2020-02

Hybrid Load Balancing Algorithm In Cloud Computing

Gardie, Birhanu

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND POSTGRADUATE STUDIES

FACULTY OF COMPUTING

HYBRID LOAD BALANCING ALGORITHM IN CLOUD COMPUTING

By

Birhanu Gardie

Bahir Dar, Ethiopia

February 2020

HYBRID LOAD BALANCING ALGORITHM IN CLOUD COMPUTING

BIRHANU GARDIE

A thesis submitted to the school of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in partial fulfillment of the requirements for the degree of
Master of Science in Information Technology in the Faculty of Computing.

Advisor: Mekuanint Agegnehu (PhD)

Bahir Dar, Ethiopia
February 24, 2020

DECLARATION

I, the undersigned, declare that the thesis comprises my own work. In compliance with internationally accepted practices, I have acknowledged and refereed all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation/ fabrication of any idea/data/fact/source will constitute sufficient ground for disciplinary action by the University and can also evoke penal action from the sources which have not been properly cited or acknowledged.

Name of the student Birhanu Gardie Signature

Date of submission: 24/2/2020

Place: Bahir Dar

This thesis has been submitted for examination with my approval as a university advisor.

Advisor Name: Mekonnint A. (PhD)

Advisor's Signature:

Bahir Dar University

Bahir Dar Institute of Technology-

School of Research and Graduate Studies

Faculty of Computing

THESIS APPROVAL SHEET

Student:

Birhanu Gardie		21/02/2020
Name	Signature	Date

The following graduate faculty members certify that this student has successfully presented the necessary written final thesis and oral presentation for partial fulfillment of the thesis requirements for the Degree of Master of Science in *Information Technology*.

Approved By:

Advisor:

Mekuanint A (PhD)		Feb 21, 2020
Name	Signature	Date

External Examiner:

Sosina Mengistu (Ph.D.)		21/02/2020
Name	Signature	Date

Internal Examiner:

Gebeyehu B (Dr)		21/02/20
Name	Signature	Date

Chair Holder:

Dereje W Lake		21/02/20
Name	Signature	Date

Faculty Dean:

Belete B.		21/02/20
Name	Signature	Date

Acknowledgment

First, to God who makes everything possible then I would like to forward my heartfelt gratitude to my research advisor **Dr. Mekuanint Agegnehu** for his constructive guidance, his readiness for consultation at all times and his overall encouragement since the conception of this research work. He have listened me to all my problems I faced during this thesis work and showed me the way to address them.

I would like to deeply thank my mother **Banchayehu Alemu** who spent all her life to make the best and always me happy. Thanks for your pray, for your patience, for helping, guiding and supporting me throughout all my life. I need to thank my eagle friends Daniel Aklog, Gubala Getu, Bereket Belayneh, Desalegn Ashebir, Abrham Chalachew, Denkinesh Getachew (ዶሲ) and Yonas Takele who provided support and motivation throughout my study time.

Finally, I extend my heartfelt thanks and respect to all those people who were not have mentioned here but their contributions have been inspiring for the completion of this work. In the end, very special thanks to my emperor friend **Abita Gebrie** for his support in infrastructure and finance so far.

Abstract

Cloud computing is a virtual pool of common computing resources, which has presented to the customer through the internet. It gives unlimited pay per use of computing resources virtually without burdening the user by managing the underlying computing infrastructure. Providing cloud computing service is not straight forward task it needs appropriately balancing of load on virtual machines to achieve optimum allocation of bandwidth, memory utilization, processing speed and instruction size between virtual machines in the data center.

The cloud system resource should coordinate to provide users request response that needs intercommunication among different parts of the system, this leads to challenge in an imbalanced charge in the diversified networking system where some node get involved in over charge, some get in light charge and others might involve idle. To alleviate such problem service providers required to apply a solution on load balancing for allocation of tasks over datacenters, network, hard drives, physical hosts, and virtual machines across these virtual cloud centered resources. Applying genetic algorithm and fuzzy set theory separately have should not improve on iteration time.

The aim of this study was load balancing in cloud computing using hybrid algorithm to improve the overall performance of cloud-based system. On this study hybrid of genetic algorithm and fuzzy set theory has implemented to get optimal load balance among virtual machines in the datacenter. This study has been simulated on ten datacenters, fifty virtual machine and time-shared virtual machine provisioning policy using CloudSim simulation toolkit. The experimental simulation result showed an average of execution time 2.1 and 1.5 milliseconds for genetic algorithm and hybrid genetic algorithm respectively. Resource utilization is found 90.1 % and 53.2 % for hybrid algorithm and genetic algorithm respectively. The hybrid algorithm found less imbalance value of jobs in VMs as compared with genetic algorithm. In conclusion, proposed hybrid algorithm has found highest resource utilization and lower execution time.

Contents

Abbreviations

QoS	Quality of service
CPU	Central processing unit
IOT	Internet of things
NIST	National Institute of Standards and Technology
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
VM	Virtual machine
JIQ	Join-idle-queue
PALB	Power aware load balancing
RR	Round robin
RAM	Random access memory
GA	Genetic algorithm
FGA	Fuzzy Guided Genetic algorithm
FFF	Fuzzy Fitness finder
LIF	Load imbalance factor
IT	Information Technology

List of Figure

List of Table

CHAPTER ONE
1. INTRODUCTION

Distributed computing leads to an advanced technology called cloud computing implemented in academia and industry to store and retrieve files and necessary documents[1]. Currently, cloud computing becomes an essential computing model emerged from the rapid development of an internet [2]. A cloud introduces an information technology industry, which is created to initiate remotely rapid provisioning and DE provisioning of measured and scalable computing resources. Historically, the internet is has been represented by a metaphor of cloud. This representation were formerly taken from its collective delineation in network diagrams as a sketch of a cloud, applied to represent the movement of data through carrier mainstays that possessed the cloud to a destination that departs on the cloud in the other end. In the advancement and evolution of on demand services and products cloud computing, would be the next step in the information technology development[3]. Cloud service providers present services based on "pay-as-you-go" model instead of "own and use" technique for cost minimization purpose.

Industries, experts and providers in cloud computing give their own definition to the terminology Cloud Computing. There is not yet a consensus for what this terminology exactly means currently. Investigating a little of the existing definitions helps to give a clue for the term what it involves or might involve.

''cloud computing is a diversified and concomitant system in which several coordinated and virtualized resources with shared memory which dynamically accessible and presented based on one or several unified computing resource founded based on quality of service level agreement through negotiation between the cloud service providers and the enterprises consumers that rent infrastructure and service" [4].

"In simple terms, cloud computing is the distribution allocation and integration of computing facilities and products like servers, analytics, software, networking, database, and intelligence across the cloud to offer quicker invention, elastic computational resources, and markets of scale" [5].

Barrie Sosinsky [6] defines cloud computing as follows:

"*Cloud computing denotes to the infrastructures and facilities that route on scattered of network system implemented with virtualized computing resources and which are available through shared internet regulations and networking standards. It is determined by the fact that the ideas of computing resources are virtual and unlimited in which the fine points of the physical machine on which the application routes are abstracted from the consumers*."

"The term c*loud computing can be defined as both the facilities presented as service across the distributed network and the physical hardware and applications of a system in the datacenters that offer those facilities"*[7]. These facilities themselves have been determined long as software as a service.

Cloud computing is a virtual pool of common computing resources, which are presented to the customer through the internet. It gives unlimited pay per use of computing resources virtually without burdening the user by managing the underlying computing infrastructure. The user can scale up when needed and can minimize as much as the resource necessity. When we store our files and upload photos in an online through Gmail, Yahoo, and Hot-mail or by using social media networking sites is through by means of cloud computing service. Due to its large-scale distributed computing, cloud computing is to utilize the available computing services efficiently and gaining maximum profit for cloud service providers.

Load balancing is a vexed precarious problem in cloud computing, due a cloud service provider has to give a service to many cloud computing users, load balancing is an involved fundamental issue in founding cloud computing system environment. To overcome this issue load balancing algorithms should make order the tasks in a way that balance between enhancing the performance and enhancing the quality of service (QoS) through service level agreement by the cloud provider and the users meanwhile, sustaining the efficiency and fairness allocation of resource between the jobs.

Cloud computing becomes rapidly growing area, state of art, prominent technology in the information technology industry today with the advancement of science and technology. It provides infrastructure, platform and applications as a service. Cloud computing presents the capability of computing resources and storage on a metered basis and minimizes the IT investment cost in enterprises cloud infrastructure.

Cloud computing can distribute secure access to applications as shown in the following figure. However, high-level security is still an involved and vexed challenge for cloud computing model developers.

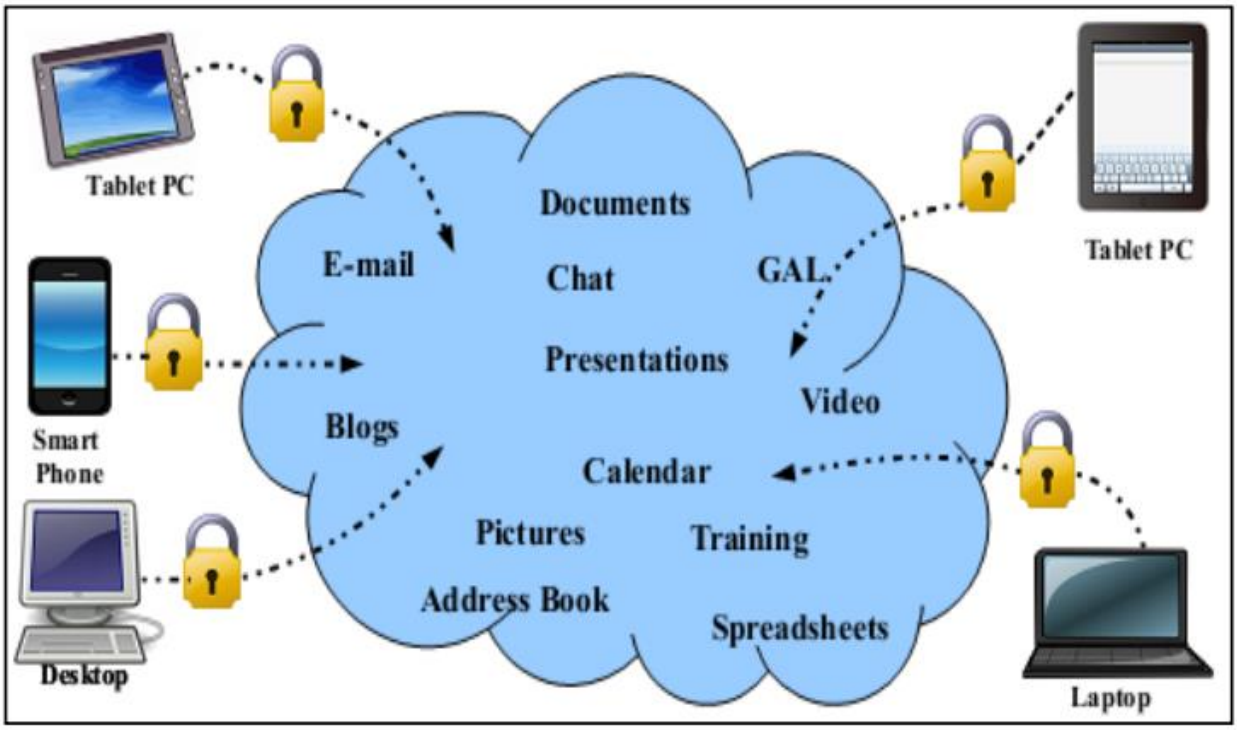

Figure: 1 cloud computing Applications

Customer request on cloud services is increasing day to day for better performance, resource utilization, proper resource allocation and to satisfy users demand it should be balanced in such a way that the performance of the system should not be reduced, degraded and works in efficient way. In cloud, computing environment data is stored and resources are distributed in an open environment as a result the extent of storage space escalated rapidly in an open environment. As a result, load balancing come to be a big challenge in cloud computing environment. Load balancing is the process of assigning the total workload to the separately performing nodes of the distributed system to make use of effective resource utilization, to ensure that no single node is overwhelmed and some other nodes not to be underutilized, thus raising the overall performance of the system.

In cloud computing environment there are involved challenges like load balancing, system monitoring, fault tolerance, resource discovery, task scheduling, resource allocation, computing

service sensing, cloud security and protection. Load balancing is the major central issues in cloud computing. Imagine an organization have a website with only one web server which operates all incoming customer requests in the organization, when cloud computing is started it is tough to handle all the user requests in a single web server. As the organization, business grows up through time a single server cannot be sufficient to operate and if not having extra server the single web server becomes loaded, slow and customer requests will be waiting until to become free to process client requests. The growing use of web services demands for high scalability, availability and reliability of web servers to provide quick response for customer request with high throughput occurring at every time. Using disseminated web servers offered an effective operational resolution technique for enhancing the quality of web infrastructure. A collection of different distributed web servers have implemented as a shared pool of virtual resources to present parallel services to the consumers. The arriving customer requests should disseminated into different existing virtual machines by considering its bandwidth and processing speed by ensuring the machine is not at its maximum load threshold level.

1.1 Statement of the problem

Load balancing is the central basic issue in cloud computing, it is the main aspect to improve cloud system performance. The recent algorithms of load balancing in cloud environment are not highly efficient. Load balancing in cloud computing environment is a complex task until today, because prediction of user request arrivals on the server is not possible. Each virtual machine has various specifications, as a result, it becomes a very difficult task to allocate arriving jobs and balance the load among machines. In cloud based environments job allocation to a particular resource is a basic problem in which the system performance is in unceasing abruptly of state devoid because of the drastic increment on demand use of the cloud service by the enterprises and users[8]. The major problem on performance degradation in datacenter is due improper allocation of tasks and resources in virtual machine. The cloud system resource should coordinate to provide users request response that needs intercommunication among different parts of the system, this might leads to a challenge in an imbalanced charge in the diversified networking system where some node get involved in over charge, some get in light charge and others might involve idle[9]. This is because of the uneven allocation of tasks, user requests changing over time, newly joining machines and a high possibility of failure in overprovisioned nodes. In contrast, if machine get idle virtual machine still consumes power without any gain in

turn. When a computing resource is freely available, it utilizes 70W of power without performing task[10]. On cloud computing service user side, In cloud based environment users would pay based on the service usage or utilization, as a result it is very essential to reduce the processing time and the makespan[11]. If machines are over loaded execution time will be increase as a result users pay for delay happen due to poor execution time this may affect customer satisfaction on the service. As the cloud service user growth up dramatically currently and the service providers needed to address the massive task requests[12], [13],[14].

In cloud, system due to improper task allocation resources overprovisioning and under provisioning happens and this leads to resource wastage, which implies CPU wastage, bandwidth wastage and memory wastage in virtual machines and hosts and task starvation problem occur in which, large tasks might take long processing time by handling the virtual machine CPU time. Lightweight tasks may wait until it finishes the current running tasks. This is caused by load imbalance, leakage of resource and which deliberately caused by service deny.

In parallel and distributed cloud environment, load balancing is the basic precondition for efficient utilization of computational resources, and improves system performance. Load balancing lets the allocation of tasks over datacenters, network, hard drives, physical hosts, virtual machines thus service providers required to apply a solution to assign the incoming loads across these virtual cloud centered resources[12], [15]. As the massive increment in cloud service users in the cloud environment, efficient load balancing mechanism is needed to handle user requests[16]. Overcharging of datacenter; leads to low system performance and denial of service issues. To overcome the above mention problems and for optimal resource consumption of available resources to enhance performance of virtual machines efficient load balancing in the cloud environment is very significantly imperative. This can be addresses through implementing efficient (intelligent) load balancing algorithms in cloud computing environment.

1.2 Objective of the study

1.2.1 General objective

The main goal of this study is load balancing in cloud computing using hybrid algorithm to improve the overall performance of cloud-based system.

1.2.2 Specific objectives

To meet the general objective of the study the following specific objectives have identified.

- To enhance resource utilization and reduce execution time using hybrid algorithm
- To evaluate the proposed algorithm system in execution time, makespan and resource utilization.
- To balance the load throughout virtual machines using hybrid algorithm.
- To improve cloud based system performance at reasonable cost.

1.3 Significance of the study

Load balancing is a mechanism to allocate and reallocate the entire task to each node in the cloud computing system for better resource utilization to reduce the execution time of task, parallel removing over resource provisioning and less provisioning of resource to the processed tasks to meet user requirement. Customers and enterprises who rent infrastructure and service from the cloud vendors would pay based on the service they have accessed. Effective load balancing can help in efficient utilization of available resources in optimal way. Load balancing in cloud system can also help in carrying out failover, avoid over and under provisioning, enabling scalability and elasticity, thereby improving the whole performance of the system. Several people dynamically balance the task load by relocating loads nearby the node too far away nodes that are less loaded or idle. More on this, green cloud computing can be realized through load balancing in the distributed cloud system. Many people balance the work by dynamically relocating workloads nearby to the system to far away nodes, which are less loaded or idle. Additionally, green cloud computing can be accomplished through load balancing. Certain features with this regard, the following very important points[17].

Limited Energy Consumption: load balancing can simply censored the capacity power utilization by removing overheating of system nodes or virtual machines that that arise due to extreme task load.

Reducing Carbon Emission: carbon emission and power consumption are the two most important and coordinated features in green computing. Therefore, as reducing power consumption load balancing, can also automatically minimize the carbon emission, and hence develop green computing.

1.4 Scope of the study

The intent of this study is to propose a load-balancing algorithm that targets on addressing the performance issues of the recent algorithms through by maximizing computational resource utilization and minimizing makespan and execution time.

1.5. Methodology

There a number of method of load balancing algorithms in cloud computing. We used a hybrid load balancing algorithm of Fuzzy set theory and Genetic algorithm to enhance system performance in cloud environment and to effectively utilize the available computational resources in cloud. A single objective optimization model cannot serve in the objective of fitness measuring index because we are looking at multiple input variables that could bring VM processing speed, cloudlet length, VM memory size and bandwidth together. In this regard, hybrid of Fuzzy set theory and Genetic algorithm are used.

1.6 Thesis organization

This section describes the organization of the work in the thesis. It has structured as follows: Chapter 1 discusses about background information about the cloud-computing environment, the statement of problem, objectives, methodologies and related concepts have presented. Chapter 2, presents the conceptual discussion about load balancing in cloud computing and detailed literature review and related works. What the trends in cloud computing tell us, Services and deployment types Cloud Computing. Chapter 3 illustrates the used and proposed methodology. Chapter 4 is all about the experiments results. Chapter 5 is conclusions and future directions that need further investigations.

CHAPTER TWO

2. LITERATURE REVIEW

A new time sharing system technology was created in 1961 after McCarthy in the history of computer technology suggested that likewise water, electricity power and telephone are public utility, utility of computing may structure as public in MIT anniversary celebration of centenary[18]. As a computer science professional, he was the first who established timesharing technology would lead to the most powerful and dominant controlling computing paradigm. His state of art ideas was very popular during the time however, slowly faded away in 1990 following the starting of the 20^{th} century that his creative thoughts are winding up within a new approach which is now called as cloud computing.

From the beginning of 1970, computer age groups have vanished through rapid changes and modifications during when the mainframe computers were announce into the information technology industry. In the first 1980, also called recessionary phase, in IT enterprises computers were manufacture to maximize the efficiency and effective level of the commercial industries and the individual customers in by increasing profitability. Client-server computer architecture was offer in new capabilities and efficiency such as LANs to improve enterprises and user's productivity using shared network model in 1990.

Later on, following the introduction of the internet as an important invention in the IT enterprises Internet of Things were announced to link users with the virtual world, which brings the appearance of cloud technology[19]. The time is 1990 that John Romkey produced a toaster which can be turned on and turned off over the internet, was the forerunner device working on the internet. IOT describes the organization of a virtual network, which handles physical objects controlled through internet. Internet of Things waterlogged the user's lives by offering them by means of controlling tricks, which might intelligence, compute and can interconnect the users demand accurately and quickly. Linking many varieties of things in together can help users in getting a huge source of information to improve data management. Additionally, IOT offers researchers with dominant controlling computing devices with real-time data processing and decision-making. Motivating by IOT model, currently the novelty of the fifth IT industry named as cloud computing, produced a rapid IT transformation by providing dynamic provisioning of computing resources to customers.

In October 2009, conference is seized authorized as "Effectively and securely using the cloud computing paradigm" by [20] NIST (National Institute of Standards and Technology) is an Information Technology Laboratory, Cloud computing is defined as follows:

"*cloud computing is an architecture which allows suitable, on-demand network access to a common pool of computing resource which can be easily arrange and reliable (services, networks, storage, datacenter) that can be quickly provisioned and released with less user control right or service vendor interaction*"

The cloud computing model consists of five significant characteristics, three service models and four deployment models. These things are highlight here under:

2.1 Characteristics of cloud computing

The five significant characteristics of cloud computing is as follows:

On-demand self-service: users can assign and release computing resources like server time, network storage and others when they need automatically without necessitating human interactions in cloud service providers.

Ubiquitous network access: cloud computing capabilities are available with reachability over the network and accessed via standard methods that promote use by heterogeneous thin and thick client platforms.

Resource pooling: cloud computing services are pooled in an organized manner to work for several users using multi-tenant model in which various physical and virtual resources are assigned and released dynamically.

Rapid Elasticity: cloud service consumers can scale up or down the cloud resources, they are going too used; in some cases, it can be automatically scale out when is required and rapidly released when users do not need.

Measured service: customers and enterprises rented infrastructure and facilities from cloud service vendor have charged according to the resource they utilize.

2.3 Cloud computing service models

Service means various types of software offered by various servers over the cloud. The three services models in cloud are the following.

1. Software as a service (SaaS)

In SaaS, a service provider licenses different software application from various servers to the customers for use through the internet as service on demand[21]. The customer uses the application without change and does not to do many modifications or no need to integrate to other systems. The service providers do the changes and maintenance despite the fact that the infrastructure is operating.

2. Platform as a service (PaaS)

In PaaS, providers offer a computing resource delivery platform that are required to build applications, where users can deploy their application and run on it without monitoring the underlying hardware and software layers, however can maintain the control over the deployed applications.

3. Infrastructure as a service (IaaS)

IaaS is a way of providing computing resources such as Network, Storage, processing unit and operating system in over the internet. Instead of buying servers, storage and software enterprises can enlarge their resource competences by getting these computing resources on demand service over the internet as IaaS. Infrastructure as a service provider serves enterprises through private and public clouds.

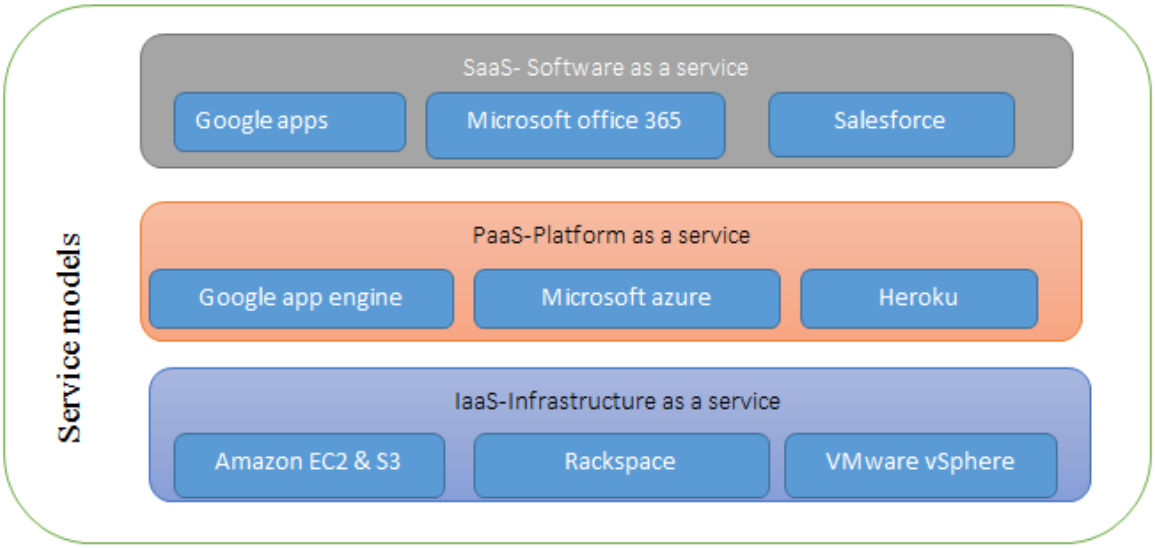

Figure: 2 Cloud service models

2.4 Cloud Deployment Models

Today the cloud substantiated cloud service delivery models, with in the rising number enterprises realizing remarkable agility efficiency benefits. Regardless of delivery service, models (Iaas, Paas, and Saas) cloud computing service delivery models are deployed in four techniques.

Public cloud: a cloud service provider offers cloud services and infrastructures to all business, academics institutions and government organizations with access over the internet. It may offer either a single-tenant or multitenant operating atmosphere with benefits and functionality of scalability and utilization of the cloud.

Private cloud: cloud services are provision for use by a single organization or their designated services and offer a dedicated or single tenant operating environment. Private cloud targets to address data security, provides greater control, which is lacking in public cloud. Private clouds have deployed to benefit of implementing, serving and using cloud storage of flexibility, management simplicity of cloud model.

Community cloud: cloud services are provided for use to specific community users which having common cloud infrastructure requirements.

Hybrid Cloud: this category of cloud is a combination of two deployment models of cloud (for example a combination of private and public cloud). Hybrid cloud elements are remaining unique however, bound together by a standardized or commercial technology to enables data and application compatibility, convenience and reliability.

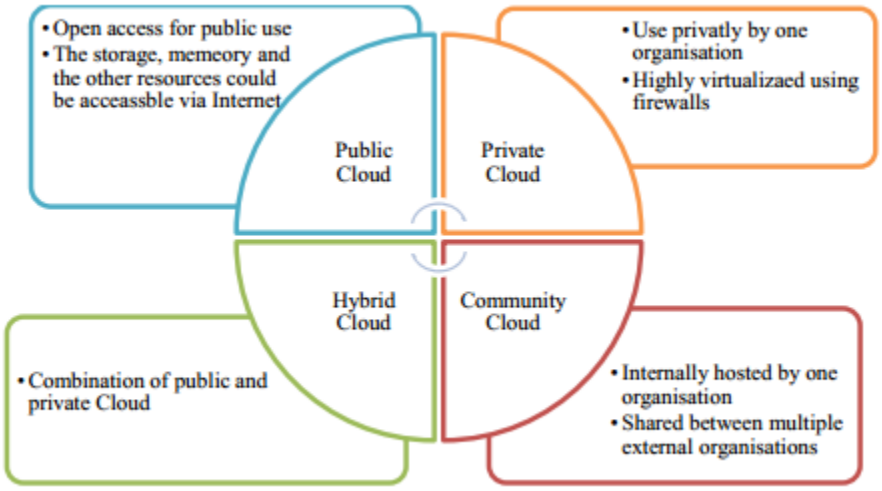

Figure: 3 cloud deployment models

2.5 Advantages of cloud computing

Cloud computing provides several benefits to the cloud service users. The following are highlight as the main advantages that cloud computing offers.

Scalability and Elasticity: scalability is an incorporated feature of cloud computing. Service providers have several enormous numbers of infrastructures and services. So those providers can simply extend and lower its facilities to monitor the raising services based on the users[22]. Whereas elasticity is, the capability of scaling up and down provided resources when demanded. In fact, elasticity expounds the description of "pay-as-you-go" model that is assigning and releasing of services based on user requirement[23].

Reduction in cost: cloud service resources is based on provisioning and DE provisioning on demand access strategy then this can offer a substantial saving on operating economical costs. Several applications deployed on cloud do not require labor and can have easily configured freely such as emails and Google apps.

Ease of management: cloud services have delivered across web-based services; users can access it through any device that supports internet connections without installing applications for computation and repairing of the infrastructure needs less time and cost.

More storage and capacity: storage capacity is among the several significant elements of cloud computing environment. It can store large data as compared to personal computers and disks; it eradicates the fears of running out of storage space. Everything is on the internet, can store your whole data on cloud and can access at any time.

Disaster recovery: through virtual backups, disaster recovery is much faster than using another system other than cloud services.

No up-front investment: The principles of cloud regarding payments modes of services are based on the model of pay-per-use. Enterprises can then getting the capability of rent services from cloud, as they require.

Green computing: currently, energy consumption is an involved scenario consisting electronic waste with the improvement in time, extensive usage of system resources. This can be addressed with help of cloud computing. Less e-wastage produces results that preserve the environment.

2.6 Barriers of cloud computing

Alongside all the cloud computing benefits provided, cloud computing shall able to overcome several barriers to make certain a capable implementation of elastic scalable, secure and private reliable platform [24]. The main challenges and issues are point out here under in table:

Barriers	Opportunity
Interoperability	Non-existence of principles for service convenience amongst cloud service providers.
Security and privacy	Less enhanced approaches in consumer authorization to access their information
Resiliency	The capacity of the system to deliver users with levels of services upon suffering faults in the system.
Reliability	Failure opportunity in standard period of time
Energy saving	Describing a standard metric for efficient power usage and effective standard of infrastructure usage
Resource monitoring	No improved control techniques through sensors that collect data from CPU load, memory load etc.
Load balancing	No, a standard way of load management for various cloud applications

Table 1: Barriers of cloud computing

2.7 Virtualization

Virtualization is the software execution of a machine that will perform in various programs like a real machine. In cloud-computing environment's platform, dynamic resources can be successfully controlled through virtualization technology. Subscribers with many demanding service level agreements can be warranted by accommodating all the needed services through virtual machine image and by mapping it on a physical server. This helps to address the issue of resource heterogeneity and platform inappropriateness. Using virtualization clients can access cloud services. It affords the capability to run several operating systems on a single physical machine that shares the underlying resources.

Virtualization is a basic empowering technology in cloud computing environments that enables to run many operating systems and applications on same physical machine at the same time[7]. It hides a computing platform physical characteristic from the clients[25]. Virtualization allows the separation and abstraction of principal hardware has and lowers level functionalities. It consents abstraction and isolation of lower level functionalities and underlying hardware. It can enhance

hardware utilization, lowering costs for recovery and backup of disasters to realize automated controlling for the entire hosts. Although it is very problematic to allocate a large extent of jobs to dynamic resources for cloud computing. There are various types of virtualization in cloud spectrum. These two types of virtualization are:

1. Full virtualization: full virtualization is a complete installation of a machine on another machine. The virtual machine delivers all the software has and functions of the original physical machine. Its services when an actual machine is not free then the user use virtual machine.
2. Para virtualization: is when the hardware allows multiple operating systems to run on a single machine. It also allows the efficient use of system resources like memory and processor.
3. Emulation virtualization: this type of virtualization occurs when the VM pretends the hardware part and develops self-determining of it.

2.8 Virtualization Benefits

Virtualization is an efficient way to minimize IT expenses upon boosting efficiency and agility. The benefit consists:

- Multiple applications can run in one physical machine.
- Consolidate hardware to have high production from few servers.
- Save 50%+ on the entire IT expenses.
- Increase speed and simplify maintenance, deployment of new applications.

2.9 Resource Allocation

Concerning to cloud computing settings, resource allocation defines away of assigning the available cloud resources to cloud services through the internet[26]. The main target is to keep track of overloaded nodes therefore, no wastage of resources. The wastage defines that the wastage of bandwidth, CPU, storage and memory. The resource mapping have performed in two ways:

The first way is that mapping the virtual machine on to the smaller number of hosts. The virtual machine is located on the physical server termed as the host. A VM is draws to the host and the procedure is depends on the availability and capacity thus, allocation mainly depends on the on-

demand resources. The second way is that mapping applications to virtual machines, which is increasingly significant in cloud environments. Additionally, monitoring power is highly required because of the increasing demands of computing power and utilization of the power in cooling resources of a datacenter. A VM presents a power as the applications are processed on the virtual machines, thus depends on the availability and configuration of resources. However, the cost of power is an essential factor to a provider because reduction in power utilization leads to cost minimization in cloud infrastructure.

2.10 Task Scheduling

Task scheduling regarding to cloud computing implies the dynamic distribution of cloudlets across the cloud resources to accomplish best outcomes[27]. Concerning to task scheduling: tasks have programmed across virtual machines for running objective. Task scheduling is the most fundamental involved issues in cloud computing because clients of cloud service will have to pay based on the resource using based on time. The aim of task scheduling is to reallocate the whole load evenly all over the system by increasing the consumption of resources and reducing execution time of task[28]. It has a substantial effect on ensuring distribution of computing resources efficiently and fairly.

In cloud, task scheduling is mainly having generalized into three steps:

Resource discovering and filtering: the data broker finds out the resources available in the network structure and assembles status information regarding to it.
Resource selection: aimed resources are carefully has chosen based on some resources and task.
Task submission: a task has submitted to a resource that has been chosen.

2.11 Load balancing

Load balancing is a technique of sharing capacities or allocation of overloads to the entire nodes of the system, which are idle, and under loaded nodes. These involved problem is tackled by resource utilization in which each available machine should utilized in the way that the algorithm should first confirms whether free machine is available or not and assign tasks to free machines. The basic target of load balancing is enhancing the performance of the system, to increase the network efficiency by increasing the throughput of each nodes in the system[29]. In order to undertake these objective many algorithms has deployed even if they are under problem.

2.12 Types of load balancing algorithms

Upon on the initiation process, algorithms of load balancing can be characterized into three types as described by[30]:

Sender initiated: This is type of load balancing approach is happening when the sender of a particular task or activity to be performed initiates the algorithm.
Receiver initiated: This type of technique of balancing load is occurred during when a particular receiver of jobs initiates the algorithm.
Symmetric: it is an amalgamation of both the sender and receiver initiation processes.

Based on the existing state of the system load balanced algorithms can be characterized into two classes as given in[30].

Static: the previous status of the system is required. It cannot adapt run time changes of the load.
Dynamic: load balancing decisions are done based on the existing state of the system. No prior knowledge is required; consequently, it is better than static approach. Here in the following briefing numerous dynamic load balancing algorithms are in detail.

2.13 Dynamic load balancing algorithms

In cloud computing spectrum, dynamic load balancing is performed in two various techniques: Distributed, non-distributed (centralized) and hierarchical. In distributed type of dynamic load balancing algorithm, all nodes are responsible to share the jobs of load balancing. Tasks are distributed in multiple entire domains, which are in charge for the entire function. Here no single node is overloaded. The interactions between nodes are achieved in cooperative and non-cooperative ways. In cooperative form, nodes are operating side to side to achieve a common objective whereas in non-cooperative, nodes are operating independently to realize an aim local to it. In distributed nature of algorithms usually, produce several communications than the non-distributed because each node needs to communicate to other nodes. An advantage of this is, if one node in the system fails, it will not cause the entire system function to stop.it may in effect system performance in some extent.

In non-distributed dynamic load balancing algorithm, a single centralized node makes decisions, and this node is responsible to the whole operation in monitoring the network. The process may be static and dynamic, based on the specifications required. This approach can cause a bottleneck

at the central node and has high intensity failure rate, no fault tolerance because of the overhead, mainly on the single centralized node; also, the failure recovery is not an easy task.

In hierarchical load balancing algorithms, this algorithm mainly operates in the master slave node. Parent nodes are responsible and perform to balance the nodes. Several levels of the cloud are taking part, and is based on slave mode functioning. This architecture is deployed based on tree data structure, where balancing of each node in the tree is supervised by its root node. Master slave node can get information through its agents.

Nature of algorithm	Base knowledge	Advantages	Limitations
Static	Previous system status has needed.	Used in homogeneous cluster	If change is happen, not scalable.
Dynamic	Run time knowledge status is required	Used in heterogeneous cluster	Complex structure Time consuming
Distributed	Load balancing is done by all nodes	Used in large heterogeneous cluster.	Complexity
Centralized	Load balancing is done by single node	Used in small networks	Single node overhead No fault tolerance
Hierarchical	Nodes at different stages of hierarchy	Used in medium and large heterogeneous cluster	Less fault tolerance

Table 2: Load balancing algorithm categories

2.14 Dynamic load balancing strategies and policies

Dynamic load balancing algorithms[31] are complex, but are offering better system performance and fault tolerance. In dynamic algorithms, some strategies are applied. These can be highlight as follow:

Transfer policy: a portion of dynamic load balancing strategy or policy that specifies the selection of tasks for moving from a local node to a distant remote node.

Selection policy: it determines processors that get involved in the interchange of load (matching of processors).

Location policy: a part of dynamic load balancing algorithm, which specifies the selection of a terminus end node for task transferring.

Information policy: a part of dynamic load balancing algorithm, which is undertaking and responsible for collecting nodes information in the system, is denoted as information policy or information strategy.

2.15 Qualitative metrics and important resources in Load balancing

Qualitative metrics comprises just about parameters that are good important to discover the productive algorithm amongst them. Various parameters or metrics, which have deemed significant for balancing workload in cloud computing environment, are expound as follows:

- **Throughput:** mainly for cloud computing systems, it is the time taken from submission of the cloudlet until the first reaction have generated. To have a better system performance a higher throughput is required. Maximize the throughput for users and cloud service provider's benefits for them both.
- **Migration time:** is the time taken to migrate a task or a resource from one node to another node in the system. It has to minimize to enhance the system performance.
- **Response time:** response time is the extent of time taken to respond for processing by a specific load balancing algorithm in a cloud based system. It needs to optimize because in the cloud system payment is based on the time one accessed.
- **Resource Utilization**: It is the amount to which the resources of the system are effectively and efficiently consume. A better load balancing algorithm provides maximum resource utilization for an efficient load balancing.
- **Fault tolerant:** Fault tolerance is the capability of the load balancing algorithms that empowers a system to continue functioning well and uniformly even in conditions of the failures of at any arbitrary node in the system.
- **Reliability:** this metrics have related with the capability of presenting the users requirement in the cloud load balancing algorithms even in case of some machine failures.
- **Scalability:** It determines the extent to which a load balancing algorithm is to accomplish to workload changes through provisioning and de-provisioning computing resources in an

automated way, in such a way that at each time computing resources should mapped with the recent required service as diligently as possible.

- **Energy consumption:** Determines the energy utilization of all the resources in the system. Load balancing benefits in sidestepping hotness through balancing the tasks across all over the nodes in the Cloud system, therefore, decreasing energy consumption.
- **Performance:** As the great computing power in cloud, application performance should have guaranteed**.** It signifies that the effectiveness and efficiency of the system after operating load balancing. This should have enhanced at reasonable cost. If all above aforementioned metrics are contented at optimum, then it will extremely enhance system performance.

As expounded by [32] the fundamental important resources in load balancing consists the following:

Computer processor time: this is the most basic and essential resource in computer system. While distributed system is used computer CPU-time needs to be balance.

Computer memory: memory is another significant resource in computer system, used when distributed computers have connected over the cloud; this computing resource needs to be balance to heighten the presentation of the system.

Computer input-output resources: In mapping the load of the computer processor time and RAM is not reasonably well, enough to perform efficiently in some time, therefore I/O resources, which is upon on the efficient consumption of storage, required to have shared. Load balancing is an essential and basic mechanism to maximize the service level agreement (SLA) and better consumption of resources in cloud based spectrum.

2.16 Load balanced resources

There are several computing and network resources, which have needed to balance in the cloud systems. The following resources are among the these which are very essential in the distributed network system[33]. The basic resources are the cloud storage, network service and interface, intelligent switches with connection, application instances and processing via system allocation process. Without implementing load balancing in the cloud computing environment, it is very complex to control its technology. It provides involved redundancy procedures to make unreliable environment reliable in managing redirection.

2.17 Related works

In [34] a hybrid job scheduling in cloud computing environment is presented, which is based on the Fuzzy set theory and Genetic algorithm that mainly targets to reduce execution time and cost. In this work, the algorithm allocates jobs to virtual machines but it fails in efficiently utilization of all existing computing resources (VM). It repetitively allocates tasks to some of the machine but still there is idle virtual machine. While the algorithm allocate jobs to some of the machines overloaded occurs and some idle and that consumes power, which needed to be removed. This issue is address in by modifying the algorithm in our proposed algorithm. The proposed algorithm monitors all the existing virtual machines. While a new job arrives, the algorithm confirms available free machine. If exists it allocates the job to the machine based on its processing capacity. If there is no free available machine the job allocate to a machine with less time to finish the ongoing job by contrasting with other machines.

In [35] proposed an a load balancing algorithm that is upon on a particular behavior of the ants, in which they move toward the place where high amount of nourishing food is available. They always find their nourishment and the food source to deliver back to their home. In ant's colony optimization algorithm, a head node first is selected based on nodes that have high number of neighboring nodes. Every ant in the colony moves for food in identical way and same time. After they get their food they move back to the direction to their home, similarly the ant colony optimization algorithm proposed that when an ant moves forward to the way that encounters the overloaded, under loaded and free available nodes. During the time, ants get an overloaded node while in the earlier time it searches under loaded nodes then it moves backward and confirms whether the node is still under loaded or freely available. If it is confirms that the node is free or under loaded, the task is now distribute throughout all under loaded and freely available nodes in the system. As a result, this algorithm is an efficient algorithm in utilization of computing resources. Despite the fact that it is limited because of very high extent of ants in a colony, the network traffic might be congested. The status of nodes after ant's observation is also not take into account.

In [36] proposed a virtual machine mapping policy that depend on multi-computing resource load balancing algorithm, which deployed on private cloud setting in which virtual machines includes central scheduling controller to monitor resource availability for a work and free

available resources are allocated for nodes in a system for processing in a cluster of nodes. Resource controller is there to calculate and investigate the detail information of resources, which are available free. This algorithm is based on mapping of virtual machines in private cloud. Load balancing procedure in this algorithm includes many levels such as acceptances of request, gaining detailed available resource information, and controllers of resource and scheduling which performs scheduling of tasks and investigate resource availability for scheduled tasks. Resources have allocated for processing of tasks in which they have high amount then clients can access through the application. The limitation over this algorithm is that it does not take into account capabilities of node and the load of the network and if single point of failure happened the whole system is getting stop functioning.

Efficient load balancing technique [37] is performed using divisible load scheduling and weighted round robin method. In this approach, requests have divided into sub tasks, which are executed one after the other, and some can run in parallel. Every server have allocated with his or her corresponding weight and his or her previous allocation status. During a task is arrived to a cloud server, it is conceded to the load balancer which can divide requests to many arbitrary sub tasks based on divisible load balancing technique, which can implement a priority-based task allocation to the available servers. The load is assign to a server having a higher weight to all sub tasks and the status of the server is change to busy and again changed to ready. Transforms from busy to ready status for a server supports that it is no longer available in the ready server list, in such kind of way overlapping of tasks have prevented. The server again joins the ready server list after it finished its allocated work. In this load balancing scheme, network performance is increased whereas request completion time is minimized and it can eliminate task starvation problems. The limitation of this algorithm is in the distribution of loads to server is poor allocation.

Honey bee behavior inspired load balancing [38] is presented that is based on the behavior of honeybees colony, which can be categorized under two types. The first honeybee is that it searches honey nourishment whereas the other is the honeybee that reaps for honey. The first category of bees goes in to have honey and to search honey nectar origins. After getting nectar sources these bees turns back to their home and they perform waggle dance, which have executed in the dance floor to where inactive forager bees can attend and following, to indicate

the qualitative and quantitative amount of the nectar sources they searched. Every active forager bee efficiently and effectively delivers answers on their local flower area at the same time as attending bees have contact a set of beautiful nectar resources that would be exploited through the honey bee colony. However, no forager bee demands the overall global understanding but also, can randomly select a waggle dance to observe from which area they can absorb the environments of the flower and they leave their home for foraging. The technique response delivers a feedback threshold using enlisting signal. The assimilation of feedback inception generates the distance of the enlisting signal, which is waggle dance in the honeybee colony.

Likewise, in the honeybee inspired load balancing algorithm of tasks in the cloud environments, several internet servers are grouped together as virtual server. By considering virtual servers like the honey bee and the web services as food resource origins like flower patches. The self-motivated behavior of the received cloudlet appearances, which is user operation of daily activities, is corresponding to the flower area volatility, which is a function of a day-to- day changes in environmental weather conditions. A single server is associating to a single forager bee. A forager bee selects a virtual server randomly, that is engaged in giving services for a particular web requests queue, which have deemed as a cluster of honeybees that get involved through accumulating nectars from a definite flower environment. The cluster of servers as colony of honeybees, and the web service queue associated to the co-hosted web application services as the origins of food resource nectars to be exploited profitability. Depending on the nature of contributions of the profit virtual server processes the cloudlets. If after they do calculations and get less profit the server turns to their forage as forager bee and do migrations from one flower patch to the other flower patch. As a result, this conserves that the balance of the load in the system which improves performance. The computation in the profit may cause an additional overhead, which result in a whole decrease in throughput.

In [12] a genetic algorithm based strategy of load balancing is presented, which is built on a way of natural selection strategy. A hypothetical configuration has done by dividing the world into six regions representing continents in the world. Six user bases modeling used by representing the six regions considered. A single time zone depicted for all user bases have assumed that different number of customers registered online during peak hours. In this technique, all simulated data Centre hosts have a specific extent of virtual machines dedicated for the

application. The usage of this algorithm is consenting load balancing in cloud computing through minimizing the make span of the given cloud jobs. The method pledges the quality of service requirements of the user tasks.

Dynamic biased random sampling balances the system workload is built on the dynamic and random sampling of the nodes. In this algorithm, the system load is determined by constructing a virtual graph, which depicts the connectivity of each node in the system. Depiction of nodes in the virtual graph is constructing by the edges and free resources in the system are indomitable by the in-degree. The work distribution and resource updating required for balancing the load are determined in the configuration of the network. The free resource has decreased by the value one when a job have assigned. After the completion of the task, the node generated an edge and the weight of the in-degree is incremented by the value of one, which means that the readiness of the free resources is increased. Thus, increment and decrement of the node in the free resources is done via Biased Random Sampling [39]. Processes have calculated by contrasting threshold parameters, which is depict as maximum walk length that indomitable by computing the navigating from one node to the other up to a final destination is arrived. When completing a job, the load balancer chooses a random node and the recent node walk length is contrasted with the threshold value. A task have allocated to a node when the threshold value becomes less than the walk length. If the walk length of the nodule is lower than the threshold value, a task has allocated to the following node and the recent walk length is increased by the value one. The next node is depicting as the neighbor of the recent node. This algorithm offers a highly reliable and scalable approach, which balances the workflow, but the computation of the walk length generates an additional overhead.

Active clustering [40] is a technique in balancing load in cloud environment. Active clustering is an upgraded technique of random sampling where, it works based on the thought of clustering similar nodes into one, rest on the matchmaker node, which then performs in cooperation. The system performance have improved by maximizing the resource parameters. The increase system diversification will degrade the algorithm. A node inspires the process and chooses a matchmaker, which establishes connection with neighboring node which of the same nature as of the initial node. Then the matchmaker disconnects between it and the initial node. This algorithm follows iterative approach.

Join-Idle-Queue [41] which is used in the enormous scale work load balancing in dynamically scalable distributed information. JIQ algorithm balances the workload through the dispatchers, which mainly concerned on the availability of the idle processes. The central attention is to decouple the location of easily weighty loaded processors from the job, which is in charging. This technique can reduce the average queue length in every processor and deserves less communication overhead due to during job arrivals and therefor reduces response time of the system, hence this rejects load balancing work from the critical path of request processing. The vexing limitation of the join idle queue is that, it is not scalability in web services. Currently, in nature web services are dynamic. JIQ is subject to on two level systems possessions. The initial is the dispatcher possessions, which receive tasks from the client, and it checks the availability of servers in the queue. If server is available, a task have assigned to it otherwise, it selects a server randomly. This has known as primary load balancing. The second possession is the server behavior, which is after processing of the tasks, the workload balancing of the idle processors through the dispatcher

Power Aware Load Balancing [42] Algorithm proposed by which is efficient in power that rest on the size of request of customers. Power aware load balancing was designed with the main objective to save powers from switching of unused compute nodes which delivers the operation atmosphere for virtual machines in cloud computing and to contribute computation management to cluster controller, which is aware of the entire standing of cloud assets. The PALB receives incoming task requests in the way that virtual machines, balancing the load through compute nodes and proceeds the power consumed through those computed nodes in the specified time period

This PALB, technique has three main parts consisting balancing part, upscale part and downscale part. The first balancing part orders the state-run of the virtual machine. When all the on the go nodes computed be greater than 75% resource utilization, then the workload is reassigned to the fresh virtual machine with the least load or idle node, otherwise the workload is reassigned to the operated node with the objective of balancing the workload. The onset level of 75% resource utilization has selected. The upscale part of the technique is accountable to power up the extra nodes when the recent consumption of the nodes is greater than 75%. In addition, the downscale part is responsible to power down the under consumed nodes to cost of the power. Therefore,

PALB provides the blackout indication to the work-shy node. The blackout indication offered to the nodes, which are under 25% resource utilization. This process is essentially detecting and judges which virtual machine is to be instantiated. It can also be resolves that have to activated.

In [43] presented a scheduling strategy for virtual machine resources that procedure on the prior logs and the recent server status. By having Genetic algorithm methodologies, the strategy benefits to in lessening the dynamic migrations in the system. This technique assists in resolving the involved and vexing issues of load unbalancing and in elevation cost of migration in the system, which can do realization of improved utilization of resource of the system. The main central limitation of this technique is that, sometimes the previous logs cannot give the recent scenario at its best level.

In [44] an effective load balancing algorithm using fuzzy logic with Round Robin (RR) algorithm in virtual machine atmospheres of cloud computing to attain response time processing time. The technique gains information about the number of cloudlets currently assigned to the virtual machine and each VM. It specifies a little bit loaded machine, when a new request has received to assign, but if there are additional little loaded machine, the selection has made by based on processor speed and the VM load in the fuzzy logic algorithm. It improves the load balancer performance and reduced the response time of the system, additionally it shows that the algorithm results are better than the round robin algorithm. The main drawback of this algorithm it only emphases on to decrease the response time of the task scheduling and it ignored the processing cost.

A scheduling strategy [45] suggested a scheduling strategy in load balancing on Virtual machine resources that depends on genetic information to advance scheduling of tasks to be processed to realize load balancing by using the previous historical data and the recent status of the system. This algorithm makes a plotting relationship between the set of physical machines and the set of virtual machines and it finds the least effective solution by computing ahead in effect of the system afterwards the setting out of the needed virtual machine resources. It uses the same formula to find the finest load balancing scheduling using population. The output result of the experiment indicates that an enhancement in resource utilization, whereas the algorithm has high cost to store and retrieve the previous state of the data of the system nodes, and it may rise the response time and the processing cost.

A job scheduling algorithm [46] attempts to effort to achieve better utilization of computing resources to encounter the self-motivated requirements of tasks in offering job scheduling strategy that built on load balancing in cloud computing atmospheres. They try to enhance the scheduling policy on load balancing in improving the response time, consumption of resources and by plotting jobs to virtual machines then virtual machines to host resources. They procedures the initial stage of scheduling which is from user's application to virtual machine to host resources to generate an expound of virtual machine consisting of tasks of computing resources, network resources and storage resources and used the second level scheduling strategy to find a called for resources for virtual machine. The algorithm improves the resource utilization but using two level scheduling would maximize the response time compared to another load balancing.

Algorithm	**Advantage**	**Drawback**
Ant colony optimization	Idle and under loaded nodes is discovered at the opening of searching. No single point of failure due to it is distributed.	The network may be congesting because of many several ants in the search. The status of the node after visit is not taking into account.
Honey bee foraging	It realizes global load search using local server applied for large-scale cloud-based systems.	Profit computation may cause overhead resulted in the entire throughput decrement.
Active clustering	It reduces the task allocation by connecting active similar services.	While in the increase in the node diversity, the system performance lowers.
Power aware load balancing	It is implementing with the objectives of computation control to the cluster controller.	Idle, under loaded, overloaded nodes are not considered.
Join idle queue	It allocates idle processors to the dispatcher for availability and assigns tasks to processors to minimize task	It can't be used to dynamic web content services because of scalability and reliability.

	length.	
Dynamic biased random sampling	Realize load balancing throughout all nodes using random sampling of system domain.	Additional overhead has created due to walk length computing.
Min-Min load balancing	Task with the smallest time is process first.	Some large jobs may experience starvation problem.
Weighted round robin	Selects the first node randomly and assigns to all other nodes in circular fashion.	Distribution of loads to the server is poor allocation.
Fuzzy logic	It improves performance of the load balancer and reduces response time of the system.	Emphasizes to minimize response time and ignore processing cost.

Table 3: summary of algorithms

CHAPTER THREE

METHODOLOGY

In the current, load balancing algorithms in cloud-based systems, which have diversified processor power, is not highly efficient. As the data volume is increasing rapidly because of the intensification in the extent of users of the internet worldwide, there is a need to monitoring the load efficiently in cloud systems. In this regard, cloud computing paved a ground-breaking in this direction of distributed system in realizing optimized system performance, minimizing response time, maximizing efficient resource utilization, and adaptability of service level agreement. The fundamental target of this research work is to optimize the load balance in the diversified processing power in cloud computing setting area through decreasing the execution time of cloudlets in virtual machines at hand.

In this specific section of methodology, we try to highlight the research method, which employed in this study. It expounds the techniques adopted for this work. The thesis topic focuses on the wide range feature of involved issues on balancing the task load in the cloud computing environments. In the way to inquire into these features, the method that we need to apply here is using fuzzy set theory that can support genetic algorithm to enhance the standing load balancing trials in cloud environment.

3. 1 load balancing strategy

By having the default allocation of computing resources, cloudlets have to be assigned to cloud server of virtual resources [47] The main question here is *"how to allocate more important and suitable computing resources to cloudlets or tasks in order to realize a well-balanced load across virtual servers with less extent of servers."*

In cloud computing environmental configuration, load balancing is a technique of allocations of tasks across two or more several distributed datacenters, network accesses, processing units, storage and computing resources in general. Emblematic datacenter implementations are based on a massive, essential computing services and network accesses that depends on public shared risks regarding with physical device, failures of hardware's, power commotions, and limited extent of resources in case of having several user requests

Currently, Load balancing algorithms in cloud computing system environment are not enough efficient in reducing response time and its resource utilization on-demand basis. However, artificial intelligence approaches can increase the whole system performance of cloud environment applications. There is no a wide ranging of an approach that can address cloud involved issues in several varieties of parameters. Every load balancing algorithm has their own qualities in a particular area.

Hence, the estimated model allies' different approaches to interchange the appropriate load balancing techniques as per the system performance level. The main target of this research work is to realize the resourceful based performance in the cloud computing environment. In this section, we will expound the hybrid load balancing algorithm, which is based load optimization in cloud systems that can take advantages from both of the fuzzy set theory principle and the genetic algorithm techniques. Currently much attention has given to these artificial intelligence methods because of their inferred parallelism.

3.2 Proposed load balancing scheme

In the existing fuzzy and genetic hybrid algorithm[34] it allocate jobs to virtual machines to balance the load but it is not efficient in resource utilization in which it fails to allocate. In the existing hybrid, algorithm jobs have allocated to some virtual machines repetitively that leads to overloading and some machines remain idle. This leads to under resource utilization. Even if computational resource were free, they have used power, so resources should have effectively utilized. This problem have addressed in by modifying the existing algorithm in our proposed algorithm. The proposed algorithm can control all the existing virtual machines through utilization model that have employed in order to present a fine grained monitoring over available computing resources used by cloudlets.

In our proposed load-balancing scheme, we use a hybrid algorithm of fuzzy set theory that supports the genetic algorithm in finding fitness of the individual population, which reapplied in the crossover stage. In the first place, initial population that is common variables uses in the solution set for the problem has generated. Then datacenters, broker, virtual machines have created. Virtual machines have submitted to a broker, which specifies which host then provides a service to a particular VM. Required number of user cloudlets or tasks would create in which a response is route back to this requests. The algorithm calculates free available processing

elements in the VM allocation that represents the provisioning policy of hosts to virtual machines in a datacenter. Based on the result, fitness value of each chromosome is calculated using fuzzy set theory. For better resource utilization, when a new cloudlet arrives the proposed algorithm confirms available free virtual machines. If it exists, the algorithm allocates the task to the VM based on its processing capacity. If there is no available idle virtual machine, the task has allocated to the VM whose recent task is going to be finished with shortest time as compared to the rest virtual machines. In this regard, virtual machines are efficiently utilized, no virtual machine leftover free or idle and no over utilization of virtual machines. While it checks the utilization threshold level and if it did not exceed the maximum level it applies the genetic operators and it reiterates until the set conditions is meet. When the threshold level is exceeds the maximum level it submits cloudlets to the broker that determine the datacenter, which provide service and call to start the simulation function that finally produces the expected output. In the genetic operation step, selection of population have performed initially based on their fitness score, those with high fitness are selected for the next offspring that they have high probability to generate one or more new offspring. After selection operation, it applies crossover operation based on fuzzy set theory (we did not use the standard single point and two point crossover) in which two classes of chromosomes undergo hybridization to produce new generation that will added to the initial population for extra reproduction. Mutation changes gene values in each chromosome from its first state.

In this algorithm, we produce two classes of chromosomes that will undergo reproduction; the first chromosome is upon the cloudlet length, processing speed, and memory usage of the available computing resources. The second chromosome is based on cloudlet length and bandwidth of the virtual machines. These chromosomes are the input parameters of the proposed algorithm. In this work, cloudlets are depicting as a gene. A set of gene creates chromosomes. Computational resources have allocated to these genes of chromosomes upon their fittest processing capacity.

In this proposed algorithm, the main objective of applying fuzzy set theory is minimize the rate of iteration generating offspring and to allocate fittest resources cloudlets depend on its length. The set theory easily can determine the fittest resource to process the task because in fuzzy system it is simple to identify clear-cut fitness values and more or less types.

In computing the fitness score of genes in chromosomes, the inference system of the theory obtains the input parameters of fuzzy set theory that identifies the extent to which it goes to its appropriate fuzzy set in the membership function. There are two kinds of membership function: Mamdani and Segeno type. In this work, Mamdani type of membership function have used because of its simplicity and prerogative in determining fuzzy inference rules used for reasoning. To do these three kinds of overlapping fuzzy sets are produced that lies in between zero and one and intervals have defined in a way that the endpoint of the first fuzzy set becomes the beginning of the third fuzzy set. A membership function is a curve that specifies how every point in the fuzzy input space has drawn to a membership function. The detail of each section of the proposed hybrid algorithm present in the next section.

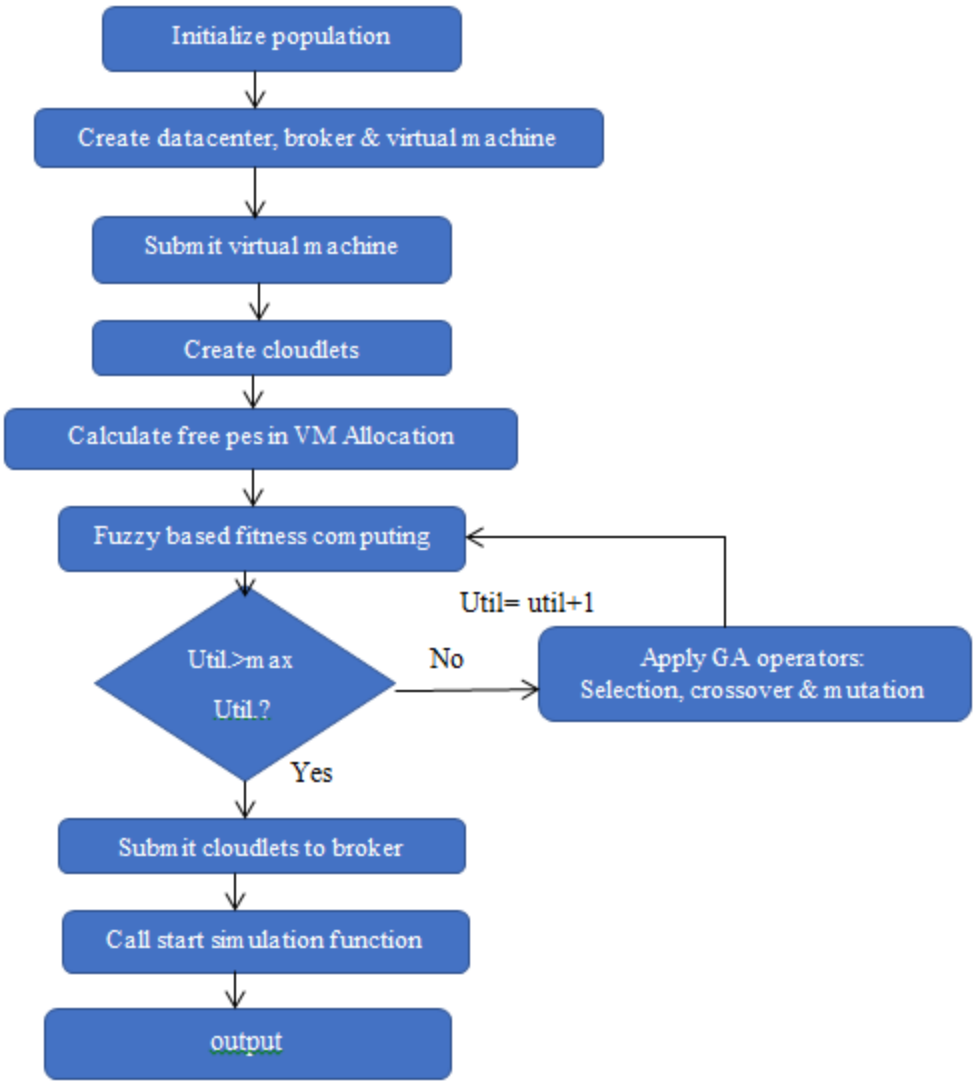

Figure: 4 Flowchart of the proposed algorithm

3.3 The Hybrid Genetic Algorithm

In this thesis, we need to follow the thought that applying the approximate optimization approach namely, GA to have used for the process of optimization. Moreover, the matching between clouds requests with cloud service providers has based on minimum response time. We also make known to the idea of a crossover operation rate as well as mutation operation rates using fuzzy set theory, which is suitable for controlling throughout the genetic algorithm optimization that will enhance solution quality.

Computational intelligence approaches like fuzzy set theory, and genetic algorithm recently gain a significant attention with targeting in improving the performance of the search algorithms

because of they deal with complex engineering issues, which are difficult to address through traditional methods. In this thesis work, we present hybrid soft computing fuzzy system support genetic algorithm approach, which makes use of the fuzzy set theory in helping the stochastic genetic search technique that can provide us a better solution for optimum resource allocation in cloud datacenter environments. Fuzzy set theory section changes the fitness score, crossover rate and mutation operation rate of evolution for each generation in the population that can lead to enhance solution quality applied to resource allocations of cloudlets. Fuzzy set theory was introduced by [48] in which the limitation set are not clearly specified, but in definite set limitations it is gradational and currently it is extensively used because of its intelligence within imprecise, uncertain knowledge and vague information and it highly used for addressing uncertainties within time. It is a computational paradigm based on how human beings think. In similar with human beings done their judgments, fuzzy set systems employed they strategy of approximate reasoning that lets it to deal with fuzzy and inadequate information. It we used in embedded, network, distributed and complex engineering.

Furthermore, [49] fuzzy set theory play an essential role in information system fields in in the development of intelligent and flexible machine interfaces and the storage of incorrect linguistic information. In addition to this, it has highly robust logical groundwork's in the classical of many-objective optimization approaches. It also has a significant advantage in the development of information technology industry. Fuzzy logic varies from the classical logic in that statements are no longer black or white, true or false, on and off. It recognizes not only the clear-cut alternatives but also the infinite gradations in between. Fuzzy logic reasoning rejects the fuzziness through allocation of a particular number to those gradations these numeric points are then employed to the correct solutions to the problems[50].

The algorithm has two computational elements performing in cooperation to produce an improved qualified solution.

- Genetic algorithm
- Fuzzy fitness finder

The pseudo code of the hybrid genetic algorithm has given below:

Algorithm1. Pseudo code of Genetic algorithm

```
Initialize (P)                          //initialize the population
Create common variables                 // datacenter, broker, VM
Submit VM to broker                     // submission of VMs to broker.
Create cloudlets                        // create required cloudlets
Calculate free Pes in VM                // calculate free processing elements
                                            In VM allocation policy
While (the termination condition satisfied) do
    Call GA which makes call to FFF     // to find the fitness value
    Check Util.?                        // confirm free available VM
    Best=Select (P)                     //select best fitness
    Crossover (P)                       //to produce new solution
    Mutation (P)                        //replace worst solution by best one
End while
Return Best                             // return best solution
End Procedure
```

The fundamental approach here is the genetic algorithm, and the fuzzy fitness finder is to have embedded within the genetic approach. To expound clearly, we will present these approaches separately as follows.

3.3.1 Genetic approach

Genetic algorithm is a rapidly growing space of evolutionary algorithm in artificial intelligence based on biological inspired evolutionary heuristic random searching method that can have generated from ecological world, which commonly applied to generate high quality solutions to optimization and searching problems by implementing genetic operators. GA can achieve optimal estimated solutions through the Darwinian law of “survival of the fittest” tasks are distributed by rendering to the value of fitness from the generations, after the initial population is produced. Selections of individual population in each generation is based on the fitness value evaluation in various individuals in definite problem domains and during various individuals are

united, hybridize and differed by genetic operators in natural genetics, a new set of population representing a new solution set is created. Solutions from one population have taken and used to generate new population. If you need to use genetic algorithm, you have to embody reply your problem as a genome. Genetic algorithm then produces a population of offspring has and applies genetic operators such as crossover and mutation to evolve the offspring has to find out the fittest. Repetitively the rule alters a population of individual solutions. Individuals from the population have selected randomly by the genetic algorithm at each step and uses and used them as parents to generate the Childs for future generation. Over serial generations, the population evolves toward a best solution.

The significance of this algorithm is that it can achieve a massive search space, used to complex objective function problems and can escape from trapping into local optimal solution [12]

The fundamental model of genetic approach has presented. It has the following main components. In order to do a better run of genetic approach, the values of the parameters of the GA have specified such as size of population, genetic operator and the terminating condition. The presentation details of each of these parts have defined in this section independently but the fuzzy fitness calculation scheme, which is FFF, is defined in the next section.

Initial population: The first population is the set of all individual that have used in the Genetic Algorithm to search out the best optimal solution. Every solution in the population has called as an individual. A set of population in an N number of individuals embraces N number of alternate solutions in the problem of addressing a cloudlet mapping of a gene. We begin with by creating initial population generation in which individuals have produced by representing cloudlets into a gene.

Selection: the selection operator determines which individual should go for offspring's in the next generation from the population.

In our approach, these have performed with help of fuzzy fitness finder by evaluating the fitness of individuals upon the GA call. In this point, the idea is that a fitness value relating to each individual is existing. Individuals with a higher fitness have selected for the next offspring in which they have a higher possibility of producing one or more offspring.

Crossover: a recombination, which two classes of chromosomes (two parents) have undergo reproduction to produce new generation. The algorithm creates two types of chromosome in the population. The first kind of chromosome is produced based on cloudlet size, virtual machine processing speed, size of the RAM the resource owns. The second types of chromosome have produced based on the cloudlet size and bandwidth of the resource. These two types of chromosomes are the input parameters of the fuzzy fitness finder part. The algorithm weighs the fitness score of the individuals in the two chromosomes through the support of fuzzy set theory. Here we do not apply the standard uniform single point or two-point crossover method because of the result are stochastic, better or worse than the previous in this way of combination. Rather we apply fuzzy set theory-based crossover which can aid the GA to meet the more targeted issue based on the fitness evaluations in every iteration.

VM4	VM3	VM1	VM2	VM8	VM6	VM8	VM7
C1	C2	C3	C4	C5	C6	C7	C8

Table 4: chromosome 1

VM2	VM8	VM5	VM1	VM3	VM4	VM6	VM7
C1	C2	C3	C4	C5	C6	C7	C8

Table 5: chromosome 2

VM2	VM8	VM1	VM1	VM8	VM4	VM8	VM7
C1	C2	C3	C4	C5	C6	C7	C8

Table 6: chromosome result

In this sample, proposed crossover the objective is to hybridize two selected chromosomes to get better offspring chromosome in which it will remain run for recombination with other

generations to get good generation. The first chromosome in table 3, is with eight chromosome genes from cloudlet C1 to cloudlet C8 that allocated to different virtual machines. In addition, in the second class of chromosome input there are eight genes from cloudlet C1 to C8 in which every cloudlet is allocated to different Virtual machines the same to in the first class of chromosome. The output chromosome result here is that it has genes allocated to more suitable or powerful virtual machines. According to a fuzzy crossover result cloudlet were re-allocated to various virtual machines. As we see in table above in the first chromosome the cloudlet C1 have allocated to VM4 and in the second chromosome, it have allocated to VM2. The output chromosome result after fuzzy crossover have allocated to VM2. In this, it allocates to computational resources that is more powerful. In the second chromosome, C2 have allocated to VM8 and in the output chromosome, it allocated to VM8. All cloudlets have allocated to VMs in the output although some VMs have rejected. The population size will remain constant subsequently the crossover operation because the gained chromosome gene is added back to the population.

Mutation: this operator has needed to respond the loss of some possibly useful genetic material during recombination and selection of individuals in the population. In the artificial chromosome, this has influenced by the infrequent random modification of the value of a string position. The mutation operator of GA has applied as: If the two tasks belong to the same membership function component and the pairwise fitness between the last gene of the first chromosome and the first gene of the second chromosome is very high then the two chromosomes merged. If the genes at the last in the chromosomes have very low pairwise fitness, the final gene has removed from the chromosome. The process of fitness evaluation, selection, crossover and mutation process will undergo until there is no a significant variation in the fitness values in the successive generations.

Algorithm1. Pseudo code of Genetic algorithm

Procedure Genetic ()

1. Initialize () //initialize the population
2. **while** (the termination condition satisfied) **do**
3. Evaluate()
4. Best=Select() //select best fitness

5. Crossover () //to produce new solution
6. Mutation () //replace worst solution by best one
7. **end while**
8. **return** Best // return best solution

End Procedure

Algorithm 1.0 expounds the main stages involved in genetic algorithm that outlines a set of solutions that in together is represent a population. The algorithm begins with initializing a population by having randomly producing a collection of solutions in line1. Then the solution is weighed by calculating the fitness function (line 3), following which the result drawn is paralleled with the current new solution. The new offspring solution has produced by performing a crossover function in line 5. The following step from the crossover step is the mutation function that has called to substitute the worst solution by the best new solution in line 6. At the end, these steps have iterated while the stop condition is met (lines 2-7).

3.3.2 Fuzzy fitness finder

Fuzzy set theory is presented by Zadeh [48] in 1965 in which the set limitations are not clearly defined. Nevertheless, it is gradational in definite limitations in that it is recently has applied in artificial intelligence without imprecise terms, uncertain knowledge and incorrect information that employed for handling uncertainties in time. Similarly, to human beings, the way they decide their decisions, fuzzy systems use the way approximate reasoning that lets it to deal with vague and incomplete information. Fuzzy set theory has used in embedded, networked and distributed systems

During the population of the GA experiences evolution at every generation, the comparatively "good" solutions reproduce and the comparatively bad solutions would "die". To determine between solutions a targeted fitness evaluation is essential and applied. In single objective problems only, a single criterion is enough for optimization but in several the real-world decision-making problems, there is a requirement of optimization for multi objective optimizations concurrently, and here is better to implement fuzzy set theory to address multi criterion optimizations.

In our proposed approach fuzzy set theory has applied in which for fitness evaluation upon the GA call to reduce iterations of generations, which is the objective of the research. Fuzzy set

system takes two classes of chromosome inputs. The first class chromosome has cloudlet length, processor speed and memory usage as input parameter. The second class chromosome has Bandwidth usage and cloudlet length as input variable. The algorithm allocates the resources to the genes randomly. After this fitness score of every gene in a chromosome is calculated using fuzzy fitness function. Those having the highest value of fitness in each chromosome have selected for crossover. After this crossover have applied based on fuzzy set theory. Moreover, the selected chromosomes again recombine and produce new offspring. Finally, the highest value chromosome have generated in the first generation.

A single criterion optimization model cannot support for the objective of fitness evaluation parameter due to we are looking for several objectives that could be drawn for solutions by winding together in the chromosome to perform hybridization. It is true fuzzy fitness finder is better applied in multiple criteria optimization problems using the genetic algorithm. In multi optimization objective the concept of optimality is not clearly determined. A solution may be best in single criteria but may not be good in multiple criteria.

The Genetic algorithm invoked upon the fuzzy fitness finder to evaluate the individual's fitness value in the population set. The solution is the mapping of the whole jobs to the presented virtual machines.

The fuzzy fitness finder is an inference system of fuzzy set theory, which has implemented in robotics navigation and control engineering systems. In general, FFF comprises the following components.

- Fuzzification
- Rule base
- Inference engine and
- Defuzzification

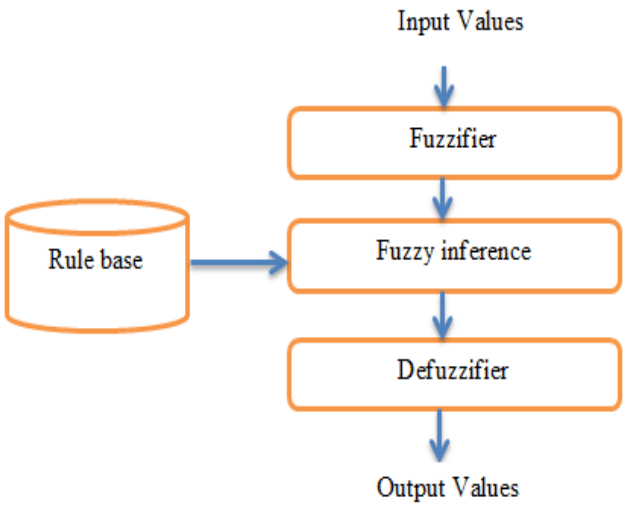

Figure: 5 structure of fuzzy inference engine

3.3.2.1 Fuzzification

First, we need to define the input parameters. The input parameters here are task or cloudlet length, bandwidth, storage and processing speed. The initial step is to take input variables for fuzzy set for fuzzification purpose. A fuzzy input variable have considered as fuzzy sets. Fuzzification of input data value determines the evaluation of the membership function of the variables in which the output result of this valuation is a membership value. After naming the fuzzy sets has created, the membership function has designed. For our proposed algorithm a Gaussian, membership function has defined with a triad (low, medium and high) in which low and high are the ends and the medium is the center of gravity of the membership function. The input variables take a range between [0 1] fuzzy sets with their linguistic explanations and the value of the triad (low, medium and high) which determines fuzzy sets. In the fuzzification process, the Y-axis of the graphs shows the fuzzy sets of the input variables between [0 -1] with their corresponding input parameter values in the X-axis triad membership values of low, medium and high.

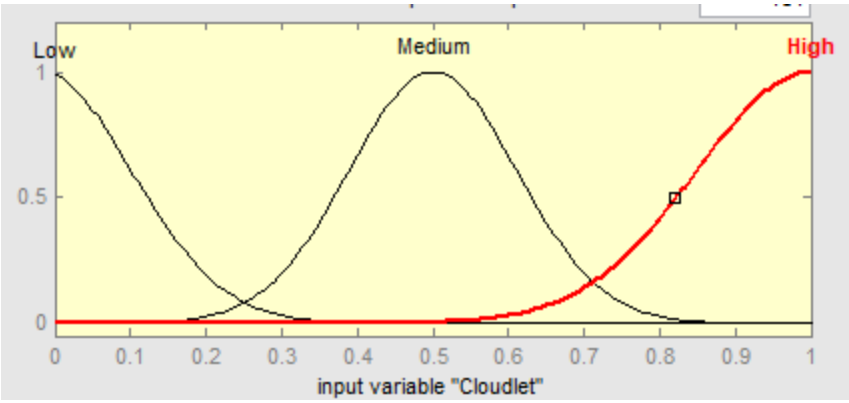

Figure: 6 Fuzzy sets of cloudlet length

In computing the fitness score, the fuzzification system applied membership functions to define the input variables that have its place to every fuzzy set of the Gaussian membership function. For this objective three overlying sets have produced. For example, the cloudlet variations values from 0 to 0.3 are in the low range, values between 0.2 to 0.8 are the medium range and variable values from 0.6 to 1 are the high range of the fuzzy set in membership function.

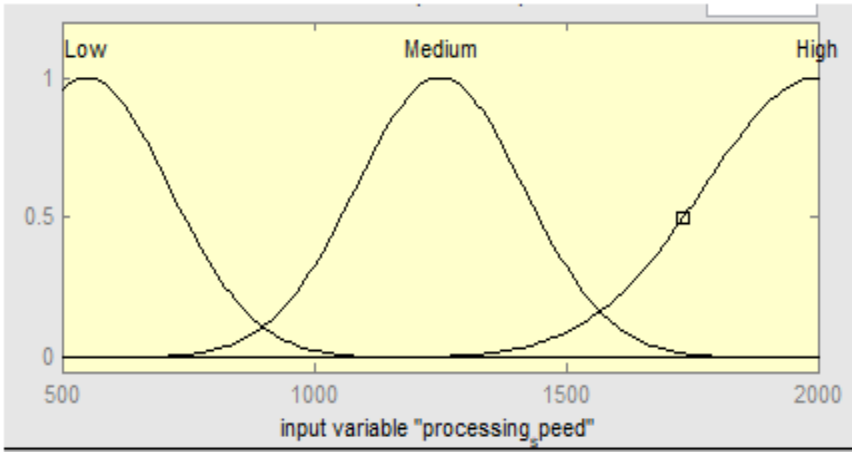

Figure: 7 Fuzzy sets of VM processing speed

In this membership function, the input parameter of virtual machine processing speed is from 500 to 2000. The processing speed value 700 has degree values of membership fuzzy set on the 0.3 in the low-level range, 0.7 in the mid-range and 1 high level fuzzy set membership function.

This value has applied in the fuzzy reasoning, which provides basis for decision-making or pattern recognition. In our proposed work, the two types of chromosomes are the input parameters in the fuzzy inference system and the value have taken from these variables to assign powerful computational resources to various instructions, which have varied in their length. The low-medium-high inference membership sets have used to coordinate the output values of the inference rule. In our proposed work the rule base organization of the fuzzy set theory is a series of IF-THEN rules are determined for the result reply presented in the input situations, in which the rules are a set of semantic control rules that have needed to realize control target systems. In this work in first chromosome 18 and in the second input chromosome 9 logical rules used in the inference system.

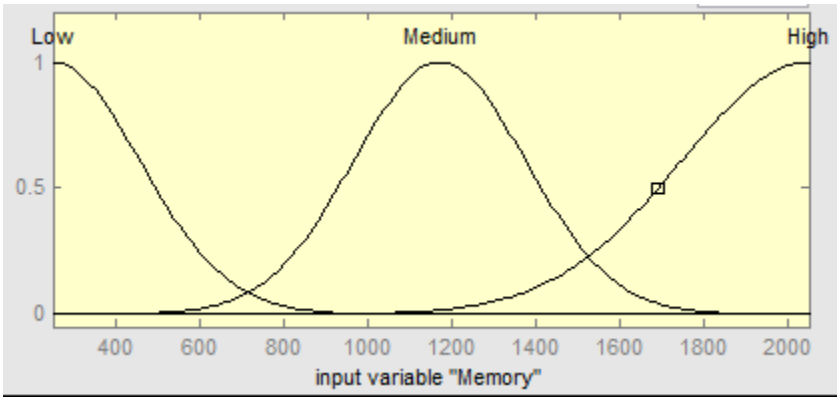

Figure: 8 Fuzzy sets of Memory size

3.3.2.2 Fuzzy inference engine

Now the output value of the fuzzified process of various input parameter have presented. Every parameter goes to one or more fuzzy sets with non-zero chance based upon overlapping and non-overlapping parts of fuzzy sets. The main target of the inference engine is to associate the effects of different input parameter existed as fuzzy input to evaluate the fitness function of the chromosomes using fuzzy rules. The fundamental steam is resulting from the rule base, which consists a set of semantic rules that are defined in the form of IF-THEN rules. These rules expound the subsequent of the design based on the given linguistic variable such as low, medium and high. The inference engine uses IF-THEN rules defined in the rule base consisted from a semantic control rules and accompanying control objectives in the system and based on that it reaches at the fuzzy output from a given input mapping, which offers a basis from which a pattern is recognized or decisions are made.

Moreover, it specifies the type and amount of the involved membership functions of input and output parameters. The result is a value that can have taken as the measurement of the fitness of the chromosomes. Here is the format of the rule:

Number of variables =2

Variable 1 may take values in the fuzzy set low, medium and High
Variable 2 may take values in the fuzzy set low, medium and High
Variable 3 may take values in the fuzzy set low, medium and High

Let the result or fitness take values in the fuzzy set as insufficient, medium and suitable. Rules in the rule base of the fuzzy set are the grouping of two or more antecedent and consequent. Some of the rules in our work have presented as follows:

IF variable 1 is medium **or** variable 2 low **or** variable 3 low **THEN** Fitness is insufficient.
IF variable 1 is medium **or** variable 2 low **or** variable 3 low **THEN** Fitness is medium.
IF variable 1 is medium **or** variable 2 low **or** variable 3 low **THEN** Fitness is suitable.

Based on the above rule instruction the following 18 rules we tried to generate in the first input chromosomes and in the second chromosome, we create 9 semantic rules which are used in inference engine.

1. If cloudlet length is low or processing speed is low, or memory is low then result is insufficient.
2. If cloudlet length is low or processing speed is medium, or memory is medium then result is medium.
3. If cloudlet length is low or processing speed is medium, or memory is high then result is medium.
4. If cloudlet length is low or processing speed is high, or memory is low then result is medium.
5. If cloudlet length is low or processing speed is high, or memory is medium then result is insufficient.

6. If cloudlet length is low or processing speed is high, or memory is high then result is suitable.
7. If cloudlet length is medium or, processing speed is low, or memory is low then result is insufficient.
8. If cloudlet length is medium or processing speed is low, or memory is medium then result is insufficient.
9. If cloudlet length is medium or processing speed is medium, or memory is medium then result is insufficient.
10. If cloudlet length is medium or processing speed is medium, or memory is low then result is insufficient.
11. If cloudlet length is medium or processing speed is high, or memory is low then result is suitable.
12. If cloudlet length is high or processing speed is low, or memory is low then result is suitable.
13. If cloudlet length is high or processing speed is low, or memory is medium then result is suitable.
14. If cloudlet length is high or processing speed is low, or memory is high then result is medium.
15. If cloudlet length is high or processing speed is medium, or memory is low then result is medium.
16. If cloudlet length is high or processing speed is medium, or memory is medium then result is insufficient.
17. If cloudlet length is high or processing speed is medium, or memory is high then result is insufficient.
18. If cloudlet length is medium or processing speed is high, or memory is medium then result is medium.

In the above semantic rules, three variables are determined in the fuzzy rule set in the first input chromosome which are cloudlet size, processing speed and the memory size which are defined in the membership function of triad value (low, medium and high) used in the fuzzy inferencing engine for reasoning.

1. If cloudlet length is low or bandwidth is low then the result is insufficient.
2. If cloudlet length is low or bandwidth is medium then the result is moderate.
3. If cloudlet length is low or bandwidth is high then the result is suitable.
4. If cloudlet length is medium or bandwidth is low then the result is moderate.
5. If cloudlet length is medium or bandwidth is medium then the result is insufficient.
6. If cloudlet length is medium or bandwidth is high then the result is medium.
7. If cloudlet length is high or bandwidth is low then the result is suitable.
8. If cloudlet length is high or bandwidth is medium then the result is insufficient.
9. If cloudlet length is high or bandwidth is high then the result is insufficient.

In the above semantic rules, the input variables in the second chromosome are cloudlet size and bandwidth with their membership function. These chromosomes are would be used in the fuzzy based crossover to produce new offspring generation.

The input and output variable to fuzzy rule bases are fuzzy set variables. Each crisp input variable has non-zero membership values in one or more fuzzy set due to the overlap of the Gaussian membership function. This leads to several inputs of rules being had initiated which result diverse output fuzzy set to fire.

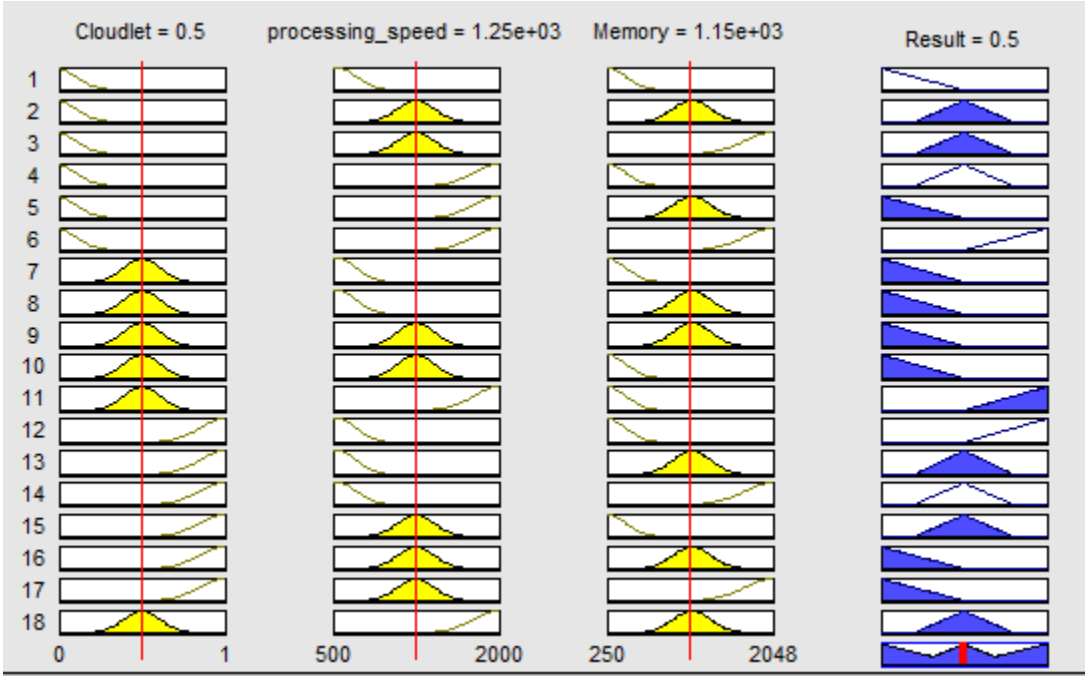

Figure: 9 Fired aggregation of fuzzy set rule

The fire aggregation operation of fuzzy set produces a suitable fuzzy single set by combining different input parameters of the fuzzy inference system, in the above figure 13 the blue color is the aggregated of the cloudlet size, processing speed of virtual machines and memory size of the first chromosome in the fuzzy input parameters.

3.3.2.3 Defuzzification

Defuzzification process is a way of representing back the output crisp values, which is an inverse transformation of that maps the result from the fuzzy set to into the crisp value. The defuzzification of the result output crisp value is achieved through associating the result of the fuzzy inference way and calculating the fuzzy centroid of the area. Defuzzifier embraces the collected linguistic values from the latent fuzzy controller and generates a non-fuzzy control result, which represents the balanced load adapted to load conditions. The defuzzification process is utilized to weight the membership function for the accumulated output[51].

3.4 Development tool

A research approach that we will use in this thesis work consists of fundamental theoretical concepts with experimental simulation through programming languages. During studying the scalability and large size of the cloud computing environment, experimental simulation can help in testing the proposed approaches in small measure environment.

In the research work, a simulation java run time environment will be used for realizing load balancing by minimizing response time and processing time of the data center in the cloud environment through fuzzy set theory guided- genetic algorithm design. Java programming language is a high-level object-oriented programming language with CloudSim library that will helps developers by delivering useful libraries and classes for easier simulation.

The proposed load balancing algorithm in this thesis work have performed through CloudSim. It is an open source environment, which initially have developed for cloud-based case studies in university of Melbourne, Australia. CloudSim comprises several modules and design graphics, which makes the cloud infrastructure modeling easier [52], [53]. Prior to applying large environment, various cloud enterprises simulate their idea using CloudSim. HP is among those industries that used CloudSim for their research findings[47].

The CloudSim toolkit can support the simulation and modeling of both cloud computing system parts like virtual machines, datacenter, and resource provisioning policies. Moreover, currently CloudSim is an extensible toolkit and can also supports the simulation of cloud computing environments including both single and inter networked clouds (Federation of Cloud) consisting dedicated management interfaces for VM's, memory, storage and bandwidth. The basic hurdles of provisioning of hosts to VMs, managing application execution, and monitoring dynamic system state can be easily handle through this architecture.

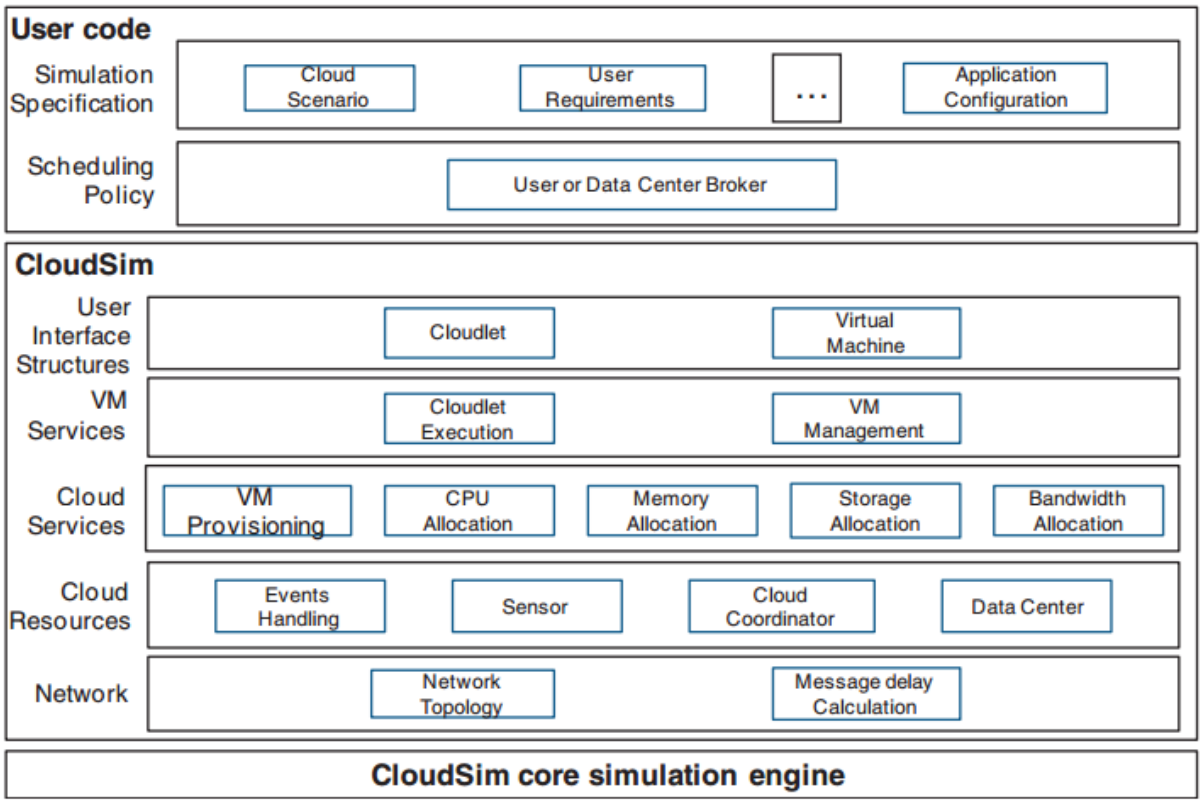

Figure : 10 CloudSim architecture [54]

In the CloudSim is a layered architecture that comprises several components or classes to implement cloud-based provisioning policies. The various layers of CloudSim have shown above in the figure. These layers of CloudSim have responsibility to do communications possible between layers. Here we need to provide a finer highlight of the fundamental classes of the architecture.

Cloudlet: this abstract class determines the set of user requests. It contains user identification, name of the user base that is the source, which a response route back.

Cloudlet Scheduler: this class provides the execution of various provisioning policies, which determines the processing power shared among cloud requests in virtual machines.

Datacenter: this class of CloudSim models the infrastructure level services, which are offer through various cloud service providers (Amazon, Azure, and App Engine).

Cloud broker: the cloud broker specifies which datacenter should provide the service for a particular cloudlet that emanates from the user base.

Datacenter characteristics: this abstract class models the information regarding with datacenter resource configuration such as host, network topology, RAM provisionary and sensor.

Vm: this class decides a virtual machine that can control and host a cloud host parts. Each virtual machine has access to memory, storage size, processor and the Vm is provisioning strategy that has extended from cloudlet scheduler.

VM allocation policy: this abstract class models provisioning strategies on the mechanisms how to allocate hosts to virtual machines. It can determine the available hosts in a datacenter that realize availability and storage for virtual machine implementation.

CHAPTER FOUR

RESULTS AND DISCUSSION

In this chapter, we will present the experimental simulation results by comparing with other load balancing algorithms in the cloud-computing environment based on some evaluation parameters. We proposed an efficient load balancing which are upon effective utilization of computing services that can remove the challenge of resource over provisioning and under provisioning based on Virtual machine time-shared provisioning policy. This efficient load balancing has networked to all hosts and customers. The host controller that provides the service manages virtual machines. Our proposed algorithm is efficient in utilizing the service resources due to the algorithm allocates computational resources to tasks is through by considering the task length (task variations). Task with high length has allocated to a resource, which are powerful to perform the task in minimum execution time.

4.1. Expression of load

The whole virtual machine running on a physical machine at a particular virtual server has defined as the load of the physical machine. Mathematically it can be express as follows.

Imagine that there is n number of cloudlets required to have allocated specified as:

$$C = \{C1, C2 \dots \dots \dots \dots Cn\}$$

In addition, there are k numbers of virtual machines in a datacenter.

$$V = \{V1, V2 \dots \dots \dots \dots Vk\}$$

The recent cloudlet load on the datacenter then is:

$$DC = \{VC1, VC2 \dots \dots \dots \dots Vk\}$$

Then we require getting a function f(C), in which the cloudlet C needed to have allocated to virtual machines V, that would make the load VC of every virtual machine V, is necessarily equal, that is:

$$VC1 \approx VC2 \approx \cdots \dots \dots . . \approx VCk$$

4.2 Response time

Response time is the amount of time elapsed to respond for a task in a system in a particular load balancing algorithm[55]. It is a warrantee of service level agreement between the service vendor and who rent for service for better quality of service because it have an impact on the payment mode, in that users are require a minimum cost with good service performance. A response time have computed through the following mathematical formula:

$$RT = Tfin - Tarr + Tdelay$$

Where T_{fin} is the execution final time of the user cloudlet, T_{arr} is the arrival time of the user cloudlet and T_{delay} is the transmission delay time.

The transmission delay time have calculated with the help of the following formula:

$$Tdelay = Tlatency + Ttran$$

Where $T_{latency}$ is the transmission latency time, T_{tran} is the transfer time of the data of a single cloudlet from a user to its destination.

$$Ttransfer = T/BWperuser$$

Where T is a single task or request that sent to destination for processing, and BWperuser can be computed as BWperuser = BWtotal/N where BWtotal is the entire bandwidth existed (specified in the characteristics), N is the number of requests currently in transmission. The cloud characteristics have consisted the number of user requests in transmission.

4.4 Simulation setup

In this work, we specified 50 virtual machines in datacenters and the size employed to host the application is 1000 MB in the experiment. Virtual machines would have from 256-2048 MB RAM memory with 500-1000 MB of existing bandwidth. Hosts in simulation have x86 system architecture, virtual machine monitor “XEN”, and Linux operating system the hosts have 10048 MB RAM memory, 12000 million instructions per second, 100 GB host storage and 100000 available bandwidths. Every virtual machine has processors having various kinds of CPU and speed. The experiment considers the effect of the CPU processing capacity with its processing

speed. The algorithms perform by effectively allocating cloudlets to appropriate resource based on based on high resource capacities and high bandwidth for better utilization of computational resources and it minimizes the available free RAM. Those instructions, which weigh more spans that have executed, would distribute to resources that have high bandwidth, good storage capacity, processing speed of the CPU. In the simulation experiment, the VM policy has based on timeshared in which, the resources have shared among the instructions. Every instruction would get a resource for execution to a specific period and after they get finished, it is just have released and allocated to other instructions.

The following is the CloudSim toolkit-based experiment simulation configuration for the algorithm as shown in table. The result of the simulation of the proposed algorithm is based the resource capacity and instruction weight is compared with other load balancing algorithms

Datacenter	#VM	Image size	Memory	bandwidth	Mips	VM policy
10	50	1000	128-2048	50-1000	100-2000	Timeshared

Table 7: Experiment configuration

Mips	Host storage	RAM	Bandwidth
12000	100 GB	10048	100000

Table 8: Host Configuration

4.5 Resource allocation

Resource allocation is a way of distributing available computational resources to the clients according to their demand. Resource provisioning strategies would solve the allocation problem by allowing the service providers to monitor for individual requests of resource. Resource provisioning strategy is all about the amount of cloudlet requests for assigning and utilizing inadequate resources within the limit of the cloud environment to realize the demands of the cloud-based applications. It needs the type and amount of the computational resource required by each application in order to complete the user task. In this research work, the resource provisioning policy has done based on the computational resource processing capacity, such as the processing capacity in the virtual machines in their RAM. CPU and bandwidth. Allocation to

user request is depends on the size of the instruction of and the fittest processing power of the virtual machine is considered.

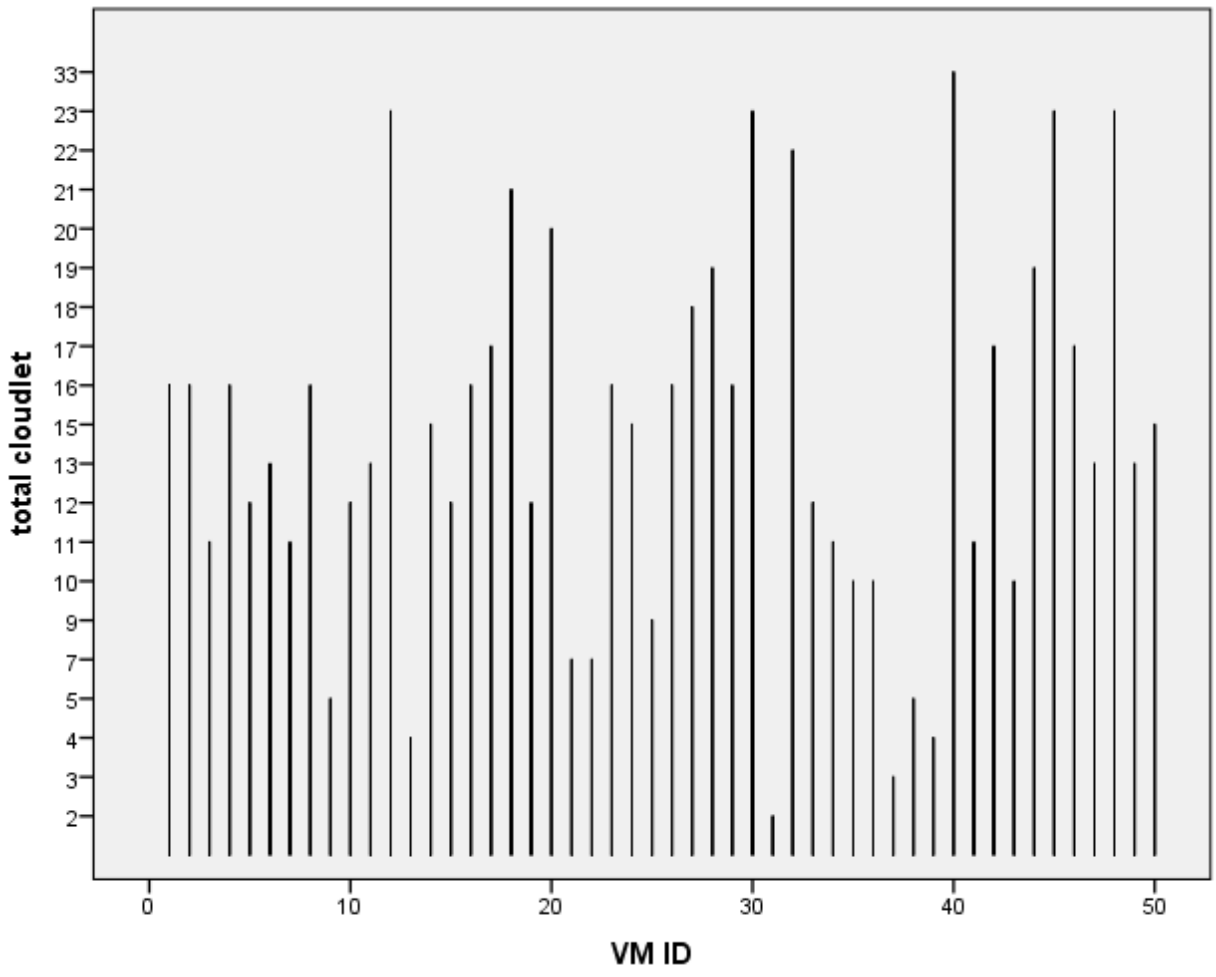

Figure: 11 Resource allocations

As we have observed on the above figure 11, we can look more cloudlets are allocated to that of virtual machines that have qualified processing power and capacity than the machine with less processing power machines. When a VM is more suitable and powerful to process instructions, it executes more tasks than those machine with less suitable to run tasks. The evaluation to allocate computational resource to instructions have based on their suitability to run it. As a result, task allocation is allocation have balanced and enhanced throughout the hosts. The proposed algorithm adds a substantial enhancement in virtual machine allocation to instructions to reduce

load inequity, execution time and response time. This would help to improve the performance of the diversified cloud computing environment. Currently resource allocation in cloud datacenter targets in offering high performance while meeting the service level agreement with limited or no consideration of energy consumptions during virtual machine allocations[56].

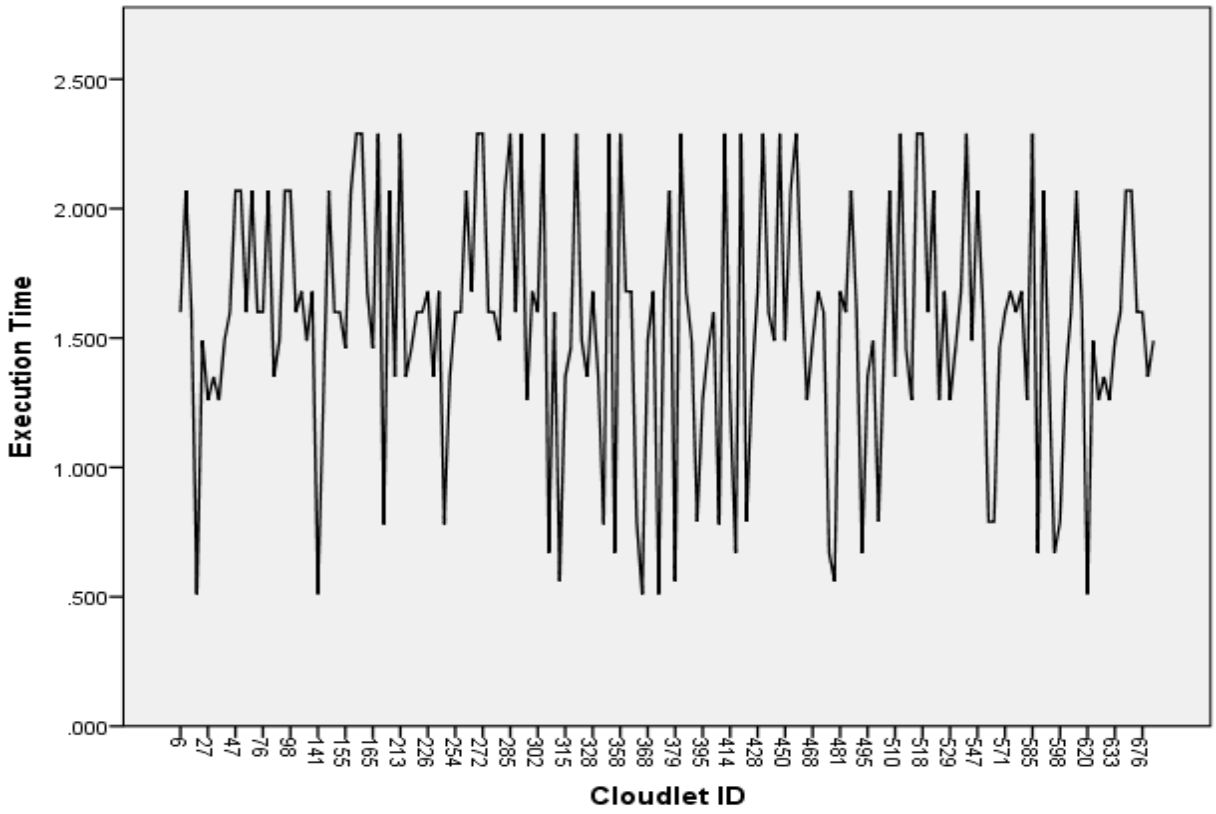

Figure: 12 Execution time of cloudlet

As we have observed from the figure above, because of the instructions are allocated to virtual machines based on their length and VMs qualification, processing capacity, the execution time of the instruction is minimized. The instruction which have high length would assign to a virtual machine suitable to it, that is enough sufficient to process the task. Most of the tasks have executed below the execution time of 2.00 milliseconds. Those have less weight would execute in less time than those have high instruction size. Here the assumption is with no network delay during the running of the instructions.

4.6 Performance evaluation and analysis

The performance of the proposed algorithm FGA have conducted and analyzed based on the simulation result done using the CloudSim via Java programming language in Net Beans IDE. This simulation mainly evaluates the execution time, makespan, imbalance factor and resource utilization in the proposed load balancing algorithm in cloud computing environment. The performance of this work under these evaluation metrics have compared with the standard genetic algorithm. The time units in the performance evaluation parameters are in Milliseconds.

4.6.1 Execution Time

Various types of resource allocation is approaches were presented [57]. In [58] execution time is well-thought-out in resource allocation strategy, it can address the involved issues of contention of computational resource and can maximize resource utilization by applying various kinds of approach of renting computing capacities. Execution time is the elapsed times of the cloudlet from submitting to it have completed.

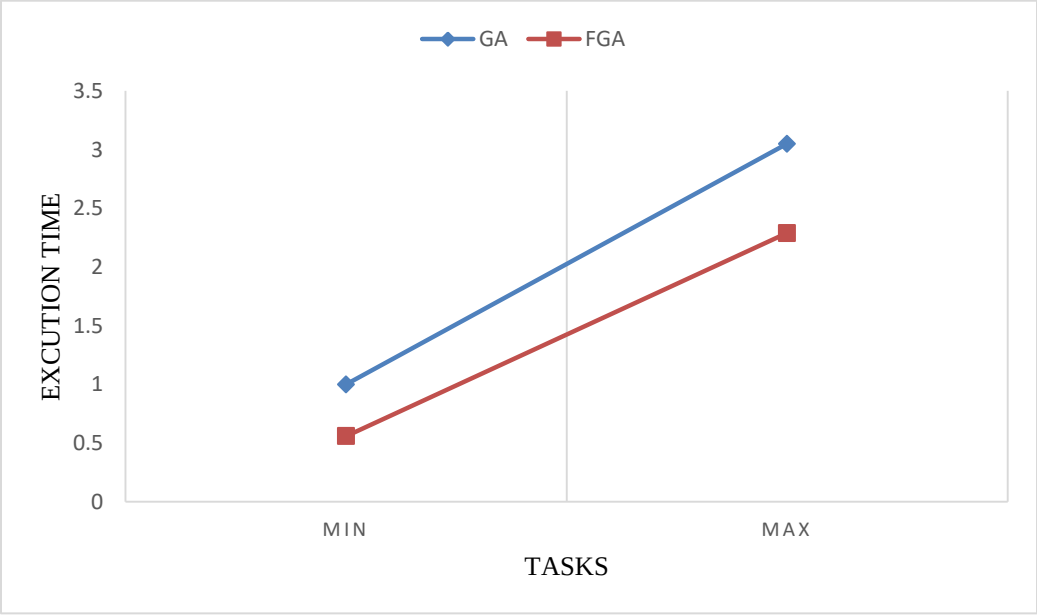

Figure: 13 Execution time Vs. Tasks

4.6.2 Load imbalance

This is an aspect, which has related to determine load balancing in virtual machines. It can determine load among the virtual machines. This can measured using the execution time of

cloudlets in virtual machines. A less imbalance factor value shows good load balancing in VM's. This can compute through the following formula.

$$IF = \frac{Tmax+Tmin}{Tavg}$$

Where IF, is load imbalance factor, $Tmin$ minimum execution time, $Tmax$ maximum and $Tavg$ average execution time of all virtual machine. In the following figure 14, the fuzzy guided genetic algorithm has lower imbalance degree than the standard genetic approach. The FGA with less imbalance value is less loaded of jobs in VMs. Load imbalance degree in needed to identify the allocation of jobs in virtual machines.

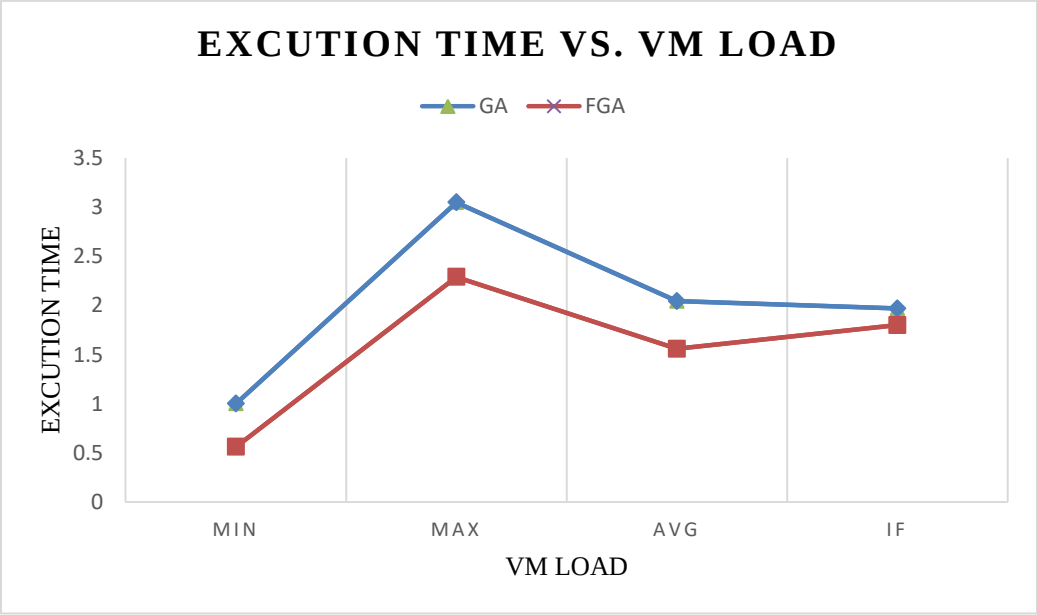

Figure: 14 VM load imbalance

4.6.2 Makespan

Makespan is the total time has needed to finish all the cloudlets submitted to the system. Makespan of the system is the maximum time taken by a host running in a particular datacenter[15]. This performance evaluation parameter should be decreased to minimize execution time and cost[59][60]. It can compute using the following formula.

$$Makespan = \sum_{i=0}^{n} Executiontime(Tasks)$$

In the following graph 15, the (Y-axis) represents the completion time of the final task in the execution of jobs within their corresponding virtual machine in the (x-axis). As the proposed hybrid algorithm, assign tasks to the available machine based on their fitness value its completion time is minimum than the genetic algorithm. As the number of VM increases the makespan time of tasks will have reduced.

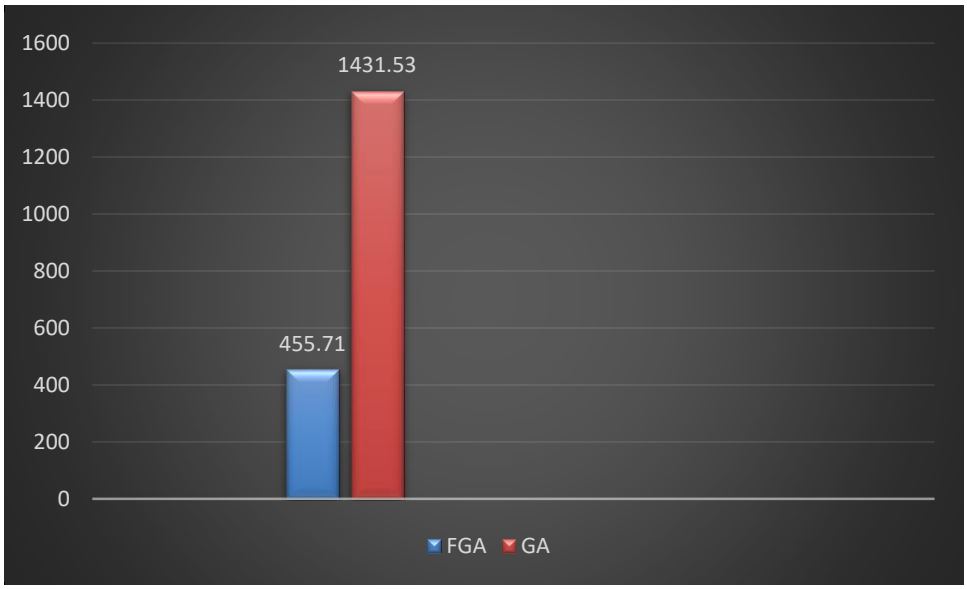

Figure: 15 Makespan

4.6.3 Resource utilization

Increasing resource utilization is an essential objective of load balancing in cloud-based systems[61]. This metrics is gaining importance because of the financial aspect of cloud system such as the contribution of infrastructure capitals by organizations and individuals in exchange for economic benefits. Realizing maximum resource utilization is a challenge in cloud environment due to the diversified heterogeneity of computational resources across the cloud computing system. Resource utilization is measured by using the following Equation [62].

$$\text{RUT} = \left(\frac{meantime}{makespan} * 100\right)$$ Where meantime is the $\sum$ processing time resource VM to execute all allocated cloudlets and the range of average resource utilization is from 0-1. 1 depicts the maximum (100%) utilization whereas 0 indicates the idle state.

Resource utilization is to use to find the capability of the cloud system to help in the course of utilization of resource parameters such as virtual machine list, task lists executed in the VMs and

their corresponding time needed for processing tasks. The FGA offers better resource utilization by instruction size, which has a quick processing time in the distributed cloud system and homogeneous tasks. The algorithms take into account instruction or cloudlet size along with the capacity of processing heterogeneous VMs to assign tasks, so a greater number of cloudlets are assigned to higher processing capacity in terms of CPU, RAM, and bandwidth of virtual machines in homogeneous tasks within distributed based cloud systems environments. This helps to in realizing or completing the cloudlets execution time in shorter time. The load balancing capability of nodes can be determined through resource utilization. In the following figure 16, FGA have better utilization than the GA because in the proposed hybrid algorithm all the available machines have utilized as it initially checks free available VM and allocate a particular task to the VM. In addition, if not all resources are free it calculates the free processing elements and assign task based on the shorter completion time of the machine. As a result, all available machines have effectively utilized in the proposed algorithm. The main target of this thesis work is to maximize utilization of available computational resources efficiently.

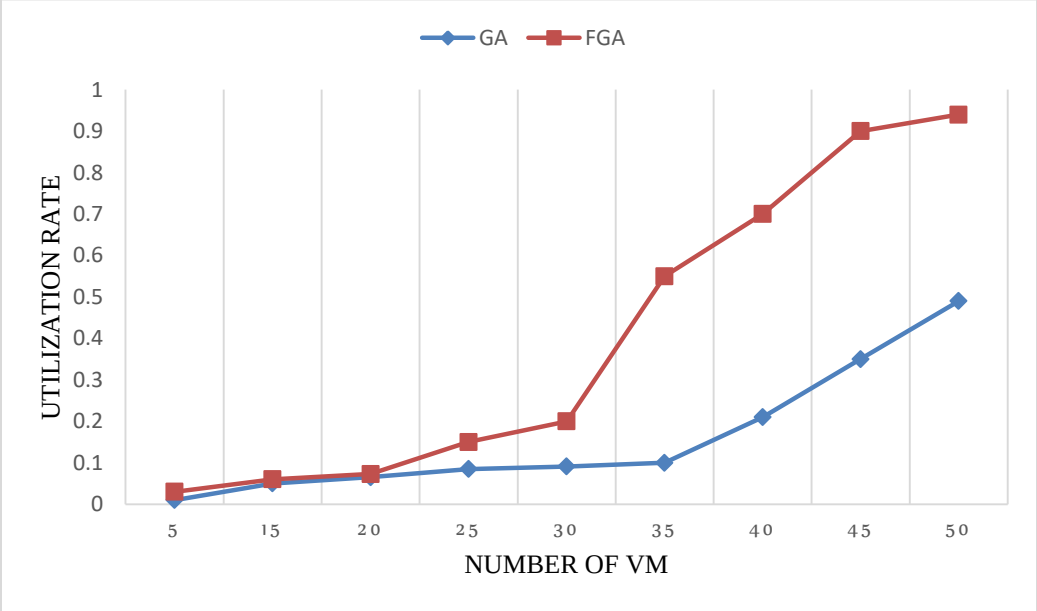

Figure: 16 Resource utilization

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 conclusion

In this thesis work, we tried to explore the current experiences of the cloud computing technology. In this thesis work, the main concept of load balancing in cloud-based systems is the primary research theme. The main involved problem in cloud-based system is load balancing and resource utilization, in which some nodes become overloaded in that computational resources are over utilized as a result response time increases and nodes may be underutilized or idle in which nodes ere under the threshold value of using computational resource and it increases in using power consumption. As a result, load balancing of systems should be efficient to improve performances of the cloud-based technology. The existing load balancing algorithms are not efficient to utilize resources efficiently and reduce execution time at the same time. Due to the limited amount of resources in the cloud environment and resources should have not be idle because of when they are free still they consume power. To remove this issue resources have needed to utilize effectively. Therefore, we tried to present a hybrid algorithm that takes the advantages of the two algorithms through considering the VMs processing speed, memory usage, cloudlet variations and bandwidth of the VMs, by having these parameters the algorithms targets to allocate computational resources to cloudlets based on their length variations. This is because of virtual machines have various processing capacity in distributed system. The main objective of this research work is to decrease the total instruction execution time.

In the hybrid fuzzy genetic approach, when VMs have disseminated over datacenters it is based on the hosts' requirement of qualification and processing capacity, the datacenter, which have powerful capacity, will have more virtual machines than a datacenter with less powerful capacity. As a result, VMs are selected based on their power of processing tasks to allocate instructions for processing, therefore the response time is enhanced due to more tasks are on the qualified virtual machines.

The simulation is experimented using CloudSim simulator and the analysis of the result shows that the execution time of the cloudlets has reduced as compared to the standard genetic algorithm. According to our proposed algorithm of load balancing resource allocation strategy is

based on utilization of computational resources would lead to better energy efficient because idle node consume power with zero resource utilization.

5.2 Recommendation

In cloud-based systems load balancing can have considered as the most vexed challenge in which it is the main aspect in enhancing the cloud computing system performances. We discussed in enhancing the performance of virtual machines distributed on various hosts in considering processing speed, memory size and task length, but there are still parameters which are needed to be considered to balance virtual machine loads like power consumption and dynamic resource utilization. We are going to contrivance a hybrid fuzzy guided genetic algorithm to enhance the virtual machine allocation policy. We tried to discuss load balancing in a normal state but there are other states needed to have studied in future such as bursty workload state. In addition, it is needed to discuss on how can address the issues of deadlocks and server overflow issues in VMs in future.

References

[1] S. G. Domanal and G. R. M. Reddy, “Optimal load balancing in cloud computing by efficient utilization of virtual machines,” in *2014 6th International Conference on Communication Systems and Networks, COMSNETS 2014*, 2014.

[2] H. J. Younis, A. Al Halees, and M. Radi, “Hybrid Load Balancing Algorithm in Heterogeneous Cloud Environment,” no. 3, pp. 61–65, 2015.

[3] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, “Cloud Computing: Distributed Internet Computing for IT and Scientific Research,” *IEEE Internet Comput.*, vol. 13, no. 5, pp. 10–13, Sep. 2009.

[4] R. Buyya, J. Broberg, and A. Goscinski, *Cloud Computing: Principles and Paradigms*. 2011.

[5] “What is cloud computing? A beginner’s guide | Microsoft Azure.” [Online]. Available: https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/. [Accessed: 24-Dec-2019].

[6] B. Sosinsky, *The book you need to succeed! Cloud Computing*. 2011.

[7] M. Armbrust, Michael and Fox, Armando and Griffith, Rean and Joseph, Anthony D. and Katz, Randy and Konwinski, Andy and Lee, Gunho and Patterson, David and Rabkin, Ariel and Stoica, Ion and Zaharia, “Above the clouds: A Berkeley view of cloud computing,” *Univ. California, Berkeley, Tech. Rep. UCB*, pp. 07–013, 2009.

[8] S. Swarnakar, Z. Raza, S. Bhattacharya, and C. Banerjee, “A novel improved hybrid model for load balancing in cloud environment,” *Proc. - 2018 4th IEEE Int. Conf. Res. Comput. Intell. Commun. Networks, ICRCICN 2018*, pp. 18–22, 2018.

[9] A. Khiyaita, H. El Bakkali, M. Zbakh, and D. El Kettani, “Load balancing cloud computing: State of art,” *Proc. 2nd Natl. Days Netw. Secur. Syst. JNS2 2012*, pp. 106–109, 2012.

[10] M. Vanitha and P. Marikkannu, “Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines,”

Comput. Electr. Eng., vol. 57, pp. 199–208, 2017.

[11] S. Singh and I. Chana, "A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges," *J. Grid Comput.*, vol. 14, no. 2, pp. 217–264, Jun. 2016.

[12] K. Dasgupta, B. Mandal, P. Dutta, and J. Kumar, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing," *Procedia Technol.*, vol. 10, pp. 340–347, 2013.

[13] M. Rana, S. Bilgaiyan, and U. Kar, "A study on load balancing in cloud computing environment using evolutionary and swarm based algorithms," *2014 Int. Conf. Control. Instrumentation, Commun. Comput. Technol. ICCICCT 2014*, pp. 245–250, 2014.

[14] S. F. Issawi, "Efficient Adaptive Load Balancing Algorithm for Cloud Computing Under Bursty Workloads," vol. 5, no. 3, pp. 795–800, 2015.

[15] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A big picture," *J. King Saud Univ. - Comput. Inf. Sci.*, 2018.

[16] M. Rahman, S. Iqbal, and J. Gao, "Load balancer as a service in cloud computing," *Proc. - IEEE 8th Int. Symp. Serv. Oriented Syst. Eng. SOSE 2014*, pp. 204–211, 2014.

[17] N. Jain and I. Chana, "Cloud Load Balancing Techniques: A Step Towards Green Computing," *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 238–246, 2012.

[18] Ari Liberman García, "THE CLOUD The Evolution of The Cloud," p. 9, 2013.

[19] S. Aslanzadeh, "Anticipatory Models of Load Balancing in Cloud Computing," *Australia*, 2016.

[20] P. M. Mell and T. Grance, "The NIST definition of cloud computing," Gaithersburg, MD, 2011.

[21] Y. Balagoni and R. R. Rao, "Importance of Load Balancing in Cloud Computing Environment: A Review," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 3, no. 5, pp. 77–82, 2014.

[22] J. L. Shah, “International Journal of Advances in Computer Science and Technology Cloud Computing : The Technology for Next Generation,” vol. 3, no. 3, pp. 152–155, 2014.

[23] G. Galante and L. C. E. De Bona, “A survey on cloud computing elasticity,” in *Proceedings - 2012 IEEE/ACM 5th International Conference on Utility and Cloud Computing, UCC 2012*, 2012, pp. 263–270.

[24] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, “Key challenges in cloud computing: Enabling the future internet of services,” *IEEE Internet Comput.*, vol. 17, no. 4, pp. 18–25, 2013.

[25] V. SureshKumar and M. Aramudhan, “Performance Analysis of Cloud under different Virtual Machine Capacity,” *Int. J. Comput. Appl.*, vol. 68, no. 8, pp. 1–4, Apr. 2013.

[26] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing*. Elsevier Inc., 2013.

[27] A. Al-Maamari and F. A. Omara, “Task Scheduling using Hybrid Algorithm in Cloud Computing Environments,” vol. 17, no. 3, pp. 96–106.

[28] . P., “Load Balancing Algorithms in Cloud Computing Environment,” *Int. J. Adv. Res. Comput. Sci.*, vol. 9, no. 2, pp. 397–401, 2018.

[29] A. Kaur Sidhu, S. Kinger, A. Professor, and S. Guru Granth, “Analysis of Load Balancing Techniques in Cloud Computing,” vol. 4, no. 2.

[30] A. M. Alakeel, “A Guide to Dynamic Load Balancing in Distributed Computer Systems,” 2010.

[31] H. B. Hitesh Bheda, “An Overview of Load balancing Techniques in Cloud Computing Environments An Overview of Load balancing Techniques in Cloud Computing Environments,” *Int. J. Eng. Comput. Sci.*, vol. 4, no. JANUARY, pp. 9874–9881, 2015.

[32] A. Y. Hamo and A. A. Saeed, “Towards a Reference Model for Surveying a Load Balancing,” 2013.

[33] “Cloud Load Balancers.” [Online]. Available: https://www.w3schools.in/cloud-

computing/load-balancing/. [Accessed: 14-Oct-2019].

[34] S. Javanmardi, M. Shojafar, D. Amendola, N. Cordeschi, H. Liu, and A. Abraham, “Hybrid Job scheduling Algorithm for Cloud computing,” pp. 1–10, 2014.

[35] K. Nishant *et al.*, “Load Balancing of Nodes in Cloud Using Ant Colony Optimization,” in *2012 UKSim 14th International Conference on Computer Modelling and Simulation*, 2012, pp. 3–8.

[36] J. Ni, Y. Huang, Z. Luan, J. Zhang, and D. Qian, “Virtual machine mapping policy based on load balancing in private cloud environment,” in *2011 International Conference on Cloud and Service Computing*, 2011, pp. 292–295.

[37] S. S. Narayanan, M. Ramakrishnan, and M. Saadique Basha, “Efficient Load Balancing Algorithm For Cloud Computing Using Divisible Load Scheduling And Weighted Round Robin Methods,” *Adv. Nat. Appl. Sci.*, vol. 11, no. 1, 2017.

[38] L. D. Dhinesh Babu and P. Venkata Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments,” *Appl. Soft Comput. J.*, vol. 13, no. 5, pp. 2292–2303, 2013.

[39] O. A. Rahmeh, P. Johnson, and A. Taleb-Bendiab, “A dynamic biased random sampling scheme for scalable and reliable grid networks.” 2008.

[40] M. Randles, D. Lamb, and A. Taleb-Bendiab, “A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing,” in *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, 2010, pp. 551–556.

[41] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg, “Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services,” in *Performance Evaluation*, 2011, vol. 68, no. 11, pp. 1056–1071.

[42] J. M. Galloway, K. L. Smith, and S. S. Vrbsky, *Power Aware Load Balancing for Cloud Computing*. .

[43] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, "LBVS: A load balancing strategy for virtual storage," in *Proceedings - 2010 International Conference on Service Science, ICSS 2010*, 2010, pp. 257–262.

[44] S. Sethi, "Efficient load Balancing in Cloud Computing using Fuzzy Logic," *IOSR J. Eng.*, vol. 02, no. 07, pp. 65–71, 2012.

[45] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *Proceedings - 3rd International Symposium on Parallel Architectures, Algorithms and Programming, PAAP 2010*, 2010, pp. 89–96.

[46] Y. Fang, F. Wang, and J. Ge, "A task scheduling algorithm based on load balancing in cloud computing," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6318 LNCS, no. M4D, pp. 271–277.

[47] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim : A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms Outline • Introduction • Related Work • CloudSim Architecture • Design and Implementation," *Softw. - Pract. Exp.*, vol. 41, no. 1, pp. 23–50, 2011.

[48] L. A. Zadeh, "Fuzzy sets and systems," *Int. J. Gen. Syst.*, vol. 17, no. 2–3, pp. 129–138, 1990.

[49] "Fuzzy Sets and Systems | RG Journal Impact Rankings 2017 and 2018." [Online]. Available: https://www.researchgate.net/journal/0165-0114_Fuzzy_Sets_and_Systems. [Accessed: 29-Sep-2019].

[50] E. R. A. Safian, "• Introduction • Fuzzy Inference Systems • Examples," p. 131.

[51] W. Pedrycz and F. Gomide, "Instructor's Manual An Introduction to Fuzzy Sets: Analysis and Design," p. 465, 1998.

[52] G. Belalem, F. Z. Tayeb, and W. Zaoui, "Approaches to Improve the Resources

Management in the Simulator CloudSim," Springer, Berlin, Heidelberg, 2010, pp. 189–196.

[53] C. Chen, J. Liu, Y. Wen, and J. Chen, "Research on Workflow Scheduling Algorithms in the Cloud," 2015, pp. 35–48.

[54] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," 2010.

[55] Z. Goudarzi, "Effective Load Balancing in Cloud Computing," *Int. J. Intell. Inf. Syst.*, vol. 3, no. 6, p. 1, 2014.

[56] "Cloud Computing Model - an overview | ScienceDirect Topics." [Online]. Available: https://www.sciencedirect.com/topics/computer-science/cloud-computing-model. [Accessed: 09-Dec-2019].

[57] V. Vinothina, S. Lecturer, and R. Sridaran, "A Survey on Resource Allocation Strategies in Cloud Computing," vol. 3, no. 6, pp. 97–104, 2012.

[58] J. Li, M. Qiu, J. W. Niu, Y. Chen, and Z. Ming, "Adaptive resource allocation for preemptable jobs in cloud systems," *Proc. 2010 10th Int. Conf. Intell. Syst. Des. Appl. ISDA'10*, pp. 31–36, 2010.

[59] A. Gupta and R. Garg, "Load Balancing Based Task Scheduling with ACO in Cloud Computing," *2017 Int. Conf. Comput. Appl. ICCA 2017*, pp. 174–179, 2017.

[60] S. G. Eladl, N. I. Ziedan, and T. S. Gaafar, "Cloud Computing Load Balancing using Genetic and Throttled Hybrid Algorithm," *Int. J. Eng. Technol.*, vol. 11, no. 3, pp. 606–626, Jun. 2019.

[61] D. Chitra Devi and V. Rhymend Uthariaraj, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks," *Sci. World J.*, vol. 2016, 2016.

[62] M. Kumar and S. C. Sharma, "Load balancing algorithm to minimize the makespan time

in cloud environment," vol. 14, no. 4, pp. 276–288, 2018.