Comparison Study of The Most Common Virtual Machine Load Balancing Algorithms in Large-Scale Cloud Environment Using Cloud Simulator

Rowaa Filimban
Technological University Dublin

Follow this and additional works at: https://arrow.tudublin.ie/scschcomdis

Recommended Citation

This Dissertation is brought to you for free and open access by the School of Computing at ARROW@TU Dublin. It has been accepted for inclusion in Dissertations by an authorized administrator of ARROW@TU Dublin. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie, brian.widdis@dit.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License
Comparison Study of The Most Common Virtual Machine Load Balancing Algorithms in Large-Scale Cloud Environment Using Cloud Simulator

Rowaa Filimban

A dissertation submitted in partial fulfilment of the requirements of Dublin Institute of Technology for the degree of M.Sc. in Computing (Advanced Software Development)

January 2016
I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Advanced Software Development), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute’s guidelines for ethics in research.

Signed: ________________________________

Date: 03 January 2017
ABSTRACT

This is era of internet. There is barely any field where internet do not play important role. Nowadays, CloudComputing links to the internet that has rebelled the whole universe. CloudComputing is a rapid enlarging domain in computing industry and research. Three main services offered by the cloud are SaaS, PaaS and IaaS. With the technology advancement of the CloudComputing, there are many new chances pioneer on how applications can be developed and how several services can be provided to the end user throughout Virtualization, on the internet. What's more, there are cloud service providers who offer and provide large-scaled computing infrastructure determined on usage, and offer the infrastructure services in a really elastic way. The establishing of an efficient load balancing algorithm and how to use CloudComputing resources expeditiously for efficient and effective cloud is one of the Cloud service provider’s absolute objective. These days, Load-Balancing is coming out as the main issue in cloud environment. Load due to user requests ought to be balanced through service provider resources. In this dissertation firstly analysis of several Virtual Machine load balancing algorithms is put through. Secondly, This dissertation emphasize comparison with different existing load balancing algorithms by using simulator which is CloudAnalyst. This study focus on only three VM load balancing algorithm because they are the most common in CloudComputing and most of the researchers are interested in. This papers achieve that there is a decrease in data center servicing time as well as decreasing in response time at client part and resources of resource providers are controlled and managed effectively by using Throttled algorithm in a cloud that uses Dynamically Reconfiguration service broker. Therefore, it can be said that the performance of Throttled algorithm is the best among the other two algorithms when the cloud uses Dynamically Reconfiguration service broker.

Key words: Cloud Computing, Load Balancing, Service Broker, Large-Scale Application, Cloud Simulator, Cloud Analyst, VM Load Balancing algorithm.
ACKNOWLEDGEMENTS

I owe a sincere thanks to all my instructors at DIT, Luca Longo, Sara Jane, Pierpaolo Dondio. John Mcauley, John Gilligan, Ciaran Cawley, Brenden Tierney, Andrew Hines, Diana Ferreira, Brian Keegan and Damian Gordan. I learned from each one of you. Thanks to all DIT staff and all those who kept the wheels turning, specially Andrea Curly.

Also, I would like to express my deepest thanks to my supervisor Aneel Rahim for all your opinions and insights. This dissertation would not have been possible without your help and support.

Thanks and appreciation to Saudi Ministry of Education for give me this scholarship and made my dream true.

What's else, I would like to say, my dissertation is dedicated with deepest love and affection to:

To my parents, Abdullah Filimban and Fatimah Simbawah.

To my husband, Rayan Dawaji.

To my daughter, Alma.

To my sisters and brother.

Their love, co-operation, wisdom, and strength have inspired me to be the best I can be and to do the best what I can do.

I could never have done this without their faith, support, and constant encouragement that teaching me to believe in myself and in my dreams.

Sincerely,
Rowaa A. Filimban
# TABLE OF CONTENTS

**ABSTRACT** .......................................................................................................................... II

**ACKNOWLEDGEMENTS** ......................................................................................................... III

**TABLE OF FIGURES** .............................................................................................................. VII

**TABLE OF TABLE** ................................................................................................................ IX

1. **INTRODUCTION** ............................................................................................................... 1
   1.1 BACKGROUND .................................................................................................................. 1
   1.2 RESEARCH DESCRIPTION ............................................................................................... 2
   1.3 RESEARCH OBJECTIVES .............................................................................................. 4
   1.4 RESEARCH METHODOLOGIES ...................................................................................... 4
   1.5 SCOPE AND LIMITATIONS ............................................................................................. 5
   1.6 DOCUMENT OUTLINE .................................................................................................... 6

2. **LITERATURE REVIEW** ....................................................................................................... 7
   2.1 INTRODUCTION ............................................................................................................... 7
   2.2 CLOUD COMPUTING ENVIRONMENT ............................................................................. 7
   2.3 TECHNOLOGICAL DEVELOPMENT IN CLOUD ............................................................ 10
   2.4 LARGE SCALE SOFTWARE SYSTEMS BASED ON CLOUD ........................................... 12
      2.4.1 Example of large scale application based on cloud: Safeer System ....................... 14
         2.4.1.1 Safeer workflow concept: .................................................................................... 14
         2.4.1.2 Sample of Safeer System Requirement Specification ....................................... 15
         2.4.1.3 Safeer System Screenshot: ................................................................................. 17
         2.4.1.4 Safeer architecture and its implications for performance in cloud ...................... 17
   2.5 THE TECHNIQUES OF LARGE SCALE CLOUD COMPUTING ......................................... 18
      2.5.1 Service broker polices ............................................................................................... 19
      2.5.2 VM load balancing algorithms ................................................................................. 21
   2.6 THE MOST COMMON VM LOAD BALANCING ALGORITHMS ..................................... 21
      2.6.1 Round Robin algorithm ............................................................................................ 21
      2.6.2 Throttled algorithm .................................................................................................. 22
      2.6.3 Active Monitoring algorithm ................................................................................... 23
   2.7 ANALYSIS OF VM LOAD BALANCING ALGORITHMS ................................................ 23
      2.7.1 Case 1: using closest data center service broker policy ....................................... 24
      2.7.2 Case 2: using optimize response time service broker policy ................................. 25
      2.7.3 Results Evaluation of Case 1 and Case 2 ................................................................. 27
   2.8 CLOUD SIMULATOR ......................................................................................................... 28
3 DESIGN AND METHODOLOGY .................................................................32

3.1 INTRODUCTION .....................................................................................32
3.2 FOCUS OF THE EXPERIMENT .............................................................32
3.3 THE EXPERIMENT REQUIREMENTS .....................................................33
3.4 DATA SET SOURCES .............................................................................34
   3.4.1 Explanation of collecting the missing date by a Survey ......................36
3.5 THE DATA SET ......................................................................................38
   3.5.1 Datacenters characteristics: ..........................................................38
   3.5.2 User bases’ characteristics .............................................................40
   3.5.3 Specification of Application deployment and internet characteristics ....41
3.6 EXPERIMENT PREPARATION .................................................................42
   3.6.1 Install CloudAnalyst and Eclipse ..................................................42
   3.6.2 Data preparation for Large Scaled Internet Application running on the Cloud ....43
3.7 SET UP THE SIMULATIONS’ PROCESS (METHODOLOGY OF THE EXPERIMENT) ..........44
3.8 METHODOLOGY (OF ALL THESIS) .......................................................49
   3.8.1 Literature review: ........................................................................50
   3.8.2 Data collection: ...............................................................................50
   3.8.3 Data preparation: ...........................................................................52
   3.8.4 Design the simulations: .................................................................54
   3.8.5 The simulations .............................................................................55
   3.8.6 Evaluation the result......................................................................55
3.9 CONCLUSION..........................................................................................55

4 IMPLEMENTATION AND RESULTS..........................................................57

4.1 INTRODUCTION .....................................................................................57
4.2 ENVIRONMENT SETUP ..........................................................................57
4.3 THE CONFIGURATION (DESIGN) OF SAFEER SYSTEM SIMULATION ........58
   4.3.1 Design Scenario 1 and Scenario 2 ..................................................61
   4.3.2 Design Scenario 3 and Scenario 4 ..................................................61
4.4 EXPERIMENT: SIMULATION AND RESULTS .......................................62
   4.4.1 Scenario 1 ......................................................................................63
   4.4.2 Scenario 2 ......................................................................................64
   4.4.3 Scenario 3 ......................................................................................66
   4.4.4 Scenario 4 ......................................................................................67
TABLE OF FIGURES

Figure 2.1: Cloud’s main models ................................................................. 8
Figure 2.2: Cloud computing layers .......................................................... 10
Figure 2.3: Virtual Machine ...................................................................... 12
Figure 2.4: a Slide by Safeer team illustrates important numbers .......... 14
Figure 2.5: Safeer system HomePage ....................................................... 17
Figure 2.6: Service broker policy .............................................................. 19
Figure 2.7: Throttled load balancing algorithm ........................................ 22
Figure 2.8: Active Monitoring load balancing algorithm ....................... 23
Figure 2.9: Overall Response Time of the three VM load balancing algorithm of Case 1 ........................................... 25
Figure 2.10: Overall data center processing time of the three VM load balancing algorithm of Case 1 .......... 25
Figure 2.11: Overall response time of the three VM load balancing algorithm of Case 2 ........... 27
Figure 2.12: Overall data center processing of Case 2 ............................. 27
Figure 2.13: Example of CloudSim results ............................................. 29
Figure 2.14: CloudSim example code in Java ......................................... 29
Figure 2.15: How to run CloudSim Example ........................................... 29
Figure 2.16: CloudAnalyst built on top of CloudSim toolkit .................. 31
Figure 3.1: The survey questions ............................................................... 37
Figure 3.2: Number of respondent per day ............................................ 38
Figure 3.3: Folder structure ..................................................................... 44
Figure 3.4: Run through the command line ............................................ 44
Figure 3.5: Cloud Analyst ....................................................................... 45
Figure 3.6: The Boundaries ..................................................................... 45
Figure 3.7: Data center configuration and adding number data centers ...... 45
Figure 3.8: Data center configuration ....................................................... 46
Figure 3.9: Details of data center configuration ....................................... 46
Figure 3.10: Copy the number of servers in each data center ................ 46
Figure 3.11: The advanced Configuration ............................................... 46
Figure 3.12: The main configuration 1 ..................................................... 47
Figure 3.13: The main configuration 2 ..................................................... 47
Figure 3.13: Save configuration ............................................................... 48
Figure 3.14: Internet characteristics configuration .................................. 48
Figure 3.15: Run simulation ................................................................. 48
Figure 3.16: Simulation results Statistics ........................................ 49
Figure 3.17: The five phases of this research .................................... 50
Figure 3.18: Number of how often users use the system ................. 51
Figure 3.19: Number of users who request certain amount of services 51
Figure 3.20: Number of users who use the system at certain time .... 52
Figure 3.21: Number of users who check on their requests after SACM office hours.... 52
Figure 3.22: Cloud Analyst in Eclipse ................................................ 53
Figure 4.1: The main configuration tab in cloud analyst ................. 58
Figure 4.2: Cost Configuration ............................................................ 59
Figure 4.3: Duration simulation ........................................................... 59
Figure 4.4: example of how to configure some parameters .......... 60
Figure 4.5: The distribution of the regions ........................................ 62
Figure 4.6: Results of scenario 1 ....................................................... 63
Figure 4.7: Result of scenario 2 ............................................................ 65
Figure 4.8: Scenario 3 results .............................................................. 66
Figure 4.9: Scenario 4 results .............................................................. 68
Figure 5.1: Results of the 4 scenarios of the overall response time ..... 71
Figure 5.2: The overall data centre processing time for all 4 scenarios . 72
**TABLE OF TABLES**

Table 2.1: Comparison among private hybrid cloud ........................................... 8
Table 2.2: Result of Round Robin algorithm using closes data center service broker........ 24
Table 2.3: Result Active Monitoring algorithm using closes data center service broker........ 24
Table 2.4: Throttled algorithm using closes data center service broker................... 24
Table 2.5: Overall Response Time using optimize response time............................ 26
Table 2.6: Overall Datacenter Processing Time using optimize response time........ 26
Table 3.1: The sources of each parameters in the dataset ....................................... 35
Table 3.2: Safeer’s users number and where they base ........................................... 38
Table 3.3: Physical hardware of each Data Centre ................................................ 40
Table 3.4: The cost of hosting of the data centres ................................................... 40
Table 3.5 User base characteristics........................................................................... 41
Table 3.6: Latency Matrix values (in milliseconds) .................................................... 42
Table 3.7: Bandwidth Matrix values (in Mbps) ......................................................... 42
Table 4.1: The differences between the design/configuration of Scenario 1 & 2...... 61
Table 4.2: The differences between the design/configuration of Scenario 3 & 4...... 62
Table 4.3: results of scenario 1................................................................................ 64
Table 4.4: Scenario 2 results.................................................................................... 65
Table 4.5: Scenario 3 results.................................................................................... 67
Table 4.6: Scenario 4 results.................................................................................... 68
Table 5.1: Summery of all 4 scenarios ..................................................................... 73
1. INTRODUCTION

1.1 Background

The 21st century is referred to as the age of the internet; the internet has played a great significance probability all domains that are within human reach and control. Among the technological advancement on the web is cloud computing (Pallis, 2010). Cloud computing enables multiple users to access shared hardware and software infrastructure over the internet. Professionals and enterprises view the technology as a significant improvement in data centers that will affect how millions of users acquire information (Pallis, 2010). It is a technology that has helped to distribute large scale amount of data and has helped users to share the data away from their laptops and desktops. Cloud computing harnesses the internet and the wide area network and uses remotely available resources to provide cost effective connectivity to users at real time (Armbrust et al., 2009).

Previously, application deployment used to be one of the major concerns while designing a web application for the Internet. However, the advancement of Cloud Computing makes the deployment more flexible by using powerful infrastructure services. Also, cloud technology helps in resources sharing over the Internet using hardware access to virtual machine virtualization. So, we can say that, this technology provides an abstraction of free virtualization technology, multi-tenancy, and web services and is used to provide a dynamically scalable infrastructure for file storage and application data (Sosinsky, 2011).

All industries have adopted cloud computing although some issues such as server consolidation, virtual machine migration, and load balancing that are yet to be adequately addressed. What's more, Virtualization in cloud technology improves data center power efficiency and make all its virtual machines turning to one physical server. Therefore, within the cloud, an essential part is the allocation of resources to all users. In addition, the service broker policy is in charge of routing the requests of users originating from several geographical locations around the world, to the data centres on cloud. The data centres are also distributed in different physical places. While the
Virtual machine load balancer is taking full responsibility for the load balance in the user requests among all the Virtual Machines (Malewar & Kapgate, 2016). There are three main Virtual Machine load balancing algorithms: Round robin Policy (RR), Throttled Policy (TLB) and Active Monitoring Policy (AMLB). Each one of these algorithms is totally different from the other in the way of handling requests (Arnikar, 2013).

### 1.2 Research Description

Cloud Computing is becoming one of the most popular technologies that taken by industry to provide an efficient and flexible way to retrieve and store data in applications or systems. However, the big issue is to organize the incoming request with minimum efficient resource utilization, a minimum time response and at the same time, resources shall not be under-utilized (Behal & Kumar, 2014). Therefore, The issue of load balancing is perhaps the greatest challenge facing this technology (Rima, Choi, & Lumb, 2009). It’s when resources are not allocated properly, there will be some servers that will be highly loaded while others are idle. This will lead to more energy consumption or more than that; it will not be possible to analyze the resource allocation in the cloud (Mohapatra et al., 2013). In other word, load balancing is used to the server workload among multiple computers or resources to create optimal utilization of resources, minimum response time, and short data processing time to avoid data overload (Rima, Choi, & Lumb, 2009).

There are many studies in terms of resources’ allocation by applying the most common load balancing algorithm, but none of them is about large-scale application using dynamic configuration service broker. The virtual machine load balancing algorithms that are studied in this paper are: Round Robin load balancing algorithm (RR), Active monitoring load balancing algorithm (AMLB) and throttled load balancing algorithm (TLB).

Many researchers have just considered few factors like the availability of nodes, processing capacity, node’s memory and etc. So, Kaur & Luthra, added more factors such as virtual machine computing capacity, virtual machine bandwidth, the image size of a virtual machine and the number of processors in a VirtualMachine
therefore all these features will simply provide the suitable resource to be processed in a cloud for the job (2012).

Bhadani made it clear that cloud technology is totally count on its infrastructure that is point to the cloud’s software and hardware components, such as networking, servers, virtualization software and storage (2011). Moreover, Ray and Sarkar said that this infrastructure of cloud computing includes a software abstraction layer which virtualizes resources which are logically presented to users over programmatic means (2012). Besides, in cloud technology, virtualized resources _such as memory, servers, firewalls, network, switches, virtual machines and load balancers_ are hosted by providers and delivered to its users through the Internet (Mahajan et al., 2013). On one hand, the existing of these providers of cloud technology service who do provide large scaled cloud infrastructure services in really flexible manners. On the other hand, large scaled applications/systems are becoming popular these days, which can get benefit by using cloud services to minimise deployment costs and improve the quality of services (services to end users) (Kherani & Vania, 2014). However, when bringing these two ends together, there are number of different features/factors that will affect on the benefit such as the dynamic nature of the usage patterns of the user base, the user bases’ distribution (geographic), the available Internet infrastructure inside those geographic areas and how well the cloud services can dynamically configure or adapt itself, etc (Agarwal & Jain, 2014). This raises a research question: Which of the VM load-balancing algorithms (RR, TLB, AMLB) is the best to improve the performance of large scale application based on cloud computing?

The reason why this research only focus on those three Virtual Machine Load Balancing Algorithms, because they are the most common and most of the researchers are interested in.

Although, users are not interested in the specific details of an application deployment, They only interested in the application itself and the time taken for the download (the application response time when the user request or command) (Wang, 2012). Nevertheless, details like cloud infrastructure components are very important to make the application works efficiently. Besides, cloud policies like virtual machine load balancing algorithm helps rise the application speed performance (Mohapatra et al., 2013).
1.3 Research Objectives

The main aim of this dissertation is to determine which Virtual Machine load balancing algorithm (RR, AMLB, TLB) is the best. And the reason why this research only focus on those three Virtual Machine Load Balancing Algorithms, because they are the most common and most of the researchers are interested in. In addition, to achieve this objective, the following other objectives need to be achieved as well.

1. A review of the cloud computing and its technology need to be taken.

2. General study about large scale application which is based on Cloud environment.

3. Deep study in to the most common virtual machine (VM) load balancing algorithms and dynamic reconfiguration service broker.

4. Determine the suitable existing cloud simulation tools and techniques, and define an approach for effectively simulating large scaled applications on Cloud environment.

5. Study one example of large application to use it as it is on the cloud.

6. Set up number of simulation of different cases for the same large application on cloud.

1.4 Research Methodologies

The researcher is conducting a secondary research (desk research) which involves the summery and synthesis of existing researches likes the ones were taken under International Journal on Cloud Computing, IEEE, ACM and others approved publishers. Besides, this secondary research will be using the quantitative method which is “a systematic empirical investigation of quantitative properties and phenomena along with their relationships”. Therefore, using the quantitative method in the secondary research will help the investigator to collect her data in order to start an empirical research, which in this case, is a simulation by using CloudAnalyst simulator tool so that the research question can be answered. Furthermore, this investigation is from the general to the specific which is a deductive reasoning research.
That’s to say, In order to figure out the answer of research Question, the objectives in section 1.3. need to be achieved.

Therefore, Objective 1, objective 2, objective 3 and objective 4 have been achieved through a literature review _using the secondary research_ of cloud computing technology, large scale application on cloud, Virtual machine load balancing algorithms and service broker policies. Information concerning the cloud simulation techniques and defining an approach for effectively simulating large scaled Internet applications on Cloud. In addition, as it is obvious, the search start from general (cloud computing) to the specific (VM load balancing algorithms and strives brokers).

Objective 5 has been achieved through gathering data either from studies, papers, survey that were taken under approved publisher like IEEE, and Saudi Ministry of Education (Since the application is for Saudi ministry of Education (MoE)). The application is SafeerSystem [safeer.moe.gov.sa]. And also some data is gathered through quantitative method which is by a survey that distributed to Safeer users around the world.

Objective 6 and objective 7 have been achieved through the detailed design of the simulations, the execution of the simulations and then the result’s evaluation of each simulation. Those simulations to analyse the VMs load balancing algorithms behaviour. However, The empirical evaluation to compare among the algorithms, the researcher is going to evaluate the statistical reports that came out from each simulation. Therefore, based on the empirical evaluation the research question can be verified and answered.

1.5 Scope and Limitations

The experiment was about a simulation using CloudAnalyst which is 100% based on Java. So, Eclipse is going to be used to Run CloudAnalyst framework. In addition, all the simulations were no more than one large application which is Safeer system [safeer.moe.gov.sa], which is the main workflow portal that delivers online services and facilitates the communication between all the Saudi Arabian Cultural Mission’s employees around the world and all Saudi students overseas.
So, the dataset is going to be used is the approximate distribution of the SafeerSystem. This data set is taken from studies and surveys that were taken under the Ministry of Higher Education (MoE) in Saudi Arabia and Saudi Arabian Cultural Mission (SACM). Moreover, the simulations was limited to only three virtual machine load balancing algorithms which are Round Robin algorithm (RR), Active Monitoring load balancing algorithm (AMLB) and Throttled load balancing algorithm (TLB). Moreover, the simulations were done for only one application. Besides, the sample of the respondents to the survey were only 1000 users. However, because this application is only for easier and faster communication about specific reasons like request or question, Safeer users are almost alike in the way of using the system.

1.6 Document Outline

Presented in this dissertation is a literature review related to Cloud Technology, a design and methodology description of the experiment performed, an explanation of the experiment, an analysis of the experimental results, and the conclusions reached as well as recommendations for directions of future research.

In the literature review several areas of research are targeted that reflect the key areas of concern for this study. Topics such as Cloud environment, technology and principles, Large scale cloud web application, various virtual machine load balancing algorithms, Service broker policy specially dynamic reconfiguration, and finally cloud simulation tools.

Following the literature review, the design discusses the background, approach and methodology and covers topics such as the CloudAnalyst Domain Model and Cloud Environment Main Components. Further, the methodology and simulation implementation are discussed.

Details of how the experiment was conducted including metrics, details of problems encountered, and motivations are discussed. Finally a statistical analysis is performed and conclusions and future direction are discussed.
LITERATURE REVIEW

1.7 Introduction

In this chapter several areas of research are targeted that reflect the key areas of concern for this study. Cloud computing environment and its technology as they relate to cloud computing policies and algorithms is discussed and lays the groundwork for the justification of this work. Literature on the types of virtual machine load balancing algorithms and service brokers have been examined as is discussed to provide details about how they work so that will help in comparing among the most common virtual machine (VM) load balancing algorithms. Several tools for simulating cloud computing environment for large scale application under specific virtual machine load balancing algorithms are discussed, focusing on their feature; their limitations and advantages are considered, followed by a discussion on large scale application in cloud.

1.8 Cloud computing environment

The 21st century is referred to as the age of the internet; the internet has played a great significance probability all domains that are within human reach and control. Among the technological advancement on the web is cloud computing (Pallis, 2010). Cloud computing enables multiple users to access shared hardware and software infrastructure over the internet. Professionals and enterprises view the technology as a significant improvement in data centers that will affect how millions of users acquire information (Pallis, 2010). There are number of cloud’s types on the fundamental of deployment, here are the important ones (Nair et al., 2010):

1. Public Cloud: IT services that provided by cloud can be used by anybody using internet.

2. Private Cloud: IT services are only provided to authorized users like an organisation or enterprise.
3. **Hybrid Clouds**: It is a combination of private and public cloud provided by an organization.

![Cloud's main models](image)

**Figure 2.1**: Cloud’s main models

The following table will illustrate the pros and cons for each model of cloud computing.

<table>
<thead>
<tr>
<th>Capabilities</th>
<th>Private</th>
<th>Public</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Control</td>
<td>IT enterprise</td>
<td>Service provider</td>
<td>Controlled by both the enterprise and the service provider</td>
</tr>
<tr>
<td>Total Cost of Ownership</td>
<td>High cost</td>
<td>Low cost</td>
<td>Moderate cost as the cost follows a pay-as-you-go model</td>
</tr>
<tr>
<td>Data Security</td>
<td>High</td>
<td>Low</td>
<td>Moderate</td>
</tr>
<tr>
<td>Service Levels</td>
<td>IT specific</td>
<td>Provider specific</td>
<td>Aggregate</td>
</tr>
<tr>
<td>Scalability</td>
<td>Limited</td>
<td>Unlimited</td>
<td>Base and burstable</td>
</tr>
</tbody>
</table>

**Table 2.1**: Comparison among private, public and hybrid cloud

According to Buyya, Broberg, & Gościński, (2011), there are six basic principles of cloud computing. These principles allow cloud computers to freely migrate, grow, and shrink.

- **Federation**
  The principle of federation state that all cloud computing providers have a finite capacity regardless of their size. To surpass this obstacle, the providers should collaborate with one another and share their resources. All the federal cloud
service providers should allow virtual application of its resources to be deployed to federated sites. Furthermore, they should be location free and can be accessed through any site (Buyya, Broberg, & Gościński, 2011).

• Independence
Just like any other utility such as telephones, where clients do not necessarily have the knowledge of internal processes of the service provider, cloud providers also require independence. Users should be able to access the cloud computing services without relying on particular tools from the service provider (Buyya, Broberg, & Gościński, 2011).

• Isolation
Cloud service providers host services that host applications from different users. Users who seek cloud services require warranties from the service provider that third information is held differently from other information hosted by the server. They need assurance that other users who are not permitted to access the information cannot gain access (Buyya, Broberg, & Gościński, 2011).

• Elasticity
Cloud computing can provide information when demanded. These capabilities are enacted automatically to meet varying needs. The service providers should be able to handle the scale demand required by users. At the same time, they should also allow the users to downscale. The scalability helps the user to maneuver the cost and effectiveness of the service.

Other principles are the trust and business orientation (Buyya, Broberg, & Gościński, 2011).

To sum up, Electrical Engineering and Computer Sciences University of California at Berkeley (2009) defines cloud computing as “applications delivery as service through the Internet and system software and hardware in the datacenters that provides those services”.

9
1.9 Technological Development in Cloud

Cloud computing requires the providers and users to adopt and implement essential protocols, interfaces and architectural components that allow the flow of information around a network. Without these components, it is virtually impossible to achieve cloud interoperability (Toosi, Calheiros, & Buyya, 2014). Cloud computing architecture refers to the designs, components, software applications and subcomponents that use Internet-accessible services to enable cloud computing. These components include user interface devices such as laptops and desktops, and back-end platforms such as data storages and servers, a network connection such as an intranet or inter-cloud, and a cloud delivery system. Cloud computing can be seen as a set of services, which can be exhibited as a layered cloud computing architecture. Those services provided by cloud computing usually include:

- Software as a Service (SaaS) refers to IT services which allows and enables users to run applications remotely from the cloud.
- Platform as a service (PaaS) provides a space for consumers to create services or applications.
- Infrastructure as a Service (IaaS) points to computing resources as a service that includes virtualized computers with ensured processing power and reserved bandwidth for Internet access and storage (Bishwkarma & Vyas, 2016).

![Figure 2.2: Cloud computing layers 1](image-url)
The technological development in Cloud is separated into Cloud Service Providers and Software System Providers (Bishwkarma & Vyas, 2016).

- **Cloud Service Providers** provide their services as computing infrastructure environment for application development (large scale infrastructure) at cost based on usage pattern.

- **Software System Providers** develop and deploy applications such as e-commerce and social networking.

Furthermore, Cloud computing infrastructure is the set of networking gear, server hardware, software and storage resources. All these resources are required to build applications that can be accessed by the cloud. In a cloud computing infrastructure, applications can be accessed remotely by using networking services such as telecom services, wide area networks, and the Internet. All of this are typically built by cloud service providers (Wickremasinghe & Buyya, 2014).

The components of cloud computing infrastructure are usually categorised into three resources which need to work together to provide a cloud service. The three categories are as follow (SDXcentral, 2011):

- **Computing:** which provides the computing power for the cloud service and is commonly provided by number of servers powered by server chips. Those servers can be attached together with virtualization software to split up the computing power to different services or clients.

- **Networking:** Switches and routers are used in order to move data among the storage systems, the computing resources, and the outside world.

- **Storage:** The cloud service usually demands large amounts of storage resources. The storage system has its own storage software and networking gear to manage high-performance connectivity with the service.

A cloud computing infrastructure probably contains an extremely expensive combinations of storage hardware, server, and networking but the software is the key to making it work all together. This software is referred as virtualization software, that is because it is eligible of taking all of the dynamically creating new networks and the
hardware pieces that tie together the virtual resources thus that they can be sold for different customers as services.

So, Chieu, Mohindra, Karve and Segal (2009) reported that the computing resources' physical characteristics are hidden from their users by Virtualization. It includes altering one single physical resource (such as an operating system, a storage device, an application, or server) to function as multiple number of virtual resources; this can also include making multiple number of physical resources (such as servers or storage devices) show as one single virtual resource.

Figure 2.3: Virtual Machine

Cloud computing, as it is mentioned in the previous sections, is designed to provide service instead of a product. Dash, Mahapatra and Chakraborty said that one of the main design issues of cloud computing is Transparency, in which services such as software, computation, storage, and data access are provided to users without the knowing of the user geographical location (2013).

Given the progress of services of cloud, large scale software systems, such as e-commerce applications, social networking sites and government to citizens applications or systems, are gaining popularity. These systems extremely utilize cloud services to improve service quality and minimize costs to end users (Radi, 2013).

1.10 Large Scale Software Systems Based on Cloud

The deployment and development of large scale systems has become increasingly common, with the rapid advancement in the area of Cloud computing. A
Large-scale application is an application which is able to handle a huge number (millions) of users/request per second. It is distributed or run on servers geographically closer to the user (Dave & Maheta, 2014). Moreover, not only E-commerce sites and Social network sites are example of large scale software, but also, E-government as Government to citizens (G2C) software in most of the countries, is a good example of large scale software based on cloud (private cloud for more security for the case of G2C). Various factors affect the quality and cost of the cloud of large scale software. Here some examples of these factors: the distribution of the user bases (geographic), the dynamic nature of the usage patterns of the user base, the availability of the infrastructure of the Internet within those geographic areas, and how well cloud services can be dynamically reconfigured or adapted. However, deployment and development of large scale applications on Cloud is cheaper and easier. As cloud applications can be deployed and developed in various geographical locations as well as it affects the user located far away from the cloud datacenter. Those applications can be accessed by users from all the world and the popularity of them also varies with geographical location and so that will vary the user experience of using that application (Sankla, 2015).

The main components of any large scale application: Virtual Machines, User Bases, Cloudlets, Data Centres, Service Brokers and Virtual Machine Load Balancing Algorithms (Radi, 2013).

Any large application must be hosted on set of servers (Virtual Machines). Those groups of servers are capsulated in data centers which are located in one region or more. Since there are lots of users from everywhere using this large application, each group of users is considered as User base. Normally, these user bases and data centres are distributed to six regions correspond to six continents in the world. So, each request (Cloudlet) from a user includes the ID of the application, name of the user base for routing back the responses, input and output files and size of request execution commands. In addition, the one who is responsible for routing the requests of users originating from different location around the globe, to the data centres on cloud is service broker policy. And the one who is responsible for the load balance in the user requests among the virtual machines is Virtual machine load balancing algorithm.
1.10.1 Example of large scale application based on cloud: Safeer System

Projects as E-government have been considered as one of the main priority areas in most of the countries which Kingdom of Saudi Arabia is one of them. (Alshehri, Drew & Alfarraj, 2012). Therefore, Saudi ministry of education (MoE) launched G2C service which is Safeer system. Safeer is the main workflow portal that delivers online services and facilitates the communication between all the (SACM) Saudi Arabian Cultural Mission employees around the world and all Saudi students who study abroad. That’s to say, the system is a workflow which handles the requests of SACM’s students (Saudi ministry of higher education, 2014, p.11).

Safeer users are more than 120,000 users as Saudi students who study overseas and the employees of SACM in those following countries: Ireland, USA, UK, Canada, France, Italy, Spain, Germany, Australia, Poland, The Netherlands, New Zealand, Hungary, Austria, Czech Republic and 10 other countries (Saudi Ministry of Higher Education, 2015).

The following figure is one slide by Safeer team presents important numbers.

![Figure 2.4: a Slide by Safeer team illustrates important numbers](image)

2.4.1.1 Safeer workflow concept:

The concept of Safeer workflow described by (AL-Zuabi & AL-Shaikhli) as the following:
Safeer collects requests from the students overseas, passes it to SACM employees for review, might forward the data to MoE that informs Safeer users of decisions, and records output in one central database. However, all transactions are reported and recorded through an electronic medium and saved at the central database in Riyadh (2012).

2.4.1.2 Sample of Safeer System Requirement Specification

The following is a study done by the researcher. The researcher analysed Safeer system [https://safeer.moe.gov.sa/Sites/Student/Pages/default.aspx] in order to understand the large-scale application more.

2.4.1.2.1 Functional Requirement:

- Registration: Each Scholarship student has an electronic- file number. So, they can register by their electronic file number and their national ID. they will have a password in order to login to the system again.
- Login: After the registration, Students can login by their national ID and password.
- Edit information: Students can Edit their information, For example:
  - add more independent.
  - if they renew their study visa, they have to update their profile.
    otherwise the system will keep alarming the student.
- Request: There are many Requests student can make, like:
  - request tickets for the student and student's family
  - request official document like:
    - Approved financial grantee, So student can submit it to the education institution that he/she go.
- Study services: here some examples:
  - students can ask for changing the current major, so they can study something else after the embassy will approved.
  - students can ask for hold the scholarship for personal reason, they have to attach all the documents so the embassy will look at them and decide to approve or not.
• Financial services: here some examples:
  o after the students renew their visa, they can ask for their money back from the embassy.
  o when the students do a test which is required fee, the embassy will give them their money back.

2.4.1.2.2 Non-Functional Requirement:

• Technical Requirements
  Safeer system is a website that's fully compatible with the following browsers using Microsoft Windows operating system (XP, Vista or 7) and Macintosh operating system (OS X 10.7 or later, OS X Mountain Lion).

• Browser Software:
  ▪ Microsoft Internet Explorer 8 or above.
  ▪ Google Chrome.
  ▪ Apple Safari 6 or above.

• Usability Requirements
  Functions in safeer system are easy to accomplish quickly and with no user errors. besides, The content of the pages are clearly written, understood, and up-to-date. And the most important thing is the interface is easy to learn and navigate; buttons, headings, and help/error messages are simple to understand.

• Performance Requirements
  Safeer's pages response time (10 to 25 seconds) when users click the links (AL-Zuabi and AL-Shaikhli, 2012).

• Safeer system availability:
  Safeer system is run 7 days a week, 24 hours a day.

• Safeer system accuracy:
  The content and information will provide by the system is trustable and accuracy.

• Supportability Requirement:
  The system supports both English and Arabic languages.

• Security Requirements
  Security should be placed to prevent unauthorized users from login to the system and to protect students information.
2.4.1.3 Safeer System Screenshot:

![Safeer System Screenshot](image)

**Figure 2.5: Safeer system HomePage**

See more screenshots about Safeer system services _for Saudi students_ in the appendix.

2.4.1.4 Safeer architecture and its implications for performance in cloud

Safeer users are more than 120,000 users as Saudi students who study overseas and the employees of SACM in those following countries: Ireland, USA, UK, Canada, France, Italy, Spain, Germany, Australia, Poland, The Netherlands, New Zealand, Hungary, Austria, Czech Republic and 10 other countries (Saudi Ministry of Higher Education, 2015).

The concept of Safeer workflow described by (AL-Zuabi& AL-Shaikhli) as the following:

Safeer collects requests from the students, passes it to SACM employees for review, might forward the data to MOHE that informs Safeer users of decisions, and records output in one central database. However, all transactions are reported and recorded through an electronic medium and saved at the central database in Riyadh (2012). The consequences of having only one datacenter that used by a huge number of users is the heavy traffic that leads to the low speed of the system performance. In 2014, a survey has been conducted to evaluate the usability of SAFEER e-services. And a large number of Safeer users answered the survey. The speed of the system is one of the
main key findings of the survey that made all participants very disappointed (Albalawi, 2014, p.4-6).

It was explained by specific decision makers that the Network and Internet infrastructure in clouds in Saudi Arabia is under development. As a result, Safeer system was deployed on simple private cloud which MOHE owned it. Safeer’s infrastructure has only one datacenter which is deployed in Asia, specifically in the capital of Saudi Arabia, Al-Riyadh city. Furthermore, the system is deployed in only 50 virtual machines (Alfarraj, Drew & AlGhamdi, 2011, p.84). Moreover, more than 120,000 users worldwide use the system (Saudi Ministry of Higher Education, 2015). With this number of users and this limited infrastructure, The system can’t handle all the requests that came from users without delay. That’s why Safeer users have been complained about the low speed of the system. Albalawi said that the cloud’s system policies needs to be revised in regard to the system delay. Besides, MOHE needs to consider the number of Safeer’s servers, virtual machines and datacenter and where they should be located in order to speed up the performance and to ensure quick access to Safeer services, especially during peak hours. (2014, p.15). Alzomily mentioned that: Safeer users can expect a response time around10 seconds, but there are several periods of time during the day, the actual response time can be expected to rise higher. Therefore, a decision needs to be addressed with strict and careful access policy (2013). The following part describes how safer system can get benefit from all cloud features and how Safeer’s cloud’s resources and polices can be modified in order to solve the problem.

1.11 The Techniques of Large Scale Cloud Computing

Cloud computing as its mentioned in the previous sections enables multiple users to access shared hardware and software infrastructure over the internet. Professionals and enterprises view the technology as a significant improvement in data centers that will affect how millions of users acquire information (Pallis, 2010). One great challenge facing large cloud computing is the allocation and migration of virtual machines that can be reconfigured and the integration of hosting machines (Armbrust et al., 2009). It is a technology that has helped to distribute large scale amount of data and has helped users to share the data away from their laptops and desktops. Cloud
computing harnesses the internet and the wide area network and uses remotely available resources to provide cost effective connectivity to users at real time (Armbrust et al., 2009). All industries have adopted cloud computing although some issues such as server consolidation, virtual machine migration, and load balancing that are yet to be adequately addressed. The issue of load balancing is perhaps the greatest challenge facing the technology (Rima, Choi, & Lumb, 2009). Load balancing is used to the server workload among multiple computers or resources to create optimal utilization of resources, minimum response time, and short data processing time to avoid data overload (Rima, Choi, & Lumb, 2009).

1.11.1 Service broker policies

The process of serving a client begins when the client requests a particular resource either for development or gain access to the data. The information received from the user is serviced through multiple steps including an intermediary known as a service broker (Young-Rok Shin & Eui-Nam huh). The service broker acts as a link between the cloud provider and the client. A service broker determines the data route between the client and the service provider. In other words, A service broker policy decides which datacenter should provide the service to the requests which coming from each userbase. Therefore, service broker policy controls the traffic routing between both DataCenters and UserBases.

![Figure 2.6: Service broker policy](image)
To efficiently perform the intermediary role, the service provider is guided by three policies;

• Closest data center policy:

  The closest data policy is also known as the Service Proximity based routing; it makes use of the proximity between the user and the data center. The service provider provides the shortest route to the data in consideration of the transmission latency (Joshi, Yesha, Finin, & Yesha, 2013. A proximity list is maintained by the broker who also sets the occurrence of data centers. In case multiple data centers with equal latency are available, then the one the centers is picked randomly regardless of the workload. This policy is beneficial to users as the information can be received within a short period as long as the data center identified can successfully satisfy a request (Manasrah, Smadi, & ALmomani, 2016).

• Optimal response time policy:

  This service broker uses a network latency parameter to identify the closest data center. This routing policy monitors the performance of the available data centers and directs incoming traffic towards the center with the lowest response time (Manasrah, Smadi, & ALmomani, 2016). If the nearest data center transmits the lowest response time then it is selected, if another data center responds faster than the closest data center, then it is selected as the suitable data center (Rani, Chauhan, & Chauhan, 2015).

• Dynamically reconfigurable routing with load balancing:

  This policy uses the current execution load to determine the most suitable route that a process with be assigned. Scaling is done to by determining the current processing time and the best processing time. The policy can decrease or increase the virtual machines (VM) in the data centers. The route that is dynamically identified as the most suitable route for a process is then assigned the task (Reimer, Abraham, & Tan, 2013).
1.11.2 VM load balancing algorithms

In large cloud computing, load balancing offers an efficient solution to the issues that arise and reside in the computing environment usage and set up. Load balancing takes into account two primary tasks, resource provisioning and scheduling the distributed environment (Katyal & Mishra, 2014). These functions aid in the ensuring that resources are available on demand, fully utilized, saves the energy used to access information and reduce the cost of obtaining the information (Katyal & Mishra, 2013).

All in all, in the scenario of Cloud Computing, user submits the task to be executed/performed, So Cloud Coordinator divides the task into cloudlets and passes it to the servers in the Data Centers. Then, these servers as Virtual Machine (VM), performs/executes user requests as cloudlets which present in the queue as they reach those virtual machines. This process of cloudlet’s scheduling is known as VM load balancing algorithm and Service brokers policy (Kapur, 2015).

1.12 The Most Common VM load Balancing Algorithms

1.12.1 Round Robin algorithm

A round robin algorithm is a traditional approach that distributes tasks evenly to all slave processors in a circular manner. Each processor is assigned a particular task, and the task is maintained locally independent from the other processors (Ko, Kim, Kim, Thota, & Jha, 2010) The workload is evenly distributed but the processors work at different processing speed and complete the tasks at various times. At some point in the processing Robin, some processors might be idle while others are overloaded with processes. The round robin scheduling is priority oriented as the operations are handled in a circular manner. It is mostly used in Http servers (Ko, Kim, Kim, Thota, & Jha, 2010.).

Round Robin algorithm works as the following (Bishwkarma & Vyas, 2016):

1. Creates same size of Cloudlets.
2. Cloudlet Coordinator devices the assigned Cloud task into same size of cloudlets.
3. Create Broker and User assigned the task to the cloudlet coordinator.
4. Coordinator sends cloudlets to VM managers and VM managers send the list of the needed resources.
5. Request for the exception of the cloudlet is set to the VM from the host
6. Cloudlet scheduling is done in VM according to First come First serve scheduling policy.
7. Sends the edited job as cloudlets in a wrapped file to VM managers.
8. VM further passes the executed cloudlet as wrapped file to the cloudlet coordinator.
9. Cloudlet coordinator combines all excited cloudlets in wrapped file from combine to form the whole task.
10. Cloudlet coordinator sends the excited task in authenticated file format to the user client.
11. Print the result.

1.12.2 Throttled algorithm

The throttled algorithm allocates a virtual machine to each request the client sends to the load balancer. The throttled balancer takes the responsibility of indexing each task and machines whether the processor is busy or idle (Wickremasinghe, Calheiros, & Buyya, 2010). Initially, all virtual machines are indexed as idle, when a task is received, it requests the load balancer to identify and issue the task to an idle virtual machine. The data center acknowledges the allocation and updates the location of the task according to the virtual machine. If the load balancer does not find an appropriate virtual machine, it returns the request to the data center which queues the request until a suitable machine can be identified (Mohialdeen, 2013).

![Figure 2.7: Throttled load balancing algorithm](image-url)
1.12.3 Active Monitoring algorithm

Active monitoring load balancing maintains the number of connections allocated to each server and assigns tasks to the server with the least amount of tasks. The Active monitoring load balancer maintains the table of virtual machines and sends information to the data center which allocates the task according to the information received (Wickremasinghe, Calheiros, & Buyya, 2010). This algorithm maintains an equal amount of processes among the available virtual machines. If the load balancer identifies more than a single idle virtual machine, then the data center will assign the task to the one identified first (Wickremasinghe, Calheiros, & Buyya, 2010). The data center also sends the information concerning the virtual machine to the load balancer about new allocations so as to update the number of allocations on the virtual machine. When a process is complete, the data center sends the information to the load balancer to reduce the allocations on the virtual machines and wait for the next task (Wickremasinghe, Calheiros, & Buyya, 2010).

![Diagram of Active Monitoring load balancing algorithm](image)

Figure 2.8: Active Monitoring load balancing algorithm

1.13 Analysis of VM Load Balancing Algorithms

This section discusses and analyzes the various load balancing algorithms that were mentioned in section 3.6. There are many papers and researches about Virtual Machine load balancing algorithms. However, as it was mentioned in Chapter 1, none of these
papers about large-scale cloud with dynamic reconfiguration route with load service broker policy. Therefore, in this section, the researcher will discuss the differences among the three VM load balancing algorithms whereas the cloud firstly using closest data center service broker policy. And then whereas the cloud using optimise response time service broker policy.

1.13.1 Case 1: using closest data center service broker policy

There was a study by Singh, Sharma and Kumar (2016). They use CloudAnalyst as cloud simulator tool. They simulated the cloud environment by taking six datacenters, having six virtual machines in each datacenter, and six user bases. They did three simulation, each simulation for one of the VM load balancing algorithm. Moreover, 60 hours is the duration of each performed simulation. Besides, they considered 1000 as average number of users at peak hour, and 100 users as at off-peak hour. After the simulations, The results obtained considering different VM load balancing algorithms as it shown in Table 2.2, Table 2.3 and Table 2.4.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Average (ms)</th>
<th>Min. (ms)</th>
<th>Max. (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Response Time</td>
<td>64.04</td>
<td>34.5</td>
<td>9818.03</td>
</tr>
<tr>
<td>Data Center Processing Time</td>
<td>13.7</td>
<td>0.02</td>
<td>9768.77</td>
</tr>
</tbody>
</table>

Table 2.2: Result of Round Robin algorithm using closest data center service broker

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Average (ms)</th>
<th>Min. (ms)</th>
<th>Max. (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Response Time</td>
<td>64.38</td>
<td>34.5</td>
<td>9818.03</td>
</tr>
<tr>
<td>Data Center Processing Time</td>
<td>14.04</td>
<td>0.02</td>
<td>9768.77</td>
</tr>
</tbody>
</table>

Table 2.3: Result Active Monitoring algorithm using closest data center service broker

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Average (ms)</th>
<th>Min. (ms)</th>
<th>Max. (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Response Time</td>
<td>57.68</td>
<td>34.5</td>
<td>87.01</td>
</tr>
<tr>
<td>Data Center Processing Time</td>
<td>7.24</td>
<td>0.02</td>
<td>25.05</td>
</tr>
</tbody>
</table>

Table 2.4: Throttled algorithm using closest data center service broker
Figure 2.9: Overall Response Time of the three VM load balancing algorithm of Case1

![Graph of Overall Response Time](image)

Figure 2.10: Overall data center processing time of the three VM load balancing algorithm of Case1

![Graph of Data Center Processing Time](image)

1.13.2 Case 2: using optimize response time service broker policy

There was a study by Jena and Ahmed (2013). As the Case 1, those two researchers used CloudAnalyst as cloud simulator tool to analyse the performance of the three VM load balancing algorithms. Each simulation for each VM load balancing algorithm was carried out for 60 hours period by taking three data centers which were having 75, 50 and 25 numbers of Virtual Machines respectively and different number of userbases.
Table 2.5: Overall Response Time using optimize response time

<table>
<thead>
<tr>
<th>Setup Description</th>
<th>Overall Response Time (in ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Round Robin</td>
</tr>
<tr>
<td>Exp1 6 user bases</td>
<td>195.91</td>
</tr>
<tr>
<td>Exp2 12 user bases</td>
<td>200.99</td>
</tr>
<tr>
<td>Exp3 18 user bases</td>
<td>201.57</td>
</tr>
<tr>
<td>Exp4 24 user bases</td>
<td>203.89</td>
</tr>
<tr>
<td>Exp5 30 user bases</td>
<td>204.18</td>
</tr>
<tr>
<td>Exp6 36 user bases</td>
<td>204.54</td>
</tr>
<tr>
<td>Exp7 42 user bases</td>
<td>204.79</td>
</tr>
<tr>
<td>Exp8 48 user bases</td>
<td>201.96</td>
</tr>
</tbody>
</table>

Table 2.6: Overall Datacenter Processing Time using optimize response time

<table>
<thead>
<tr>
<th>Setup Description</th>
<th>Overall Datacenter Processing Time (in ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Round Robin</td>
</tr>
<tr>
<td>Exp1 6 user bases</td>
<td>14.38</td>
</tr>
<tr>
<td>Exp2 12 user bases</td>
<td>14.39</td>
</tr>
<tr>
<td>Exp3 18 user bases</td>
<td>14.51</td>
</tr>
<tr>
<td>Exp4 24 user bases</td>
<td>14.40</td>
</tr>
<tr>
<td>Exp5 30 user bases</td>
<td>14.57</td>
</tr>
<tr>
<td>Exp6 36 user bases</td>
<td>14.35</td>
</tr>
<tr>
<td>Exp7 42 user bases</td>
<td>14.34</td>
</tr>
<tr>
<td>Exp8 48 user bases</td>
<td>11.42</td>
</tr>
</tbody>
</table>
1.13.3 Results Evaluation of Case 1 and Case 2

Contrasting the obtaining results from the different service broker policies and VM load balancing algorithms, the response time and datacenter processing time of Throttled load balancing algorithm is good when using Closest Datacenter service Broker policy Singh, Sharma and Kumar (2016). Although the data processing time and the response time of Active Monitoring and Throttled algorithms are nearly the same in case 2 (using optimize response time service broker), Jena and Ahmed (2013) concluded that AMLB is an efficient effective one than the other two VM load balancing algorithms when the application is deployed on a cloud that uses optimize response time service broker policy.
1.14 Cloud Simulator

1.14.1 CloudSim

A CloudSim is an extensible framework that allows for simulation, and modeling and experimentation of new infrastructures that can be used for cloud computing. By CloudSim allows researchers and developers to test their ideas within a controlled environment where they can evaluate and analyze their findings, and fine tune the service (Calheiros, Ranjan, Beloglazov, De Rose, & Buyya, 2010). Due to the problems experienced by the existing simulators, developers were unable to diagnose the extent of their services fully. Other simulators were not applicable to the cloud computing technology are proved redundant for use. It is hard to determine and evaluate the performance of cloud application, policies, and models under diverse and varying systems. To overcome these obstacles, developers created the CloudSim to aid in simulation of new cloud services.

Goyal, Singh, & Agrawal, (2012), state that CloudSim is not a framework because it does not provide a suitable environment to execute a complete scenario and successfully achieve the targeted results. They state that the users of CloudSim have to develop the scenario they wish to evaluate and define the output they expect to receive (Kumar & Sahoo, 2014). It supports the systems that allow the remodeling of certain cloud features such as the virtual machines, and data centers. It implements the application of generic applications that can be changed with limited effort. Researchers and developers have the ability to focus on specific design issues regardless of the low-level infrastructure (Wickremasinghe, Calheiros, & Buyya, 2010). That is to say, CloudSim provides developers with basic cloud computing entities.

2.8.1.1 Example of using CloudSim

The following example presents how to create one host and one cloudlet DataCenter. To see more examples of CloudSim, please see the appendix.
Figure 2.13: Example of CloudSim results

Figure 2.14: CloudSim example code in Java

To run the CloudSim examples you need to do the following steps.

In Windows:
1. cd <PATH TO CLOUDSIM PACKAGE>/jars
2. java -classpath cloudsim-<VERSION>-jar cloudsim-examples-<VERSION>-jar org.cloudsims.cloudsims.examples.CloudSimExample<EXAMPLE NUMBER>

In Unix/Linux:
1. cd <PATH TO CLOUDSIM PACKAGE>/jars
2. java -classpath cloudsim-<VERSION>-jar cloudsim-examples-<VERSION>-jar org.cloudsims.cloudsims.examples.CloudSimExample<EXAMPLE NUMBER>

Where you need to replace:

(PATH TO CLOUDSIM PACKAGE) - by the path to a directory where you have unpacked the CloudSim package
.VERSION - by the version of the downloaded CloudSim package
.EXAMPLE NUMBER - by the of number of the example you want to run

Figure 2.15: How to run CloudSim Example
1.14.2 GridSim

It is very hard and near impossible to perform an evaluation in controlled and repeatable environment such as a grid environment. To overcome this issue, researchers have developed a GridSim, a tool that enable developers to model and simulate various resources (Buyya & Murshed, 2002). The GridSim supports the different resources such as workstations, multiprocessors, clusters of information and distributed memory machines. The GridSim is used for simulation of various classes of computing systems mainly parallel systems. In a cluster, the resources administered by a single domain, in a grid system, the resources are distributed across several administrative domains (Buyya & Murshed, 2002).

The GridSim toolkit also provides the modeling platform for simulating network resources, particularly connectivity. It determines the varying capabilities, domain, and configuration of a cloud-based connection.

During simulation, several multi-threaded entities are created by the GridSim. Each of the entities created is dependent on the other and runs parallel. As dictated by SimJava, the behavior of an entity needs to be simulated within its body. Since the entities are dependent, their simulation environment needs to be abstract. When simulating GridSim entities, they contain users, resources, information service providers, brokers and an active network administrator.

1.14.3 Cloud Analyst

A cloud analyst is a toolkit with the features of an original framework that extend the capabilities of CloudSim. The CloudAnalyst is responsible for separating simulations from programming technicalities. The separation occurs to allow the modeler to focus on the processes and task being simulated (Rani, Chauhan, & Chauhan, 2015).

The CloudAnalyst provides a user-friendly interface by hiding the complex programming. With the complexities out of the simulation, the modeler conducts a series of experiments in varying environments in a quick and successive manner (Zhang, Cheng, & Boutaba, 2010). It is developed on a CloudSim toolkit by introducing concepts that emulate Internet application behaviour.
A CloudAnalyst is popular because it has an easy to navigate Graphical User Interface, can simulate tasks that require sophisticated reconfigurability and flexibility, ease of application extensions and can continuously repeat experiments (Rani, Chauhan, & Chauhan, 2015).

Figure 2.16: CloudAnalyst built on the top of CloudSim toolkit

A CloudAnalyst has several components, these are;

- The GUI package; which is responsible for user interface. The GUI is the front end of the simulation as well as interprets the background processes into what a user can understand (Rani, Chauhan, & Chauhan, 2015).

- A region; the CloudAnalyst provides six regions based on the global continent. The regions create an environment suitable enough for large scale testing

- Simulation; The simulation component holds the parameters to be used during the simulation process (Rani, Chauhan, & Chauhan, 2015).

- UserBase; the user base emulates a group of users that works as a single unit that generates traffic for the simulation process. The modeler can either represent the simulation as a single user (Rani, Chauhan, & Chauhan, 2015).

Please see Chapter 4 and Chapter 5 for more details on CloudAnalyst.

1.15 Conclusion

Load balancing is essential in cloud computing. It allows maximum utilization of resources. As the age of technology continues to deepen, cloud computing is also changing to accommodate the need for faster and efficient dissemination of information. An overloaded system is inefficient as in some cases ineffective. Load
balancing algorithm is both economical efficient and effective in cloud computing. This chapter represents three virtual machine load balancing algorithms (Round robin, active monitoring and throttled) and also talked about some cloud simulation tools. So, next chapter will represent the data collection, the data set and preparing the simulations in order to study the behaviour of each virtual machine load balancing algorithm that mentioned in this chapter.
DESIGN AND METHODOLOGY

1.16 Introduction

Those sections outline the design of the experiment (which is a number of simulations) being carried out as part of the research topic. A detail of the data used is provided along with information regarding the transformations required in order to produce a complete data set ready for simulations. Data preparation for Large Scaled Internet Application running on the Cloud and the methodology of this dissertation are also discussed.

1.17 Focus of the experiment

The focus of this investigation concerned and tests the behavior of a large scale application on the cloud that base on reconfiguring dynamic route with load balancing service broker. The application that was simulated in this experiment is Safeer System [https://safeer.moe.gov.sa/]. Malhotra & Jain reported that The best approach to study a large, distributed and dynamic environment is through simulation (2013). The test will be for the cases of three Virtual Machine load balancing algorithms: Round Robin load balancing algorithm, Throttled load balancing algorithm, Active monitoring load balancing algorithm. This determines the timing of the large-scale cloud which is very important for the developers and cloud service provider. CloudAnalyst is The simulator that is going to be used to apply the experiment. It is 100% based on Java, so, Eclipse is required to Run CloudAnalyst framework. And the reason why this papers only focus on those three algorithms is because they are the most common and most of the researchers are interested in. What’s more, the dataset is going to be used, is from studies and surveys that were taken under the Ministry of Education in Saudi Arabia (MoE) and SACM which is Saudi Arabian Cultural Mission. The dataset is about the approximate distribution of SafeerSystem [safeer.moe.gov.sa], which is a vast scale web application. It is the main workflow portal that delivers online services and facilitates the communication between all the SACM’s employees around the world and all Saudi students overseas (SaudiMinistryOfEducation, 2014. p.11). Safeer
users are more than 120,000 in more than ten countries.  
http://www.moe.gov.sa/ar/Pages/default.aspx is one of the sources that I’m going to use it as a Dataset.

1.18 The Experiment Requirements

As the start, Microsoft Windows operating system or Macintosh operating system (OSX) is required so the investigator could install JAVA which is one of the requirement as well. That's to say, that NetBeans or Eclipse are also required since that the CloudAnalyst is 100% based on Java. And obviously, Cloud simulator is required, like CloudSim or Cloud Analyst, but it is already mentioned that the researcher used Cloud Analyst as Cloud Simulator. Moreover, the value of each criteria that was needed to perform the experiment is also considered as requirements. Those Criteria as the following:

- Data Centre Characteristics which include the detailed hardware specification of the server farm.

- User bases’ characteristics: the frequency of the traffic generation, users’ number, the peak and off-peak hours of usage.

- Grouping factors for user requests [when a message is assigned to Virtual Machines in the Data Centre and when the message is sent from User Bases].

- Internet-specific characteristics: available bandwidth and network latency.

- Specifications of Application deployment: the number of virtual machines and in which data centers should be located, also those virtual machines’ detailed specification.

Collecting the values of those criteria is considering as a dataset that was used in the experiment.
1.19 Data Set Sources

In order to study the behavior of the Virtual Machine load balancing algorithm, the researcher needed to use a cloud simulator. Therefore, the experiment is a simulation by Cloud Analyst of deploying an application on cloud that using Dynamically reconfiguration service broker. The application that is going to be simulated is Safeer system, which the researcher had a full study of in Chapter 2. In addition, regarding to simulate this application by Cloud Analyst, there are some parameters needs to be determined in terms of the application and the cloud that the application deployed on.

First of all, the parameters which is related to the application is User base characteristics, such as: the period of time when the application is used by the highest number of users continually (Peak hours start (GMT)), Off-Peak hours end (GMT), number of users use the system during the peak hour (Average peak users) and Average off-peak users. All this kind of data was taken by the mentioned websites:

- Safeer Al-Talaba https://safeer.moe.gov.sa/Sites/Student/Pages/default.aspx

However, there were some of those data was needed and they were not in those websites. So, since some of those data was missing, the researcher needed to conduct a survey _See more about the survey in Section_, which was distributed to Safeer’s users around the world, so she can collect the data she needed to do her experiment.

Secondly, in terms of the parameters which are related to the cloud that the application deployed on, such as: Data Center characteristics. the researcher followed other researchers' steps in their papers and their studies in some of the data that is required to do the experiment. For example, the studies that were mentioned in Chapter 2, which are: The study by Jena and Ahmed(2013) and the study by Singh,sharma and Kumar (2016). Both of those papers used some universal standards like Amazon EC2 standards for the cost of hosting of data centers. Both of those studies were published in International Journal of Computer Science which can be found in Google Schooler. Any way most of the researchers who published their work on library like ACM and IEEE used the same standards that Jena and Ahmed(2013) and by Singh,sharma and Kumar (2016) used in their papers.
The dataset and where it is from would be as the following:

<table>
<thead>
<tr>
<th>The criteria and what includes</th>
<th>The source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region (geographical/physical location of the data center)</td>
<td>It depends on the scenario that the investigator designed.</td>
</tr>
<tr>
<td>Arch</td>
<td>Journal of Computer Science</td>
</tr>
<tr>
<td>Operating System</td>
<td>Journal of Computer Science</td>
</tr>
<tr>
<td>Virtual Machine Management VMM</td>
<td>Journal of Computer Science</td>
</tr>
<tr>
<td>Memory (MB)</td>
<td>Journal of Computer Science</td>
</tr>
<tr>
<td>Storage (MB)</td>
<td>Journal of Computer Science</td>
</tr>
<tr>
<td>Available Bandwidth</td>
<td>Journal of Computer Science</td>
</tr>
<tr>
<td>Number of processors</td>
<td>It depends on the scenario that the investigator designed.</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>Journal of Computer Science</td>
</tr>
<tr>
<td>Virtual Machine Policy</td>
<td>It is either RR, AMLB or TLB</td>
</tr>
<tr>
<td>Cost per VM per hour</td>
<td>Amazon EC2</td>
</tr>
<tr>
<td>Memory Cost $</td>
<td>Amazon EC2</td>
</tr>
<tr>
<td>Storage Cost $</td>
<td>Amazon EC2</td>
</tr>
<tr>
<td>Data transfer cost</td>
<td>Amazon EC2</td>
</tr>
<tr>
<td>Physical Hardware unit</td>
<td>It depends on the scenario that the researcher designed.</td>
</tr>
<tr>
<td>Region (geographical/physical location of the user bases)</td>
<td>It depends on the scenario that the researcher designed.</td>
</tr>
<tr>
<td>Number of requests per user per hour</td>
<td>Survey</td>
</tr>
<tr>
<td>Data size per request (bytes)</td>
<td>Journal of Computer Science</td>
</tr>
<tr>
<td>Peak hours start (GMT)</td>
<td>Data gathered from survey</td>
</tr>
</tbody>
</table>

**Data Centre Characteristics (the detailed hardware specification of the server farm)**

**User bases’ characteristics**
Table 3.1: The sources of each parameter in the dataset

<table>
<thead>
<tr>
<th>Specifications of Application deployment</th>
<th>Internet-specific characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak hours end (GMT)</td>
<td>Journal of Computer science</td>
</tr>
<tr>
<td>Data gathered from survey</td>
<td></td>
</tr>
<tr>
<td>Average peak users</td>
<td>It is depend on the scenario that the investigator designed.</td>
</tr>
<tr>
<td>Data gathered from survey</td>
<td></td>
</tr>
<tr>
<td>Average off-peak users</td>
<td>Journal of Computer science</td>
</tr>
<tr>
<td>Data gathered from survey</td>
<td></td>
</tr>
<tr>
<td>Number of servers (Virtual Machines VM)</td>
<td>Journal of Computer science</td>
</tr>
<tr>
<td>Image size</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>Journal of Computer science</td>
</tr>
<tr>
<td>Bandwidth</td>
<td></td>
</tr>
<tr>
<td>The transition delay between region</td>
<td>Journal of Computer science</td>
</tr>
<tr>
<td>The available bandwidth between regions</td>
<td></td>
</tr>
</tbody>
</table>

1.19.1 Explanation of collecting the missing data by a Survey

Because of the time, the investigator used a SurveyMonkey which is an online survey development cloud-based software. It was only 4 questions, so from analyzing the results of these questions, the researcher can figure out the missing information she needs to complete the simulations.

The survey was only for Safeer system’s users. In addition, since that the researcher is using Safeer system as a Saudi student who study in Ireland, it was easier to her to distributed the survey quickly and got many respondents back. In fact, the researcher has contacted with about 400 users just in Ireland. Actually, as overseas students, they have What'sApp groups to feel like they are with people who are familiar with. Besides that the researcher is joined 4 What'sApp groups with 50 different students in each one of them. So it was easy to just send the link of the survey and asked those other students to fill it. And because it was only 4 questions, so almost all of them were done it. What's more, the survey link was distributed also by those students in the What'sApp groups. They sent it to their partners (husbands/ wives) and their friends _who also Safeer's users_ either in Ireland or in other countries.
Moreover, the researcher also know some friends and relatives who study in US, Canada, UK, Japan, Malaysia, Germany and Korea. So the link of the survey was also sent to them and they sent it to their friends who they study abroad as well. That's how the investigator got 1287 Safeer's users responded to the survey.

The questions of the survey were as it is shown in Figure 3.1. Please see the appendix for the Survey questions.

![Figure 3.1: The online survey](image)

From these simple questions, The researcher found when is the peak hour and off-peak hour and how many users during these two durations. Also, she figured out how many services the user request per hour. And, as it's mentioned that the respondents were only 10% of all Safeer users. That is to say, 1287 Safeer users from around the world had done the survey from 7\11\2016 to 18\11\2016.
1.20 The Data Set

In order to study these three Load Balancing Algorithm, number of scenarios of a large-scale cloud environment simulation must be set up by configuring the criteria that was explained in Section 3.3. All these criteria will be entered through the simulator tool (CloudAnalyst) number of times until it achieved the examination of all three VM load balancing algorithms with all their scenarios. Although most of the data which the simulation needs are available in papers were taken under the Ministry of Education in Saudi Arabia (MoE), SACM which is Saudi Arabian Cultural Mission, and Computer Science Journals. However, there is still information the researcher need to collect that’s why a survey was conducted, see section 3.4.

1.20.1 Datacenters characteristics:

To specify the number and the actual geographic location for each data center of Safeer system, the researcher investigated the number of Safeer’s users and where they base.

These data was extracted from the website of Ministry of Education of Saudi Arabia as the following:

<table>
<thead>
<tr>
<th>Location</th>
<th>Number of users</th>
<th>The continent</th>
<th>Time zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>57211</td>
<td>North America</td>
<td>GMT-6</td>
</tr>
</tbody>
</table>

Figure 3.2: Number of respondent per day
<table>
<thead>
<tr>
<th>Country</th>
<th>Number</th>
<th>Region</th>
<th>Time Zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK</td>
<td>14459</td>
<td>Western European</td>
<td>GMT 00:00</td>
</tr>
<tr>
<td>Canada</td>
<td>13801</td>
<td>North America</td>
<td>GMT -6</td>
</tr>
<tr>
<td>Australia</td>
<td>8789</td>
<td>Pacific</td>
<td>GMT +9:30</td>
</tr>
<tr>
<td>New Zealand</td>
<td>2049</td>
<td>Pacific</td>
<td>GMT +13:00</td>
</tr>
<tr>
<td>Ireland</td>
<td>1707</td>
<td>Western European</td>
<td>GMT 00:00</td>
</tr>
<tr>
<td>China</td>
<td>1143</td>
<td>Asia</td>
<td>GMT +8:00</td>
</tr>
<tr>
<td>Malaysia</td>
<td>1105</td>
<td>Asia</td>
<td>GMT +8:00</td>
</tr>
<tr>
<td>Germany</td>
<td>945</td>
<td>Central European</td>
<td>GMT +1:00</td>
</tr>
<tr>
<td>France</td>
<td>923</td>
<td>Central European</td>
<td>GMT +1:00</td>
</tr>
<tr>
<td>Poland</td>
<td>774</td>
<td>Central European</td>
<td>GMT +1:00</td>
</tr>
<tr>
<td>India</td>
<td>609</td>
<td>Asia</td>
<td>GMT +5:30</td>
</tr>
<tr>
<td>Japan</td>
<td>499</td>
<td>Asia</td>
<td>GMT +9:00</td>
</tr>
<tr>
<td>The Netherland</td>
<td>326</td>
<td>Central European</td>
<td>GMT +1:00</td>
</tr>
<tr>
<td>South Korea</td>
<td>200</td>
<td>Asia</td>
<td>GMT +9:00</td>
</tr>
<tr>
<td>Switzerland</td>
<td>100</td>
<td>Central European</td>
<td>GMT +1:00</td>
</tr>
<tr>
<td>Slovakia</td>
<td>77</td>
<td>Central European</td>
<td>GMT +1:00</td>
</tr>
<tr>
<td>Italy</td>
<td>40</td>
<td>Central European</td>
<td>GMT +1:00</td>
</tr>
<tr>
<td>Africa</td>
<td>15719</td>
<td>Africa</td>
<td>GMT +3:00</td>
</tr>
</tbody>
</table>

**Table 3.2: Safeer’s user's number and where they base**

Moreover, in terms of the rest of the criteria, the investigator assumed a plan which almost follows the real and actual plan of Amazon EC2—one of the most popular Cloud Service providers these days_. That's to say, the investigator followed other researchers' steps in configuring the hardware architecture of a cloud. The presumed plan is:

Physical hardware of each Data Centre:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value Used</th>
</tr>
</thead>
</table>
### Arch

**Arch**

- **x86**

### Operating System

**Operating System**

- **Linux**

### Virtual Machine Management VMM

**Virtual Machine Management VMM**

- **Xen**

### Memory (MB)

**Memory (MB)**

- **1024 Mb**

### Storage (MB)

**Storage (MB)**

- **1000**

### Available Bandwidth

**Available Bandwidth**

- **1000**

### Number of processors

**Number of processors**

- **4**

### Processor Speed

**Processor Speed**

- **100 MIPS**

### Virtual Machine Policy

**Virtual Machine Policy**

- **Time Shared**

#### Table 3.3: Physical hardware of each Data Centre

The cost of hosting of the data centers:

<table>
<thead>
<tr>
<th>Cost per VM per hour (1024Mb, 100MIPS)</th>
<th>$0.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost per 1Gb of data transfer (from/to the Internet)</td>
<td>$0.10</td>
</tr>
</tbody>
</table>

#### Table 3.4: The cost of hosting of the data centers

1.20.2 User bases’ characteristics

The world is divided to six different regions that coincide roughly with North America, South America, Asia, Africa, Europe and Australia. Those regions are the six main continents. Therefore, The user base locations is depending on table 3.2 where Safeer users are.

Each user base is contained in a single time zone and according to the survey results that most of Safeer users use the application during the SACM office work, from 9 am to 4 pm. Also, according to the survey, 30% of the registered users will be online during the peak time simultaneously and only one tenth of that number during the off-peak hours. What else, the survey showed to us that each user makes a new request every 15 minutes when online.
<table>
<thead>
<tr>
<th>Region</th>
<th>Userbase</th>
<th>Country</th>
<th>Number of users</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0</td>
<td>UB1</td>
<td>US</td>
<td>57211</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Canada</td>
<td>14459</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UK</td>
<td>14459</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ireland</td>
<td>1707</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Germany</td>
<td>945</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>France</td>
<td>923</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Poland</td>
<td>774</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Netherland</td>
<td>326</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Switzerland</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slovakia</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Italy</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>UB2</td>
<td>China</td>
<td>1143</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Malaysia</td>
<td>1105</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>India</td>
<td>609</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Japan</td>
<td>499</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>South Korea</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>UB3</td>
<td>Africa</td>
<td>15719</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>UB4</td>
<td>Australia</td>
<td>8789</td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>UB5</td>
<td>New zealand</td>
<td>2049</td>
<td></td>
</tr>
</tbody>
</table>

3.5 User base characteristics

1.20.3 Specification of Application deployment and internet characteristics

Since this kind of large scale application which has a different amount of load at different datacenter in various locations at different time, the design of the cloud needs to use Reconfigure dynamically with rout load. So, for some cases case, 4 Datacenters
were set up with 100, 75, 50, 25 virtual machines within for each one of them. And other cases, only 2 Datacenter were set up with 100 virtual machines in each one of them.

However, the rest of the application deployment specification and the internet characteristics are according to Amazon E2 as the following:

<table>
<thead>
<tr>
<th>Region/Region</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>25.0</td>
<td>100.0</td>
<td>150.0</td>
<td>250.0</td>
<td>250.0</td>
<td>100.0</td>
</tr>
<tr>
<td>1</td>
<td>100.0</td>
<td>25.0</td>
<td>250.0</td>
<td>500.0</td>
<td>350.0</td>
<td>200.0</td>
</tr>
<tr>
<td>2</td>
<td>150.0</td>
<td>250.0</td>
<td>25.0</td>
<td>150.0</td>
<td>150.0</td>
<td>200.0</td>
</tr>
<tr>
<td>3</td>
<td>250.0</td>
<td>500.0</td>
<td>150.0</td>
<td>25.0</td>
<td>500.0</td>
<td>500.0</td>
</tr>
<tr>
<td>4</td>
<td>250.0</td>
<td>350.0</td>
<td>150.0</td>
<td>500.0</td>
<td>25.0</td>
<td>500.0</td>
</tr>
<tr>
<td>5</td>
<td>100.0</td>
<td>200.0</td>
<td>200.0</td>
<td>500.0</td>
<td>500.0</td>
<td>25.0</td>
</tr>
</tbody>
</table>

Table 3.6: Latency Matrix values (in milliseconds)

<table>
<thead>
<tr>
<th>Region/Region</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
</tr>
<tr>
<td>1</td>
<td>1000.0</td>
<td>800.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
</tr>
<tr>
<td>2</td>
<td>1000.0</td>
<td>1000.0</td>
<td>2500.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
</tr>
<tr>
<td>3</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1500.0</td>
<td>1000.0</td>
<td>1000.0</td>
</tr>
<tr>
<td>4</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>500.0</td>
<td>1000.0</td>
</tr>
<tr>
<td>5</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>1000.0</td>
<td>2000.0</td>
</tr>
</tbody>
</table>

Table 3.7: Bandwidth Matrix values (in Mbps)

1.21 Experiment Preparation

1.21.1 Install CloudAnalyst and Eclipse

Firstly, the source code of Cloud Analyst and Eclipse are needed to be downloaded. This Cloud Analyst package contains all the CloudSim classes and all the Jars that are needed to make Cloud Analyst run. Then, A Java project is acquired to be created to
install Cloud Analyst and configure it in Eclipse. After that, all Cloud Analyst file resources should be imported to that Java project. Therefore, Cloud Analyst, now, can be run.

1.21.2 Data preparation for Large Scaled Internet Application running on the Cloud

Large scale application means it is used by a huge number of users and usually from all over the globe. So, It makes sense to make this application hosted on several servers (datacenters) around the world. And since the geographical location affects the speed of the application’s performance, it is really important to host the application in datacenters where most of the users are. Moreover, As much as the number of users in the specific region, the requests from that region to the application are. In addition, since there are timings when users use the application and it’s not fixed, number of Virtual Machines which perform users requests should be dynamically allocated relying on the current workload. For example, the highest load from a region occurs from 15:00-17:00 GMT and through this time the datacenter servicing and processing these requests (the datacenter located in the region itself) must have a higher number of VMs allocated. However, when the peak has passed, the number of VM’s could be dynamically reduced, and the number in a different datacenter facing higher load can be increased. That’s why Reconfigure dynamic route with load balancing is the proper service broker for this kind of application to consume the cost and the time.

Malhotra & Jain reported that The best approach to study a large, distributed and dynamic environment _like a cloud_ is through simulation (2013). Cloud Analyst is the simulator that is going to be used to examine the behavior of a large scaled application (Safeer system) on the Internet. However, to set up a simulation for Safeer system on Cloud Analyst tool, certain parameters should be configured in the tool, See Section 3.5.

To sum up the experiment preparation section, the duration of the simulation must be specified which can be given in minutes, hours or days. In Safeer system case, 60 hours is good enough to study the behavior of the system. In addition, depending on table 3.5, the investigator had the ability of specifying how many data centre does
Safeer system deploy on, how many virtual machines does each data centre has, and where is the physical location for each data centre.

1.22 Set up the simulations' process (Methodology of the Experiment)

Firstly, download and install Cloud Analyst using this website:

http://www.cloudbus.org/cloudsim/CloudAnalyst.zip

Secondly, extract all Cloud Analyst folders and files from the Zip folder which is giving the following structure of folders as it is shown in figure 3.3.

Figure 3.3 : Folder structure

Or run it from the command line then click on run.bat file as it is shown in figure 3.4

Figure 3.4 : Run through the command line
Then Cloud Analyst is ready to be used as it shown in Figure 3.5, then click on the icon of Show Region Boundaries as it is shown in figure 3.5.

![Figure 3.5: Cloud Analyst](image)

After that Click on the icon of Configure Simulation

From here the configuration can be done:

1. Data Center Configuration
   1.1 Physical Hardware Details of Data center

![Figure 3.6: The Boundaries](image)

![Figure 3.7: Data center configuration and adding number data centers](image)
2. Advanced configuration

Figure 3.8 : Data center configuration

Figure 3.9: Details of data center configuration

Figure 3.10 : Copy the number of severs in each data center

Figure 3.11: The advanced Configuration.
2.3. User Base configuration which are a group of users and their generates traffic representing the application's users.

2.4. Configurations of the Application Deployment (Cloudlets)

2.5. Define the Service Broker Policy and in this study the service broker policy is Dynamically Reconfiguration.

![Figure 3.12: The main configuration 1](image1)

![Figure 3.13: The main configuration 2](image2)

Also, there is an ability to Save the Configuration to use it again. The configuration will be stored as .sim file which is an XML data that is generated and stored as Sim file.
How ever the internet characteristics need to be configured as well.

Once the Configuration is done just click on Done icon and then click on Run Simulation icon.
So, the Main Window will give all the results as the statistics

![Figure 3.16: Simulation results Statistics](image)

1.23 **Methodology (of all thesis)**

To undertake the comparison of the most popular virtual machine load balancing algorithms in the cloud environment, an investigation of cloud computing, the Technological Development in the Cloud, had been taken. What’s more, a deep study about large scale application based on cloud and the policies that made the cloud more efficient for this kind of applications had been taken as well. This study took about three weeks to be done. After that, the investigator figured out that the most suitable way to study the behavior of the virtual machine load balancing algorithms is through a simulation. That’s why the researcher did some study on Cloud simulation tools and did a study of one of the largest applications in Saudi Arabia which are G2C platform (Safeer system) to help communication between all the student overseas and their Saudi Arabia Cultural Mission. This study had been done by a survey distributed online through Safeer system and going through a number of papers were published by SACMs and Computer science journals. This process also took about two weeks to be done and covered. Although, after this five weeks, the vision had been completed and all the data has been collected, the researcher must prepare some data to be suitable with the study and the simulation experiment. Then, finally the researcher could do the simulation in one of the best tools that had been studied and has been recommended by many researchers which is cloud analyst.
Figure 3.17: The five phases of this research

1.23.1 Literature review:

The literature review was a deep study about cloud computing and all its technology and policies. Specially the service broker polices and the virtual machine load balancing algorithms. And of course, the study was all about the large scale application on cloud. In addition, the researcher considered one example of large scale application as an example and so the researcher can use this example in the simulations experiment. Besides, a study about the most suitable cloud simulator tool so the simulation can be undertaken.

In the literature review, the investigator knew exactly what information she is going to need to complete the simulations and to answer the research question.

1.23.2 Data collection:

This chapter illustrates this part of data collection and preparation, from where the researcher collect her data and how the data became after the preparation. Table 3.1 explain in detail from where each information was extracted. Therefore, the data was collected from different resources like papers were published under International Journal of computer science, ACM, IEEE, Amazon EC2, Saudi ministry of education (MoE) and Saudi Arabian Commission Mission (SACM). Moreover, as it’s explained at the beginning of this chapter that some information about the application’s user was missing, so that’s why the investigator needed to distribute a survey to collect those information from the applications’ users. Since the users are either Saudi students or their supervisors as SACM employees, the investigator distributed the survey online.
through What’sApp. Because it’s the most popular social network application is using by Saudi’s people, especially the ones who want to communicate with their families or friends. So, What’sApp is the fastest way to distribute information, news, or in this dissertation case a link to the survey. See more about the Survey in Section 3.4.1.

What is important from the survey results is that 30% of the respondents says they are using the system every day during SACM office hour from 9:00 to 16:00 and they request about 6-10 services each time they are using Safeer system. Besides 10% of these users check their request (using the system again) out of the office hour. The following figures illustrate the result of the survey in numbers.

![Figure 3.18: Number of how often users use the system](image1)

![Figure 3.19: Number of users who request a certain amount of services](image2)
1.23.3 Data preparation:

To do the simulations, there are some of the data is needed to be prepared. For example, CloudAnalyst simulator accepts the peak and off-peak hour only in Greenwich Mean Time (GMT). Moreover, there are many user bases distributed in different time zone. And the only information the researcher has, is the peak hour and off-peak hour which starts and ends at as the SACM’s office time. Therefore, the researcher used https://greenwichmeantime.com/time-gadgets/time-zone-converter/ to convert the time of each user base to the GMT.

Furthermore, since the respondents of the survey was only 10% of the overall Safeer’s users. The researcher needed to generalize all the results of the 10% to the whole number of users.
For instance, 30% of the respondents use the system during the peak hour. So for each User base, the researcher will only include 30% as peak-hour users.

For example, Number of users in North America is 71760 users. So the 30% of those users is \((71,760 \times 0.30)\) which is 21,528 users. And this is the number of users in North America who use the system during the SACM’s office hour.

3.5.4 Using Cloud Analyst

CloudAnalyst is based on CloudSim which is 100% java. So the researcher used Eclipse in order to run CloudAnalyst and set up the simulations. Firstly, the source code of CloudAnalyst and Eclipse are needed to be downloaded. This CloudAnalyst package contains all the CloudSim classes and all the Jars that are needed to make CloudAnalyst run. Then, A Java project is acquired to be created to install CloudAnalyst and configure it in Eclipse. After that, all CloudAnalyst file resources should be imported to that Java project. Therefore, CloudAnalyst, now, can be run. Figure 3.5 shows the CloudAnalyst java codes using Eclipse. Figure 3.5 also shows CloudAnalyst files in the project explorer tab.

---

![CloudAnalyst in Eclipse](image-url)
1.23.4 Design the simulations:

In order to compare among the Virtual Machine (VM) Load Balancing algorithms (Round Robin, Active Monitoring and Throttled load balancing algorithm), The researcher designed number of simulations for each VM load balancing algorithm.

The scenario of each simulation is different from the others in one criterion or two criteria. Each simulation about deploying Safeer application on the cloud using one of the VM load balancing algorithms that were mentioned above and Dynamically reconfiguration service broker. The researcher designed four scenarios for each VM load balancing algorithms to study the behavior of each VM load balancing algorithm under number of cases as the following.

Scenario 1:

When the application is deployed on many datacenter with small number of servers that are virtualized into different amount of Virtual Machine: to test and see if is worth it for the service provider to deploy the clients application on many data centers in different locations around the world which will cost the client much money.

Scenario 2:

When the application is deployed on the same number of data centre in Scenario 1 with more servers then Scenario 1 which are virtualized into that the same number in Scenario 1: to test the affect of servers number.

Scenario 3:

When only small number of data centers distributed on locations in the middle of all user bases: to approve the importance of allocating the data centers where all the users requests can reach as fast as it could.

Scenario 4:

When the data centers have the same criteria except they are distributed on locations which is not in the centre of the user bases with huge number of servers that are virtualized into huge number of virtual machine: to see if it is important to allocate the data centre in the middle of the higher user bases.
Next chapter will explain in detail each scenario in the simulation.

1.23.5 The simulations

The experiment includes 12 simulations, four simulations (scenario) for each VM load balancing algorithm. Basically, each simulation is about deploying Safeer application on a cloud which is using Dynamically reconfigure route with load service broker policy. Most of the parameters’ value are the same in those four scenarios. However, some of them are different from scenario to another. Each scenario is a simulation of 60 hours of running the application on the cloud. The configuration of each simulation uses the data had been collected and prepared by the investigator in the previous sections (Section 3.5). However, next chapter will illustrate in detail about each scenario for each VM load balancing algorithms. In general, the researcher divided her experiment into 3 phases, one phase for each VM load balancing algorithm like the following:

Phase1: 4 different scenario (simulations) for RoundRobin Load balancing algorithm.

Phase2: 4 different scenario(simulations) ActiveMonitoring Load balancing algorithm.

Phase3: 4 different scenario(simulations) Throttled Load balancing algorithm.

1.23.6 Evaluation the result

To evaluate the results of each simulation, the researcher needs to analyze the generated report by CloudAnalyst fro each simulation. By using Tableau and Excel, the researcher can easily present the results and the data analysis of the experiment.

1.24 Conclusion

This chapter discussed the methodology that the researcher followed in order to do her dissertation. Also, this chapter presented in detail the data that’s used in the experiment.
and from where it was collected and how it was prepared. Besides, this chapter gave an overview of designing the simulations and the simulations themselves, which are, in other words, the simulations setup and configurations. However, in the next chapter, those two topics will be discussed in detail.
IMPLEMENTATION AND RESULTS

1.25 Introduction

The investigator had to simulate a large application in order to compare the Virtual Machine load balancing algorithms which play the main role in the behavior of the application’s cloud. Therefore, the investigator did some research on large-scale application as well as collected an application’s data that is needed to be deployed on the cloud. The application is Safeer system. safer system is a large application that is studied by the investigator regarding to do the simulation. As it is mentioned in chapter 3 how the researcher collected and prepared the data that is needed in the experiment. And also last chapter (chapter 3) gives a preview how to install cloud analyst and run it in Eclipse. This chapter is talking about set up a simulation (design the simulation) to study the behavior of cloud which is effected by virtual machine load balancing algorithm under 4 cases to determine the affect of the number and the locations of the datacenters and their hardware architecture. The simulation should be done four times for each VM load balancing algorithm to figure out what other factors that affect on a cloud as well.

1.26 Environment Setup

To set up a simulation the researcher carried out the following steps using CloudAnalyst:

1. Define the duration of the simulation.
2. Define UserBases characteristics
3. Define DataCenter characteristics
4. Allocate Virtual Machines for the application in each Data Centers
5. Define the load balancing policy across VMs in a data center.
6. Review and adjust the Internet Characteristics (bandwidth matrices and network latency).
As it is explained in the previous list, the simulation had been carried out using Cloud Analyst tool. Figure 4.1 shows the main configuration tab in cloud analyst. From this tab, the simulation duration should be assigned as well as the service broker policy. In addition, in this tab, both of user bases characteristics and Application deployment configuration must be defined. However, in the data center configuration tab, all the values of each data centre parameters and the details of each physical hardware of each data center must be configured in this tab. The Advanced tab includes “Load balancing policy across virtual machines in data center” options.

![Figure 4.1: The main configuration tab in cloud analyst](image)

1.27 The Configuration (Design) of Safeer system simulation

As it’s mentioned in the previous sections that is going to be four simulations as four scenarios for each VM load balancing algorithm. Those four scenarios have the same value for some parameters. So, they were sharing the pricing plan that most of the cloud providers use it, which is Amazon EC2 original pricing plan. The pricing plan as it was explained in table 3.4, which is: Cost per 1Gb of data transfer (from/to Internet): $0.10; Cost per Virtual Machine per hour (1024Mb, 100MIPS): $ 0.10. That is to say,
the investigator followed other researchers’ step in terms of the cost of hosting applications in a Cloud, besides, the duration of the simulation. Therefore, each simulation/scenario was 60 hours which is a simulation of almost three days of Safeer application running on the cloud.

Figure 4.2: Cost Configuration

Figure 4.3: Duration simulation
According to the data that had been gathered from Computer science journals, most of the cloud service provider use the same parameters of data centers and its virtual machine and Internet characteristics. The parameter’s value as it is shown in the tables in Chapter 3, which is as the following: Virtual machines size used to host applications within the experiment is 100MB. Also, they have 10MB of available bandwidth and 1GB of RAM memory. In addition, Simulated hosts have Linux operating system, x86 architecture and virtual machine monitor Xen. However, Each simulated datacenter hosts a different number of virtual machines in each scenario. Those Machines have 100GB of storage and 2 GB of RAM. In addition, the policy that used to schedule resources to VMs is a time-shared policy. Figure 4.4 is another example of how to configure some parameters in the simulation.

![Figure 4.4: example of how to configure some parameters](image)

Moreover, the researcher defined number user bases representing the users of Safeer system, and their locations are depending on which continent and time zone they are. These user bases with their parameters described in Table 3.2. And as illustrated last chapter, Only 30% of the users use the application simultaneously during the office hour of Saudi Arabia Culture Mission in each country, and that’s between 9:00 to 16:00 and this is considered as peak hour. Only 10% of that number of
users is using the system during the off-peak hours. That's to say, each user makes a new request every 15 minutes when he or she is online.

1.27.1 Design Scenario 1 and Scenario 2

In scenario 1 and 2, five user bases were defined representing which region/continent and time zone they are. These user bases with their parameters described in Table 3.8. Also, four datacenters was defined in different regions: R0, R2, R3, R4, and R5. However, the characteristics of these data centers are different within scenario 1 and 2. In scenario one, the number of physical hardware within each data center is only 2. On the other hand, each data center, in scenario 2, contains ten physical hardware. In addition, Users are grouped by a factor of 100 in scenario 1, and 1000 in scenario 2. And requests are grouped by a factor of 10 in scenario 1, and 100 in scenario 2. Each user request requires 25 instructions to be executed in scenario 1, and 250 in scenario 2. This part is already explained and represented in tables (table 3.7, table 3.8 and table 3.9) within last chapter (chapter 3).

<table>
<thead>
<tr>
<th>Region</th>
<th>Number of data center</th>
<th>Name of data center</th>
<th>Number of Virtual Machine</th>
<th>Number of physical hardware unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Scenario 1</td>
</tr>
<tr>
<td>R0</td>
<td>1</td>
<td>DC1</td>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>R2</td>
<td>1</td>
<td>DC2</td>
<td>75</td>
<td>2</td>
</tr>
<tr>
<td>R4</td>
<td>1</td>
<td>DC3</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>R5</td>
<td>1</td>
<td>DC4</td>
<td>25</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.1: The differences between the design/configuration of Scenario 1 & 2

1.27.2 Design Scenario 3 and Scenario 4

In scenario 3 and 4, eight user bases were defined representing which region/continent and time zone they are. These user bases with their parameters described in Table 3.10. Also, two datacenters was defined in different locations in each scenario: In scenario 3; R0, R2; In scenario 4 R0, R4. Moreover, the characteristics of these data centers are
the same in both scenario 3 and 4. The number of physical hardware in each data centre is 20. In addition, in both scenarios, users are grouped by a factor of 1000. And requests are grouped by a factor of 100. And each user request requires 250 instructions to be executed. This part is already explained and represented in tables (table 3.10, table 3.11 and table 3.12) within last chapter (chapter 3).

<table>
<thead>
<tr>
<th>Region</th>
<th>Scenario 3</th>
<th>Scenario 4</th>
<th>Number of data center</th>
<th>Name of data center</th>
<th>Number of Virtual Machine</th>
<th>Number of physical hardware unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0</td>
<td>R0</td>
<td>1</td>
<td>DC1</td>
<td>100</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>R3</td>
<td>1</td>
<td>DC2</td>
<td>100</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: The differences between the design/configuration of Scenario 3 & 4

Figure 4.5: The distribution of the regions.

1.28 Experiment: Simulation and results

In the experiments, the effects of the virtual machine load balancing algorithms were investigated. The overall average response time was the main performance metrics. The simulations were conducted for 12 times to study the behavior of each VM load balancing algorithm under specific cases related to the number of datacenter and its
hardware architecture and its distributions. Number of variants were changed each time: the number and distribution of datacenter, the physical hardware of each data center. Next section illustrated the response time and the data processing time for each scenario for each VM load balancing algorithm.

1.28.1 Scenario 1

This scenario was done three times for each VM load balancing algorithm. Five user bases were defined in R0, R2, R3, R4, R5 depending on which continent they are. R0 represents North America, R2: Europe, R3: Asia, R4, Africa, R5: Pacific. Moreover, 4 data centers were distributed in different region depending on number of users in each user base. D1 in R0, D2 in R2, D3 in R3, D4 in R5. In each of those data centers contains only two physical hardware units. As it can be seen, this scenario has this number of data center with this low number of servers to test and see if is worth it for the service provider to deploy the clients application on many data centers in different locations around the world which will cost the client much money. Figure 4.6 shows the enormous difference in the results as the average response time of each VM load balancing algorithms. However, Table 4.3 illustrates the response time by each user base and the processing time by each data center.

Figure 4.6: Results of scenario 1
<table>
<thead>
<tr>
<th></th>
<th>Round Robin</th>
<th>Active Monitoring</th>
<th>Throttled</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg (ms)</td>
<td>Min (ms)</td>
<td>Max (ms)</td>
</tr>
<tr>
<td>UB1</td>
<td>58.25</td>
<td>44.74</td>
<td>75.81</td>
</tr>
<tr>
<td></td>
<td>58.19</td>
<td>44.74</td>
<td>75.67</td>
</tr>
<tr>
<td></td>
<td>86.69</td>
<td>47.09</td>
<td>146.56</td>
</tr>
<tr>
<td>UB2</td>
<td>68.60</td>
<td>42.67</td>
<td>235.32</td>
</tr>
<tr>
<td></td>
<td>54.64</td>
<td>43.67</td>
<td>82.56</td>
</tr>
<tr>
<td></td>
<td>71.95</td>
<td>45.33</td>
<td>92.15</td>
</tr>
<tr>
<td>UB3</td>
<td>325.46</td>
<td>249.94</td>
<td>618.80</td>
</tr>
<tr>
<td></td>
<td>31658.80</td>
<td>42.72</td>
<td>990042.23</td>
</tr>
<tr>
<td></td>
<td>61630.80</td>
<td>44.34</td>
<td>1107044.75</td>
</tr>
<tr>
<td>UB4</td>
<td>8183.64</td>
<td>43.64</td>
<td>1107051.38</td>
</tr>
<tr>
<td></td>
<td>303.78</td>
<td>252.33</td>
<td>371.73</td>
</tr>
<tr>
<td></td>
<td>320.77</td>
<td>259.51</td>
<td>380.08</td>
</tr>
<tr>
<td>UB5</td>
<td>59.85</td>
<td>44.81</td>
<td>81.60</td>
</tr>
<tr>
<td></td>
<td>53.79</td>
<td>42.15</td>
<td>78.10</td>
</tr>
<tr>
<td></td>
<td>65.00</td>
<td>43.28</td>
<td>86.86</td>
</tr>
<tr>
<td>DC1</td>
<td>8.07</td>
<td>0.44</td>
<td>19.17</td>
</tr>
<tr>
<td></td>
<td>11.09</td>
<td>0.21</td>
<td>61.00</td>
</tr>
<tr>
<td></td>
<td>36.65</td>
<td>0.69</td>
<td>90.05</td>
</tr>
<tr>
<td>DC2</td>
<td>22.64</td>
<td>0.27</td>
<td>314.30</td>
</tr>
<tr>
<td></td>
<td>4.36</td>
<td>0.14</td>
<td>103.90</td>
</tr>
<tr>
<td></td>
<td>21.92</td>
<td>0.57</td>
<td>32.23</td>
</tr>
<tr>
<td>DC3</td>
<td>8133.69</td>
<td>0.27</td>
<td>1106996.17</td>
</tr>
<tr>
<td></td>
<td>98601.03</td>
<td>0.18</td>
<td>1106990.95</td>
</tr>
<tr>
<td></td>
<td>61580.82</td>
<td>1.03</td>
<td>1106992.85</td>
</tr>
<tr>
<td>DC4</td>
<td>10.01</td>
<td>0.56</td>
<td>27.16</td>
</tr>
<tr>
<td></td>
<td>1.69</td>
<td>0.11</td>
<td>20.51</td>
</tr>
<tr>
<td></td>
<td>15.28</td>
<td>0.64</td>
<td>32.02</td>
</tr>
</tbody>
</table>

Table 4.3: results of scenario 1

1.28.2 Scenario 2

This scenario was done three times for each VM load balancing algorithm. Five user bases were defined in R0, R2, R3, R4, R5 depending on which continent they are. R0 represents North America, R2: Europe, R3: Asia, R4, Africa, R5: Pacific. Moreover, 4 data centers were distributed in the different region depending on number of users in each user base. D1 in R0, D2 in R2, D3 in R3, D4 in R5. In each of those data centers contains only 10 physical hardware units. Apparently, this scenario has the same value of most criteria as in the Scenario 1. However, the only difference is the number of servers which is in here 10 servers in each data center. And this is to test the affect of servers number. Figure 4.7 shows the huge difference among the average
response time of each VM load balancing algorithms. However, Table 4.4 illustrates the response time by each user base and the processing time by each data center.

![Image of Figure 4.7: Result of scenario 2]

<table>
<thead>
<tr>
<th></th>
<th>Round Robin</th>
<th>Active Monitoring</th>
<th>Throttled</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg (ms)</td>
<td>Min (ms)</td>
<td>Max (ms)</td>
</tr>
<tr>
<td>UB1</td>
<td>58.27</td>
<td>44.74</td>
<td>75.81</td>
</tr>
<tr>
<td></td>
<td>58.19</td>
<td>44.74</td>
<td>75.67</td>
</tr>
<tr>
<td></td>
<td>57.10</td>
<td>44.74</td>
<td>71.90</td>
</tr>
<tr>
<td>UB2</td>
<td>66.83</td>
<td>44.05</td>
<td>238.46</td>
</tr>
<tr>
<td></td>
<td>54.64</td>
<td>43.67</td>
<td>82.56</td>
</tr>
<tr>
<td></td>
<td>54.51</td>
<td>42.67</td>
<td>70.93</td>
</tr>
<tr>
<td>UB3</td>
<td>58665.79</td>
<td>42.75</td>
<td>108045.43</td>
</tr>
<tr>
<td></td>
<td>31658.80</td>
<td>42.72</td>
<td>99042.23</td>
</tr>
<tr>
<td></td>
<td>306.39</td>
<td>242.11</td>
<td>379.96</td>
</tr>
<tr>
<td>UB4</td>
<td>313.24</td>
<td>252.33</td>
<td>472.72</td>
</tr>
<tr>
<td></td>
<td>303.78</td>
<td>252.33</td>
<td>371.73</td>
</tr>
<tr>
<td></td>
<td>54.67</td>
<td>43.66</td>
<td>65.90</td>
</tr>
<tr>
<td>UB5</td>
<td>59.56</td>
<td>42.83</td>
<td>80.93</td>
</tr>
<tr>
<td></td>
<td>53.79</td>
<td>42.15</td>
<td>78.10</td>
</tr>
<tr>
<td></td>
<td>53.18</td>
<td>43.40</td>
<td>66.46</td>
</tr>
<tr>
<td>DC1</td>
<td>8.07</td>
<td>0.44</td>
<td>19.17</td>
</tr>
<tr>
<td></td>
<td>8.02</td>
<td>0.44</td>
<td>18.82</td>
</tr>
<tr>
<td></td>
<td>6.91</td>
<td>0.44</td>
<td>14.20</td>
</tr>
<tr>
<td>DC2</td>
<td>15.79</td>
<td>0.23</td>
<td>314.30</td>
</tr>
<tr>
<td></td>
<td>4.78</td>
<td>0.23</td>
<td>38.28</td>
</tr>
<tr>
<td></td>
<td>5.33</td>
<td>0.27</td>
<td>7.48</td>
</tr>
<tr>
<td>DC3</td>
<td>58615.77</td>
<td>0.27</td>
<td>1106996.17</td>
</tr>
<tr>
<td></td>
<td>31608.78</td>
<td>0.28</td>
<td>989993.03</td>
</tr>
<tr>
<td></td>
<td>4.66</td>
<td>0.27</td>
<td>7.24</td>
</tr>
<tr>
<td>DC4</td>
<td>9.69</td>
<td>0.14</td>
<td>27.16</td>
</tr>
<tr>
<td></td>
<td>3.94</td>
<td>0.14</td>
<td>25.83</td>
</tr>
<tr>
<td></td>
<td>3.35</td>
<td>0.14</td>
<td>6.95</td>
</tr>
</tbody>
</table>

**Table 4.4: Scenario 2 results**
1.28.3 Scenario 3

8 user bases were defined UB1: R0, UB2/UB3: R2, UB4/UB5, UB6: R3, UB7: R4 and UB8: R5 depending on which Time zone they are. R0 represents North America, R2: Europe, R3: Asia, R4, Africa, R5: Pacific. Moreover, only 2 data centers were distributed in two different regions depending on number of users in each user base. DC1 in R0 and DC2 in R2. In each of those data centers contains 20 physical hardware units. As it can be seen that the data centers are distributed in North America and Europe. And Europe is considered as the center of all user bases. So, this Scenario to test and approve the importance of allocating the data centers where all the users requests can reach as fast as it could. Figure 4.8 shows the huge difference among the average response time of each VM load balancing algorithms. However, table illustrates the response time by each user base and the processing time by each data centre.

![Figure 4.8: Scenario 3 results](image-url)
Table 4.5: Scenario 3 results

<table>
<thead>
<tr>
<th></th>
<th>Round Robin</th>
<th>Active Monitoring</th>
<th>Throttled</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg (ms)</td>
<td>Min (ms)</td>
<td>Max (ms)</td>
</tr>
<tr>
<td>UB1</td>
<td>53.63</td>
<td>39.88</td>
<td>68.86</td>
</tr>
<tr>
<td>UB2</td>
<td>52.42</td>
<td>39.79</td>
<td>67.60</td>
</tr>
<tr>
<td>UB3</td>
<td>52.12</td>
<td>39.83</td>
<td>64.67</td>
</tr>
<tr>
<td>UB4</td>
<td>301.95</td>
<td>237.58</td>
<td>376.73</td>
</tr>
<tr>
<td>UB5</td>
<td>299.89</td>
<td>233.10</td>
<td>369.47</td>
</tr>
<tr>
<td>UB6</td>
<td>300.30</td>
<td>237.89</td>
<td>368.03</td>
</tr>
<tr>
<td>UB7</td>
<td>301.81</td>
<td>228.66</td>
<td>369.64</td>
</tr>
<tr>
<td>UB8</td>
<td>201.59</td>
<td>153.08</td>
<td>249.14</td>
</tr>
<tr>
<td>DC1</td>
<td>3.17</td>
<td>0.04</td>
<td>7.48</td>
</tr>
<tr>
<td>DC2</td>
<td>2.06</td>
<td>0.03</td>
<td>6.61</td>
</tr>
</tbody>
</table>

1.28.4 Scenario 4

In this scenario, eight user bases were defined UB1: R0, UB2/UB3: R2, UB4/UB5, UB6:R3, UB7:R4 and UB8:R5 depending on which Time zone they are. R0 represents North America, R2: Europe, R3: Asia, R4, Africa, R5: Pacific. Moreover, only 2 data centers were distributed in two different regions depending on number of users in each user base. DC1 in R0 and DC2 in R3. In each of those data centers contains 20 physical hardware units. As also it can be seen, that the data center are distributed where not in the center of all users, in North America and Asia. This distribution to see if it is important to allocate the data centre in the middle of the higher user bases Figure 4.9 shows the huge difference among the average response time of each VM load balancing algorithms. However, Table 4.6 illustrates the response time by each user base and the processing time by each data center.
### Figure 4.9: Scenario 4 results

The table below summarizes the performance metrics for Round Robin, Active Monitoring, and Throttled modes over differentUb (UB) and DC (DC) scenarios. The metrics include average (Avg), minimum (Min), and maximum (Max) values in milliseconds (ms).

<table>
<thead>
<tr>
<th></th>
<th>Round Robin</th>
<th></th>
<th></th>
<th>Active Monitoring</th>
<th></th>
<th></th>
<th>Throttled</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg (ms)</td>
<td>Min (ms)</td>
<td>Max (ms)</td>
<td>Avg (ms)</td>
<td>Min (ms)</td>
<td>Max (ms)</td>
<td>Avg (ms)</td>
<td>Min (ms)</td>
<td>Max (ms)</td>
</tr>
<tr>
<td>UB1</td>
<td>53.60</td>
<td>39.88</td>
<td>69.09</td>
<td>53.51</td>
<td>39.88</td>
<td>67.54</td>
<td>53.47</td>
<td>39.88</td>
<td>67.54</td>
</tr>
<tr>
<td>UB2</td>
<td>303.38</td>
<td>237.35</td>
<td>393.91</td>
<td>303.47</td>
<td>237.35</td>
<td>393.91</td>
<td>303.40</td>
<td>237.35</td>
<td>393.91</td>
</tr>
<tr>
<td>UB3</td>
<td>301.81</td>
<td>232.45</td>
<td>360.52</td>
<td>302.32</td>
<td>232.70</td>
<td>372.11</td>
<td>302.15</td>
<td>232.70</td>
<td>360.52</td>
</tr>
<tr>
<td>UB4</td>
<td>52.06</td>
<td>41.32</td>
<td>64.23</td>
<td>52.39</td>
<td>41.73</td>
<td>64.57</td>
<td>52.38</td>
<td>41.73</td>
<td>64.57</td>
</tr>
<tr>
<td>UB5</td>
<td>50.62</td>
<td>39.35</td>
<td>61.97</td>
<td>50.68</td>
<td>39.47</td>
<td>62.06</td>
<td>50.70</td>
<td>39.47</td>
<td>62.06</td>
</tr>
<tr>
<td>UB6</td>
<td>50.61</td>
<td>38.31</td>
<td>60.63</td>
<td>50.71</td>
<td>40.37</td>
<td>60.75</td>
<td>50.71</td>
<td>38.43</td>
<td>60.75</td>
</tr>
<tr>
<td>UB7</td>
<td>501.87</td>
<td>406.21</td>
<td>613.64</td>
<td>501.74</td>
<td>404.66</td>
<td>613.64</td>
<td>501.82</td>
<td>406.21</td>
<td>613.64</td>
</tr>
<tr>
<td>UB8</td>
<td>201.71</td>
<td>153.08</td>
<td>249.77</td>
<td>201.88</td>
<td>153.69</td>
<td>249.77</td>
<td>201.70</td>
<td>153.69</td>
<td>249.77</td>
</tr>
<tr>
<td>DC1</td>
<td>0.04</td>
<td>7.18</td>
<td>0.04</td>
<td>2.90</td>
<td>0.04</td>
<td>7.03</td>
<td>2.87</td>
<td>0.04</td>
<td>5.60</td>
</tr>
</tbody>
</table>
Table 4.6: Scenario 4 results

| DC2 | 0.07 | 7.71 | 0.07 | 1.71 | 0.07 | 4.09 | 1.71 | 0.07 | 4.09 |

1.29 Conclusion

Four cases as four simulations were conducted for each Virtual Machine load balancing algorithms (Round Robin, Active Monitoring, Throttled) to study the behavior of each one of these algorithms in a cloud that using dynamically configuration service broker under number of cases in terms of the distribution of data centers and their hardware units numbers.

The simulation was about deploying a large scale application on cloud. The investigator simulated Safeer system which is a large Saudi application that is used by all Saudi students who study overseas. The simulations were done by cloudAnalyst. CloudAnalyst generates reports with the results. This chapter presented the result that was generated from the simulator. However, The result’s evaluation will be discussed in the next chapter.
EVALUATION AND ANALYSIS

1.30 Introduction

After performing the simulations of the most three common VM load balancing algorithms, the result computed by cloud analyst as is shown within the previous chapter (Chapter 4). And the reason why this papers only focus on those three algorithms is because they are the most common and most of the researchers are interested in. The higher than outlined configuration has been used for every load reconciliation policy one by one and looking on that the result calculated for the metrics like over all response time and data center processing time has been shown. Parameters like average overall response time by each user base and datacenter service time have taken for analysis in this chapter.

1.31 Simulation Results Evaluation

After performing the simulations the result computed by cloud analyst as is shown within the previous chapter. Figure 5.1 below shows there is a significant decrease of each VM load balancing algorithms average response time from scenario 1 to scenario 3 and Scenario 4.

The overall response time of Round Robin algorithm and Active Monitoring load balancing algorithm is extremely high in scenario 1 comparing to the overall response time of the Throttled algorithm in the same scenario. Also, the figure shows the overall response time of Active monitoring load balancing algorithm and Throttled algorithm is way lower than the overall response time of Round robin algorithm. In addition, in scenario 1, it is very clear that active monitoring response time is higher than Round Robin response time. In contrast, in scenario 2, Active monitoring load balancing algorithm is lower than Round Robin algorithm response time.

In Scenario 3 and Scenario4, the response time of the three VM load balancing algorithms are almost the same, however, none of them has the same value of their response time.
To sum up, as it shown in Figure 5.1, the first scenario has the highest overall response time and scenario 3 and scenario 4 are almost have closest value of overall response time. however in between scenario 3 and scenario 4, scenario 3 is having the lowest value among those 4 scenarios. In addition, by taking deep insight in the results, we can see that Throttled load balancing algorithm has the lowest value of the overall response time among those three VM load balancing algorithms in all four scenarios.

meanwhile, the average data centre processing time in Figure 5.1, shows the similar results of Figure 5.2. it shows that scenario 3 is the best to have the lowest data centre processing time and it shows that throttled algorithm has also the smallest value among the other three balancer.

Figure 5.1: Results of the 4 scenarios of the overall response time
The figure 5.2 below shows the average data center processing time for each VM load balancing algorithm in each scenario which is the same of the overall response time in Figure 5.1.

![Figure 5.2: The overall data centre processing time for all 4 scenarios](image)

**1.32 Data Analysis**

After the simulations and studying its results, here some analytics. According to the result of scenario 1, it doesn’t matter if the application was deployed on number of datacenter, since the number of the physical hardware unit of each data center is really small. However, distributing the data center in regions where most of the user bases are, may help this situation a little bit, but still, the overall time is very high such as in scenario 2. What is really important is the number of physical hardware in each data center. So, even though, in scenario 3, the number of data centers that the application
was deployed on is only 2 data centers, the overall time was really reasonable and much better than the overall time in first two scenario. This is because each data centers based on a vast number of physical hardware units. In addition, the geographical location of each data centre is really vital. In scenario 4, one of the data centers was located in a region that most user requests were directed to another one. That’s why in scenario 4 the overall time was a slightly higher than scenario 3. Table 5.1 illustrates the differences in the criteria, the overall response time and the overall datacenter processing time of each scenario.

<table>
<thead>
<tr>
<th>Scenario’s number</th>
<th>Description</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
<th>Scenario 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4 data center in R0, R2, R3 and R5 In each of those data centers contains only 2 physical hardware units. Users are grouped by a factor of 100, and requests are grouped by a factor of 10, and each user request requires 25 instructions to be executed.</td>
<td>4 data center in R0, R2, R3 and R5 In each of those data centers contains only 2 physical hardware units. Users are grouped by a factor of 100, and requests are grouped by a factor of 100, and each user request requires 250 instructions to be executed.</td>
<td>2 data center in R0 and R2 In each of those data centers contains 20 physical hardware units. Users are grouped by a factor of 1000, and requests are grouped by a factor of 100, and each user request requires 250 instructions to be executed.</td>
<td>2 data center in R0 and R3. In each of those data centres contains 20 physical hardware units. Users are grouped by a factor of 1000, and requests are grouped by a factor of 100, and each user request requires 250 instructions to be executed.</td>
<td></td>
</tr>
<tr>
<td>VM load balancing algorithm</td>
<td>RR</td>
<td>AMLB</td>
<td>TLB</td>
<td>RR</td>
<td>AMLB</td>
</tr>
<tr>
<td>Average overall response time</td>
<td>18274.58</td>
<td>15547.87</td>
<td>15547.87</td>
<td>14789.06</td>
<td>8011.12</td>
</tr>
<tr>
<td>Average overall datacenter processing time</td>
<td>18197.87</td>
<td>15471.30</td>
<td>15471.30</td>
<td>14712.41</td>
<td>7934.48</td>
</tr>
</tbody>
</table>

Table 5.1: Summery of all 4 scenarios
From the table 5.1, it can be seen that by using Round Robin algorithm in deploying an application, the overall response time by each user bases can be reduced to 108 times. Moreover, the datacenter processing time can be decreased to 6 times, if the application be deployed on only 2 datacenter with 20 physical hardware which are located in two different regions where most of users are, instead of deploying the application in 4 datacenter contain only 2 physical hardware units in each one of them. However, the overall response time for each user base can be reduced to 118 times and the datacenter processing time can be decreased to 6 times, if the application also be deployed on 2 datacenter with also 20 physical hardware but are located in 2 different regions where are in the centre of all users bases.

Moreover, by using Active Monitoring load balancing algorithm in deploying an application, the overall response time by each user bases can be reduced to 181 times and the datacenter processing time can be decreased to 8 times, if the application be deployed on only 2 datacenter with 20 physical hardware which are located in two different regions where most of users are, instead of deploying the application in 4 datacenter contain only 2 physical hardware units in each one of them. However the overall response time for each user base can be reduced to 148 and the datacenter processing time can be decreased to 9 times, times and if the application also be deployed on 2 datacenter with also 20 physical hardware but are located in 2 different regions where are in the centre of all users bases.

What’s more, by using Throttled load balancing algorithm in deploying an application, the overall response time by each user bases can be reduced to 106 times and the datacenter processing time can be decreased to 334 times, if the application be deployed on only 2 datacenter with 20 physical hardware which are located in two different regions where most of users are. However the overall response time for each user base can be reduced to 6 times and the datacenter processing time can be decreased to 353 times, if the application also be deployed on 2 datacenter with also 20 physical hardware but are located in 2 different regions where are in the centre of all users bases.

All in all, by looking to the scenario 1 results, it is really not worth it that a clients pay for all those data centers and their small number of servers and then they get this kind of delay which is too much comparing to what they can get with other choices. In
addition, by looking to the second scenario results, it is figured out that number of servers inside each data centre is very important to improve the performance of large scale cloud that used dynamically reconfigure service broker.

what’s else, looking to the scenario 3 and scenario 4, by increasing the number of servers in each data centers that include high number of virtual machines is the best option for the clients to choose for deploying their large applications. however as I said distributing the data centers in a way that where ever the high number of users are, the data centre should be located in the centre of them.

Nevertheless, Throttled load balancing algorithm always has the lowest response time in any situation has been discussed above.

1.33 Addressing Research Questions

All 12 simulations showed that Throttled load balancing algorithm always has the lowest overall response time and data centre processing time. Furthermore, in order to improve the performance of large scale cloud application, the response time of each request should not be just reasonable but also very low so the user can be satisfied when he/she send a request and get the respond back quickly. In addition, according to the table 5.1, Throttled load balancing algorithm has the lowest value of the overall response time and the lowest data centre processing time, which make This algorithms the best to improve the performance of any large scale application on cloud. However, Throttled load balancing algorithm works in its best way when the application is deployed in two data centre or more which are located in geographical places where all user can access them. And those data centers contain good number of their physical hardware units.

1.34 Conclusions

This chapter evaluates the simulations results in order to address the research question. So, this chapter presents Throttled load balancing algorithm as the most suitable algorithm for deploying large scale application on cloud. What’s more, this chapter illustrates the reasons why Throttled load balancing algorithm is the best and how this algorithm be in is best behave. The reason why Throttled algorithm is the best among
the other two algorithms because in any situation that using dynamically configuration, this algorithms has the lowest value of the overall response time of each user base and the lowest value of the overall datacenter processing time. Also, throttled load balancing algorithm works in its best behavior when the data centers where the large application deployed are located in the centre of all the user bases. And each of these data centers have a good and reasonable number of the physical hardware units.
CONCLUSION

Chapter 6 revisits the objectives of this dissertation. The main findings that were found out during the research and experiment are outlined, and conclusions are discussed. Areas of supplemental research are described, particularly in relevance to this research topic. eventually, the contribution of this research is demonstrate as well.

1.35 Research Overview

The 21st century is an age of the internet; which is the driving forces of the aspects of communication and has turned the world into global village. The internet has been a great influence because the probability of all domains that is within human reach and control. Among the technological expansion on the web is cloud computing. The research study has focused on the key aspects of the cloud computing. As this section enables numerous access to people who make good use of both hardware and software infrastructure of the Internet. The researchers, Professionals as well as the entrepreneurs perceive the technology as a noteworthy improvement in statistics center that will influence ways in which millions of users acquire information (Shiau & Chau, 2016).

Advancement of cloud computing has helped in resolving the challenges faced when deploying web applications. This technology enables resource sharing over the internet using hardware access to virtual machines. Moreover, the cloud technology provides an abstraction of free virtualization technology, multi-tenancy, and web services. It is also used to provide a dynamically scalable infrastructure for application, file, and data storage. However, Virtualization in cloud technology improves the power efficiency of data centers by ensuring that different virtual machines can utilize the resources of a single physical server. Furthermore, The service broker policy is responsible for routing the requests of users originating from different locations to the data center on the cloud. On the contrary, the virtual machine load balancer is responsible for load balancing the user request coming from different locations. Therefore, The three main load balancing algorithms are Round robin Policy (RR),
Throttled Policy (TLB) and Active Monitoring Policy (AMLB). Those three algorithms are different in the manner they handle requests. So, number of simulations had be done on the three algorithms to determine the best virtual Machine load balancing algorithm that can improve the performance of large-scale applications based on cloud computing. And the reason why this papers only focus on those three algorithms is because they are the most common and most of the researchers are interested in.

1.36 Problem Definition

Cloud Computing is becoming one of the most popular technologies that taken by industry to provide an efficient and flexible way to retrieve and store data in applications or systems. However, the big issue is to organize the incoming request with minimum efficient resource utilization, a minimum time response and at the same time, resources shall not be under-utilized (Behal & Kumar, 2014). Therefore, The issue of load balancing is perhaps the greatest challenge facing this technology (Rima, Choi, & Lumb, 2009). It’s when resources are not allocated properly, there will be some servers that will be highly loaded while others are idle. This will lead to more energy consumption or more than that; it will not be possible to analyze the resource allocation in the cloud (Mohapatra et al., 2013). In other word, load balancing is used to the server workload among multiple computers or resources to create optimal utilization of resources, minimum response time, and short data processing time to avoid data overload (Rima, Choi, & Lumb, 2009).

There are many studies in terms of resources’ allocation by applying the most common load balancing algorithm, but none of them is about large-scale application using dynamic configuration service broker.

1.37 Design/Experimentation, Evaluation & Results

The investigator had to simulate a large application In order to compare the Virtual Machine load balancing algorithms which play the main role in the behavior of the application’s cloud. The simulation had been done four times for each VM load balancing algorithm (Round Robin, Active Monitoring, Throttled load balancing algorithm) to figure out what other factors that affect on a cloud as well. The
evaluation of the simulations results presents Throttled load balancing algorithm as the most suitable algorithm for deploying large scale application on cloud. What’s more, the research illustrated the reasons why Throttled load balancing algorithm is the best and how this algorithm be in is best behave. This algorithm is the best among the other two algorithms because in any situation, this algorithms has the lowest value of the overall response time of each user base and the lowest value of the overall datacenter processing time. Also, throttled load balancing algorithm works in its best behavior when the data centers where the large application e deployed are located in the centre of all the user bases. And each of these data centers have a good and reasonable number of the physical hardware units.

1.38 Contributions and impact

The current firms and industries have adopted cloud computing in the production of goods and services though some issue such as attendant consolidation, practical mechanism relocation, and consignment complementary that are up till now to be sufficiently addressed (Da Cunha, et al. pg.23). A key contribution of Virtualization is improving data center power efficiency, and makes all its virtual machines revolving to one physical server (Jain, Willke, Datta, & Yigitbasi, 2016). Also, the service broker policy is responsible for routing the requests of users originate from various locations around the world to the data center on a cloud. The data centers are also distributed in different geographical regions. While the Virtual machine loads, balancer is taking full responsibility for the load equilibrium in the user requirements in the middle of all the Virtual machines (Jermain, Rowlands, Buhrman & Ralph, 2016). That’s to say, my papers will give the cloud service providers a clear idea about using virtual machine load balancing algorithms with dynamically reconfiguring service broker in large scale application.

To sum up, the contribution and the impact of this papers as the following:

- One of the main contribution is the Virtualization which help improving the data center power efficiency in large scale of cloud, and makes all its virtual machines revolving to each physical server.
- The importance of the geographical locations of the data centres for the service broker policy
• Also, the importance of the Virtual machine load algorithms in organising the users requests to the available virtual machines in the servers of each data centers.

• Additionally to the studies I mentioned: AMLB is the best when the cloud uses Optimise Response Time broker, and TLB is the best while the cloud using Closest Data Centre. this papers will add to them that TLB is the best when the cloud uses Dynamically Reconfiguration Service Broker.

1.39 Future Work & recommendations

As the major goal of cloud computing is to offer services to the clients as their demand. Thence to get a preferable services, it can be concluded that the waiting status may occur in the coming days and in the future, as the cloud computing being the most beneficial computing domain these days. The issue of discovering the availability of Virtual Machines to the clients will improve and make the performance level of the cloud computing better. And after finding out the most efficient load balancing algorithm which is Throttled load balancing algorithm, that will be useful for allocation of effective Virtual Machines on demand.

Therefore, whereas caring about these issue leads to the ability of having a better service from the cloud computing. The modification of Throttled load balancing algorithm and proposing a new modified VM load balancing algorithm would help decreasing the overall response time even more and this modification will be carried out in the future work.

In conclusion, this papers recommend that the best way to use the dynamically reconfiguration service broker in large scale cloud is when the data centre is distributed in a way it is the centre location of all users, and considering that those data centers should have a good number of servers which are virtualized in also a good number of virtual machines. and to more improvement in the performance, it would say that using throttled load balancing algorithm will improve the cloud performance even more.
BIBLIOGRAPHY


Kumar, R., & Sahoo, G. (2014). Cloud Computing Simulation Using CloudSim. IJETT, 8(2), 82-86. doi:10.14445/22315381/ijett-v8p216


RituKapur —A Cost Effective approach for Resource Scheduling in Cloud ComputingIEEE International Conference on Computer, Communication and Control (IC4-2015).asemg,


APPENDIX

1.40 Cloud Analyst report of Scenario 1

1.40.1 Round Robin load balancing algorithm

<table>
<thead>
<tr>
<th>Overall Response Time Summary</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response time:</td>
<td>18274.58</td>
<td>35.39</td>
<td>1080052.17</td>
</tr>
<tr>
<td>Data Center processing time:</td>
<td>18197.87</td>
<td>0.14</td>
<td>1079994.55</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response Time by Region</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>61.36</td>
<td>35.39</td>
<td>121.39</td>
</tr>
<tr>
<td>UB2</td>
<td>225.33</td>
<td>38.03</td>
<td>621128.77</td>
</tr>
<tr>
<td>UB3</td>
<td>72460.41</td>
<td>36.65</td>
<td>1080052.17</td>
</tr>
<tr>
<td>UB4</td>
<td>325.06</td>
<td>226.32</td>
<td>626.46</td>
</tr>
<tr>
<td>UB5</td>
<td>72.85</td>
<td>38.23</td>
<td>102.53</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User Base Hourly Response Times</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UB1</strong> Response Time (ms)</td>
</tr>
<tr>
<td>0.0000</td>
</tr>
<tr>
<td>4.0000</td>
</tr>
<tr>
<td>20.0000</td>
</tr>
<tr>
<td>60.0000</td>
</tr>
<tr>
<td>60.0000</td>
</tr>
<tr>
<td>4.0000</td>
</tr>
<tr>
<td>7.0000</td>
</tr>
</tbody>
</table>

| **UB2** Response Time (ms)       |
| 0.0000                           |
| 4.0000                           |
| 30.0000                          |
| 60.0000                          |
| 60.0000                          |
| 4.0000                           |
| 7.0000                           |

| **UB3** Response Time (ms)       |

| **UB4** Response Time (ms)       |
UB5

Data Center Request Servicing Times

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC1</td>
<td>11.59</td>
<td>0.21</td>
<td>62.38</td>
</tr>
<tr>
<td>DC2</td>
<td>109.73</td>
<td>0.13</td>
<td>621077.02</td>
</tr>
<tr>
<td>DC3</td>
<td>72410.46</td>
<td>0.17</td>
<td>1079994.53</td>
</tr>
<tr>
<td>DC4</td>
<td>33.28</td>
<td>0.14</td>
<td>45.51</td>
</tr>
</tbody>
</table>

Data Center Hourly Average Processing Times

DC1

DC2
1.41 Active Monitoring load balancing algorithm:

Overall Response Time Summary

<table>
<thead>
<tr>
<th></th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response time</td>
<td>24818.22</td>
<td>35.42</td>
<td>1107049.60</td>
</tr>
<tr>
<td>Data Center processing time</td>
<td>24741.49</td>
<td>0.11</td>
<td>1106990.95</td>
</tr>
</tbody>
</table>

Response Time by Region

<table>
<thead>
<tr>
<th>Userbase</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>61.27</td>
<td>35.42</td>
<td>119.10</td>
</tr>
<tr>
<td>UB2</td>
<td>56.23</td>
<td>38.05</td>
<td>159.10</td>
</tr>
<tr>
<td>UB3</td>
<td>96851.04</td>
<td>37.61</td>
<td>1107049.60</td>
</tr>
<tr>
<td>UB4</td>
<td>301.77</td>
<td>226.33</td>
<td>388.77</td>
</tr>
<tr>
<td>UB5</td>
<td>51.31</td>
<td>37.80</td>
<td>71.32</td>
</tr>
</tbody>
</table>

User Base Hourly Response Times

UB1

Response Time (ms)

<table>
<thead>
<tr>
<th>0.0000</th>
<th>40.0000</th>
<th>20.0000</th>
<th>60.0000</th>
<th>80.0000</th>
<th>60.0000</th>
<th>40.0000</th>
<th>20.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>01.01</td>
<td>12.01</td>
<td>13.01</td>
<td>14.01</td>
<td>15.01</td>
<td>16.01</td>
<td>17.01</td>
<td>18.01</td>
</tr>
</tbody>
</table>

UB2

Response Time (ms)

<table>
<thead>
<tr>
<th>0.0000</th>
<th>40.0000</th>
<th>20.0000</th>
<th>60.0000</th>
<th>80.0000</th>
<th>60.0000</th>
<th>40.0000</th>
<th>20.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>01.01</td>
<td>12.01</td>
<td>13.01</td>
<td>14.01</td>
<td>15.01</td>
<td>16.01</td>
<td>17.01</td>
<td>18.01</td>
</tr>
</tbody>
</table>

UB3

UB4
Data Center Request Servicing Times

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC1</td>
<td>11.09</td>
<td>0.21</td>
<td>61.00</td>
</tr>
<tr>
<td>DC2</td>
<td>4.36</td>
<td>0.14</td>
<td>103.90</td>
</tr>
<tr>
<td>DC3</td>
<td>9860.03</td>
<td>0.18</td>
<td>110690.95</td>
</tr>
<tr>
<td>DC4</td>
<td>1.69</td>
<td>0.11</td>
<td>20.51</td>
</tr>
</tbody>
</table>

Data Center Hourly Average Processing Times

DC1

Processing Time (ms)

DC2
1.41.1 Throttled load balancing algorithm

### Overall Response Time Summary

<table>
<thead>
<tr>
<th></th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response time</td>
<td>15547.87</td>
<td>43.28</td>
<td>1107044.73</td>
</tr>
<tr>
<td>Data Center processing</td>
<td>15471.30</td>
<td>0.37</td>
<td>1106992.83</td>
</tr>
</tbody>
</table>

### Response Time by Region

<table>
<thead>
<tr>
<th>Userbase</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>86.69</td>
<td>47.09</td>
<td>146.56</td>
</tr>
<tr>
<td>UB2</td>
<td>71.93</td>
<td>45.33</td>
<td>92.13</td>
</tr>
<tr>
<td>UB3</td>
<td>61630.80</td>
<td>44.34</td>
<td>1107044.75</td>
</tr>
<tr>
<td>UB4</td>
<td>320.77</td>
<td>259.51</td>
<td>380.08</td>
</tr>
<tr>
<td>UB5</td>
<td>65.00</td>
<td>43.28</td>
<td>86.86</td>
</tr>
</tbody>
</table>

### User Base Hourly Response Times

- **UB1**
  - Response Time (ms)

- **UB2**
  - Response Time (ms)

- **UB3**

- **UB4**
Data Center Request Servicing Times

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC1</td>
<td>56.65</td>
<td>0.69</td>
<td>90.03</td>
</tr>
<tr>
<td>DC2</td>
<td>21.92</td>
<td>0.57</td>
<td>32.23</td>
</tr>
<tr>
<td>DC3</td>
<td>61580.82</td>
<td>1.03</td>
<td>1106992.83</td>
</tr>
<tr>
<td>DC4</td>
<td>15.28</td>
<td>0.64</td>
<td>32.02</td>
</tr>
</tbody>
</table>

Data Center Hourly Average Processing Times

DC1

<table>
<thead>
<tr>
<th>Processing Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9</td>
</tr>
</tbody>
</table>

DC2
Data Center Request Servicing Times

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC1</td>
<td>6.28</td>
<td>0.21</td>
<td>25.57</td>
</tr>
<tr>
<td>DC2</td>
<td>207.86</td>
<td>0.14</td>
<td>863994.59</td>
</tr>
<tr>
<td>DC3</td>
<td>3775.69</td>
<td>0.18</td>
<td>1079996.65</td>
</tr>
<tr>
<td>DC4</td>
<td>1.61</td>
<td>0.14</td>
<td>2.05</td>
</tr>
</tbody>
</table>

Data Center Hourly Average Processing Times

DC1

DC2
1.42 Cloud Analyst Report Scenario 2

1.42.1 Round Robin Load Balancing Algorithm
Data Center Request Servicing Times

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC1</td>
<td>8.07</td>
<td>0.44</td>
<td>19.17</td>
</tr>
<tr>
<td>DC2</td>
<td>15.79</td>
<td>0.23</td>
<td>184.26</td>
</tr>
<tr>
<td>DC3</td>
<td>58615.77</td>
<td>0.27</td>
<td>1075992.31</td>
</tr>
<tr>
<td>DC4</td>
<td>9.69</td>
<td>0.14</td>
<td>26.66</td>
</tr>
</tbody>
</table>

Data Center Hourly Average Processing Times

<table>
<thead>
<tr>
<th>Processing Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07</td>
</tr>
<tr>
<td>0.08</td>
</tr>
<tr>
<td>0.09</td>
</tr>
<tr>
<td>0.10</td>
</tr>
<tr>
<td>0.11</td>
</tr>
<tr>
<td>0.12</td>
</tr>
<tr>
<td>0.13</td>
</tr>
<tr>
<td>0.14</td>
</tr>
<tr>
<td>0.15</td>
</tr>
<tr>
<td>0.16</td>
</tr>
<tr>
<td>0.17</td>
</tr>
<tr>
<td>0.18</td>
</tr>
<tr>
<td>0.19</td>
</tr>
<tr>
<td>0.20</td>
</tr>
<tr>
<td>0.21</td>
</tr>
<tr>
<td>0.22</td>
</tr>
<tr>
<td>0.23</td>
</tr>
<tr>
<td>0.24</td>
</tr>
<tr>
<td>0.25</td>
</tr>
</tbody>
</table>

DC2
1.42.2 Active Monitoring Load Balancing Algorithm

### Overall Response Time Summary

<table>
<thead>
<tr>
<th></th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response</td>
<td>8011.12</td>
<td>42.15</td>
<td>990042.23</td>
</tr>
<tr>
<td>Data Center processing</td>
<td>7994.48</td>
<td>0.14</td>
<td>989993.03</td>
</tr>
</tbody>
</table>

### Response Time by Region

<table>
<thead>
<tr>
<th>Userbase</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>58.19</td>
<td>44.74</td>
<td>75.67</td>
</tr>
<tr>
<td>UB2</td>
<td>54.64</td>
<td>43.67</td>
<td>82.56</td>
</tr>
<tr>
<td>UB3</td>
<td>31658.80</td>
<td>42.72</td>
<td>990042.23</td>
</tr>
<tr>
<td>UB4</td>
<td>303.78</td>
<td>252.33</td>
<td>371.73</td>
</tr>
<tr>
<td>UB5</td>
<td>53.79</td>
<td>42.15</td>
<td>78.10</td>
</tr>
</tbody>
</table>

### User Base Hourly Response Times

#### UB1

![Response Time Chart for UB1]

#### UB2

![Response Time Chart for UB2]

#### UB3

#### UB4
Data Center Request Servicing Times

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC1</td>
<td>8.02</td>
<td>0.44</td>
<td>18.82</td>
</tr>
<tr>
<td>DC2</td>
<td>4.78</td>
<td>0.23</td>
<td>38.28</td>
</tr>
<tr>
<td>DC3</td>
<td>31608.78</td>
<td>0.28</td>
<td>999999.03</td>
</tr>
<tr>
<td>DC4</td>
<td>3.94</td>
<td>0.14</td>
<td>25.83</td>
</tr>
</tbody>
</table>

Data Center Hourly Average Processing Times

DC1

<table>
<thead>
<tr>
<th>Processing Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>90,000</td>
</tr>
<tr>
<td>80,000</td>
</tr>
<tr>
<td>70,000</td>
</tr>
<tr>
<td>60,000</td>
</tr>
<tr>
<td>50,000</td>
</tr>
<tr>
<td>40,000</td>
</tr>
<tr>
<td>30,000</td>
</tr>
<tr>
<td>20,000</td>
</tr>
<tr>
<td>10,000</td>
</tr>
</tbody>
</table>

DC2
1.42.3 Throttled Load Balancing Algorithm

---

Overall Response Time Summary

<table>
<thead>
<tr>
<th></th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response time</td>
<td>7088.21</td>
<td>42.15</td>
<td>1107044.75</td>
</tr>
<tr>
<td>Data Center processing time</td>
<td>7011.57</td>
<td>0.14</td>
<td>1106893.85</td>
</tr>
</tbody>
</table>

Response Time by Region

<table>
<thead>
<tr>
<th>Userbase</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>37.12</td>
<td>45.10</td>
<td>71.90</td>
</tr>
<tr>
<td>UB2</td>
<td>54.53</td>
<td>43.67</td>
<td>70.93</td>
</tr>
<tr>
<td>UB3</td>
<td>27982.06</td>
<td>42.73</td>
<td>1107044.75</td>
</tr>
<tr>
<td>UB4</td>
<td>303.62</td>
<td>250.90</td>
<td>371.73</td>
</tr>
<tr>
<td>UB5</td>
<td>53.14</td>
<td>42.15</td>
<td>66.46</td>
</tr>
</tbody>
</table>

User Base Hourly Response Times

UB1

UB2

UB3

UB4
Data Center Request Servicing Times

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC1</td>
<td>6.91</td>
<td>0.44</td>
<td>14.20</td>
</tr>
<tr>
<td>DC2</td>
<td>4.69</td>
<td>0.23</td>
<td>7.25</td>
</tr>
<tr>
<td>DC3</td>
<td>27932.05</td>
<td>0.28</td>
<td>1106993.85</td>
</tr>
<tr>
<td>DC4</td>
<td>3.34</td>
<td>0.14</td>
<td>6.95</td>
</tr>
</tbody>
</table>

Data Center Hourly Average Processing Times

DC1

DC2
1.43 Cloud Analyst Report of Scenario 3

1.43.1 Round Robin Load Balancing Algorithm

### Overall Response Time Summary

<table>
<thead>
<tr>
<th></th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response time</td>
<td>100.70</td>
<td>39.79</td>
<td>376.73</td>
</tr>
<tr>
<td>Data Center processing time</td>
<td>2.77</td>
<td>0.03</td>
<td>7.48</td>
</tr>
</tbody>
</table>

### Response Time by Region

<table>
<thead>
<tr>
<th>Userbase</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>53.63</td>
<td>39.88</td>
<td>68.86</td>
</tr>
<tr>
<td>UB2</td>
<td>52.42</td>
<td>39.79</td>
<td>67.60</td>
</tr>
<tr>
<td>UB3</td>
<td>52.12</td>
<td>39.83</td>
<td>64.57</td>
</tr>
<tr>
<td>UB4</td>
<td>301.95</td>
<td>237.58</td>
<td>376.73</td>
</tr>
<tr>
<td>UB5</td>
<td>299.89</td>
<td>233.10</td>
<td>369.47</td>
</tr>
<tr>
<td>UB6</td>
<td>300.30</td>
<td>237.89</td>
<td>368.03</td>
</tr>
<tr>
<td>UB7</td>
<td>301.81</td>
<td>228.66</td>
<td>369.64</td>
</tr>
<tr>
<td>UB8</td>
<td>201.59</td>
<td>153.08</td>
<td>249.14</td>
</tr>
</tbody>
</table>

### User Base Hourly Response Times

- **UB1**
  - Response Time (ms)
  - Graph showing response times over hours.

- **UB2**
  - Response Time (ms)
  - Graph showing response times over hours.
Data Center Request Servicing Times
1.43.2 Active Monitoring Load Balancing Algorithm

**Overall Response Time Summary**

<table>
<thead>
<tr>
<th></th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response</td>
<td>100.66</td>
<td>39.79</td>
<td>377.07</td>
</tr>
<tr>
<td>Data Center</td>
<td>2.79</td>
<td>0.03</td>
<td>7.03</td>
</tr>
</tbody>
</table>

**Response Time by Region**

<table>
<thead>
<tr>
<th>Userbase</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>53.52</td>
<td>39.88</td>
<td>67.54</td>
</tr>
<tr>
<td>UB2</td>
<td>52.40</td>
<td>39.79</td>
<td>64.60</td>
</tr>
<tr>
<td>UB3</td>
<td>52.44</td>
<td>40.08</td>
<td>64.57</td>
</tr>
<tr>
<td>UB4</td>
<td>302.27</td>
<td>237.99</td>
<td>377.07</td>
</tr>
<tr>
<td>UB5</td>
<td>300.08</td>
<td>233.22</td>
<td>369.57</td>
</tr>
<tr>
<td>UB6</td>
<td>300.40</td>
<td>237.89</td>
<td>368.14</td>
</tr>
<tr>
<td>UB7</td>
<td>301.88</td>
<td>228.66</td>
<td>369.64</td>
</tr>
<tr>
<td>UB8</td>
<td>201.87</td>
<td>153.69</td>
<td>249.77</td>
</tr>
</tbody>
</table>

**User Base Hourly Response Times**

- **UB1**
  - Response Time (ms)
  - Hours

- **UB2**
  - Response Time (ms)
  - Hours
Data Center Request Servicing Times
1.43.3 Throttled Loud Balancing Algorithm

Overall Response Time Summary

<table>
<thead>
<tr>
<th></th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response time:</td>
<td>100.64</td>
<td>39.79</td>
<td>377.07</td>
</tr>
<tr>
<td>Data Center processing time:</td>
<td>2.76</td>
<td>0.03</td>
<td>5.60</td>
</tr>
</tbody>
</table>

Response Time by Region

<table>
<thead>
<tr>
<th>Userbase</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>53.47</td>
<td>39.88</td>
<td>67.54</td>
</tr>
<tr>
<td>UB2</td>
<td>52.40</td>
<td>39.79</td>
<td>67.60</td>
</tr>
<tr>
<td>UB3</td>
<td>52.46</td>
<td>40.08</td>
<td>64.57</td>
</tr>
<tr>
<td>UB4</td>
<td>302.30</td>
<td>237.99</td>
<td>377.67</td>
</tr>
<tr>
<td>UB5</td>
<td>299.89</td>
<td>233.22</td>
<td>369.57</td>
</tr>
<tr>
<td>UB6</td>
<td>300.32</td>
<td>227.18</td>
<td>368.14</td>
</tr>
<tr>
<td>UB7</td>
<td>301.98</td>
<td>244.21</td>
<td>369.64</td>
</tr>
<tr>
<td>UB8</td>
<td>201.78</td>
<td>153.69</td>
<td>249.77</td>
</tr>
</tbody>
</table>

User Base Hourly Response Times

UB1

Response Time (ms)

UB2

Response Time (ms)
Data Center Request Servicing Times
1.44 Cloud Analyst Report of Scenario 4

1.44.1 Round Robin Load Balancing algorithm

Results of the Simulation Completed at: 08/12/2016 16:04:40

Overall Response Time Summary

<table>
<thead>
<tr>
<th></th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response time:</td>
<td>167.90</td>
<td>38.31</td>
<td>613.64</td>
</tr>
<tr>
<td>Data Center processing time:</td>
<td>2.84</td>
<td>0.04</td>
<td>7.71</td>
</tr>
</tbody>
</table>

Response Time by Region

<table>
<thead>
<tr>
<th>Userbase</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>53.60</td>
<td>39.88</td>
<td>69.09</td>
</tr>
<tr>
<td>UB2</td>
<td>303.38</td>
<td>237.35</td>
<td>393.91</td>
</tr>
<tr>
<td>UB3</td>
<td>301.81</td>
<td>232.45</td>
<td>360.52</td>
</tr>
<tr>
<td>UB4</td>
<td>52.06</td>
<td>41.32</td>
<td>64.23</td>
</tr>
<tr>
<td>UB5</td>
<td>50.62</td>
<td>39.35</td>
<td>61.97</td>
</tr>
<tr>
<td>UB6</td>
<td>50.61</td>
<td>38.31</td>
<td>60.63</td>
</tr>
<tr>
<td>UB7</td>
<td>501.87</td>
<td>406.21</td>
<td>613.64</td>
</tr>
<tr>
<td>UB8</td>
<td>201.71</td>
<td>153.08</td>
<td>249.77</td>
</tr>
</tbody>
</table>

User Base Hourly Response Times

UB1

UB2
1.44.2 Active Monitoring Load Balancing Algorithm
Data Center Request Servicing Times
1.44.3 Throttled Loud Balancing Algorithm

### Overall Response Time Summary

<table>
<thead>
<tr>
<th></th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response time</td>
<td>167.83</td>
<td>38.43</td>
<td>613.64</td>
</tr>
<tr>
<td>Data Center processing time</td>
<td>2.83</td>
<td>0.04</td>
<td>5.60</td>
</tr>
</tbody>
</table>

### Response Time by Region

<table>
<thead>
<tr>
<th>Userbase</th>
<th>Avg (ms)</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB1</td>
<td>53.47</td>
<td>30.88</td>
<td>67.54</td>
</tr>
<tr>
<td>UB2</td>
<td>303.40</td>
<td>237.35</td>
<td>393.91</td>
</tr>
<tr>
<td>UB3</td>
<td>302.13</td>
<td>232.70</td>
<td>360.32</td>
</tr>
<tr>
<td>UB4</td>
<td>52.38</td>
<td>41.73</td>
<td>64.57</td>
</tr>
<tr>
<td>UB5</td>
<td>50.70</td>
<td>30.47</td>
<td>62.06</td>
</tr>
<tr>
<td>UB6</td>
<td>50.71</td>
<td>38.43</td>
<td>60.73</td>
</tr>
<tr>
<td>UB7</td>
<td>501.82</td>
<td>406.21</td>
<td>613.64</td>
</tr>
<tr>
<td>UB8</td>
<td>201.70</td>
<td>153.69</td>
<td>249.77</td>
</tr>
</tbody>
</table>

### User Base Hourly Response Times

**UB1**

![Graph of UB1 response times](image)

**UB2**

![Graph of UB2 response times](image)
UB3

Response Time (ms)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 Hrs

UB4

Response Time (ms)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 Hrs

UB5

Response Time (ms)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 Hrs

UB6

Response Time (ms)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 Hrs

UB7

Response Time (ms)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 Hrs

UB8

Data Center Request Servicing Times
1.45 Screenshot of Safeer System

The figure below shows Student's home page that contains all functions they can perform.

1.46 Example of CloudSim

```
<table>
<thead>
<tr>
<th>Cloudlet ID</th>
<th>Status</th>
<th>Data center ID</th>
<th>VM ID</th>
<th>Time</th>
<th>Start Time</th>
<th>Finish Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SUCCESS</td>
<td>2</td>
<td>0</td>
<td>320</td>
<td>0.1</td>
<td>320.1</td>
</tr>
<tr>
<td>1</td>
<td>SUCCESS</td>
<td>2</td>
<td>4</td>
<td>320</td>
<td>0.1</td>
<td>320.1</td>
</tr>
<tr>
<td>2</td>
<td>SUCCESS</td>
<td>2</td>
<td>2</td>
<td>320</td>
<td>0.1</td>
<td>320.1</td>
</tr>
<tr>
<td>3</td>
<td>SUCCESS</td>
<td>3</td>
<td>1</td>
<td>320</td>
<td>0.1</td>
<td>320.1</td>
</tr>
<tr>
<td>4</td>
<td>SUCCESS</td>
<td>3</td>
<td>2</td>
<td>320</td>
<td>0.1</td>
<td>320.1</td>
</tr>
<tr>
<td>5</td>
<td>SUCCESS</td>
<td>3</td>
<td>3</td>
<td>320</td>
<td>0.1</td>
<td>320.1</td>
</tr>
</tbody>
</table>

org.cloudsSim.examples.CloudSimExample finished
```
**1.47 Online Survey Questions**

---

**Safeer Program**

---

This's a very simple survey _ For Safeer’s users only_ that won't take more than one minute. Thanks in advance for filling it.

1. How often do you use Safeer system?
   - Every day
   - Every week
   - Every Month

2. How many services do you request each time you log in?
   - 2-5 services
   - 6-10 services
   - Above 10 services

3. When do you usually use the system?
   - From 9:00 to 16:00
   - Other time

4. For the ones who use the system during SACM office hour, Do you check on your request later the day?
   - Yes
   - No

---

[Done]