A

Dissertation on (Major Project-II)

"Application of Enhanced WRR as Load Balancing Algorithm in Cloud Computing"

Submitted in Partial Fulfillment of the Requirement
For the Award of Degree of

Master of Technology
In
Software Technology

By

Ravi Jaiswal
University Roll No. 2K16/SWT/514

Under the Esteemed Guidance of

Dr. Rajesh Kumar Yadav
Assistant Professor, Department of Computer Science & Engineering

2016-2020

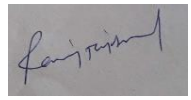
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
DELHI – 110042, INDIA

DECLARATION

Delhi Technological University
(Government of Delhi NCR)
Bawana Road, Delhi- 110042

I hereby declare that the thesis entitled **"Application of Enhanced WRR as Load Balancing Algorithm in Cloud Computing"** which is being submitted to the **Delhi Technological University**, in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Software Technology** is an authentic work carried out by me. The material contained in this thesis has not been submitted to any university or institution for the award of any degree.

Signature:



Student Name
Ravi Jaiswal
2K16/SWT/514
11/12/2020

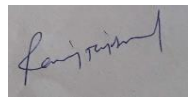
CERTIFICATE

Delhi Technological University

(Formerly Delhi College of Engineering)

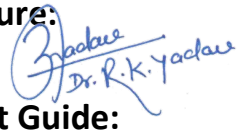
Bawana Road, New Delhi-42

This is to certify that thesis entitled "**Application of Enhanced WRR as Load Balancing Algorithm in Cloud Computing**" is a bona fide work done by Mr. Ravi Jaiswal (Roll No: 2K16/SWT/514) in partial fulfillment of the requirements for the award of **Master of Technology Degree in Software Technology** at Delhi Technological University, Delhi, is an authentic work carried out by me. The content embodied in this thesis has not been submitted earlier to any University or Institution for the award of any Degree or Diploma to the best of my knowledge and belief.

Signature: 

Student Name
Ravi Jaiswal
2K16/SWT/514

Above Statement given by Student is Correct.

Signature: 

12.12.2020

Project Guide:
Dr. Rajesh Kumar Yadav
Assistant Professor, Department of Computer Science & Engineering, DTU

ACKNOWLEDGEMENT

I would like to express sincere thanks and respect towards my guide **Dr. Rajesh Kumar Yadav, Assistant Professor, Department of Computer Science & Engineering, Delhi Technological University Delhi.**

I consider myself very fortunate to get the opportunity for work with him and for the guidance I have received from him, while working on this project. Without his support and timely guidance, the completion of the project would have seemed a far. Special thanks for not only providing me necessary project information but also teaching the proper style and techniques of documentation and presentation.

I am also thankful to **Dr. Rajni Jindal**, HOD of Computer Science & Engineering Department and **Dr. Ruchika Malhotra**, Coordinator, for the motivation and inspiration that triggered me for the project work. Besides this, I would like to thank entire teaching and non-teaching staff in the Department of Computer Engineering, DTU for all their help during my tenure at DTU.

Ravi Jaiswal
M. Tech (Software Technology)
2K16/SWT/514
11/12/2020

ABSTRACT

The recent emerging framework for cloud computing is set in the exchange of data resources in a decentralized dais, a technology based on clouds. Consequently, the requirements of clients are meeting with the widespread boom of technological developments. Furthermore it involves sellers that extend their approval as well as introduction of the new high performance computing model. The primary questions concerning the distributed computing framework, nevertheless, are the effective distribution of resources, task scheduling as well as load balancing. The paradigm of cloud / use computing includes a complex cloud-site function, in order to accomplish the productivity and control of demands as deliberately as it can be. Effective load management and appropriate resource distribution are critical mechanisms for optimizing the efficiency of various resources and for allowing reasonable use of current cloud storage properties. The cloud-based architecture therefore has many types of load issues, for example CPU loading, server loading, memory dump, network loading etc. This helps to achieve failure, reduce the backlog issues, ability to adapt, utilization of resources appropriately, increase reliability and customer performance and so on in the cloud environments by an effective load balancing model. This thesis studied different popular algorithms for load balancing in cloud computing. Revised round robin models are typically used for scheduling problems as well as load balancing by multiple big corporations to overcome the said issues. A EWRR algorithm has been implemented in this thesis concentrating on effective load balancing and efficient resource management as well as task scheduling problems.

Keywords: Resource allocation, load balancing algorithms, cloud computing, task scheduling.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

- **IAAS** Infrastructure as a service
- **SAAS** Software as a Service
- **PAAS** Platform as a Service
- **VM** virtual machines
- **ASIC** Application Specific Integrated Circuits
- **LBAAS** Load Balancer as a Service
- **QOS** Quality of Service
- **VLAN** Virtual Local Area Network
- **CBR** Case Based Reasoning
- **ACO** Accountable care organization
- **RS** Relative Superiority
- **VFS** Voltage Frequency Scaling
- **TRACON** Task and Resource Allocation Control
- **PSO** Particle Swarm Optimization
- **NDN** Named Data Networking
- **DRS** Distributed Resource Scheduler
- **ESX** Elastic Sky X
- **SRT** Shortest remaining time
- **SJF** Shortest Job First

- **FCFS** First come first serve
- **ESCE** Equally Spread Current Execution
- **FIFO** First in first out
- **WRR** weighted round robin
- **EWRR** Enhanced Weighted Round Robin

CHAPTER 1

1.1 INTRODUCTION

In the researches relating to computer field, cloud computing is becoming quite famous these days. Broadly if we try to categorize the services being rendered by cloud, one can find three services that also in the form of first is IAAS, second is SAAS and third to be PAAS. While the cloud is developing same time the chances are also increasing with regard to applications getting opened up and how various offices can be given to the end endorser by utilizing virtualization, on the web. The distributed computing climate requires the customary administration providers to have two different ways; foundation and specialist organizations. Framework provider's game plan of cloud stages and rent assets as indicated by use. Administration providers offer assets from framework suppliers to help end endorsers. Distributed computing has baited the mammoth organizations, for example Google, Microsoft and Amazon are considered as an extraordinary effect in the present Information Technology organizations in building up the up and coming age of innovation.

Distributed computing conveys a versatile, adaptable and simple approach to save and recall information and records as a piece of its administrations. Especially for making huge informational collections and documents available for expanding number of clients around the world. Managing such conspicuous informational indexes include a few methods to enhance as well as providing tasteful execution dimensions to the benefit of clients. Among the many significant problems related to distributed compute region, dynamic burden adjusting is a significant one. When it comes to Cloud Computing, fundamental reason requests about the task of errands to cloud hubs with specific goal that the exertion and solicitation handling is done as proficiently as could be expected under the circumstances [1], while having the option to persevere through the different including requirements, for example, heterogeneity and high correspondence delays.

Presently as the fundamental idea of distributed computing is to offer assets for example virtual machines (VMs) as administrations on interest. Relegating successful VM on interest is being lead with the help of the heap adjusting calculations in the distributed computing. As the heap adjusting calculation has a huge impact while picking,

Which VM is to be appointed on interest of the endorser while offering offices it is conceivable to have various demands at once and in light of that a few requesters require to stay in line however they have likelihood to advance solicitation to other administration provider Subsequently with the help of the heap adjusting calculation endorser will be skilled to choose whether they require to stay in the line or get office from the other Administration providers [2].

Various burden adjusting calculations in the distributed computing are existing for the task of the compelling assignment for virtual machines (VMs). Among such existed calculation which is to be used is the fundamental choice is to be considered. A portion of those calculations have been clarified in this thesis. Thus for having exact use of assets and being reliable with every one of the assets, thought of burden adjusting is being directed.

1.2 CLOUD COMPUTING OVERVIEW

A distributed computing can be reviewed as a web based structure in which the processing assets, for example, stockpiling, equipment, databases, programming applications, arranges and even basic working frameworks in origination of data innovation (IT) administrations are accessible to the end customers according to the-request of administrations. Also, there very little consideration of new advancements in distributed computing; yet, it enables to expand the determinative leads in overseeing IT administrations and free cost issues [3].

Distributed computing utilizes a procedure for the net where in the purpose is to oversee data along with the processing of data. Distributed compute grants organizations and purchasers to use processed data with no access as well as establishment of matters of themselves on their systems which has an internet connection.

This method grants for inadequate incorporating of memory. With virtualization arranged at the same time isolates into a few detached "virtual servers”, all working autonomously and appear to the customer to be a solitary physical gadget. These virtual servers don't physically comprise and would thus be able to be moved every which way and flaky up or down on the fly without affecting the end supporter [4].

1.3 CLOUD VIRTUALIZATION

In setting of distributed computing, virtualization is a method which permits sharing single physical occurrence of an asset among different associations or clients. Virtualization is synonymous to something that isn't genuine, however supplies all offices that are existing in reality. Virtualization gives every single distinctive administration of distributed computing to the end client by remote server farm with incomplete virtualization or full virtualization way [5]. It can help in burden adjusting by empowering amazingly responsive provisioning and maintaining a strategic distance from hotspots in server farm. Chiefly two sorts of virtualization basically exist:

a. Para Virtualization

In Para virtualization the equipment framework isn't copied. The customer programming runs their very own disengaged field. In this whole the administration isn't completely accessible, yet incomplete administrations are provided. The Para virtualization capacities with a working framework that has been changed to work in a virtual machine. Better efficiencies of this virtualization can likewise prompt better scaling.

b. Full Virtualization

In full virtualization the total establishment of one framework is done on another framework, with the goal that all the product which is accessible in real server will likewise be available in virtual framework. It additionally permits sharing of PC framework among numerous clients and recreated equipment arranged on various frameworks are accessible.

1.4 CLOUD COMPUTING AND LOAD BALANCING ALGORITHMS

Burden adjusting is the innovation of spreading the heap among a few assets in any framework. Henceforth burden should be dispersed over the assets in cloud-based design, so every asset does nearly a similar measure of work anytime of time. Fundamental need is to offer a few strategies to adjust solicitations to offer the arrangement of quick answer for solicitation. Cloud Load Balancers keep up online traffic by spreading outstanding burdens among various servers and assets consequently. They increment throughput, decline reaction time, and maintain a strategic distance from over-burden. In this thesis, a general study of the new burden adjusting strategies in the Cloud Computing environment is submitted [6].

The ideas of every calculation are discussed lastly summed up. Be that as it may, there are different issues while taking care of with burden adjusting in a distributed computing climate. Each heap adjusting calculation must be, for example, to instate the required target.

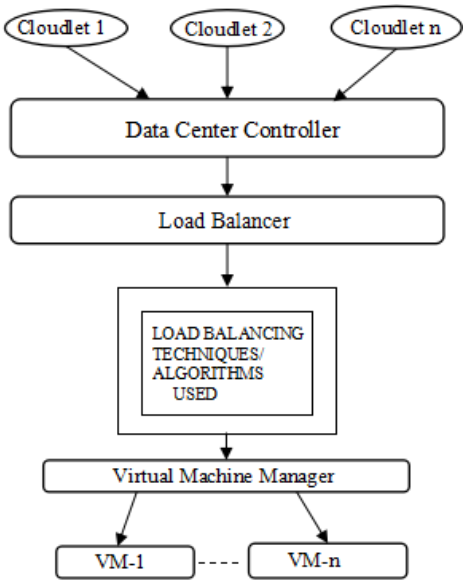


Fig. 1.1: Load balancing algorithm execution.

1.5 CATEGORIES OF LOAD BALANCING ALGORITHMS

Burden adjusting calculations are extensively partitioned into two noteworthy classes [7]:

- Based on how the heap is circulated and how procedures are apportioned to framework hubs.
- Based on the data status of the hubs.

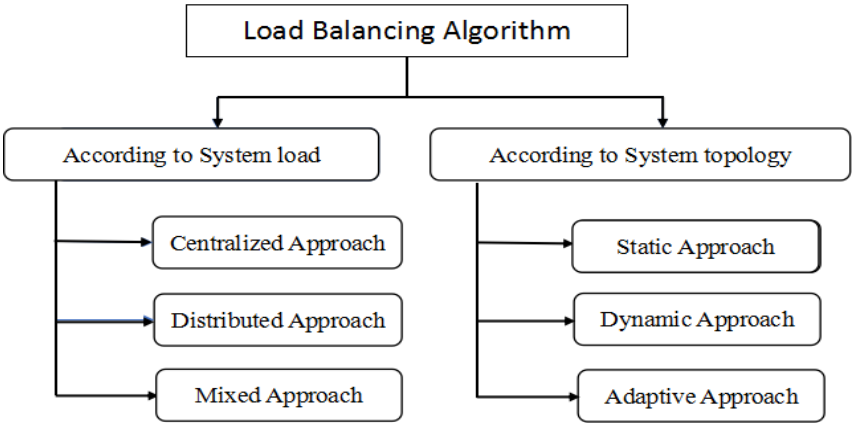


Fig. 1.2: LB algorithms categorization

A. Based on the framework topology -

1) Static Approach: Static calculations are commonly appropriate for homogeneous just as steady situations and can yield better outcomes in these conditions. By and by, they are commonly not adaptable and are unfit to coordinate the dynamic changes to the traits all through the execution time [8] (Hameed et al. 2016).

2) Dynamic Approach: Dynamic calculations are progressively adaptable and can consider various sorts of qualities in the framework, including both preceding and during run-time.

3) Adaptive Approach: These methodologies are fit to adjust the appropriation of burden to framework, by changing their parameters powerfully just as their calculations.

They can give better execution when the framework state changes much of the time and are increasingly appropriate for versatile circulated frameworks, for example, cloud systems.

B. Based on the framework load -

1) Mixed methodology: It takes the advantages of both incorporated and disseminated approach.

2) Distributed methodology: In this, every single hub freely frames its very own heap vector by get-together the heap data of different hubs. Choices are privately made utilizing neighborhood load vectors. This methodology is progressively alluring for broad circulated frameworks, for example, distributed computing.

3) Centralized methodology: Here, A solitary hub is in charge of dealing with the circulation of assets inside the entire system.

These calculations can comply with improves and permit results in heterogeneous and dynamic conditions. As the appropriation of qualities become progressively intricate and dynamic, a portion of these calculations could wind up wasteful and may cause more overhead than required, bringing about by and large corruption of the administrations execution.

1.6 EXISTING LOAD BALANCING ALGORITHMS

Burden adjusting arrangements can be partitioned into equipment based burden balancers and programming based burden balancers. Equipment based burden balancers can deal with the rapid system traffic. They are particular boxes that incorporate Application Specific Integrated Circuits (ASICs) tweaked for a particular use though Software-put together burden balancers keep running with respect to standard equipment segments and standard working frameworks. Disseminate outstanding burden of numerous system connects to accomplish least reaction time and most extreme throughput and to abstain from over-burdening. There are diverse burden adjusting calculations for the heap dispersion.

1.6.1 Round Robin Load Balancer:

This calculation works on irregular determination of the virtual machines. The server farm controller allots the solicitations to a rundown of Virtual machines on a turning premise [9]. The principal solicitation is doled out to a VM picked self-assertively from the gathering and after that the Data Center controller dispenses the solicitations in a roundabout manner. When the VM is dispensed the solicitation, the VM is moved to the rundown end [10].

1.6.2 Equally Spread Current Execution Algorithm:

Similarly equally spread current execution calculation procedure makes do with needs. It spreads the heap self-assertively by analyzing the size and move the heap to that VM which is gently stacked or deal with that assignment simple and devote less time, and give most noteworthy throughput.

1.6.3 Throttled Load Balancer:

In this calculation the customer first demands the heap balancer to decide a proper VM to play out the required task. In this customer initially mentioning the heap balancer to analyze the privilege VM.

1.6.4 Distributed Dynamic Load Balancing Algorithm:

In the appropriated technique, the dynamic burden adjusting calculation is executed by all hubs accessible in the framework and the booking undertaking is shared between them. The correspondence among the hubs to acquire burden adjusting can take two structures: helpful and non-agreeable.

1.6.5 Non-Distributed Load Balancing Algorithm:

In the undistributed or non-disseminated strategy, the hubs work individual for instating a typical reason. Non-conveyed dynamic burden adjusting calculations are ahead classified into two: brought together and semi-unified.

1.7 CHALLENGES

There are a few measurements that can be improved for the better burden adjusting in distributed computing [11].

a. Performance: It is the general productivity of the cloud framework. In the event that every one of the parameters of the framework are improved, at that point perhaps by and large execution can be improved.

b. Resource usage: It is the degree to which the assets of the framework are used. An effective burden adjusting calculation must make ideal use of the accessible assets.

c. Response time: It is characterized as the base measure of time that a particular burden adjusting calculation requires to react in a dispersed framework. This time should be decreased for better execution.

d. Migration time: The time taken by the procedure to move starting with one framework hub then onto the next for the execution is known as Migration time. This time ought to dependably be less for better the exhibition of the cloud framework.

e. Fault Tolerant: It is characterized as the capacity a calculation to perform precisely and consistently even in the conditions of disappointment at any self-assertive hub in the framework.

f. Scalability: It discovers the capacity of the framework to achieve burden offsetting calculation with a limited number of hubs or processors.

g. Throughput: It is characterized as the greatest number of employments that have finished their execution for a given timeframe. A high throughput is required as a parameter for better execution of the framework.

CHAPTER 2

LITERATURE REVIEW

A broad investigation on different written works has been performed to conceptualize the procedure of burden adjusting under the haze condition. The examination investigation is gathered under different classes dependent on the idea of the executed framework structure.

2.1 LOAD BALANCING

An adaptable and vitality mindful structure for appropriation of virtual machines in the server farms is proposed by Dam et al. [12]. Specialized imperatives of various classifications like equipment, QOS, accessibility and so forth are considered while playing out the distribution of VM‟s in server farms. A power mini-computer is utilized to anticipate the power utilization of each piece of the server farm. The proposed framework is guaranteed to spare the vitality 18% more when contrasted with the current approaches.

Certain methods to perform burden offsetting alongside unique asset the board and diminishing the power usage over the physical machines in a distributed computing condition is proposed Akiyama et al. [13]. VM movement is engaged so as to lessen the server disappointments. Movements can happen because of hotspots, intermittent activity or abundance save limit. The relocation prompts server union better asset usage and problem area lightening.

Two methodologies named Case Based Reasoning (CBR) and guideline based methodology for appropriate asset planning to meet the nature of administrations prerequisites of the customers are created Fei et al. [14]. The remaining burden taken for the experimentation is the engineered outstanding task at hand and this present reality logical work process from the field of bioinformatics.

The target of the calculations created is to beat the SLA infringement and to evoke an appropriate asset portion plan to meet the QoS necessities.

A successful burden adjusting where they have embroiled fluffy rationale in the process is proposed by Xu et al. [15]. The fundamental rationale of this calculation created is to keep up a steady state over the virtual machines.

The parameters considered are speed of the processor, quantities undertakings apportioned. The interior parameters of the virtual machine and the server farm are used for the test set up. The outcome demonstrates to demonstrate a considerable advancement in the server farm preparing time the general execution time.

A load balancing methodology in distributed computing condition which utilized apportioning system for its working is executed Ghomi et al. by [16]. The best segment is chosen to plan the occupations. The segment additionally contains a nearby balancer which deals with the segments" load status. The investigation is led to dissect the execution time of the assignments when contrasted with the current systems.

E-STAB calculation created by Duan et al. [17] considers the traffic needs while performing errands planning for the distributed computing condition. The specialists express that the throughput of the framework can be improved by controlling the traffic dispersion and channelizing the correspondence hindrance and the clog misfortunes.

A calculation is performed for a proficient asset the executives by contemplating the exhibition and accessibility as the QoS parameters [18]. An assorted whole number nonlinear improvement asset the board strategy is created to deal with the cloud condition. The framework execution and the power model are watched contrasting and the current methodologies. The framework has demonstrated to create a superior Quality of Service in multi value-based administration conspire.

A Stratus framework created by Mehta et al. [19] works on Voronoi allotments to discover where to course the server farm prerequisites dependent on the near priority of the cloud administrators. The framework is intended to limit the vitality utilization subsequently lessening the carbon outflow and the expense too by finding the closest cloud servers to post the solicitations. Voronoi allotments are the development of subsets and the chart model called stratus is intended to screen and move the outstanding tasks at hand. A couple savvy dividing principle is assigned to pick the fitting allotments. Analyses are researched to break down the cooling cost and the carbon outflow proportion.

Eco Cloud as mentioned in Ibrahim et al. [20], in which the CPU as well as different assets are engaged to perform VM union. State data of the servers is used to touch base at a choice in the movement and solidification procedures of virtual machines. The exhibition measurements are the quantity of servers, number of movements relative to the servers, control utilization and so on. Likelihood capacities are intended for the errand assignments and the VM relocations. Trials are led to watch the presentation of CPU and RAM usage under dynamic servers and to demonstrate a reliable improvement in the proposed framework.

A Honey Bee Behavior enlivened load balancing calculation was proposed by Hashem et al., [21] which plans to accomplish very much adjusted burden crosswise over VMs to amplify the throughput and to adjust the needs of undertakings on the VMs. Thus, the measure of holding up time of the undertakings in the line is negligible. Utilizing this calculation normal execution time and decrease in holding up time of undertakings on line were improved. This calculation works for heterogeneous kind of frameworks and for adjusting non-preemptive free assignments.

As indicated by Sivagami and Kumar [22] Load Balancer as a Service (LBaaS) is one of the more prominent techniques; as opposed to building a heap balancer for each cloud, organizations want to expend LbaaS. In this technique, the heap balancer will be used through the system, and it requires data about the whole system, including data about each hub inside it. Most of LbaaS suppliers use round-robin or weighted round-robin calculation as their static calculation.

A power improvement calculation to adjust the outstanding task at hand among the virtual machines adequately so as to demonstrate an exchange – off between the exhibition and the vitality utilization is created by Mishra et al. [23]. Various sorts of calculations like round robin, throttled burden adjusting, similarly spread current execution calculation are taken for investigation. A power utilization number cruncher is intended to screen the vitality usage.

Han et al. [24] proposed in an improved weighted round robin calculation considering the capacities of each virtual machine and the length of each undertaking, where each mentioned errand allots to the most appropriate virtual machines. The investigation results and execution examination of this calculation demonstrated that the

improved weighted round robin calculation is the most appropriate one where heterogeneous/homogeneous undertaking with heterogeneous assets (virtual machines) contrasted with the other round robin and weighted round robin calculations.

As per Megharaj and Mohan, [25] in circulated figuring burden changing is required to achieve consistently spread stack among the center points and to gainfully make usage of the benefits Load adjusting ensures that all the processor.

In the structure or every center in the framework does around the proportionate proportion of work at any snapshot of time. This technique can be sender begun, authority began or symmetric. This complete presents the parts and analyzed about the upsides and drawbacks of various burden changing computation in the creator's technique. Diverse idea of the count like throughput, execution, adjustment to non-basic disappointment, development time, response time, etc. has been discussed.

A calculation where the mixed media remaining task at hand is adjusted over the various servers of a cloud system is proposed by Moganarangan et al. [26]. In view of the clients‟ demands the administrations are diverted to the suitable servers to improve the framework execution by lessening the outstanding burden of the servers. Hereditary calculation is utilized for the procedure of undertakings task.

Dynamic situations are produced with the end goal of experimentation and the outcomes present a huge improvement in the presentation and throughput of the framework.

A vitality effective burden adjusting approach using round robin approach was proposed by Pasha et al. [27] which focused on limiting the asset use along these lines decreasing the vitality usage and carbon discharge. The paper concentrated on the throughput of the framework and the adaptation to internal failure level. Distinctive existing burden adjusting methodologies are investigated and potential benefits and bad marks are expressed.

A load adjusting calculation for proactive outstanding task at hand administration by recognizing the remaining tasks at hand as base and blaze group is created by Dhari, A. and Arif [28]. The calculation concentrated on information gauge in order to liven up the framework productivity with a commonsense burden adjusting. The examinations are attitude with spotlight on the enhancement of asset productivity in the based zone and buildup in the information replications in the glimmer mass area. The calculation performs

outstanding load figuring and quick continuous information discovery so as to make the framework skillful with the current procedures.

An asset arrangement calculation for Forwarding and Control Element Separation – For CES system is created by Liu et al. [29].

The calculation is planned so that the best possible effectiveness can be accomplished if the asset booking is executed dependent on the expense and the need of the client applications. The investigation is directed in a Clouds in domain and the errands finish time and expenses are broke down to display a superior outcome.

A traffic supervision procedure portrayed by Ejaz et al. [30] oversaw burden adjusting utilizing a valuable VLAN mapping plan. Segment age technique is utilized to decide the issue of mapping. The scientists have avowed to deliver a powerful throughput by finding the pursuit space in under 1% with an optimality hole of 4% when contrasted with the current procedures.

A load adjusting calculation utilizing game hypothesis is created by Ruitao et al. [31] to keep up a legitimate errand booking crosswise over various virtual machines. The fundamental rule of cloud segment under different classifications like ordinary, inactive and substantial burden conditions are done basically. Various calculations like irregular calculation, weighted round robin, dynamic round robin is executed for the inert mode segment. The idea of game hypothesis which includes strife and participation is used in the ordinary mode cloud parcel to perform burden adjusting.

TRACON – a Task and Resource Allocation Control Framework is proposed by Babu and Samuel [32] for improving the framework execution in a virtualized domain. The scientists have broadcasted that 25% improvement is demonstrated in the throughput of the proposed framework when contrasted with the current methodologies. Tests are researched to demonstrate a noteworthy improvement regarding the quantity of assignments, execution time and the standardized throughput.

I – Aware (Interference Aware) system is created by Singh et al. [33] to direct virtual relocation in a distributed computing condition. The primary thought is to limit the quantity of movements while performing burden adjusting. The I/O and system throughput, CPU use and adaptability are treated as parameters while directing the trial investigation.

The outcomes demonstrate a reliable improvement in the VM movement level and decreased VM impedance.

Volkova et al., [34] in their investigation tended to various elements of the Load Balancing Approach. They revealed that few sorts of burden issues are connected with distributed computing for instance CPU load, memory burden, and framework load, and so on. Burden adjusting is the route toward passing on the stack over the unmistakable center points which gives the incredible resource use when centers are over-trouble with business. Burden adjusting needs to deal with the heap when a hub is over-troubled.

The inspiration driving this execute genuine issues appropriated figuring Power Consumption and Lloyd Balance. Objective in burden adjusting is to manufacture as well as expanding asset use and upgrade the cloud execution.

Scattering the virtual machine assets capably regarding the applications prerequisites is examined by Ali et al., [35] which focused on cost cutting plan as far as vitality and advancing the quantity of servers being used. The scientists have demonstrated that the proposed technique can trim down the expense of servers all in all, at the same time promise the fullest asset use and QoS request in different extents of server assets.

An arrangement of dealing with the cloud virtual machines during the procedure of burden adjusting and asset the board is talked about. Lessening the asset wastage and power utilization is the principle thought process behind the framework. A similar investigation is directed against Distributed Resource Scheduler to give a superior arrangement as far as the presentation of the framework. The tests directed present an extensive distinction in the power utilization, arrange use and the exhibition assessment of the cool cloud when contrasted with DRS [36].

A framework is actualized to impact the NDN to help VM movement in the cloud server farms in a faultless way [37]. The heap adjusting calculation is proposed to decrease the quantity of administration interferences. The heap on various VM‟s is broke down relative to the bounce tallies of the solicitation and the reaction. The framework demonstrates to improve the presentation by decreasing the quantity of movements and furthermore the administration disturbances.

A broad research is led by Gangadhar et al. [38] to apply hereditary calculation to perform burden adjusting in a distributed computing condition.

Via cautious asset use the vitality utilization is decreased to a corresponding proportion. The objective is to decrease the VM movement cost. The examination is led to break down the exhibition of the virtual machines over the ESX servers regarding the memory and CPU use. The yield of vitality effective burden offsetting when contrasted and the typical burden adjusting strategy has demonstrated to demonstrate a significant improvement in the throughput of the framework.

An examination work called multi – objective settled Fuzzy Logic Technology for remaining task at hand adjusting is proposed by Velde et al. [39].

The focal thought is to play out a compelling activity planning in order to lessen the handling time in the cloud server farms in this way advancing the vitality.

Numerous targets like change administrators, particles and so on are utilized in the PSO enhancement by taking different parameters of server farms, errands landing, length of the assignments, task consummation time.

A tale calculation is proposed by Guo et al. [40] for memory reusing during VM movements. A VM relocation helps in adjusting the heap and augments the asset use. It will likewise prompt high memory utilization in putting away the foundation information before the exchange. This calculation as opposed to moving the information stores it in the first host and reuses the equivalent once when the VM is moved back to the first host. Results have demonstrated that memory utilization is diminished by 33% when the size of the memory picture is decreased and when reestablished in the first host.

In their examination Kaur et al. [41], delivered a broad diagram on cloud-based issues and give a point by point examination of some best arranging strategies from the memory portion area and their execution in disseminated distributed computing. Travel arranging and booking is one of the indispensable parts that impact the asset use and cloud execution. A broad measure of research work is accessible to capably design dispersions assets. Examination outlined improvement calculations area lovely distributed and memory and CPU use with their appropriateness and ideal conditions.

A heap adjusting calculation is executed to work in a heterogeneous cloud radio access arrange in order to reduce the pinnacle transmission load in the remote radio leaders of the versatile systems [42]. Adjusting the remaining burden in the backhauls is a significant undertaking in the portable system.

The heap is adjusted dependent on the closest neighbor area. The investigations are led to demonstrate that the heap adjusted in the backhauls is superior to anything the current pressure approaches winning in the portable system.

INS – the record name servers were proposed by Raj et al. [43] to diminish the heap in light of the event of copy documents.

The framework concentrated on directing record stockpiling, information de-duplication, determination of the best hub and assignment allotment among servers. Via cautiously checking the nearness of the records in the capacity frameworks, the replications are stayed away from and the capacity proficiency is improved in this way. The presentation of such framework is improved with productive capacity and get to and the remaining burden is adjusted over the server in a powerful way. Information duplication rate and the stacking rate over the customers are watched and the defer time in access is significantly diminished.

Stochastic Hill Climbing is talked about by Kumar, M. and Sharma [44], wherein a delicate registering based burden adjusting method has been presented. A neighborhood enhancement procedure has been applied task approaching occupations virtual machines (VMs) servers. There are two huge groups of procedures for comprehending an improvement issue. Complete strategies which guarantee either to decide a legitimate distribution of qualities to factors or demonstrate that no such portion accessible. These methods habitually have great execution, and guarantee an ideal and right response for all data sources.

Authors Ragmani et al., [45] proposed another errand planning calculation based on ant colony optimization to limit the make span and augment the assets usage considering free tasks. The proposed calculation computes the complete preparing intensity of the accessible assets and the all-out mentioned handling intensity of the clients' undertakings. At that point dispense a gathering of clients' assignments to each virtual machine dependent on the proportion of its required power comparing to the all-out preparing intensity of every single virtual machine. Assess the presentation results demonstrated that the proposed calculation outflanks of different calculations by limiting make range and augmenting the assets usage.

Chen et al., [46], in the investigation on “CLB: A novel load balancing architecture and algorithm for cloud services.”, has given a nitty gritty examination in the different calculations that were proposed before.

An investigation into the different calculations has been made to locate a legitimate answer for load scheduling and adjusting in distributed computing condition and analyze those calculations dependent on different parameters. The benefits and faults of the calculations are talked about in the study paper.

In another examination by Kang and Choo [47], an SDN-based calculation method was proposed to accomplish least finishing time by thinking about need as a factor. In this calculation, rather than requesting the approaching assignments dependent on need, the need of each errand is utilized to distinguish the VM to which the undertaking ought to be moved.

In this proposed bumble bee calculation, when the undertaking scheduler discovers unevenness in the heap among the hubs in the system, it relocates the high need task from the over-burden VM to any VMs with a lower number of comparative need errands.

This calculation additionally utilizes the preemptive strategy for moving ward undertakings. This outcomes in improving the base fruition time and boosting use.

A half breed calculation is executed by Krishnadoss and Jacob [48] coordinating the rationale of ACO and cuckoo search calculations. This exploration work center around diminishing the vitality utilization in occupation planning task by including a make-range limitation. A heuristic methodology of intensity mindful errand portion is performed by utilizing the rationale of relative predominance (RS) of the processors. This heuristic methodology refines a neighborhood ideal outcome. A Voltage Frequency Scaling (VFS) is utilized to evaluate the processor vitality in this manner settling on a choice on the CPU usage. Make-length model thinks about on compelling occupation planning to the virtual machines so as to diminish the fulfillment time of the errands. The outcomes demonstrate a huge improvement in the vitality when contrasted with the ACO calculation.

Different existing load balancing techniques in cloud computing that have been referenced in different written works generally focus on lessening related overhead, to help response time and upgrading execution, and so on. Notwithstanding, none of the strategies has thought about the vitality usage or the components of carbon emanation. Distributed

computing offices and venture server farms devour a tremendous measure of intensity. A distributed computing server farm devours fundamentally more for it runs the figuring activities for different clients.

CHAPTER 3

METHODOLOGY

3.1 BACKGROUND

The load balancing alludes to a procedure wherein the measure of work done by a solitary PC is partitioned among at least two PCs which advantages the client to get their undertaking served speedier and each PC that are included completions their assignment in same measure of time. Burden adjusting can be executed as far as equipment, programming, or a mix of both. Different servers are included during the heap adjusting process. A similar procedure can be clarified in straightforward terms with the end goal that, when two servers adjusts the remaining burden, at that point likewise record figure out must appointed.

It can be commonly finished with the help of programming different planning calculations proficient remaining task at hand task. The circulation of outstanding burden and adjusting it between at least two servers. Hence, foundation has been molded for matching movement, accordingly streamlining portion assets guarantee an insignificant reaction. Load balancers have been very prescribed ensured progression of administrations, taking care of high traffic and confronting abrupt solicitation spikes. Over-burdens can be averted by expanding the quantity of adjusted Cloud servers. The process was implemented in CloudSim software bundles under Eclipse framework.

3.2 CLOUD SCHEDULER AND LOAD BALANCER

The scheduler and load balancer are the two most significant parts of any cloud-based framework. Figure 3.1 gives a schematic plan of a scheduler and burden balancer in cloud organize. The assignment administrator gathers data about each errand landing through the undertaking line. The scheduler and load balancer are the segments that utilization a heap adjusting calculation to disseminate the heaps uniformly over the assets. The scheduler is a part that aides in choosing the suitable VM to which an arrived undertaking ought to be assigned. It guarantees that the errand is allocated to a VM that sets

aside less effort to finish the assignment by considering the measure of burden in that particular.

VM and the absolute time expected to finish the whole process [47]. The scheduler utilizes the data about the assets from the asset supervisor to locate the proper,

VM to deal with the approaching assignment. Each VM has an in-coming errand line, and the assignments distributed to each VM is accessible in the approaching undertaking line.

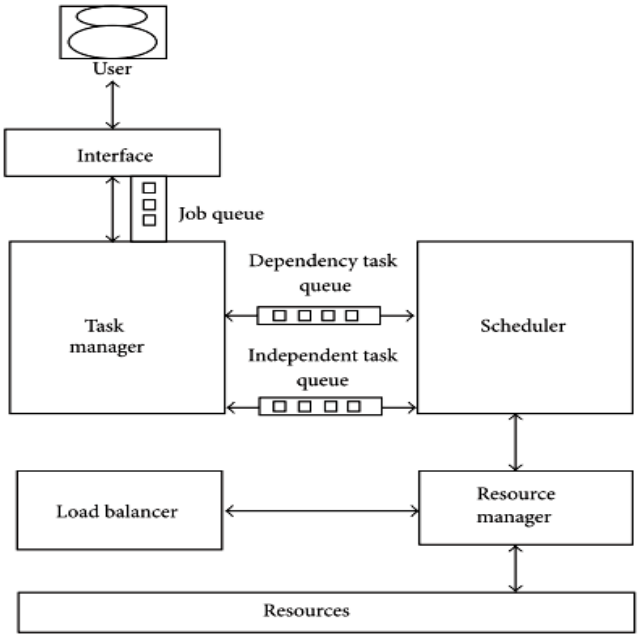


Fig. 3.1: Design of scheduler and load balancer in cloud

The load balancer is another significant segment in this framework; it guarantees that the heap in the cloud system is equitably appropriated. In the event that a heap balancer recognizes any of the VMs as over-burden, it promptly relocates the assignment from an over-burden VM to an under-stacked or preferably stacked VM. An approaching undertaking first achieves the assignment supervisor, which distinguishes the length and need of the errand. The asset chief computes the weight to each VM by utilizing different data, for example, the handling limit of VMs, the quantity of errands in each VM approaching assignment line, and all out time required to finish the undertakings of every need.

The load of each VM is determined each time the scheduler gets an approaching solicitation. With the learning of this data, the scheduler finds a suitable VM for every

approaching undertaking. The load balancer helps in discovering load unevenness in the framework by ascertaining the proportion between the complete number of occupations in the line and the quantity of accessible VMs.

It chooses which VM is less used and speak with the scheduler to designate the approaching undertaking to that VM. The heap on each VM is likewise estimated to guarantee it isn't over-burden.

The scheduler puts the run time landing occupations in the most appropriate VMs dependent upon minimum used VM specific employment entry.

The Load Balancer chooses that relocation in errand an intensely stacked VM to an inert VM less stacked at run time, at whatever point it finds an inactive VM or less stacked using the assets current status data.

Asset screen speaks asset prober and gathers abilities, burden employments holding up line. Undertaking prerequisite is given client which incorporates assignments moves necessities to employable choices.

3.2.1 Computation of Load Imbalance Factor:

The total of heaps of every single virtual machine can be figure by:

$$L = \sum_{i=1}^{k} l_i$$

Here *i* demonstrates the quantity of VMs in the datacenter. In this manner, the load imbalance factor of a specific virtual machine can be distinguished by:

$$LPC = \frac{L}{\sum_{i=1}^{m} c_i}$$

$$Threshold\ T_i = LPC \times c_i$$

Where edge here, LPC is the load per unit of capacity and c_i is the limit of the hub.

$$If\ VM \begin{cases} < \left| Ti - \sum_{v=1}^{k} L_v \right| & Underloaded \\ > \left| Ti - \sum_{v=1}^{k} L_v \right| & Overloaded \\ = \left| Ti - \sum_{i=1}^{k} li \right| & Balanced \end{cases}$$

The relocation of undertaking from the over-load stacked permitted load over-load dips under the edge and the thing that matters the difference stored in μi (scheduling constant).

I : number of VMs

LPC : load per unit capacity

μi : scheduling constant for loads

3.3 SHORTEST REMAINING TIME ALGORITHM

The shortest time (SRT) algorithm as the name implies chooses the development cycle, which requires the least amount of time to perform. Moslty these shortest remaining time scheduler calculation may experience starvation. On the off chance that the short procedures are consistently added to the CPU scheduler, at that point the existing running procedure will always be unable to execute, consequently SRT might lead to starvation in such conditions.

The booking can be ordered into two sections:

i) *Preemptive:* The procedure which is right now in execution, keeps running until it complete or another procedure is included the CPU scheduler that requires littler measure of time for execution.

ii) *Non-preemptive:* Once chose for execution, a procedure keeps on running until the finish of its CPU burst. It is otherwise called Shortest Job First (SJF). The two most as often as possible utilized booking calculations in a non-preemptive technique are rand rub-in along with weighed rand rub-in calculations. Improvised weighed rand rub-in has been accepted calculation. Prevailing calculations assumed to be executed for similar investigation.

3.4 Round Robin Algorithm

Round-robin (RR) is one of the calculations utilized by procedure and system schedulers in figuring. As the term is commonly utilized, time cuts (time quanta) are doled out to each procedure in equivalent parts and in round request, taking care of all procedures without need (otherwise called cyclic official). Round-robin booking is straightforward, simple to execute, and without starvation. Round-robin booking can likewise be connected to other planning issues, for example, information bundle planning for PC systems. It is a working framework idea. The name of the calculation originates from the round-robin standard known from different fields, where every individual takes an equivalent offer of something thusly.

Procedure scheduling

To schedule multiple transactions reliably, a round-robin scheduler by its capacitive computations utilizes time-sharing, giving each incoming task a vacancy or quantum (its payload of CPU time), and checking on the activity until it isn't finished by, at that point. The activity is continued next time a schedule opening is shared out to that procedure. In the event that the procedure ends or changes its state to busy during its assigned time measures.

The scheduler chooses the primary procedure in the prepared line to execute. Without time-sharing, or if the quanta were huge in respect to the sizes of the employments, a procedure that delivered huge occupations would be supported over different procedures. Round-robin calculation is a pre-emptive calculation as the scheduler powers the procedure out of the CPU once the time portion terminates.

Round-robin planning results in max-min decency if the information bundles are similarly estimated, since the information stream that has held up the longest time is given booking need. It may not be attractive if the size of the information bundles shifts generally starting with one employment then onto the next. A client that produces enormous bundles would be supported over different clients. All things considered reasonable lining would be best. Whenever ensured or separated nature of administration is offered, and not just best-exertion correspondence, shortfall round-robin (DRR) booking, weighted round-robin (WRR) planning, or weighted reasonable lining (WFQ) might be considered.

3.5 WEIGHTED ROUND ROBIN ALGORITHM

The weighed rounded robin considers the benefit capacities of the VMs and consigns higher number of endeavors to as far as possible VMs subject to the weightage given to all of the VMs. In this computation, at whatever point another requesting is gotten, it finds a fitting VM by figuring the holding up time. Be that as it may, it as such fails to consider the length of the errands for the decision of the fitting VM.

3.6 PROPOSED ALGORITHM

There are diverse weight changing computations available, for instance, round robin, weighted round robin, dynamic weight altering, Equally Spread Current Execution (ESCE) Algorithm, First Come First Serve, Ant Colony estimation, and Throttled count. The frequently used arranging techniques for a non-preemptive system are first in first out (FIFO) and weighted round robin (WRR). The proposed figuring is the Enhanced Weighted Round Robin (EWRR) estimation.

The computation takes its start from the Weighted Round Robin figuring. The Weighted Round Robin computation works by consigning the most extraordinary burden to the server with the best judgments. For instance if there are two servers one with the Intel i7 processors and extended getting ready speed and the other server with Intel i3 processors and moderately least taking care of rate then the server with the best specific is designated the best weight. Essentially, the Weighted Round Robin count is a phenomenal case of the Round Robin estimation that capacities splendidly with servers of different subtleties.

The EWRR attempts to such a degree, that despite considering the specific of the servers it in like manner considers the execution time of the endeavors to such a degree, that it doles out the task with the most outrageous execution time to the server with the best weight. This ensures the load is scattered similarly among the servers along these lines constraining the response time.

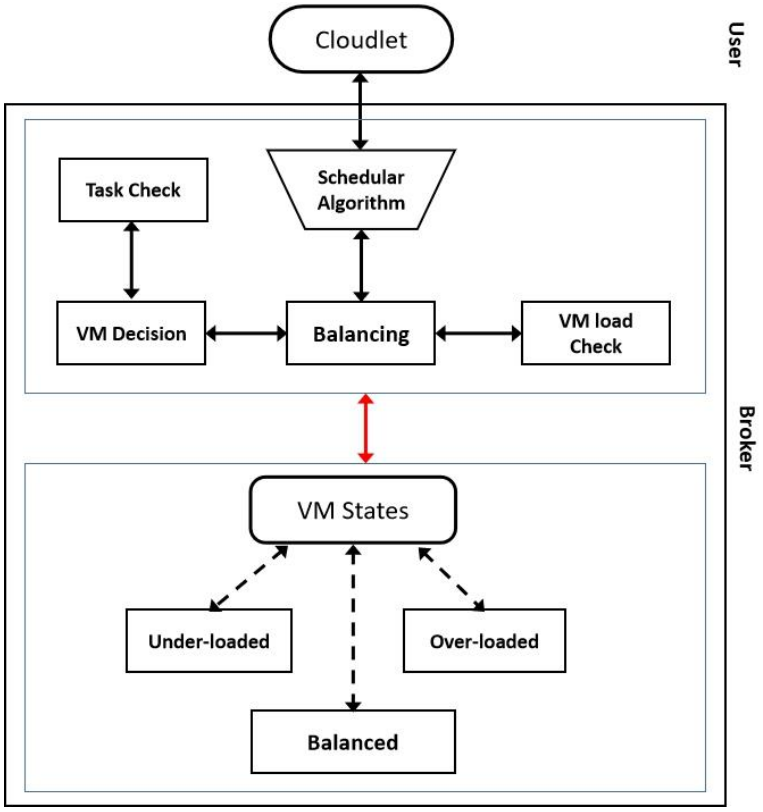


Fig. 3.2: Block diagram of architecture of the proposed EWRR.

"The proposed EWRR suits calculation the most and it assigns the employments to the most appropriate VMs dependent on the VM's data like its handling limit, load on the VMs, and length of the arrived undertakings with its need (figure 3.2).

The static planning of this calculation utilizes the preparing limit of the VMs, the quantity of approaching undertakings, and the length of each errand to choose the designation on the suitable VM."

"The proposed EWRR calculation utilizes crafted by the WRR calculation as a base, yet it moves the calculation from static to dynamic. This calculation finds a proper VM with the assistance of data that incorporates the preparing intensity of VM, length of the assignments, need of approaching errands, and handling time required for VMs to finish the undertakings of equivalent or higher needs. When an errand touches base in the framework, data about the undertaking, for example, the length and the need of the assignment is noted. At that point the heaviness of each VM is determined, as referenced prior. Next, the approaching assignment is sent to the VM with more weight. Fundamentally,

the heaviness of a VM alludes to the measure of time that particular VM requires to finish the errands with equivalent or higher needs in its line. The heaviness of VM increments if the fruition time is shorter, and the other way around. All in all, the time taken to process an errand can differ in runtime; in that circumstance, task relocation is performed.

The high need task from the line of the over-burden VM is relocated to the under-stacked or in a perfect world stacked VM, by discovering which VM has less of equivalent or higher need assignments.

This helps the high-need assignments to be finished quicker, and the relocation helps in expanding the best possible use of assets.

The holding up time of each VM, WTvm is determined as pursues:

Where p is the need of errands and T is an execution time of an undertaking. The above condition is utilized to compute the holding up time of any VM when it gets an approaching undertaking of need I. In this investigation, the cloudlet with a modest number as need is considered as a high-need task, and the cloudlet with a huge number is considered as low-need assignments.

3.6.1 Dynamic Scheduler in EWRR Load Balancer

Dynamic scheduling in EWRR load balancer is processed by Initialization, mapping (planning), load equalization, and finally process execution. The execution is finalized by gathering the pending execution time of cloudlets sizes (in million instruction (MI)) from each of the interconnected VMs and organizing it in upward request of pending time pursued by organizing the run time of the arrived assignments in line, in light of the need. Mapping (process schedule) includes choice of selecting job which is in top of the line and computation of its finish time in each VM. At that point errand is relegated to the most fitting VM dependent on finish and pending execution time. Load adjusting is finished by including the comparing assignments and execution time to the VMs pending time.

In the event that any VM comes to over the edge level, at that point the VM is considered an over-load VM. The undertaking is moved from the over-load VM to an under-stacked

VM whose heap is not exactly the limit level. The under-stacked VM is prepared to acknowledge the undertaking until it achieves the edge level.

On the off chance that there are no under-stacked VMs, at that point no relocation is performed. According to the proposed calculation, while picking the under-stacked VM, it chooses the VM that has the most minimal number of assignments with equivalent or higher needs.

3.7 PRESCRIBED METHODOLOGY

In view of the writing studies of different existing burden adjusting calculations a similar report will be imagined. After the assessment of different burden adjusting calculations and finding the best out of every one of these calculation.

The one with higher likelihood and effectiveness in undertaking execution is assigned for the trial investigation. Shroud system is used for the programming and trial investigation of the proposed ideas.

i) Eclipse structure: This use to be original instrument in case of JIDF (Java Integrated Development Framework). A stage structured to build an incorporated advancement instruments. Overshadowing underpins quicker advancement of incorporated highlights based on module model. It underpins a wide scope of stages and can be acquainted with keep running on any stage. The engineering exists at the center of the Eclipse programming for dynamic disclosure, stacking, and running of modules. The stage is likewise in charge of the coordination of running the code. A wide scope of usefulness is upheld by capacity of augmentation for engineers. Components like records and information are constrained by regular stage asset model. Working with apparatuses and giving administration of incorporated assets in which the obscuration gives client an apparent plan to make with modules. This design benefits the module engineers. Intricacy of different runtime situations like distinctive working frameworks is dealt with by this stage.

ii) Eclipse Requirements: The Eclipse requires the Java Runtime Environment and the Java Development unit. The Java Runtime Environment is expected to run the Java programs and the Java Development Kit is a product pack that can be utilized for the advancement of Java applications.

3.8 IMPLEMENTATION SETUP

The accompanying equipment and programming parts were utilized for the execution:

- Processor: Intel Core i7 @ 2.9 GHz
- RAM: 8 GB
- Operating Systems: Windows 10
- Programming Language: Java
- IDE: Eclipse
- Framework: Eclipse

3.8.1 Processing of Algorithms

Processing of Algorithms records the well-ordered methodology for the round-robin calculation. This is a standout amongst the most generally utilized static calculations. In this calculation, the approaching solicitations is doled out to VMs in cyclic request. The primary solicitation is relegated to any arbitrary VM, and the accompanying solicitations are handled in cyclic request. This calculation is generally utilized among cloud suppliers for perfect static conditions.

Pseudo-code for RR algorithm:-

1) Scheduler moves the first incoming request or cloudlet to the ready queue.
2) The data center controller select which VM should get the 1st cloudlets.
3) The 1st cloudlet is assigned to any random VM.
4) Once the first cloudlet is assigned, VMs are ordered in a cycle manner
5) The VM which received the 1st cloudlet ids moved back to all VMs.
6) Next cloudlet is assigned to next VM in cyclic order
7) Go to step 3 for each cloudlet request until the scheduler processes all cloudlets

Lists the step-by-step approach for the weighted round-robin algorithm [48-50]. This algorithm is an improvement of the RR algorithm. In RR, the processing capacity of the processor or VM is not considered while scheduling the task. But in the WRR algorithm, the weight for each VM is calculated based on its processing power. VMs with more processing power have more weight.

VMs with less processing power have less weight. Once the weight is calculated, VMs are ordered based on weight, and VMs with more weight are assigned more tasks than the other VMs.

Pseudo –code for WRR algorithm:-

1) Scheduler moves the first incoming request or cloudlet to the ready queue.
2) A weight is given to each VM based on the processing power, a VM with more processing power has more weight.
3) VMs are arranged in the order of decreasing weight from high to low in a cycle fashion.
4) Assign an incoming task or cloudlet to VM which has more weight, more task are assigned to VM with more Weight than the others.
5) Once the desirable cloudlets are assigned to VM, it move back to other VMs.
6) The next cloudlets is assigned to the next VM in cycle order.
7) Go to step 4 for each cloudlets request until the scheduler processes all cloudlets.

Pseudo-code for dynamic Weighted Round Robin algorithm:-

1) Identity the length of an incoming task or cloudlet.
2) Calculate waiting time for each VM based on the available loads on each VM.
3) A weight is given to each VM based on the waiting time, a VM with less waiting time will have more weight.
4) Assign an incoming task to VM which has less waiting time.
5) Once the incoming task is assigned, calculate threshold and load in each VM.
6) If any VM is found to be overloaded
    a) Pick any task from the task waiting queue of the VM.
    b) Identity the under loaded VMs.
    c) Select the VM which has less waiting time
    d) Continue the process till it has overloaded VMs.
7) Go to step 2 for each cloudlet request until the scheduler processes all cloudlets.

In this lists the step-by-step approach for the priority weighted round-robin algorithm. This algorithm is an improvement of the dynamic weighted round-robin algorithm. In the dynamic weighted round-robin algorithm, the waiting time of each VM is calculated for each request, and an incoming cloudlet is assigned to VM with less waiting time. But in this algorithm, the priority of each task is considered while calculating the waiting time of VMs.

Pseudo-code for dynamic Enhanced Weighted Round Robin algorithm:-

1) Identify the priority of an incoming task.
2) Calculate waiting for each VM based on the priority (weight) of an incoming task.
3) A weight is given to each VM based on the weighting time VM with less waiting time should have more weight
4) Assign incoming task to VM which has less waiting time based on priority
5) Once the incoming task is assigned calculate the threshold and load in each VM
6) If any VM is found to be overloaded
    a) pick high priority task from to be overloaded
    b) identity the under loaded VMs
    c) select the VM which has les the number of tasks of same or high priority
    d) continue the process till it has overloaded VMs
7) Process the high priority task from the waiting queue
8) If the low priority task is waiting for more than fixed number of iteration
    a) increase the priority of the waiting tasks by one level
9) Allow the VM to pick the next task from the waiting queue, it should select the high priority tasks from the queue and scheduler will continue from step 1 simultaneously.

3.8.2 Execution Cost

The cost of execution applies to the overall expense of the cloudlet response activity. Total running costs for the suggested algorithm are measured in contrast to current algorithms in this experiment. Total performance costs are proportional to the overall execution costs needed to complete the execution of all cloud networked cloudlets. Such expenses cover processing, transmission, networking and network charges. With each program, the execution cost varies because each regulation works differently. Cost of execution results are computed from below given:

$$C_e = \sum_{i=1}^{k} \frac{S_i}{V_m} \times C_s$$

Where S_i is the cloudlet size in MI, k is the number of cloudlets within a cloud system, V_m indicates the process capacity of a VM (in million instructions per second (MIPS)), and C_s is the cost/second needed for processing a cloudlet.

CHAPTER 4

RESULTS AND DISCUSSION

The migrations of active tasks in the EWRR framework are very limited because of the comprehensive dynamic and static task scheduling algorithm whereby each activity is defined by VM, which is shown in (Figure 4.1). The load balancer did not consider further design and implementation, such that the task could be done in the shortest period practicable. Both the dynamic and static scheduling algorithm did not take into consideration task lengths throughout the cases of WRR or RR implementations. Rather, it takes into account only the capacity of resources and the list of jobs arrived.

In addition to the server specifications, the EWRR algorithm also acknowledges the run-time to perform activities so that it allocates the jobs with the highest expected run-time of maximum weighted servers. This guarantees that the load is equally dispersed between the VMs, thereby reducing the overall processing time (or execution cost). This ensures that the load balancers will consider the more priority optimizations in time, moving the job between higher loaded VM to under-used VMs.

Only at end of every assignment, the load balancer in the EWRR with task length is going to run. Unless the load balancer considers one of the VMs completing their whole assignment, only then the highest loaded VM of the category will be listed and the work-flow of the highest-loaded VM as well as the least-charged VM will be determined. If any task existing in the high load VM is completed by the least loaded VM as quickly as practical, otherwise the job would be transferred to the less loaded VM. (Figure 4.2). The VM efficiency is taken into consideration by the EWRR and the percentage of the tasks arriving in the Virtual Machine is allocated to the overall VM efficiency. It then takes place on the next point. However, if long jobs are provided to the reduced-capacity VMs, otherwise the successful completion time would be delayed (Figure 4.3). The simplistic RR has no parameters on these setting, VM functions and work intervals are perceived and worked upon. It simply allocates the task activities to the VM entries in an orderly way. Its working time is therefore greater than the other 2 corresponding algorithms. In space

sharing and time sharing mode owing to the aggregate completion rate of EWRR is higher than either of RR as well as WRR algorithms (Figure 4.4).

The load balancing mechanism divides the loads within all the resources which are available. The dynamic cloud computing used many of load balancing strategies. In the EWRR algorithm for each task, the minimum completion time (or task execution time) is calculated, then the task with the maximum of minimum completion time is mapped to the corresponding VM.

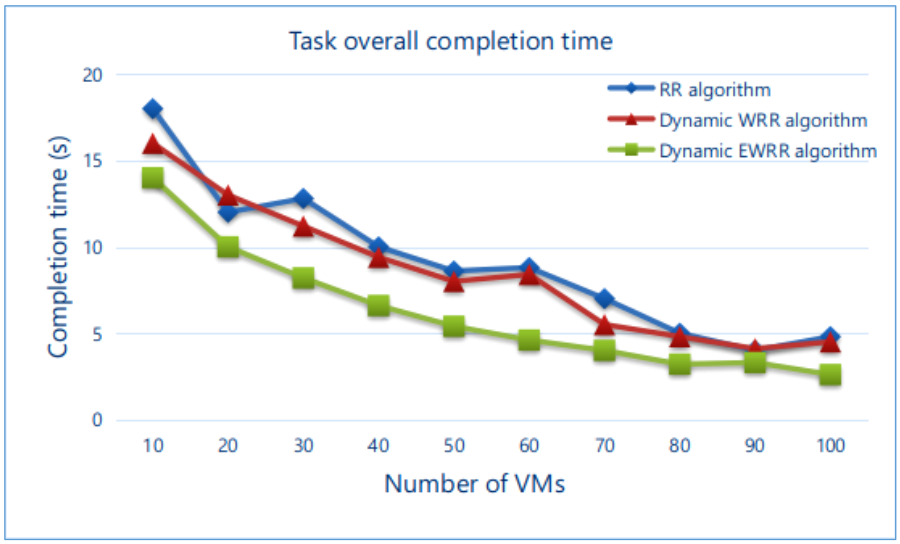


Figure 4.1: VM space shared time of execution completion

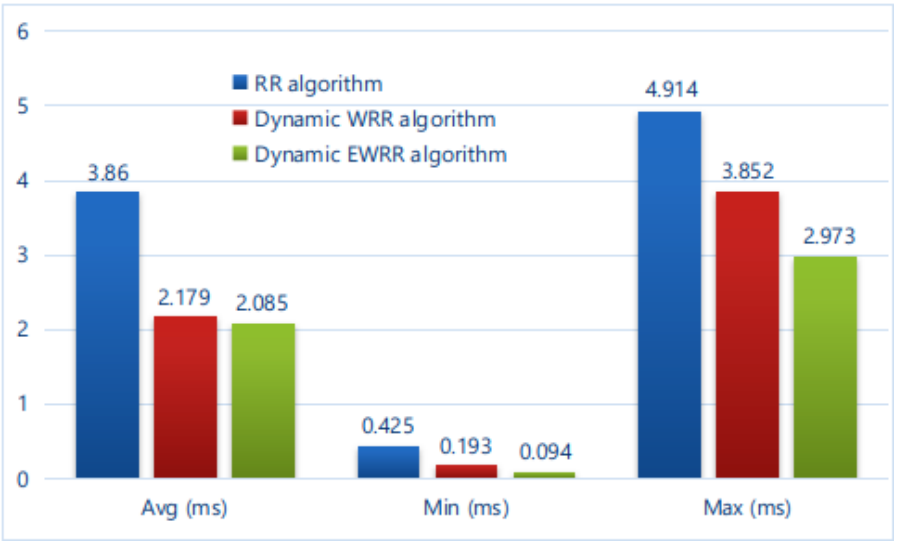


Figure 4.2: Task execution time of dynamic EWRR

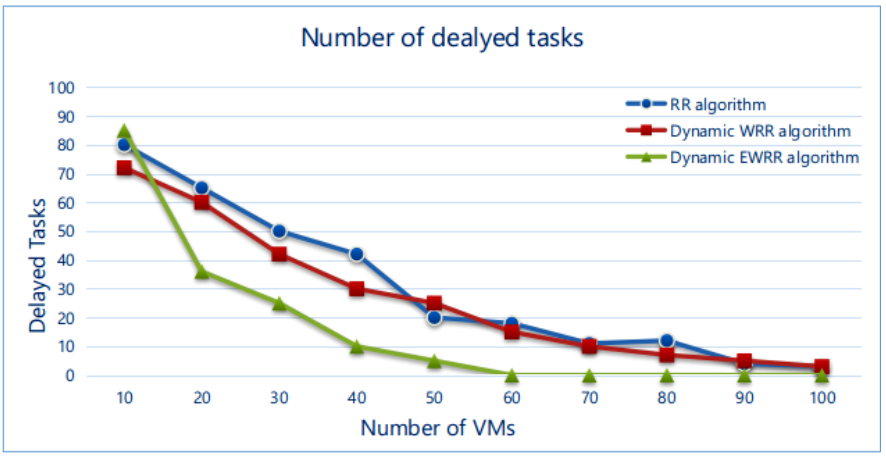


Figure 4.3: VM space shared number of tasks delayed

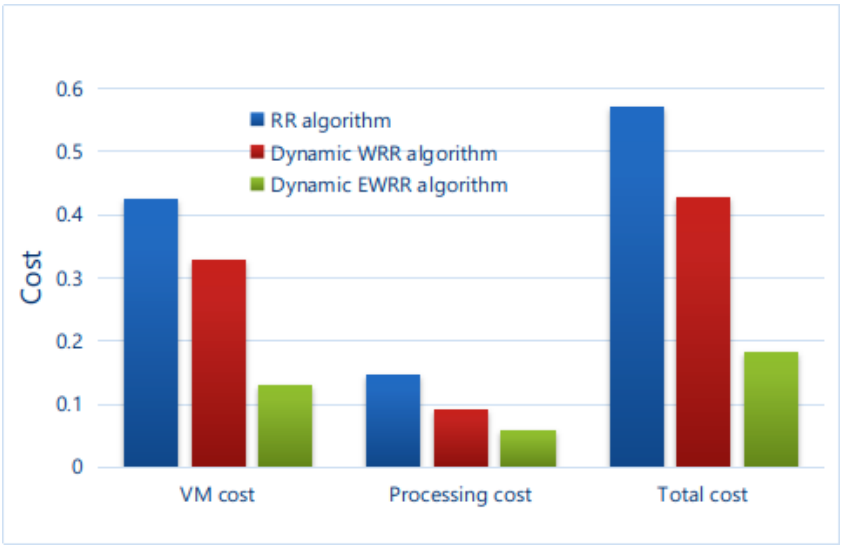


Figure 4.4: Overall execution cost under time share mode

Moreover, it was observed that dynamic EWRR load balancing algorithm gives better results in terms of avg., min-max., as well as active response time, number of delayed tasks, time of data transfer and resource costs in comparison to other load scheduling algorithms.

CHAPTER 5

DISCUSSIONS AND CONCLUSION

The cloud computing is an emerging and rapidly growing concept with an expanding technological development which transforms the IT environment quickly. Despite the important advantages of cloud computing, even so, load balancing in the existing framework is a major problem. This thesis covered a variety of recent findings of load balancing methods, primarily based on reducing workload, reaction time, capacity enhancement, maximization of efficiency as well as improved utilization of resources, etc. Within cloud, a huge set of variables as well as software methodologies for the efficient utilization of resources or even design constraints could be implemented in the coming years.

In this thesis we have analyzed the reliability of cloud-related networks of specific cloud computing modules dependent on load balancing parameters. For cloud-based applications, the load balancer performs a significant function. The applications in the cloud communicate with the customer and the associated computing resources. User-cloud interaction requires the load to be transferred to VMs as well as other facilities. Different academics and system programmers have used like swarm optimization-based jobs and activities forecasting methods to enhance load performance in cloud technology.

The non-preemptive task scheduling techniques such as round-robin algorithm are remarkably effective in comparison to older as well as traditional methodologies like FCFS and round SRT. Furthermore, in this thesis we introduced an enhanced-WRR (EWRR) algorithm as well as analyzed its effectiveness. In contrast to WRR and traditional RR independently, it was discovered that the new EWRR operated considerably better in experimental studies with migrations as well as utilization of the algorithm.

The EWRR algorithm has worked relatively well to balance load as well as to prevent overloading or under-loading of servers and thus it reduces response time to a minimum. The algorithm operates by determining the best weights to the list and then measuring how long the activities are to be carried out therefore the job with a longer duration passes to the highest weight of the list. The EWRR algorithm also prevents VM

starvations more extensively. With all the required specifications listed, the EWRR algorithm operates well with variable loads. The proposed EWRR showed improved performances in overall task completion time, task migrations over RR and WRR algorithms.

REFERENCES

1. Abdulhussein Abdulmohson, Sudha Pelluri & Ramachandram Sirandas, 2015, "Energy Efficient Load Balancing of Virtual Machines in Cloud Environments", *International Journal of Cloud-Computing and Super- Computing,* Vol.2, No.1, pp 21-34
2. Singh, A., Juneja, D. and Malhotra, M., 2015. "Autonomous agent based load balancing algorithm in cloud computing." ***Procedia Computer Science***, 45, pp.832-841.
3. Thakur, A. and Goraya, M.S., 2017. "A taxonomic survey on load balancing in cloud." ***Journal of Network and Computer Applications***, 98, pp.43-57.
4. Bernardetta Addis, Danilo Ardagna, Barbara Panicucci, Mark S, Squillante & Li Zhang, 2013, "A Hierarchical Approach for the Resource Management of Very Large Cloud Platforms", *Transactions on Dependable and Secure Computing*, Vol 10, (5), pp. 253 – 272.
5. Babu, K.R. and Samuel, P., 2016. "Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud." ***Innovations in bio-inspired computing and applications***, Springer, Cham. pp. 67-78.
6. Gajjar, D., Kotak, H., & Joshi, H. 2016. "Round Robin Load Balancer using Software Defined Networking (SDN)." *Capstone Team Research Project, University of Colorado Boulder.*
7. Carlo Mastroianni, Michela Meo & Giuseppe Papuzzo, 2013, "Probabilistic consolidation of Virtual Machines in Self-Organizing Cloud Data Centers", *IEEE Transactions on Cloud Computing,* Vol. 1, No. 2, pp 215 – 228
8. Chadi Assi, Sara Ayoubi, Samir Sebbah & Khaled Shaban, 2014, "Towards Scalable Traffic Management in Cloud Data Centers", *IEEE Transactions on Communications*, Vol. 62, No. 3, pp 1033 – 1045
9. Chen Ran, Shaowei Wang, & Chonggang Wang, 2015, "Balancing Backhaul Load In Heterogeneous Cloud Radio Access Networks." *Ieee Wireless Communications*, pp 42 – 48
10. Devi, C. K., and K. Kousalya, 2018. "Level-Wised Directed Acyclic Graph Scheduling on Cloud Resources." *Journal of Computational and Theoretical Nanoscience* 15.5, pp. 1530-1533.

11. Dong, M.T. and Zhou, X., 2016. “Fog computing: Comprehensive approach for security data theft attack using elliptic curve cryptography and decoy technology.” *Open Access Library J*, *3*(09), p.1.
12. Dam, S., Mandal, G., Dasgupta, K. and Dutta, P., 2018. “An ant-colony-based meta-heuristic approach for load balancing in cloud computing.” ***Applied Computational Intelligence and Soft Computing in Engineering***, IGI Global, pp. 204-232.
13. Akiyama, Soramichi, Takahiro Hirofuchi, Ryousei Takano, and Shinichi Honiden. "Fast wide area live migration with a low overhead through page cache teleportation." ***In 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing***, pp. 78-82. IEEE, 2013.
14. Fei Xu, Fangming Liu, Linghui Liu, Hai Jin, Bo Li & Baochun Li, 2014, “iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud”, *IEEE Transactions on Computers*, Vol 63, No.12, pp 3012 – 3025
15. Xu, M., Tian, W. and Buyya, R., 2017. “A survey on load balancing algorithms for virtual machines placement in cloud computing.” ***Concurrency and Computation: Practice and Experience***, 29(12), p.e4123.
16. Ghomi, Einollah Jafarnejad, Amir Masoud Rahmani, and Nooruldeen Nasih Qader, 2017. "Load-balancing algorithms in cloud computing: A survey." *Journal of Network and Computer Applications* 88, pp. 50-71. https://doi.org/10.1016/j.jnca.2017.04.007.
17. Duan, H., Chen, C., Min, G. and Wu, Y., 2017. “Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems.” ***Future Generation Computer Systems***, 74, pp.142-150.
18. Priya, V., Kumar, C.S. and Kannan, R., 2019. “Resource scheduling algorithm with load balancing for cloud service provisioning.” ***Applied Soft Computing***, 76, pp.416-424.
19. Mehta, A., Malik, K., Gunturi, V.M., Goel, A., Sethia, P. and Aggarwal, A., 2018. “Load Balancing in Network Voronoi Diagrams Under Overload Penalties.” ***In International Conference on Database and Expert Systems Applications***, Springer, Cham. pp. 457-475.

20. Ibrahim E, El-Bahnasawy NA, Omara FA. 2016, “Task scheduling algorithm in cloud computing environment based on cloud pricing models.” In 2016 *World Symposium on Computer Applications & Research* (WSCAR), pp. 65-71.
21. Hashem, W., Nashaat, H. and Rizk, R., 2017. “Honey bee based load balancing in cloud computing.” ***KSII Transactions on Internet & Information Systems***, 11(12).
22. Sivagami, V.M. and Kumar, K.E., 2018. “Performance analysis of Load balancing algorithms using LBaaS.” ***International Journal of Research and Analytical Reviews (IJRAR)***, 5(3), pp.140-150.
23. Mishra, S.K., Sahoo, B. and Parida, P.P., 2020. “Load balancing in cloud computing: a big picture.” ***Journal of King Saud University-Computer and Information Sciences***, 32(2), pp.149-158.
24. Han, P., Ye, Q., Jiang, X. and Chen, Z., 2018. “Research on Improved Load Balancing Algorithm of Weighted Round Robin.” ***Journal of Changchun University of Science and Technology (Natural Science Edition)***, (3), p.28.
25. Megharaj G, Mohan KG. 2016, A survey on load balancing techniques in cloud computing. *IOSR Journal of Computer Engineering* (IOSR-JCE). Vol. 18(2), pp. 55-61.
26. Moganarangan N., R,G, Babukarthik, S, Bhuvaneswari, M,S, Saleem Basha, P, Dhavachelvan, 2016, “A novel algorithm for reducing energy consumption in cloud computing environment: Web service computing approach.” *Journal of King Saud University – Computer and Information Sciences*. –Vol. 28, pp. 55–67.
27. Pasha, Nusrat, Amit Agarwal, and Ravi Rastogi. 2014, "Round robin approach for VM load balancing algorithm in cloud computing environment." *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, no. 5, pp. 34-39.
28. Dhari, A. and Arif, K.I., 2017. “An efficient load balancing scheme for cloud computing.” ***Indian Journal of Science and Technology***, 10(11), pp.1-8.
29. Liu, J., Zhang, N., Kang, C., Kirschen, D.S. and Xia, Q., 2017. “Decision-making models for the participants in cloud energy storage.” ***IEEE Transactions on Smart Grid***, 9(6), pp.5512-5521.

30. Ejaz, S., Iqbal, Z., Shah, P.A., Bukhari, B.H., Ali, A. and Aadil, F., 2019. "Traffic load balancing using software defined networking (SDN) controller as virtualized network function." ***IEEE Access***, 7, pp.46646-46658.
31. Ruitao Xie, Yonggang Wen, Xiaohua Jia & Haiyong Xie, 2015, "Supporting Seamless Virtual Machine Migration via Named Data Networking in Cloud Data Center", *IEEE Transactions on Parallel and Distributed Systems*, Vol.26, No.12, pp 3485 – 3497.
32. Babu, K.R. and Samuel, P., 2018. "Interference aware prediction mechanism for auto scaling in cloud." ***Computers & Electrical Engineering***, 69, pp.351-363.
33. Singh, Priyanka, Palak Baaga, and Saurabh Gupta. 2016, "Assorted Load Balancing Algorithms in Cloud Computing: A Survey." *International Journal of Computer Applications*. Vol. 143, No. 7, pp. 34-40.
34. Volkova, V.N., Chemenkaya, L.V., Desyatirikova, E.N., Hajali, M., Khodar, A. and Osama, A., 2018. "Load balancing in cloud computing." ***In 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)***, pp. 387-390.
35. Ali, H.G.E.D.H., Saroit, I.A. and Kotb, A.M., 2017. "Grouped tasks scheduling algorithm based on QoS in cloud computing network." ***Egyptian informatics journal***, 18(1), pp.11-19.
36. Naqvi, S.A.A., Javaid, N., Butt, H., Kamal, M.B., Hamza, A. and Kashif, M., 2018. "Metaheuristic optimization technique for load balancing in cloud-fog environment integrated with smart grid." ***In International Conference on Network-Based Information Systems,*** Springer, Cham. pp. 700-711.
37. Mansour, D., Osman, H. and Tschudin, C., 2020. "Load Balancing in the Presence of Services in Named-Data Networking." ***Journal of Network and Systems Management***, 28(2), pp. 298-339.
38. Gangadhar, P.V.S.S., Rao, M.V., Hota, A.K. and Rao, V.V., 2017. "Distributed Memory and CPU Management in Cloud Computing Environment." ***International Journal of Applied Engineering Research***, 12(24), pp. 15972-15978.
39. Velde, V. and Rama, B., 2017. "An advanced algorithm for load balancing in cloud computing using fuzzy technique." 2***017 International Conference on Intelligent Computing and Control Systems (ICICCS).*** pp. 1042-1047.

40. Guo, M., Guan, Q. and Ke, W., 2018. “Optimal scheduling of VMs in queueing cloud computing systems with a heterogeneous workload.” ***IEEE Access***, 6, pp.15178-15191.
41. Kaur, A., Kaur, B. and Singh, D., 2017. “Optimization techniques for resource provisioning and load balancing in cloud environment: a review.” ***International Journal of Information Engineering and Electronic Business***, 9(1), p.28.
42. Fahim, Y., Rahhali, H., Hanine, M., Benlahmar, E.H., Labriji, E.H., Hanoune, M. and Eddaoui, A., 2018. “Load Balancing in Cloud Computing Using Meta-Heuristic Algorithm.” ***Journal of Information Processing Systems***, 14(3).
43. Raj, P.H., Kumar, P.R. and Jelciana, P., 2018. “Load Balancing in Mobile Cloud Computing using Bin Packing’s First Fit Decreasing Method.” ***In International Conference on Computational Intelligence in Information System,*** Springer, Cham. pp. 97-106.
44. Kumar, M. and Sharma, S.C., 2017. “Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing.” ***Procedia computer science***, 115, pp.322-329.
45. Ragmani, A., El Omri, A., Abghour, N., Moussaid, K. and Rida, M., 2018. “A performed load balancing algorithm for public Cloud computing using ant colony optimization.” ***Recent Patents on Computer Science***, 11(3), pp.179-195.
46. Chen, S.L., Chen, Y.Y. and Kuo, S.H., 2017. “CLB: A novel load balancing architecture and algorithm for cloud services.” ***Computers & Electrical Engineering***, 58, pp.154-160.
47. Kang, B. and Choo, H., 2018. “An SDN-enhanced load-balancing technique in the cloud system.” ***The Journal of Supercomputing***, 74(11), pp.5706-5729.
48. Krishnadoss, P. and Jacob, P., 2018. “OCSA: task scheduling algorithm in cloud computing environment.” ***International Journal of Intelligent Engineering and Systems***, 11(3), pp.271-279.