHM

YAZEED ALSSAYEED ALI AL MOAIAD

DOCTOR OF PHILOSOPHY (PHD) IN INFORMATION AND COMMUNICATIONS TECHNOLOGY

FACULTY OF COMPUTER & INFO. TECH
AL-MADINAH INTERNATIONAL UNIVERSITY

2017/1439H

ENHANCED MODEL FOR SELECTING CLOUD SERVICE PROVIDER USING DECISION MAKING ALGORITHM

YAZEED ALSSAYEED ALI AL MOAIAD

PIT141BG563

Thesis Submitted in Fulfillment
of the Requirements for the **Degree of Doctor of Philosophy (PhD) in Information and Communications Technology**

Faculty of Computer and Information Technology

Al-Madinah International University

Supervised by:
Prof. Dr. Zainab Binti Abu Bakar

Co-Supervisor:
Assoc. Dr. Najeeb Abbas Al-Sammarraie

December 2017 /Rabulaul 1439H

بسم الله الرحمن الرحيم

CERTIFICATION OF DISSERTATION WORK PAGE

The thesis of student: YAZEED ALSAYYEED ALI AL MOAIAD under the title: ENHANCE MODEL FOR SELECTING CLOUD SERVICE PROVIDER USING DECISION MAKING ALGORITHM has been approved by the following:

Supervisor Academic

Name	**……………………….**
Signature	**……………………….**

Supervisor of Amendments

Name	**……………………….**
Signature	**……………………….**

Head of Department

Name	**……………………….**
Signature	**……………………….**

Dean of the Faculty

Name	**……………………….**
Signature	**……………………….**

Deanship of Postgraduate Studies

Name	**……………………….**
Signature	**……………………….**

DECLARATION

I declare that the work in this thesis is my original work; it has not submitted previously or concurrently for any degree or qualification at any other institutions, I hereby confirm that there is no plagiarism or data falsification/ fabrication in the thesis, and scholarly integrity is upheld as according to the Al-Madinah International University (MEDIU).

Name: ………………………….

Signature ……………………….

Date: ……………………………

COPYRIGHT

Al-MADINAH INTERNATIONAL UNIVERSITY

DECLARATION OF COPYRIGHT AND AFFIRMATION

OF FAIR USE OF UNPUBLISHED RESEARCH

[ENHANCE MODEL FOR SELECTING CLOUD SERVICE PROVIDER USING DECISION MAKING ALGORITHM]

No part of this unpublished research may be reproduced, stored in a retrieval system, or Transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of the copyright holder except as provided below.

1. Any material contained in or derived from this unpublished research may only be used by others in their writing with due acknowledgement.
2. MEDIU will have the right to make, store in a retrieval system and supply copies of this unpublished research if requested by other Universities and research libraries for instructional and academic purposes.

Name: ………………………...

Signature ……………………..

Date: ………………………….

ABSTRACT

The number of Cloud providers are expanding their services differ from one another. These have caused difficulty for the user to choose the best services for a particular application. It is a tedious process for a user to search and select services from each provider before a chosen service can be extracted and compiled according to user preference. This process will repeat until all the desired services by the user are compiled. Then the user has to rank the providers and make a decision based on their requirements. Unfortunately, the user has to search the information again as the information frequently updated by the providers. Thus the objectives of the research are to identify providers based on user's specific applications and requirements; to build a model that user can use to decide appropriate services based on user requirements; to construct and evaluate a decision model that overcome user's difficulty to consider changing services and recommend the ranking of providers with similar services; and to enhance the decision model to grab changing services to show updated data of providers instantly. These objectives are achieved through the construction of measuring preferred service (MPS) model. Three models, Static Infrastructure as a Service (SIaaS), Dynamic Infrastructure as a Service (DIaaS) and Enhance Dynamic Infrastructure as a Service (EDIaaS) are evaluated before the MPS can be employed. The MPS model are flexible to other the cloud services and providers. The linear equation model has successfully help user in making the best decision according to their requirement and recommended providers.

Keywords: *Cloud Computing, Decision Making, CLOUD SERVICE Provider, User Requirement, Linear Equation, Web Services.*

ACKNOWLEDGEMENT

First and foremost, I am grateful to Allah for providing me the strength and inspiration needed for carrying out this study. I would like to extend my appreciation to my supervisors, Prof. Dr. Zainab Binti Abu Bakar, Dr. Najeeb Al-Sammarraie Dr. Wael Yafooz, for their thorough supervision, encouragement, and willingness to support me throughout this study. To them, I would like to express my sincere gratitude for their enthusiasm and guidance. I also acknowledge my close friends Dr. Maged Mustafa, Eng. Marwan Mustafa, and Dr. Ahmad Salah who assisted during the writing thesis.

I wish to dedicate this study to my family, as they have always been my strongest supporters. I am indebted to all my family members for their love and appreciation during my study.

DEDICATION

All praise, glory, and honour to God for His everlasting and steadfast love for me throughout my life and throughout this journey of completing my doctorate degree. With deep gratitude to my parents, Mr. Mohammed Al Moayed and Dr. Abubakr for instilling in me the values of faith, character, virtue, discipline, patience, perseverance, and the encouragement to pursue and achieve my educational goals. With great appreciation and love to my darling wife, Mrs. Wafa Abdulkarem for her wonderful love, encouragement, and support throughout this arduous journey.

I express my deep gratitude to my siblings, cousins, aunties, uncles, nieces, nephews, in-laws, friends, and extended family relatives for their support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

Term	Description
AHP:	Analytical Hierarchical Process
AWS:	Amazon Web Services
CPNI:	Centre for the Protection of National Infrastructure
CSMIC:	The CLOUD SERVICE Measurement Index Consortium
CSP:	CLOUD SERVICE Provider
CSUs:	CLOUD SERVICE Users
EC2:	Elastic Compute Cloud
HPC	High Performance Computing
LEMPS:	Linear Equitation for Measuring the Preferred Service
IaaS:	Infrastructure as a Service
ICs:	Integrated Circuits
GCE:	Google Compute Engine
MPS:	Measuring Preferred Service
MPSU:	Measuring Preferred Service of User
MRPC:	Measuring Reputation of Provider Company
PaaS:	Platform as a Service
QoS:	Quality of Service
TR:	Trust and Reputation
SOA	Service-oriented architecture
SaaS:	Software as a Service
VISE:	Value of Isolated Service Equation
VTSE:	Value of Total Services Equation
DIAAS:	Dynamic Model the Infrastructure as a Service
NGS:	National Grid Service
PTUR:	Prioritization Tool on User Requirement
ITOOL	Intelligence Tool
EITOOL	Enhanced Intelligence Tool

CHAPTER ONE

INTRODUCTION

This chapter provides background of the Cloud Computing, Cloud service providers, scope, problems and objectives of this research. The advantages of Cloud Computing and the problems faced in Cloud service selection in defining the best service provider based on user requirement are discussed. This chapter ended with the thesis organization.

1.1 BACKGROUND OF STUDY

The Cloud Computing is represented as the mover of resources and systems across the Internet. Cloud services allow the users to use the software, systems, and services from anywhere. Companies providing these services are very effective and competitive, for instance, Dropbox, Google Docs, Pixlr Editor and Jaycut (Whaiduzzaman et al., 2014).

There are four types of Cloud Computing such as Public, Private, Hybrid, and Community. Cloud services models are normally isolated into three layers, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Mell and Grance, 2009).

Well-known companies, which have established large high-capacity servers for Cloud Computing applications are Google, Amazon, Microsoft, and IBM (Antoniou, 2012). As known, the number of providers and their services is increasing. In addition, their costs are on the rise. They provide services that are changeable and not adaptable to user requirements, for instance, automatically scalable storage and CPU, depending on the load on the servers.

Additionally, the users face some difficulty when requesting any service needed with a certain cost whenever and from anywhere (Malik and Nazir, 2012). Examples of widely known providers are Google Compute Engine (GCE), Microsoft Azure (Azure), Amazon Web Services (AWS) and Rackspace Cloud (Rackspace). Users are interested in knowing which providers give high-quality services versus costs. At the same time providers are always seeking to improve and develop their services.

This makes a robust competition among providers to provide many options which make the users have difficulty and challenging in selecting the appropriate Cloud services (Gui et al., 2014; Jahani et al., 2014).

If the users are not knowledgeable, they will have difficulty and take a long time in order to find their preferred services (Cao, Li, and Xia, 2009). With all these features, benefits and deployment, the users still face obstacles to select and ascertain the Quality of Service (QoS) of the providers (Zheng et al. 2013; Garg, Versteeg, and Buyya, 2013). In addition, the users need to know more regarding providers and their services (Saravanan and Kantham, 2013). These increases the difficulty of the user's ability to make the decision to choose the best Cloud services based on their needs (Wang et al., 2011).

Currently, there are no tools that can help the user to make the right decision in choosing the providers with regard to services (Sun, 2014) and Quality of Services (QoS) provided (Garg, Versteeg and Buyya, 2013).

There are many applications that employ Linear and Quadratic Models Optimization for Decision Making (Murty 2009, Shaw et al. 2012; Hall and Miller, 2012; Kannan et al, 2013; Harrell, 2015). Rezaei (2015) has proved a real-world decision-making problem using multi-criteria decision-making (MCDM) method based on linear models. Since linear models have been successfully implemented in many applications, this model will be implemented to assist the users in making decision in choosing cloud computing services.

1.2 PROBLEM STATEMENT

Cloud Computing providers have similar services and faces failures such as interrupted services or downtime. This creates difficulty for the user to choose the best service provider that considers the user requirement. Currently there is no optimal decision method employed to help user to select, compare and to rank providers based on user requirement and reputation of the provider. Therefore, in this research, the problems that are identified are as follows:

Recently, there is a demand for Cloud Computing services. Unfortunately, the user is not an expert and as a result, the user will need more time to look and define user preferred services (Zheng et al., 2013; Kumar and Agarwal, 2014; Mamoun and Ibrahim, 2014). It is critical challenging to select the best service provider for users based on specific applications (Zheng et al., 2013; Kumar and Agarwal, 2014; Mamoun and Ibrahim, 2014).

There are varieties of services from providers for the users to decide based on user requirements. Currently, there are difficulties to find all functional and essential services' data for any virtual server to be combined in one application (Saravanan and Kantham, 2013).

The users confront problem in considering the provided services as the services vary in volumes, performance and costs. Moreover, the information on the services change from time to time (Garg, Versteeg, and Buyya, 2013; Jahani and Khanli, 2014).

It is difficult to get updated data for all providers instantly, leading to potential difficulty for user to make a decision about which the best service provider that considers all the user requirements is because there are many providers of similar services with different range of costs (Garg, Versteeg, and Buyya, 2013).

1.3 RESEARCH QUESTIONS

This thesis addressed the following research questions.

1. What are the reputable providers based on user's specific applications and requirements?

What is the model that combined all functional and essential services' data for any virtual server in one application and helps user to decide appropriate services based on user requirements?

What is the decision model that overcomes user's difficulty to consider changing services and recommend the ranking of providers with similar services?

How to grab changing services to show updated information of providers instantly?

1.4 RESEARCH OBJECTIVES

The main goal of this research is to propose a model, which helps the user to select the best Cloud Computing service provider, which can satisfy the user requirements of Cloud services.

Therefore, to achieve this goal, the following objectives have to be met:

1. To identify providers based on user's specific applications and requirements.

To build a model which is combined all functional and essential services' data for any virtual server in one application and helps user to decide appropriate services based on user requirements.

To construct and evaluate a decision model that overcome user's difficulty to consider changing services and recommend the ranking of providers with similar services.

To enhance the decision model to grab changing services to show updated data of providers instantly. It briefly presents the measuring preferred service (MPS) model based on a set of services preferred that is validated by users.

1.5 SCOPE AND LIMITATION

The study focuses on investigating and improving solutions, which have many processes and mechanisms as a part of produced work. It will help to resolve the problem's research in Cloud Computing for selecting the optimal Cloud service provider and the rest of phases in order to improve the quality of the model to achieve the main goal and get true results. So that will focus on:

1. Four Cloud providers which are AWS, GCE, AZUR, and RACKSPACE who are the famous in Cloud Computing.

Service model only involves Infrastructure as a Service (IaaS).

Five essential services such as SSD, RAM, CPU, Bandwidth, and cost.

Users who use Cloud Computing such as IT personnel from MEDIU who faces problems in infrastructures such as overload on the servers and Internet speed.

Important techniques that are relevant employed to enhance the performance of the models.

Thus the limitations will not cover: Service models as Software as a Service (SaaS), Platform as a Service (PaaS).

1.6 CONTRIBUTIONS

1. A method to help users to select suitable services according to the application from amongst many Cloud providers.

Techniques such as worker role, cache redis, and SignalR has increase system efficiency and speed display instantly up to date information and the results. These techniques can be employed in other Cloud Computing applications.

A flexible model called Measuring Preferred Service (MPS) for measuring services according to the requirements of users. MPS performs search and ranking efficiently for the user to select the best Cloud service provider.

There are three models which are known as Static Infrastructure as a Service (SIaaS), Dynamic Infrastructure as a Service (DIaaS) and Enhance Dynamic Infrastructure as a Service (EDIaaS). These models can be used for other Cloud services and providers.

Intelligence Tool (ITOOL) which grabs and parses data from providers to gain specific data for user.

Prioritization Tool on User Requirement (PTUR) which can be used to sort services and providers based on user's requirements.

Linear equations can be used to make decision in selecting Cloud services from providers.

1.7 RESEARCH GAP

Cloud Computing has been growing exponentially as hundreds of companies competing in this field. Based on the importance of this field, there are several leading and experienced companies called Cloud Providers (CSP). There are three layers to offer the service - SaaS, PaaS, and IaaS, in which each layer has certain characteristics.

Through previous studies, the researchers have been interested in one of these layers. In this research, focused on IaaS, because there is a great practical experience to know the characteristics of this layer in Mediu University. The focus on measuring the services provided, which was the main importance of researchers in all models in that studies. Unfortunately, researchers did not focus on measuring the service based on the user requirements. The research gap is how to do the measurement of the services based on the user requirements. Figure 1.1 shows the relationship between all elements in Cloud Computing and research gap.

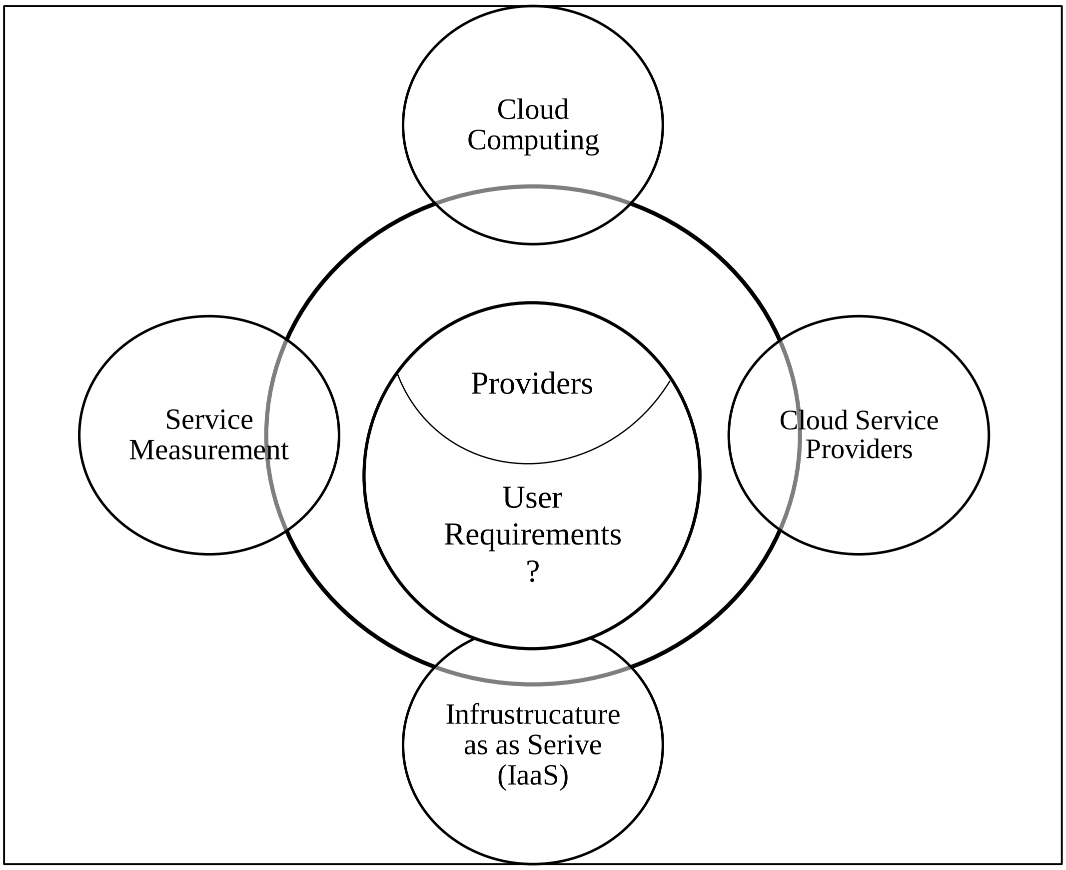


Figure 1.1: *RESEARCH GAP*

1.8 THESIS OVERVIEW

1. Proposed model algorithms and machines for selecting the best service provider. Before proposing algorithms, will begins by providing a motivational application that is currently used in the real-world. It describes the challenges that this model faces while in practice. It then formally defines the workflow model of selection problem. Then it proceeds towards implementing the workflow system to address these challenges.
2. Proposed several workflows algorithms that are implemented in the workflow MPS model. It first describes linear equation of model based measuring services and scheduling algorithms that produce optimal results, which would then be an ideal case for comparison approaches. It proposes static, dynamic and enhances the dynamic model. These algorithms are used to minimize the total execution time of measuring data (services according to the priority of users). In addition, it is designed to achieve a minimum cost of time execution for outputting data. Due to its fast convergence data property, MPS reduces the significant cost for

sample application executed on Cloud Computing.

3. MPS model executes real-time data measuring. The experimental users evaluating the proposed linear equation algorithm of the model through inputting data from different resources and outputting data from specific users.
4. The rest of this thesis is organized as follows:

Chapter 2:

This chapter presents an overview of Cloud Computing, some of the previous models for measuring QOS, selecting services in Cloud Computing and web services, and proposes characterization and classifies providers of Cloud Computing and Cloud services selection. This thesis then presents the design of the MPS model.

Chapter 3:

This chapter describes the support of the model through a phase test data collection, which is a help to list of providers, user, and services their characteristic needed in the proposed model. Then the MPS model will analyze it. (Identification of the Possible Solution).

Chapter 4:

This chapter moves on the suggestions to work out on the mentioned issues as discussed in the previous study and presents the executed steps in achieving the stated research objectives. In this chapter, MPS model of Cloud provider selection is based on measuring preferred service of user Cloud.

Chapter 5:

This chapter presents how the model works to suggest another solution for the problem of existing models for selecting the best service provider, and how to implement, and evaluate its environments.

Chapter 6:

This chapter concludes the work in achieving MPS model which not only help users to select the best service provider but also understand their own requirements and services to assign ranking provider companies to help users to save time in process of searching of Cloud Computing provider. The future work will be more comprehensive and it will take into account more numbers than the mentioned companies and their reputation of providers which will contribute a high trust efficiency for providers of Cloud Computing.

1.9 SUMMARY

In this chapter, the motivation goals of this work are discussed. Cloud Computing focusing on problems that are given by a brief overview of, difficulties and challenges faced by users when selecting the service and provider. By introduction of Cloud Computing that dives deeper into the IaaS Cloud services, and also the reputation of the provider as a service in Section 4.2. Hence, everything related to Cloud Computing, providers, quality of service and all previous models that help to choose the provider will be discussed in the next chapter.

CHAPTER TWO

LITERATURE REVIEW

This chapter emphasizes on the difficulties faced by the user when selecting the best service provider that can satisfy its needs in Cloud Computing. Different available models in the literature related to examination of varying models and the common problems. It also describes the definition and the classification of Cloud services and Cloud service and provider selection together with the benchmark datasets used in this work. Note that the discussion on the available approaches in this chapter is arranged based on the different models and models to measure and classify services and providers. Due to the large number of related work in this area, this chapter only focuses on the most significant related works, which is concentrated on measuring the quality of service and the optimal choice to define the best service provider. Algorithms and mechanisms are also investigated.

2.1 INTRODUCTION

Cloud Computing allows a service provisioning model, which commonly includes the provisioning over the Internet, of powerfully versatile and virtualized services. Applications or services offered by this method for Cloud Computing are called Cloud services. The three primary models of Cloud services are Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) (Mell and Grance, 2009).

Cloud Computing is the following stage in the development of Internet. The Cloud Computing provides everything from registering energy to figure foundation, applications, and business procedures to the individual coordinated effort that can be conveyed as a service wherever and at whatever required point. It utilizes a productive way to deal with the execution of ideal web service Computing and positioning in order to satisfying service requesters utilitarian and non-practical necessities (Zou, Xiang, Gan, Wang and Liu, 2009).

Cloud Computing is a model for conveying data innovation services in which assets are recovered from the web through online devices and applications, as opposed to an immediate association with a server. Information and programming bundles are stored away in servers. Nevertheless, Cloud Computing structure permits access to data as long as an electronic gadget has admittance to the web. This type of system allows employees to work remotely.

The essential concept of Cloud Computing is to transmit computational assets as service over the web. The clients do not have to put resources into an immense PC model to manage their works, instead of that can buy the Cloud services and gain benefits (Fujiwara, 2011).

Cloud Computing has gained a large importance as of late. Increasing numbers of people, associations, and companies have been transferring their businesses into Cloud in regard of its versatility and negligible exertion. Amazon, Google Compute Engine (GCE), Sun Microsystems Salesforce, IBM, Microsoft, and Sun Microsystems have begun to set up new server branches to encourage Cloud Computing applications in various ranges everywhere to give abundance and assurance-enduring quality if there ought to be an event of site disillusionments, (Antoniou, 2012). Because of the variety of providers and their assortment services in Cloud Computing, they make computational environment particularly in gigantic and acclaimed organizations, for example, AWS, AZURE, Google Compute Engine (GCE), and Rackspace and in all sort of associations.

Every one of them can provide a high caliber of services with high expenses to appealing clients who need their necessities and most likely are aware that before the rise of Cloud Computing, the clients need to pay a lot of money for costly IT equipment and specialize in using them. However, Cloud clients without IT foundation today can oversee and characterize their own requirements. Thus, there is no requirement to search for specialists with high compensations.

The advantage of Cloud Computing is that the clients will save their money, time, enhance the execution of their own works, and many advantages. Cloud Computing is the innovation for the decade. It is needed by Cloud client for storing a huge measure of information in Cloud storage and use as and when needed from any place around the world, by means of any terminal hardware (Malik and Nazir, 2012).

There is an immense number of Web services giving comparable functionalities, and more emphasis is being given on the most proficient method to search for a provider which best fits the shopper's necessities.

Cloud Computing offers alterable, adaptable, shared resources, for example, figuring force, stockpiling and programming over the web from remote server farms to the clients which for example are business associations, government powers, and people. The substance of Cloud Computing is to give services to the system. To the extent of client’s concerns, assets in the "Cloud " can be augmented inconclusively whenever, procured whenever utilized on interest, and pay-per-use.

Joining with Multi-Agents and SOA innovation is considered to have the most advantages but the problem is, if the users are new and do not have the expertise, they will spend more time looking for their preferred services (Cao, Li, and Xia, 2009). As mentioned above about the advantages, spreading and services of Cloud Computing, the user will still face difficulties in defining optimal choices for the best service provider.

Table 2.1, illustrates groups of definitions for the Cloud Computing based on different researchers as follows:

Table 2.1
Cloud Computing Definitions

Definition	Reference
“Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers which provide those services.”	Armbrust, Fox, Griffith, Joseph, Katz, Konwinski, Lee, Patterson, Rabkin, Stoica and Zaharia (2010)
“Cloud Computing can broadly define as several different methods to deliver information or services to customers who pay for what they use.”	Baca (2010)
“A Cloud is a pool of virtualized computer resources. A Cloud can host a variety of different workloads, including batch-style back-end jobs and interactive, user-facing applications; allow workloads to be deployed and scaled-out quickly through the rapid provisioning of virtual machines or physical machines; support redundant, self-recovering, highly scalable programming models that allow workloads to recover from many unavoidable hardware/software failures; and monitor resource use in real time to enable rebalancing of allocations when needed.”	Boss et al. (2007)

"A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers which are dynamically provisioned and presented as one or more Unified Computing resources based on service-level agreements established through negotiation between the service provider and consumers."	Buyya, Yeo, Venugopal, Broberg, and Brandic (2009)
"The Cloud is a collection of Internet-based or private network services providing users with scalable, abstracted IT capabilities, including software, development platforms and virtualized servers and storage."	Callewaert et al. (2009)
"Cloud Computing provides on-demand, a shared pool of configurable IT resources (e.g. processing, network, software, information, and storage), as a scalable and elastic service, through a networked infrastructure, on a measured (pay-per-use or subscription) basis, which needs minimal management effort, is based on service level agreements between the Cloud provider and Cloud consumers, and often utilises virtualization resources."	Carroll, Kotze and Van der Merwe (2012)
"Cloud separates application and information resources from the underlying infrastructure, and the mechanisms used to deliver them. Cloud enhances collaboration, agility, scaling, and availability, and provides the potential for cost reduction through optimized and efficient Computing. More specifically, Cloud describes the use of a collection of services, applications, information"	Cloud Security Alliance (2009)

Source: Carroll, Kotze and Van der Merwe (2012)

There are four types of Cloud Computing. Through its definitions and the differences between these types of Cloud Computing, the right type can be determined for the organization. The main types of Cloud Computing and services are as follows:

2.2 TYPES OF CLOUD COMPUTING

Four different ways in which the Cloud services can be deployed: Public Cloud, Private Cloud, Hybrid Cloud and Community Cloud.

2.2.1 Public Cloud

The servers and other physical register assets are situated off-premises, typically possessed, and facilitated by a third party who also gives Cloud services to different clients. Accordingly, numerous disconnected associations share the assets. So it increases the productivity of scale and does not obligate the client to support the capital uses or upkeep costs for equipment. However, it can also expose security and protection issues and gives clients minimal control over their information and applications) Shinder, 2013).

In Public Cloud, clients can access web applications and services across the web. Every individual client has its own assets that are progressively given by Cloud providers. These providers encourage numerous clients from various server farms, deals with all the efforts to establish safety and gives equipment and foundation to the Cloud clients to work. The client has no clue about how the Cloud is overseen or what model is accessible. Clients of Public Cloud services are thought to be untrusted (Mamoun and Ibrahim, 2014).

2.2.2 Private Cloud

The servers and the different segments are located in a private system, which might possibly be on organization premises. A private Cloud is a form of Cloud Computing that carries identical advantages to public Cloud, inclusive self-service and scalability, through a proprietary architecture. A private Cloud is determined to a single organization. It gives more protection and better control over the information, applications, and security of the model, however, it can be more expensive with the fact that the equipment has to be purchased and maintained (Shinder, 2013).

In private Cloud, clients have complete control over how information is overseen and what efforts to ensure security are set up while information is preparing in Cloud. The clients of the organization are viewed as "trusted". Thus, clients of the organization are the individuals considered to be a part of an association including representatives, temporary workers, and business accomplices (Mamoun and Ibrahim, 2014).

2.2.3 Hybrid Cloud

As the name shows it is a hybrid of the public and private Cloud whereby a few assets are facilitated by an open Cloud provider and others are facilitated in a private

Cloud. Thus, permits the client to maintain closer control over the information and applications that are important, maintained with high security and in contrast, reduce expenses and exploit the advantages of the public cloud. The major issue is that the provider needs to manage dual levels of infrastructure, a private cloud which is controlled by the client and the public cloud that does not, that typically one that is neatly arranged (Shinder, 2013). Hybrid Cloud is a blend of public and private Cloud inside of the same system. Private Cloud clients can store individual data on their private Cloud and utilize in public Cloud for taking reduce excess expenses from increased requests (Mamoun and Ibrahim, 2014).

2.3 COMMUNITY CLOUD

A community Cloud sits some place in the middle of public and private Cloud, however, it is not a blend of the two such as hybrid Cloud Cloud sits some place in the middle of the general population and private Cloud, however, it is not a blend of the two such as a cross breed Cloud. Its Cloud Foundation may likewise be known as a "mutual Private Cloud." Several associations have basic concerns or attributes team up to share Cloud assets. A community Cloud can be facilitated by a third party or the physical assets can be kept up by one or a greater amount of the interested associations (Shinder, 2013).

2.4 MODELS OF CLOUD SERVICES

There are three main Cloud Computing service models, namely, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). This three service can be shown in the following Figure 2.1:

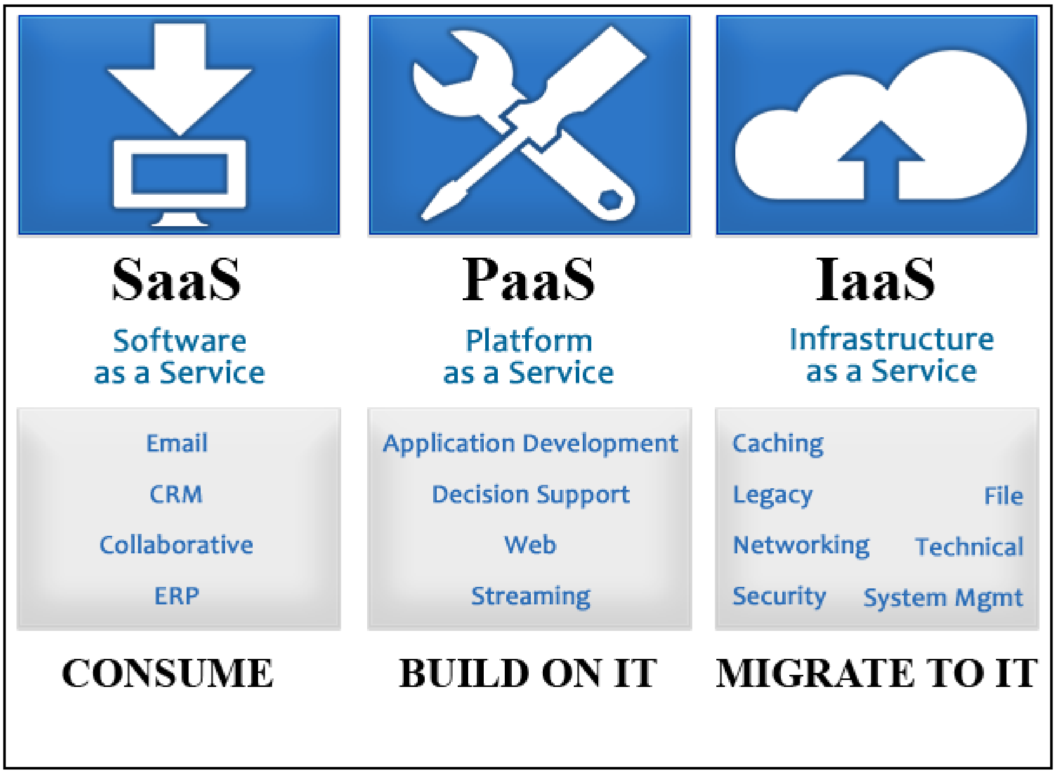


Figure 2.1: *Categories of Cloud Services*

Source: (Technologicasolcom, 2017)

As in Figure 1, which show the three layers, namely Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The definition of each layer, its characteristics, its workflow and how the customer will benefit from them will be explained in the next section

2.5 SOFTWARE AS A SERVICE (SaaS)

This is the Cloud services model allows users, and organizations to connect and use cloud applications over the Internet. Webmail services, for example, Outlook.com (once in the past Hotmail) and Gmail have been around for quite a while. Lately, the service of office efficiency applications as Cloud services (Google Apps, Microsoft Office Web Apps) has become more famous. In addition, TeamViewer (web conferencing), Cloud Analytics (Business Intelligence) and Box (stockpiling) are different SaaS providers (Shinder, 2013).

2.5.1 Platform as a Service (PaaS)

This model is regularly utilized by developers to make, test and run programming without investing in the equipment and keeping up the fundamental working models. The Cloud providers offer base, stage, and programming to the clients in a conservative and reliable way (Manuel, 2015). PaaS can be delivered in two ways: as a public cloud service from a provider, where the client controls applications deployment with minimal configuration options, and the provider provides the networks, servers, storage, operating system (OS), 'middleware' (e.g. Java runtime, NET runtime, integration (Manuel, 2015).

2.5.2 Infrastructure as a Service (IaaS)

The first model is IAAS when IT professionals think of about “going to the Cloud.” In this model, the constructions are usually located in the physical infrastructure and virtual machines are usually hosted by servers and network resources running on the provider’s equipment, although on-premises IaaS solutions are also available from Citrix, Microsoft, and VMware which can only access client application remotely, the accessing of user (almost always virtual) server across the Internet or carrier network, and client can install and run applications anytime (Shinder, 2013). Amazon Web Services is the best example of an IaaS provider, including Google’s Compute Engine, Rackspace Microsoft’s Azure Infrastructure Services and Amazon’s Elastic Compute Cloud (EC2) and others. Additionally, AT&T and Verizon are transforming into this model. The providers who offer Computing and storage resource Infrastructure as a service and the capacity which allows physical resources to be assigned and split it dynamically (Whaiduzzaman et al., 2014).

The Virtual Machine (VM) is a key asset on offer from an IaaS Cloud service and also a reproduction of a physical machine (Computer). A VM picture can be conveyed rapidly as it takes a couple of minutes to setup a VM. The Cloud provider has control on the physical machines while the VMs are controlled by the user from numerous points of view including online interfaces and SaaS customers, where VMs are taken (Buyya et al., 2009).

2.5.3 IAAS and Preferring Services

The user's preference of criterion and values should be taken into account when selecting web services with same functionalities (Fan et al., 2010). Different preferences or requirements are taken from different users. It is essential to be capable of representing QoS from the point of views of service user’s preference. For instance, regardless of the time, it takes to execute the service, selecting service Cloud can be paid completely in price (Liu, Ngu, and Zeng, 2004) Many users at the same time exceeding its capacity can be hosted by the provider. An application can be created by the user, using services offered by multiple providers, in order to meet his need (Fujiwara, 2011).

Infrastructure as a service (IaaS) is responsible for hosting virtualized Computing resources over the internet. MPS model is dependent on the IaaS model which can host hardware, software, servers, storage and other infrastructure components by the provider.

A. Functional Services

Basically, any virtual server including function services such as:

1. Solid-State Drive (SSD)

This has an arrangement of semiconductor memory organized as a disk drive, for using integrated circuits (ICs) instead of optical or magnetic storage media.

CPU

The abbreviation for central processing unit is CPU. It is referred to fully as the central processor, but more generally called processor, the brain of the computer is the CPU where most calculations take place.

RAM

An acronym for random access memory is RAM, a type of computer memory which can be accessed randomly where any byte of memory can be accessed without touching the preceding bytes. The most common type of memory found in computers is RAM and other devices, such as printers.

Bandwidth

The size of data that can be inherited in a fixed amount of time is a bandwidth, which is usually expressed in bytes per second for digital devices.

Cost

The cost is an important service for the customer, as a low cost attracts the customer to move their business into Cloud space because of its scalability and low-price (Qu, Wang, and Orgun, 2013). Before selecting Cloud Computing, the first thing that arises in the mind of Cloud users is, if it is cost-effective or not. Therefore, the cost is clearly one of the vital services for the business and IT (Mamoun and Ibrahim, 2014). The same services at different prices with different features are offered by many providers. For instance, cheaper storage might be offered by one provider whereas to rent powerful virtual machines from them are very expensive (Saravanan and Kantham, 2013).

A study published in February 2013 by KPMG International shows that the cost reduction is clearly the most important objective for organizations' Cloud adoption. Almost half of the respondents consisting of business and IT executives Include cost as a key objective with regard to a cloud strategy. Nearly, 70% of the organizations who are using Cloud answered that Cloud is delivering efficiencies and cost savings today. Though this is a high number, there were around 20% who were not certain of any efficiencies or cost savings and the rest responded that the Cloud is actually hindering their efficiencies.

Infrastructure costs in some cases can be up to 60% of the total costs of the software development. The costs can be divided into operational attributes and business premises. Operational attributes refer to three elements: hardware costs, software costs, and license fees. Business premises are personnel expenses and costs of physical locations, such as rental and electricity costs (Bibi, Katsaros and Bozanis, 2010).

Table 2.2 shows a percentage of the required operational qualities. In the Table, the second section is marked if the quality is pertinent when utilizing Cloud services and the third segment is marked if the property is significant when utilizing on-premises improvement. Note that the last line in Table, software licenses for the server, is dependent on chosen programming decisions. At the point when just open-source software is utilized, there is no matter of any server side software permit charges. Table A.2 presents the results of the work for (Bibi, Katsaros and Bozanis, 2010).

Table 2.2
Infrastructure Costs

Operational attribute	Cloud	On-premises
Development devices (computers)	*	*
Peripheral devices (accessories)	*	*
Device maintenance	*	*
Server infrastructure	-	*
Server maintenance	-	*
Subscription fees	*	-
Server software licenses	*	*

Source: Bibi, Katsaros and Bozanis, 2010

Not all fundamental base expenses are avoided by essentially picking the Cloud approach. Nonetheless, server base contains the server PCs themselves as well as the physical space, system associations, save parts and support faculty. Therefore, there are numerous cases that support Cloud IaaS model over ordinary facilitating Iosup, Yigitbasi and Epema.

B. Non-Functional Services

There is related research classified the reputation of Cloud provider as one of the non-functional services, which include the age of company and availability. The reputation of Cloud provider can attract client turn to the certain provider, based on clients' recommendations, and users' experiences are important features and may increase trust granted service-level agreement (SLA). As a result, the experience of users is an important feature which helps two sides, to satisfy the new users who are not experts."QoS properties provide more realistic measurements of the user usage experience model (Zheng et al., 2013).

The Cloud provider needs to define users' requirements through the users' preferences, then Cloud provider can present the service which satisfies their need. Thus, the reputation and users' preferences will help users to select best Cloud provider. The comprehension about one's QoS criteria originates from experience, may the fast development in the quantity of Cloud applications forces Cloud provider to consider QoS criteria from a new perspective (Kumar and Agarwal, 2014).

The reputation service will encourage the Cloud provider to improve its provided services to attract attention of a larger prospective user base (Itani et al., 2014).

The reputation is the reliability of the service provider, described as users' ratings and advertising messages reliability, (Salama et al., 2012). In the markets, the provider places sell orders with a market when they have a capacity of service with specific QoS. The user places a buy order with a

1. Age of Company

Since 2000, Cloud Computing has come into existence. Amazon.com is one of the most prominent Cloud providers, and the first provider to employ the IaaS model, in 2006 (Antoniou, 2012).

The Cloud providers in its long life can increase and expand experiences in Cloud Computing. Age of the provider company in MPS is important in showing how the users need the old capabilities and abilities of providers. When considering a Cloud service's age, users want to understand whether the provider is professional, have a high understanding and have flexible experiences from the past users to attain their preferred services. It is important for users to attain the best cost for their services which help them to select their services according to the best price from the provider (Raderbauer, 2011).

2. Availability

The different Cloud providers offer many different services; each one has different performance in terms of functionality, uptime and down time. The providers of Cloud services need to be more understandable of how their applications will perform on the different Clouds and fulfill their clients’ expectations (Salama et al., 2012).

In the case study based on defined providers which include Google, IBM, Amazon, and Microsoft, they have started to offer different Cloud services to their customers (Garg, Versteeg and Buyya, 2013).

2.6 THE QUALITY OF SERVICES QOS

The quality of services (QoS) in distributed models applications is very important. In Cloud Computing, QoS requires the consideration of specific strategies that give the clients a nature of experience while utilizing the services (Adinolfi, Cristaldi, Coppolino and Romano, 2012).

Kourtesis, Alvarez-Rodríguez and Paraskakis (2014), clarified the difficulties confronted in meeting QoS guidelines in Cloud models and introduced a semantic based service system which gives the intelligent and interoperable environment to observe assorted services in Cloud.

In a comparative work (Kafetzakis, Koumaras, Kourtis and Koumaras, 2012), QoS in Cloud viewpoint is measured as far as the degree up to which the provisions of the service level agreement (SLA) are met. Infringement of SLA brings down the client's trust in the Cloud service provider.

Ding, Yang, Zhang, Liang and Xia (2014), displayed a plan for assessing service reliability, taking into account the anticipated QoS and level of consumer loyalty. However, utilizing the extent of using past clients' experience of service fulfillment has not been done in the present work.

Yuchao, Bo and Fuyang (2012), exhibited the connection between QoS conveyance and service provider workload. They ensure that with a specific end goal to guarantee the craved level of value, a service provider must be reasonably stacked so it does not leave the guaranteed quality properties, for example, reaction time and accessibility of assets, because of work over-burden. Along these lines, QoS is also identified with burden adjusting and in this way, at whatever point applications are scaled up the QoS must also be kept up. Then again, correlation of providers' workload and positioning as indicated by minimum or most over-burden service provider is not examined.

Alhamazani, Ranjan, Rabhi, Wang and Mitra (2012), proposed the utilization of computerized quality checking models to identify varieties in service execution. Nevertheless, their works do not encourage the client to choose the service provider conveying slightest variety in service quality. As depicted in the area the test is handled by utilizing a system called irregularity calculation as depicted in the area.

Dynamic solicitations by Cloud clients lead to the organization of services, a great record of which is given by Jula, Sundararajan and Othman, (2014). Therefore, straightforward and composite service Computing from a pool of accessible services on the Internet is an NP-difficult issue (Zhao et al., 2012) and a few critical thinking methodologies are utilized to get the answer for an ideal choice of services.

For quick and best choice of Cloud services (Zheng et al., 2013) exhibited a strategy of comparability calculation that recognizes past clients with comparative QoS prerequisites and after that utilized the past decisions to determine on current choices. This methodology may not be suitable situations where there is a vast irregularity in the execution of Cloud providers.

From numerous writing study, it is observed that although a lot of work is dedicated to the strategies and instruments for upgrading QoS on the providers' end, little thought has been given to taking care of client's issue of finding the best decision for service necessities.

Garg, Versteeg and Buyya (2013), recorded a few measurements for the ranking of Cloud service has been given and Cloud providers are ranked based on the given measurements. However, the likelihood of varieties in the measurements, which might change the rank of a service provider, has not been investigated. It applies Analytic Hierarchy Process (AHP) (Saaty, 2008a), which is a promising way to deal with Multi-Criteria Decision Making (MCDM) problem, for taking care of the positioning issue. AHP has an endless scope of use in tackling designing issues as given by Lee and Walsh, (2011).

2.7 CLOUD SERVICE SELECTION METHODS

Zhang, Ranjan, Nepal, Menzel and Haller (2012), Selection methods are to identify objects that meet the needs and objectives of individuals who are able to choose. When the amount of choice standard is exactly done, all viewpoints are gathered into the grouping of multi-criteria choice making (MCDM) (Triantaphyllou, 2000). The main issues focused on the decision-making strategy are to examine the cloud providers by surveying the different criteria as well as QoS, which helps the client to make the decision (Triantaphyllou, 2000). The common multi-criteria choice making (MCDM) models used in Cloud Service Computing are the experiential hierarchy of leadership methodology/symptomatic model process, Multi-Attribute Utility Theory (MAUT), Analytic Hierarchy Process (AHP) and essentially, Substance Weighting (SAW) is included.

2.7.1 Analytic Hierarchy Process Method (AHP)

Saaty (2008a), analytic hierarchy process (AHP) is an across the board service positioning strategy whereas the AHP chain of command is built to compose the service data, then support a sensible demonstration of paradigm conditions. An AHP-based classification step includes three principal steps: disintegration, near judgment and combination. In the disintegration stage, a chain (c1, c2, and c3) of importance is made to show the relationship between the choice components (ac1, ac2, and ac3). Pair-wise correlations (Pa and Pb) are connected in order to decide how components at one level affect a component with the larger amount. AHP has a type of this equation (1).

$$[\text{c1 c2 c3}] . \begin{bmatrix} \text{ac1} & \text{bc1} \\ \text{ac2} & \text{bc2} \\ \text{ac3} & \text{bc3} \end{bmatrix} = [\text{Pa} \quad \text{Pb}] \qquad (1)$$

2.7.2 Multi-Attribute Utility Theory Method (MAUT)

The AHP concentrates mostly on this proportional significance of this choice standard during pair-wise examinations, therefore, Multi-Attribute Utility Theory (MAUT) depends on the utility capacities (Garg, Versteeg and Buyya, 2011). This utility capacity which quantifies the inclinations the leader makes and the total difference of the chief's degrees of fulfillment of a specific measure. MAUT has a type of this equation (2).

$$U_j = \sum_j w_j u_{ij} \quad \ldots\ldots\ldots\ldots \quad (2)$$

(San Cristóbal, 2012).

Zeng, Zhao and Zeng (2009); Limam and Boutaba, (2010); Cavalcante et al., (2011, 2012); Salama et al., (2012) said, specific works connected to MAUT-based on Cloud service Computing methods.

Zeng, et al. (2009) The Cloud service was examined the key mathematical calculations for choosing this Cloud service. Choices largely rely upon these exchanges between an expanded addition and this decreased cost that is controlled via a multi-attribute used capacity. Thus, presented the service conduct method of Cloud services which have three capacities: service register, service revelation, and service representative. In view of this model, two stages performed in the service Computing technique: first defining a rundown of services to satisfy client prerequisites, secondly a specific service must be selected based on service upgrade with cost increase accordingly.

Limam and Boutaba (2010) suggested a reliable services-based Computing strategy. In this method, a notoriety based service choice system is intended to rate SaaS services and diminish the danger and time of the choice with the usage of programming services. The suggested Computing system helps service clients to choose services in view of three main standards: expense, notoriety, and quality. The notoriety of a service provider depends on the criticism of users that can reflect on their utilization experience. Dissimilar to the other works, client input is naturally created by watching the clients' long haul service appropriation inclinations as opposed to their genuine assessments, which empowers the notoriety of the service provider to be evaluated more equitably. The target input is framed by conglomerating the apparent utility of the client's "pattern" fulfillment and the apparent disconfirmation of the client's normal fulfillment. The utility of the services can be computed by checking the outcomes.

The notoriety can be resolved later to help clients gauge the reliability of the service provider sooner rather than later. Four determining methods – moving normal, weighted moving normal, (SES) straightforward exponential smoothing and Holt's direct exponential (HLE) (Brockwell and Davis, 2006) – are connected for notoriety estimating. Pragmatic analyses are made to demonstrate the efficiency and legitimacy of these strategies. Taking after the notoriety computation, a service Computing procedure is portrayed which includes service assessment, service positioning, and service matchmaking.

In the matchmaking stage, competitor services will be filtered, taking into account the coordinating degree between this service depiction and client prerequisite to the extent of addition and expense, where the services are then assessed. A final score for every service can be Figured by a weighted mean-like score capacity that combines notoriety, pick up and cost connected with every service. The service with the most noteworthy score will be the base decision for services.

Outranking-based methodologies for Cloud service choice, outranking strategies (Roy, 1991) are another gathering of MCDM techniques. They are selected as follows: Alternative Ai outranks Aj if on a generous percentage of the criteria Ai performs better at any rate, and in addition, Aj (concordance condition), while the execution is worse, it is still worthy on the other criteria (non-dissonance condition) (Fülöp, 2005).

Garg, Versteeg and Buyya (2013), surveys hopefuls in every foundation and identifies the strength level of one competitor over another.

The key contrast in the middle of MAUT and the Outranking techniques is that the former results in the best decision while the latter offers a shortlist of options. The portion of the tasks, which have connected outranking-based Cloud service Computing methodologies, are examined in this segment. Elimination and Choice Expressing Reality (ELECTRE) is primary outranking technique. It applies concordance and conflict records to develop a halfway positioning of an arrangement of options (Fülöp, 2005). In light of ELECTRE (Silas, Rajsingh, and Ezra, 2012), suggested a Cloud service choice middleware from clients (i.e. client inclination for every choice basis) and from providers (i.e. service portrayal), the concordance file, the dissonance record and the validity degree are Figured.

The concordance file introduces the honesty of the outranking connection between two competitors regarding a given rule. The harshness list is utilized to judge the accuracy of the outranking connection as per the execution comparison between the two options in view of the basis. The believability degree totals the concordance and conflict file to demonstrate the outranking connection regarding the entire arrangement of criteria. Lastly, two types of refining strategy – dropping and rising – are performed to accomplish the final positioning of the service.

2.7.3 Simple Added Substance Weighting

Simply added substance weighting (SAW) (Afshari, Mojahed and Yusuff, 2010) is a basic MCDM strategy and widely recognized. It assesses choices by the mathematical statement Ai ¼∑wjxij, where Ai is the ith elective, wj is the heaviness of the jth foundation, and xij is the score of the ith elective concerning the jth measure. A general rating result for every service can be computed to help with choice making.

Saripalli and Pingali (2011) talked about SAW-based techniques to classify choices in a decision problem in the area of Cloud service appropriation. In light of general choice making hypothesis, the creators broke down the conceivable choice issues that may be experienced by service clients when they plan to utilize Cloud services and identify the choice, establish standard and optional answers for these choice issues.

They presented a trait chain of importance involving six property tuples that shape the choice criteria for the proposed Cloud service Computing system – Multiple Attribute Decision Methodology for Adoption of Clouds (MADMAC). A Wide-band Delphi-based strategy is proposed to survey the relative weights for every standard by directing a specialist meeting. The SAW strategy is then utilized to rank the service competitors based on the created rank values.

Zhao et al. (2012) displayed the SAW-based service seeking and planning calculation – Service Provider Search Engine (SPSE) – to get an arrangement of positioned services. This exploration concentrates on service Computing with the flexible backing of client personalization and numerous targets in SOA and the Cloud environment. The creators dissected the issues of existing Web service choice strategies and identified the difficulties of planning a viable service choice technique. The proposed strategy incorporates four essential operations: seek, filter, rank, and overhaul. In the first process, services with needed service sorts and accessible providers are seeking to utilize indexing innovation.

The sought services are then filtered by means of a Pareto ideal based choice technique to enhance planning efficiency. The third step is service designed to position the SAW-based technique; that is, every parameter of the service is first positioned in light of the qualities given by the service provider. The positioned values duplicated with client inclinations are then added to land at the final positioning of the service. The upgrade operation naturally computes and overhauls the client inclinations of the criteria as indicated by the beginning inclinations of clients and their consequent service choosing. Other MCDM-based methodologies for Cloud service choice, Research on Cloud service Computing has additionally been conducted, utilizing other MCDM strategies (Han and Sim, 2010; Martens, Teuteberg and Gräuler, 2011; Wittern, Kuhlenkamp and Menzel, 2012) to bolster the service Computing process.

Wittern, Kuhlenkamp and Menzel (2012) proposed an organized model, in particular, the Cloud highlight model (CFM), taking into account the element demonstrating strategy. CFMs can catch the abilities of Cloud services and necessities by a method for variability demonstrating and shape the learning of Cloud service areas, Cloud services, service-particular configurations, client prerequisites, and the totaled relations between service properties in a unified representation design.

A model contains four sub-models, specifically an area model, service model, prerequisites model and option model. A space model is utilized to portray the theoretical parts of decision problems; a service model alludes to specific Cloud benefits; a necessities model signifies the prerequisites of a leader for the service Computing choice, and an option model stands for a reasonable configuration coming about because of a service model.

The move strategies between various models are defined to get elective models from service models consolidated with the prerequisites models; this is alluded to as a specialization of an element model. On the premise of CFMs, a CLOUD SERVICE Selection Process (CSSP) is acquainted with a presently reasonable service Computing workflow. In a CSSP, a space model, numerous service models, and a prerequisites model are defined as the info in choice making. Attainable service configurations are then produced, assessed and prescribed to service clients. The creators proposed MCDM based strategies for service evaluation and selection.

Martens, Teuteberg and Gräuler (2011) introduced a group stage that helps organizations and clients to choose appreciate Cloud services. The service data is put away in the fundamental database of the stage and can be measured by a development model. The model measures the separation between the development object (i.e. Distributed Computing Service) and the client prerequisites to demonstrate the development degree as far as every assessment paradigm. In this way, the development level of a service reflects the level of consistence with client prerequisites and is ascertained by a weighted math normal technique.

Every paradigm in the development model can be evaluated by five development degrees – Initial, Acceptable, Industry Standard, High Standard, and High Quality. In addition, the creators who gave the first investigation of the business sector circumstance furthermore gave counsel on Cloud service organizations and research.

2.7.4 HPC Grids, IaaS Cloud Processing Services

Computational services offer physical assets such as capacity, correspondence and preparing as services. These services have stayed in broad use by the exploratory examination group and to some degree by the business (Foster, Kesselman and Tuecke, 2001). Bunch processing and Grid figuring are the predecessors to Cloud registering. The services Grid Computing services network registering is centered on substantial scale sharing of computational assets, for example, stockpiling and handling, (Foster, Kesselman and Tuecke, 2001) proposed the idea of Grid figuring. It proposed the making of a computational Grid for explaining huge register serious undertakings. Lattice figuring is expanded on exploration in the field of Cluster processing, as a computational Grid is an arrangement of approximately coupled registering machines or bunches.

The Grid is an equipment and programming model that gives trustworthy, predictable pervasive and in costly access to the top of the line computational capacities. The lattices were made by pooling computational assets from various associations to fill a typical need.

The pooling required equipment foundation to accomplish the fundamental interconnections and programming to screen and control the subsequent gathering. (Foster, Kesselman and Tuecke, 2001) matrices were extremely prevalent among examination organizations exceptionally colleges and various computational lattices were made that incorporate the White Rose Grid (Dew, Schmidt, Thompson and Morris, 2003), Nordu Grid (Eerola, Kónya, Smirnova, Ekelöf, Ellert, Hansen and Vinter, 2003) and the Sun Grid (Gentzsch, 2001).

Issues confronted by computational lattices incorporate substantial forthright expenses, connected with purchasing equipment and programming assets. An even more of a problem that needs to be addressed is viewing the asset allotment as because of the substantial number of clients, each user’s employment is submitted to a vocation line. As of late, the utilization of Grids has been diminished because of the accessibility of Cloud. One such case is National Grid Service (NGS) service Cloud services. NGS is the biggest open part provider of HPC assets in the UK. It serves various instructive and research associations (“National Grid Service,” 2016).

2.7.5 Optimization-Based Methods in CLOUD SERVICE

In the area of services choice, enhancement is determined as "to find appreciate services for the customers or providers that amplifies or decreases one or a few criteria and still holds fast to the imperatives" (Dastjerdi and Buyya, 2011). The writing, Martens, Teuteberg and Gräuler (2011); Chang, Liu, and Wu (2012a); He, Han, Yang, Grundy and Jin (2012) ; Nizamani (2012); Sundareswaran, Squicciarini and Lin (2012); Jung, Mukherjee, Kunde, Kim, Sharma and Goetz (2013) ; Yang, Lin and Dou (2013); Zheng et al. (2013), demonstrate that an assortment of streamlining techniques have been connected to Cloud service choice, for example, dynamic programming, whole number programming, covetous calculation, and so forth. Dynamic writing computer programs are basically used to handle advancement issues in light of the guideline of optimality (Wang, Zhang and Liu, 2009).

Chang, Lai and Huang (2012b) added to a dynamic programming-based calculation to choose Cloud stockpiling providers which can boost the information survival likelihood or the measure of surviving information, which is subject to a fixed spending plan. It planned the issue of numerous capacity service provider Computing into a likelihood model with plainly defined object capacities and cost estimations. The accessibility of the capacity service is quantitatively broken down into two techniques least disappointment likelihood with a given spending plan, and most extreme legitimacy with a given spending plan. The previous cases a Computing to minimize the general disappointment likelihood under a given spending plan."

2.8 THE CHALLENGE OF CLOUD SERVICE SELECTION

Goscinski and Brock (2010), present a detailed description of the process of Cloud service publication, discovery, and selection. Through a higher deliberation of Cloud in light of dynamical and trademark characteristics, they propose a general structure which appeared to be sound and plausible by a proof of the idea. Then again, there is no solid choice methodology proposed in this work. In the literature, the problem of Cloud service selection can usually be solved by approaches of Cloud service comparison in view of target execution investigation (Li, Yang, Kandula and Zhang, 2011).

Proposal of an efficient comparator called Cloud Cmp. Cloud Cmp can be connected to look at three parts of the execution and expense of a Cloud (i.e., flexible registering, diligent capacity, and intra-Cloud and wide range organizing). These examinations are acknowledged by an arrangement of standard benchmark instruments, whose outcomes show the target evaluation of a Cloud (Li and Wang, 2010).

In Binnig et al. (2009), consider the differences of Cloud and the measurements of versatility, cost, top load, and adaptation to internal failure. Another discourse of Cloud benchmark testing is introduced by (Lenk, Menzel, Lipsky, Tai and Offermann, 2011). In their work, they call attention to the issue that the execution markers given by Cloud providers may not be sufficient to judge the genuine execution of a virtual machine and propose another execution estimation system which considers the sorts of services executed on a virtual machine for Infrastructure-as-a-Service Cloud. As of late, some outsider associations (e.g., Cloud Harmony) have begun to offer Cloud observing and benchmarking services (Leitner and Cito, 2016).

At the point when compared with the execution markers given by Cloud providers, such outsider testing might be more valid because of no direct profits included. All the methodologies specified above focused on breaking down the target parts of a Cloud service in light of quantitative testing. No subjective viewpoint is considered in order to reflect the general execution of a Cloud service.

In customary e-trade or e-service situations, service choice, for the most part, relies on the notoriety based trust assessment of services. Comparing with right on time trust assessment approaches in view of processing a solitary trust esteem for a service, (Li and Wang, 2008; Li, Wang and Varadharajan, 2009) proposed a few trust vector based assessment approaches, where a trust vector is ascertained to reflect both the present dependability of a service and its trust trend. Such trust values or vectors are all evaluated from evaluations which speak to the subjective appraisal of services given by service buyers, keeping in mind the end goal to consider subjective parts of a Cloud service (Li and Wang, 2010).

Rehman, Hussain, Parvin and Hussain (2012) proposed a basic system for checking Cloud execution in view of client criticism, in which the execution of a Cloud service is observed and anticipated by clients' input. Their methodology only considers Cloud clients' subjective evaluation. There is no component to check the dependability of clients' criticism. Moreover, the target appraisal of a Cloud service is not considered in their structure. Another answer for Cloud service choice issue is to demonstrate the issue as a multi-criteria choice making (MCDM) issue (Sakhuja, Jain and Dweiri, 2015), which can be regularly fathomed by Analytic Hierarchy Process (AHP) (Avikal, Mishra and Jain, 2013).

In (Godse and Mulik , 2009), focused on the choice of Software-as-a-Service Cloud in light of AHP. Five elements (i.e., usefulness, construction modeling, and ease of use, seller notoriety and expense) are considered in their methodology. It ought to be noticed that every one of these elements aside from expense can barely be quantified by a goal measure. Hence, their methodology is still for the most part in view of subjective appraisal. Another AHP-based Cloud examination methodology is proposed by (Garg, Versteeg and Buyya, 2013). In their work, they endeavor to institutionalize the execution qualities for Cloud correlation. Be that as it may, the institutionalization for a few qualities (e.g., supportability and straightforwardness) is excessively basic, making it impossible to reflect the intricate circumstances of Cloud services in this present real world.

2.9 DATA GRABBING APPROACHES

In this section, the existing service selection approaches (Espadas, Molina, Jiménez, Molina, Ramírez and Concha, 2013; Chunlin and Layuan, 2014), for example, Grid web services and the accessible Cloud strategies, have been produced for execution assessment and cost-benefit examination (Walterbusch, Martens and Teuteberg, 2013). The present utility capacity will be joined in their proposed model. In the matrix connection, the utility has been for the most part utilized for asset allotment. The creators have presented a utility model for asset allotment on computational matrices (Espadas et al., 2013).

The creators have considered different QoS-based network asset booking. Despite the fact that Cloud and network figuring standards have numerous focuses in same manners, QoS based models created for various purposes cannot be embraced in the Cloud Computing, particularly while tending to the service provider Computing issue. In the web services, setting, utility capacities have been utilized for service structure and enhancement (Chunlin and Layuan, 2014). The authors tended to the issue of selecting web services with the end goal of their piece in a way that boosts client fulfillment communicated as utility capacities over QoS traits. Subsequently, utility values and works planned for different purposes cannot fit with the provider choice issue for a typical client whose objective is selecting a Cloud provider (Wu,Chen, Feng, Zheng, Zhou and Wu , 2013).

To sum up, methodologies produced for matrices cannot be specifically relocated to the Cloud environment. Past results from Grid Computing cannot be linked, as various applications are being tended to and attributes of the issue in every domain essentially vary (Dastjerdi and Buyya, 2011). In spite of the fact that back-end innovations might be comparative, the objective gatherings and normal use states for Grids and Clouds are altogether different. Concerning Cloud execution assessment, reaction time and throughput variables were considered.

The creators concentrated on the reaction time from the viewpoint of the client, who is more disposed to ask for a factual bound on its reaction time than a normal reaction time. Case in point, a client can ask for the reaction time of his service to not exceed 0.005 seconds in 95% of the time. Consequently, the paper is worried about a percentile of the reaction time. The metric has been utilized by IBM's specialists (Xiong, and Perros, 2009).

The metric is called percentile delay by researchers at MIT and Cisco Communications Future Program (Jacob and Davie, 2005). The throughput is likewise considered as a critical execution variable in an extensive variety of uses, for example, content delivery networks (CDN) (Buyya et al.2009; Pathan, Broberg, and Buyya, 2009), where the service organization is enhanced, taking into account normal throughput; and the utility is measured as part of prepared solicitations (throughput) or the aggregate assessment (weighted throughput). Then again, money saving advantage investigation has also been led by the Cloud Computing services.

A correlation in the middle of Cloud and matrices has been led in the execution and asset necessity in view of money saving advantages. It has been reasoned that cost proficiency differs depending on the stage size, where a base number of hubs is adequate for a lattice to end up financially savvy for less time and whereas a Cloud of the same size is effective enough to bolster a long haul experimental venture (Chunlin and Layuan, 2014).

Figuring cost metric and evaluating the estimation of the Cloud as far as circumstance expenses have been proposed in (Menzel, Schönherr and Tai, 2013). Such structure helps chiefs to gauge and think about the Cloud Computing costs for ordinary IT arrangements. By and large, regardless of the exertion done to investigate the money saving advantage, purchase utility, however, is not considered.

Here, it is worth to note that there were few endeavors done to address the QoS assessment point, which are standalone measurements such as straightforwardness, accessibility, unwavering quality, and notoriety has been considered. The creator is concentrated on how a business evaluates Cloud providers' straightforwardness (Pauley, 2010). A tool has been created to assess a Cloud provider's straightforwardness from security, protection, and service level capabilities viewpoints.

Cloud service purchasers fundamentally focus on accessibility, dependability, and notoriety, notwithstanding the pre-considered element: reaction time and cost (Cao, Li, and Xia, 2009).

Utilizing existing methodologies of Distributed Computing, the chosen methodology will be to construct either in light of a solitary quality parameter or on execution assessment. For this situation, the provider assessment will not think about the quality concerns that a shopper wants and the basic requirements. Compared to the use of QoS, the buyer's basic requirements for a particular application will be fully explored in the choice of the ideal provider.

In any case, the optimal procedure cannot be performed by utilizing accessible systems to execute assessment or money saving advantage examination. Moreover, ideal service provider choice cannot be accomplished just on a specific quality parameter; rather it must be dissected as a multi-dimensional QoS issue, considering every service class in an arrangement of quantifiable quality parameters connected with a weight that characterizes the significance of that element, and adding to a model for ideal provider Computing Clouds the need of consolidating scientific devices that boost the Cloud purchaser's utility quality.

In this manner, the stress is on the utility capacity as one of those qualities. The author said that the utility capacity has already been presented to bring the model's state to a typical scale, speaking in reaction time and a number of QoS objectives met (Paton, De Aragão, Lee, Fernandes and Sakellariou, 2009). The creators defined utility-amplification issue with quantitative expressions; with enhancing content serving utility of Content Delivery Clouds. From their point of view, utility-amplification can be accomplished from the utility that is measured by handled solicitations (throughput) (Pathan, Broberg and Buyya, 2009).

Therefore, today a client may have a decision of different stages, for example, Windows, Mac, and Linux. These stages work diversely and programming produced for one cannot be utilized by the other. This absence of interpretability focuses to a requirement for programming that can be utilized and all-inclusive without being secured to a particular stage. This stage autonomy is accomplished by Web Services which collaborate over the Internet; utilizing a Web program (Lemos, Daniel and Benatallah).

A Web service is a product application that can distribute its capacities and messages to the world through the Internet and is open through numerous processing gadgets. The key point of preference of a Web service over a customary programming application is its worldwide availability and stage autonomy (Becker, Nickolas and Vicknair, 2014).

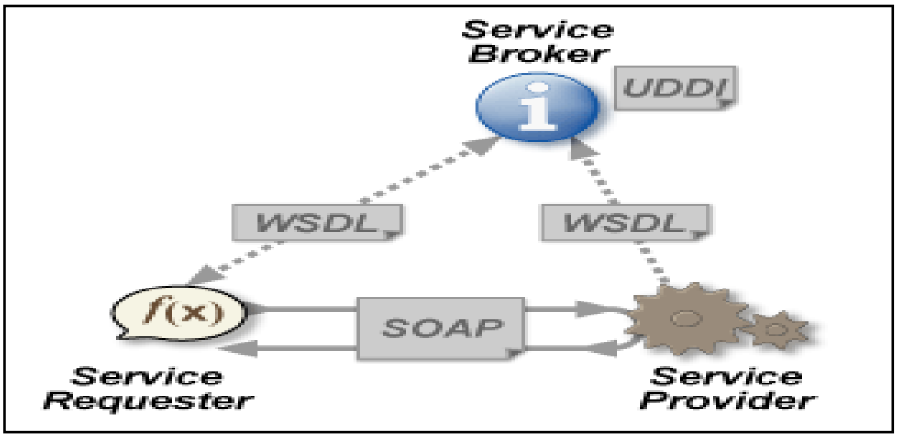


Figure 2.2: *Web Service Architecture*

Source: Perrey and Lycett (2003)

As in Figure 2.2, a web service has three related gatherings, in particular, the service provider, the service requester (additionally alluded to as client or shopper) and the service representative. The correspondence channel between a service client and a provider is Simple Object Access Protocol (SOAP). Cleanser uses Extensible Markup Language (XML) for informing. The XML message is designed by utilizing Hypertext Transfer Protocol (HTTP) and utilizing Simple Message Transfer Protocol (SMTP) for transmission (Box, Ehnebuske, Kakivaya, Layman, Mendelsohn, Nielsen, Thatte, and Winer, 2000; Curbera, Duftler, Khalaf, Nagy, Mukhi and Weerawarana, 2002).

Web services are depicted utilizing Web Service Description Language (WSDL) (Duftler et al., 2002). WSDL depictions contain data identifying with sort, operations and tying. Sorts portray the kind of information being handled, for example, string or number. The operations are the rundown of capacities that the service can perform, the coupling contains subtle elements of the physical system essential for correspondence, for instance, IPs addresses and ports.

2.9.1 Web Services Evolution

One of the significant advances for web services was the improvement of composite services. A composite service consolidates the usefulness of numerous services to achieve a particular objective. An individual service can do a solitary undertaking while a composite service could accomplish an extensive complex errand. (Claro, Albers and Hao, 2006). A case of this could be reserving an occasion where a client needs to book a flight, taxi, and an inn. These errands can either be reserved through three individual services or by one composite service. Composite services are subject to a compelling choice calculation, as these needs to be distinguished and selecting the most suitable service for every errand. Web service Computing is not just significant to the compelling working of composite services, additionally, it is fundamental to the choice of individual services by a client.

2.9.2 Web Services Selecting

The least complex type of service Computing includes coordinating users' demand for the useful properties of the accessible services. The Computing process includes utilizing contingent programming, for example, if-else develops. These are utilized to look at two values and achieve a choice. (Moghaddam and Davis, 2014) The key issue confronted by the useful Computing is its failure to separate among services with the same usefulness. With an end goal to enhance the choice, new calculations were proposed which utilized QoS parameters.

Serhani, Dssouli, Hafid and Sahraoui (2005); Tran, Tsuji and Masuda, 2009) contends that: "With various Web services having comparative usefulness, it is important to rank those services to choose the best Web services for a solicitation. QoS data which can mirror client's desire and experience of utilizing a service is frequently utilized as the recognizing element as a part of a service positioning calculation."

The utilization of QoS parameters is additionally contended by others, including those who express that numerous providers can coordinate a user's useful necessities in the manner QoS pre requisites go about as the separating rule (Godse and Bellur, 2011). As contended over a viable method for utilizing useful choice is a part of conjunction with the QoS-based choice. Various existing Computing calculations, for example, Euclidean Distance (Danielsson, 1980) and Analytical Hierarchy Process (AHP) (Saaty, 2008b) have been utilized to handle the users QoS prerequisites. Since 2000, Microsoft proposed web services, which utilized XML, SOAP, and WSDL. The key inspiration driving Web services was e-trade (Levitt, 2001).

2.9.3 Semantic Web Services

The key favorable position of SWS is the upgraded level of robotization for Web service revelation, structure, and conjuring. Different focal points incorporate institutionalization of naming plans and standard configuration for the portrayal, stockpiling and trade of information (McIlraith, Son and Zeng, 2001; Sirin and Hendler, 2003; Sycara et al., 2003). If there is an occurrence of Grid or Cluster, each user's employment is submitted to a line, while if there is an occurrence of Cloud, the client can get to any measure of assets whenever. Replication expands the unwavering quality of the service and because of this component, a user's employment won't be hindered now and again because of catastrophe. Killing forthright costs, pay per use, and high usage is lucrative to bosses or individuals in-control as these assists in dealing with the expenses (Armbrust et al., 2010).

2.10 USER SELECTION MODELS

There are relevant models devoted to defining the quality of services, based on previous service description model and how the user could define the services of Cloud Computing and select the best service provider in different overviews of researchers. In addition, they help users in understanding their service needs, which can satisfy their requirements in Cloud Computing. A trust and reputation system is an instrument utilizing customers' criticisms to distinguish great services from awful ones (Habib, Riesy and Muhlhauser, 2011).

2.10.1 Ranking Models

A proposed structure offers the Cloud clients to pick the best organization provider who needs some assistance with satisfying their QoS prerequisites and for the client who needs to pick the best providers, taking into account their necessities and utilize the proposed system by disregarding all superfluous characteristics. Likewise, the system requires no extra services of Cloud services while selecting the best provider, they propose a structure for positioning and reservation of the Cloud services which depend on an arrangement of Cloud Computing particular execution and QoS properties. The QoS information is gathered from a different assessment of three IaaS Cloud providers: Amazon EC2, Windows Azure, and Rackspace (Mamoun and Ibrahim, 2014).

Cloud Rank model is proposed for ranking QoS Cloud services by taking advantage of the past service usage experiences of other consumers, there can be a prediction of Two Clouds. The first one belongs to the past user (client side) and the second belongs to the current user (Zheng et al., 2013).

There is a belief in the importance of reputation of the provider in the market, and the needs of users depend on their conditions, such as money and application. A web service description model that considers service QoS information, and then presents an overall service selection and ranking model with QoS (WSSR-Q) based on previous service description model (Zou et al., 2009).

The proposed Cloud Service Provider Selection Engine (CSPSE) model is not just to assist the users to understand one's own demand but also serves to specify a ranking order to the Cloud providers on QoS that are necessary for a specific use, (Kumar and Agarwa, 2014).

A proposed sorting model that compares the providers on different QoS, ranks them in line with their performance and shows the performance of four Cloud providers with respect to all the QoS attributes (response time- Elasticity- Cost-Availability) and ranking them. The hierarchical structure of QoS suggests that the classification of QoS attributes needed by the customers for selecting the appropriate providers are based on cost, performance, assurance, security, usability, agility and accountability, (Saravanan and Kantham, 2013). They proposed a sorting model, that compares the providers on different QoS and ranks them according to their performances. The four pivotal QoS attributes are cost, availability, response time and elasticity (Kumar and Agarwa, 2014).

The work is analyzed, ranked and compared with the existing approaches for ranking Cloud Computing services. They believed this work can enable users to use existing approaches and Cloud Computing providers can compare their quality services with other adversary and can increase quality services. In this work, a review is done on the approaches for ranking Cloud services and analyzes them. It resolves some of the user's challenges about selecting best service which satisfies their requirements. In addition, this works mentions advantages and disadvantages or limitations of the existing approaches (Jahani and Khanli, 2016).

The customers need to know more about their requirement when making optimal CLOUD SERVICE selection; the model proposes a QoS ranking prediction model for Cloud services by taking advantage of the past service usage experiences of other consumers and personalized QoS ranking prediction approaches are proposed to predict the QoS rankings directly (Zheng et al., 2013). The proposed model helps the Cloud customers to choose the best service provider who satisfies their QoS requirements (Mamoun and Ibrahim, 2014).

2.10.2 Measurement Models

This model gives a choice to the client to assess different accessible CSPs in view of their notoriety in the business sector for the sake of QoS given and chooses the most dependable CSP after profound examination of clients' needs and pre requisites qualities. The proposed CLOUD SERVICE Users (CSUs) model will offer the Cloud some assistance with servicing users in discovering effective and reliable CLOUD SERVICE Provider (CSP) on the premise of information taken from administrative powers. Execution of the CSP in most recent years, and inputs taken from the user's notoriety in the business sector for the sake of QoS given and chooses the most dependable CSP (Naseer, Jabbar and Zafar, 2014).

The proposed model of context-aware Cloud service selection based on comparison and aggregation of subjective assessment from Cloud users and objective assessment from quantitative QoS monitoring and benchmark testing (Qu et al., 2013). In structure, the general model depends on combined Cloud, which consists of various Cloud and a Cloud trade unit. Every Cloud has an organizer component, which gathers the solicitation from the clients and checks it and then gives the requested asset with five methods.

The mathematical model addresses the Cloud service provider selection optimization based on QoS guarantees, as well as displaying the suggested QoS utility-based model for best Cloud service provider selecting in the market of Cloud Computing platforms to match the characteristics of various kinds of Cloud services (Salama et al., 2012).

The CLOUD SERVICE Measurement Index Consortium (CSMIC) proposed a frame for best provider selection using Ranked Voting Method. In this work, a model has been designed which considers not only metrics defined by SMI, but additionally, other metrics will act as a voter and compare its required value with providers providing the value of metrics to rank providers accordingly (Baranwal and Vidyarthi, 2014).

This work presents SMIC Cloud, to systematically measure all the QoS attributes and rank the Cloud services based on mentioned attributes, and also creates metrics for every accounTable QoS to measure the service level of each Cloud provider accurately. The Analytical Hierarchical Process (AHP) based ranking mechanism, which can evaluate the Cloud services related to various applications depending on QoS needs. Service measurement index (SMICloud) is able to compare various Cloud providers based on user requirements and the SMICloud would allow users to compare various Cloud services, in line with their preferences (Garg, Versteeg and Buyya, 2013).

Proposal of a novel context-aware Cloud service selection model based on the comparison and aggregation of subjective assessment extracted from Cloud user feedback and objective assessment from quantitative performance testing (Qu et al., 2013). The proposed mathematical model produced salutation for the service selection problem, and QoS dimensions (Salama et al., 2012).

The Cloud service recommendations are based on user preferences and practical Cloud provisioning; and visually presents and compares solutions through an interactive web Graphical User Interface (GUI) (Gui, et al 2014).

With the proposed novel model of Cloud service selection based on aggregating the information from both the feedback from Cloud users and objective performance analysis from a trusted third party, a method is proposed for filtering feedback from such users (Qu et al., 2013). Figure 2.3 Proposed model displaying processes of selecting service as following:

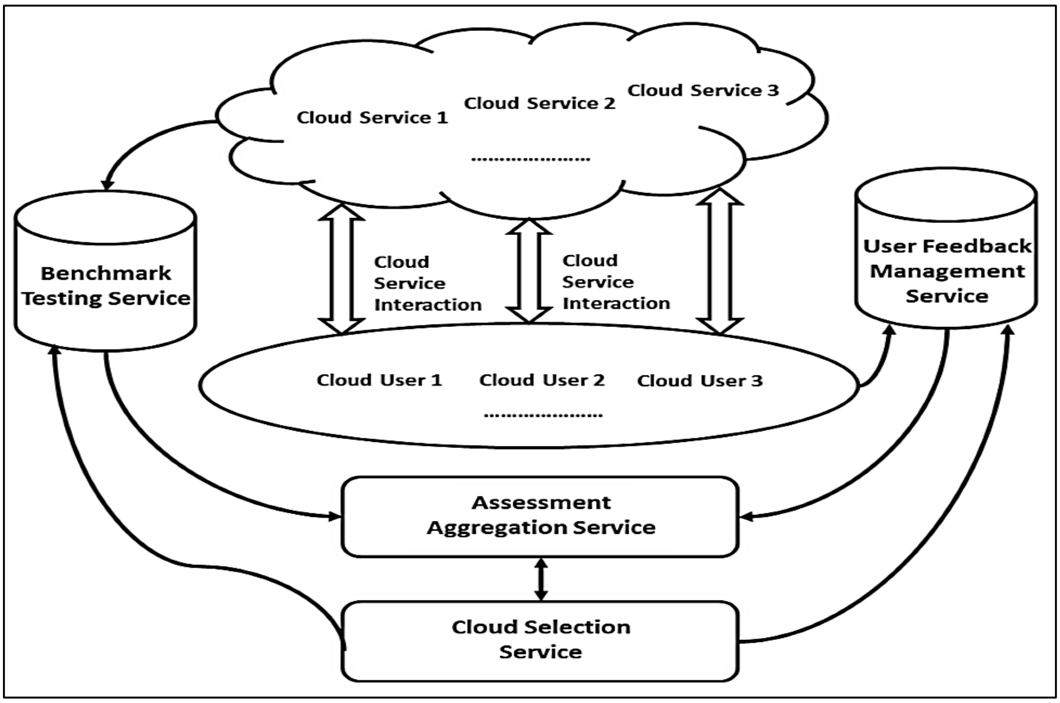


Figure 2.3: *Proposed Model for CLOUD SERVICE Selection*

Source: Qu et al. (2013)

Figure 2.3 illustrates the model, which consists of four components, namely, Cloud selection service, benchmark testing service, user feedback management service, and assessment aggregation service. Cloud Selection Service is in charge of accepting and preliminarily process the requests for Cloud service selection from potential Cloud consumers and issues requests for the services from the lower layer components. Cloud selection services are in a higher layer in this model. Therefore, Table 2.3 shows the case study of one advantage of the proposed model.

Table 2.3

Case Study of one advantage of the proposed model

Top Level QoS (Weights)	1st Level Attributes (Weights)	2nd Level Attributes (Weights)	Service 1 (S1)	Service 2 (S2)	Service 3 (S3)	Requirement 1	Requirement 2
Accountability (w_1)	Level: 0-10 (1.0)		4	8	4	4	4
Agility (w_2)	Capacity (0.6)	CPU (0.5)	9.6	12.8	8.8	6.4 GHZ	9 GHZ
		Memory (0.3)	15	14	15	10 GB	12 GB
		Disk (0.2)	1690	2040	630	500 GB	700 GB
	Elasticity (9.4)	Time (1)	80-120	520-780	20-200	60-120 Sec	70-120 Sec
Assurance (w_3)	Availability (0.7)		99.95%	99.99%	100%	99.9%	99%
	Service Stability (0.2)	Upload Time (0.3)	13.6	15	21	--	--
		CPU (0.4)	17.9	16	23	--	--
		Memory (0.3)	7	12	5	--	--
	Serviceability (0.1)	Free Support (0.7)	0	1	1	--	--

		Memory (0.3)	24/7, Phone, Urgent Response, Diagnostic Tools	24/7, Phone, Urgent Response, Diagnostic Tools	24/7, Phone, Urgent Response	24/7, Phone	24/7, Phone, Urgent Response
Cost **(w_4)**	On-Going Cost (1.0)	VM Cost (0.6)	0.68	0.96	0.96	<1 $/hour	<1 $/hour
		Data (0.2)	10	10	8	100 GB/month	120 GB/month
		Storage (0.2)	12	15	15	1000 GB	1000 GB
Performance **(w_5)**	Service Response Time (1.0)	Range (0.5)	80-120	520-780	20-200	60-120 Sec	70-120 Sec
		Average Value (0.5)	100	600	30	--	--
Security (w_6)	Level: 0-10 (1.0)		4	8	4	4	4

Source: Mamoun and Ibrahim (2014)

Table 2.3 illustrates the user who wants to choose the best providers based on their requirements, the user must neglect all unnecessary attributes or increase the weight of their required attributes. The Cloud environment is created for three Cloud services from three Cloud providers and two users' requests ($requirement_1$, requirement2). The QoS data is collected from the various evaluation of previous studies for three IaaS Cloud providers: Amazon EC2, Windows Azure, and Rackspace (Mamoun and Ibrahim, 2014).

In addition, Table 2.4 sets different priorities and numeric weights for user request the results coming from the Computing process is as follows:

1. The relative ranking of all the Cloud services can be decided based on the resultant Relative service ranking value (RSRV) for each requirement as:
2. For requirement1: RSRV1 = (0.3424, 0.2702, 0.3874). Based on the user 1 requirements, then Cloud services are ranked as S3 > S1 > S2.
3. For requirement2: RSRV2 = (0.3770, 0.2919, 0.3269). Based on the user 2 requirements, then Cloud services are ranked as S1 > S3 > S2.

Table 2.4
User Request Weights

weight user request	w_1	w_2	w_3	w_4	w_5	w_6
$Requirement_1$	0.05	0.10	0.20	0.30	0.30	0.05
$Requirement_2$	0.05	0.05	0.20	0.40	0.20	0.10

Source: Mamoun and Ibrahim (2014)

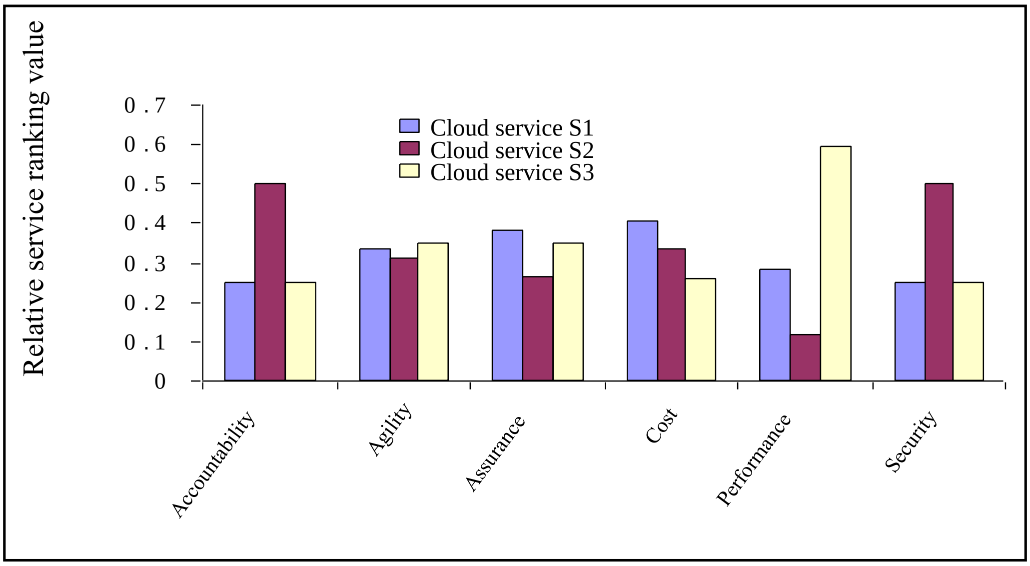


Figure 2.4: *Clouds Services Comparison Requirement for User1*

Source: Mamoun and Ibrahim (2014)

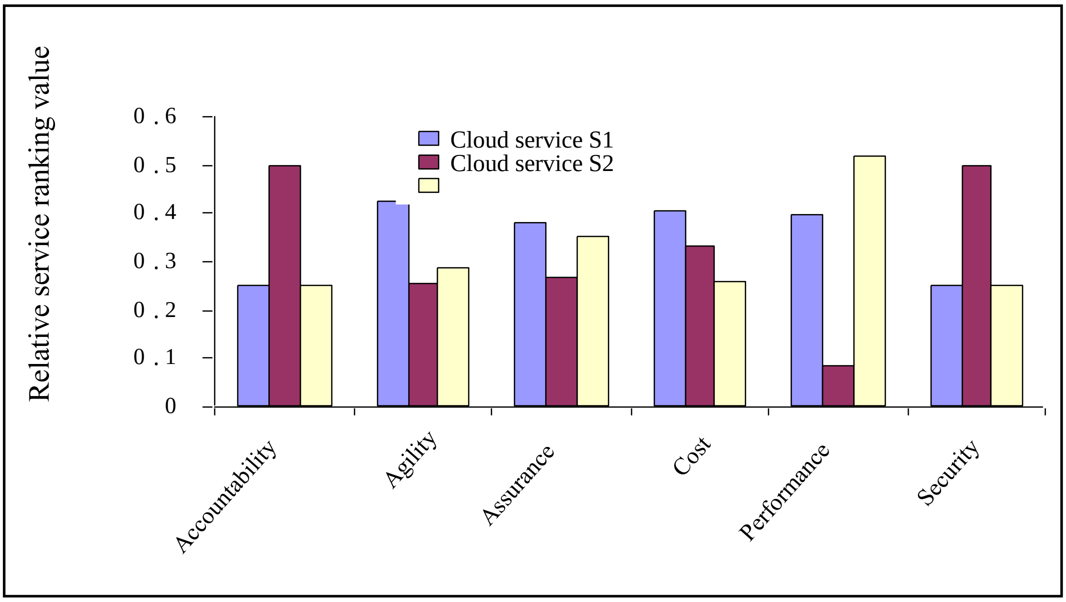


Figure 2.5: *Clouds Services Comparison Requirement for User2*

Source: Mamoun and Ibrahim (2014)

Cloud services comparison for two clients' solicitations can be pictured in Figure 2.4 and Figure 2.5 It can be observed that, for requirement1 and requirement2, Cloud provider S3 is the best in execution of the machine and the least cost, nevertheless it is one of the lowest for security for the studies of the three IaaS Cloud providers: Amazon EC2, Windows Azure, and Rackspace. Cloud services are positioned as S1 > S3>S2. Breaking down the outcomes, S3 is a decent option for mainstream researchers where security is a lower need prerequisite and information is openly accessible. Then again, Cloud provider S2 is the best in security and responsibility, which might be a key prerequisite for a client from a business association.

In the Cloud service provider choice model, the utility-based model is proposed for ideal Cloud service provider choice. It can be connected by the determination of providers in a wide range of services. For example, IaaS, PaaS, and SaaS. To evaluate the QoS among various providers meaningfully, it is vital to reliably characterize the qualities and measurements, for example, reaction time, throughput, and accessibility time. Table 2.5 and 2.6 are a case for displaying QoS properties, and also QoS measurements and their comparing quantifiable parameters, as required in this model, are shown in two Tables below (Salama et al., 2012).

Table 2.5

QoS Attributes

QoS Attribute	Description
Response Time	the time experienced by the consumer to get serviced
Throughput	the number of requests completed within a time frame
Availability Time	total time that the service is available
Service Value	denotes the economic dimension for a unit of service (1 GB storage, 1 CPU hour)

Source: Salama et al. (2012)

Table 2.5 shows the attributes and their description, the measuring in the utility-based model.

Table 2.6

QoS Dimensions and Measurable Parameters

QoS Dimension	Description	Measurable QoS Parameters
Performance	how fast and successfully the Cloud can provide the requested service	response time, worst-case execution time, throughput
Dependability	how probable the Cloud can successfully provide the requested service	Availability time, reliability level (high/medium/low)
Reputation	the creditability of the service provider	Consumers' rating, advertising messages credibility
Service Value	the economic dimension	service fees to be paid

Source: Salama et al. (2012)

Table 2.6 shows, QoS dimensions, and its corresponding measurable parameters, as needed in the mentioned model.

Table 2.7

Simulation Parameters - Services Data

Provider	**Fees**	**Uptime**	**MB/SF**	**TTR**	**ATTM**
0	74.0	97.99	165	14	0
1	98.0	97.99	184	25	86
2	81.0	99.99	133	43	44
3	71.0	98.99	151	47	83
4	98.0	97.99	105	11	71

5	75.0	99.99	177	26	43
6	76.0	97.99	182	42	28
7	100.0	99.99	156	44	92
8	77.0	99.99	116	16	46
9	81.0	99.99	188	32	82
10	74.0	99.99	123	26	89
11	75.0	98.99	111	48	71
12	98.0	97.99	186	17	63
13	81.0	97.99	142	34	66
14	81.0	99.99	180	42	65
15	73.0	98.99	120	44	73
16	80.0	99.99	149	29	10
17	81.0	99.99	176	20	6
18	88.0	97.99	102	30	6
19	92.0	97.99	185	14	41

Source: Salama et al. (2012)

Table 2.7 showing the list of the explicit quality of service, its upper or lower bounds. The consumer assigned these values, which are based on his own interest, within a range of values, previously specified to make it easier for the user.

Table 2.8

Simulation Parameters - Quality Attributes

QoS Dimension	W_j	QoS Parameters	w_j
1. Performance	0.40	1.1 Time Between Failures (MB/SF)	0.60
		1.2 Time to Recovery (TTR)	0.40
2. Dependability	0.25	2.1 Uptime	1.0
3. Reputation	0.10	3.1 Arrival Time To Market (ATTM)	1.0
4. Service Value	0.25	4.1 service fees to be paid	1.0

Source: Salama et al. (2012)

Table 2.8 includes the set of QoS dimensions and its assigned weights, as well as the measurable parameters for each dimension.

2.10.3 Linear Equation

Linear Equation of Y on X is represented by Y= mX + b where m and b are unknown constants which are known as intercept and slope of the equation. It is used to predict the unknown value of variable X when the value of variable Y is known.

A fuzzy linear regression (FLR) technique is proposed that uses real-time facts to accurately predict daily height float rate for the Bow and Elbow Rivers in southern Alberta. FLR version overall performance turned into in comparison to a non-fuzzy, errors-in-variables model (EIV).

Mean each day drift charge, with a postpone of one, two, three or seven days become used as the unbiased variable. In imposing the FLR, a unique hybrid modeling method becomes devised that handled peak waft price as probabilistic and imply daily flow fee as the possibility. Three gauge mistakes, 5%, 10% and 20%, had been examined and compared to quantify uncertainty in discovered drift charge.

The research proposed a brand new approach of computing the exceedance probability of top waft fee the use of fuzzy numbers. NSE, PBIAS and RSR and a proposed rating machine were used to evaluate and examine the methods. two exclusive calibration schemes had been used, along with a quasi-actual time gadget. The tests validated that FLR with a sooner or later lag became a superb predictor of top go with the flow rate and outperformed EIV for 2 stations on the Bow River.

A test statistics set from the floods of June 2013 in Calgary changed into used for risk assessment. The FLR results proven better flexibility and sensitivity to the flood because it approached Calgary. the bushy method became capable of seizing the height drift price for the general public of the high glide charge days, at the same time as the EIV version become unable to are expecting this information within the 95% which is confidence interval (Khan and Valeo, 2016).

Fuzzy linear regression analysis (FLRA) by way of quadratic programming (QP) is proposed for unique manufacturing structures. In our take a look at, information is provided by using certainly one of the most important casting and machining companies in Europe. The database includes items which have comparable production strategies. A fuzzy linear regression version is built by using the previously measured general times of a product circle of relatives. The version evolved is used for estimating the standard instances of the ultimate merchandise.

FLRA primarily based on QP technique enables the combination of the vital tendency of least squares and viable residences of fuzzy regression. the main factors that at once effect fashionable times are determined and used for the estimation of the fuzzy widespread times (Atalay, Eraslan, and Çinar, 2015).

Moon and Weidner (2015), studied the least squares (LS) estimator in a linear panel regression model with an unknown quantity of things acting as interactive fixed results. Assuming that the number of factors utilized in estimation is larger than the authentic range of factors inside the statistics, the establish the proscribing distribution of the LS estimator for the regression coefficients as the range of time periods and the number of pass-sectional gadgets at the same time go to infinity.

The primary result is sure assumptions; the limiting distribution of the LS estimator is unbiased of the variety of factors used inside the estimation so long as this quantity is not underestimated. The important practical implication of this end result is that for inference at the regression coefficients, one does no longer necessarily want to estimate the range of interactive fixed results always.

Castillo, Schmidt-Hieber and Van der Vaart (2015), studied full Bayesian processes for excessive-dimensional linear regression beneath sparsity constraints. The previous is a combination of factor masses at zero and non-stop distributions. Below compatibility conditions on the design matrix, the posterior distribution is proven to contract on the choicest price for healing of the unknown sparse vector, and to offer the most beneficial prediction of the response vector.

It is also proven to pick the appropriate sparse model or as a minimum the coefficients which can be significantly special from zero. The asymptotic form of the posterior distribution is characterized and employed to the construction and have a look at of credible units for uncertainty quantification.

Miyashiro and Takano (2015) concerned a method of choosing the best subset of explanatory variables in more than one linear regression models. Goodness-of-in shape measures, as an instance, adjusted R^2, AIC, and BIC are generally used to evaluate a subset regression version. even though variable selection with regard to those measures is generally done with a stepwise regression technique, it does no longer continually offer the nice subset of explanatory variables.

The endorse combined integer 2d-order cone programming formulations for selecting the quality subset of variables with respect to adjusted R^2, AIC, and BIC. Computational experiments display that, in phrases of these measures, the proposed formulations yield better answers than those furnished by common stepwise regression strategies.

2.10.4 Linear Models

Utilizing the customary linear model to execute variable determining can be performed successfully now and again, giving the reaction to applicable parts around. Thus, a methodology is recommended that is based on ranking and taking into account empirical correlations between the reaction variable and segments of the illustrative vector. The technique used is not forecast based and can recognize variables that are powerful, however, it is not unequivocally part of a prescient model.

In addition, investigate the strategy's execution thoroughly and the reproduced information, and give a hypothetical contention illustrating its legitimacy. The strategy can likewise be utilized as a part of conjunction with, as opposed to as an option to traditional expectation-based variable choices, by giving a preparatory "enormous measurement diminishment" venture as a prelude to utilizing selective strategies (e.g., the adaptive lasso) that do not generally adapt well to high measurements.

The assortment of linear model-based methods has been proposed for variable determination. In this methodology, it is contended that a reaction variable,Y_i, may be expressible as a linear shape in a long p-vector, X_i, of informative variables, in addition of an error, that is, $Y_i = \alpha + \beta_1 X_i + \cdots + \beta_p X_{ip} + error$ (Hall and Miller, 2012).

Natural manageability of a supply chain relies on the buying procedure of supply chain individuals. The majority of the prior models have focused on cost, quality, lead time, and so forth, however, issues were not sufficiently given significance to carbon outflow for provider assessment. As of late, there is a developing weight on store network individuals for diminishing the carbon emanation of their supply chain. This study shows a coordinated methodology for selecting the fitting provider in the production network, tending to the carbon emanation issue, utilizing fuzzy-AHP and fuzzy multi-objective linear programming.

Fuzzy AHP (FAHP) is used first to analyze the weights of the numerous elements. The considered variables are cost, quality dismissal rate, late conveyance rate, greenhouse gas outflow and interest. The weights of the various elements are utilized as a part of fuzzy multi-objective linear programming for provider determination and quantity assignment. An outline with an information set from a practical circumstance is displayed to show the adequacy of the proposed model. The proposed methodology can deal with practical circumstance when there is data dubiousness identified with inputs (Shaw, Shankar, Yadav, and Thakur, 2012).

The proposal of an empirical Bayes method for variable selection and coefficient estimation in linear regression models. The method is based on a particular hierarchical Bayes formulation, and the empirical Bayes estimator is shown to be closely related to the LASSO estimator. Such a connection allows us to take advantage of the recently developed quick LASSO algorithm to compute the empirical Bayes estimate and provides a new way to select the tuning parameter in the LASSO method.

Unlike previous empirical Bayes variable selection methods, which in most practical situations can be implemented only through a greedy stepwise algorithm, our method gives a global solution efficiently. Simulations and real examples show that the proposed method is very competitive in terms of variable selection, estimation accuracy, and computation speed compared with other variable selection and estimation methods. In addition, it considered the issue of variable determination and coefficient estimation in the regular typical linear equation model have *n* perceptions on a dependent variable *Y* and P predictors ($x_{1\,,}\ x_{2\,,}\ x_{3\,,} x_p$), and Y= Xβ + ϵ, (Harrell, 2015).

2.11 PROCESSING SERVER

Many companies resort to complete their operations on separating server from server applications, in order to improve performance and speed of data processing. Azure Provider Company is providing one technique that is used to process data called worker roles.

2.12 AZURE WORKER ROLES

The MPS system will be using the modern technology of Windows Azure, called worker roles.

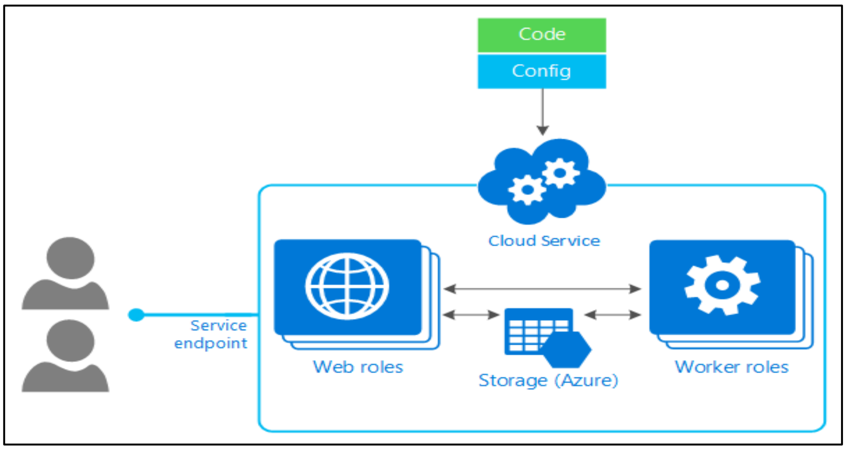


Figure 2.6: *Worker-and-Web-Role*

Source: Cloudmonix (2017)

Figure 2.6 displayed the processes for Worker Role which is the modern technology of Windows Azure. The explanation in details as the following:

1. What are Web and Worker Roles in Microsoft Azure?

"Before delving into the difference between Web Role and Worker Role. What is CLOUD SERVICE Role, and its two varieties: Web Role and Worker Role actually are in Azure.

What is an Azure CLOUD SERVICE Role?

In Azure, a CLOUD SERVICE Role is a collection of managed, load-balanced, platform-as-a-service virtual machines that work together to perform common tasks. Cloud service roles are managed by Azure fabric controller and provide the ultimate combination of scalability, control, and customization.

What is a Web Role?

Web Role is a CLOUD SERVICE role in Azure that is configured and customized to run web applications developed on programming languages/technologies that are supported by Internet Information Services (IIS), such as ASP.NET, PHP, Windows communication foundation and fast CGI.

What is a Worker Role?

Worker Role is any role in Azure that runs applications and services level tasks, which generally do not require IIS. In Worker Roles, IIS is not installed by default. They are mainly used to perform supporting background processes along with Web Roles and do tasks such as automatically compressing uploaded images, run scripts when something changes in the database, get new messages from queue and process, and more.

Differences between Web and Worker Roles

The main difference between the two is that a Web Role support and runs Internet Information Services (IIS), while an instance of a Worker Role does not. As being deployed and delivered through Azure Service Platform, both can be managed in the same way and can be deployed on a similar Azure Instance. In most scenarios, Web Role and Worker Role instance work together and are often used by an application simultaneously.

For example, a web role instance might accept requests from users, then pass them to a worker role instance for processing" (Cloudmonix, 2016).

2.12.1 AWS Lambda

"AWS Lambda is a compute service where one can transfer the code to AWS Lambda, then the service can run the code by using AWS infrastructure. Afterward, upload the code and build a *Lambda function.* AWS Lambda is responsible for taking care of managing and provisioning the servers that allow the code to run. Using AWS Lambda is as following:

1. In event-driven, compute service where AWS Lambda runs the code in response to events, such as changes to data in an Amazon S3 bucket or an Amazon Dynamo DB Table.

In a compute service to run the code in response to HTTP requests using Amazon API Gateway or API calls made using AWS SDKs.

AWS Lambda runs a code on a high-availability compute infrastructure and performs all of the procedures of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code monitoring and logging. All one needs to do is to supply the code in one of the languages that AWS Lambda supports (currently Node.js, Java, and Python).

AWS Lambda executes the code only when needed and scales automatically, from a few requests per day to thousands per second. With these capabilities, it can use Lambda to easily build data processing triggers for AWS services like Amazon S3 and Amazon DynamoDB, process streaming data stored in Amazon Kinesis, or create your own backend that operates at AWS scale, performance, and security (AWS 2016).

2.12.2 Using AWS Lambda

AWS Lambda is an ideal computing platform for many application scenarios, provided that one can write the application code in languages supported by AWS Lambda (which are Node.js, Java, and Python), and run within the AWS Lambda standard runtime environment and resources provided by Lambda.

When using AWS Lambda, users are responsible only for their codes. AWS Lambda manages the compute fleet that offers a balance of memory, CPU, network, and other resources. This is in exchange for flexibility, which means the user cannot log in to compute instances, or customize the operating system or language runtime. These constraints enable AWS Lambda to perform operational and administrative activities on user's behalf, including provisioning capacity, monitoring fleet health, applying security patches, deploying the code of the user, and monitoring and logging Lambda functions (AWS 2016).

If the user requirements to manage their own computing resources, Amazon Web Services also offers other compute services to meet the user requirements.

1. Amazon Elastic Compute Cloud (Amazon EC2) service offers flexibility and a wide range of EC2 instance types to choose from. The users have the option to customize operating systems, network and security settings, and the entire software stack, but they are responsible for provisioning capacity, monitoring fleet health and performance, and using Availability Zones for fault tolerance.

Elastic Beanstalk offers an easy-to-use service for deploying and scaling applications onto Amazon EC2 in which user retains ownership and full control over the underlying EC2 instances (AWS 2016).

2.12.3 Worker Dynos

1. Overview

“Many tasks that should be processed as background jobs are fetching data from remote APIs, reading RSS feeds, resizing images, and uploading data to S3. The web process that requests the job schedules it for processing and immediately returns it to the client. The client can then request for updates to see when their job is complete.

Consider the example of a web-based RSS reader. An application like this will have a form where users can submit a new feed URL to be read. After a delay, the user will be taken to a page where they can see the contents of the feed. A simple but non-scalable way to do this would be to retrieve the RSS from the third-party site directly inside the web request (Heroku Dev Center 2017).

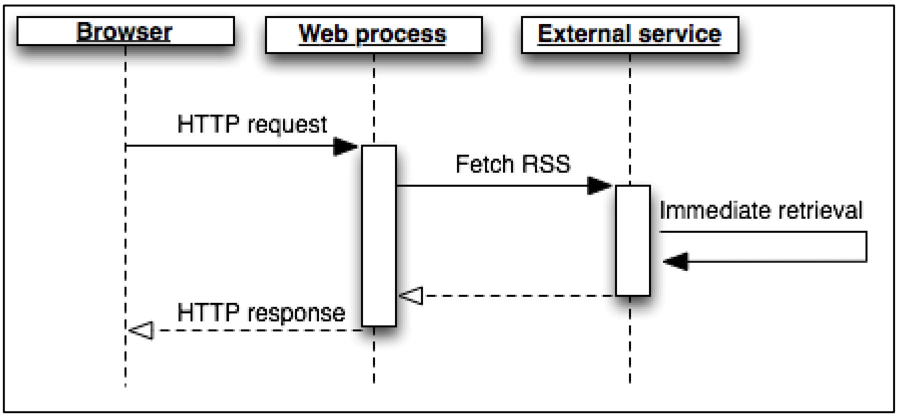


Figure 2.7: *Worker Dynos 1*

Source: (Heroku Dev Center 2017)

Figure 2.7 shows the fetching data from external sources will sometimes happen in as little as a few hundred milliseconds. Other times it may take several seconds. If the feed's server is down, it could hang for 30 seconds or more until the request times out.

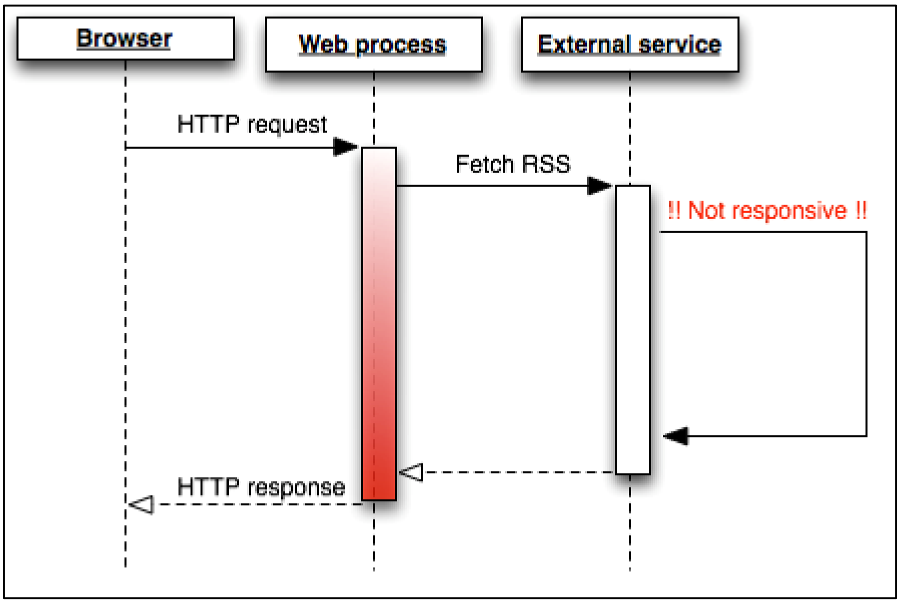


Figure 2.8: *Worker Dynos 2*

Source: (Heroku Dev Center 2017)

Figure 2.8 shows how tying up your application processes during this time prevents it from handling other requests and results in a very poor user experience. This may not manifest itself under low load but as soon as the application has multiple simultaneous users, response times will become more and more inconsistent and may experience H12 or other error statuses. As a result, users' application will not be able to scale very well.

Approach

A more predictable and scalable architecture is to keep the high-latency or long-running work in a background process separate from the web layer and immediately respond to the user's request with some indicator of work progress.

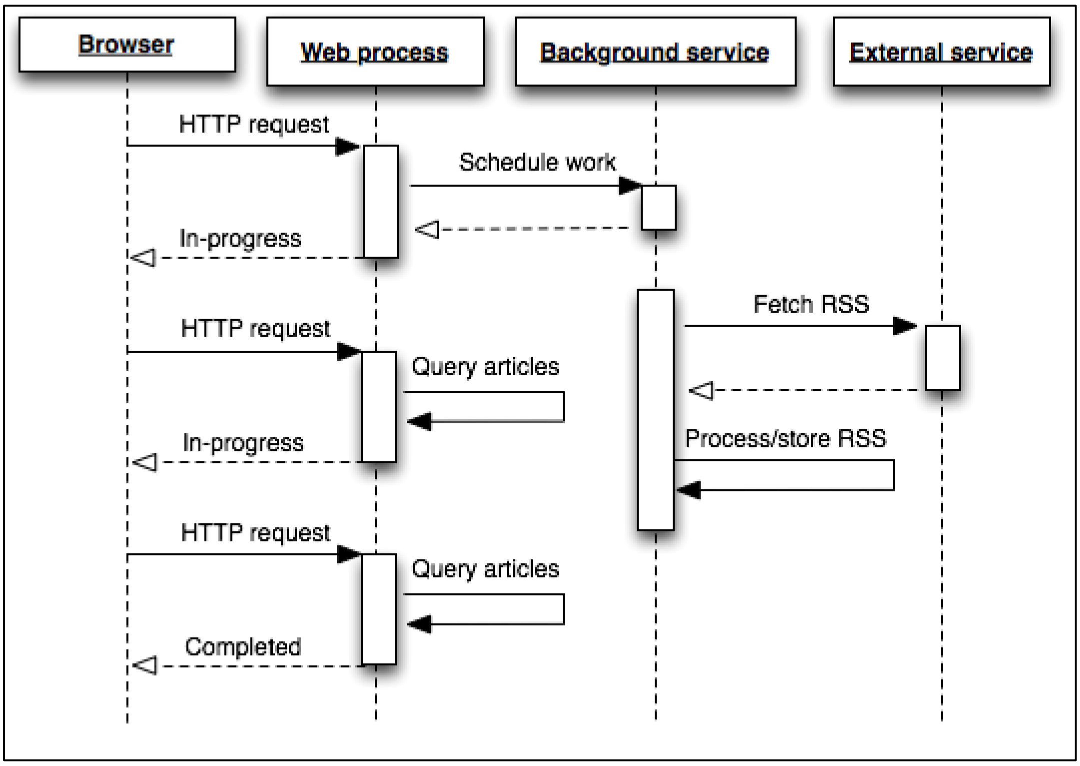


Figure 2.9: *Worker Dynos Approach*

Source: (Heroku Dev Center 2017)

Figure 2.9 show, one or more background services running separate from the web process and not serving web requests, will read items from their work queue one by one and do the work asynchronously. The results will be placed in local storage (DB, Memcached etc…) when finished.

Judging by the sequence diagram, the background approach may not appear to be of any benefit as there are now more client (HTTP) requests than before. This is true but it masks the real gain. While the browser may have to make more than one request to retrieve the backgrounded work the benefit is that these are very low-latency and predictable requests. Any single user request will no longer have to be in waiting, or hanging, for a long-running task to complete.

Handling long-running work with background workers has many benefits. It avoids tying up the web dynos, preventing them from serving other requests, and keeps the site snappy. It can now monitor, control and scale the worker processes independently in response to site load. The user experience is also greatly improved when all requests are immediately served, even if only to indicate the current work progress (Heroku Dev Center 2017).

Process model

Heroku allows specification of an application-specific process model, which can include background workers retrieving and processing jobs from the work queue. Below is an example of a profile for Clojure application that has both a web process type and a process type for handling background (Heroku Dev Center 2017).

Table 2.9

Measuring Services Models in Cloud Computing

Methods	Models	Services	Contribution	Research Work
Cloud Ranking	Ranking Approaches for Cloud services Based on Quality of Service	1- Accountability 2- Agility 3- Cost, 4- Performance 5- Assurance 6- Usability 7- Security	The existing approaches for ranking Cloud services are analyzed. The overall performance of each method is presented by reviewing and comparing of them. Finally, the essential features of an efficient rating system are indicated.	(Jahani and Khanli ,2014)
	A model for selection of best CLOUD SERVICE provider using ranked voting method	1- Reliability 2- Availability 3- Security 4- Data centers 5- Cost 6- Operating Systems Support 7- Platforms Supported 8-Virtualization Technique 9- Customer support facility 10- Response Time 11- Throughput and Efficiency 12- Capacity	Identifies QoS metrics and defines it in such a way that user and provider both can express their expectation and offers respectively into quantified form. A dynamic and flexible model using Ranked Voting Method is proposed which takes requirement of the user as an input and provides the best provider as output.	(Baranwa and Vidyarth,2014)
	Model for Ranking and Reservation the Cloud Services	1-cost, 2-performance 3-assurance 4-security 5- usability, 6-agility, 7-accountability	Develop a model to pick the best Cloud providers from a group of available providers based on many Quality-of-Service ("QoS") criteria attributes in order to enhance efficiency, accuracy and service provisioning. Also, provide a classification for QoS criteria attributes, which is divided into five main attributes and sub-attributes, helping in enhancing ranking process in the provided model.	(Mamoun and Ibrahim,2014)
	Achieve Better Ranking Accuracy Using Cloud Rank Model for Cloud Services	1- Response time 2- Throughput 3- Failure probability	A Cloud Rank model which predicts the QoS ranking directly without predicting the corresponding QoS values. In addition, the model provides an accurate ranking but the QoS values are same in both algorithms so, an optimal VM allocation policy is used to improve the QoS performance of Cloud services and it also provides better ranking accuracy than Cloud Rank2 algorithm.	(Subha et al.,2013)
	QoS based Enhanced Model for Ranking CLOUD Providers	1-Response Time 2- Elasticity 3- Cost	Comparing various QoS. And rank them according to the performance of four Cloud providers. Focus on specific	(Kumar and Agarwal,2014)

		4- Availability	attributes the disadvantage is ignoring the preference's user or basic services which meet their needs	
	An enhanced QoS Architecture based Model for Ranking of Cloud Services.	1- Cost 2- Performance 3- Assurance 4- Security 5- Usability 6- Agility 7- Accountability	A novel model for ranking and advanced reservation of Cloud services is proposed which is based on a set of Cloud Computing specific performance and a Quality of Service (QoS) attributes. It provides an automatic best fit and a guaranteed delivery.	(Saravanan and Kantham ,2013)
	QoS ranking prediction for Cloud services	1- Response time 2- Throughput 3- Failure probability	A QoS ranking prediction model for Cloud services by taking advantage of the past service usage experiences of other consumers. In addition, the model requires no additional invocations of Cloud services when making QoS ranking prediction. Two personalized QoS ranking prediction approaches are proposed to predict the QoS rankings directly.	(Zheng et al.,2013)
Service Measurement	A service brokering and recommendation mechanism for better-selecting Cloud services	1-Application requirements 2-Business expectations 3-Capacity provisioning 4-Cloud information collection and process	A ranking model, which compares the providers on various QoS criteria and ranks them according to their performances. Also, perform a consistency evaluation of the providers by comparing the entropy and hyper entropy of delivered QoS values to past users.	(Gui et al., 2014)
	A novel Trust Model for Selection of Cloud Services Provider .	1-Security Measures 2- Compliance with regulatory body's standards 3-Down time 4-Up time 5- Customer support: 6- Performance of a Specific Service 7- Latency (Response time) 8- Fault Tolerance Capability 9- Application update frequency 10- Customer service experience	A model which will help the CLOUD SERVICE Users (CSUs) in finding out the efficient and trustworthy CLOUD SERVICE Provider (CSP) on the basis of data taken from regulatory authorities, the performance of CSP in last one year and feedbacks taken from the customers. Moreover, the model is flexible enough to be customized according to the precedence level of parameters for the CSUs.	(Naseer, Jabbar and Zafar, 2014).
	CLOUD SERVICE Selection Based on the Aggregation of User Feedback and Quantitative Performance Assessment	1- Availability 2- Elasticity 3- Response time 4- Cost 5- Performance	A novel model of Cloud service selection by aggregating the information from both the feedback from Cloud users and objective performance analysis from a trusted third party. Based on this model, proposed a model, which supports Cloud service selection approach.	(Qu, Wang and Orgun, 2013)

	Integrated QoS utility-based model for Cloud Computing service provider selection	1- Response time 2- Throughput 3- Availability 4- Reputation 5- Performance 6- Dependability 7- Service value	A mathematical model addressing the Cloud service provider selection optimization problem based on QoS guarantees. The proposed model efficiently matches with the characteristics of market-oriented platforms covering a wide range of service provider selection problems.	(Salama et al., 2012).
	Towards a Trust Management System for Cloud Computing	1- SLA 2- Compliance 3- Portability 4- Interoperability 5- Geographical Location 6- Customer 7- User Feedback 8- Performance 9- Security	A multi-faceted Trust Management (TM) system architecture for a Cloud Computing marketplace. This system provides means to identify the trustworthy Cloud providers in terms of different attributes.	(Habib, Ries, and Muhlhauser, 2011)
Linear Equation	Using generalized correlation to effect variable selection in very high dimensional problems	1- Cardiomyopathy microarray data 2- Acute leukemia microarray data 3- Breast tumor X-ray data	An approach based on ranking generalized empirical correlations between the response variable and components of the explanatory vector. This technique is not prediction-based and can identify variables that are influential but not explicitly part of a predictive model. In addition, explore the method's performance for real and simulated data, and give a theoretical argument demonstrating its validity	(Hall & Miller, 2012)
	Provider selection using fuzzy AHP and fuzzy multi-objective linear programming for developing low carbon supply chain	1- Cost 2- Quality rejection percentage 3- Late delivery percentage, 4- Greenhouse gas emission and demand	An integrated approach for selecting the appropriate provider in the supply chain, addressing the carbon emission issue, using fuzzy-AHP and fuzzy multi-objective linear programming. Fuzzy AHP (FAHP) is applied first for analyzing the weights of the multiple factors	(Shaw et al., 2012)
	Integrated fuzzy multi-criteria decision-making method and multi-objective programming approach for provider selection and order allocation in a green supply chain	1- Capacity 2- Speed 3- Performance	An empirical Bayes method for variable selection and coefficient estimation in linear regression models. The method is based on a particular hierarchical Bayes formulation, and the empirical Bayes estimator is shown to be closely related to the LASSO estimator.	(Yuan and Lin, 2005)

2.13 SUMMARY

In prior studies, the researchers have approved the optimal choice for defining the best service provider which is a common problem faced by the user in Cloud Computing. The user is in a critical situation to decide which provider can satisfy their specific needs. There is no comprehensive model that can help the users to take crucial decision for the best service provider of Cloud Computing. This common problem is still an active thesis domain where many models have been offered to solve this problem. Models based on measuring the quality of services as well as defining specific services are considered more suitable for users according to their overviews. The proposed models, methods and algorithms are investigated. On the other hand, MPS model is based on measuring preferred service of the user. Finally, examination of Cloud selection service on this problem is used in this thesis to classify the service provider based on priority user's needs. While others models focus on produced services from Cloud providers which at the end leads to the satisfaction of a user.

CHAPTER THREE

RESEARCH METHODOLOGY

The methodological details of this study are discussed in this chapter. A number of resolved issues and solutions encourage the work in this thesis and moves on the suggestions to work out on the mentioned issues as discussed in the previous study. It presents the executed steps in achieving the stated research objectives. The present overview findings obtained from the theoretical study as in Chapter Two will be the basis to pre-research through the development and evaluation of a new model of Cloud provider selection called measuring preferred service (MPS), based on the priority of the user's cloud computing service.

3.1 INTRODUCTION

The research methodology consists of four phases that will be discussed in details in this chapter. Section 3.2.1 presents the details of the experimental design and a summary of the chapter followed in Section 3.3.

In prior studies and existing models, the selecting and comparing of Cloud provider selection focus on performance of QoS or specific services, which is attained from a provider in order to help users to take the right decision. Therefore, the service measured without taking into account the viewpoint of users who are indeed utilizing the Cloud services and interested in the reputation of the provider.

Measuring preferred service (MPS) can allow users to evaluate the provider companies as a first step to select the best service provider based on the user's priorities. In addition, it will be easy for users to search and request for the basic and main services to be the basis for the provider selection.

The MPS model provides a complete view of the preferential services by users to choose the best service provider. The architecture of MPS consists of three phases which are input, processing, and output. In the input phase, the service provider data and the values of the service preferences specified by the user, there are two services, which are functional services and non-function service.

1. Function services: SSD, CPU, RAM, Bandwidth, and Cost.
2. Non-function service: reputation, which includes the age of company and availability.

Linear equation, which measures the preferred services of users, approved that MPS model achieved the goals and help users to define the provider who can provide services that meet their needs. The Linear equation gives a distinct solution, which applies to the latest and new technologies of Measuring Preferred Services (MPS), the reputation of Cloud provider, and Multi-objective and constraints optimization to satisfy the requirements of Cloud user. Through the methodology that will be one of its objectives to provide easy user use, semantic interoperability, and the precise measurement process, to find and select a more reliable provider, control and automate negotiation strategy. Figure 3.1 depicted the research methodology and its phases.

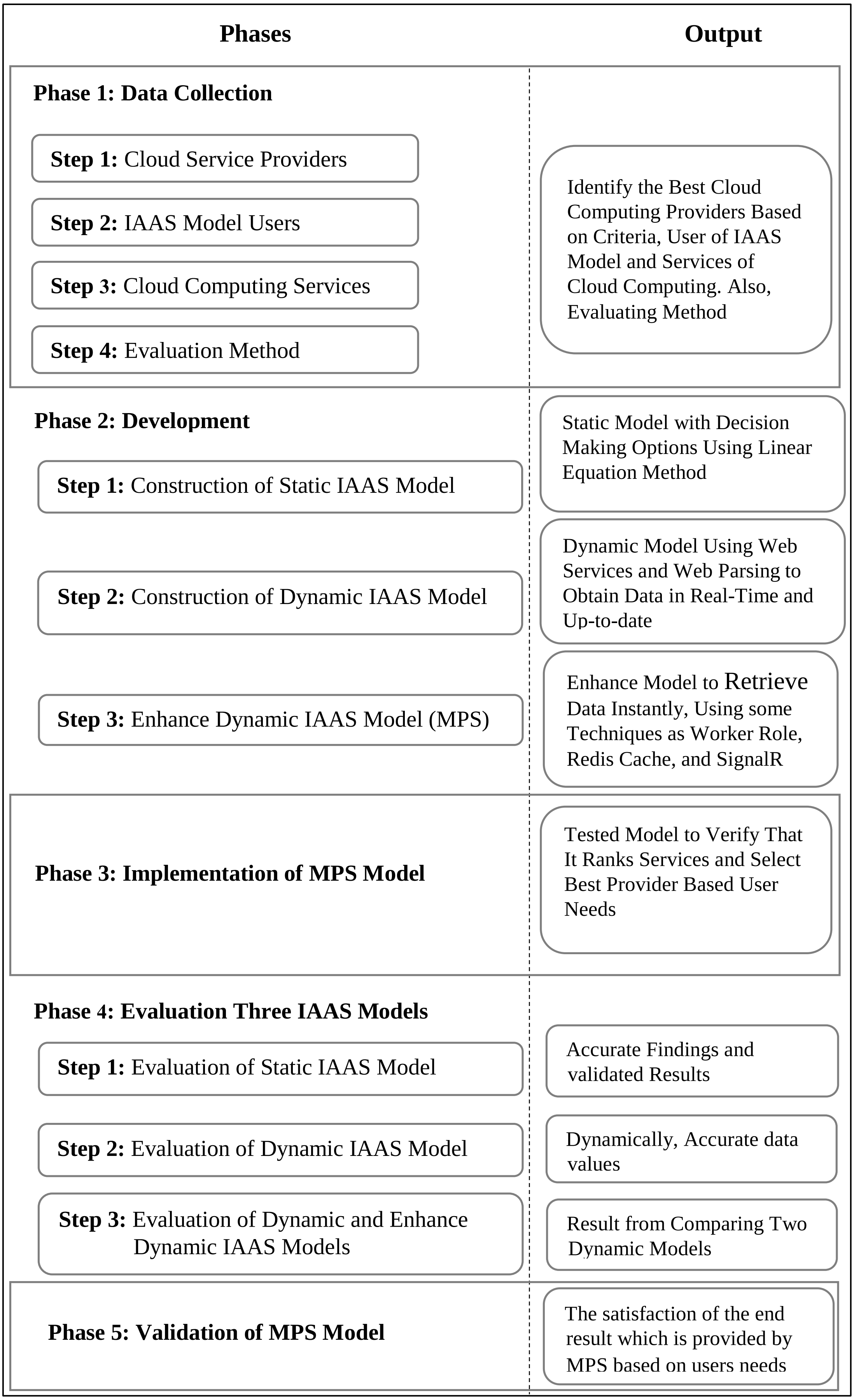


Figure 3.1: *Research Methodology Framework*

Figure 3.1 highlighted the Four phases of this research, these phases are:

- **Phase 1: Data Collection**
- **Phase 2: Development**
- **Phase 3: Implementation of IAAS Model**
- **Phase 4: Evaluation Three IAAS Models**
- **Phase 5: Validation of MPS Model**

3.2 PHASE ONE: DATA COLLECTION

Data collection in this thesis is focused on the primary and secondary data. This phase includes four steps:

Step 1: Identify Cloud Providers

Step 2: Identify IAAS Model Users

Step 3: Cloud Computing Services

Step4: Evaluation Method

In this phase, investigate and achieve a comprehensive data collection related to Cloud Computing, in a conference paper, journals, and website to achieve all steps and preparing for the next phase.

3.2.1 Step 1: Cloud Providers

In this step, a list of all Cloud providers is defined and obtained, which will enable us to know the most important companies, physical capabilities, and characteristics.T here are more than a thousand providers but the four best providers will be chosen based on ranking in Cloud Harmony website (Cloud Harmony, 2014).

3.2.2 Step 2: IAAS Model Users

In this step, each Cloud Computing users are defined to find out who are the users of IAAS model, which is more popular among users into research and higher Computing areas such as universities and customer service companies. In addition, users are in charge of overseeing more: applications, information, runtime, middleware, and O/S. Providers still oversee virtualization, servers, storage, hard drives, and networking. What users get with IaaS is infrastructure in which they can install any required platforms. Additionally, users are in charge of updating the platforms if new versions are released.

3.2.3 Step 3: Cloud services

The services of Cloud Computing are various; there are functional and non-functional services depending on the user's needs and vision such as availability, security, reliability, operating systems support, data centers, cost and platforms supported, virtualization technique, customer support facility, throughput, efficiency, capacity and response time, and more. However, we will be focused on essential and functions services for any virtual private server (VPS), which are SSD, CPU, RAM, bandwidth, and cost and additionally, non-function service: reputation, which includes the age of company and availability.

There are definitions of services, which are measured, in proposed model as following in Table3.1:

Table 3.1
Definition of Services of MPS Model

Services	Description
Cost	The fees for services.
Availability	The failure time of receiving service in an hour.
CPU	A central processing unit (CPU) is an important part of almost every computer and measured in hertz.
RAM	A random-access memory device allows data items to be read and written in the same amount of time and measured in bytes.
Bandwidth	The rate of data transfer in bits per second (bit/s).
SSD	A solid-state drive (SSD) is a data storage device, typically used in a computer and measured in Bytes.
Reputation	The age of Cloud Computing company, which provided the service to the user of Cloud is computed per year.
Service Volume	Denotes the size and volume
Service Value	The cost of each quantity attribute.
Percentages of each service	Amount in each hundred that belongs the function services

3.2.4 Step4: Evaluation Method

The linear equation for measuring preferred service (LEMPS) is procured by two processes, the first process is finding out the value of isolated service equation (VISE) according to the calculation of the percentage of priority of services user and the second process is finding out the summation values of total services equation (VTSE) to produce the final result.

1. **Measuring of Priority of Services in MPS**

The formula of measuring preferred service is not a complex equation; the linear equation is used to compute the values of services. An example is explained below to show the ranking of companies in Cloud Computing according to the preferences services of user and reputation of Cloud provider. The experimental user has many choices when calculating the percentage of preference of services. The potential user can select services according to their need and priority and the final results produced by the VTSE as shown are based on provided services of the four mentioned companies.

2. **How to Select a Cloud Provider Given its Services?**

By assumption, the user A has the following demands: The user browses catalog service and finds that there are N Cloud providers C1, C2 … CN who can provide service that meets the user requirements.

When the user submits the list of services providers C1, C2 … CN with the list of requirements according to its priority to MPS System. The System forwards the list of resource providers and requirements to the MPS and retrieves the trust values of N Cloud providers from the MPS system. Then it sorts the provider companies based on the trust values. Here is a sample of four Cloud companies P1, P2, P3 P4 and their services.

The offered services of P1 is better than P2. On the other hand, the trusted values of P1 is better than P2. It requires more thinking to make an optimal decision. This is done by MPS model in consultation with the experimental user through the MPS System. Assume the experimental user decides to select P1. After the final results and trusted values are prepared by MPS model, based on the MPS, Cloud provider P1 provided the service and delivers the processed data to the user through the MPS System. After the services are submitted, the MPS system updates the trusted values sorting for companies with the values of services. The mathematical equation below is explaining the process of Computing functions services. The experimental user has many optimal choices to select the best service provider among GCE, Rackspace, AWS and Windows Azure by comparing the final results. The comparison among the maximum volume of services in each company and the experimental use through the differences results according to the user preferences, who must choose the best provider who has the best service (highest volume takes the highest value).

For example, if the user submits the percentages services in which their summation must be a hundred percent. For selection, percentage takes the highest value among the maximum values that are mentioned in provider companies. In contrast, the lowest value among the maximum values takes one percent. In the terms of equation to compute VPS services, they are computed by the following algorithms; the result of subtracting the largest volume of service from the smallest volume for each isolated service in every provider company. The volume of two services represents the variable value (Temp).

The result of the subtraction is divided by the constant value (priority of service) which the experimental user must define the priority of services in percentage, where the MPS model will force them to select from 1% which represents the lowest to 100% which represents the highest value. There are three levels in the selection system, which are low, medium and high. These levels are used for the user's preference and evaluate services based on its requirements. Each level represents a percentage, where the lowest level represented 34%, 67 % represents medium and the high level is 100%.

Measuring preferred service that is selected is independent. The linear equation can be defined by:

Let *r* be Pearson's *r*, $\boldsymbol{s}_{\mathbf{x}}$ the standard deviation and $\overline{\boldsymbol{x}}_{\boldsymbol{x}}$ the mean of all the numbers on the x-axis, $\boldsymbol{s}_{\mathbf{y}}$ the standard deviation and $\overline{\boldsymbol{x}}_{\boldsymbol{y}}$ the mean of all the numbers on the y-axis. Then the slope will be $\boldsymbol{m} = \boldsymbol{r}\frac{\overline{x}_y}{\overline{x}_x}$ and y-intercept $\boldsymbol{b} = \overline{\boldsymbol{x}}_{\boldsymbol{y}-\boldsymbol{m}}\,\overline{\boldsymbol{x}}_{\boldsymbol{x}}$, for linear trend line. Thus, formula (1) is:

$$\mathbf{y} = \mathbf{mx} + \mathbf{b} \qquad (1)$$

Professional researchers in different researchers use this equation as formula (1) (Chang, Lai, and Huang, 2012a; Adamuthe, Tomke and Thampi, 2014; Gupte, and Wang, 2015). e.g., a geographical feature or a set of points on a graph slope, or deviation. Scientists often apply trend lines, or best fit lines, to their data after they graph it on an x, y plot. The idea of a trend line is to reveal a linear relationship between two variables, x and y, in the y = mx + b form. Deriving the linear equation that links two variables allows scientists to extrapolate, or predict, how one variable will change given any change in the other. Most of the time, a line cannot simply be drawn through real life data because it will rarely fit neatly. A statistical tool is required to calculate the best fit line accurately. Regression analysis of a large dataset will easily fill both sides of a paper with numbers, so if you can find a program to do it, it saves a lot of time.

3.3 PHASE TWO: DEVELOPMENT

In this phase, there are five steps, which are: 1. Construct IAAS model provides, 2. Construct IAAS model for providers and users, 3. Evaluate best combination option IAAS model according to user requirements, 4. The dynamic construct of model MPS using web service and 5. Evaluating static and dynamic services values. These steps help to construct Enhance Dynamic IAAS Model (MPS).

3.3.1 Step 1: Construction of Static IAAS Model

In this step, each data values of the four Cloud Service Providers (CSP), which are GCE, Rackspace, AWS and Windows Azure, obtained in a specific time. Providing these data to construct static IAAS model is the first phase of construction. It will benefit the user who wants high-quality decisions with strong support for follow through, willing to invest time in order to create a proposal or plan, and will benefit from consensus decision-making. Involving all group users in the debate of issues and making decisions together is a powerful process. Additionally, finding out a model for users with decision-making options is very important. Linear equation matrix is used to find the percentages of each service, which is then collected and ranked based on the user's requirements.

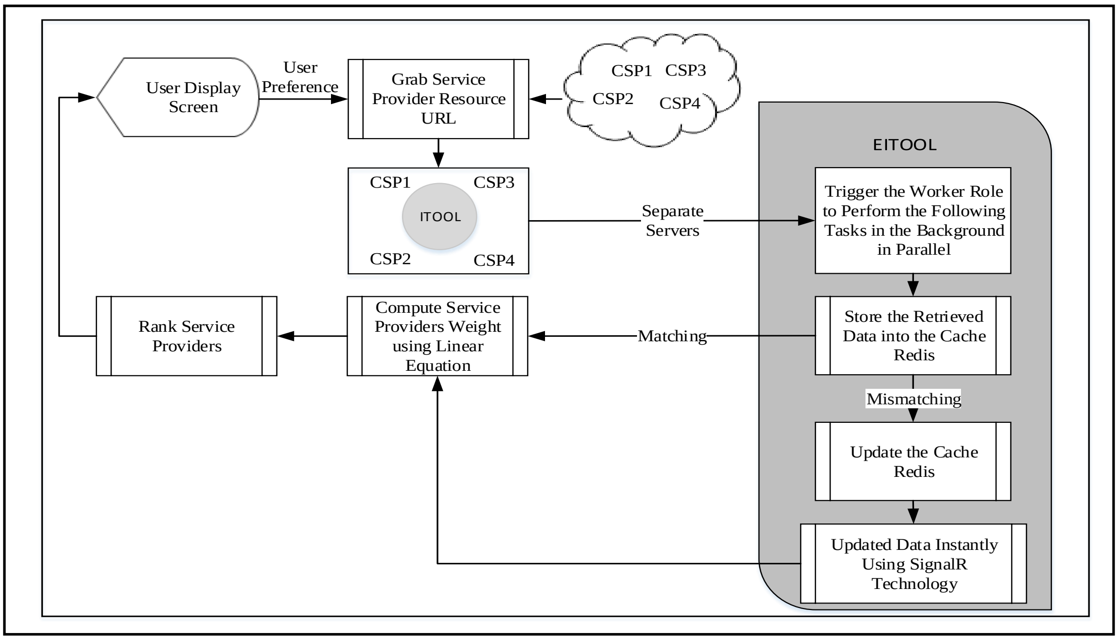


Figure 3.2: *Static Infrastructure as a Service Model (SIAAS)*

- **Analyzing of quality of static model**

There are four steps display in a short, static model which is achieving its task successfully through analyzing the collective data and measuring the value of services, to accomplish the final results and achieve the goal in this thesis, which is to produce a new model called (MPS) to select best Cloud provider based on measuring preferred service of user, Figure 3.2, displayed the construction of Static IAAS Model process as the following:

1. Step 1: Collecting the essential services according to the user orders and needs.
2. Step 2: Calculate the overall ranking of functional services (preferred service of the customer). Total values of importance are calculated by using equation is given below in section 4.3.
3. Step 3: Selecting provided services in terms of decision criteria of customers which are mentioned in this thesis. Providers of four huge companies in this thesis are compared to each other in terms of decision criteria of customers, the best service provider that meets the needs of customer in MPS model (age of companies, a viability, cost etc.), through gathering data from websites which each company produce its services with volume and cost for last year (2014).
4. Step 4: Sorting the result according to preferences services of user lead to the best service provider. The priority services are obtained by Computing the importance of services. The specific services are as shown below in Table 4.1:

Table 3.2

VPS Services for Prestigious Companies in Cloud Computing

VPS	Azure	AWS	GCE	Rackspace
SSD	800 GB	1000 GB	1500 GB	1,200 GB
RAM	112 GB	60 GB	104 GB	120 GB
CPU Core	16 cores	36 cores	16 Cores	32 Cores
Bandwidth	2000 Mb/s	2000 Mb/s	10 Mb/s	524 Mb/s
Cost/ Month	$1271.33	$1828.48	$923.47	$4529.79

Table 3.2 summarizes the data for each Virtual Private Server (VPS) obtained from the published data in the website of Cloud Computing providers regarding perceptions and adoption of selecting services of Cloud Computing. The collected data for SSD, RAM, CPU Core, bandwidth and cost of various services providers are mentioned in Table 3.2.

These services are (RAM, SSD, CPU, and Bandwidth) measured after the maximum and minimum volume are defined according to the preferred service of users and their application. The cost is the most important service, where the customer asks for a low cost which is normal but at the same time, the increasing or decreasing of price depends on the performance and reputation of the provider.

3.3.2 Step 2: Construction of Dynamic IAAS Model

In this step, the second phase of construction will be based on the first phase. Each data values for the service provider is dynamic. Data is retrieved and grabbed using web services. To get data from each provider who are not providing a standard data or any web services for grabbing data, will need some intelligent features and abilities in the MPS system. An intelligent tool will be used to grab data automatically. This tool will go to the data page on each site, do intelligence searching for the highest value plan and record it in a standard way. This allows changing arbitrary data to a standard data that can be processed. The tool will make a lot of loops and conditions to examine the values of each row on each Table, inside each provider site.

The intelligence tool algorithm is doing the following:

1. Fetching the pricing web page and detect all the Tables on that page.

Detecting the needed pricing Tables using regular expressions to find the needed data.
Looping over all of the Tables and data sets until reaching the Table with the highest values in CPU, RAM, SSD, bandwidth, and cost.
The highest value will be stored in a new variable and will be populated using standard way JSON.
To process all the HTML tags, using special libraries to do that. Importing the data in text format, and needs to know where a Table and a cell is.

Figure 3.3 clarify Comprehensive procedures for dynamic IaaS Model (DIAAS) from the request to initiate a connection to get service provider UR. Then, the processes for achievement in all tasks in order to find the location of the required values and retrieve it based on the location of the Table containing those values, until the use of linear equation algorithm to calculate the Cloud provider's rankings. Finally, display the Cloud providers' rankings to the user.

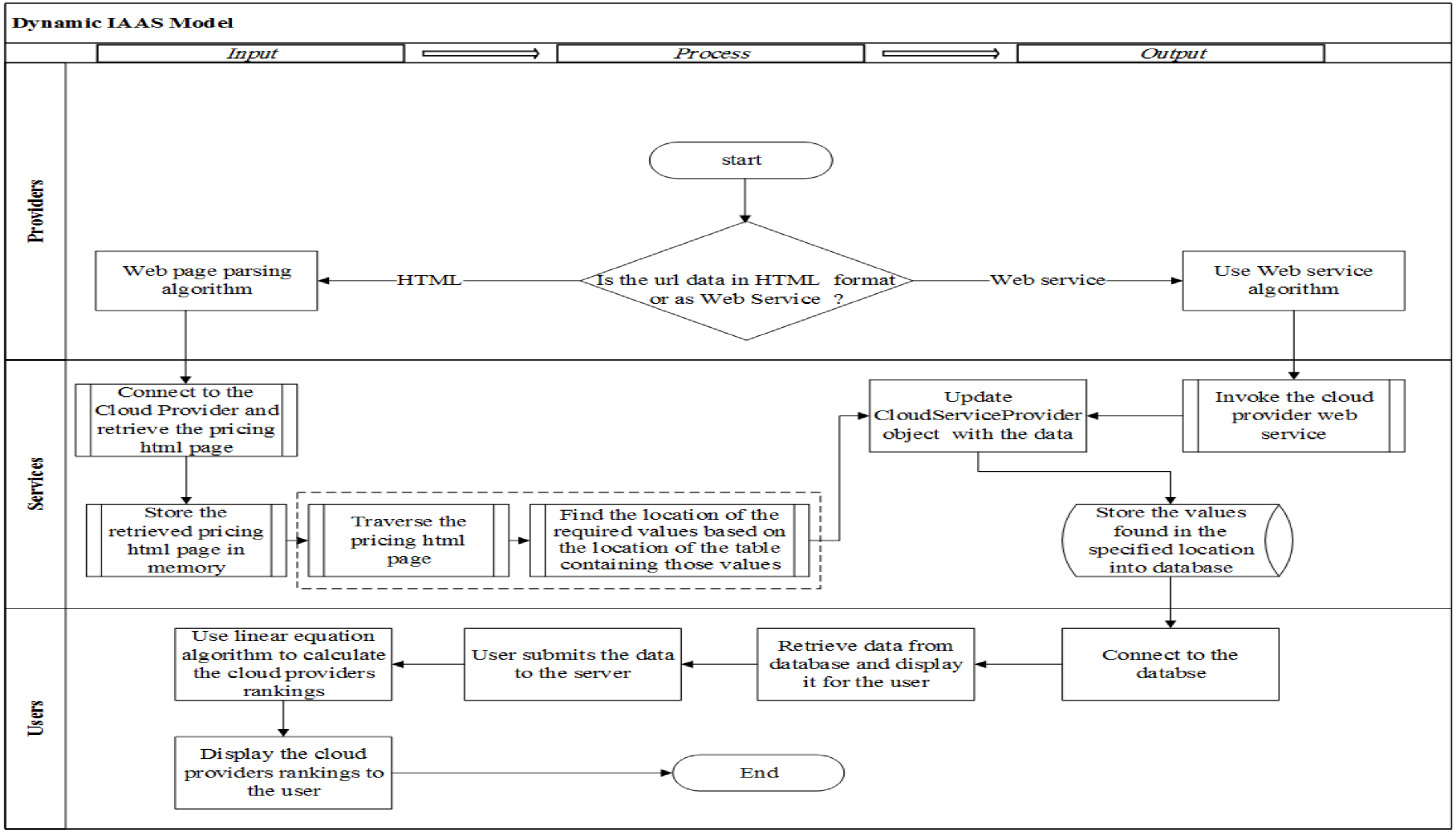


Figure 3.3 *Comprehensive Procedures for Dynamic IAAS Model*

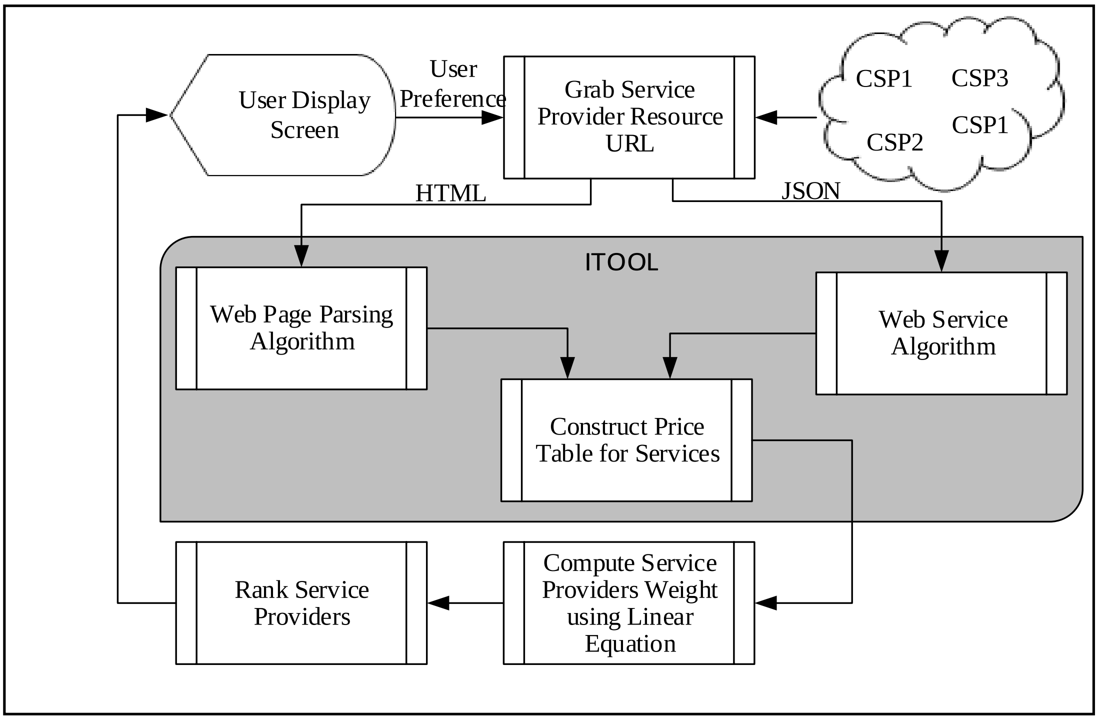


Figure 3.4: *Simplified Dynamic IAAS Model*

- **Dynamic IAAS Model Algorithm**

Dynamic IAAS Model Algorithm includes several procedures:

Phase 1: Grab Service Provider Resource URL

Concept: The basic process of establishing a connection between client and server includes three steps: the client sends a synchronize message; the server sends a message that combines an acknowledgment for the client's synchronize and contains the servers synchronize, and then the client sends an acknowledgment for the server's synchronize. The process is called the TCP three-way handshake. Figure 3.3 and 3.4 shows all these processes, the first one explains in detail and the other is simpler.

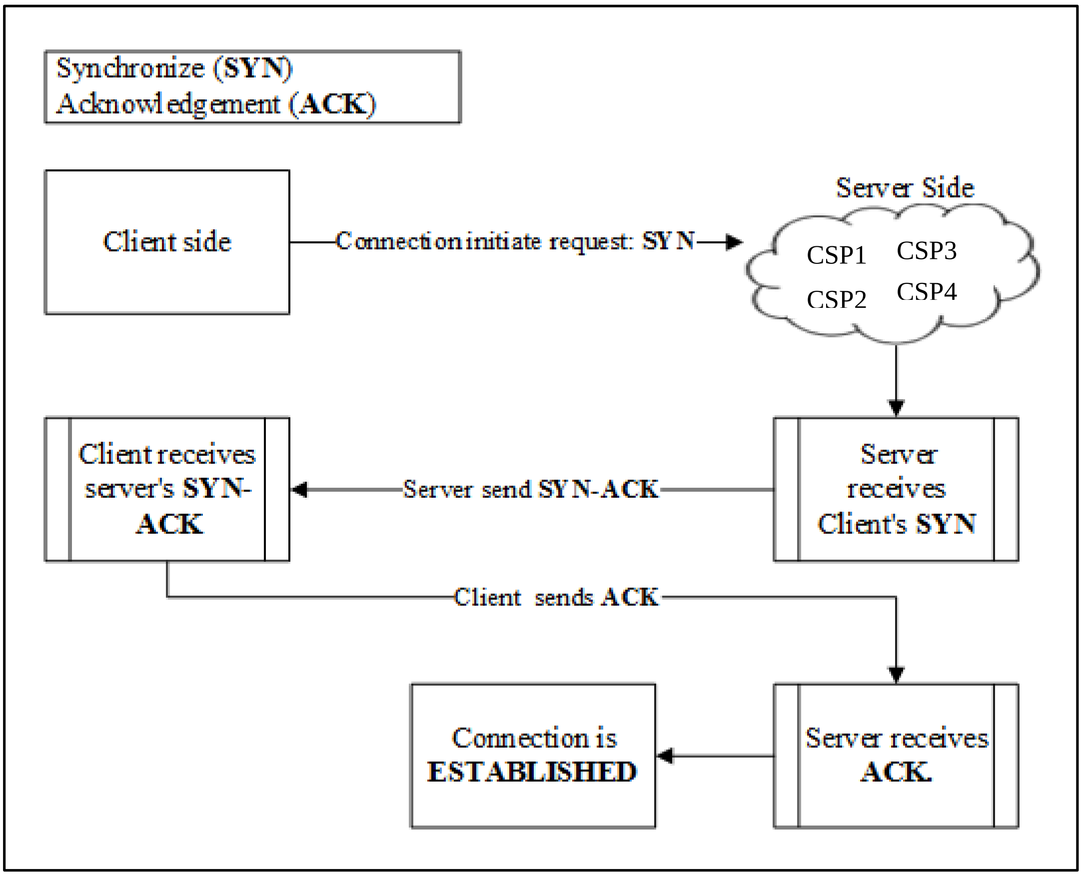


Figure 3.5: *Grab Service Provider Resource URL Process*

Figure 3.5 clarifies how the usual connection is established between a client and server, execution of the three messages sent through the process and how each device transitions from the request to initiate a connection until the session is established.

1. Client sends connection to initiate request (Synchronize) to server.

The server receives clients synchronize.

The server sends a Synchronize –Acknowledgement.

The client receives server's Synchronize –Acknowledgement.

The client sends Acknowledge.

The server receives Acknowledge.

The connection is established.

Store the retrieved data into the database.

Phase 2: The parsing algorithm

1. The parsing algorithms consist of several processes as displayed in Figure 3.6.

Making Parsing the web page (HTML) content into Document Object Model (DOM) nodes inside a tree called "content tree".

Making Parsing the CSS data, inline style elements.

Making a combination of the visual instruction from the web page (HTML) with the styling, and create a "render tree ". The render tree contains rectangles with visual attributes. The rectangles will be placed in the right position which should be appearing on the screen.

Layout process – moving through the render tree, and allowing each node the exact assortment where it should be appearing on the screen.

Painting process – traversing the render tree and painting each node using the UI Backend layer. (Vazana, 2016).

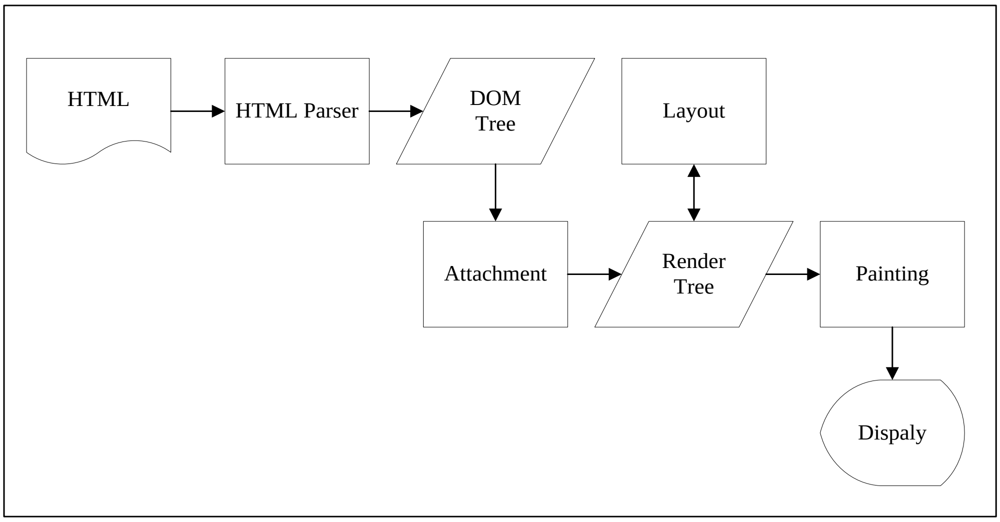


Figure 3.6: *The Parsing Algorithm*

- **How do HTML Parsers Work?**

  As displayed in Figure 3.7 the parsing algorithm of HTML work as the following:

  Step 1: Connect to the Cloud provider and retrieve the pricing HTML page.

  Step 3: Traverse the pricing HTML page.

  Step 4: Find the location of the required values based on the location of the Table containing those values.

  Step 5: Display the values found in the specified location into a database.

  For example: suppose this input HTML markup.

  <html>

  <head>

  <title> </title>

  <meta name=" description" content=" "/>

  </head>

  <body>

  <h2> </h2>

```
<p></p>
<Table>
<tr>
<td>Analytics Values</td>
</tr>
</Table>
</body>
</html>
```

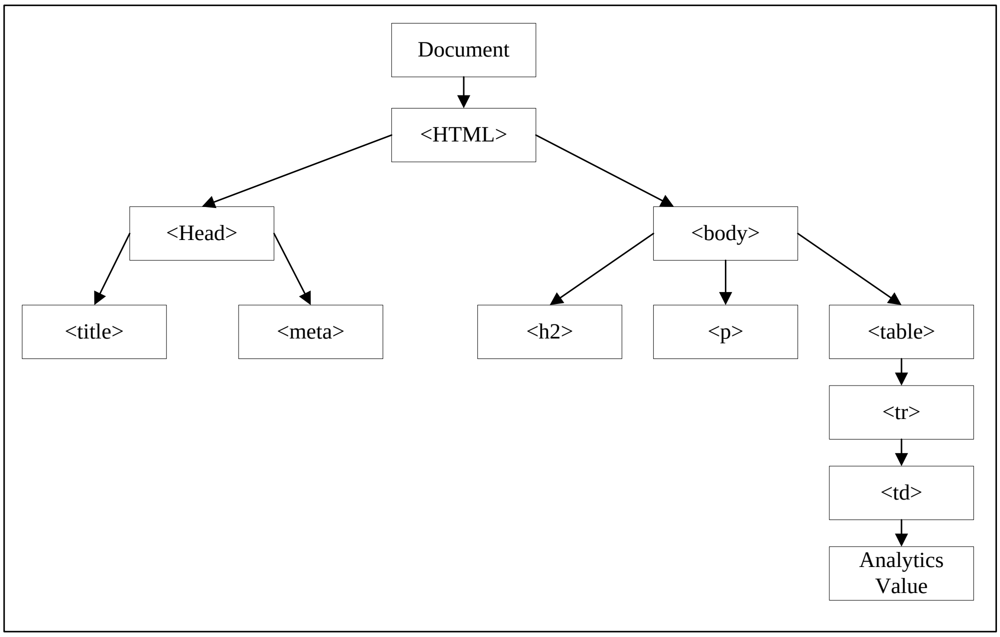


Figure 3.7: *How HTML Parsers Work*

3.3.3 Step 3: Enhance Dynamic IAAS Model

In this step, the focus is on processing server and how it enhances the performance of the dynamic model. This enhanced model is known as measuring preferred service (MPS).

1. To make processing server, we will be resorting to complete the operations on separating server from server applications, in order to improve performance and speed of data processing. Azure Provider Company is providing one technique that is used to process data called worker roles.

Database Caching is a process included in MPS system to generate web pages dynamically by accessing backend databases. The benefit of caching is that it can improve scalability via distributing inquiry workload from backend to front-end systems. It permits elasticity in the processing of data. In addition, caching can enhance the availability of data, via supporting continued service for applications that are based only on cached Tables even if the backend server is unavailable. Another advantage is enhanced data access speeds brought on by locality of data.

ASP.NET SignalR, which is a library for ASP.NET developers improves the way of adding real-time web functionality to applications. Real-time web functionality is the capacity to have the server code push content to connected clients instantly as it becomes available, instead of having the server wait for a client to request new data. (Microsoft ASP.NET 2016). All those processes and technical tools used will be shown in the Figure 3.

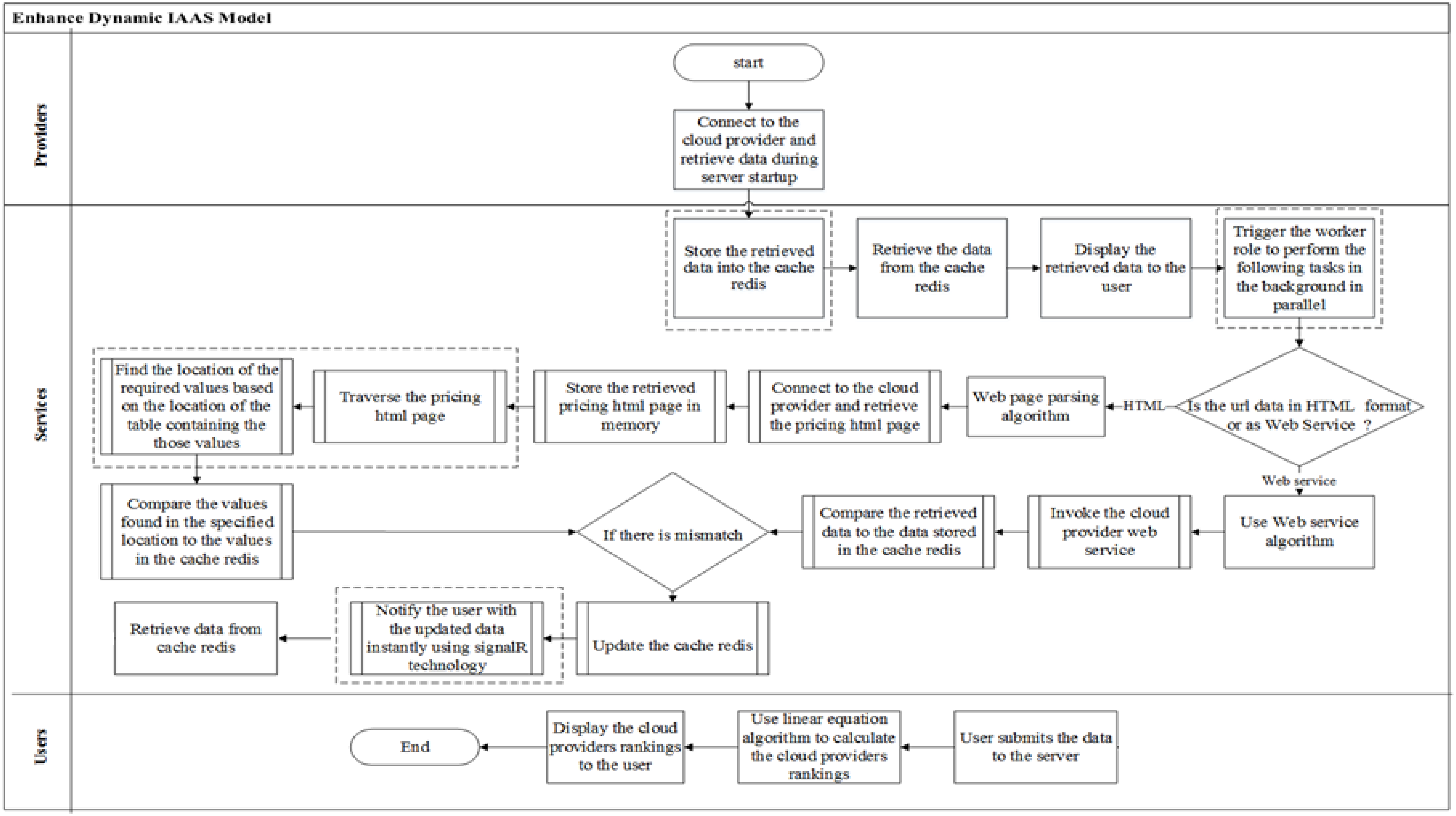


Figure 3.8: Workflow Algorithm of Enhance Dynamic IAAS Model

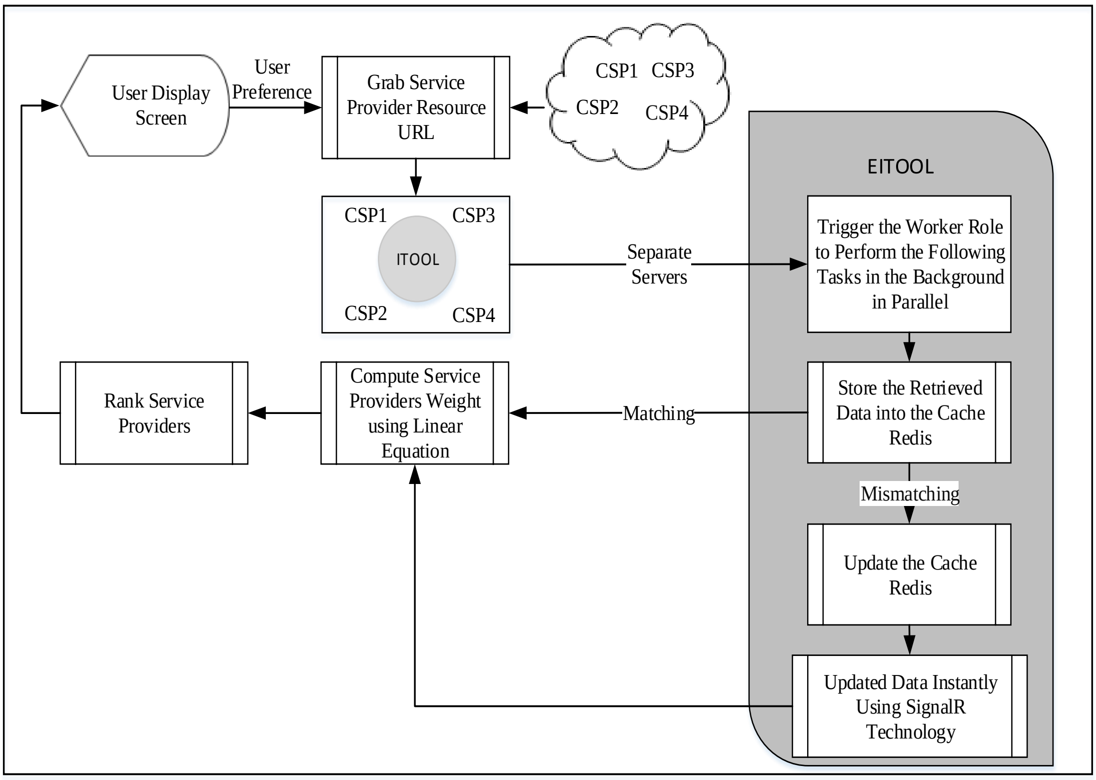


Figure 3.9: *Simplified Enhance Dynamic IAAS Model*

Figure 3.9 explains the DIAAS algorithm, phases and steps which are related to the providers, services, and users. In addition, all the important techniques as worker role, cache redis and SignalR helped to increase system efficiency and reduce the time to display the data and the results instantly.

- **Enhance Dynamic IAAS Model Algorithm**

Phase 1: Connect to server

1. Connect to the Cloud provider and retrieve data during server startup.

Store the retrieved data into the cache redis.

Retrieve the data from the cache redis.

Display the retrieved data to the user.

Trigger the worker role to perform the following tasks in the background in parallel.

Phase 2: If web service

1. Invoke the Cloud provider web service.

Compare the retrieved data to the data stored in the cache redis.

If there is mismatch;

Update the cache redis

Notify the user with the updated data using SignalR technology

Phase 3: If web page parsing

1. Connect to the Cloud provider and retrieve the pricing HTML page.

Store the retrieved pricing HTML page in memory.

Traverse the pricing HTML page.

Find the location of the required values based on the location of the Table containing those values.

Compare the values found in the specified location to the values in the cache redis.

If there is mismatch;

Update the cache redis.

Notify the user with the updated data using SignalR technology.

Phase 4: Selecting

The user selects data preference.

User submits the data to the server.

Use linear equation algorithm to calculate the Cloud provider's rankings.

Display the Cloud provider's rankings to the user.

By using these weights and values, all results are recorded in order to compare and aggregate the preferred services and offered services of provider companies, the values of these services should be accurate to sort Cloud provider companies which meet the requirement of the user. Here, a method to propose linear equation is applied for measuring the preferred service based on collecting dynamic data from websites by web services. The services of Cloud provider companies of Cloud Computing details are shown in chapter 4.

3.4 PHASE THREE: IMPLEMENTATION OF MEASURING PREFERRED SERVICE (MPS) DYNAMIC MODEL

In this phase, services of MPS models are considered, implementing a simulation model by using ASP.NET programming, using some of the supporting technologies such as Azure worker role and database caching, and additionally performing an input analysis of the data and an output analysis of the simulation results. The MPS model will be developed in the future. Here, the preference equation for service will be offered by MPS conducted model. Through mathematical equation, we can get the combination results. From two sides, user and provider of Cloud Computing and priority provider with its function services.

The process created during the investigation phase (define the requirements, data, and services needed in MPS model) and will be analyzed by the MPS model. (Identification of the Possible Solution). The overall system relies on selection Cloud, which consists of two Clouds, Cloud provider services, and Cloud user selection.

- **The main process for analyzing**

  The main processes are collection service, prioritization of CSP and sorted the list of CSP based on customer. The collected, stored and converted data through all these processes will be tested in MPS model to achieve the successful goal of sorting and ranking providers according to the preferences of service's user.

1. Collection service

   In this process, the data is created as a result of the first investigation phase, the data are transferred into percentages which are inspired by reputation of provider and preferred services from user's site-based MPS model, the linear equation proposed in this thesis is to measure services of Cloud Computing for the user according to its importance and preferences. In this section, we proposed a mathematical MPS model of Cloud selection. By collecting the data from the user and provider Clouds, which consists of two main Clouds selection.

A. The Cloud user selection is in charge of selecting preferred service from potential Cloud users when the user submits a request for selecting the preferred service.

B. The Cloud provider selection is in charge of available services from the provider side. The values of the preferred attributes from the user Computing the final result for services is according to the importance of weights of service.

1. Prioritization of CSP (Computing and calculating services)

   In this process, we present the mathematical linear equation for measuring the preferred service LEMPS which consist of two processes. The first one is finding out the value of isolated service VIS according to our equation, to calculate the percentage of importance for user's service and the second process is finding out VTS the total (sum) values of services to produce the final result. Trust Value of services is explained as following where w1, w2, w3, and w4 are total weights of the trust services such as w1+w2+w3+w4 = 100. The weights of the trust attributes are pre-determined based on their priority. For example, w1 = 20%, w2 = 30%, w3= 10%, w4 = 40%. In this example, storage data is given the highest priority whereas turn around CPU is given the lowest priority.

The basic metric set of trust values consists of services according to user requirements.

Sorted list of CSP based on user

The task is very difficult from two points, first collecting information from websites and filtering them is considered critical task. In addition, this information is available and must be trusted. Second, the offered services provided by ranking providers are measured and tested by MPS, and the variety of available services provided by Cloud provider (cost, CPU, RAM, and storage) by the mathematical linear equation. In addition, (age of the company, availability, is considered non-functional services and depend on intending of the user and not measured in MPS system. All specific tests can be designed and ran according to Cloud users' needs such as testing the reliability of mathematical calculations. The final results and values are organized and ranked the Cloud provider companies according to preferred users' requirements. As a simple illustrated in Figure 3.10

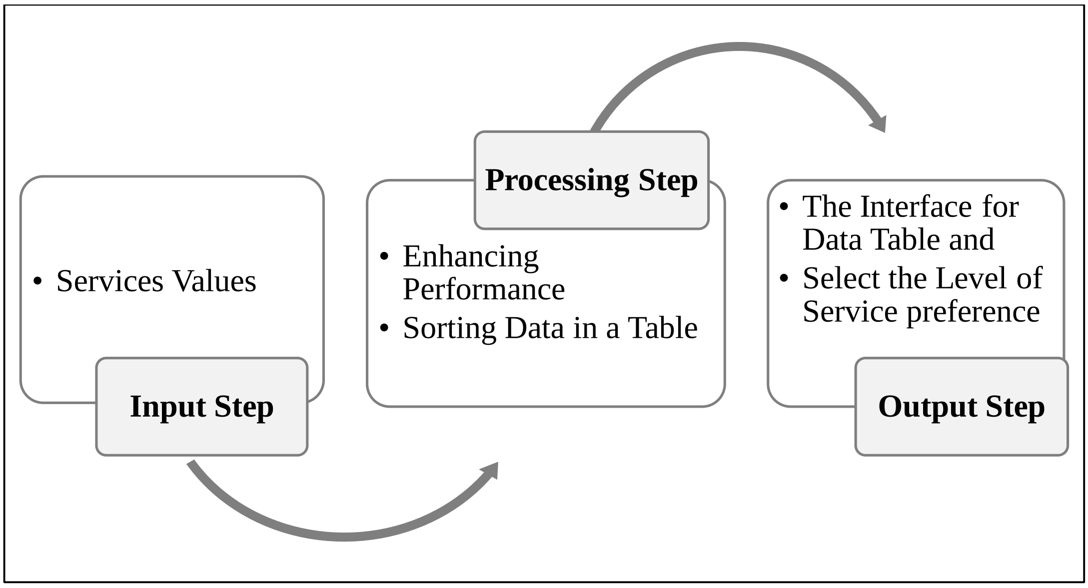


Figure 3.10: *System Diagram for MPS*

The Figure 3.10 for applied MPS system illustrate the mechanism summary details of MPS system work.

A. Input Step

In this phase, the values of the service will be grabbed dynamically via web service to make data up to date, find out the latest service values offerings, which are provided by the service provider companies.

B. Processing Step

1. In this phase, enhancing the performance of MPS system is very important. Therefore, we will use a new technique called worker role, which is provided by Azure Provider Company. This technique is a collection of managed, load-balanced, that work together to perform common tasks. Cloud service roles are managed by Azure fabric controller and provide the ultimate combination of scalability, control, and customization.
2. It is mainly used to perform background processes supports along with Web Roles and performs tasks such as automatically compressing uploaded images, run scripts when something changes in the database, get new messages from queue and process and more. In addition, it is used as database caching mechanism, which is a process included in MPS system to generate web pages dynamically by accessing backend databases. The benefit of caching is that it can improve scalability via distributing inquiry workload from backend to front-end systems and reduce the time to fetch data from the database.
3. Finally, the data, which is processed, is sorted inside the Table and categorized based on the value for each service provider.
4. Output Step
5. In this phase, the attractive interface is implemented to view the data Table values and range selector between high, medium and low level based on the user's needs. Finally, the three models which are static, dynamic and MPS models are implemented and ready to be used by users. The details are discussed in Chapter Five.

3.5 PHASE 4: EVALUATION THREE IAAS MODELS

3.5.1 Step 1: Evaluation of Static IAAS Model

In this step, evaluation of IAAS Model will be measured as input data and its results using linear equation matrix

($y = mx + b$) . The reports and graphs, which describes the percentage of the best service provider based on user requirements, are embedded and significant. In addition, the system also revealed an immediate data that will help to enhance the decision-making.

The users are usually comprised of researchers and Computing personnel that are from universities and customer service companies. These users are involved heavily in developing or using applications, information, runtime, middleware and operating system. For this research, simulated users and their requirement needs are used to test Static IAAS Model as shown in Table 3.3. The score values of 1-3 indicate the essential degree of requirement. A score value of 1 indicates user requires a high level of the service; value 2 implies medium level; whereas the value 3 indicates low level.

Table 3.3

Requirement Level of Service Needs of User

User	SSD	RAM	CPU	Bandwidth	Cost/Month
1	1	2	3	1	2
2	2	1	2	3	1
3	3	2	1	2	3
4	1	3	2	1	2
5	2	1	3	2	1
6	3	2	2	2	3
7	1	3	2	1	3

The simulated users have been accepted the Static IAAS model according to the factors specified and compare it with traditional method to collect data as shown in Table 3.6. These factors are friendly-use, time and decision-making.

Table 3.4

Result of Comparing between Two Methods

(Four SP)	Friendly-Use %	Time /sec	Decision-Making Support %
Collect data/ Online	75	1800	50
	70	2100	40
	80	1920	60
	85	1980	70
	85	2220	50
	80	2160	75
	75	1920	70
Average	**78.571**	**2014.286**	**59.286**
Collect data/ Static	90	30	70
	90	21	70
	100	11	75
	90	14	80
	95	20	70
	100	40	75
	100	13	80
Average	**95**	**21.286**	**74.286**

Table 3.4 illustrates the result of these factors based on the testing and evaluating by a sample of developers who have knowledge of cloud computing, then calculate the average each one of them for validating which method is the best to help the user, in order to select Cloud service provider based on user requirements. The find out of Table 3.6 display comparison of data collection via the web and through static IAAS model developed to validates which method is the best. Thus, notes that the average rating (95, 21.286 and 74.286) of the factors in the second method based on static IAAS model is better than the first.

3.5.2 Step 2: Evaluation of Dynamic IAAS Model

In this step, DIAAS is constructed based on static IAAS model and use new techniques provided by Microsoft Azure Cloud, to enable the user to make an up-to-date decision in choosing the service provider based on his preferences. The construction of DIAAS considers the speed of central processing unit (CPU), the size of random-access memory (RAM), the size solid-state drive (SSD), the bandwidth in bits per second (bit/s), and the cost of service. DIAAS uses an intelligent tool (ITOOL) for grabbing current provider functional services, give weights by employing linear equation and rank the providers by averaging the summation of the weights for all services for providers. From the results, DIASS presented the ranked list according to user preference. The user has the choice to accept or choose another service provider from the ranked list.

Table 3.5 shows the performance measures of DIAAS in terms of capacity, load and time based on experimental user’s requirement. Capacity, load and time are measured using JMeter tool.

Table 3.5
Performance Measures of DIAAS Models

Factors	DIAAS Model
Capacity	68%
Load	500/responds
Time/Average	23.424/s

Table 3.5 shows the need for increased system efficiency and reduce the time to display up to date information and the results instantly. This will ease the effects of user’s search and ranking task in selecting Cloud service provider.

3.5.3 Step 3: Evaluation of Dynamic and Enhance Dynamic IAAS Models

In this step, the two dynamic models will be compared in terms of three factor. These factors are capacity, load and time.

Evaluate. The MPS model is applied to select any kind of sensitive data of Cloud storage in practical life, tests this model and confirms the advantage of the proposed mechanisms, the experiments have been carried out to test and evaluate the MPS model. All the simulations in this chapter are carried out using our MPS simulator running on the environment, which is mentioned in the following.

The following Table shows the performance measures of DIAAS and EDIAAS in terms of capacity, load and time based on experimental user's requirement. Capacity, load and time are measured JMeter tool.

Table 3.6
Performance Measures between DIAAS and EDIAAS Models

Factors	DIAAS Model	EDIAAS Model
Capacity	68%	93%
Load	500/responds	500/ responds
Time/Average	23.424/s	11.834/s

The experiments conducted show that incorporating techniques such as worker role, cache redis and SignalR has increase system efficiency and reduce the time to display up to date information and the results instantly. This will ease the effects of user's search and ranking task in selecting Cloud service provider. Table 3.6 summarizes the result of comparing between static and dynamic models.

Table 3.7
Result of comparing between DIAAS and EDIAAS Models

Factors	DIAAS Model	EDIAAS Model
Capacity	Acceptable	Capability
Load	Reasonable	Sustainable
Time	Fast	Instantly

The capacity designed is to handle the number of users and processes. DIAAS has the capacity of 68%, which is acceptable. In contrast, EDIAAS model has the capacity of 93%. The load testing for 500 concurrent users is considered reasonable as the number approaching 450 loading become slower and the errors increased to 32% when the users reach 490. Thus, DIAAS is considered a failure. While EDIAAS is able to load, continuously with only 7% error. Finally, the response time is the average time to fetch the homepage is 23.424/s for DIAAS and 11.834/s for EDIAAS. This implies that the user instantly receives the result although the number of users is amount to 500. Table 3.7 summarizes the result of comparing between enhance and dynamic models.

3.6 PHASE 5: VALIDATION OF MEASURING PREFERRED SERVICE MODEL (MPS)

The validation of model and design process provide a consistent methodology and systematic framework which connects system identification, the construction of robust models, and controller structure with experimental data. The mechanism to validate the MPS application by using a case study on an expert using the web application in Cloud Computing for four providers: GCE, Azure, AWS and Rackspace.

Firstly, an expert uses the web application in Cloud Computing for four providers: GCE, Azure, AWS and Rackspace. The expert will select preferred services such as SSD, RAM, CPU, Bandwidth, and cost. For each service the expert will prioritize the provider based on his needs. After he has tabled all the values of services according to providers, he will rank and decide the most suitable provider accordingly for his need. It is not necessary that one provider will give the best values for all services.

Secondly, the expert uses MPS application to select preferred services and then ranking these in the provider table in the application.

Finally, the results from the two tables mentioned above are compared to validate the MPS. The results will either give similar or better results. Besides that, the MPS application provides dynamic, up to date and an efficient method to retrieve, rank and present the most suitable provider according to the services for the expert to make decision.

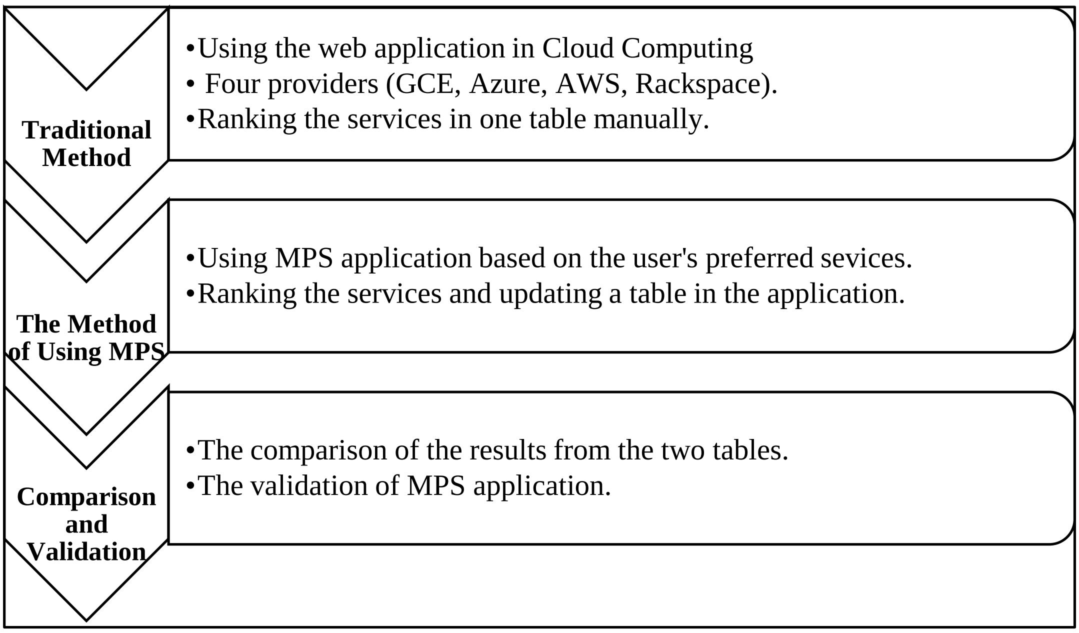


Figure 3.11: *Validation Mechanism*

Figure 3.11 shows the mechanism of validation to prove that the MPS application meets the expert's need to select the most suitable provider in a dynamic, up to date and an efficient method. The results are explained in details in Section 5.6.

3.7 SUMMARY

A proposed model is achieved to lead the user to an optimal selection of the best service provider. With the increase of numerous providers in Cloud Computing, the user faces trouble and problem to define which of the trusted provider company in Cloud Computing can meet their requirements and needs. Since each Cloud provider has their own features related to its reputation, which includes some attributes (age of the company, cost...etc.), a model is proposed which depends on the preferred services of user and reputation of the company of provider. In the dynamic model which called (MPS), after collecting data from both the preferred services of the user and available services of Cloud service, the values of the preferred services that are functional are converted into ratings through data as illustrated in this chapter.

Secondly, the values of the functional services are also converted into the percentages and compared with the values of the same offered attributes in all the provider Cloud selection. Thirdly, as introduced above, for Cloud service section in Phase 2 (analyzing). In Chapter 4, there are many sets of values for the preferred services from a user's feedback, which is considered as functional services and only one set of values is the optimal selection for the best service provider. Computing the values of services and the summation values of volume/service and service/cost, the final result of summation values would be considered a reasonable choice to select the best service provider company, as shown in details in Chapter Five, and the MPS model would be tested for measuring functional services and conclude with future work for developing.

CHAPTER FOUR

RESULT AND DISCUSSION

Chapter Four shows the analysis, findings, and interpretation of information about Cloud Computing companies organized in support of the requirements for a proposed model to select service in Cloud Computing. It represents the second phase, ‘solution identification’ in proposed methodology research of the design thesis phases and process as described in Section 3.1 and the production from this part is used in the development of analyzing steps to substantiate the needs and requirements of MPS model for selecting the best service provider of Cloud Computing from a user’s priority.

The aim of the related studies regarding the current status of research on selecting service provider Cloud Computing was to get the outcomes of reports. In the current research, these results were used to identify shortcomings, and for the results to be obtained in support and fortification is by collecting data from prestigious companies for both function and non- functional services (mentioned in Section 4.2). Data and results already published will be collected and reviewed in 2014. Moreover, all data is trusted and reliable.

4.1 INTRODUCTION

In this chapter, the purpose is to build and display the requirements for the development of a proposed model, based on the previous analysis to select the best service provider in Cloud Computing. In the starting stages of this study, the need analysis was performed. The intended purpose of identifying shortcomings in current research, defining and guiding this thesis is essentially broken down into two main sections that are based on the sub-research objectives.

The results of user requirements are discussed for functional services of SSD, RAM, CPU, bandwidth, and cost.

Section 4.2. Displayed the analysis of the quality of MPS model for the development of a proposed model for selecting a service provider. The equations and algorithms are displayed in detail in Section 4.2. In Section 4.4 a brief summary is provided and concludes the chapter. Figure 4.1 represents a Table of services for prestigious companies in Cloud Computing.

4.2 STATIC IAAS MODEL

In this model, there are two types of services, which are the functional and non-functional service. Functional services are considered using the linear equation which is described. As for the non-functional services, we do not use the equation but it may be of interest to the user. The data values in the static model are static which are grabbed from the companies' websites manually. In addition, this is in the first phase to test the model in terms of the use of the linear equation and the graphic display and the final report to the ratio preference for each service.

4.2.1 Functional Services

The functional services contain SSD, RAM, CPU, Bandwidth, and cost, which are essential to virtual private server (VPS).To calculate the percentage of the user's preference selected by the user, the linear equation will be used, through which we can find out the percentage of each service value (Shaw et al., 2012). The range level start from 1% to 100% and it is divided into three levels (high, medium and low level). The high level is equal to 100% and the medium level equal to 67% and the low is equal to 34%. Table 4.1 summarizes the information obtained from websites of Cloud Computing providers on the current perceptions and adoption of selecting services of Cloud Computing (AWS, Azure, GCE, and Rackspace) which are famous based on the CloudHarmony site that is interested in cloud computing companies.

These services are SSD, RAM, CPU, and bandwidth are measured by defining the maximum and minimum volume according to the preferred service of users and their application. The cost is the most important service that the user wants to know. This is the common concern of user, but at the same time, the increasing or decreasing of the price depends on the performance and reputation of the provider.

Table 4.1
Cloud Computing Providers Services

VPS	AWS	Azure	GCE	Rackspace
SSD	1000 GB	800 GB	1500 GB	1,200 GB
RAM	60 GB	112 GB	104 GB	120 GB
CPU Core	36 cores	16 cores	16 Cores	32 Cores
Bandwidth	2000 Mb/s	2000 Mb/s	10000 Mb/s	5000 Mb/s
Cost/ Month	$1828.48	$1271.33	$923.47	$4529.79

The users are usually comprised of researchers and Computing personnel that are from universities and customer service companies. These users are involved heavily in developing or using applications, information, runtime, middleware and operating system. Thus, simulated users at Al-Madinah International University (MEDIU) and their requirement needs are used to test the decision-making tool which will be applied later as an example of the experimental user. The rank value is calculated as shown in Figure 4.2. This algorithm is repeated for other services.

Table 4.2
Rank Value Calculation for a Service

For each service

1. Get the values of the service from each provider
2. Sort these values
3. Get the max and min service values
4. Get user preference (low, medium or high)
5. Calculate the upper and lower % from user preference
6. Let *m* = (max-min service values)/upper-lower %)

For provider with highest service value

7. Let y_1 be the service value of provider
8. Let x_1 be upper percentage value of user preference
9. Get value of *b* from $b = y_1 - mx_1$

For the rest of the provider

10. Let y be the service value of provider
11. Use *m* and *b* calculated from Step 6 and 9 respectively
12. Determine x_j
13. Plot y_i vs x_j for each service value and provider.

The values for these functional services are calculated by the linear equation (1) and the calculation performed is referred to Table 4.1 above.

$$y = mx + b \qquad (1)$$

A. Functional Services - SSD

The experimental user decides on the lowest level of service for SSD, this implies the percentage of requirement and evaluation for the service is 34%. Therefore, this percentage will be given for the highest value, which is 1500GB provided by GCE. In contrast, the lowest percentage of 1% is given to the lowest value which is 800GB provided by Azure. The preference and evaluation values between 1500GB and 800GB can be calculated by the linear equation (1) as shown in Table 4.3.

Table 4.3
Percentage of User Requirement for SSD

Percentage	SSD/GB	SP
34	1500	GCE
19.9	1200	Rackspace
10.4	1000	AWS
1	800	Azure

Table 4.3 is showing the percentage for SSD after calculating the preferred service of the experimental user which is 34%. GCE is presented as the highest ratio amongst the mentioned providers. At the same time, Azure obtained the lowest ratio according to the used equation for calculating the preferred services.

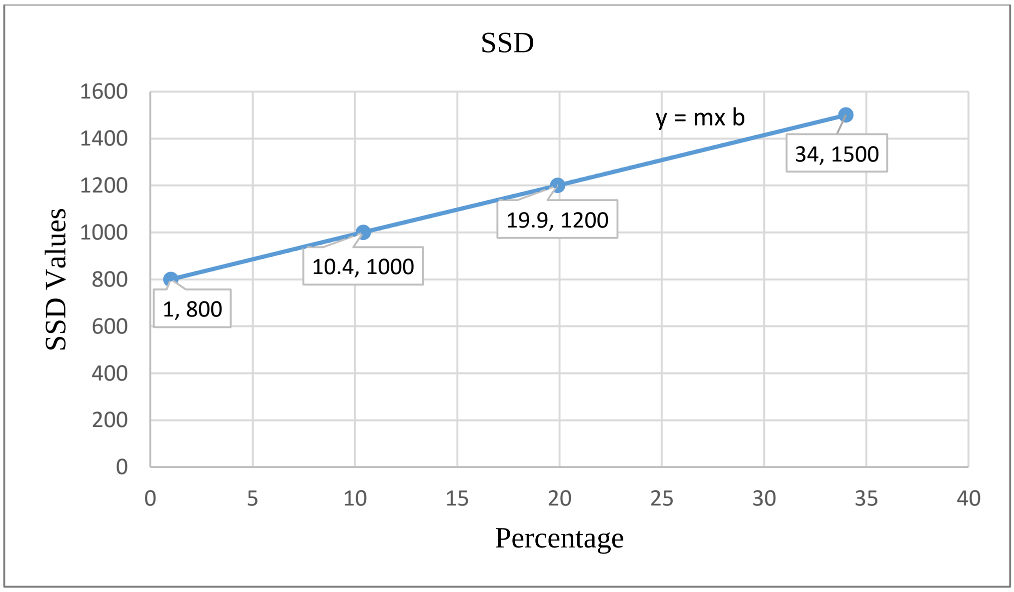


Figure 4.1: *Preference against SSD*

Figure 4.1 is showing the slope line for the SSD that is measured by the linear equation. Throughout the drawing of a set of points on a graph slope, the highest point represented Azure company. In contrast, the lowest point is represented by Rackspace Company.

B. Functional Services - RAM

The experimental user chooses the medium level for the RAM service, which is evaluated as 67%. This percentage is related to the highest RAM value of 120GB provided by Rackspace. In contrast, the lowest value which is 60GB provided by AWS is evaluated as 1% requirement and evaluation. The requirement and evaluation values between 120GB and 60GB are shown in Table 4.4.

Table 4.4

Percentage of User Requirement for RAM

Percentage	RAM/GB	SP
67	120	Rackspace
58.2	112	Azure
49.4	104	GCE
1	60	AWS

Table 4.4 is showing the percentage of the RAM after calculating the preferred service of the experimental user which is 67% for Rackspace Provider which represented the highest ratio amongst the mentioned providers. At the same time, AWS obtained the lowest ratio according to the used equation for calculating the preferred services.

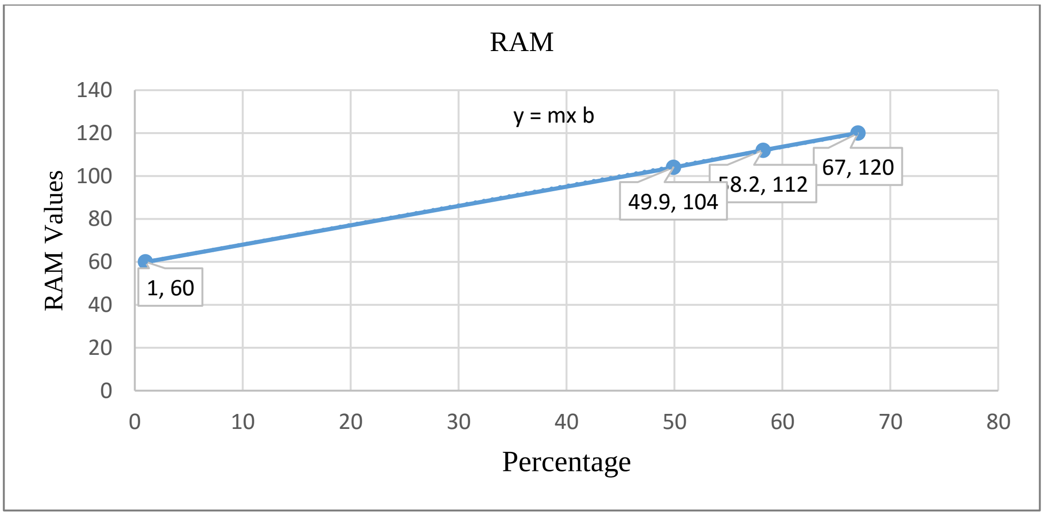


Figure 0.2: *Preference against RAM*

Figure 4.2 is showing the slope line for the RAM that is measured by the linear equation. Throughout the drawing of a set of points on a graph slope, the highest point is represented by Azure company. In contrast, the lowest point is represented by Rackspace Provider.

C. Functional Services - CPU

For CPU function service, the experimental user determined the medium level for the CPU service, which is evaluated as 67%. This percentage is related to the highest CPU value as 36 cores provided by AWS Provider. In contrast, the lowest value which is 16 cores are provided by Azure and GCE evaluated as 1% requirement and evaluation. The requirement and evaluation of the values are 36 cores and 16 cores.

Table 4.5

Percentage of User Requirement for CPU Core

Percentage	CPU Core	SP
67	36	AWS
52.9	32	Rackspace
1	16	GCE
1	16	Azure

Table 4.5 is showing the percentage for CPU after, calculating the preferred service of the experimental user which is 67% for AWS Provider presented the highest ratio amongst the mentioned providers. At the same time, Azure and GCE obtained the lowest ratio according to the used equation for calculating the preferred services.

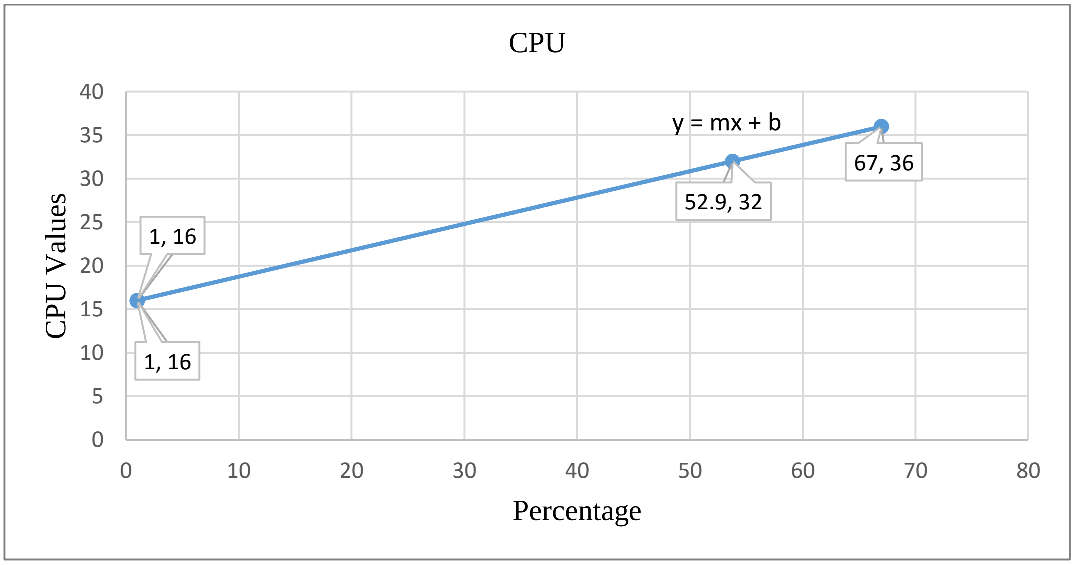


Figure 4.3: *Preference against CPU Core*

Figure 4.3 is showing the slope line for the CPU that is measured by the linear equation. Throughout, the drawing of a set of points on a graph slope, the highest point is represented by AWS company. In contrast, the lowest point is represented by Azure and GCE providers.

D. Functional Services - Bandwidth

In the case of bandwidth function service, the experimental user requires the medium level for the bandwidth service, which is evaluated as 67%. This percentage is related to the highest bandwidth value of 10,000 Mb/s provided by GCE. In contrast, the lowest value which is 2000Mb/s is provided by Azure and AWS evaluated at 1% requirement and evaluation. The requirement and evaluation of the values are 10,000Mb/s and 2000Mb/s.

Table 4.6

Percentage of User Requirement for Bandwidth

Percentage	Bandwidth-Mb/s	SP
67	10,000	GCE
25.8	5,000	Rackspace
1	2,000	AWS
1	2,000	Azure

Table 4.6 is showing the percentage of Bandwidth after calculating the preferred service of the experimental user which is 67% for GCE Provider is presented the highest ratio amongst the mentioned providers. At the same time, Azure and AWS have obtained the lowest ratio according to the used equation for calculating the preferred services.

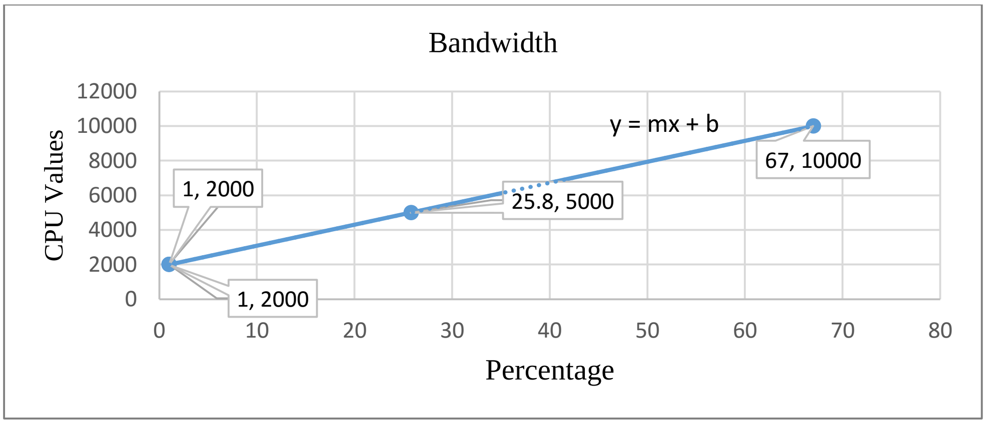


Figure 4.4: *Preference against Bandwidth*

Figure 4.4 is showing the slope line for the Bandwidth that is measured by the linear equation. Throughout, the drawing of a set of points on a graph slope, the highest point is represented by Rackspace company. In contrast, the lowest points are represented by Azure provider and AWS.

E. Functional Services - Cost

The experimental user wants the highest level of service for the cost. The highest percentage of requirement and evaluation for the service was 100%. This percentage has been given for the lowest value which $923.47 is provided by GCE Provider. In contrast, the lowest percentage of 1% is given to the highest value which is $4529.79 provided by Rackspace. The requirement and evaluation of the values are $4529.79 and $923.47. The percentage of cost after calculating the preferred service of the experimental user which is 100% for GCE is presented as the highest percentage amongst the mentioned providers. At the same time, Rackspace obtained the lowest percentage which is 1% according to the used equation for calculating the preferred services (Table 4.7).

Table 4.7

Percentage of User Requirement for Cost

SP	Cost/month	Percentage
GCE	$923.47	100
Azure	$1271.33	90.5
AWS	$1828.48	75.2
Rackspace	$4529.79	1

Table 4.7 is showing the percentage of cost after calculating the preferred service of the experimental user which is 100% for GCE which is presented as the highest ratio amongst the mentioned providers. At the same time, Rackspace obtained the lowest ratio which is 1% according to the used equation for calculating the preferred services.

This Figure 4.5 is showing the slope line for the cost measured by the linear equation. Throughout, the drawing of a set of points on a graph slope, the highest point is represented by GCE company. In contrast, the lowest point is represented by Rackspace Company.

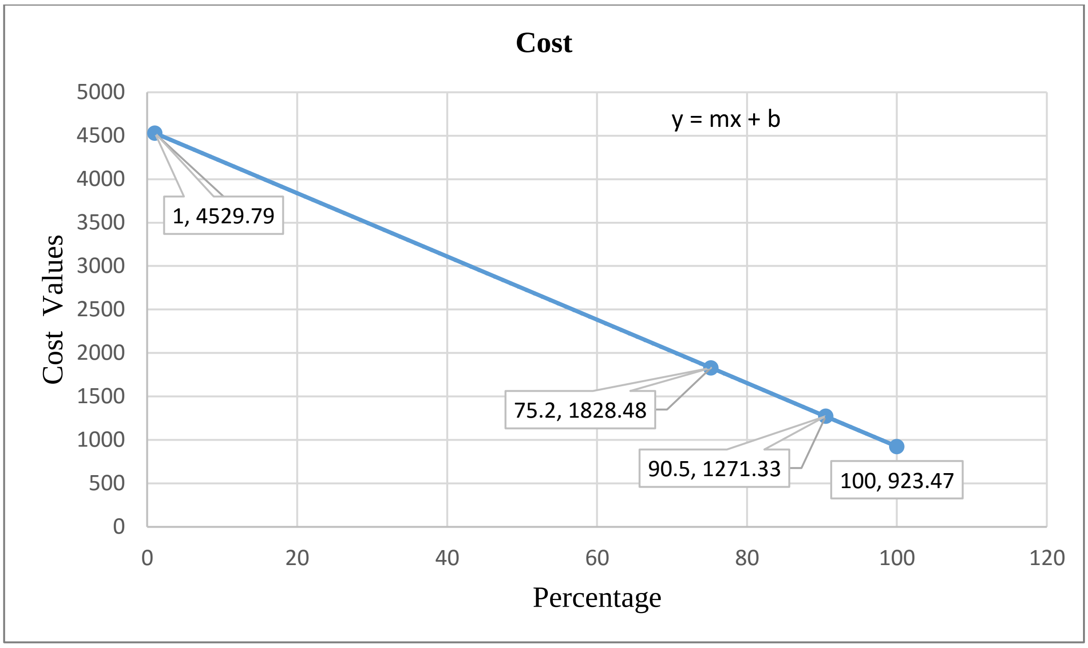


Figure 4.5: *Preference against Cost*

The final result of measuring VPS services and cost is achieved through the mathematical equations which are applied above. The experimental user selected the preferred services depending on their need and priority. The ranking of provider companies in Cloud Computing is produced by the MPS model. Table 4.6 displayed the trust values and results.

$$\sum_{i=1}^{5} x_i = x1 + x2 + x3 + x4 + x5 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2)}$$

Table 4.8

Trust Value Based on User Requirement

	SSD	RAM	CPU	Bandwidth	Cost/ Month	Percentage
AWS	10.4	1.0	67.0	1.0	75.2	30.9%
GCE	34.0	49.4	1.0	67.0	100	50.2%
Rackspace	19.9	67.0	53.8	25.8	1.0	28.3%
Azure	1.0	58.2	1.0	1.0	90.5	30.3%

Table 4.6 displayed the final results of the experimental user who asked about services that can meet with their preference. Moreover, the final result produced GCE Company as the best services provider. The experimental user from this analysis and results has the optimal choice of GCE Company, considering it to be the optimal choice which meets their requirements and application needs. In addition, Figure 4.6 shows the ranking of four providers companies in Cloud Computing according to the priority

of experimental user.

Finally, the results of user experiment that appear in Table 4.8 shows that the highest percentage of preference and evaluation in preference has been given to GCE provider which is 50.2% due to the user's preference for Bandwidth and RAM, which is reflected in the values provided by the company. This indicates that it is not the all-famous companies are the best for their needs.

The experimental user chooses these services according to its priority, final result of sum percentages functional services and the rest of non- function services.

The first impression by analyzing the final results according to the above mathematical formula which is shown in Table 4.8, is that there are simple differences between final values of services in percentages. Overall, the GCE provider stands out from the remaining companies with the highest percent especially in the final results as can be testified in the MPS model, however, the function services are similar.

Due to trusted values of services being the most informative in Tables, GCE which is at the top of the Table is the best services provider among the rest of providers. Nevertheless, these are definitely very interesting values and preferred services for the user in Cloud Computing. In contrast, Azure is in the last rank which has the lowest ratio of 30.3%.

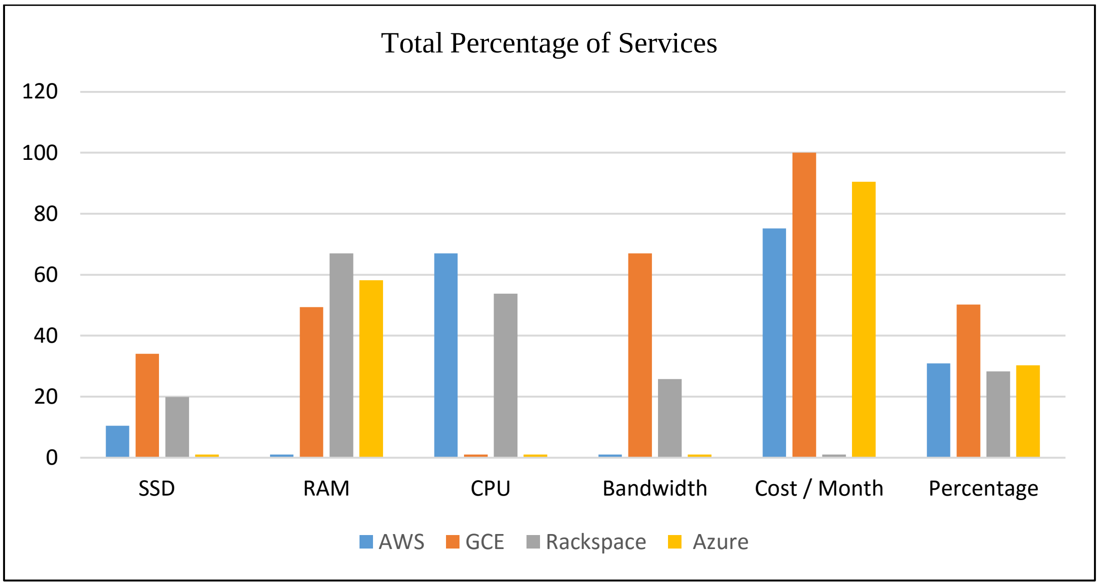


Figure 4.6: *Sorting of the Provider Companies in Cloud Computing*

4.2.2 Non-Functional Services

Non-functional service: reputation, which includes the age of company and availability. These optional services are not included in the calculation. These services are only for information if the user wants it.

A. Availability

The availability is defined as the percentage of time a user can reach the service. The performance of provider in Cloud Computing is achieved in the reputation of the company. In MPS the ranking of companies is taken from the website (networkworld.com, 2015).

Figure 4.7 is showing the downtime or failure of four, where the companies are cutting or stopping services and Computing by hours in 2014. Figure 4.7 shows the total of failures of services (downtime) in 2014.

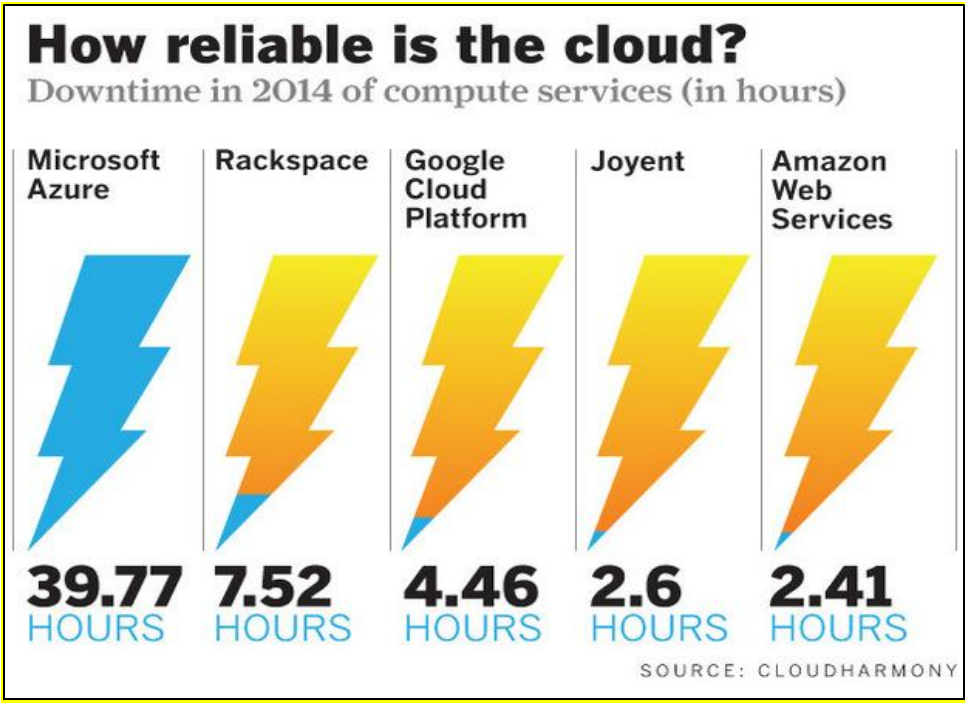


Figure 4.7 *Downtime in 2014 of Computing Services (in Hours)*

Source: Cloud Harmony (2014).

Figure 4.7 reports updated comparisons related to the performance of down time for four prestigious Cloud providers in 2014. Many researchers depended on the Cloud harmony in their work (André, 2011; Figiela, 2013; Arias, 2013).

In the case of the experimental user interest in downtime service, which stands for the failure of receiving services by the user and is given by hours in a year, assuming that the experimental user chooses this service, clicks and submits in the monitor of MPS system. Then the performance of provider in a year can be shown. The experimental user can also show the numbers of hours. It can be represented by Yrs/Hr which shows downtime (the failure time of service) for each provider. All the data is obtained from harmony website that is trusted and updated to help users understand more and determine their best service provider in their interests.

B. Age of Company

This service can be also served according to the intent of the customer in terms of the importance of the provider's age. For example, if a user is interested about the historical age of provider and considers this as an important service of a Cloud provider, then Cloud provider years in Cloud Computing is represented and shown. A number of years will be shown when the experimental user clicks for the age of the company. Table 4.4 shows the ranking of companies in Cloud Computing according to the age of the company. The user has one choice if they are interested in this service and MPS gives related data with the preferred service. Table 4.9 below is showing the history of four prestigious companies:

Table 4.9
Age of Company Providers

SPs	Age of Company
AWS	2006
GCE	2012
Rackspace	2006
Windows Azure	2010

Table 4.9 displayed the established years for each company in Cloud Computing. The first and second rows are the best services providers for users if they are interested in the history of the service provider. The experience of provider is an important service which is related to its age.

4.3 DYNAMIC IAAS MODEL

Dynamic IAAS model is a second phase in the development model, where each data values for the service provider is dynamic. Data is retrieved and grabbed using web services. To get data from each provider who are not providing a standard data or any web services for grabbing data will need some intelligent features and abilities in the

MPS system, where an intelligent tool will be developed to do the job automatically. This tool will go to the data page on each site, do the intelligent searching for the highest value plan and record it in a standard way. This allows changing arbitrary data to a standard data that can be processed. The tool is making a lot of loops and conditions to examine the values of each row on each Table inside each provider site.

The intelligent tool algorithm is doing the following:

1. Fetching the pricing web page and detecting all the Tables on that page.

Detecting the needed pricing Tables using regular expressions to find the needed data.
Looping over all of the Tables and data sets until reaching the Table with the highest values in CPU, RAM, SSD, bandwidth, and cost.
The highest value will be stored in a new variable and will be populated using standard way JSON.

To process all the HTML tags, special libraries are used to do that. Importing the data in text format will need to the knowledge of where a Table and a cell is. The rank value is calculated as previously mentioned in Table 4.1. This algorithm is repeated for other services. The same as the previous user's preferences will be used in the static IAAS model. The values of services providers will be grabbed dynamically as shown in Table 4.10.

Table 4.10
Cloud Computing Providers Services

VPS	AWS	Azure	GCE	Rackspace
SSD	640 GB	800 GB	1500 GB	320 GB
RAM	60 GB	112 GB	120 GB	60 GB
CPU Core	32 cores	16 cores	32 Cores	32 Cores
Bandwidth	2000 MB/S	2000 MB/S	10000 MB/S	5000 MB/S
Cost/ Month	$ 1828.48	$ 1271.33	$923.47	$ 921.6

Table 4.10 displayed the values of provider’s services, which are grabbed from services provider’s sites dynamically.

A. Dynamic Functional Services - SSD

The experimental user decides for the lowest level of service for SSD, this implies the percentage of requirement and evaluation for the service is 34%. Therefore, this percentage will be given to the highest value, 1500GB which is provided by GCE. In contrast, the lowest percentage of 1% is given to the lowest value of 320GB which is provided by AWS. The preference and evaluation values between 1500GB and 320GB can be calculated by linear equation (1) and as shown in Table 4.11.

Table 4.11

Percentage of User Requirement for SSD

Percentage	SSD/GB	SP
34	1500	GCE
14.4	800	Azure
9.95	640	AWS
1	320	Rackspace

Table 4.11 is showing the percentage for SSD after calculating the preferred service of the experimental user which is 34%. GCE is presented as having the highest ratio amongst the mentioned providers. At the same time, Rackspace obtained the lowest ratio according to the used equation for calculating the preferred services.

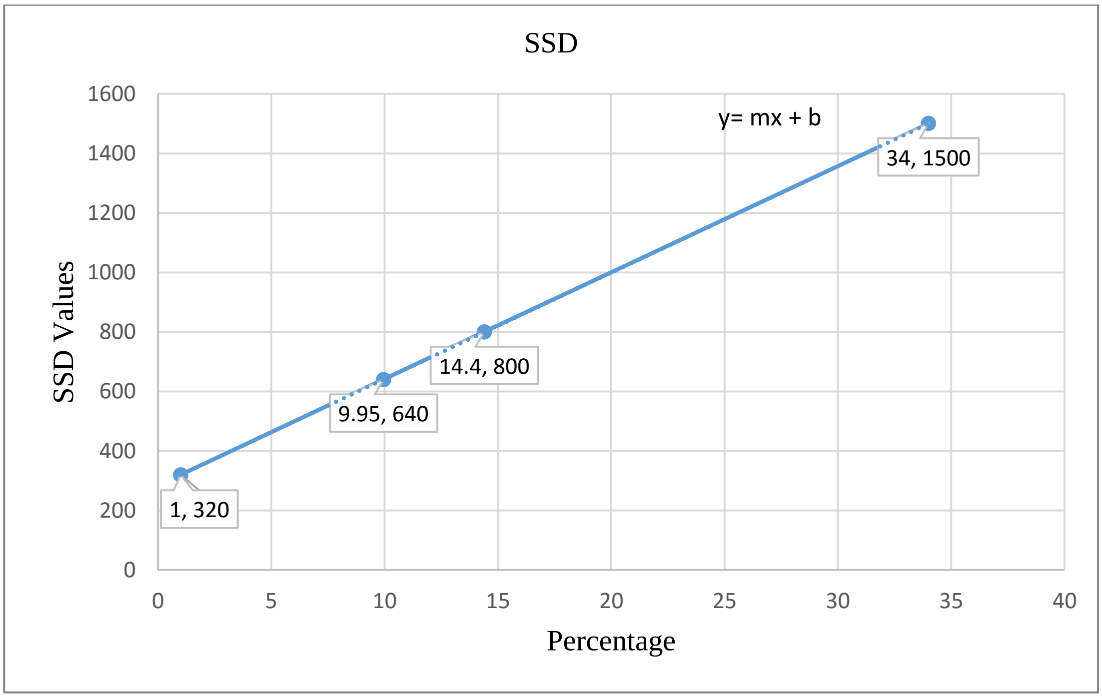


Figure 4.8: *Preference against SSD*

Figure 4.8 is showing the slope line for the SSD that is measured by the linear equation. According to the drawing of a set of points on a graph slope, the highest point is represented by Azure Company. In contrast, the lowest point is represented by Rackspace Company.

B. Dynamic Functional Services - RAM

The experimental user chooses the medium level for the RAM service, which is evaluated as 67%. This percentage is related to the highest RAM value of 120GB provided by GCE. In contrast, the lowest value which is 60GB is provided by AWS and Rackspace, evaluated as 1% requirement and evaluation. The requirement and evaluation values between 120GB and 60GB are shown in Table 4.12.

Table 4.12

Percentage of User Requirement for RAM

Percentage	RAM/GB	SP
67	60	Rackspace
57.4	112	Azure
49.4	120	GCE
1	60	AWS

Table 4.12 is showing the percentage of RAM after calculating the preferred service of the experimental user which is 67%. Rackspace Provider is presented as having the highest ratio amongst the mentioned providers. At the same time, AWS and Rackspace obtained the lowest ratio according to the used equation for calculating the preferred services.

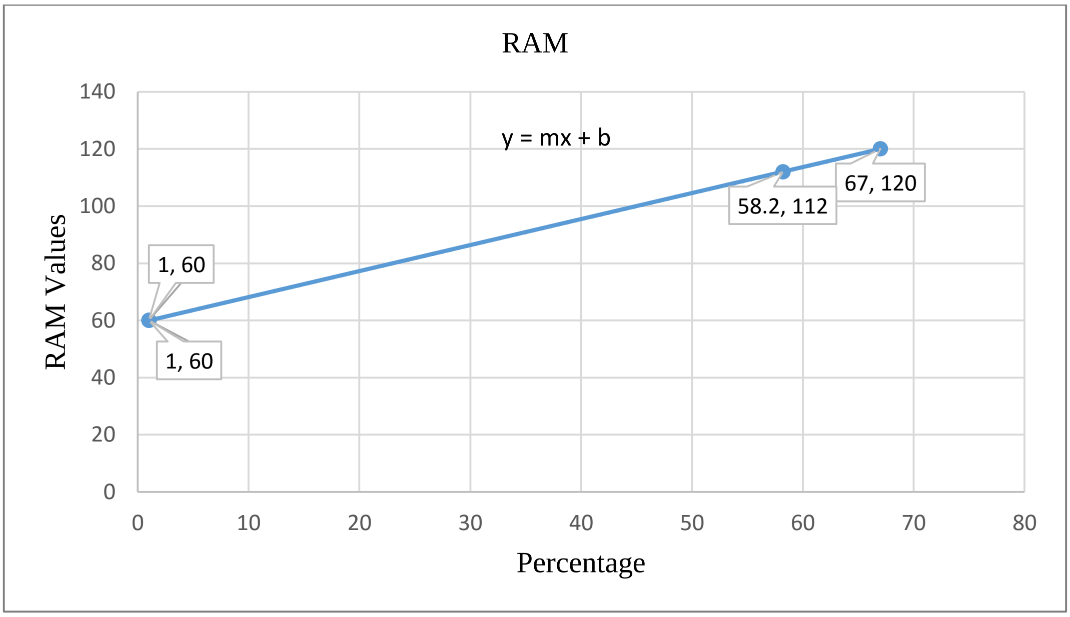


Figure 4.9: *Preference against RAM*

Figure 4.9 is showing the slope line for the RAM which is measured by the linear equation. According to the drawing of a set of points on the graph slope, the highest point is represented by GCE provider. In contrast, the lowest point is represented by Rackspace and AWS Providers.

C. Dynamic Functional Services - CPU

For CPU function service, if the experimental user chooses the medium level for the CPU service, the evaluation is 67%. This percentage is related to the highest CPU value of 32 cores provided by AWS, Rackspace and GCE Providers. In contrast, the lowest value which is 16 cores is provided by Azure and evaluated as 1% requirement and evaluation. The requirement and evaluation of the values are 32 cores and 16 cores. After calculating the preferred service of the experimental user which is 67%, AWS, Rackspace and GCE Providers is presented as having the highest percentage amongst the mentioned providers. At the same time, Azure obtained the lowest percentage according to the used equation for calculating the preferred services (Table 4.13).

Table 4.13

Percentage of User Requirement for CPU Core

Percentage	CPU Core	SP
67	32	AWS
67	32	Rackspace
67	32	GCE
1	20	Azure

Table 4.13 is showing the percentage for CPU after calculating the preferred service of the experimental user which is 67%. AWS, Rackspace and GCE Providers are presented as having the highest ratio amongst the mentioned providers. At the same time, Azure and GCE obtained the lowest ratio according to the used equation for calculating the preferred services.

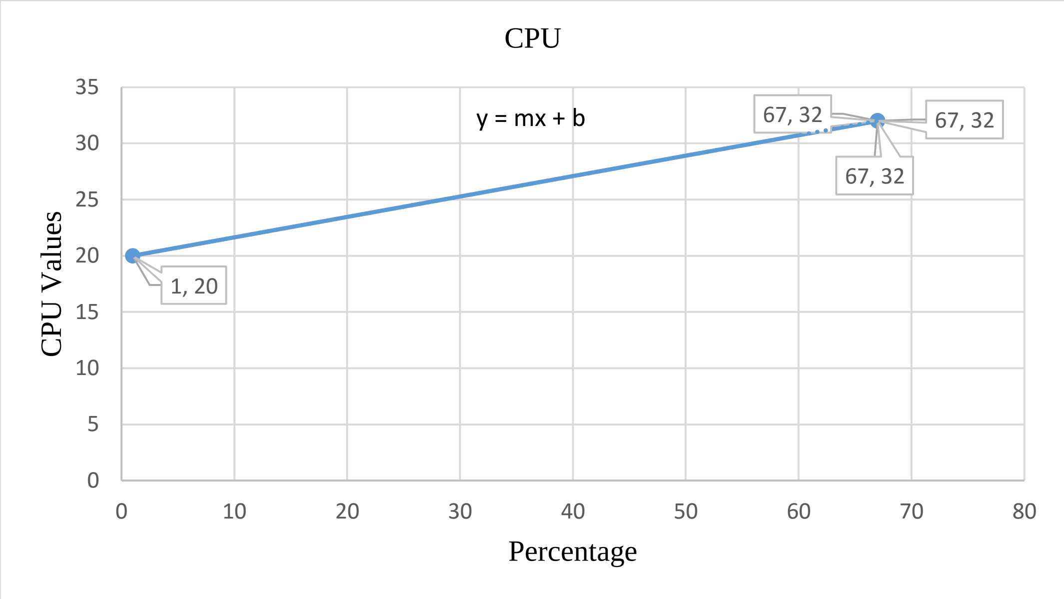


Figure 4.10: *Preference against CPU Core*

Figure 0.10 is showing the slope line for the CPU that is measured by the linear equation. According to the drawing of a set of points on a graph slope, the highest point is represented by AWS, Rackspace and GCE providers. In contrast, the lowest point is represented by Azure provider.

D. Dynamic Functional Services - Bandwidth

In the case of bandwidth functional service, the experimental user requires the medium level for the bandwidth service, which is evaluated as 67%. This percentage is related to the highest bandwidth value of 10,000 Mb/s, provided by GCE. In contrast, the lowest value which is 2000Mb/s is provided by Azure and AWS and evaluated as 1% requirement and evaluation. The requirement and evaluation of the values are 10,000Mb/s and 2000Mb/s. The percentage of Bandwidth after calculating the preferred service of the experimental user which is 67% for GCE Provider is presented as the highest percentage amongst the mentioned providers. At the same time, Azure and AWS obtained the lowest percentage according to the used equation for calculating the preferred services (Table 4.14).

Table 4.14

Percentage of User Requirement for Bandwidth

Percentage	Bandwidth-Mb/s	SP
67	10,000	GCE
25.8	5,000	Rackspace
1	2,000	AWS
1	2,000	Azure

Table 4.14 is showing the percentage of Bandwidth after calculating the preferred service of the experimental user which is 67%. GCE Provider is presented as having the highest ratio amongst the mentioned providers. At the same time, Azure and AWS obtained the lowest ratio according to the used equation for calculating the preferred services.

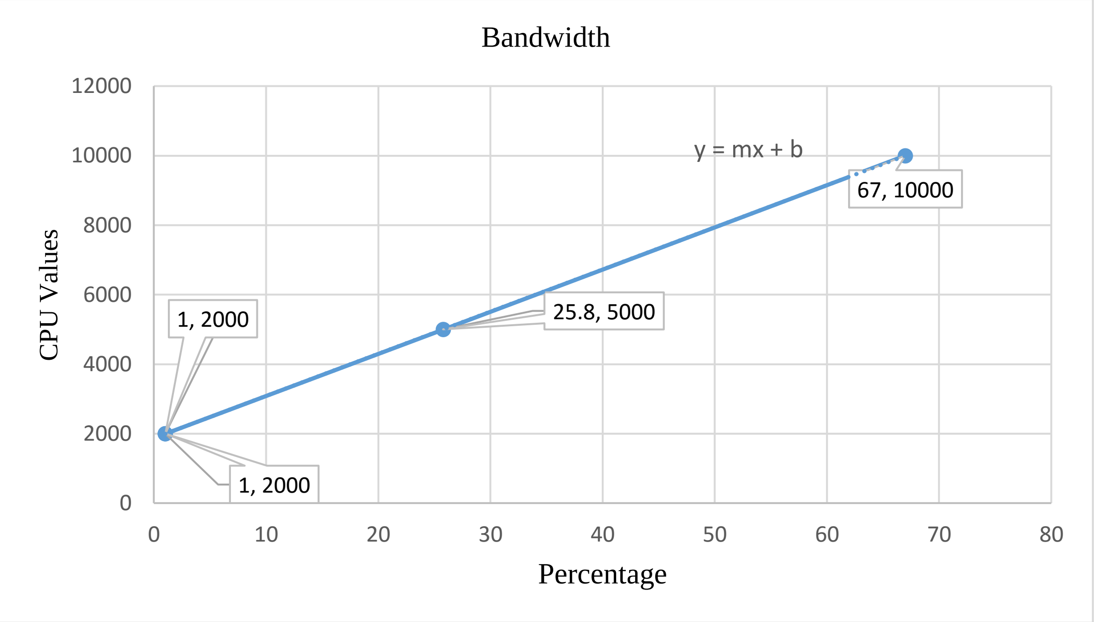


Figure 4.11*: Preference against Bandwidth*

Figure 4.11 is showing the slope line for the Bandwidth that is measured by the linear equation. According to the drawing of a set of points on the graph slope, the highest point is represented by Rackspace Company. In contrast, the lowest points are represented by Azure provider and AWS.

E. Functional Services - Cost

The experimental user wants the highest level of service for the cost. The highest percentage of requirement and evaluation for the service is 100%. This percentage has been given to the lowest value which $923.47 as provided by GCE Provider. In contrast, the lowest percentage of 1% is given to the highest value which is $4529.79 provided by Rackspace. The requirement and evaluation of the values are $4529.79 and $923.47. The percentage of cost after calculating the preferred service of the experimental user which is 100% for GCE is presented as the highest percentage amongst the mentioned providers. At the same time, Rackspace obtained the lowest percentage which is 1% according to the used equation for calculating the preferred services (Table 4.15).

Table 4.15

Percentage of User Requirement for Cost

SP	Cost/month	Percentage
GCE	$923.47	99.8
Azure	$1271.33	61.8
AWS	$1828.48	1
Rackspace	$921.6	100

Table 4.15 is showing the percentage of cost after calculating the preferred service of the experimental user which is 100%. Rackspace is presented as having the highest ratio amongst the mentioned providers. At the same time, AWS obtained the lowest ratio which is 1% according to the used equation for calculating the preferred services.

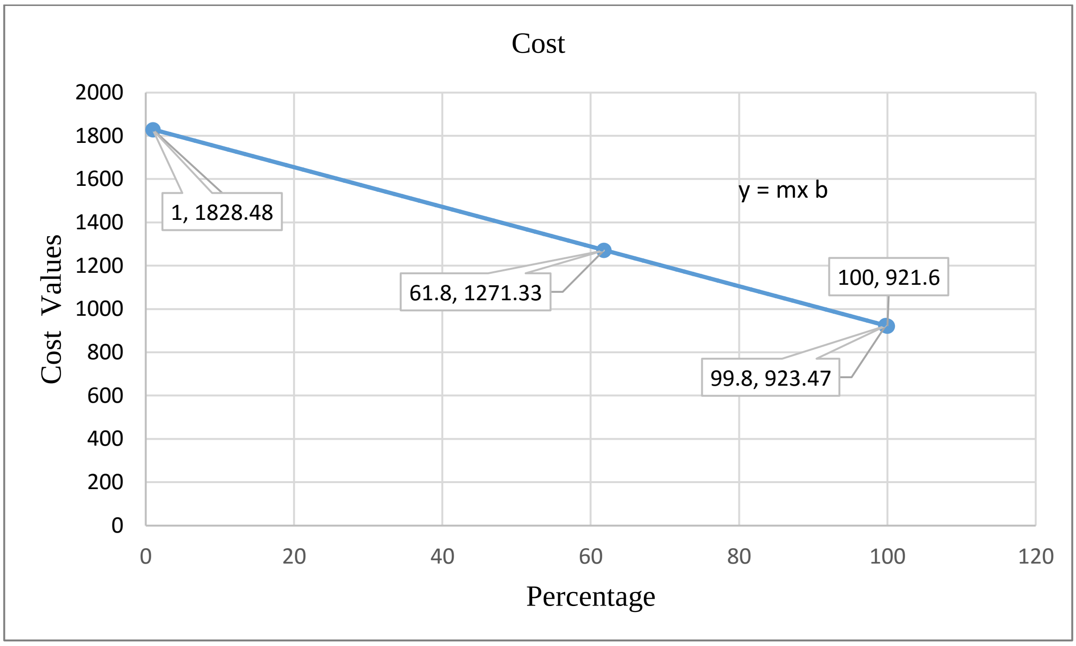


Figure 4.12: *Preference against Cost*

Figure 4.12 is showing the slope line for the cost that is measured by the linear equation. According to t, the drawing of a set of points on the graph slope, the highest point is represented by Rackspace company. In contrast, the lowest point is represented by Rackspace AWS.

The final result of measuring services and cost is achieved through the mathematical equations which are applied above. The experimental user selected the preferred services depending on their need and priority. The ranking of provider companies in Cloud Computing is produced by the MPS model. Table 4.16 displayed the trust values and results.

$$\sum_{i=1}^{5} x_i = x1 + x2 + x3 + x4 + x5 \qquad (2)$$

Table 4.16

Trust Value Based on User Requirement

	SSD	RAM	CPU	Bandwidth	Cost / Month	Percentage
AWS	1.0	1.0	67	1.0	1.0	14.2%
GCE	34	49.4	67	67	99.8	63.44%
Rackspace	19.9	67	67	25.8	100	55.94%
Azure	1.0	58.2	1.0	1.0	61.8	24.4%

Table 4.16 displayed the final results of the experimental user that asked about services that can meet their preferences. Moreover, the final result produced GCE Company as the best services provider. From this analysis and results, the user's optimal choice is to select GCE Company, considering it meets their requirements and application need. In addition, Figure 4.13 shows the ranking of four providers companies in Cloud Computing according to the priority of experimental user.

Finally, the results of user experiment that appear in Table 4.16 shows that the highest percentage of preference and evaluation in preference has been given to GCE provider which is 63.44% due to the user's preference for Bandwidth and CPU, which is reflected in the values provided by the company. This indicates that the all-famous companies are not the best for their needs.

The experimental user chooses these services according to its priority, final result of sum percentages functional services, and the rest of non-function services.

The first impression from analyzing the final results according to the above mathematical formula (2) which is shown in Table 4.16, there are simple differences between final values of services in percentage. Overall, the GCE provider stands out from the remaining companies with the highest percentage especially in the final results as can be testified in the MPS model. However, function services are similar.

Due to trusted values of services being the most informative in Tables, at the top of the Table, Rackspace is the best services provider among the rest of providers. Nevertheless, these are definitely very interesting values and preferred services for the user in Cloud Computing. In contrast, AWS is in the last rank which has the lowest ratio of 14.2%.

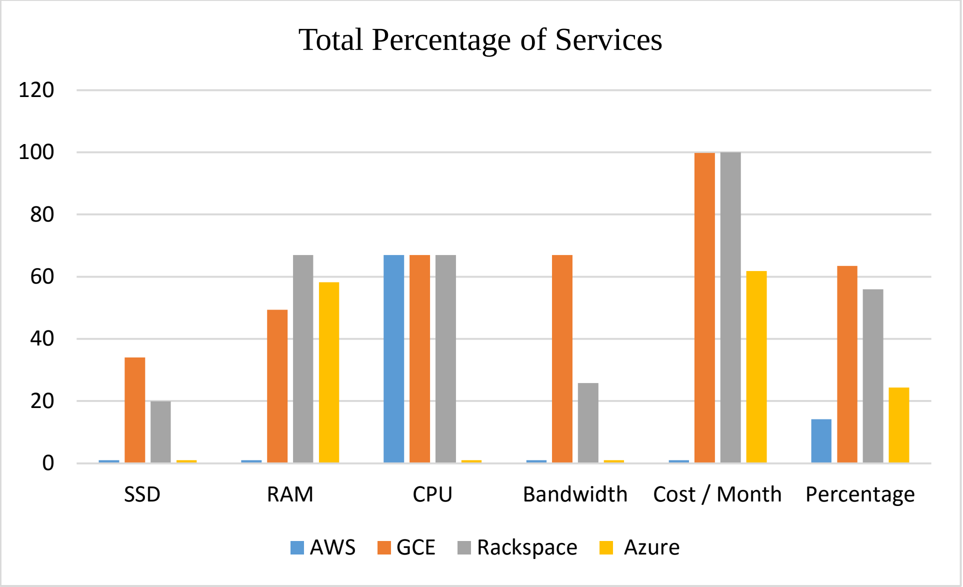


Figure 4.13: *Sorting of the Provider Companies in Cloud Computing*

- **The Properties of Dynamic IAAS Model**

Dynamic IAAS model will be provided with accuracy data, updated data and an appropriate time to grab data. Finally, the result is perfected; all providers' data are real-time and updatable automatically.

4.4 ENHANCE DYNAMIC IAAS MODEL

Enhance Dynamic IAAS model is the third phase of the development of the model. This model will get great features due to the use of Azure Redis Cache technology that is based on the popular open-source Redis cache. It gives access to a secure, dedicated Redis cache, managed by Microsoft and reachable from any application within Azure. In addition, "Azure Redis Cache helps the application become more responsive even as user load increases. It influences the low-latency, high-throughput capabilities of the Redis engine. This separate, distributed cache layer allows the data tier to scale independently for more efficient use of computing resources in your application layer”. In addition, when the user visits the site, data will be displayed immediately, meanwhile, the server checks whether the data has been modified or not. If the data have changed, the user will be notified that the data has been changed, then the process of updating data will start all over again for all providers at the same time due to the use of ASP.NET SignalR which is a new library for ASP.NET developers that makes developing real-time web functionality easy.

The rank value is calculated as previously mentioned in Table 4.16. This algorithm is repeated for other services. The same preferences as the previous user will be used in the static IAAS model. The values of services providers will be grabbed dynamically as shown in 4.17.

Table 4.17
Cloud Computing Providers Services

VPS	AWS	Azure	GCE	Rackspace
SSD	640 GB	800 GB	1500 GB	320 GB
RAM	60 GB	112 GB	120 GB	60 GB
CPU Core	32 cores	20 cores	32 Cores	32 Cores
Bandwidth	2000 MB/S	2000 MB/S	10000 MB/S	5000 MB/S
Cost/ Month	$ 1209.6	$ 1110.24	$ 1152	$ 921.6

Table 4.17 displayed the values of provider’s services which are grabbed from services provider sites dynamically.

A. Enhance Dynamic Functional Services - SSD

The experimental user decides for the lowest level of service for SSD, this implies the percentage of requirement and evaluation for the service is 34%. Therefore, this percentage will be given to the highest value, which is 1500GB as provided by GCE. In contrast, the lowest percentage of 1% is given to the lowest value which is 320GB provided by AWS. The preference and evaluation values between 1500GB and 320GB can be calculated by (1) and as shown in Table 4.18.

Table 4.18

Percentage of User Requirement for SSD

Percentage	SSD/GB	SP
34	1500	GCE
14.4	800	Azure
9.95	640	AWS
1	320	Rackspace

Table 4.18 is showing the percentage for SSD after calculating the preferred service of the experimental user which is 34%. GCE is presented as having the highest ratio amongst the mentioned providers. At the same time, Rackspace obtained the lowest ratio according to the used equation for calculating the preferred services.

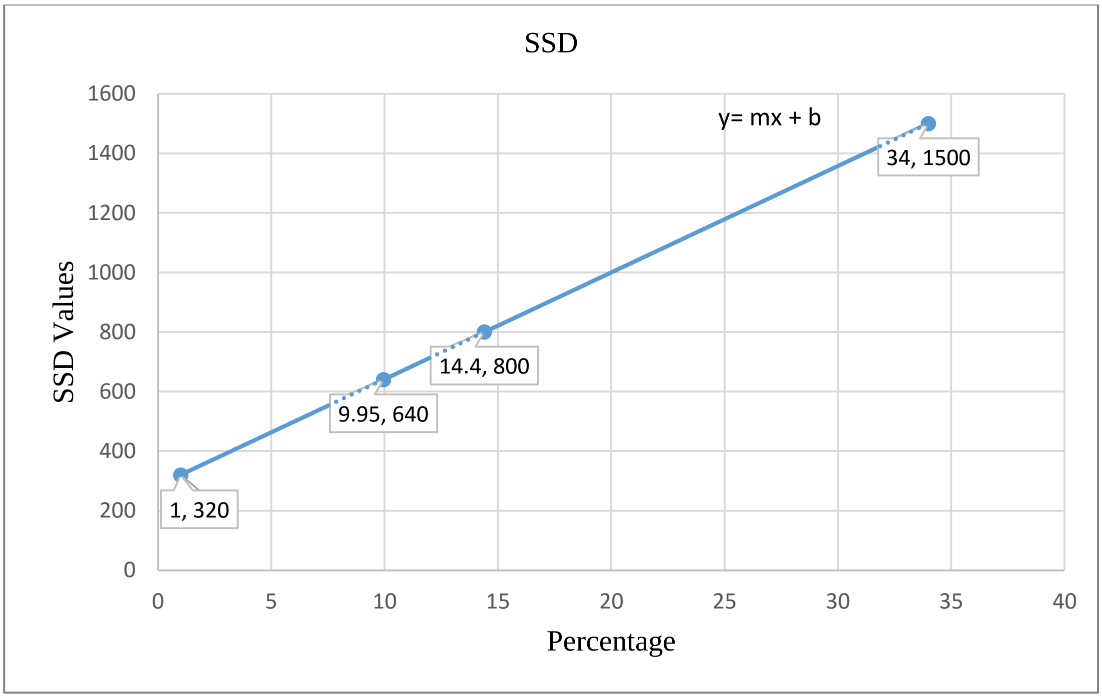


Figure 4.14: *Preference against SSD*

Figure 4.14 **is** showing the slope line for the SSD that is measured by the linear equation. According to the drawing of a set of points on the graph slope, the highest point is represented by Azure company. In contrast, the lowest point is represented by Rackspace Company.

B. Enhance Dynamic Functional Services - RAM

The experimental user chooses the medium level for the RAM service, which is evaluated as 67%. This percentage is related to the highest RAM value of 120GB provided by GCE. In contrast, the lowest value which is 60GB is provided by AWS and Rackspace is evaluated as 1% requirement and evaluation. The requirement and evaluation values between 120GB and 60GB are shown in Table 4.12.

Table 4.19
Percentage of User Requirement for RAM

Percentage	RAM/GB	SP
67	60	Rackspace
57.4	112	Azure
49.4	120	GCE
1	60	AWS

Table 4.19 is showing the percentage of RAM after calculating the preferred service of the experimental user which is 67%. Rackspace Provider is presented as having the highest ratio amongst the mentioned providers. At the same time, AWS and Rackspace obtained the lowest ratio according to the used equation for calculating the preferred services.

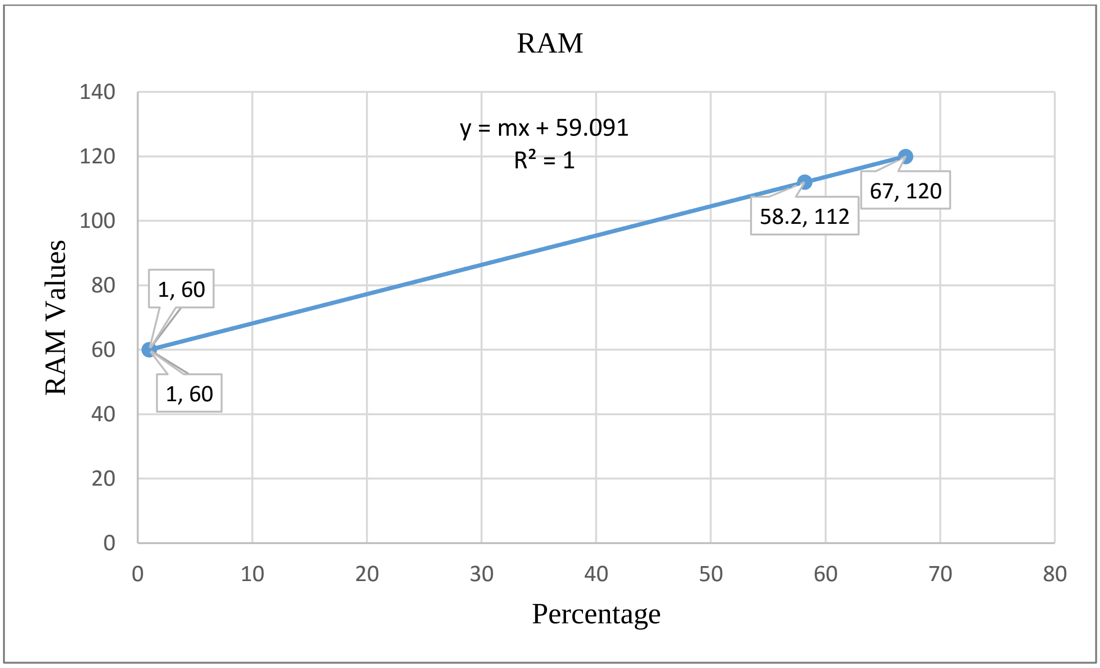


Figure 4.15: *Preference against RAM*

Figure 4.15 is showing the slope line for the RAM that is measured by the linear equation. According to the drawing of a set of points on the graph slope, the highest point is represented by GCE provider. In contrast, the lowest point is represented by Rackspace and AWS Providers.

C. Enhance Dynamic Functional Services - CPU

For CPU function service, the experimental user determined the medium level for the CPU service, which is evaluated as 67%. This percentage is related to the highest

CPU value as 32 cores provided by AWS, Rackspace and GCE Providers. In contrast, the lowest value which is 20 cores is provided by Azure and evaluated as 1% requirement and evaluation. The requirement and evaluation of the values are 32 cores and 20 cores. After calculating the preferred service of the experimental user which is 67% AWS, Rackspace and GCE Providers are presented as having the highest percentage amongst the mentioned providers. At the same time, Azure obtained the lowest percentage according to the used equation for calculating the preferred services (Table 4.20).

Table 4.20

Percentage of User Requirement for CPU Core

Percentage	CPU Core	SP
67	32	AWS
67	32	Rackspace
67	32	GCE
1	20	Azure

Table 4.20 is showing the percentage for CPU after calculating the preferred service of the experimental user which is 67%. AWS, Rackspace and GCE Providers are presented as having the highest ratio amongst the mentioned providers. At the same time, Azure and GCE obtained the lowest ratio according to the used equation for calculating the preferred services.

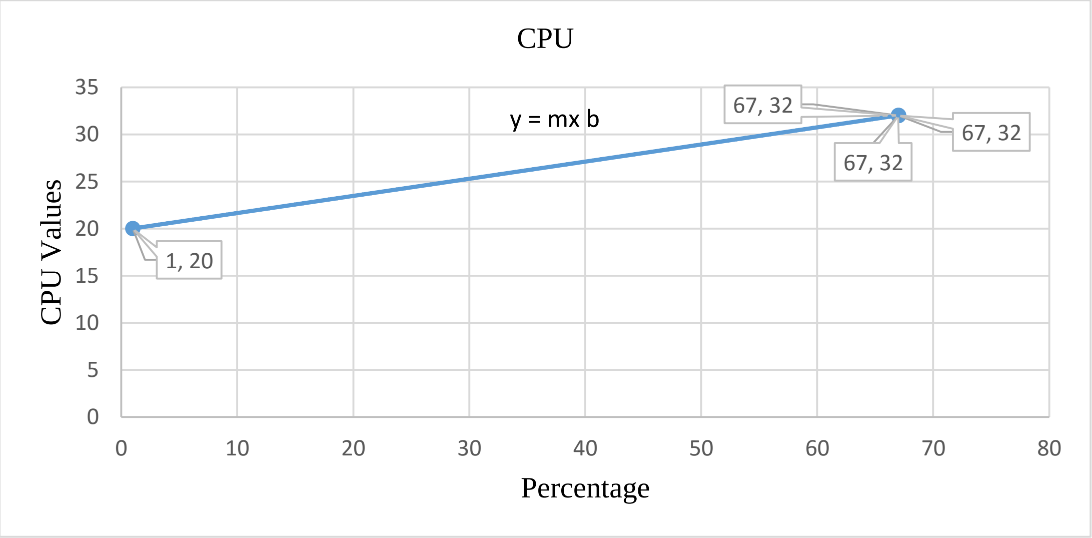


Figure 4.16: *Preference against CPU Core*

Figure 4.16 is showing the slope line for the CPU that is measured by the linear equation. According to the drawing of a set of points on the graph slope, the highest point is represented by AWS, Rackspace and GCE providers. In contrast, the lowest point is represented by Azure provider.

D. Enhance Dynamic Functional Services - Bandwidth

In the case of bandwidth function service, the experimental user requires medium level for the bandwidth service, which is evaluated as 67%. This percentage is related to the highest bandwidth value of 10,000 Mb/s provided by GCE. In contrast, the lowest value which is 2000Mb/s is provided by Azure and AWS and evaluated as 1% requirement and evaluation. The requirement and evaluation of the values are 10,000Mb/s and 2000Mb/s. The percentage of Bandwidth after calculating the preferred service of the experimental user which is 67% for GCE Provider is presented as the highest percentage amongst the mentioned providers. At the same time, Azure and AWS obtained the lowest percentage according to the used equation for calculating the preferred services (Table 4.21).

Table 4.21

Percentage of User Requirement for Bandwidth

Percentage	Bandwidth-Mb/s	SP
67	10,000	GCE
25.8	5,000	Rackspace
1	2,000	AWS
1	2,000	Azure

Table 4.21 is showing the percentage of Bandwidth after calculating the preferred service of the experimental user which is 67%. GCE Provider is presented as having the highest ratio amongst the mentioned providers. At the same time, Azure and AWS obtained the lowest ratio according to the used equation for calculating the preferred services.

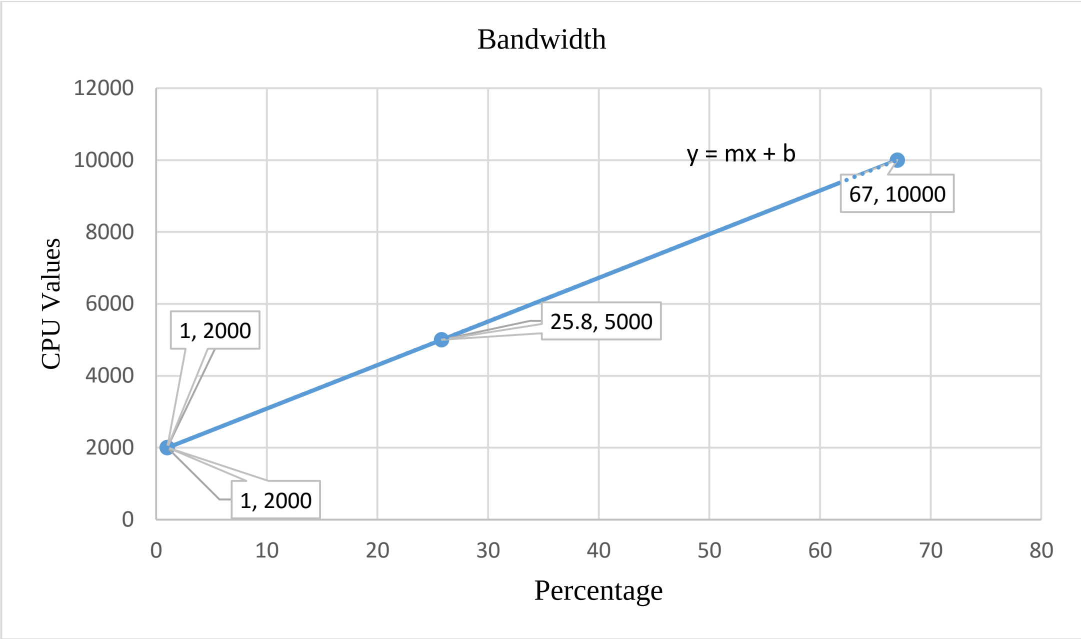


Figure 4.17: *Preference against Bandwidth*

Figure 4.17 is showing the slope line for the Bandwidth that is measured by the linear equation. According to the drawing of a set of points on the graph slope, the highest point is represented by Rackspace Company. In contrast, the lowest points are represented by Azure provider and AWS.

E. Functional Services - Cost

The experimental user wants the highest level of service for the cost. The highest percentage of requirement and evaluation for the service is 100%. This percentage has been given to the lowest value which is $921.6 as provided by Rackspace Provider. In contrast, the lowest percentage of 1% is given to the highest value which is $1209.6 as provided by AWS. The requirement and evaluation of the values are $1209.6 and $921.6. The percentage of cost after calculating the preferred service of the experimental user which is 100% for Rackspace is presented as the highest percentage amongst the mentioned providers. At the same time, AWS obtained the lowest percentage which is 1% according to the used equation for calculating the preferred services (Table 4.22).

Table 4.22

Percentage of User Requirement for Cost

SP	Cost/month	Percentage
GCE	$1152	20.98
Azure	$1110.24	35.2
AWS	$1209.6	1
Rackspace	$921.6	100

Table 4.22 is showing the percentage of cost after calculating the preferred service of the experimental user which is 100%. Rackspace is presented as having the highest ratio amongst the mentioned providers. At the same time, AWS obtained the lowest ratio which is 1% according to the used equation for calculating the preferred services.

Figure 4.18 is showing the slope line for the cost that is measured by the linear equation. According to the drawing of a set of points on the graph slope, the highest point is represented by Rackspace Company. In contrast, the lowest point is represented by AWS Company.

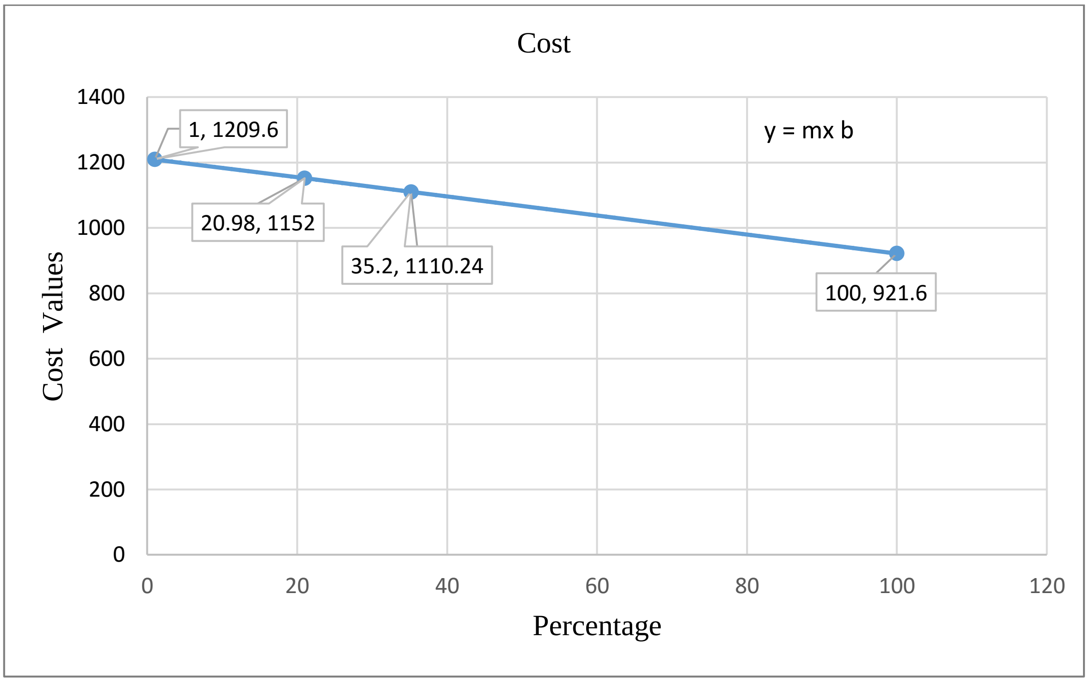


Figure 4.18: *Preference against Cost*

The final result of measuring services and cost is achieved through the mathematical equations. The experimental user selected the preferred services depending on their need and priority. The ranking of provider companies in Cloud

Computing is produced by the MPS model. Table 4.23 displayed the trust values and results.

$$\sum_{i=1}^{5} x_i = x1 + x2 + x3 + x4 + x5 \qquad (2)$$

Table 4.23

Trust Value Based on User Requirement

	SSD	RAM	CPU	Bandwidth	Cost / Month	Percentage
AWS	10.4	1.0	67	1.0	1.0	16.08%
GCE	34	49.4	1.0	67	20.98	34.48%
Rackspace	19.9	67.0	53.8	25.8	100	53.3%
Azure	1.0	58.2	1.0	1.0	35.2	19.28%

Table 4.23 displayed the final results of the experimental user that asked about services that can meet their preferences. Moreover, the final result produced Rackspace Company as the best services provider. From this analysis and results, the user's optimal choice is to select Rackspace Company, considering it meets their requirements and application need. In addition, Figure 4.19 shows the ranking of four providers companies in Cloud Computing according to the priority of experimental user.

Finally, the results of user experiment that appear in Table 4.23 shows that the highest percentage of preference and evaluation in preference has been given to Rackspace provider which is 53.3% due to the user's preference for CPU and RAM, which is reflected in the values provided by the company. This indicates that it is not the all-famous companies which are the best for their needs.

The experimental user chooses these services according to their priority, final result of sum percentages functional services and the rest of non-function services.

The first impression by analyzing the final results according to the above mathematical formula, there are simple differences between final values of services in percentage. Overall, the Rackspace provider stands out from the remaining companies with the highest percent especially in the final results, as can be testified in the MPS model. However, the function services are similar.

Due to trusted values of services being the most informative in Tables, at the top of the Table, Rackspace is the best services provider among the rest of providers. Nevertheless, these are definitely very interesting values and preferred services for the user in Cloud Computing. In contrast, Azure is in the last rank which has the lowest ratio of 16.08%.

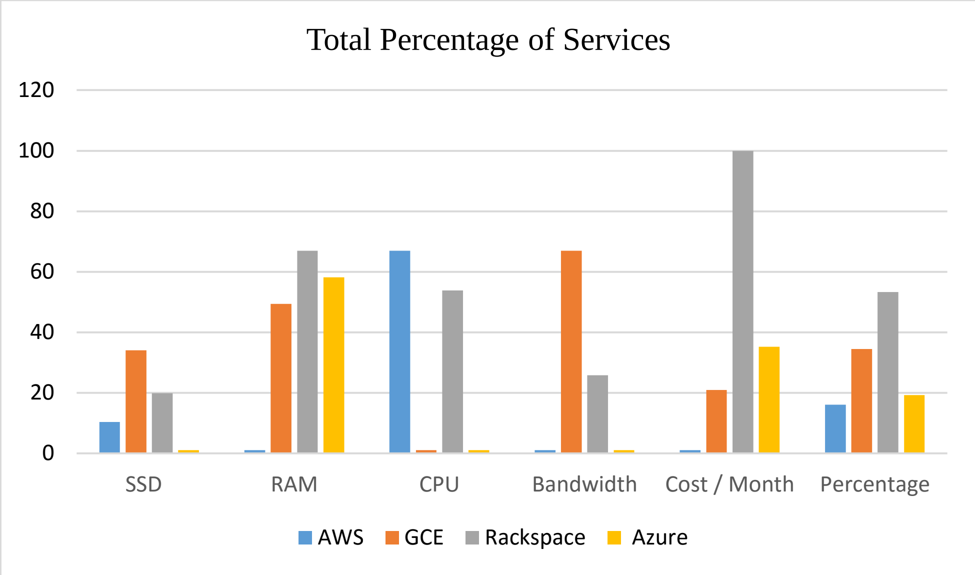


Figure 4.19: *Sorting of the Provider Companies in Cloud Computing*

- The Properties of Enhance Dynamic IAAS Model

The application will become more responsive even as user load increases. The Redis Engine has high-throughput capabilities and independent scaling for more efficient use of computing resources in the application layer. In addition, the real-time data retrieval is instant which means that the servers can now push content to connected clients instantly as it becomes available.

4.5 SUMMARY

This chapter presents the execution of comprehensive proposed model for selecting the best service provider in Cloud Computing resources, with an objective of the optimal choices for the user to select the best service provider. The second objective is to make a rank of these four providers and similarly to acknowledge if the offered services by them are considerably different from each other and the best service provider satisfies the needs of the user according to their priority. The selecting provider problem is modeled and solved using MPS model. Based on the solution, the aggregation data from multiple data is collected from the user side and provider side to compute the preferred services of the user, which could help the user to select the best service provider according to its needs.

In MPS model, the data was aggregated from trusted and reliable resources. Using the MPS model achieves an optimal method for choosing a company, compared to using algorithms and equations to compute the preferred services of the user. It achieves that by summation of the total priority of services data and final results to find out the provider company that satisfies their needs.

The MPS produced optimal results in terms of task ranking provider companies. The MPS model would be highly suitable for reducing the problem in selecting a provider, because of the services in Cloud Computing still increasing and with huge numbers of provider companies, which puts the user in a critical situation on finding the best Cloud service provider. The proposed heuristics based equations in this chapter is implemented in following chapter where performance and optimality are balanced to resolve the thesis's problem through MPS model.

CHAPTER FIVE

MEASURING PREFERRED SERVICE (MPS) SYSTEM

In this chapter, the main observation from the experimental analysis was that the measurement selection strategy embedded in the Cloud selection algorithms was able to bring all the solutions in the optimal selection to the provider in Cloud Computing together. To achieve and enhance the realistic search, the MPS system based on different selection strategies and self-selection mechanism is illustrated in this chapter. It incorporates two selection strategies (namely service user selection and best service provider selection) and tested on measuring preferred service MPS system later, a self-selection mechanism is embedded with the MPS algorithms (based on the optimal decision for selection strategy) in finding better solutions.

The work presented in this chapter tried to find out answers to the research questions i.e., to find an optimal selection strategy that brings the selecting best service provider together, between investigation, exploration and results phases. To enhance the realistic search in the web site-based approaches during the search phases, respectively.

5.1 INTRODUCTION

Technical details of the model were improved by using ASP.NET. It is the new version of a middleware software platform which gives full support to develop models such as the one proposed in the design of the Measuring of Preferred Services MPS system, has been developed to explore optimal selecting for a service provider in Cloud Computing by multi-provider simulations. The comprehensive strategies of MPS model are illustrated in the following Figures below in detail. The MPS model system consists of provided services in MPS model, two agents' provider and user. Both types of agents simultaneously are dealt with by the provided services which are designed in the model. In the dedicated system of MPS model, the agents are in contact. All the system's components are built on the NET platform and SQL Server database. The following sections give the overview of MPS system and describe the selection strategies.

5.2 MEASURING PREFERRED SERVICES (MPS) SYSTEM

To explain and estimate the performance of the proposed model, to implement a web-based system that includes the Measuring Preferred Services (MPS) (MPS) for the user and the functionality to prioritize users by priority requirements included in the optimal selection decision for the best services provider. This section provides the details of this implementation. The interesting characteristic of the implemented available requirements is able to add more services so that each selected services are created by Cloud service. Moreover, the system also can work with application elements.

In the first screen, the system administrator has to submit permission to Login into the MPS system as shown down in Figure 5.1.

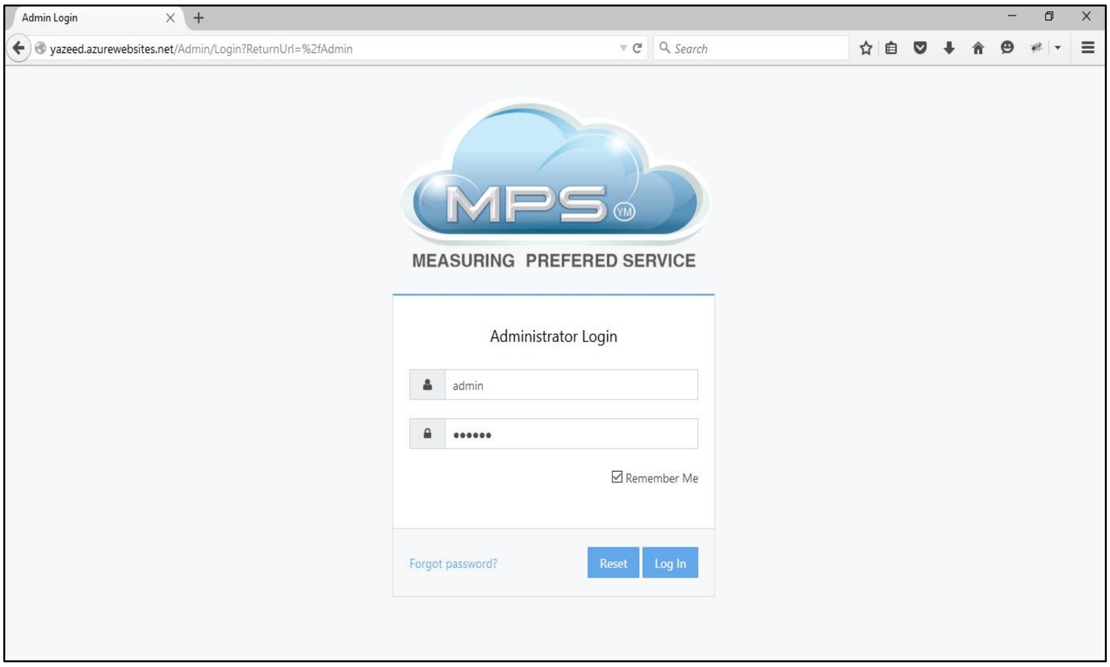


Figure 5.1: *Administrator Login*

To facilitate the multi-process to add, remove, and prioritize services in election list by order, as is shown in Figure 5.1, was used. There are three buttons; new, edit and delete services, the admin of the system is responsible for updating the system as shown in Figure 5.2.

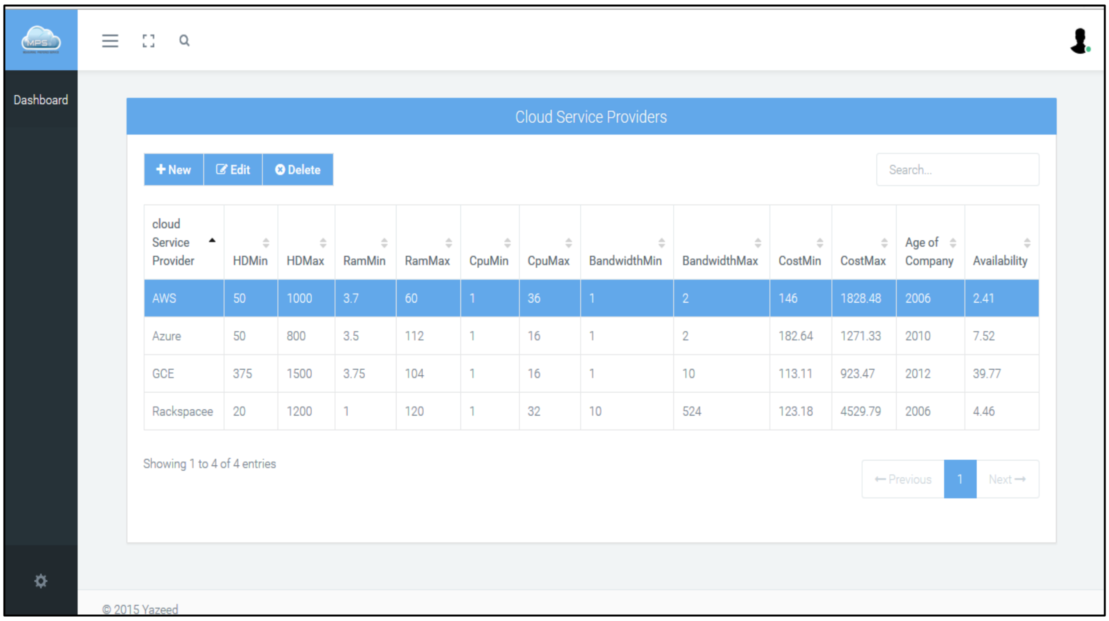


Figure 5.2: *Pick-list for Updating System*

The targeted data of provider and order are given by the user is stored to use and measure. Considering requirements in the prioritization of Cloud user, the ranking of providers was utilized, which is explained in the next subsection.

5.3 SELECTION STRATEGIES

This section proposes two selection strategies in the basic MPS model that are enhanced by algorithms to improve the quality of solutions in obtaining an optimal decision. The mathematical equations used in the thesis are discussed in the previous chapter. Two selection strategies experimented in this chapter are presented in the following sections.

5.3.1 Participant Agents

In MPS model, participant agents to motivate the provider and the user of Cloud services and an unlimited diversity of agents can be developed and used. For the experiment of this thesis, this section introduces seller agents, four prestigious companies, as shown in MPS system used for the experiments of this thesis, allows user agents to share in selecting their services and requirements. Two of them are used in the experiments of this study. MPS system consider on both agent provider and user.

- The Seller Agent

The seller agent refers to the provider's services in Cloud Computing. The provider who is responsible for providing cost and quantity of services which are not stable. The computational resources in huge data centers are where these services are usually hosted on. The seller serves its own services in its websites as an effort to sell the services at a fixed and competitive price, by assuming that multi sellers supply variety services to the user.

- The Buyer Agent

The buyer agent refers to a user’s services in Cloud Computing. According to user priority, the user's demands change dynamically in a realistic situation. MPS model is designed to follow generated and changed services. Seller agent provides services to manage the services and spends the effort to convince the user to buy all the services.

5.3.2 Functional Services User Selecting

Firstly, the experimental user shows provider companies with volume services and individual cost for each providers, as shown in Figure 5.3, (AWS, GCE, Rackspace and Azure companies in Cloud Computing).

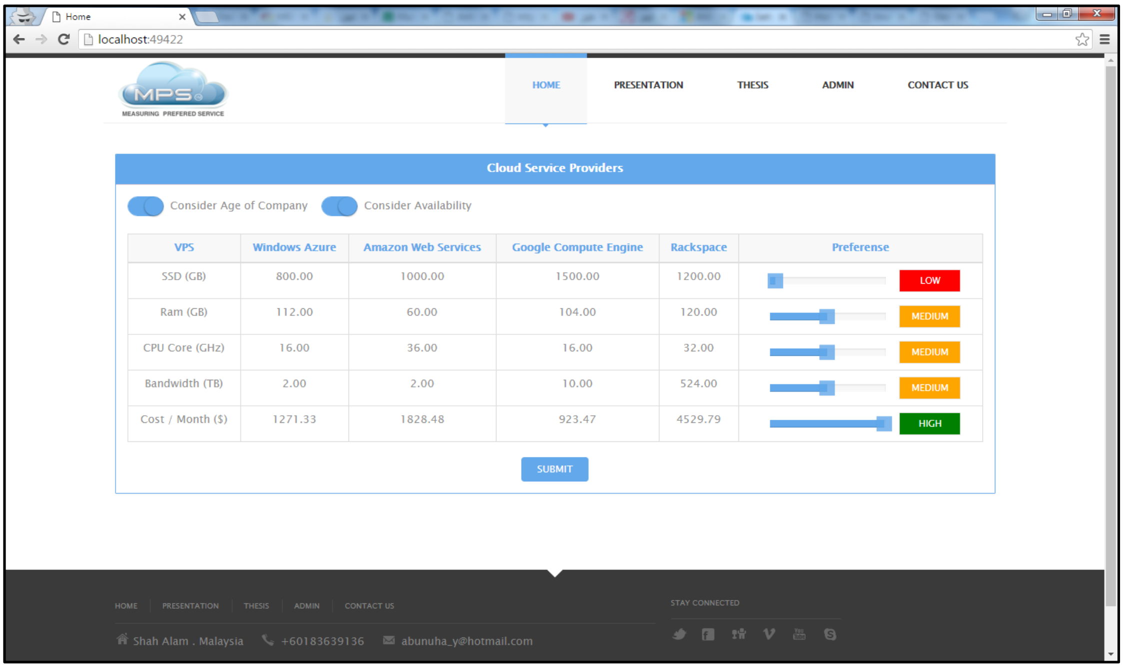
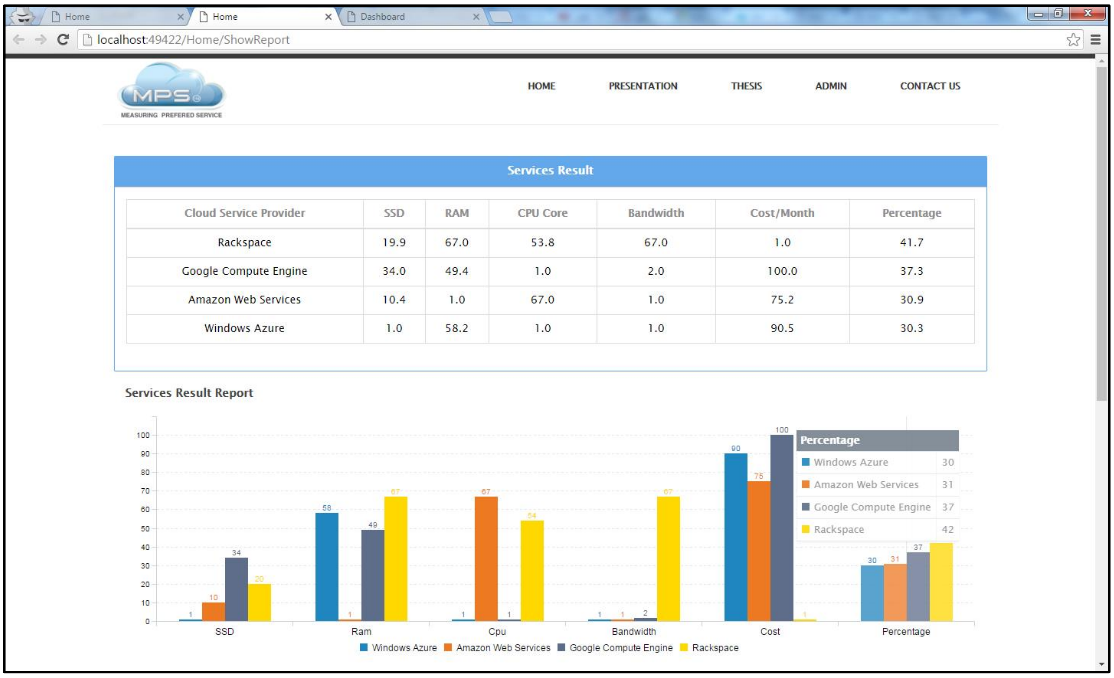


Figure 5.3: *Preferred Services of (VPS) for User in Cloud Computing*

Service selection ranks individuals from the best provider to the worst, the priority of service according to the need of experimental user which is 1% takes the lowest value and 100% takes the highest value services. Additionally, there are also CSP Cloud services providers. P1 represents AWS, P2 represents GCE, P3 represents Azure and P4 represents Rackspace. CSP is Cloud providers who can provide services meeting the user requirements and requirements based on their priority.

When the user submits the list of services to MPS System. Through Figure 5.3, the users choose their preferred services as following: A low level which is 34% for SSD, a Medium level which is 67% for RAM, a Medium level which is for CPU, a Medium level which is for Bandwidth and High level which is 100% for Cost service. The results of adding the total services must be 100%, which is then calculated using mathematical equations for measuring preferred service previous chapter.



Cloud Service Provider	SSD	RAM	CPU Core	Bandwidth	Cost/Month	Percentage
Rackspace	19.9	67.0	53.8	67.0	1.0	41.7
Google Compute Engine	34.0	49.4	1.0	2.0	100.0	37.3
Amazon Web Services	10.4	1.0	67.0	1.0	75.2	30.9
Windows Azure	1.0	58.2	1.0	1.0	90.5	30.3

Figure 5.4: Services Result According to the Priority of Experimental user

The output of the Measuring Preferred Services (MPS) of provider is a set of possible disposition arranged for the best service, which is gained by the provider. Figure 5.4 shows the output screen provided by the system.

5.3.3 Functional Services and Availability Service

In Figure 5.5 as displayed below, by adding availability service that is non-function service, the result will not be affected because this is not a measuring service. To receive this configuration and order this service according to the priority given to function services, just choose and click to display data about availability service that the experimental user is interested in. This is achieved with previous services (see Figure 5.5). It keeps the evaluation of each preferred requirement received by the different providers.

This data could be maintained by using the information from Cloud monitoring services and aggregating the experience of users. The user's applications are deployed in several of four Cloud providers.

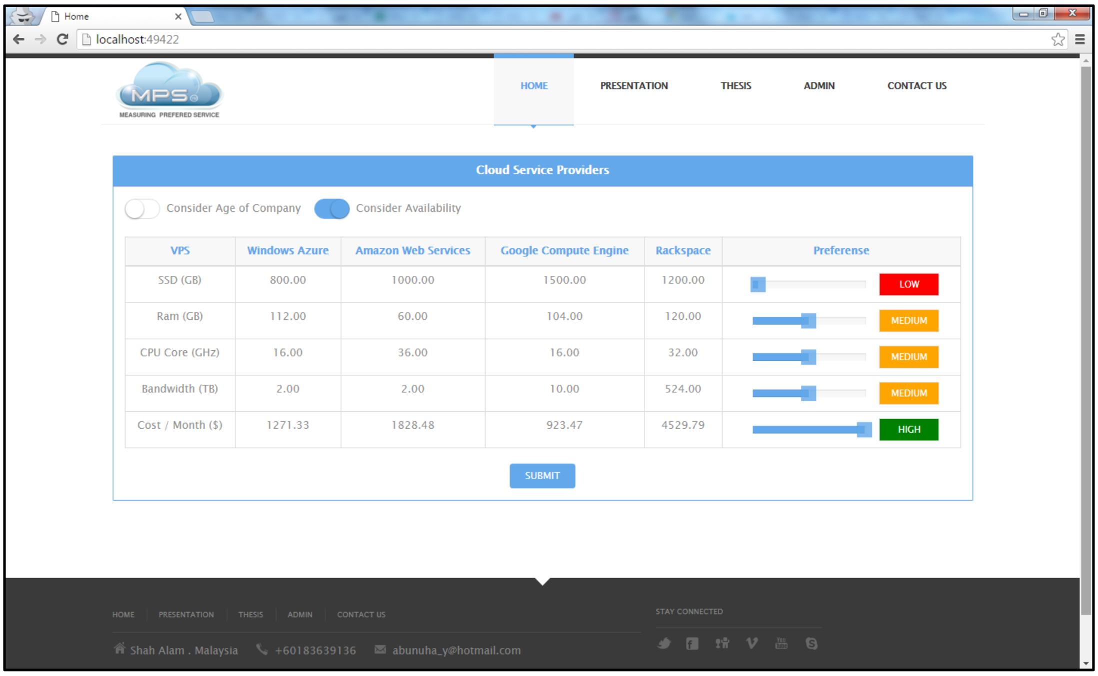


Figure 5.5: *Displayed the Services and Availability Service*

In the case of experimental user deciding to select the availability service, the system can input this automated service and output updated data, as shown in Figure 5.5.

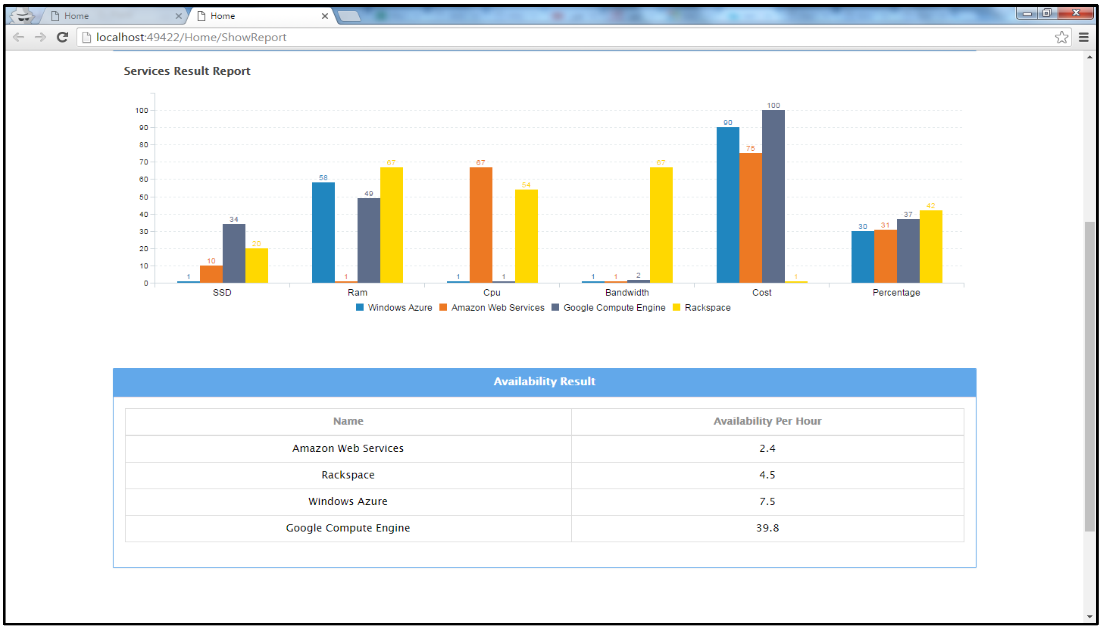


Figure 5.6: *Final Result of Availability and Services*

After arrangement, the list keeps the same layout as the input data, which may be utilized as the input of the following next service of the model, which is the automated service distributor. The screen of this operation is displayed in Figure 5.6.

5.3.4 Functional Services and Age of Company Service

The age of company service input which is used as input for non- function that is not measuring service in the proposed system service as shown in Figure 5.7.

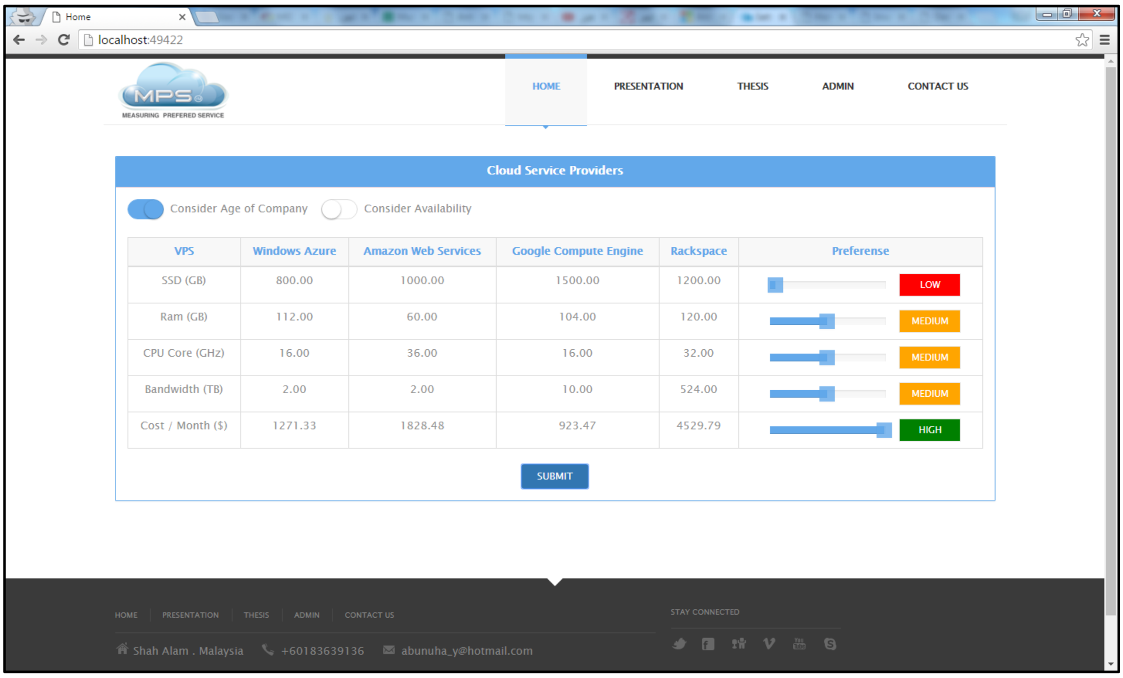
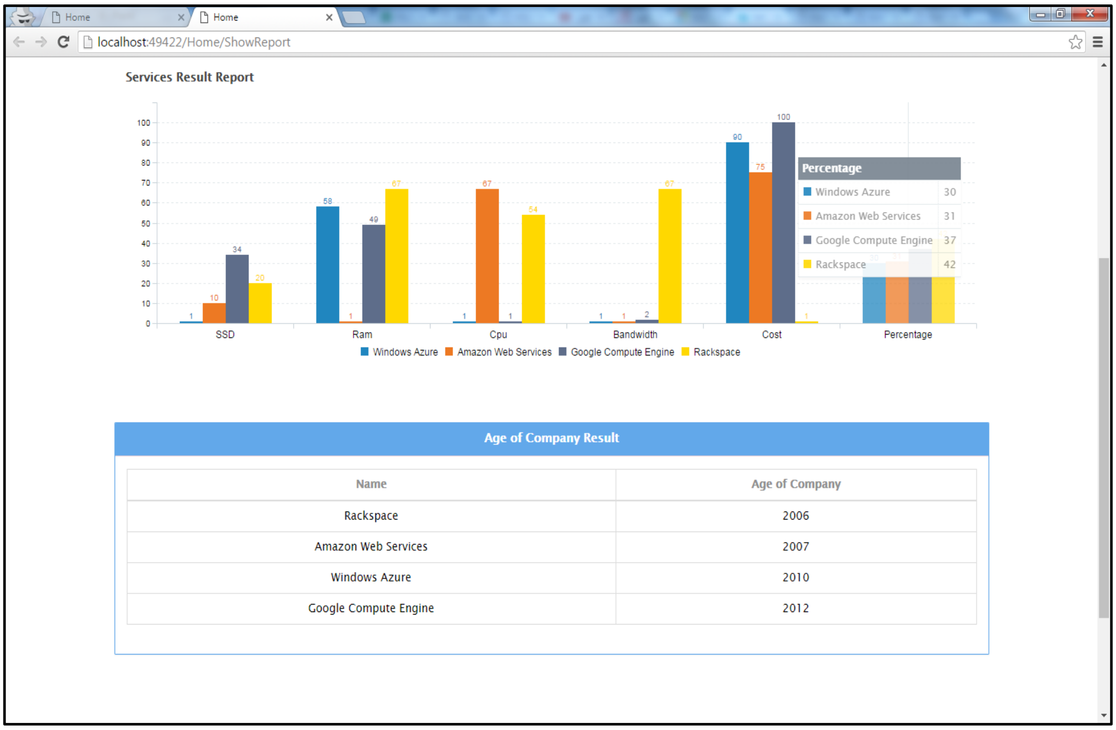


Figure 5.7: *Displayed the Services and Age of Company*

Figure 5.7 shows the screen of this functionality. The VPS previous services, in addition to the age of company (input) as shown in Figure 5.7. The output results are shown below in Figure 5.8.



Name	Age of Company
Rackspace	2006
Amazon Web Services	2007
Windows Azure	2010
Google Compute Engine	2012

Figure 5.8: *Final Result of Age of Company and Services*

If the experimental user is interested in the age of company service, the report outputs the final data as mentioned in Figure 5.8.

5.3.5 Functional and Non-Functional Services Selection

Assume that the experimental user selected reputation services. In this case, the user chooses the services that are mentioned in Figure 5.9, after the Cloud user submitted the services, the screen shows trust information of functional and non-functional services and finds that there are CSP Cloud providers. P1 represents AWS, P2 represents GCE, P3 represents Azure and P4 represents Rackspace. CP can provide services that meet the user requirements and requirements based on their intent.

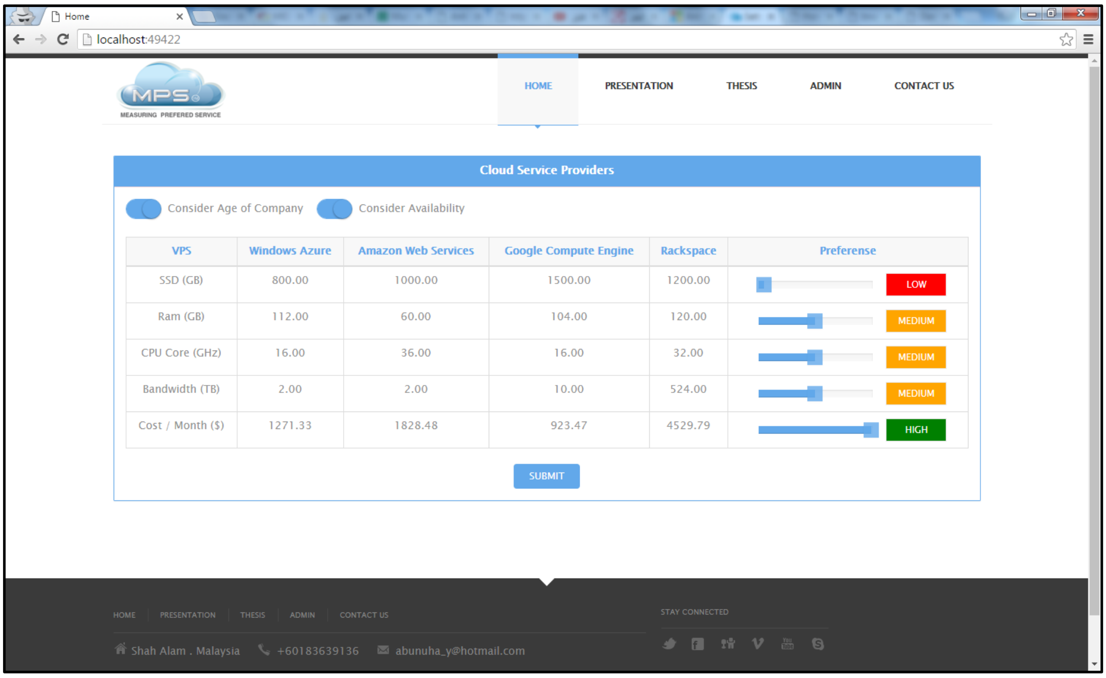


Figure 5.9: *Displayed VPS Services and Reputation*

This Figure leads the user to the optimal choice of the best service provider according to their preferences of function and non-function services.

5.4 BEST SERVICE PROVIDER SELECTION

The final results of preferred services are achieved by ranking the best service provider. The trusted way for provider evaluation is to make the comparison for the preferences of similar Cloud services. This comparison is usually dependent on the priority of function service and non-functional services, and measuring its functional services may be appropriately applied in Cloud best service provider selection. By combining these results, the final results usually reflect the real preferences of services for Cloud users.

5.5 TESTING AND EVALUATION

The proposed architecture is to test and evaluate presented experiments through its Cloud service selection, which is given to measure performance and to assess the Cloud service selection and evaluate their benefits including priority services for Cloud provider' candidates. When testing the Cloud service selection, figuring out so that the proposed system has the ability to weight dynamically when the number of services increase, to get the best cost and save time by using electronic searching rather than traditional ways that required more time and efforts.

For this purpose, the ranking system for four prestigious companies' services was created. This rank was made in the requirement of the user as the dynamic rank that evaluates the priority of users in Cloud Computing.

5.5.1 Testing MPS System

The proposed mechanism enables trust values of converted data to be given for workflows and Cloud Computing selection. This section tests and investigates one simple case to see how trust and realistic data are achieved in the Cloud services selection. The application of the MPS system is tested by platform for the experiments, which is utilized as priority services of the user, assumed to have percentage selecting (i.e. from 1% to 100 for each preferred service). Four providers offered volume and cost of services: the first provider is GCE which offers services. The second provider is AWS which offers services B, and the third provider is Rackspace which offers services C and the fourth provider is Azure which offers services D, with different prices and volume. A user asked for these services according to its priority.

The user requirements the services that can satisfy its need and priority. The user requirements to define the services that meet its needs and priority sequentially for a workflow. The required services of the task equals (volume/ service & cost /service), which means it is chosen according to its priority and application.

Only one provider provided the best available services for the user in MPS model. There are varying providers and various services are not the same. The user requires the same services (volume and price) but the valuation of the provider 1 which has the highest values among others and considered the best service provider, produces several services. The results were achieved in the MPS which shows that demands of services of providers are fulfilled.

Specifically, the services from the provider are properly computed and measured to the tasks. The proposed mechanism measuring the preferred services properly allocated the services to workflow tasks are indicated by these results.

Note that, the user achieved the optimal selection and defined the best service provider who satisfied its need. Provider 1 was the best in the competition to sell the services A for the user because their services meet with the priority of user, and the provided service meets its need and his application. The priority of user meets with provided services and the application offered by Provider 1 provided a total satisfaction.

The accurate results are written in the MPS Cloud selection. The possibility of satisfying the user of other services by the rest of providers who are not the best providers for the user because they have lower percentages in the final result values when measuring the priority of user. Provider 1 indeed still has enough capacity of satisfaction and acceptable for the service. In brief, the proposed mechanism has properly measured the services provided amongst a group of providers, various services and preferring needs for users, as confirmed through this primary experiment.

The outcome of the mechanism is reliable in general. However, how many providers can be added in our MPS model? The last section evaluates the provided services with reputation services provider in Cloud Computing when the estimation of mechanism overhead of MPS model does not intend to spend a long time when facing problems in the strategy of selecting. The evaluation estimates the effect of a number of providers and their various services on the runtime.

Generating a set of orders and running the MPS system mechanism are carried out by the simulation. The workflow in MPS model is assessed by the evaluation, and the result of matchmaking of orders does not affect the next orders. The percentages levels of services as preferred Low for SSD, Medium for RAM, Medium for CPU, Medium for Bandwidth and High for Cost service. Additionally, reputation is considered. The result of adding the total services must be 100%. The case of submission services of Cloud Computing represents trading in the model. The preferred services by the user depend on measuring the priority of user and reputation provider. For example, these services represent the cost in one month with volume. The number of providers is set to four, and one user who has a range of priority of services {Low/34%, Medium/67%, Medium/67%, Medium/67%, and High /100%}.

Details are discussed and arranged above in Figures which show the sell services of the providers. Each provider offers a unique service with cost and volume and all the services are available anytime and anywhere. The user is required to choose one to five services randomly out of seven services to be co-allocated and then the user can add reputation services. The task of Computing and measuring the various services from 1% to 100 % of the total of preferred services are added. This setting is intended to reflect the current situation of Cloud Computing, where some big companies provide their own large services and many users use services without defining which best service provider can satisfy their needs according to the requirements, and to execute its task. Other parameters are set as constant for the sake of simplicity.

The cost services of providers are offered to the experimental user. The valuation of a service depends on the priority of user in defining services. This setting means a loose supply-demand condition from a provider with price competition and different volume services, where the user's requirements are likely to be fulfilled. The simulation was conducted on cost service for providers in Cloud Computing and the final results are presented.

5.5.2 Evaluation the Performance of MPS Model

In this part, an experiment is carried out to evaluate the performance of measuring services in proposed model by evaluating the Measuring Preferred Services (MPS) in order to have its performance measured. The selection of best service provider is assessed, in order to evaluate the benefits of including preferred services in the prioritization of Cloud user’s candidates and showing that the proposed model is able to scale dynamically when the experimental user first chooses VPS services and cost when adding availability and age of company services. This selection allows services to be selected. Rank selection could change the value for each service and Computing isolated service and Compute the total sum of aggregation services produced new and different results according to the priority of user. Best selection to each individual provider is based on the best-provided service from the opinion of the user according to its preferred services that meet its need and application.

The best selection probability for each individual provider is calculated using the equation in Section 4.3.2. The experimental user chooses their preferred services as illustrated. The system produces one optimal choice among the providers AWS, GCE, Rackspace, and Azure companies in Cloud Computing.

For each isolated services and features, the time spent in performing requirements evaluation was measured, and the time to compute the priority of user for achieving results and goal was reasonable and quick, which is not more than 20 seconds.

5.6 VALIDATION OF MEASURING PREFERRED SERVICE MODEL (MPS)

In this section, will explain in details traditional method, the method of using MPS and then comparison and validation.

5.6.1 Traditional Method

First, in Table 5.1, the expert user by the traditional method will collect the values of services for each provider.

Second, determines the best value provided for each service by the providers.

Third, prioritize the best provider based his need. Hence the competition will become between the GCE and Rackspace as displayed in Table 5.2.

Then, the expert compares the value of each service for both providers. Hence, finds that Rackspace provider is better in terms of RAM, CPU, and Bandwidth as displayed in Table 5.2.

Finally, based on the expert's needs, focusing on the cost of the virtual server as a whole. The expert found that the cost of a Rackspace provider is much higher than that of the GCE. So, he decided to choose the GCE provider.

Table 5.1
Values of Services by The Traditional Method

	SSD	RAM	CPU	Bandwidth	Cost
GCE	**1500**	104	16	10,000	**$923.47**
Azure	800	112	16	2,000	$1271.33
AWS	1000	60	**36**	2,000	$1828.48
Rackspace	1200	**120**	32	**524,000**	$4529.79

Table 5.2
Comparison between the GCE and Rackspace

	SSD	RAM	CPU	Bandwidth	Cost
GCE	**1500**	104	16	10,000	**$923.47**
Rackspace	1200	**120**	**32**	**524,000**	$4529.79

5.6.2 The Method of Using MPS

The expert uses the MPS application through the main screen in Figure 5.10 and display the result. In Figure 5.10, the low level equal 34%, a medium level equal 67%, a high level equal 100 %.

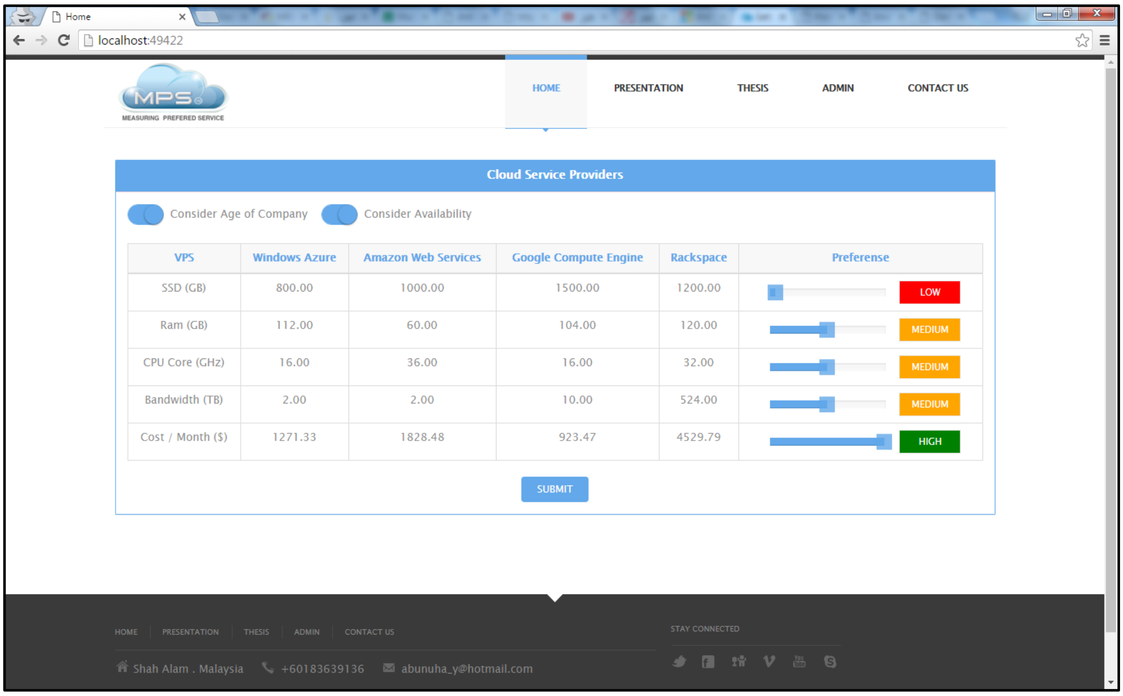


Figure 5.10: *Preferred Services of (VPS) for User in Cloud Computing*

First, the expert selected preferred services based on his need, which is displayed on Figure 5.10. The cost service got highest level, which is equal 100%.

Second, The MPS application calculated the values of services using the linear equation and then prioritize the providers accordingly as displayed in Figure 5.11.

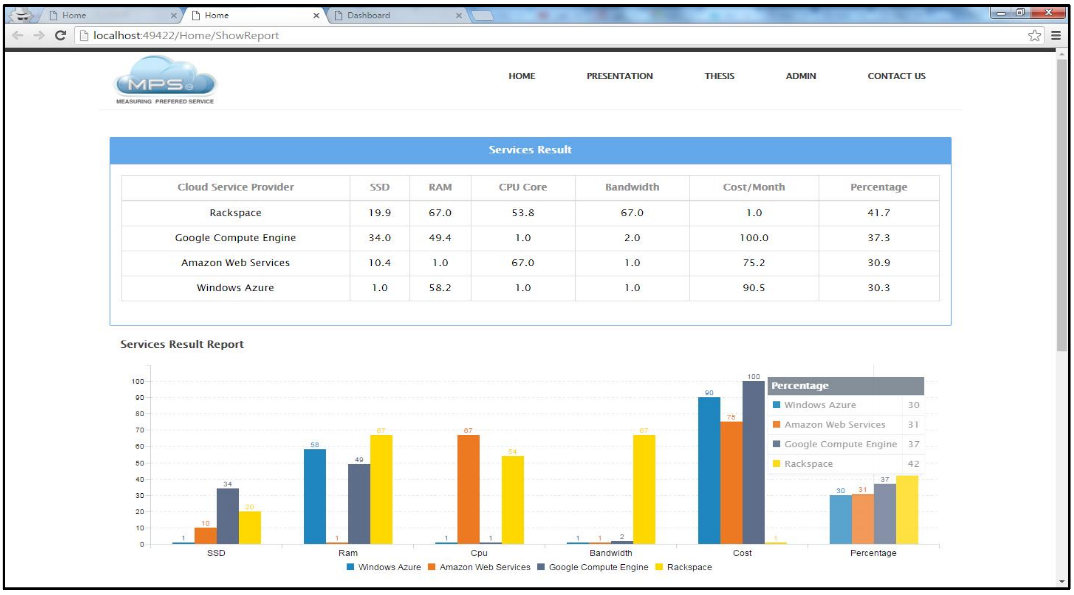


Figure 5.11: *The Priority of the Providers*

Finally, the MPS application gave preference to the Rackspace provider, with a total percentage of 41.7% for the virtual server as a whole.

5.6.3 Comparison and Validation

Based on Table 5.1 the expert chooses GCE provider based on cost with high priority. To reach this result he struggled to find the best option as the needed service provider must also provide CPU, RAM and Bandwidth services with medium priority. After searching providers' websites, spending time in studying those options, analyzing them he reached that result.

The expert tried his best to find a solution that mixes between cost with its high priority and other mentioned needed services that have medium priority. At the end he chooses GCE, but when looking deeply into the available options and with the help of MPS application result. Look in a table inside Figure 5.11, it will be found that GCE is less efficient than Rackspace in term of RAM, CPU, and Bandwidth which are all important to the end user, that's why Rackspace is the best provider for this case.

The expert found that the MPS application results are more efficient and reliable as it covered all needed aspects based on priority levels needed.

5.7 SUMMARY

Experimental results on selection strategies in MPS algorithms based on the two selection strategies are tested on examination timetabling problem, and the comparison between them is done to examine the performance of these providers of Cloud Computing with different selection strategies from the user side. The following algorithms are based on two selection strategies mentioned above in Section 3. MPS system has been developed to explore optimal selecting for the best service provider in Cloud Computing by means of multi-provider simulations.

This selection allows both higher/best and lower/good services to be selected. Rank selection could change the value for each service and Computing isolated service and Compute the total sum of aggregation services produced new and different results according to the priority of user. Best selection to each individual provider is based on best-provided service from the opinion of the user according to their preferred services that meet its need and application. The best selection probability for each individual provider is calculated using the equation in Chapter 4.

To suggest a set of possible configurations prioritized by the user, the optimal selection decision for the best service provider receives the Cloud resources requirements. Preferred features and services which include new limitations along with the priority given by the administrator to selected services should be considered to select adequate providers. The model is based on the priority of users, but it has basically been improved. First, propose that extending the database plan and adding new screens to store the values that Cloud providers have for new services. For example, downtime and age of company services could be added to store the information related to the history of different Cloud providers and availability to include a minimum percentage of downtime and maximum of the age of provider to sort the candidate providers.

CHAPTER SIX

CONCLUSION

This chapter summarizes the research work on making optimal selection mechanisms for best service of IaaS Cloud providers' and highlights the major findings in this thesis. The research contributions are described in Section 7.2. Section 7.3 outlines the future research work and outlook that may be considered. Finally, Section 7.4 lists the papers that have been published or submitted for journals and conference proceedings.

6.1 RESEARCH CONCLUSION

It has become difficult for the Cloud users to select the best Cloud provider that suits an individual's basic requirements with the expansion of numerous Cloud providers in the technology world. The new user is unable to understand which quality of basic attributes are more important for their application and thinks more on which company from the vast and various providers can be trusted. MPS model not only helps users to select the best service provider but also to understand their own requirements and serves to assign a ranking on provider companies to make the process of researching for providers easier.

Future works will be more comprehensive and take into account the reputation of providers which contributes to a high trust efficiency. It is intended to apply this in the model in a practical way. In addition, it helps provider companies to judge their performance by trying to face the challenges that come with increased users with the various priority of services. Knowing more about the requirements of users, competitor companies and create a computational environment for achieving the main goal of this thesis which is a proposed model that helps the user to easily discover the optimal choice for best service Provider Company that can satisfy its needs of Cloud services.

6.2 CONTRIBUTIONS

An intelligent model for best service provider selection in the Cloud Computing is desired to be established. This thesis proposed a flexible model for measuring services according to preferences of users and a reputation-of-provider based mechanism to support user's workflow in Cloud Computing-based applications built on the Cloud services selection. It employs an exact optimization technique to achieve a proposed model of measuring services. The proposed model consists of a Cloud selection for an advance selecting and defining of services and provider for immediate preferred and defined services so that the users can reliably plan the selection of Cloud services provider within needs and preferences.

Some experiments have been carried out in this thesis: testing the measuring services based on the priority of user in MPS model and evaluation of its performance. Some experiments demonstrated that the proposed mechanism succeeded to do matchmaking between preferences services of users and combinatorial demands according to the selecting system, as well as to determine their needs according to its priority and reputation of the provider.

The performance of measuring services mechanism of matchmaking and schemes of the proposed mechanism is also evaluated, and showed that the overheads and results are acceptable in an expected Cloud Computing selection. When comparing four prestigious providers in the selection mechanism, it showed that the proposed mechanism brought a superior performance in both individual usability and system-wide efficiency. In addition, the experiments simulated four providers with their provided services and users' preference services and concluded that the more users preferred the provider who has a high reputation in MPS model, the more performance, and comprehension in selection could be achieved in our model.

6.3 FUTURE WORKS

Sophisticated strategies of optimal selecting for the best service provider can significantly improve the performance of user's selection of services depending on preferred services of user and reputation of the company. For instance, a user can reduce wasted resources and time, and can increase workflow completion rate by employing a smarter strategy to make their orders from provider companies. Moreover, it is essential for user/provider to adjust their services price and volume according to the needs of users and available service from providers. The future work, therefore, investigates the MPS model behavior using more sophisticated strategies of user/provider in Cloud Computing.

Interesting research objectives include the autonomous behavior of the services providers, particularly the interaction between the user and the provider, where the services type, volume, and price are expected to be more trusted and accurate. The best service provider with high reputation in the market indicates the high reputation of the provider. By observing the provided services, a user can define the preferred services and avoid purchasing an unnecessarily high-priced service or unneeded services which cannot satisfy their requirement. As a result, the preferred service ratio and the provider cost are expected to be smooth and not stable. This can be seen as balancing selection of provider Cloud achieved by the optimal design, which cannot be realized by making the optimal decision of the provider alone without putting the priority of user and reputation of the provider in an account. MPS model can be extended to make an automated selection of web services in Cloud Computing viable and changeable and extend the MPS system capabilities to include new and added web services.

6.4 OUTLOOK

At the end of the thesis, imagine the world where the Cloud service selection comes true. The proposed model mechanism is so generic that any kind of service selection can be traded and measured equally in it. In a realistic scenario, however, the MPS model is used hierarchically as "model for measuring preference service of the user and sorting the Cloud provider according to user's priority". For instance, a bundle of a low percentage - services provider" and a high-percentage.". The user is likely to be ordered a high percentage of services of provider". There appears to be a hierarchical structure within the MPS model that helps new users define their needs in Cloud Computing and make the optimal decision for selecting the best services provider.

The overall concern of the research presented in this thesis was to investigate how the different models of selecting the best service provider and measuring the QOS behavior algorithms could improve the available search methodologies for examining the optimal selection problems.

The idea behind the use of measuring and Computing services algorithms (which is classified as optimal decision of Cloud selection based on priority of user and reputation of provider) is to bring solutions in the Cloud Computing service to be as good as possible simultaneously, and to create a balance between exploration and exploitation (algorithms and previous search algorithms) provider Cloud selection and user Cloud selection, in order to improve the quality of the examination model.

In this thesis, a number of research questions have been drawn to identify the requirements of a fruitful approach with respect to the problem being tackled. Firstly, the impact of using two selection strategies, namely user Cloud selection, and provider in the basic MPS algorithms over examination of selecting the best service provider problems in order to achieve the main goal of this thesis and better solutions are investigated. The findings have shown that the disruptive selection strategy was best compared to the standard selection strategy embedded in the basic measuring preferred service of user algorithms. Secondly, in order to increase and improve the performance of MPS model and the mechanism search in the algorithms, an MPS system mechanism that can adaptively select a neighborhood structure to be employed during the search progress was investigated. This was proven by the obtained results that were better than the one without the self-adaptive mechanism.

Thirdly, in order to create a balance between the exploration and exploitation, and Cloud selection behavior algorithms. Some algorithms experimented are the simulated annealing and the late acceptance hill climbing. (As presented in Chapter five). Finally, from the list of experimental results, the Measuring Preferred Services (MPS) with different modifications has proven to be the best approach in this work.

REFERENCES

Aca, S. (2010). Cloud Computing: What it is and what it can do for you. Retrieved 20 Jun 2015, from www.globalknowledge.com

Adamuthe, A. C., Tomke, J. V, & Thampi, G. T. (2014). Technology Forecasting: The Case of Cloud Computing and Sub-technologies. *International Journal of Computer Applications*, *106*(2).

Adinolfi, O., Cristaldi, R., Coppolino, L., & Romano, L. (2012). QoS-MONaaS: a portable architecture for QoS monitoring in the Cloud. In *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on* (pp. 527–532).

Afshari, A., Mojahed, M., & Yusuff, R. M. (2010). Simple additive weighting approach to personnel selection problem. International Journal of Innovation, Management, and Technology, 1(5), 511.

Alhamazani, K., Ranjan, R., Rabhi, F., Wang, L., & Mitra, K. (2012). Cloud monitoring for optimizing the QoS of hosted applications. In *Cloud Computing Technology and Science (Cloud Com), 2012 IEEE 4th International Conference on* (pp. 765–770).

Alpert, C., Quay, S., & Gandham, G. (2003). Computer-Aided Design of Integrated Circuits and Systems. *IEEE Transactions on*, *22*(4).

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., & Mock, S. (2004, June). Kepler: an extensible system for design and execution of scientific workflows. In *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on* (pp. 423-424). IEEE.

Amazon EC2 Pricing, retrieved July 22, 2014, from https://aws.amazon.com/ec2/pricing/

André, F. M. (2011). Availability and Confidentiality in Storage Clouds Availability and Confidentiality. CARNEGIE MELLON UNIVERSITY.

Antoniou, A. (2012). *Performance Evaluation of Cloud Infrastructure using Complex Workloads*. TU Delft, Delft University of Technology.

Arias, D. C. B. (2013). Resource Provisioning in Clouds via Non-Functional Requirements (Doctoral dissertation, The University of Melbourne).

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M., 2010. A view of Cloud Computing. *Communications of the ACM, 53(4), pp.50-58.*

Atalay, K. D., Eraslan, E., & Çinar, M. O. (2015). A hybrid algorithm based on fuzzy linear regression analysis by quadratic programming for time estimation: an experimental study in manufacturing industry. Journal of Manufacturing Systems, 36, 182-188.

Avikal, S., Mishra, P. K., & Jain, R. (2013). An AHP and PROMETHEE methods-based environment friendly heuristic for disassembly line balancing problems. *Interdisciplinary Environmental Review, 14(1), 69-85.*

AWS. (2017, February 17). What Is AWS Lambda? Retrieved July 07, 2017, from http://docs.aws.amazon.com/lambda/latest/dg/welcome.html

Baranwal, G., & Vidyarthi, D. P. (2014). A model for selection of best Cloud service provider using ranked voting method. In *Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014* (pp. 831–837).

Becker, C. H., Nickolas, S. E., & Vicknair, W. E. (2014). *U.S. Patent No. 8,645,862.* Washington, DC: U.S. Patent and Trademark Office.

Bibi, S., Katsaros, D., & Bozanis, P. (2010). Application development: Fly to the Clouds or stay in-house? In *Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on* (pp. 60–65).

Binnig, C., Kossmann, D., Kraska, T., & Loesing, S. (2009). How is the weather tomorrow?: towards a benchmark for the Cloud. In *Proceedings of the Second International Workshop on Testing Database Systems* (p. 9).

Boss, G., Malladi, P., Quan, D., Legregni, L., & Hall, H. (2007). Cloud Computing. IBM white paper, 321, 224-231.

Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H.F., Thatte, S. and Winer, D., 2000. Simple object access protocol (SOAP) 1.1.

Brockwell, P. J., & Davis, R. A. (2006). *Introduction to Time Series and Forecasting. Design* (Vol. 102). Springer Science {&} Business Media.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud Computing and emerging IT platforms: Vision, hype, and reality for delivering

Computing as the 5th utility. *Future Generation Computer Systems*, *25*(6), 599–616.

Callewaert, P., Robinson, P. A., & Blatman, P. (2009). Cloud Computing Forecasting change. White Paper.

Cao, B.-Q., Li, B., & Xia, Q.-M. (2009). A Service-Oriented QoS-Assured and Multi-Agent Cloud Computing Architecture. In *Cloud Computing, Proceedings* (Vol. 5931, pp. 644–649). Springer.

Carroll, M., Kotze, P., & Van der Merwe, A. (2012). A risk and control framework for Cloud Computing and Virtualisation: enabling technologies. CSIR Science Scope, 6(2), 62-63.

Castillo, I., Schmidt-Hieber, J., & Van der Vaart, A. (2015). Bayesian linear regression with sparse priors. The Annals of Statistics, 43(5), 1986-2018.

Cavalcante, E., Batista, T., Lopes, F., Delicato, F. C., Pires, P. F., Rodriguez, N., Mendes, R. (2012). Optimizing services selection in a Cloud multiplatform scenario. In *Proceedings of the 2012 IEEE Latin America Conference on Cloud Computing and Communications, Latin Cloud 2012* (pp. 31–36).

Cavalcante, E., Lopes, F., Batista, T., Cacho, N., Delicato, F. C., & Pires, P. F. (2011). Cloud Integrator: Building value-added services on the Cloud. In *Proceedings - 2011 1st International Symposium on Network Cloud Computing and Applications, NCCA 2011* (pp. 135–142).

Chang, C.-W., Liu, P., & Wu, J.-J. (2012a). Probability-based Cloud Storage Providers Selection Algorithms. In *Parallel Processing (ICPP), 2012 41st International Conference on* (pp. 199–208).

Chang, S. Y., Lai, C. F., & Huang, Y. M. (2012b). Dynamic adjusTable multimedia streaming service architecture over Cloud Computing. *Computer Communications*, *35*(15), 1798–1808.

Chang, V., Wills, G., Walters, R. J., & Currie, W. (2012c). Towards a Structured Cloud ROI: The University of Southampton Cost. *Sustainable ICTs and Management Systems for Green Computing*, 179–200.

Chunlin, L., & Layuan, L. (2014). Multi-layer resource management in Cloud Computing. *Journal of network and systems management, 22(1), 100.*

Claro, D. B., Albers, P., & Hao, J. (2006). Web services composition. In *Semantic Web Services, Processes and Applications* (pp. 195–225). Springer.

Clavister. (2010). Security in the Cloud. Retrieved 13 November 2015, from https://www.clavister.com/globalassets/documents/resources/white-papers/clavister-whp-Cloud -security-en.pdf

Cloud Computing. (2016). Retrieved August 30, 2016, from https://en.wikipedia.org/wiki/Cloud _Computing.

Cloud Harmony. (2014). Retrieved January 2, 2014, from https://Cloud harmony.com/status-1year-of-storage-and-compute-group-by-regions-and-provider.

Cloud Survey Report | KPMG | GLOBAL. Retrieved March 19, 2016 from https://home.kpmg.com/xx/en/home/insights/2012/02/business-implications-Cloud .html.

Cloudmonix. (2017, February 17). What is Web Role and Worker Role in Azure? Retrieved July 07, 2017, from http://www.cloudmonix.com/blog/what-is-web-and-worker-role-in-microsoft-azure/

Cost savings, efficiencies lead IT pros to Cloud Computing. (2016). Retrieved August 28, 2016, from http://searchCloud Computing.techtarget.com/feature/Cost-savings-efficiencies-lead-IT-pros-to-Cloud -Computing

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002). Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, *6*(2), 86-93.

Danielsson, P. E. (1980). Euclidean distance mapping. *Computer Graphics and image processing*, *14*(3), 227-248.

Dastjerdi, A., & Buyya, R. (2011). A Taxonomy of QoS Management and Service Selection Methodologies for Cloud Computing.

Deelman, E., Singh, G., Livny, M., Berriman, B., & Good, J. (2008). The cost of doing science on the Cloud: the montage example. In *Proceedings of the 2008 ACM/IEEE Conference on SuperComputing* (p. 50).

Dew, P. M., Schmidt, J. G., Thompson, M., & Morris, P. (2003). The white rose grid: practice and experience. In *Proceedings of the 2nd UK all hands e-science meeting. EPSRC*.

Ding, S., Yang, S., Zhang, Y., Liang, C., & Xia, C. (2014). Combining QoS prediction and customer satisfaction estimation to solve Cloud service trustworthiness evaluation problems. *Knowledge-Based Systems*, *56*, 216–225.

Eerola, P., Kónya, B., Smirnova, O., Ekelöf, T., Ellert, M., Hansen, J. R., Vinter, B. (2003). The Nordugrid production grid infrastructure, status and plans. In

Proceedings - IEEE/ACM International Workshop on Grid Computing (Vol. 2003-Janua, pp. 158–165).

Espadas, J., Molina, A., Jiménez, G., Molina, M., Ramírez, R., & Concha, D. (2013). A tenant-based resource allocation model for scaling Software-as-a-Service applications over Cloud Computing infrastructures. *Future Generation Computer Systems, 29(1), 273-286.*

Evangelinos, C., & Hill, C. (2008). Cloud Computing for parallel scientific HPC applications: Feasibility of running coupled atmosphere-ocean climate models on Amazon's EC2. *Ratio*, *2*(2.40), 2–34.

Fan, Z., Zhang, L., Shen, J., & Wang, S. (2010). A User's Preference Based Method for Web Service Selection. In *2010 Second International Conference on Computer Research and Development* (pp. 39–45).

Figiela, K. (2013). Optimization of Resource Allocation on the Cloud.

Foster, I., Kesselman, C., & Tuecke, S. (2001). The Anatomy of the Grid-Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, *15*(3), 200–222.

Fujiwara, I. (2011). *Study on Combinatorial Auction Mechanism for Resource Allocation in Cloud Computing Environment*. *Nii.Jp*. Ph. D. dissertation, The Graduate University for Advanced Studies :(SOKENDAI).

Fülöp, J. (2005). Introduction to decision making methods. In *BDEI-3 workshop, Washington.*

Garg, S. K., Versteeg, S., & Buyya, R. (2011). SmiCloud: A Model for Comparing and Ranking Cloud Services. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on* (pp. 210–218).

Garg, S. K., Versteeg, S., & Buyya, R. (2013). A model for ranking of Cloud Computing services. *Future Generation Computer Systems*, *29*(4), 1012–1023.

Gentzsch, W. (2001). Sun Grid Engine: Towards creating a compute power grid. In *Proceedings - 1st IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGrid 2001* (pp. 35–36.

Godse, M., & Mulik, T. (2009). An Approach for Selecting Software-as-a-Service (SaaS) Product. In *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on* (pp. 155–158).

Godse, M., Bellur, U., & Sonar, R. (2011). A Taxonomy and Classification of Web Service QoS Elements. *Int. J. Commun. Netw. Distrib. Syst.*, *6*(2), 118–141.

Google Compute Engine Pricing, retrieved July 22, 2014, from https://Cloud.google.com/compute/pricing, 2014.

Goscinski, A., & Brock, M. (2010). Toward dynamic and attribute based publication, discovery and selection for Cloud Computing. *Future Generation Computer Systems*, *26*(7), 947–970.

Gui, Z., Yang, C., Xia, J., Huang, Q., Liu, K., Li, Z., Jin, B. (2014). A service brokering and recommendation mechanism for better selecting Cloud services. *PloS One*, *9*(8), e105297.

Gupte, N., & Wang, J. (2015). Secure power grid simulation on Cloud. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*,*34*(3), 422-432.

Habib, S. M., Ries, S., & Muhlhauser, M. (2011). Towards a Trust Management System for Cloud Computing. In *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications* (pp. 933–939).

Hall, P., & Miller, H. (2012). Using generalized correlation to effect variable selection in very high dimensional problems. *Journal of Computational and Graphical Statistics*.

Han, S. M., Hassan, M. M., Yoon, C. W., & Huh, E. N. (2009, November). Efficient service recommendation system for Cloud Computing market. In *Proceedings of the 2nd international conference on interaction sciences: information technology, culture and human* (pp. 839-845). ACM.

Han, T., & Sim, K. M. (2010, March). An ontology-enhanced cloud service discovery system. *In Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1, No. 2010, pp. 17-19).

Harrell, F. (2015). *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer.

He, Q., Han, J., Yang, Y., Grundy, J., & Jin, H. (2012, June). QoS-driven service selection for multi-tenant SaaS. In *Cloud Computing (Cloud), 2012 IEEE 5th international Conference on* (pp. 566-573). IEEE.

Heroku Dev Center. (2017, February 17). Worker Dynos, Background Jobs and Queueing. Retrieved July 07, 2017, from https://devcenter.heroku.com/articles/background-jobs-queueing
https://devcenter.heroku.com/articles/background-jobs-queueing

Iosup, A., Yigitbasi, N., & Epema, D. (2011, May). On the performance variability of production Cloud services. In *Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on* (pp. 104-113). IEEE.

Itani, W., Ghali, C., Kayssi, A., & Chehab, A. (2014). Reputation as a Service: A System for Ranking Providers in Cloud Systems. In Security, Privacy and Trust in Cloud Systems (pp. 375-406). Springer Berlin Heidelberg.

Jacobs, P., & Davie, B. (2005). Technical challenges in the delivery of interprovider QoS. *IEEE Communications Magazine*, *43*(6), 112–118.

Jahani, A., & Khanli, L. (2016). Cloud service ranking as a multi objective optimization problem. *Journal of Supercomputing, 72*(5).

Jahani, A., Khanli, L. M., & Razavi, S. N. (2014). A QoS Based Ranking Approach for Cloud Computing. *Computer Engineering and Applications Journal*, *4274*(2), 55–62.

Jula, A., Sundararajan, E., & Othman, Z. (2014). Cloud Computing service composition: A systematic literature review. Expert Systems with Applications, 41(8), 3809-3824.

Jung, G., Mukherjee, T., Kunde, S., Kim, H., Sharma, N., & Goetz, F. (2013, June). Cloud Advisor: A recommendation-as-a-service platform for Cloud configuration and pricing. In *2013 IEEE Ninth World Congress on Services* (pp. 456-463). IEEE.

Kafetzakis, E., Koumaras, H., Kourtis, M. A., & Koumaras, V. (2012, July). QoE4CLOUD: A QoE-driven multidimensional framework for cloud environments. In *Telecommunications and Multimedia (TEMU), 2012 International Conference on* (pp. 77-82). IEEE.

Khan, U. T., & Valeo, C. (2016). Short-Term Peak Flow Rate Prediction and Flood Risk Assessment Using Fuzzy Linear Regression. Journal of Environmental Informatics, 28(2).

Kourtesis, D., Alvarez-Rodríguez, J. M., & Paraskakis, I. (2014). Semantic-based QoS management in Cloud systems: current status and future challenges. Future Generation Computer Systems, 32, 307-323.

Kumar, N., & Agarwal, S. (2014). *QoS based Enhanced Model for Ranking Cloud Providers*. MAGNT Research Report (ISSN. 1444-8939) Vol. 2 (6). PP: 32-39. DOI: dx. doi. org/14.9831/14448939.2014/2-6/MAGNT. 4.

Lee, S., & Walsh, P. (2011). SWOT and AHP hybrid model for sport marketing outsourcing using a case of intercollegiate sport. *Sport Management Review*, *14*(4), 361–369.

Leitner, P., & Cito, J. (2016). Patterns in the chaos—a study of performance variation and predictability in public IaaS clouds. ACM Transactions on Internet Technology (TOIT), 16(3), 15.

Lemos, A. L., Daniel, F., & Benatallah, B. (2016). Web service composition: a survey of techniques and tools. *ACM Computing Surveys (CSUR), 48*(3), 33.

Lenk, A., Menzel, M., Lipsky, J., Tai, S., & Offermann, P. (2011, July). What are you paying for? performance benchmarking for infrastructure-as-a-service offerings. *Cloud Computing (CLOUD), 2011 IEEE International Conference on* (pp. 484-491). IEEE.

Levitt, J. (2001). From EDI to XML and UDDI: A brief history of web services. *Information Week*.

Li, A., Yang, X., Kandula, S., & Zhang, M. (2011). Comparing public-Cloud providers. *IEEE Internet Computing*, *15*(2), 50–53.

Li, L., & Wang, Y. (2008). A trust vector approach to service-oriented applications. In *Proceedings of the IEEE International Conference on Web Services, ICWS 2008* (pp. 270–277).

Li, L., & Wang, Y. (2010). Subjective Trust Inference in Composite Services. In *Aaai* (pp. 1377–1384).

Li, L., Wang, Y., & Varadharajan, V. (2009, July). Fuzzy regression based trust prediction in service-oriented applications. In International Conference on Autonomic and Trusted Computing (pp. 221-235). Springer Berlin Heidelberg.

Limam, N., & Boutaba, R. (2010). Assessing software service quality and trustworthiness at selection time. *IEEE Transactions on Software Engineering*, *36*(4), 559–574.

Liu, Y., Ngu, A. H., & Zeng, L. Z. (2004). QoS computation and policing in dynamic web service selection. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (pp. 66–73).

Malik, A., & Nazir, M. M. (2012). Security Framework for Cloud Computing environment: A review. *Journal of Emerging Trends in Computing and Information Sciences,* 3(3), 390-394.

Mamoun, M. H., & Ibrahim, E. M. (2014). A Proposed Model for Ranking and Reservation of Cloud Services. *International Journal of Engineering and Technology*, *4*(9), 536–541.

Manuel, P. (2015). A trust model of Cloud Computing based on Quality of Service. *Annals of Operations Research*, *233*(1), 281–292.

Martens, B., Teuteberg, F., & Gräuler, M. (2011). Design and Implementation of a Community Platform for the Evaluation and Selection of Cloud Computing Services: A Market Analysis. In *ECIS 2011 Proceedings*.

McIlraith, S. A., Son, T. C., & Zeng, H. (2001). Semantic web services. *IEEE intelligent systems*, 16(2), 46-53.

Mell, P., & Grance, T. (2009). Cloud Computing Definition. Retrieved February 26, 2015, from http://csrc.nist.gov/groups/SNS/Cloud -Computing

Menzel, M., Schönherr, M., & Tai, S. (2013). (MC2) 2: criteria, requirements and a software prototype for Cloud infrastructure decisions. *Software: Practice and experience, 43(11), 1283-1297.*

Microsoft ASP.NET. Retrieved July 2, 2016, from http://www.asp.net/signalr/overview/getting-started/introduction-to-signalr

Miyashiro, R., & Takano, Y. (2015). Mixed integer second-order cone programming formulations for variable selection in linear regression. European Journal of Operational Research, 247(3), 721-731.

Moghaddam, M., & Davis, J. G. (2014). Service selection in web service composition: A comparative review of existing approaches. *In Web Services Foundations* (pp. 321-346). Springer New York.

Moon, H. R., & Weidner, M. (2015). Linear regression for panel with unknown number of factors as interactive fixed effects. Econometrica, 83(4), 1543-1579.

Murty, K. G. (2009). Optimization for decision making. Springer.

Naseer, M. K., Jabbar, S., & Zafar, I. (2014, January). A novel trust model for selection of Cloud service provider. In Computer Applications & Research (WSCAR), 2014 World Symposium on (pp. 1-6). IEEE.

National Grid Service. (2016). Retrieved September 1, 2016, from https://en.wikipedia.org/wiki/National_Grid_Service

Nizamani, S. A. (2012). A quality-aware Cloud selection service for computational modellers. University of Leeds.

Pathan, M., Broberg, J., & Buyya, R. (2009). *Maximizing utility for content delivery Clouds*. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5802 LNCS). Springer.

Paton, N., De Aragão, M. A., Lee, K., Fernandes, A. A., & Sakellariou, R. (2009). Optimizing utility in Cloud Computing through autonomic workload execution. *Bulletin of the Technical Committee on Data Engineering*, *32*(1), 51-58.

Pauley, W. (2010). Cloud provider transparency: An empirical evaluation. *IEEE Security and Privacy*, *8*(6), 32–39.

Perrey, R., & Lycett, M. (2003, January). Service-oriented architecture. In *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on* (pp. 116-119). IEEE.

Pratt, T. W., Zelkowitz, M. V, & Gopal, T. V. (1984). *Programming languages: design and implementation*.

Qu, L., Wang, Y., & Orgun, M. a. (2013). Cloud Service Selection Based on the Aggregation of User Feedback and Quantitative Performance Assessment. In *2013 IEEE International Conference on Services Computing* (pp. 152–159).

Qu, L., Wang, Y., Orgun, M. A., Liu, L., & Bouguettaya, A. (2014). Cloud service Selection based on contextual subjective assessment and objective assessment. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems* (pp. 1483–1484). inproceedings.

Rackspace Public Cloud, retrieved July 22, 2014, from https://www.rackspace.com/Cloud /public-pricing, 2014

Raderbauer, M. (2011). The importance of sustainable business practices in the viennese accommodation industry. *University of Exeter as a Thesis for the Degree of Master of Science*.

Rehman, Z., Hussain, O. K., Parvin, S., & Hussain, F. K. (2012). A Model for User Feedback Based CLOUD SERVICE Monitoring. In *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems* (pp. 257–262). http://doi.org/10.1109/CISIS.2012.157

Rezaei, J. (2015). Best-worst multi-criteria decision-making method: Some properties and a linear model. *Omega*.

Roy, B. (1991). The outranking approach and the foundations of ELECTRE methods. *Theory and Decision*, *31*(1), 49–73.

Saaty, T. L. (2008a). Decision making with the analytic hierarchy process. *International Journal of Services Sciences, 1*(1), 83–98.

Saaty, T. L. (2008b). Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process. *RACSAM-Revista de La Real Academia de Ciencias Exactas, Fisicas Y Naturales. Serie A. Matematicas*, *102*(2), 251–318.

Sakhuja, S., Jain, V., & Dweiri, F. (2015). Application of an integrated MCDM approach in selecting outsourcing strategies in hotel industry. *International Journal of Logistics Systems and Management, 20(3), 304-324.*

Salama, M., Shawish, A., Zeid, A., & Kouta, M. (2012). Integrated QoS utility-based model for Cloud Computing service provider selection. In *Proceedings - International Computer Software and Applications Conference* (pp. 45–50).

San Cristóbal, J. R. (2012). *Multi criteria analysis in the renewable energy industry*. Springer Science & Business Media.

Saravanan, K., & Kantham, M. L. (2013). An enhanced QoS Architecture based Model for Ranking of Cloud Services. *International Journal of Engineering Trends and Technology (IJETT)*, 4(4), 1022–1031.

Saripalli, P., & Pingali, G. (2011). Madmac: Multiple attribute decision methodology for adoption of Clouds. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on* (pp. 316–323).

Serhani, M. A., Dssouli, R., Hafid, A., & Sahraoui, H. (2005, July). A QoS broker based architecture for efficient web services selection. In *IEEE International Conference on Web Services (ICWS'05)* (pp. 113-120). IEEE.

Shaw, K., Shankar, R., Yadav, S. S., & Thakur, L. S. (2012). Provider selection using fuzzy AHP and fuzzy multi-objective linear programming for developing low carbon supply chain. *Expert Systems with Applications*, *39*(9), 8182–8192.

Shinder, D. (2013). Selecting a Cloud provider (Part 1). Retrieved January 1, 2014, from http://www.cloudcomputingadmin.com/articles-tutorials/architecture-design/selecting-cloud-provider-part1.html.

Silas, S., Rajsingh, E. B., & Ezra, K. (2012). Efficient service selection middleware using ELECTRE methodology for Cloud environments. *Information Technology Journal*, *11*(7), 868.

Sirin, E., & Hendler, J. (2003). Semi-automatic composition of Web services using semantic descriptions. In *1st Workshop on Web Services:* (pp. 17–24).

Subha, M. M., & Saravanan, K. (2013). Achieve Better Ranking Accuracy Using Cloud Rank Model for Cloud Services. *arXiv Preprint arXiv:1402.2509*, *6*(6), 307–312.

Sun, L., Dong, H., Hussain, F. K., Hussain, O. K., & Chang, E. (2014). Cloud service selection: State-of-the-art and future research directions. *Journal of Network and Computer Applications*, *45*, 134–150.

Sundareswaran, S., Squicciarini, A., & Lin, D. (2012, June). A brokerage-based approach for Cloud service selection. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on* (pp. 558-565). IEEE.

Sycara, K., Paolucci, M., Ankolekar, A., & Srinivasan, N. (2003). Automated discovery, interaction and composition of Semantic Web services. *Web Semantics*, *1*(1), 27–46.

Technologicasolcom. (2017). Technologicasolcom. Retrieved 21 June, 2017, from http://www.technologicasol.com/cloud-services.html

Tran, V. X., Tsuji, H., & Masuda, R. (2009). A new QoS ontology and its QoS-based ranking algorithm for Web services. *Simulation Modelling Practice and Theory*, 17(8), 1378–1398.

Triantaphyllou, E. (2000). Multi-Criteria Decision Making Methods. In *Multi-Criteria Decision Making Methods: A Comparative Study* (pp. 5–21). Springer.

Vazana, Y. (2016). How Browsers Render Web Pages, And What Can You Do To Help Them. Retrieved March 25, 2016, from https://www.linkedin.com/pulse/how-browsers-render-web-pages-what-can-you-do-help-them-yaron-vazana

Virtual Machines Pricing, retrieved July 22, 2014, from https://azure.microsoft.com/en-us/pricing/details/virtual-machines/#Linux, 2014

Walterbusch, M., Martens, B., & Teuteberg, F. (2013). Evaluating Cloud services from a total cost of ownership perspective. *Management Research Review, 36(6), 613-638.*

Wang, F.-Y., Zhang, H., & Liu, D. (2009). Adaptive Dynamic Programming: An Introduction. *IEEE Computational Intelligence Magazine*, *4*(May), 39–47. http://doi.org/10.1109/MCI.2009.932261.

Wang, S., Zheng, Z., Sun, Q., Zou, H., & Yang, F. (2011, April). Cloud model for service selection. In Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on (pp. 666-671). IEEE.

Web services - Gartner IT Glossary. Retrieved March 16, 2016, from http://www.gartner.com/it-glossary/web-services/

Whaiduzzaman, M., Haque, M. N., Rejaul Karim Chowdhury, M., & Gani, A. (2014). A study on strategic provisioning of Cloud Computing services. The Scientific World Journal, 2014.

Wittern, E., Kuhlenkamp, J., & Menzel, M. (2012, November). Cloud service selection based on variability modeling. In *International Conference on Service-Oriented Computing* (pp. 127-141). Springer Berlin Heidelberg.

Worker Dynos, Background Jobs and Queueing. Retrieved March 16, 2016, from

Wu, J., Chen, L., Feng, Y., Zheng, Z., Zhou, M. C., & Wu, Z. (2013). Predicting quality of service for selection by neighborhood-based collaborative filtering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43(2), 428-439.*

Xiong, K., & Perros, H. (2009). Service performance and analysis in Cloud Computing. In *Services-I, 2009 World Conference on* (pp. 693–700).

XML Web Services. (2016). Retrieved August 30, 2016, from http://www.w3schools.com/xml/xml_services.asp

Yang, J., Lin, W., & Dou, W. (2013). An adaptive service selection method for cross-Cloud service composition. *Concurrency Computation Practice and Experience, 25*(18), 2435–2454.

Yuan, M., & Lin, Y. (2005). Efficient empirical Bayes variable selection and estimation in linear models. *Journal of the American Statistical Association, 100*(472), 1215-1225.

Yuchao, Z., Bo, D., & Fuyang, P. (2012). An adaptive QoS-aware Cloud. In *Cloud Computing Technologies, Applications and Management (ICCCTAM), 2012 International Conference on* (pp. 160–163).

Zeng, W., Zhao, Y., & Zeng, J. (2009). Cloud service and service selection algorithm research. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation* (pp. 1045–1048).

Zhang, M., Ranjan, R., Nepal, S., Menzel, M., & Haller, A. (2012, November). A declarative recommender system for Cloud infrastructure services selection. In *International Conference on Grid Economics and Business Models* (pp. 102-113). Springer Berlin Heidelberg.

Zhao, L., Ren, Y., Li, M., & Sakurai, K. (2012). Flexible service selection with user-specific QoS support in service-oriented architecture. *Journal of Network and Computer Applications, 35*(3), 962–973. http://doi.org/10.1016/j.jnca.2011.03.013

Zheng, Z., Wu, X., Zhang, Y., Lyu, M. R., & Wang, J. (2013). QoS ranking prediction for Cloud services. *IEEE Transactions on Parallel and Distributed Systems, 24*(6), 1213–1222.

Zou, G., Xiang, Y., Gan, Y., Wang, D., & Liu, Z. (2009). An agent-based web service selection and ranking model with QoS. In *Proceedings - 2009 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009* (pp. 37–42).

Appendix

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using Yazeed.Entity;
using Yazeed.Infrastructure;
using Yazeed.Abstract;
using Yazeed.Concrete;
using Yazeed.Dummy;
using System.Collections.Specialized;
using MoreLinq;
using Yazeed.Models;
using System.Diagnostics;
using System.Web.Security;
namespace Yazeed.Controllers
{
  [Authorize]
  public class AdminController : Controller
  {
    private YazeedDBContext db = new YazeedDBContext();
    //     private ICloud ServiceProviderRepository cspRepo = new Cloud
ServiceProviderRepositoryDummy();

    [HttpGet]
    [AllowAnonymous]
    public ActionResult Login()
    {
      return View();
    }

    [HttpPost]
    [AllowAnonymous]
    public ActionResult Login(LoginViewModel model, string ReturnUrl)
    {
      var result = FormsAuthentication.Authenticate(model.Username, model.Password);
      if (result)
      {
        FormsAuthentication.SetAuthCookie(model.Username, false);
        return RedirectToAction("Index");
      }
      else
      {
        return RedirectToAction("Login", new { ReturnUrl });
      }
      return View();
    }

    public RedirectToRouteResult Logout()
```

```
{
    FormsAuthentication.SignOut();
    return RedirectToAction("Login");
}


public ActionResult NewCSP()
{
    ViewData["NewCsp"] = "active";
    return View();
}

public JsonResult ListCloud ServiceProviders()
{
    return Json(new
    {
        id = "",
        error = "",
        fieldErrors = new List<string>(),
        data = new List<string>(),
        aaData = db.Cloud ServiceProviders.ToList()
    }, JsonRequesMb/sehavior.AllowGet);
}

public JsonResult AddCloud ServiceProvider()
{
    int count = db.Cloud ServiceProviders.Count() + 1;
    var Cloud ServiceProvider = new Cloud ServiceProvider
    {
        Name = Request.Form["data[Name]"],
        HdMin = Decimal.Parse(Request.Form["data[HdMin]"]),
        HdMax = Decimal.Parse(Request.Form["data[HdMax]"]),
        RamMin = Decimal.Parse(Request.Form["data[RamMin]"]),
        RamMax = Decimal.Parse(Request.Form["data[RamMax]"]),
        CpuCoreMin = Decimal.Parse(Request.Form["data[CpuCoreMin]"]),
        CpuCoreMax = Decimal.Parse(Request.Form["data[CpuCoreMax]"]),
        BandwidthMin = Decimal.Parse(Request.Form["data[BandwidthMin]"]),
        BandwidthMax = Decimal.Parse(Request.Form["data[BandwidthMax]"]),
        CostPerMonthMin = Decimal.Parse(Request.Form["data[CostPerMonthMin]"]),
        CostPerMonthMax = Decimal.Parse(Request.Form["data[CostPerMonthMax]"]),
        Availability = Decimal.Parse(Request.Form["data[Availability]"]),
        AgeOfCompany = int.Parse(Request.Form["data[CostPerMonthMax]"]),
        DT_RowId = "row_" + count

    };
    db.Cloud ServiceProviders.Add(Cloud ServiceProvider);
    db.SaveChanges();
    //foreach (var key in Request.Form.AllKeys)
    //{
    //    Debug.WriteLine(Request.Form[key]);
    //}

    return Json(new
    {
        id = "",
```

```
            error = "",
            fieldErrors = new List<string>(),
            data = new List<string>(),
            aaData = new List<object>()
        }, JsonRequesMb/sehavior.AllowGet);
    }

    public JsonResult EditCloud ServiceProvider()
    {
        var key = Request.Form["id"];
        var modelKey = db.Cloud ServiceProviders.FirstOrDefault(x => x.DT_RowId == key);
        var model = db.Cloud ServiceProviders.Find(modelKey.ID);
        model.Name = Request.Form["data[Name]"];
        model.HdMin = Decimal.Parse(Request.Form["data[HdMin]"]);
        model.HdMax = Decimal.Parse(Request.Form["data[HdMax]"]);
        model.RamMin = Decimal.Parse(Request.Form["data[RamMin]"]);
        model.RamMax = Decimal.Parse(Request.Form["data[RamMax]"]);
        model.CpuCoreMin = Decimal.Parse(Request.Form["data[CpuCoreMin]"]);
        model.CpuCoreMax = Decimal.Parse(Request.Form["data[CpuCoreMax]"]);
        model.BandwidthMin = Decimal.Parse(Request.Form["data[BandwidthMin]"]);
        model.BandwidthMax = Decimal.Parse(Request.Form["data[BandwidthMax]"]);
        model.CostPerMonthMin =
Decimal.Parse(Request.Form["data[CostPerMonthMin]"]);
        model.CostPerMonthMax =
Decimal.Parse(Request.Form["data[CostPerMonthMax]"]);
        model.AgeOfCompany = int.Parse(Request.Form["data[AgeOfCompany]"]);
        model.Availability = Decimal.Parse(Request.Form["data[Availability]"]);
        db.SaveChanges();
        //foreach (var key in Request.Form.AllKeys)
        //{
        //    Debug.WriteLine(Request.Form[key]);
        //}
        return Json(new
        {
            id = 2,
            error = "",
            fieldErrors = new List<string>(),
            data = new List<string>(),
            aaData = new List<object>{ new  {
                    DT_RowId="row_2",
                    Cloud ServiceProvider="Google",
                    HDMin=50,
                    HDMax=800,
                    RamMin=3.5,
                    RamMax=800,
                    CpuMin=50,
                    CpuMax=800,
                    BandwidthMin=50,
                    BandwidthMax=800,
                    CostMin=50,
                    CostMax=800,
                }}
        }, JsonRequesMb/sehavior.AllowGet);
    }
```

```
        public JsonResult RemoveCloud ServiceProvide()
        {
            var toBeremovedCsps = Request.Form["data[]"].Split(',');
            db.Cloud ServiceProviders.RemoveRange(db.Cloud ServiceProviders.Where(x =>
toBeremovedCsps.Contains(x.DT_RowId)));
            int counter = 1;
            db.SaveChanges();
            db.Cloud ServiceProviders.ForEach(x =>
            {
                x.DT_RowId = "row_" + counter;
                counter++;
            });
            db.SaveChanges();
            return Json(new
            {
                id = 2,
                error = "",
                fieldErrors = new List<string>(),
                data = new List<string>(),
                aaData = new List<object>()
            }, JsonRequesMb/sehavior.AllowGet);
        }

        public ActionResult Index()
        {
            ViewData["dashboard"] = "active";
            return View(db.Cloud ServiceProviders.ToList());
        }

    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Yazeed.Abstract;
using Yazeed.Dummy;
using Yazeed.Entity;
using Yazeed.Infrastructure;
using Yazeed.Models;
using MoreLinq;
namespace Yazeed.Controllers
{
    public class HomeController : Controller
    {
        private YazeedDBContext db = new YazeedDBContext();
        private readonly decimal x1 = 1;
        private readonly int StartOfCloud =2006;
        private const int NUMBER_OF_FEATURES = 5;
        //  private ICloud ServiceProviderRepository cspRepo = new Cloud
ServiceProviderRepositoryDummy();
        public ActionResult Index()
        {
```

```
    var models = db.Cloud ServiceProviders.ToList();
    var preferenceMaximumValues = new PreferenceMaximumValues
    {
        HD = 15,
        Ram = 15,
        CpuCores = 10,
        Bandwidth = 10,
        CostPerMonth = 50
    };
    ViewData["preferenceMaximumValues"] = preferenceMaximumValues;
    ViewData["home"] = "current";
    return View(models);
}

public ActionResult ShowReport(UserInputModel model)
{
    List<Cloud ServiceProvider> csps = db.Cloud ServiceProviders.ToList();
    CalculateHd(csps,(decimal)model.HD);
    CalculateRam(csps,(decimal)model.Ram);
    CalculateCpu(csps,(decimal)model.Cpu);
    CalculateBandwidth(csps,(decimal)model.Bandwidth);
    CalculateCost(csps,(decimal)model.Cost);
    CalculateTotalCredit(csps);

    if (model.AgeOfCompany)
    {
        CalculateAgeOfCompany(csps);
    }
    if (model.Availability)
    {
        CalculateAvailability(csps);
    }
    CalculateReputation(csps, model.AgeOfCompany, model.Availability);
    CalculateFinalResult(csps);
    ViewData["AgeOfCompany"] = model.AgeOfCompany;
    ViewData["Availability"] = model.Availability;
    Session["csps"] = csps;
    return View(csps);

}

[Authorize]
public ActionResult ShowThesis()
{
    ViewData["thesis"] = "current";
    return View();
}

[Authorize]
public ActionResult ShowSlides()
{
    ViewData["presentation"] = "current";
    return View();
}
```

```
public JsonResult UserPreferenceChart()
{
   List<Cloud ServiceProvider> csps = Session["csps"] as List<Cloud ServiceProvider>;
   Dictionary<string, List<object>> result = new Dictionary<string, List<object>>();
   result.Add("x", new List<object>
{"SSD","Ram","Cpu","Bandwidth","Cost","Percentage"});
   foreach (var item in csps)
   {
      result.Add(item.Name, new List<object> {Math.Round(item.HdCredit),
Math.Round(item.RamCredit),Math.Round(item.CpuCredit),Math.Round(item.BandwidthCre
dit),Math.Round(item.CostCredit),Math.Round(item.TotalCredit) });
   }
   return Json(result, JsonRequesMb/sehavior.AllowGet);

}

public JsonResult ReputationChar(bool ageOfCompany,bool availability)
{
   List<Cloud ServiceProvider> csps = Session["csps"] as List<Cloud ServiceProvider>;
   Dictionary<string, List<object>> result = new Dictionary<string, List<object>>();
   if (ageOfCompany && !availability)
   {
      result.Add("x", new List<object> { "Age of Company "});
      foreach (var item in csps)
      {
         result.Add(item.Name, new List<object> {
item.AgeOfCompanyPercentage.ToString("0.00") });
      }
   }
   else if (availability && !ageOfCompany)
   {
      result.Add("x", new List<object> {"Availability"});
      foreach (var item in csps)
      {
         result.Add(item.Name, new List<object>
{item.AvailabilityPercentage.ToString("0.00")});
      }
   }
   else if (ageOfCompany && availability)
   {
      result.Add("x", new List<object> { "Age of Company ", "Availability" });
      foreach (var item in csps)
      {
         result.Add(item.Name, new List<object> {
item.AgeOfCompanyPercentage.ToString("0.00"),
item.AvailabilityPercentage.ToString("0.00") });
      }
   }

   return Json(result, JsonRequesMb/sehavior.AllowGet);
}

public JsonResult FinalResultChart(bool ageOfCompany, bool availability)
{
   List<Cloud ServiceProvider> csps = Session["csps"] as List<Cloud ServiceProvider>;
```

```
            Dictionary<string, List<object>> result = new Dictionary<string, List<object>>();
            string reputation = String.Empty;
            if (ageOfCompany && availability)
            {
                reputation="Reputation";
            }
            else if (ageOfCompany && !availability)
            {
                reputation = "Age of Company";
            }
            else if (!ageOfCompany && availability)
            {
                reputation = "Availability";
            }
            result.Add("x", new List<object> { "Features",reputation,"Total" });
            foreach (var item in csps)
            {
                result.Add(item.Name, new List<object> { item.TotalCredit.ToString("0.00"),
item.ReputationResult.ToString("0.00"), item.FinalResult.ToString("0.00") });
            }
            return Json(result, JsonRequesMb/sehavior.AllowGet);
        }

        private void CalculateHd(List<Cloud ServiceProvider> csps, decimal userInput)
        {
            decimal y1 = csps.Min(x => x.HdMax);
            decimal y2 = csps.Max(x => x.HdMax);
            var Cloud ServiceProvider = csps.OrderByDescending(x => x.HdMax).ToList();
            foreach (var item in Cloud ServiceProvider)
            {
                item.HdCredit = LinearEquationAlgorithm(userInput,y1,y2,item.HdMax);
                item.TotalCredit = item.TotalCredit + item.HdCredit;
            }
        }

        private void CalculateRam(List<Cloud ServiceProvider> csps, decimal userInput)
        {
            decimal y1 = csps.Min(x => x.RamMax);
            decimal y2 = csps.Max(x => x.RamMax);
            var Cloud ServiceProvider = csps.OrderByDescending(x => x.RamMax).ToList();
            foreach (var item in Cloud ServiceProvider)
            {
                item.RamCredit = LinearEquationAlgorithm(userInput, y1, y2, item.RamMax);
                item.TotalCredit = item.TotalCredit + item.RamCredit;
            }
        }

        private void CalculateCpu(List<Cloud ServiceProvider> csps, decimal userInput)
        {
            decimal y1 = csps.Min(x => x.CpuCoreMax);
            decimal y2 = csps.Max(x => x.CpuCoreMax);
            var Cloud ServiceProvider = csps.OrderByDescending(x => x.CpuCoreMax).ToList();
            foreach (var item in Cloud ServiceProvider)
            {
                item.CpuCredit = LinearEquationAlgorithm(userInput, y1, y2, item.CpuCoreMax);
```

```
            item.TotalCredit = item.TotalCredit + item.CpuCredit;
        }
    }

    private void CalculateBandwidth(List<Cloud ServiceProvider> csps, decimal userInput)
    {
        decimal y1 = csps.Min(x => x.BandwidthMax);
        decimal y2 = csps.Max(x => x.BandwidthMax);
        var Cloud ServiceProvider = csps.OrderByDescending(x =>
x.BandwidthMax).ToList();
        foreach (var item in Cloud ServiceProvider)
        {
            item.BandwidthCredit = LinearEquationAlgorithm(userInput, y1, y2,
item.BandwidthMax);
            item.TotalCredit = item.TotalCredit + item.BandwidthCredit;
        }
    }

    private void CalculateCost(List<Cloud ServiceProvider> csps, decimal userInput)
    {
        decimal y1 = csps.Max(x => x.CostPerMonthMax);
        decimal y2 = csps.Min(x => x.CostPerMonthMax);
        var Cloud ServiceProvider = csps.OrderByDescending(x =>
x.CostPerMonthMax).ToList();
        foreach (var item in Cloud ServiceProvider)
        {
            item.CostCredit = LinearEquationAlgorithm(userInput, y1, y2,
item.CostPerMonthMax);
            item.TotalCredit = item.TotalCredit + item.CostCredit;
        }
    }

    private void CalculateAgeOfCompany(List<Cloud ServiceProvider> csps)
    {
        decimal divider= DateTime.Now.Year-StartOfCloud ;
        foreach (var item in csps)
        {
            item.AgeOfCompanyPercentage = ((DateTime.Now.Year - item.AgeOfCompany) /
divider) * 100;
        }
    }

    private void CalculateAvailability(List<Cloud ServiceProvider> csps)
    {
        decimal hoursPerYear = 8765.81M;
        foreach (var item in csps)
        {
            item.AvailabilityPercentage = ((hoursPerYear - item.Availability) / hoursPerYear) *
100;
        }
    }

    private void CalculateReputation(List<Cloud ServiceProvider> csps,bool
ageOfCompany,bool availability)
    {
```

```
            int divider = 1;
            if (ageOfCompany && availability)
            {
                divider = 2;
            }
            foreach (var item in csps)
            {
                item.ReputationResult = (item.AgeOfCompanyPercentage +
item.AvailabilityPercentage) / divider;
            }
        }

        private void CalculateTotalCredit(List<Cloud ServiceProvider> csps)
        {

            foreach (var item in csps)
            {
                item.TotalCredit = item.TotalCredit / NUMBER_OF_FEATURES;
            }
        }
        private void CalculateFinalResult(List<Cloud ServiceProvider> csps)
        {

            foreach (var item in csps)
            {
                item.FinalResult = (item.TotalCredit + item.ReputationResult) / 2;
            }
        }

        private decimal LinearEquationAlgorithm(decimal x2, decimal y1, decimal y2, decimal
featureValue)
        {
            decimal m = (y2 - y1) / (x2 - x1);
            decimal b = y2 - (m * x2);
            decimal result = (featureValue - b) / m;
            return result;
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Yazeed.Models
{
    public class CspSelectionViewModel
    {
        public int ID { get; set; }
        public List<CspFeature> CspFeature { get; set; }
    }

    public class CspFeature
    {
```

```
        public bool IsPrefered { get; set; }
        public string Type { get; set; }
        public decimal Credit { get; set; }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Yazeed.Models
{
    public class LoginViewModel
    {
        public string Username { get; set; }
        public string Password { get; set; }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Yazeed.Models
{
    public class PreferenceMaximumValues
    {
        public decimal HD { get; set; }
        public decimal Ram { get; set; }
        public decimal CpuCores { get; set; }
        public decimal Bandwidth { get; set; }
        public decimal CostPerMonth { get; set; }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Yazeed.Models
{
    public class UserInputModel
    {
        public Level HD { get; set; }
        public Level Ram { get; set; }
        public Level Cpu { get; set; }
```

```
        public Level Bandwidth { get; set; }
        public Level Cost { get; set; }
        public bool AgeOfCompany  { get; set; }
        public bool Availability { get; set; }
    }

    public enum Level
{
         LOW=34,
            MEDIUM=67,
            HIGH=100
}
}
```