EFFICIENT LOAD BALANCING IN CLOUD ENVIRONMENT WITH MODIFIED HONEY BEE ALGORITHM

Department of Computer Science and Engineering

Pabna University of Science and Technology,Pabna-6600

Course Title: Thesis

Course Code: CSE 4100 and CSE 4200

A thesis has been submitted to the Department of Computer Science and Engineering for the partial fulfillment of the requirement of Bachelor Degree in Computer Science and Engineering

Submitted By:

Sumaya Binte Habib

Roll Number: 150123

Registration Number: 101648, Session: 2014-15

Supervised By:
Subir Saha

Assistant Professor, Department of Computer Science andEngineering

Pabna University of Science and Technology

February, 2020

DECLARATION

In accordance with rules and regulations of Pabna University of Science and Technology following declarations are made:

I hereby declare that this thesis has been done by me under the supervision of Subir Saha, Assistant Professor, Department of Computer Science and Engineering, Pabna University of Science and Technology, Pabna-6600.

I also declare that neither this thesis nor any part of this thesis has been submitted elsewhere for awarding of any degree and any material reproduced in this thesis has been properly acknowledged.

Signature of the Examinee

CERTIFICATE

This is to certify that research work embodied in this thesis entitled “Efficient Load Balancing In Cloud Environment With Modified Honey Bee Algorithm” was carried out by Sumaya Binte Habib (Enrollment No.150123) at Pabna University of Science and Technology, Pabna-6600 is approved for the degree of Bachelor of Engineering with specialization in Computer Science & Engineering by Pabna University of Science Technology.

To the best of my knowledge, this paper has not been duplicated from any other paper or submitted to elsewhere prior submission to the department.

Subir Saha
Assistant Professor,
Department of Computer Science and Engineering
Pabna University of Science and Technology,Pabna-6600.

ACKNOWLEDGEMENT

First of all I would like to admit my gratefulness to the Almighty ALLAH for enabling me to perform this thesis successfully. I would like to express my deepest sense of gratitude to my honorable supervisor Subir Saha, Assistant Professor, Department of Computer Science and Engineering (CSE), Pabna University of Science Technology (PUST), for his scholastic supervision, valuable guidance, adequate encouragement and helpful discussion throughout the progress of this work. I am highly grateful to him for allowing me to pursuing this study under his supervision.

I am deeply thankful to honorable chairman, Md. Mahmudul Hasan, and all the respectable teachers of the Department of Computer Science and Engineering, Pabna University of Science Technology, Pabna-6600, Bangladesh, for their encouragement to my research work.

Finally, I am much grateful to my family members especially to my parents, all of my friends and well-wishers for their encouragement and supports.

February, 2020

Author

ABSTRACT

Cloud computing is an internet-based pool of heterogeneous resources where resources are available when required to online users which are provided as a service on demand with payment done on use basis. Load balancing is most important task of cloud computing to attain best machine utilization, tasks from overloaded virtual machines need to be transferred to under loaded virtual machines. Scheduling of resources are very massive problem on cloud. Scheduling of the models, cost, quality of service, time, and conditions of the request for access to services are factors is to be focused for cloud. This paper use the honey bee algorithm for load balancing in cloud to utilize its resources on cloud to cut back the makespan of processing time. With honey bee forage methodology, tasks are shifted from a overloaded Virtual machine to a VM which is idle. This saves the idle time of the process parts within the Virtual machine. The scheduling strategy was simulated using CloudSim tools. Experimental results indicated that the algorithm we have proposed is leveling work load which may reduce the span of processing and response time.

Keyword: Cloud Computing, CloudAnalyst, Honeybee Algorithm, Load Balancing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter we introduced our thesis overview, our work, motivation, objective and organization. In section 1.1 we discussed about thesis introduction; in section 1.2 we discussed about work; in section 1.3 we discussed about our thesis motivation; in section 1.4 we discussed about our thesis objective; in section 1.5 we discussed about the whole thesis paper organization; in section 1.6 we should give a short discussion about this chapter.

1.1 Introduction

Cloud computing is an entirely internet-based approach where all the applications and files are hosted on a cloud which consists of thousands of computers interlinked together in a complex manner. Cloud computing incorporates concepts of parallel and distributed computing to provide shared resources; hardware, software and information to computers or other devices on demand. These are emerging distributed systems which follows a "pay as you use" model. The customer need not buy the software or computation platforms. With internet facility, the customer can use the computation power or software resources by paying money only for the duration he/she has used the resource. Many cloud providers are available nowadays like amazon web services, google cloud, etc. Cloud provider provides many services to the user which is Reliable, Efficient and Low cost. Using virtualization technology, cloud data centers become more secure and flexible and provide better support for on-demand allocation.

1.2 Our Work

Different researchers have worked on Load Balancing in Cloud Computer in previous and find different kinds of algorithms for balancing the load. In this section we focused on what is cloud load balancing and the importance of it.

Cloud load balancing is the process of distributing workloads and computing resources in cloud computing environment. Load balancing allows enterprises to manage applications or workload demands by allocating resources among multiple computers, networks or services. In our paper we have worked on an algorithm for minimizing the work load in cloud environment which will be able to measure the overall response time. Our base algorithm is Honey Bee load balancing algorithm which we have modified for better work performance in cloud environment.

1.3 Motivation

Many of the previous works mainly focus on how balance the load in cloud environment dynamically. Mostly the paper showed a wonderful performance in balncing the load using their developed algorithm inspired from the base algorithm "Honey Bee Load Balancing Algorithm". We want to develop such a algorithm, for that case we take in concern the response time of virtual machines, datacenter processing time. We have shown our improvement result of our modified algorithm and also discussed how we can further improve our algorithm for better performance in future.

1.4 Thesis Objective

The objective of this paper is to propose a load balancing algorithm aims to distribute the dynamic workload smoothly to all the hosts in the cloud to gain an improvement in both the utilization of resources and the speed of execution time. It allocates the incoming tasks to all available VMs. In order to achieve fairness and avoid congestion, the proposed algorithm allocates tasks to the least loaded VM. This leads to a reduction of the overall response time and the processing time of hosts. In the proposed algorithm, variation of processing time of VM is the key limiting factor during the task allocation process because it avoids underutilization and over utilization of VMs.

The main objective of our proposed work is,

- To distribute the workload among server to main the load.
- To improve resource utilization, minimum completion time and improve Performance of system

.

1.5 Organization of the Thesis

In these section we discussed about the organization of the thesis.

This chapter (CHAPTER 1: Introduction) presents an overview of theback- ground of our work such as related work, motivation and our objective.

CHAPTER 2: Literature Review presents an overview of thesis literature, a clear concept of Cloud Computing, Load balancing, Different types of cloud services, About load balancing algorithms and previous works of others researcher in this field.

CHAPTER 3: System and Architecture represents has the contents about the problem statement, mathematical representation of the problem's solution. Moreover it also describes about our work flow and proposed method.

CHAPTER 4: This chapter is the most important chapter as it contains all important chapter as it contains all the details information about our work, work implementation process and how we have implemented it.

CHAPTER 5: In this chapter we have our output of work,we have shown validation of work and also compared our proposed algorithm with other base algorithms and also provide our decision in which way we have done a better job.

CHAPTER 6: Conclusion is the last chapter in this paper. These chapter represents a clear discussion about all the workflows with results analysis. Then a short description about the future work availability in these research fields. Finally a short conclusion presented as ending of our work.

1.6 Discussion

This is the introduction chapter and this chapter just introduces about our thesis, previous

related work done by different authors and our goals. Also its shows the blueprint of our work.

Chapter 2

Literature Review

In this chapter we introduced our thesis literature, Cloud Computing and load balancing description. In section 2.1 we discussed about literature introduction; in section 2.2 we discussed thesis literature as Cloud Computing, Load Balancing; in section 2.3 we discussed about different types of algorithms and benefits of using these algorithms in cloud load balancing process; in section 2.4 we should summarize the chapter.

2.1 Introduction

A thesis is a statement in a non-fiction or a fiction work that a writer intends to support and prove. To understand our thesis work, literature review is important. Our work is all about compute the overall response time of the virtual machines. Our thesis work "Efficient Load Balancing In Cloud Environment With Modified Honey Bee Algorithm" named. To understand these first we have to understand these concepts. We discussed about these literatures.

2.2 Cloud Computing

Our thesis named "Efficient Load Balancing In Cloud Environment With Modified Honey Bee Algorithm" is a work of developing a algorithm for balancing the load in cloud environment under the research area of Cloud Computing. In this section we should discuss about Cloud Computing, Load Balancing, Different types of Cloud Services, Load balancing methods.

2.2.1 Overview of Cloud Computing

Cloud computing is an entirely internet-based approach where all the applications and files are hosted on a cloud which consists of thousands of computers interlinked together in a complex manner. Cloud computing incorporates concepts of parallel and distributed computing to provide shared resources; hardware, software and information to computers or other devices on demand. These are emerging distributed systems which follows a "pay as you use" model. The customer need not buy the software or computation platforms. With internet facility, the customer can use the computation power or software resources by paying money only for the duration he/she has used the resource [1]. Many cloud

providers are available nowadays like amazon web services, google cloud, etc. Cloud provider provides many services to the user which is Reliable, Efficient and Low cost.

Using virtualization technology, cloud data centers become more secure and flexible and provide better support for on-demand allocation.

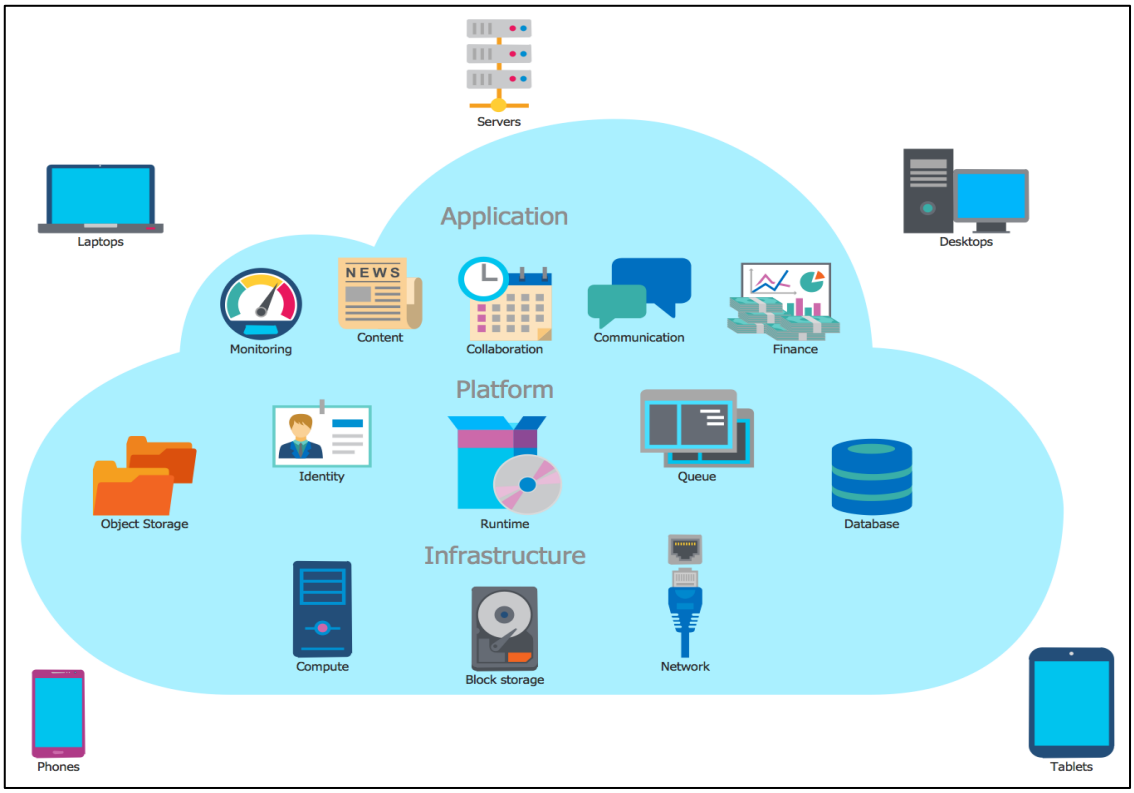

Figure 2. 1: Over View of Cloud Computing

2.2.1.1 Types of cloud computing deployment models

In cloud computing four types of deployment models are available-

1. Private Cloud: A private cloud is a cloud computing model which provides a cloud-based environment in which applications and resources can be access by the specified client only.

2. Public Cloud: It is a computing structure that is offered by the third party over the public internet which makes resources and application available for any user who wants to use them.

3. Community Cloud: A community cloud is an alliance effort which provides an infrastructure shared among many organizations that means the applications can be shared between them.

4. Hybrid Cloud: A hybrid clouds is a mix-up of two cloud that is a public and private cloud.

2.2.1.2 Types of Cloud service model

The Cloud provides three types of services model, these services modes are shown in the below figure:

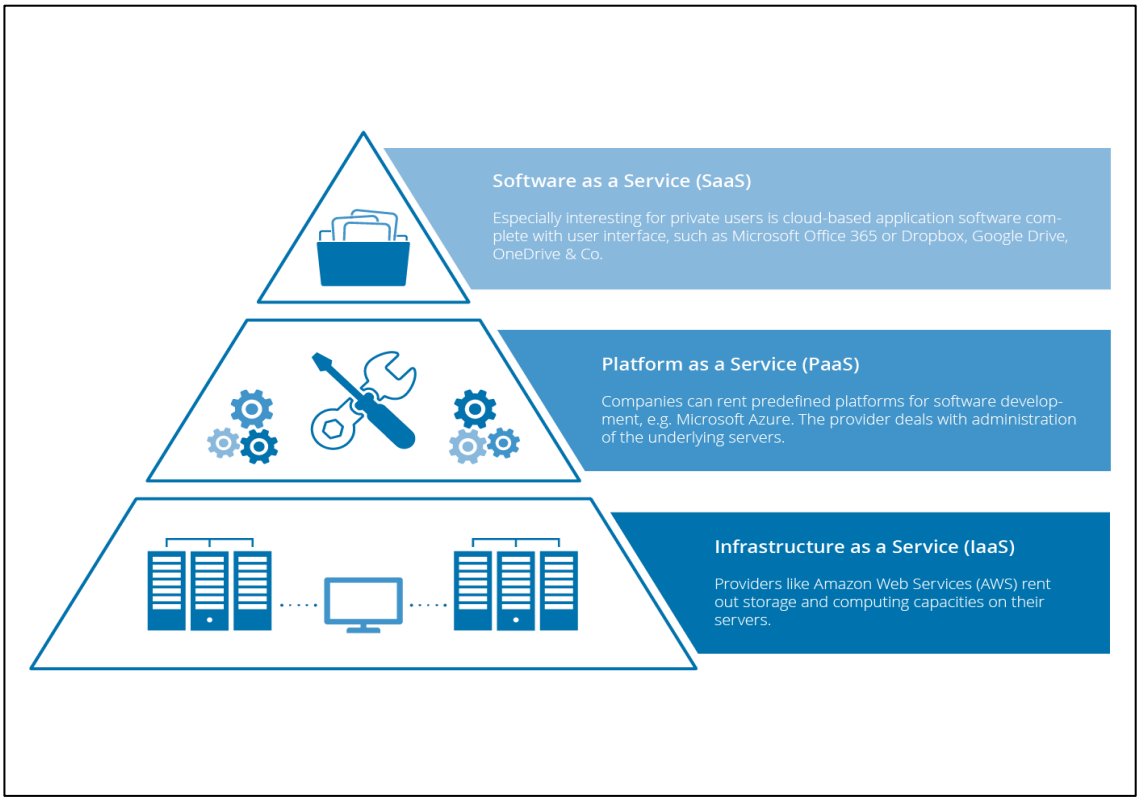

Figure 2. 2: Cloud Service Model

The models are described bellow-

Software-as-a-Service

SaaS model provides the next features:

- access to the software provided remotely via a thin client;
- payment for software usually include technical support;
- software deployment to the data center as a central core for using the app by all customers;
- a lot of users can launch application simultaneously;
- anyone can pay monthly or only for the services you actually use;
- the provider performs maintenance and update of the software.

Platform-as-a-Service

PaaS model provides the next features:

- this cloud pyramid layer allows developers to configure necessary resources for running apps, thus you can receive complete environment for software development;
- anyone can use automated scalability of resources and do not need to allocate them manually;
- anyone can manage the services he/she develops and the provider manages all the rest.

Infrastructure-as-a-Service

IaaS model provides the next features:

- anyone can rent all the tools (hardware and others) you need and use it how you need;
- anyone can receive administrator rights inside rented hardware;
- the MSP guarantees the availability of resources allocated to you;
- the provider manages all hardware issues.

2.2.1.3 Benefits of cloud computing

Cost Savings

Cost saving is the biggest benefit of cloud computing. It helps you to save substantial capital cost as it does not need any physical hardware investments. Also, you do not need trained personnel to maintain the hardware. The buying and managing of equipment is done by the cloud service provider.

Strategic edge

Cloud computing offers a competitive edge over your competitors. It helps you to access the latest and applications any time without spending your time and money on installations.

High Speed

Cloud computing allows you to deploy your service quickly in fewer clicks. This faster deployment allows you to get the resources required for your system within fewer minutes.

Back-up and restore data

Once the data is stored in a Cloud, it is easier to get the back-up and recovery of that, which is otherwise very time taking process on-premise.

Automatic Software Integration

In the cloud, software integration is something that occurs automatically. Therefore, you don't need to take additional efforts to customize and integrate your applications as per your preferences.

Reliability

Reliability is one of the biggest pluses of cloud computing. You can always get instantly updated about the changes.

Mobility

Employees who are working on the premises or at the remote locations can easily access all the cloud services. All they need is an Internet connectivity.

Unlimited storage capacity

The cloud offers almost limitless storage capacity. At any time you can quickly expand your storage capacity with very nominal monthly fees.

Collaboration

The cloud computing platform helps employees who are located in different geographies to collaborate in a highly convenient and secure manner.

Quick Deployment

Last but not least, cloud computing gives you the advantage of rapid deployment. So, when you decide to use the cloud, your entire system can be fully functional in very few minutes. Although, the amount of time taken depends on what kind of technologies are used in your business.

2.2.1.4 Open research issues in Cloud Computing

Cloud computing is the fastest growing technology. That is why so many researchers are doing research on this filed. Open Research issues in cloud computing are listed below:

- **Data Security**: Security is the greatest challenge or issue of cloud computing according to International Data Corporation (IDC). When we save our data or run our software

into others hard disk using others CPU appears to be very risky. Organization's data and software face serious risk of security issues like data loss, phishing, botnet etc.

- **Load Balancing:** The second major open research issues in cloud computing is load balancing. In this issue many research are work for maintaining a load in virtual machine. Load balancing means distributed the upcoming request among the multiple server to maintain the load.

- **Disaster Recovery:** Disaster is a suddenly accident for short time but its caused large amount of damage in society. Disaster Recovery issues are data centers down during disaster, Data Backup, Cost, Failure detection and security**.**

- **Availability of Service:** Availability refers to the ability of a user to access information or resources in a specified location and in the correct format. Simply availability mean services are available anytime and anywhere. The issues are protecting the confidentiality and integrity of data, Security and identify single point of failure.

2.3 Cloud Load Balancing

In this section we discussed about Cloud Load Balancing, Types of Load Balancing process, Goals of Load Balancing and Different types of load balancing algorithms.

2.3.1 Overview of Load Balancing

Load Balancing is a computer networking method to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload [2]. Thus load need to be distributed over the resources

in cloud-based architecture, so that each resources does approximately the equal amount of task at any point of time. Basic need is to provide some techniques to balance requests to provide the solution of the application faster. Cloud vendors are based on automatic load

balancing services, which allow clients to increase the number of CPUs or memories for their resources to scale with increased demands. This service is optional and depends on the clients business needs [3].

Load Balancing is one of the major issues related to cloud computing, the load may represent a CPU capacity, memory, network load etc. It is necessary to distribute the load equally among the nodes in a network. This results in agile and efficient performance of the system. Thereby it avoids heavily loading or under loading of nodes in a network [4].

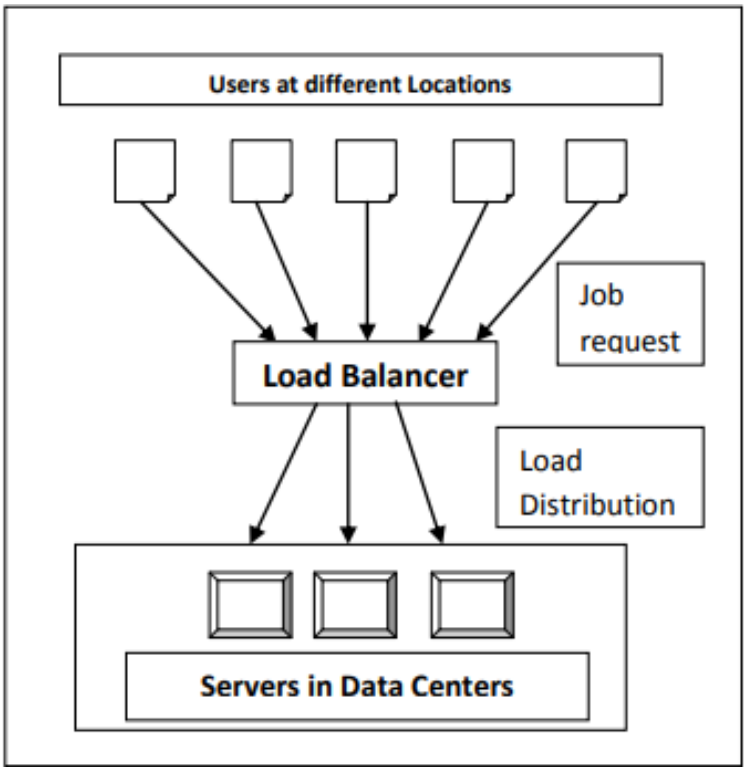

Figure 2.3. 2: Cloud Load Balancer

2.3.2 Metrics for Load Balancing in Cloud

Various metrics considered in existing load balancing techniques in cloud computing are discussed below [4] [2]-

- **Scalability** is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.
- **Resource Utilization** is used to check the utilization of re-sources. It should be optimized for an efficient load balancing.
- **Performance** is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.
- **Response Time** is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.
- **Overhead** Associated determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and interprocess communication. This should be minimized so that a load balancing technique can work efficiently.

2.3.4 Load Balancing on the basis of Cloud Environment

Cloud computing can have either static or dynamic environment based upon how developer configures the cloud demanded by the cloud provider [5].

Static Environment: In static environment the cloud provider installs homogeneous resources. Also the resources in the cloud are not flexible when environment is made static. In this scenario, the cloud requires prior knowledge of nodes capacity, processing power , memory, performance and statistics of user requirements. Round Robin algorithm provides load balancing in static environment [5].

Dynamic Environment: In dynamic environment the cloud provider installs heterogeneous resources. The resources are flexible in dynamic environment. In this scenario cloud cannot rely on the prior knowledge whereas it takes into account run-time statistics. The requirements of the users are granted flexibility (i.e. they may change at run-time). Algorithm proposed to achieve load balancing in dynamic environment can easily adapt to run time changes in load. Dynamic environment is difficult to be simulated but is highly adaptable with cloud computing environment**.** LBMM (Load Balancing Min-Min) algorithm uses for resource allocation in dynamic environment [5].

2.3.5 Goals of Load Balancing Algorithms

The goals of load balancing are to [6] [7]:

- For better performance.
- For achieving Stable System state.
- For construction of fault tolerant system.
- For accompanying further modifications.
- To have the ability to adjust itself in accordance with any modifications
- Manages resources efficiently.
- Utilizes each the systems resources as efficiently as possible.

2.3.6 Different types of Load Balancing Algorithms

- **Round Robin Algorithm:** In this algorithm [3] [2], the processes are divided between all processors. Each process is assigned to the processor in a round robin order. The process allocation order is maintained locally independent of the allocations from

remote processors. Though the work load distributions between processors are equal but the job processing time for different processes are not same. So at any point of time some nodes may be heavily loaded and others remain idle. This algorithm is mostly used in web servers where Http requests are of similar nature and distributed equally.

- **Equally Spread Current Execution Algorithm**: Equally spread current execution algorithm [9] process handle with priorities. it distribute the load randomly by checking the size and transfer the load to that virtual machine which is lightly loaded or handle that task easy and take less time , and give maximize throughput. It is spread spectrum technique in which the load balancer spread the load of the job in hand into multiple virtual machines [2].
- **Throttled Load Balancing Algorithm**: Throttled algorithm [2] is completely based on virtual machine. In this client first requesting the load balancer to check the right virtual machine which access that load easily and perform the operations which is give by the client or user. In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation.
- **BEE FORAGE BEHAVIOUR:** The artificial bee colony formula (ABC) algorithm supported the intelligent forage behavior of honey bee swarm and was planned by Karaboga in 2005. The formula is totally galvanized by natural forage behavior of honey bees [8].

2.3.7 Challenges in Cloud Computing Load Balancing

Here we discuss the challenges to be addressed when attempting to propose an optimal solution to the issue of load balancing in Cloud Computing [9]:

Spatial Distribution of the Cloud Nodes: Some algorithms are designed to be efficient only for an intranet or closely located nodes where communication delays are negligible.

However, it is a challenge to design a load balancing algorithm that can work for spatially distributed nodes.

Storage/ Replication: A full replication algorithm does not take efficient storage utilization into account. This is because the same data will be stored in all replication nodes. Full replication algorithms impose higher costs since more storage is needed.

Algorithm Complexity: Load balancing algorithms are preferred to be less complex in terms of implementation and operations. The higher implementation complexity would lead to a more complex process which could cause some negative performance issues.

Point of Failure: Controlling the load balancing and collecting data about the different nodes must be designed in a way that avoids having a single point of failure in the algorithm. Some algorithms (centralized algorithms) can provide efficient and effective mechanisms for solving the load balancing in a certain pattern. However, they have the issue of one controller for the whole system. In such cases, if the controller fails, then the whole system would fail. Any Load balancing algorithm must be designed in order to overcome this challenge.

2.4 Related Works

Load balancing mechanism distributes the work across multiple computing resources to utilize them effectively and at the same time eliminating a condition during which bound nodes are over loaded whereas others are beneath loaded.

Monika Rathore, Sarvesh Rai and Navdeep Saluja [8] have proposed an algorithm for Load Balancing of Virtual Machine Using Honey Bee Galvanizing Algorithm in Cloud. Their experimental results indicated that the mixture of the planned using honey bee forage behavior and scheduling supported the dimensions of tasks performed an scheduling strategy

in ever changing atmosphere and leveling work load which may reduce the span of processing time.

Abhishek Kumar Tiwari and Sreeja Nair has [4] improved the efficiency of load balancing in cloud environment using DAG and Honey Bee Algorithm. In this paper modified the load balancing policy based on teacher based learning optimization. The teacher based learning optimization well knows meta-heuristic function used for the purpose of optimization and searching process.

Arif Ullah, Nazri Mohd Nawi, Jamal Uddin, Samad Baseer and Ansam Hadi Rashed [10] have worked on "Artificial bee colony algorithm" in June 2019. Their review paper is a comprehensive study about load balancing in cloud computing using ABC algorithm. It also defines some basic concept about swarm intelligent and its property.

In 2016 Sakshi Dubey and Hari Mohan Singh [11] have published a research article named "Efficient Randomized Honeybee Algorithm for Allocation of Cloud Servers ". In this paper they have worked on releasing the servers from overloaded VMs and reallocate them to underutilized VMs. The simulation result shows that the proposed algorithm performs better than the honeybee algorithm, in case of arrival pattern is not predictable.

Prof. Shailendra Raghuvanshi and Priyanka Dubey [12] planned a load balancing algorithm for task scheduling based on resources segregation in cloud environment. They proposed an algorithm named honey bee behavior inspired load balancing, which aims to achieve well balanced load across virtual machines for maximizing the throughput. The proposed algorithm.

Chapter 3

System and Architecture

In this chapter we introduced our thesis Methodology, and Problem Statement. In section 3.2 we discussed about problem statement, and mathematical presentation of the problem; in section 3.3 we discussed about overview of our proposed work 3.4 .we should give a short discussion of this chapter.

3.1 Introduction

Load Balancing is the process to divide the workload among different available resource in cloud and equalize with virtual machine to achieve the performance of system. Sometimes several resources are heavy loaded, lightly loaded or idle for execution of tasks. There for task are removed from heavily loaded resources and rescheduled them to lightly loaded resources. We have proposed an modified algorithm for minimizing the overall response time.

3.2 Problem Statement

Until recently the main works on load balancing assumed solid nodes. Many instances of Cloud computing, as outlined herein, wherever dynamic and heterogeneous systems are necessary to produce on demand resources or services. The Amazon EC2, dynamic load balancing is handled by replicating instances of the precise middleware platform for internet services. This is often achieved through a traffic analyzer that tracks the time taken to method a shopper request. New instances of the platform are started once the load will increase on the far side predefined thresholds . Therefore, combos of rules impose the circumstances and answer for load balancing. Because the systems increase in size and quality, these rule sets become unwieldy and it should not be potential to take care of a viable observation and response cycle to manage the procedure work. In short, the dimensions of those systems could exceed the capabilities of connected meta systems to take care of a sufficiently agile and with efficiency organized load balancing (or general management) rule set. Once such a lot of management rules are outlined at intervals a system, there are probably to be conflicts amongst the rules; interactions and impact are normally terribly tough to research. A load balancing system is needed that self regulates the load at intervals the Cloud's entities while not essentially having to possess full information of the system. Such self organized regulation could also be delivered through distributed algorithms.

3.3 Proposed Approach

The artificial bee colony algorithm (ABC), an optimization algo-rithm based on the intelligent foraging behavior of honey bee swarm was proposed by Karaboga in 2005. Honey Bees are social insects where collective decisions are made via feedback cycles based on positive and negative signaling. This behavioral model is a powerful and tested model describing the foraging behavior of honey bees. This model is based on the behavioral structure of forager bee. An important means for communication among honey bees is through waggle dance, a dance which will give an idea to the waiting bees in the nest about a potential food source, its distance from the bee hive etc. In addition, honey bees use tremble and vibration dances also.

In our proposed work consider a minimum completion time of the task in each virtual machine. In existing honey bee algorithm only consider the virtual machine which has a minimum number of high priority task. In our modified honey bee behavior inspired algorithm, first we count the number of tasks and then sort them according their size in ascending order and then allocate them to vm and second is minimum completion time of task on each virtual machine.

3.3.1 Flow Diagram of Proposed Approach

Following figure 3.3 shows the flow diagram of the proposed algorithm. In this diagram, shows where the proposed methodology is used in cloud computing.

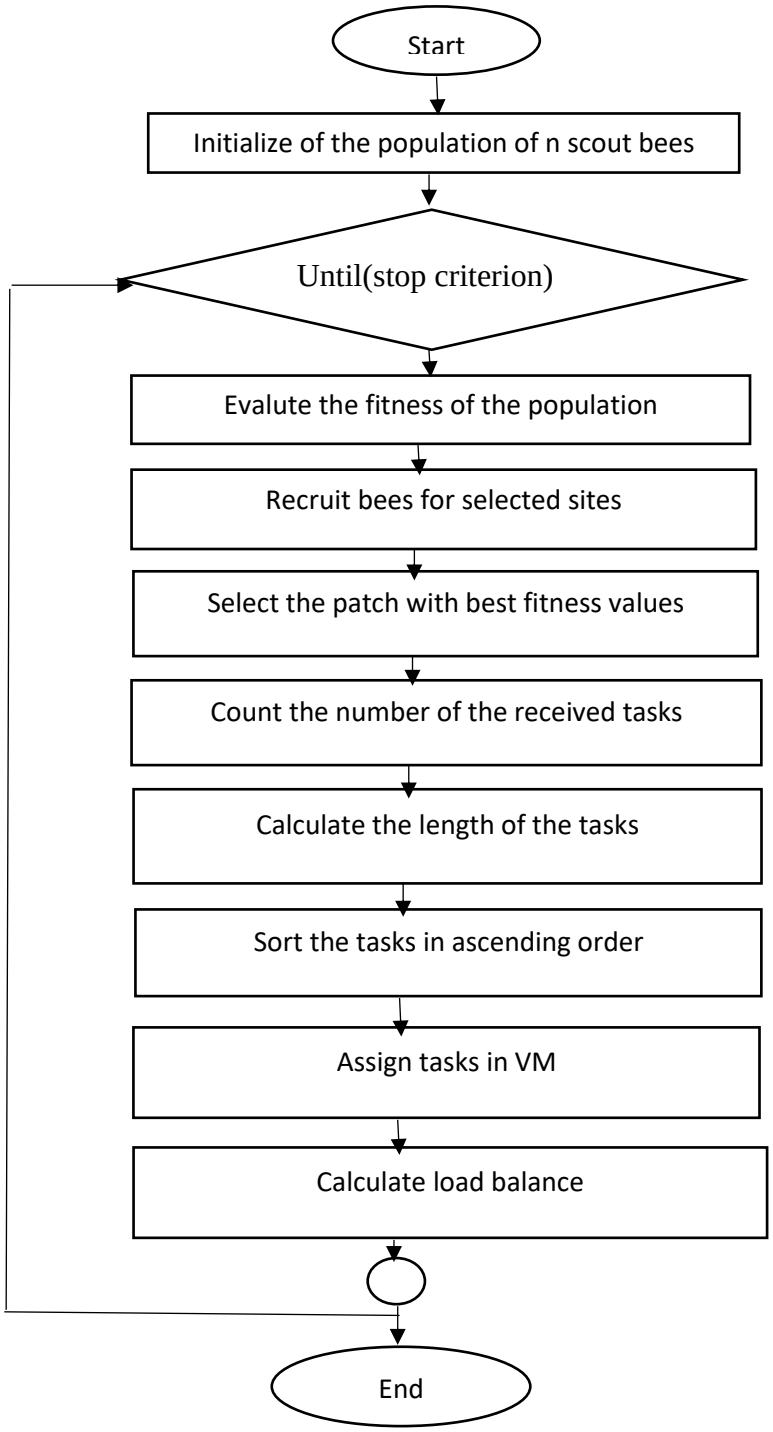

Figure 3.2. 3: Flow diagram of modified honey bee algorithm

BEE FORAGE BEHAVIOUR: The artificial bee colony formula (ABC) algorithm supported the intelligent forage behavior of honey bee swarm and was planned by Karaboga in 2005 . The formula is totally galvanized by natural forage behavior of honey bees.

INITIALIZATION METHOD: Artificial Bee Colony algorithm starts by correlating all the bees with created food sources. bound food sources are indiscriminately elect by bees and their nectar quantity is set. These bees return onto the hive and share the knowledge with bees waiting in dance space. Initialize the population of the scout bees, generate indiscriminately scout bees into the food sources and calculate the fitness values.

EMPLOYED BEE SECTION: Employed bees be the food supply and supply the neighborhood of the supply in its memory. When sharing the knowledge within the dance space, utilized bees attend food supply visited by its previous cycle and select new food supply by victimization the knowledge within the neighborhood. Then onlooker prefers a food supply counting on nectar info provided by utilized bees.

ONLOOKER BEE SECTION: Onlooker bees get the knowledge concerning food sources from utilized bees in hive and choose one in all the sources. These bee is anticipating a dance to decide on a food supply. Waggle/tremble/Vibration dances are performed by the bees to relinquish a concept concerning quality and amount of food and its distance from bee hive.

SCOUT BEE SECTION: Scout bee start random search. Once the nectar supply is abandoned by the bees, a brand new food supply is indiscriminately determined by a scout bee.

3.3.2 Pseudo Code of the Algorithm

The algorithm steps of our proposed algorithm are as below:

Algorithm: Modified Honey Bee behavior inspired load balancing algorithm
Step 1: every the scout bees search around the food sources and update the new fitness, if the new fitness is best than the previous values
Step 2: choose employee bees and recruit onlookers bees to go looking round the food sources and calculate on their fitness
Step 3: select the onlookers bees with have the most effective fitness worth
Step 4: calculate the task lengths and sort them according to their in ascending order
Step 5: allocate tasks in virtual machine $T\_l \rightarrow VM \mid \min(\sum T\_l)$ & Expected minimum completion time $T\_m \rightarrow VM \mid \min(\sum T\_m + \sum T\_l)$ & Expected minimum completion time $T\_h \rightarrow VM \mid \min(\sum T\_l)$ & Expected minimum completion time $CTij = Rj + ETij$ // Where, CT is completion time of machine ETij is expected execution time of task i on resource j Rj is ready time of resource j after completing previous assigned jobs Th, Tm, Tl are the task of high, middle and low length respectively
Step 6: Modify the number of the task as per newly added task to VM Modify the load for virtual machine both VMs and VM
Step 7: send scout bees into the food sources to get new food sources
Step 8: Until (Stopping criterion isn't met) End At the start, the initial scout bees are placed indiscriminately in VM on Cloud computing

Table 3.2. 4: Pseudo code modified for honey bee algorithm

3.3.3 Mathematical Representation

Let VM = {VM1, VM2, VM3,…,VMN } is a set of N virtual machines and Task= {task1,2, task3, … ,K} of K task to be regular and processed in VM.

We schedule non-preemptive tasks to VMs. Non-preemptive tasks means the tasks which can not be interrupt during compilation.

EVALUATE THE FITNESS OF THE POPULATION:

$$\text{fit}_{i,j}=\frac{\sum_{i=1}^{n} tasklength}{Evalute\ capacity\ of\ vm} \ldots\ldots\ldots\ldots\ldots (1)$$

Where fit i,j is that the fitness of the bees population of i in VM_j. tasklength is that the length of the task that has been submitted in VMj and capability is that the capability of VMj calculating supported the subsequent.

$$\text{Capacity} = pe\_num_j \times pe\_mips_j + vm\_bw_j \text{ ---- (2)}$$

Select m Sites for Neighborhood Search: Scout bees with the best fitness are chosen as choose Bee" and therefore the sites visiting by them are chosen from neighborhood of m VMs. Recruit Bees for elect Site: Send a lot of bees to neighborhood of the most effective VM, then judge the fitness supported.

$$\text{fit}_{i,j} = \frac{\sum_{i=1}^{n} tasklength + input\ file\ length}{Evalute\ capacity\ of\ vm} \ldots\ldots\ldots\ldots\ldots (3)$$

Where input file length is that the length of the task before execution. choose the most effective Fitness Bees from every Patch and Assign Task to Virtual Machine: for every spherical, the bee with the best fitness are chosen to assign task in Virtual Machine. Calculate Load Balance: when submitting tasks to the beneath loaded VMj, the present work of all offered VMs will be calculated by victimization the knowledge that received from the datacenter. Thus, variance (SD) is calculated so as to measure the deviations of load on VMs. variance of the load will be measured as-

$$SD=\sqrt{\frac{1}{n}\sum_{j=0}^{n}(Xj - X)} \ldots\ldots\ldots\ldots (4)$$

Processing time of VM:

$$X_j = \frac{\sum_{i=1}^{k} tasklength}{capacity} \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Mean of all processing times of all VMs:

$$X = \frac{\sum_{j=1}^{n} Xj}{n} \ldots\ldots\ldots\ldots\ldots\ldots (6)$$

If the S.D. of the loaded VM is a smaller amount than or capable the mean, then the system is during a balance state. On the opposite hand, if the S.D. is above the mean, then the system is in imbalance state.

3.3.4 Expected Outcome from the Work

Better utilization of resources and task take minimum completion time so that the performance of system will be improved. Our algorithm finding optimal least underloaded virtual machine for increasing the system performance.

3.4 Discussion

This chapter showed a clear description about the methodology we used for the various analysis of our thesis. Here also showed an overview of Analytical approach, and the Flow Diagram and pseudo code and details.

Chapter 4

Implementation

The most important chapter, mainly focused on how we have implemented our work,what tools we have used,how we have used that tools for implantation. In this chapter we have discussed about Cloud simulation tools, Eclipse and we have also discussed how we have used them in our implementation system for producing output and for further use. At last we have sumarised our whole chepter and our work in discussion portion.

4.1 Introduction

In this section learn some of the platform and simulation tools which can be used for implementing load balancing algorithm in a cloud computing environment. This tool is very useful for measuring the various QoS parameter like response time, waiting time, etc.

4.2 Design Constraints

Design and Build our proposed method, we used CloudSim simulation tool for analyzing the results of proposed method and compare it with the existing load balancing algorithm.

Software Requirement in Proposed Algorithm (minimum)

- Eclipse (version 3.5.1 and higher)
- Java Development Kit (JDK) 1.6 or Higher
- CloudSim Tool Kit
- Commons-math library

Hardware Requirement in Proposed Algorithm

- 1 GB RAM
- Intel Pentium Processor or higher

4.2.1 Eclipse

Eclipse is a free, java-based development platform knows for its plug-ins that allows developers to develop and test code written in other programming languages. Eclipse in

computer programming is an integrated development environment (IDE).It support the plug-in and workspace for customizing the environment. Development environment provided by Eclipse includes the Eclipse Java development tools (JDT) for Java, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others. Eclipse are used to run CloudSim simulation for implementing the cloud based project.

4.2.2 Java Development Kit

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) and other tools needed in Java development.

4.2.3 CloudSim Tool Kit

It is a stretchy framework for modeling and simulation of any application's performance in cloud environment on large scale. CloudSim is basically a library of cloud simulation scenarios. By considering policies of CloudSim, we can evaluate a method [15].

CloudSim framework includes [15]:

Regions: Regions are geographically division to provide resources on different locations.

Datacenter: It is a set of hosts or servers which provide infrastructure service. Datacenter configuration can be heterogeneous or homogeneous resources.

Hosts: Itis the physical entity that is resource to tasks.

Cloudlet: It is set of requests from user for computation. This class in CloudSim designs the application services like content delivery etc.

Service Broker: it decides which VM will provide service to request.

VM allocation: These policies in CloudSim model the allocation of resources to tasks.

VM scheduler: Scheduling policies of CloudSim simulate the resources allocated to cloudlet. CloudSim virtualized servers, hosts withcustom made policy for facility Virtual Machines.

The layered architecture of CloudSim is show in below figure:

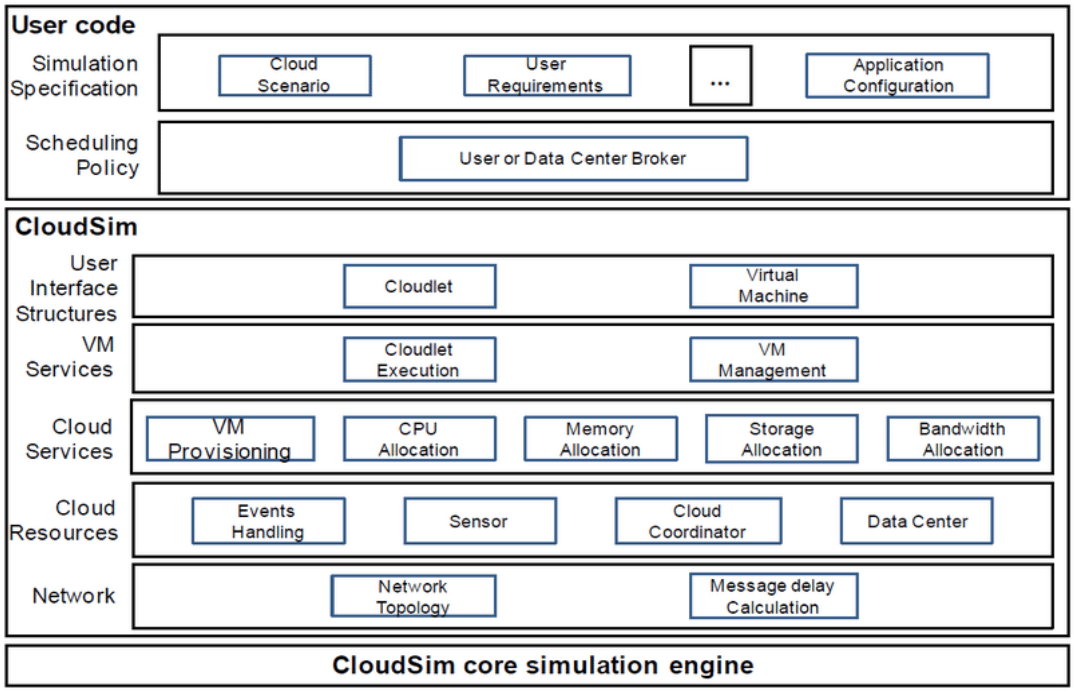

Figure 4.2.3: CloudSim Tool Kit

Figure 5.1 shows the multi-layered design of the CloudSim software framework and its architectural components. It support modeling and simulation of virtualized data center, memory, storage and bandwidth [15]. The CloudSim simulation advantages are:

• Flexibility of defining configurations.

• Ease of use and customization.

• Support for modeling and simulation of large scale Cloud computing data centers.

• Cost benefits.

• Modeling support in federating cloud.

Cloud Analyst

Cloud Analyst is a tool is at the University of Melbourne whose goal is to support evaluation of social networks tools according to geographic distribution of users and data centers. Such a tool separates the simulation experiment set up exercise from a programming exercise and enables a modeler to concentrate on the simulation parameters rather than the technicalities of programming.

4.3 Simulation Process

In this section we perform experimental process of cloud computing techniques with simulation tools. To interact with various services in the cloud and to maintain the resources in a balanced manner to fulfill the requirement of resources/infrastructure by those services, several techniques are required. To evaluate the performance of cloud computing techniques in cloud computing environment for the load balance and resource management, here we are using various numbers of techniques such as **Round Robin**, **Throttled** and **HoneyBEE** as a proposed method. For the further implementation and comparison for performance evaluation we used java programming languages with Eclipse IDE tools for complete implementation/results process.

Cloud Analyst is developed by Bhathiya Wickremasinghe at the CLOUDS Laboratory. It is built on top of CloudSim and separates the simulation experimentation from a programming task enabling one to concentrate on the imulation parameters rather than the technicalities of programming. Simulation in Cloud Analyst involves the following steps -

- Defining and configuration of User Bases.
- Defining and configuring Data Centers.
- Allocating of Virtual Machines in Data Centers.
- Review and Adjustment of various other parameters such as Packet size, No of

packets, Bandwidth, and Load balancing policies.

The Cloud Analyst enables us to model different scenarios of CSPs and User Bases, and provides a comprehensive output detailing the response time, Data Center processing time and total cost involved in the communication and computation.

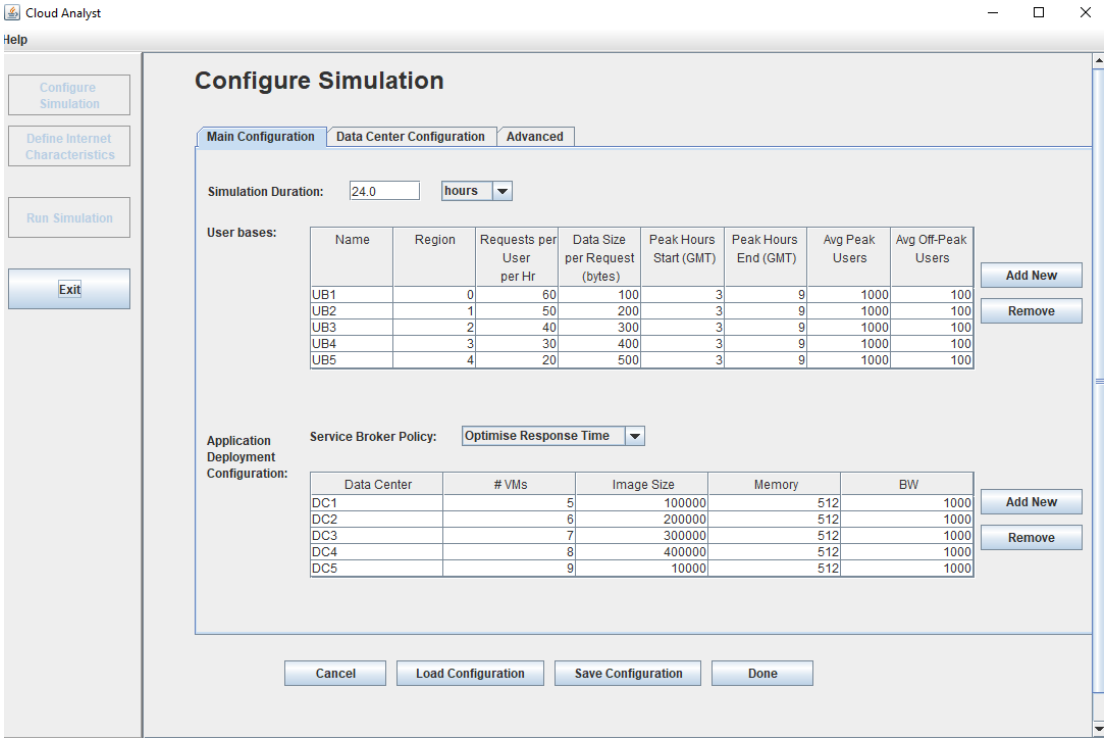

Figure 4.3.1: Simulation process for implementation of modified HoneyBee Algorithm

The above figure shows the simulation process of cloud analyst for measuring the developed algorithm's performance like- overall response time,datacenter processing time,cost for performing the whole simulation. Here we define the userbase, their number of request per hour, size of their data to be worked on. We can also modify user region and also edit the user properties, moreover add or remove users.

The next figure shows how to modify cloud datacenter in cloud analyst for further working process

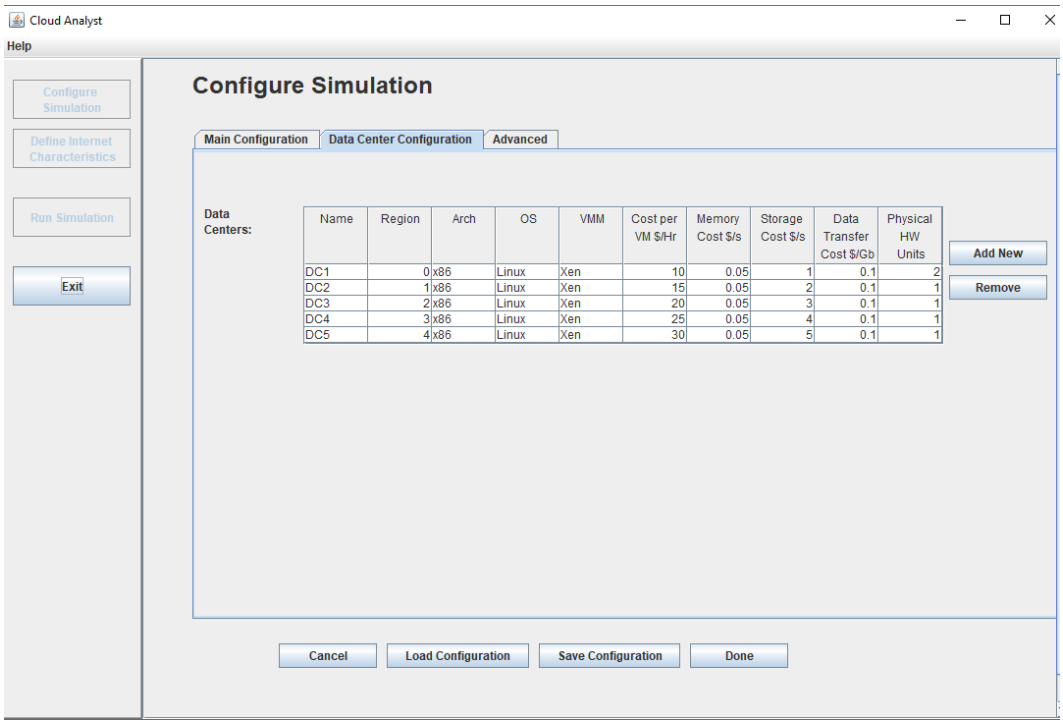

Figure 4.3. 4: Dtacenter Configuration in Cloud Analyst

The above figure shows us the configuration process of cloud datacenter. There are different types of prameters which we can configure like- OS, Cost per Vm, the region of the datacenter where they situated, memory cost, data transfer cost etc. These configuration will help us to know that how much cost will be needed to fulfill our desired work on cloud environment and other parameter also.

The next Figure is about running our own algorithm in virtual cloud environment i.e. Cloud Analyst .In this figure we have shown that how we have included our own algorithm in cloudanalyst and run our algorithm for finding out the result for the particular configuration we have done in cloudanalyst for comparing our algorithm with base algorithm.

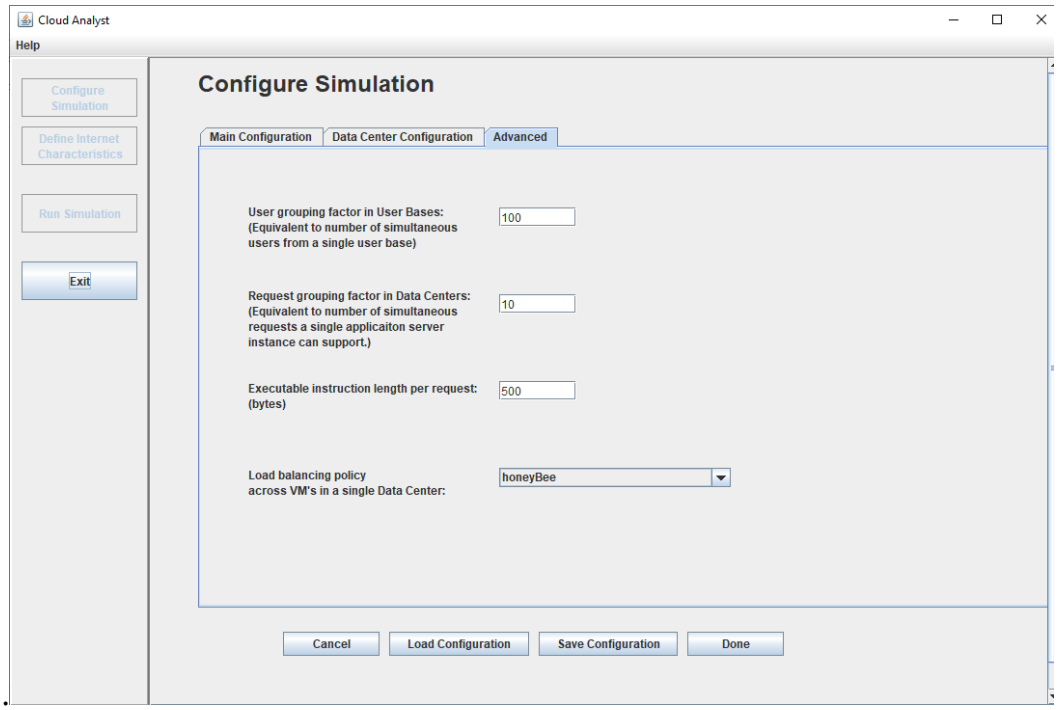

Figure 4.3. 5: Running our own algorithm in CloudAnalyst

In this above figure we have run the whole configuration we have previously done with modified honeybee algorithm for producing result.

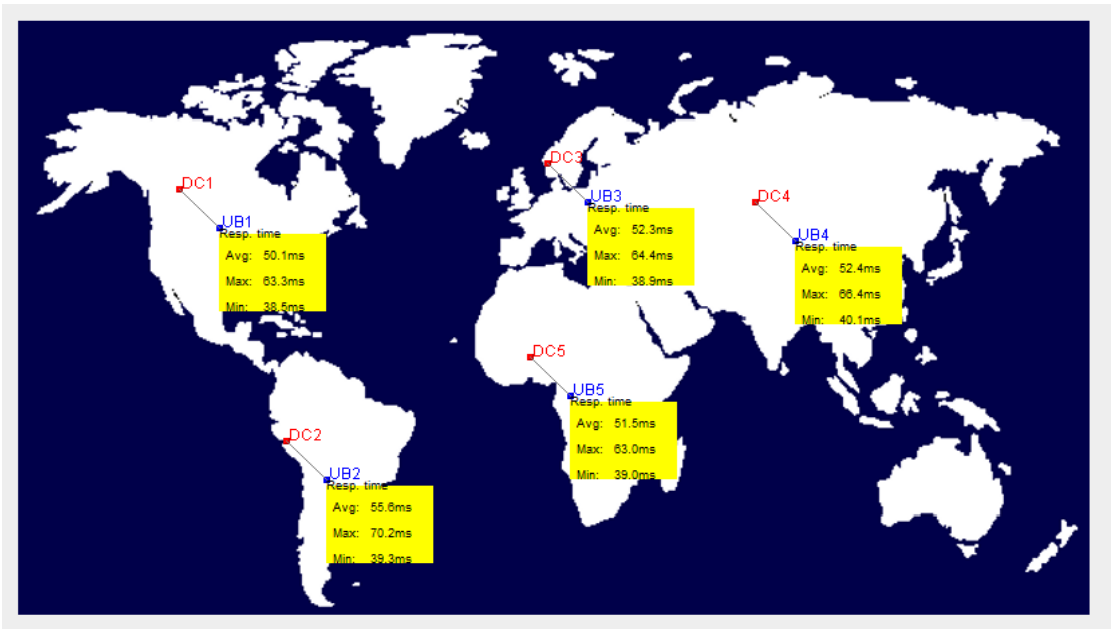

Figure 4.3. 6: Physical output of the configuration in CloudAnalyst

The above figure shows the physical output of our algorithm. The figure mainly shows that in which region we have our datacenter, in which region we have our users and the time for processing our data in particular datacenter.

Chapter 5

Result and Discussion

The most important chapter in which we have discussed about our proposed algorithm output, compare it with another algorithm output for measuring the performance of our algorithm if it is better than the other algorithm or not. At last we have given a short discussion about our work outcome.

5.1 Introduction

In this chapter we have discussed about the outcome of our proposed algorithm. We have also described the other base algorithm output so that we can compare our proposed algorithm output with these outputs and give further explanation why we have done modification in our algorithm and how it is better than other algorithms.

5.2 Result of Proposed Algorithm

At first we have include our modified algorithm in the cloud simulation tool and then run the simulation tool with the configuration and produce output. The result of the developed algorithm output is shown in following table:-

Time	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	52.43	38.53	70.21
Data Center processing time:	2.65	0.01	15.14

Table 5. 2: Output of our proposed honeyBee algorithm

The above table shows the result of our proposed algorithm after running it on cloud simulation tool. It shows the overall response time of the virtual machines and also shows the datacenter processing time.

The cost value of processing the data on vm and datacenter are given bellow:-

Total Virtual Machine Cost ($): 1801.52

Total Data Transfer Cost ($): 37.39

Grand Total: ($) 1838.91

5.3 Result of RoundRobin Algorithm

In this section we will describe about other algorithm output after running them on the cloud analyst. For comparing our algorithm performance with other algorithms we have taken in concern other two algorithms named RoundRobin and Throtolled algorithm. Here we will show the output of RoundRobin algorithm using the same cloud analyst configuration which we have used for our own algorithm implementation.

Time	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	70.11	37.56	220.54
Data Center processing time:	20.36	0.00	62.32

Table 5. 2: Result of RounRobin Algorithm after simulation

The cost value of processing the data on vm and datacenter are given bellow:-

Total Virtual Machine Cost ($): 2640.28

Total Data Transfer Cost ($): 44.86

Grand Total: ($) 2685.13

5.4 Result of Throtolled Algorithm

Here we will show the output of Throtolled algorithm using the same cloud analyst configuration which we have used for our own algorithm implementation

Time	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	52.78	36.91	79.48
Data Center processing time:	3	0.01	23.39

Table 5. 3: Output of the Throtolled Algorithm

The cost value of processing the data on vm and datacenter are given bellow:-

Total Virtual Machine Cost ($): 8640.91

Total Data Transfer Cost ($): 44.86

Grand Total: ($) 8685.76

5.5 Result Comparison

For analyzing the result of the base algorithm and our proposed algorithm we can take a two parameter first is overall response time and second is datacenter processing time. The main objective of our proposed work is to increases the utilization of cloud resources and minimum completion time of task for improving the performance of the system. Following figures shows the performance analysis of base load balancing algorithm and our proposed algorithm. In below figure overall processing time and datacenter processing time are analyzed. We have also analysed the cost for processing our data using every algorithm results we have produced in the previous section.

For analyzing the result we have used bar chart representation so that we can analyze the algorithms performance very easily. The figures are given bellow for comparison-

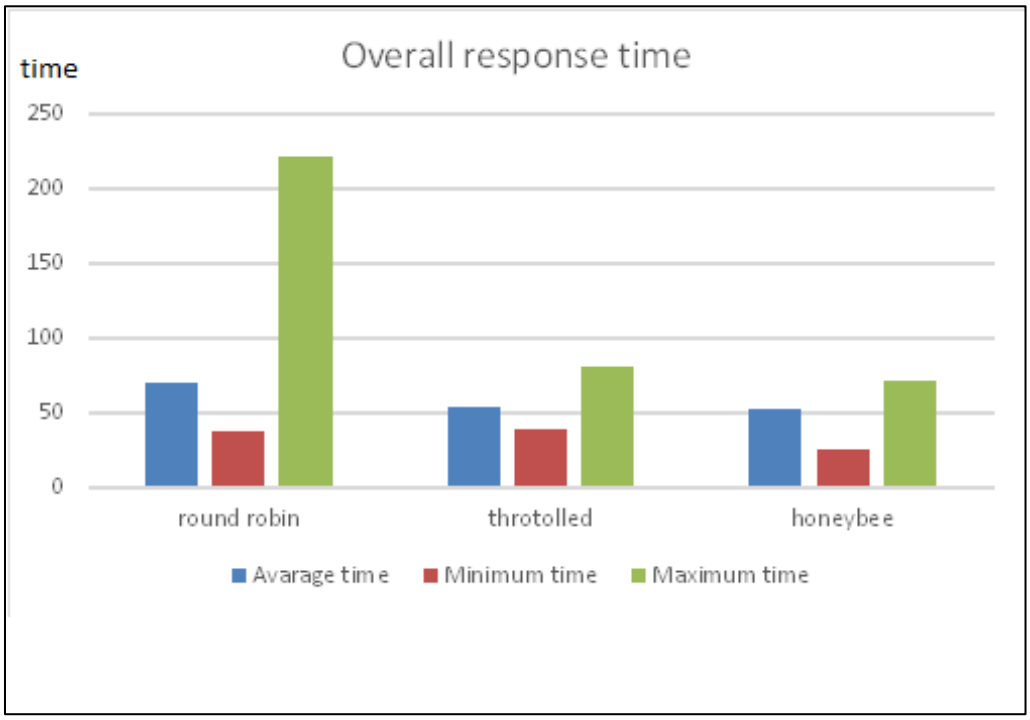

Figure 5. 5: Resultant graph for three algorithm for overall response time using 5 VM

The above figure is the comparision between three algorithms and they are- roundrobin, hrotolled and modified honeybee algorithm. The parameter of the comparision is overall response time. From the figure we can see that, the maximum response time of the honeybee algorithm is the lowest than the others. Again, the average and minimum time values are also lower for our algorithm than others. From this,we can say that the performance of load allocation using our algorithm is better.

Now,we will compare the datacenter processing time of all three algorithms with bar diagram and the diagram is given bellow :-

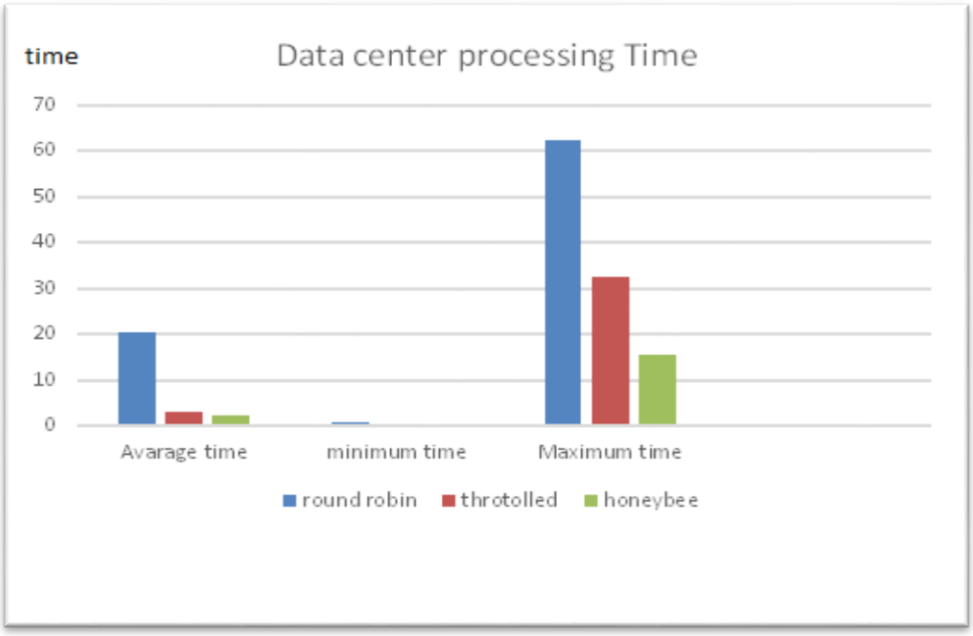

Figure 5.6: Shows the datacenter processing time comparision

The above figure is the comparision between three algorithms and they are- roundrobin, hrotolled and modified honeybee algorithm. The parameter of the comparision is datacenter processing time. From the figure we can see that, the maximum response time of the honeybee algorithm is the lowest than the others. Again, the average and minimum time values are also lower for our algorithm than others. From this,we can say that the performance of load allocation using our algorithm is better.

The cost efficiency can be measured from the following table:-

cost	Proposed algorithm	RoundRobin algorithm	Throtolled algorithm
Total Virtual Machine Cost ($)	1801.52	2640.30	8640.91
Total Data Transfer Cost ($):	36.39	40.86	44.86
Grand Total: ($)	1838.91	2680.30	8685.91

Table5. 5: Comparison of cost for processing cloudlets using different algorithms

5.6 Discussion

From the above figures and tables we can clearly say that, the modified honeybee algorithm which we have developed gives better performance than the any other algorithms which we have used for comparison. That is why we can say that, the developed honeybee algorithm will be more efficient while scheduling tasks to the virtual machines as well as cost effective. Because the, cost comparision table shows that the grand total of all transference of data is lower of honeybee algorithm than the others.

Chapter 6

Conclusions

We should summarize the problem in this chapter. In section 6.1 we should give a clear discussion about our thesis and results. In section 6.2 we discussed about the future work opportunities. In section 6.3 we should conclude the whole work in a summary.

6.1 Conclusions

Cloud computing is a recent advancement in information technology. It provides various 'cloud-based services' to end-users on the basis of usage-based payment model. The aim of the cloud service provider is to save time, save money as well as access cloud service from anywhere anytime and from any device. It provides the number of benefits, so the user of cloud services are increased nowadays. So there is a need to handle many user requests, for that the load balancing techniques are used in cloud computing. This paper presents implementation formula which can resolve the Virtual machine programming management at a lower place the dynamic atmosphere of the amount of VMs and requests on Cloud computing. Honey bee behavior load balancing, improves the general turnout of process and priority based balancing focuses on reducing the makespan , time a task must help a queue of the VM. Thus, it reduces the response of time of VMs. The experimental results show that the formula is effective when put next with other existing algorithms. Our approach illustrates that there's a big improvement in average execution time and reduction in waiting time of tasks on queue. And by this way, this proposed work is developed for achieving better resource utilization, minimum completion time and improve the system of performance in a cloud computing environment.

6.2 Future Work

In our proposed work consider a minimum completion time of the task in each virtual machine. In existing honey bee algorithm only consider the virtual machine which has a minimum number of high priority task. In our modified honey bee behavior inspired algorithm, first we count the number of tasks and then sort them according their size in

ascending order and then allocate them to vm and second is minimum completion time of task on each virtual machin

Appendix A

Vm	Virtual machine
DC	Data center
T_l	Task Length
Qos	Quality of service
Avg	Average
Min	Minimum
Max	Maximum

Reference

[1] P. V. K. Dhinesh Babu L.D., "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Applied Soft Computing,* pp. 2292-2303, 2013.

[2] S. K. Amandeep Kaur Sidhu, "Analysis of Load Balancing Techniques in Cloud Computing," *International Journal of Computers & Technology ,* vol. 4, no. 2, pp. 2277-3061 , 2013.

[3] A. D. S. Soumya Ray, "EXECUTION ANALYSIS OF LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING ENVIRONMENT," *International Journal on Cloud Computing: Services and Architecture (IJCCSA,* vol. 2, no. 5, 2012.

[4] S. N. Abhishek Kumar Tiwari, "Improve the Efficiency of Load Balancing in Cloud Environment using DAG and Honey Bee Algorithm," *International Journal of Computer Science and Information Technologies,* vol. 7, no. 2, pp. 811-815, 2016.

[5] A. M. Mayanka Katyal, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment," *International Journal of Distributed and Cloud Computing ,* vol. 1, no. 2, 2013.

[6] T. V. Gill Sukhjinder Singh, "IMPLEMENTATION OF A HYBRID LOAD BALANCING ALGORITHM FOR CLOUD COMPUTING," *International Journal of Advanced Technology in Engineering and Science,* vol. 3, no. 1, 2015.

[7] N. S. G. S. Akash Saxena, "A Study on Benefits and Classification of Load Balancing in Cloud Computing Environment," *International Conference on Internet of Things and Connected Technologies (ICIoTCT),* 2018.

[8] S. R. N. S. Monika Rathore, "Load Balancing of Virtual Machine Using Honey Bee Galvanizing Algorithm in Cloud," *International Journal of Computer Science and Information Technologies,* vol. 6, no. 4, 2015.

[9] N. M. M. A. N. A.-J. Klaithem Al Nuaimi, "A Survey of Load Balancing in Cloud Computing:," *IEEE Second Symposium on Network Cloud Computing and Applications,* 2012.

[10] N. M. N. J. U. S. B. A. H. R. Arif Ullah, "Artificial bee colony algorithm used for load balancing in cloud computing: review," *IAES International Journal of Artificial Intelligence* , vol. 8, no. 2, 2019.

[11] H. M. S. Sakshi Dubey, "Efficient Randomized Honeybee Algorithm for Allocation of Cloud Servers," *International Journal of Engineering Science and Computing,* vol. 6, no. 7, 2016.

[12] P. D. Prof. Shailendra Raghuvanshi, "Task Scheduling Algorithm based on Resources Segregation in Cloud Environment," *International Journal of Scientific Research in Science, Engineering and Technology* , vol. 4, no. 11, 2018.

[13] H. K. Ekta Rani, "STUDY ON FUNDAMENTAL USAGE OF CLOUDSIM SIMULATOR AND ALGORITHMS OF RESOURCE ALLOCATION IN CLOUD COMPUTING," *IEEE*, 2017.

[14] H. N. a. R. R. Walaa Hashem, "Honey Bee Based Load Balancing in Cloud Computing," *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS* , vol. 11, no. 12, 2017.

[15] B. K. Anureet kaur, "Load Balancing in tasks using Honey bee Behavior Algorithm in Cloud Computing," *IEEE,* 2016.