

# Dynamic Load Balancing by Round Robin and Warshall Algorithm in Cloud Computing

Nikhata Azhar, Mohd. Haroon

**Abstract:** Cloud Computing is a well-known technology that delivers scalable, fault-tolerant, consistent, secure and reliable computational services. Cloud figure out grants its services in subscriptions means user have to pay as much they use. Due to the continuously progressing and increasing area, cloud figure out is a very good topic for researchers. The main area of interest in cloud figure out is virtualization, collates, software defined network, network function virtualization and many more. There are limited load collates algorithms are used in cloud, and all algorithms having its own advantages and disadvantages. This is the requirement of time to invent a new efficient algorithm in load collate in cloud figure out, through which we can get maximum throughput, minimum response time, etc. Here in this research paper authors are proposing a new approach of dynamic load collate, which mainly depends on the distance between datacenter to host. Simulation is done by CloudSim toolkit. Authentication and effectiveness of the proposed load collate algorithm are checked by comparative analysis with existing load collate algorithm.

**Index Terms:** load collate, cloud figure out, host, virtual machine, cloud simulator

## I. INTRODUCTION

In the simplest way, we can say that cloud figure out is a system in which figure out services are achieved by the user via the internet, on a subscription manner. We will use Cloud Figure out and Cloud words interchangeable within the paper. We will use load collate and load balancing words interchangeable within the paper. There are services provider and the users, and cloud figure out is a web structure between the user and the services provider to make every figure out work easy. There is no need to establish a setup to avail any service as cloud figure out provides services at every level in the form of IAAS, PAAS, and SAAS [1].

The standard definition of Cloud Figure out, given by NIST in 2011 is:

"Cloud Figure out is a model for enabling convenient, on-demand network access to a shared pool of configurable figure out resources (e.g. networks, servers, storage, application, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

**Revised Manuscript Received on June 15, 2019.**  
Nikhata Azhar, Mohd. Haroon, Computer Science and Engineering Department, Integral University, Lucknow, India.

Due to the easy process of installation and simple work system, consumers of cloud figure out are increasing tremendously. The users' quantity is directly proportional to the increased load on the resources. Every system has a performance index, which is in the cloud is mainly based on the users and the services provided by the cloud. When the number of clients is continuously increasing and the quantity of resources is limited than a situation came, which shows a bad effect on system's performance but in the cloud, there is a solution for such circumstances in the form of scalability, virtualization, and load collate. Scalability is a characteristic of cloud figure out and Virtualization and load Collate are the core concepts of Cloud Figure out.

Nowadays two technologies are swaying in the computing environment, Cloud Computing and Machine learning. Authors have studied many techniques of machine Learning (ML) [2]. In ML, to train a machine a big amount of computational power is required. With the help of cloud figure out, one can access such high computational power for the required period on the pay on use way.

## Cloud Computing Characteristics

Cloud figure out characteristics are as following [3]:

- 1) *On-demand self-service*-- cloud figure out provides services to the user on the subscription-based manner as we have to pay for the services until we are using it.
- 2) *Resources pooling*--multiple users can use the same physical hardware with the multi-tenant aspect of cloud figure out, The basic idea behind the pooling of the resources is virtualization approaches.
- 3) *Rapid Elasticity*-- No. of cloud resources can be added or reduced based on the demand for that resource. There are two types of scaling exist.
  - a) *Horizontal Scaling (scaling out).*
  - b) *Vertical Scaling (scaling up).*
- 4) *Measured services or Pay per use*-- In cloud figure out some specific metrics are used to charge the user. The metrics are the number of used CPU cycles, storage space used during the whole process.

- 5) *Broad network access*—By applying standard access mechanisms any user can access any required cloud figure out resources over the network.

## Types of Cloud Computing Services

Authors Studied many papers related to Cloud figure out and Grid figure out to get the deep knowledge of models of these computing systems [4]. Cloud figure out system depends on two perspectives: Capability and Accessibility.

- 1) *Based on Types of capability or the services provided by the cloud figure out system*— Cloud System provides three different types of service models :
  - a) *Software as a service*
  - b) *Platform as a service*
  - c) *Infrastructure as a service*
- 2) *Based on Types of accessibility also known as Deployment Models*— Cloud System provides three different types of deployment model as follows:
  - a) *Public cloud system*
  - b) *Private cloud system*
  - c) *Hybrid cloud system*
  - d) *Community cloud system*

## Advantages of Cloud Computing

These are the advantages of cloud figure out

- 1) *Facilitate Zero-downtime rolling updates to the web server*
- 2) *Facilitate an immediate increase in capacity*
- 3) *Enhance Fault Tolerance*
- 4) *Reduce load on web servers through SSL termination*
- 5) *Facilitate Just In Time Load Collate*

## Cloud Computing Concepts And Enabling Technologies

- 1) *The key concepts behind Cloud Figure out are as follows:*
  - a) *Virtualization*

In Cloud Figure out scenario, Virtualization refers to the division of physical resources (such as Processor, Disk, Memory, and Network) into multiple virtual resources. The latest definition of Virtualization is "Virtualization is a technique which converts hardware into software". The multi-tenant aspect of cloud figure

out allows the different user to use the same physical hardware sand different virtual devices.

### b) Load Collate

Load collate is also a very useful concept of cloud figure out, the numbers of users or end users are continuously increasing, so in the same ratio load on the data centers are increasing .load collate is a concept in which load at the Datacenter distributed to a different host by the load balancer to maximize the resources utilization and user's satisfaction. When the load at any resources reached to the maximum tolerable limit then the load is a shifted from that resource to the respective resource of another host. To balance the load among all hosts of a datacenter of the cloud system becomes a major threat in cloud figure out.

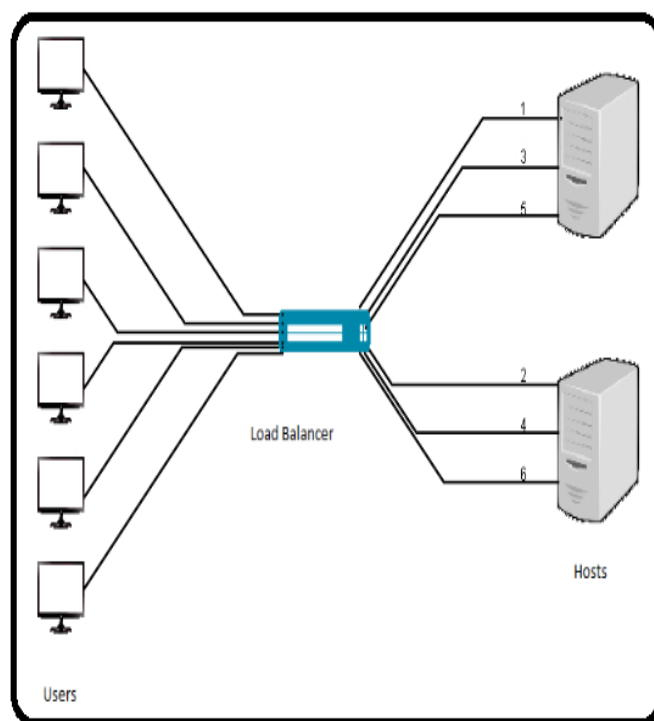


Fig.1: Load Collate

Algorithms of Load Collate are particularly of two types:

### 1) Static Load Collate Algorithms:

In static load collate algorithm, all the load distribution among all computing node done before the execution of the system. In static load collate load distribution started before the computation. In static load collate for input job statically distributed to all available computing machine. the distribution is fully based on the history of the machine[5]. if any computer machine having the good computing power and good memory the load balancer will allocate a higher amount of data.

## 2) Dynamic Load Collate Algorithms:

In dynamic Load Collate, load distribution started at run time. It means after the execution of the program .it means well all computing nodes at any moment if any computing node declares I am overloaded, At that moment dynamic Load Balancer will take the responsibility and migrate the excess loan from overloaded system to under loaded system. Several techniques are in use for load Migration of excess load from overloaded system to under loaded system.

## Metrics for Load Collate

These are the metrics for Load Collate [6].

- 1) *Throughput*: It is the calculation of completed task within particular time duration. If the throughput of a system is high means the system's performance is very good.
- 2) *Fault-Tolerance*: A system is said to be fault tolerance if it recovers from failure very speedily.
- 3) *Migration Process*: Migration is the transfer of resources from one host another host. For a good Cloud Figure out system, migration time should be less. It is of two types, task migration, and resources migration.
- 4) *Scalability*: It is the quality of a load collate algorithm that if we added more resource, there is no effect on its efficiency.
- 5) *Overhead*: Overhead is the extra charge paid to maintain the efficiency of a load collate algorithm. For a good algorithm, it should be less.
- 6) *Resource Utilization*: Resource utilization means all the resources are in the working state. Neither any resource should be overload nor should any resource be idle. For any efficient load collate algorithm, it should be maximum.
- 7) *Carbon Emission*: As much as energy is consumed by a system, Carbon emission will be more, As the number of resources increases due to more energy consumption, carbon emission is increased. Authors studied many papers on energy conservation in cloud environment and related fields as IoT [7, 8].

## Various Existing Load Collate Algorithms

### 1) Round-Robin Algorithm:

It is a very easy and well-known, static task scheduling based method, which is mainly, depends on the specific

time quantum. The task is allocated to the node (VM) in a circular manner for a predefined time quantum [9]. Every node has to finish its task within the quantum, otherwise has to wait for the next turn.

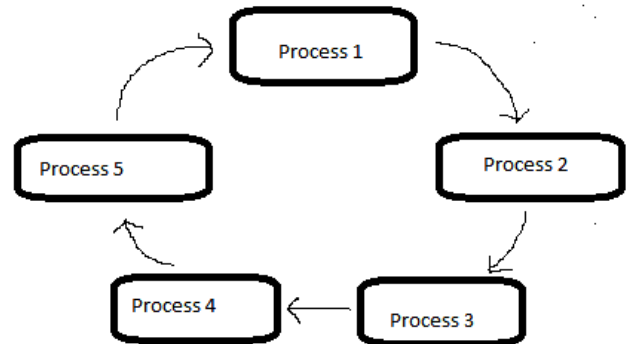


Fig.2: Round Robin Process

### 2) Min-Min Algorithm:

In Min-Min Algorithm, load balancer checks the burst time of the coming request and the completion time of the resources [10]. After that load balancer allocates the request with minimum burst time to the resource with the minimum completion time. In Min-Min algorithm, big size requests suffered from the starvation problem.

### 3) Max-Min Algorithm:

In Max-Min Algorithm, load balancer checks the burst time of the incoming request and the completion time of the resources [10]. After that load balancer allocates the request with maximum burst time to the resource with the minimum completion time. In Max-Min Algorithm, small size requests suffered from the starvation problem.

### 4) Modified Active Monitoring Algorithm:

Modified Active Monitoring algorithm is based on Active monitoring algorithm. The core point of this load Collate algorithm to find the VM with least Load along with the process to allocate free VM to any request so that optimum utilization of the virtual machine is done [11]. It checks the least loaded VM to serve the request simultaneously considering the previous allocation. If the least loaded VM is previously used; it goes for next least loaded VM for the user request.

### 5) Fair Round-Robin:

The proposed algorithm follows a simple concept of implementation of different Time Quantum. Which can be deduced by the help of the algorithm execution round? In this algorithm, first of all, ready-queue initialized burst time lists, Time Quantum (TQ), VM-state list, no of rounds [11].

If algorithm execution round is even then

$TQ = \min$  (Burst time list)

Otherwise  $TQ = \text{avg}$ (Burst time list)

Then allocate the VM to request in the ready queue according to the no. of current round.

#### 6) Reverse Round Robin Filling algorithm:

This algorithm is like the traditional RR algorithm else the condition that in traditional round robin when any request came at the load balancer, it allocates the last VM to the first request. In this algorithm, a hash table is created at the load balancer, which contains the entry of the last VM along with its availability means VM is free or busy at the first place and when any request comes, then that last VM (if free) is allocated to the request [12]. Otherwise, VM at the next place is checked.

#### 7) Round-Robin with Server Affinity

In this algorithm, the load balancer has organized data in two way:

- Hash map*: It stores the information about the userbase request and the last VM allotted to that request.
- Status list of VMs*: It shows the availability of each VM (i.e., Busy/Free).

In this algorithm, When a request comes at the load balancer, it checks the entry of that Userbase request and along with the VM's status which one last time served that request. If that VM is free, then that UserBase request is allotted to the VM. So if any userbase request comes many times, still the algorithm runs only once. It saves a considerable amount of time. Comparison Table (1) presents Load Collate Algorithms on Pre-defined Parameters.

Cloudlets, DataCenter(DC), DataCenterBroker(DCB), Host and VM are the main structural elements of cloud system. the datacenter contains a number of hosts, under a datacenter, many hosts came. Virtual Machine is the main processing element, those come under Host. The requests are actually processed by the VMs. When datacenterbroker got a job from cloudlet, load collate algorithms applied it, will decide which VMs to be allocated the incoming request. Any existing algorithms do not check the distance between the datacenter and the hosts' .when a new request came, rather than check nearest available free VM, the request is sent to the VM, which may be far from the datacenter, due to basic Round-Robin algorithm, which increases the total response Latency and datacentre processing time. The aim of the proposed algorithm is to decrease the datacenter processing time and increase the Response Latency by modifying the traditional Round Robin load collate algorithm. In the proposed algorithm, Authors are using

Round robin with Floyd-Warshall (Shortest path algorithm).

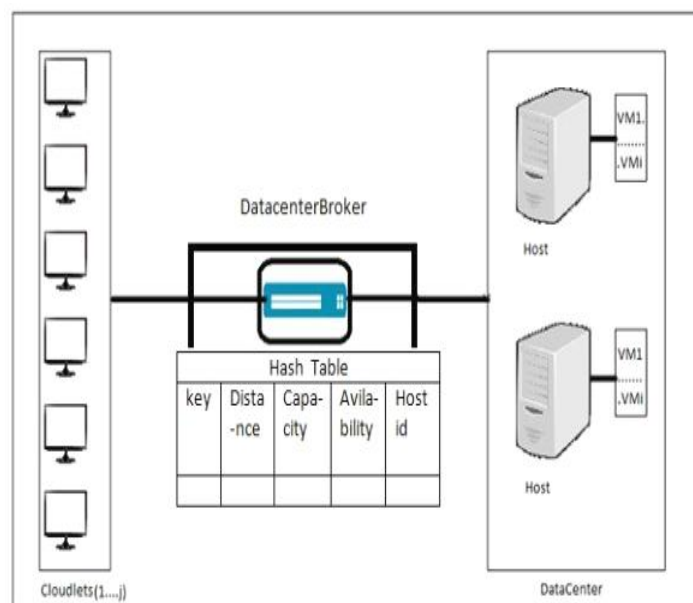


Fig.3: System Architecture of the Proposed Algorithm

#### Motivation and Problem Statement

As the quantity of internet user is increasing, in the same proportion cloud service provider are increasing so as a result, the completion between these cloud service providers (CSP) is increasing. It is mandatory for the CSPs to improve their facility by improving those parameters on which quality of their services depends as response time, request processing time and many others [13]. In Cloud figure out, there is a concept of the Service Level Agreement (SLA).

An SLA in cloud figure out confirms the level of services that are formally defined at the time of contract with the cloud service provider [14]. SLA contains a number of performance metrics and the corresponding service level objectives. With the production of huge quantity of data means big data, cloud figure is mandatory to store and manage that data properly. So CSP like Microsoft, Google and AWS are introducing their big data system [15].

They can get this by using efficient VM load collate policies, but in the present time, only a limited and efficient load collate algorithms are in use. Also in most of the existing load collate policies, a request has to waste time for free VM at the far host, whereas the free VM is available or going to be available at the nearest host.

#### Problem Definition

“To discover an efficient load collate algorithm for the cloud figures out system and obtains the efficiency of the proposed algorithm by the comparative analysis of it with keep going algorithms”.



Through the problem definition, authors get their objective to proceed in a systematic way:

- 1) To review the working load collate algorithms and try to identify the research gap.
- 2) To develop an effective load algorithm for load collate, whose core concept, Authors got from the result of the above study.
- 3) To execute the proposed algorithm on CloudSim and to check its working on parameter as Processing Timing, No. of Responses, etc.
- 4) Compare the proposed algorithm with the existing algorithms to check its efficiency and accuracy.
- 5) Show the result in the form of graphs or tables.

The remaining part of the paper is in the given order:

Section 1: Introduction + Background Study.

Section 2: Literature Review

Section 3: Proposed Work.

Section 4: Result and Discussion

Section 5: Conclusion & Future Scope.

Section 6: Acknowledgement

Section 7: References

## II. LITERATURE REVIEW

The core concept of this [1] paper is the explanation of the three load collate techniques with their architecture and their advantage and disadvantage. There are three kinds of load collate architecture based on the spatial distribution of their nodes.

- A. *Centralized load collate architecture.*
- B. *Decentralized load collate architecture.*
- C. *Hierarchical load collate architecture.*

Through this paper [3], we came to know many load collate algorithm, differentiate on the principle of the working environment and the Spatial Distribution of the cloud nodes.

On the Basis of Spatial Arrangement of the Cloud Nodes Load Collate Algorithms are Divided into Three Parts

- 1) *Centralized load collate*
- 2) *Distributed load collate*
- 3) *Hierarchy load collate*

This [5] is a very fundamental paper with respect to my research, through this paper, we get the answer of many fundamental questions about the load collate as what is the need of load collate phenomena in cloud figure out, objectives, and types of load collate algorithms. through this paper, we came to know the relationship between load collate and Green figure out.

Authors described 28 load collate techniques in the paper, Authors are explaining some of them.

### A. Two-phase (OLB + LBMM) Algorithm

Two-phase(OLB+LBMM)algorithm is a combination of two algorithms opportunist load collate (OLB) with Min-Min (LBMM) algorithm to get better results for load collate. The main quality of OLB algorithm is to put every node busy, and through LBMM we get the minimum completion time for any request.

### B. Central Load Collate For Virtual Machines

This algorithm is specially designed for distributed virtual machine environment. This is a good algorithm but the drawback of this algorithm is that the algorithm doesn't have a concept for recovery from failure.

### C. Carton

As the name suggests CARTON (a small container to keep goods) It combines the main points of load collate (LB) and distributed rate limiting.

### D. Vector Dot

The main concept of this algorithm is to handle the hierarchical complexity of the data center. Multidimensionality of resource load across servers also handles by this algorithm.

A very conceptual paper [6], for researchers, who are interested in the fundamentals of load collate in clouds. The author has published a paper based on load collate concept with the title "Load Collate in Cloud Figure out using ModifiedThrottled Algorithm" in 2013.

In this paper, the authors presented a new algorithm works on the accessibility of VM.

This paper [7], describe in detail about the Energy Efficient Workflow Scheduling in Cloud figure out Environment and also explains its reliability.

This paper [9], gives a very deep knowledge about the traditional Round Robin algorithm and also explains the reason behind the origin of the weighted Round Robin Algorithm.

In this paper, the author described many other Load Collate algorithms as least connection, weighted least connection, etc. The author proposed a new algorithm, which seems to be like a Traditional RR algorithm but the difference is that the Load Collate in this improved algorithm is based on context switching concept. Authors compare their work with one, two and three server and reached at the conclusion that as the number of the server increases response time and the average waiting time is also decreased.

In this paper [10], Min-Min and Max-Min load collate Algorithms are explained and compared by the authors. A very good paper to understand the working of the above algorithms. After describing both algorithms, authors come on simulation result to check the performance of both algorithms. They are using cloudsim toolkit simulator for their analysis. Through all the above study, we got that over makespan Max-Min gives good result than Min-Min. For any good algorithm, makespan should be less.

A very good paper [11] to understand the basic Round Robin algorithm and many other algorithms based on the existing Round Robin algorithm. The suggested algorithm follows a simple concept of implementation of Time Quantum based on algorithm execution round. In this algorithm, first of all, we initialize ready-queue, burst time list, Time Quantum (TQ), VM-state list, no of the round. If algorithm execution round is even then

$TQ = \min(\text{Burst time list})$   
Otherwise  $TQ = \text{avg}(\text{Burst time list})$

Then we allocate the VM to request in the ready queue. The author using Cloud Analyst tool for simulation Fair Round Robin algorithm.

A paper [12], which is based on basic Round Robin algorithm. In the modified algorithm, when the request comes at the load balancer, it parses the VM's table in reverse order. When it got the appropriate VM, then check its status, if that particular VM is free, then allocate the request to the chosen VM.

From the experimental result, we can see that the improved or Reverse Round Robin Filling algorithm is giving better result in comparison to basic RR algorithm, especially in two criteria response time and cost.

The main attraction of the paper [13], is the comparison between two algorithms, one is round Robin and another one is Throttled. Cloud Analyst is used for simulation and result analysis. Analyzing the comparison result, based on the cost and overall response time, we reached the conclusion that data transferring cost of the throttled policy is less than Round Robin.

In this paper [14], the author suggested a weighted Round Robin algorithm with some modification in the traditional Round Robin algorithm. In this proposed load collate technique, the author gave weight to every virtual machine, so that at the time of the load distribution, the load is distributed depending on the weight allocated to the particular virtual machine, so it is a static load collate algorithm. A master-slave concept works in the background. Load Balancer works as the master and VMs work as the slaves.

In this proposed algorithm, when the load comes, it distributed to all VM in RR manner then the slave nodes send information about remaining memory, used CPU cycle, etc to the master node. According to the information provided by the slave node, the Master node builds a weighted table.

In this paper [17], for load collate authors are using the modified Warshall algorithm with SDN(Software Defined Network) Datacenter.

The proposed algorithm is made up of two phases:

### 1. Algorithm to Compute Path Cost:

#### *Modified Floyd-Warshall APSP Algorithm*

The basic Warshall APSP algorithm has modified by authors to know the shortest path with the equal cost within all the vertexes in the given graph.

### 2. Algorithm to select Least Traffic Path:

Authors got many equal cost paths till yet. At this point, they applied an algorithm to search for an edge with the lowest traffic(load). To check the load on every path, the controller sends a probe after a time interval T sec.

With the help of the simulation result, it is proved that this algorithm gives better results than GFF(Global First Fit).

This paper [18], is the milestone paper to understand the importance and working of Clouds. The authors of the paper are also the developers of the CloudSim toolkit. The core aspect of the paper is the development and utilization of CloudSim toolkit.

In this paper [19], the authors explained very important features of cloud figure out, Server Affinity. Server Affinity is the property of a load balancer to save the history of the User base requests means to save the information about the server, who last served that particular request. They use the same concept to discover their new load collate technique (Round Robin with Server Affinity). In the proposed load collate techniques; a hash map is used to stock the history of the User base request. So as a result, when a request came, if the hash map has the entry for that request and the related particular VM is free, load balancer directly allots that VM to that request without run the load collate algorithm again, as a result, it saves a significant amount of time. Cloud Analyst is used for simulation.

## III. PROPOSED ALGORITHM: ROUND ROBIN WITH WARSHALL

Authors are taking the distance between the Datacenter and the hosts as a parameter in our proposed algorithm. Section 4 describes the simulation result of the algorithm using Clouds toolkit. Section 5 describes the comparison result of the Round Robin algorithm with proposed Algorithm. Flowchart of the proposed algorithm is given in Figure 4. The Datacenterbroker maintains one data structures, which is the Hash table [16]. Hash table: this stores the information related to the VMs.

### A. Basic Mathematical equation of the proposed Load Collate Algorithm: $\text{Balanced\_Load} = N$ , where

$N = \sum_{x=0}^n \beta x Fx (1) \beta x (\sum_{x=0}^n \beta x = 1)$ . It represents the different kinds of requests.  $Fx$ , It represents the Responses.

### A. Algorithm

As the authors stated above their research work is based on the distance between the data centre and the servers (hosts).

Steps of the algorithm are:

- 1) Authors have 1....j cloudlets.
- 2) When any request reached the data center broker, first of all, it checks it's hash table, which contains all information about  $VM_{1...i}$  for each Host. VM id is taken as the key element to distinguish any VM

among all with these qualities as the distance (a distance of host from data center), Capacity(memory of virtual machine), Availability(VM is free or busy)and hosted (id of the host, to which VM is attached).

- 3) If the VM with requires Capacity is free, then after availability checking, datacenterbroker checks the distance and allot the request to the nearest VM. But if the nearest VM is busy, then request is forwarded towards the second nearest VM and the same process is continued till datacenterbroker does not get free VM with the required capacity.
- 4) Along with the distance parameter data center broker checks one more thing, let's suppose most nearest VM is busy, then according to above-mentioned rule, the request should be forwarded, towards the second nearest VM, but here data center broker checks one more thing. If the time to reach the request at the second nearest VM is more than the waiting time at the nearest host then data center broker will forward the request towards the nearest one.
- 5) To calculate the shortest path between the datacenter and host authors are using Floyd-warshall algorithm [17].

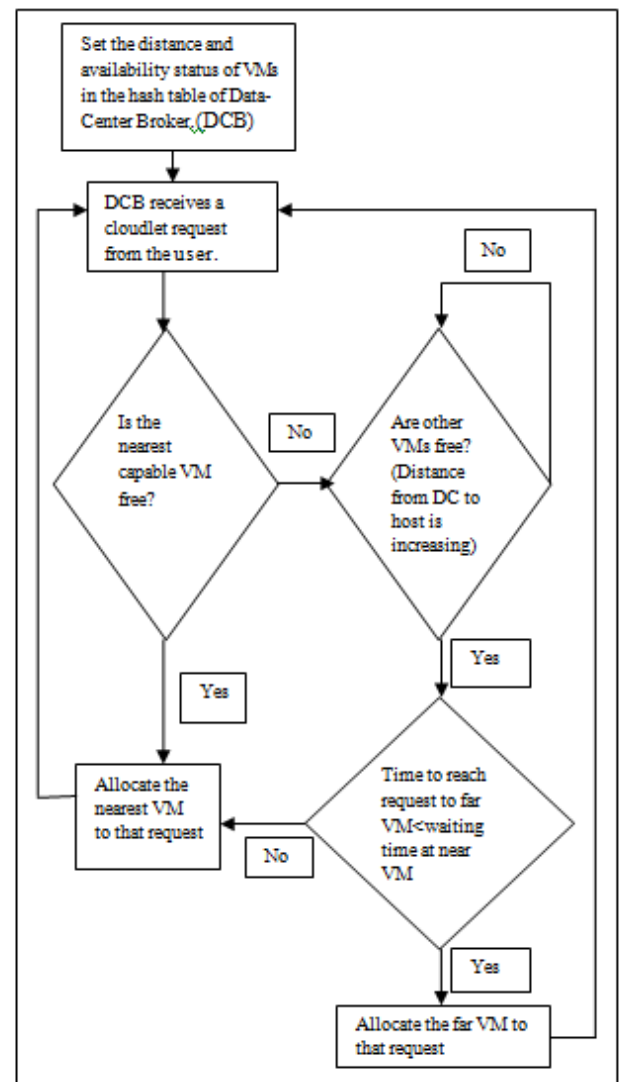


Fig.4: Flow Diagram of round-robin with warshall algorithm.

#### IV. RESULTS AND DISCUSSION

CloudSim 3.0 is used for simulation. CloudSim is run on the Java App NetBeans [18]. The CloudSim contains a large number of classes and libraries like Cloudlet Scheduler, Data Center, Hosts, VM, Data center Broker, VM Scheduler, Datacenter Characteristics, etc. It provides a simulated environment for performing research on virtual machines [19]. Java language is used for coding.

##### A. Configuration and Simulation Result

To get the working and comparative analysis of the proposed algorithm, we considered the scenario of one data centre has four hosts and eight VM. Structures shown in table (2) & (3)

- 1) Cloudlets response time
- 2) Data centre processing time

## B. Simulation of the Proposed Load Collate Method: Round Robin with Warshall Algorithm

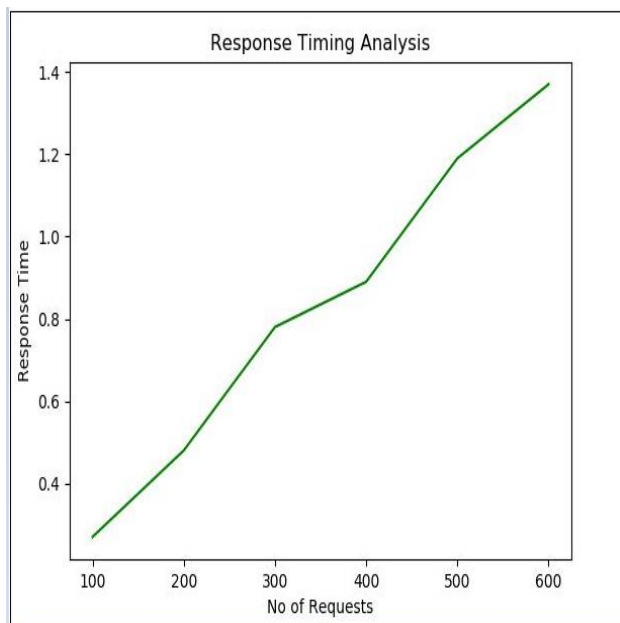


Fig.5: The Graph representing the result of the overall Cloudlet response time(Simulation output: Cloudlet Response Time).

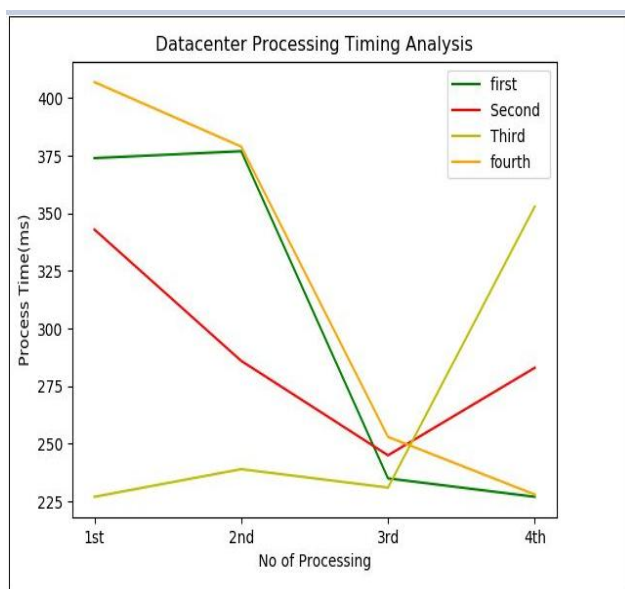


Fig 6. The Graph represents a study of the Datacenter Processing time (Simulation output: Datacenter Processing Time)

From the above graph, we came to know that when the number of requests increasing at any node, the response

time is also increasing. When no. of requests are between 300-400, then the slope of the graph is somewhat straight, it indicates that through the algorithm, we are getting more response in less time. Through graph 1, we can easily understand these variations.

From the graph and the table (4) we can see that in datacenter processing terms first, second and third, the time difference between two processing is not big but in the fourth case, this difference is very much, by this we get the result that Round robin with Warshall algorithm is working good when we do simulation alone. I took the reading at different times.

## C. Comparative Study between the Round Robin and Round Robin with Warshall Algorithm

In this step, we compare the no. of responses given by the load balancer in three conditions

- 1) Without applying any Load Balancer(WLB)
- 2) Round Robin Load Balancer(RRLB)
- 3) Round Robin with Warshall load Balancer(RRWB)

That is shown in table (5)

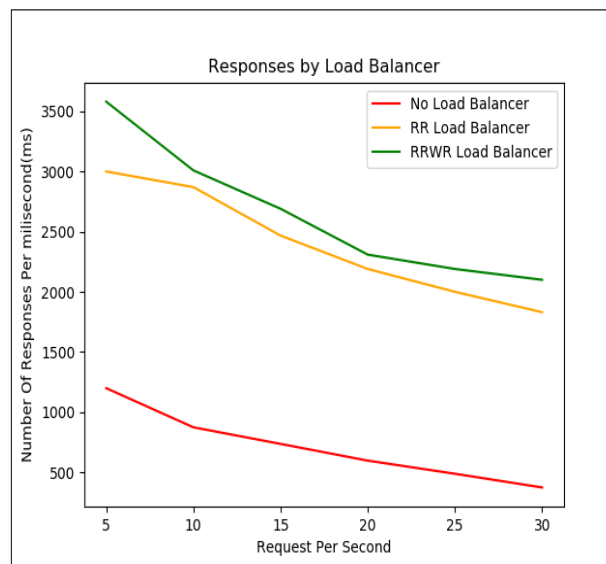


Fig. 7: The Graph Representing a Comparative Study of the No. of Response per millisecond(ms).

In the last, Round robin with Warshall Algorithm achieved more and fast response with respect to Round Robin Algorithm and gives 94% accuracies without any load balancer attached with the datacenter.



## V. CONCLUSION AND FUTURE SCOPE

With the rapid increase in internet users, services provided by cloud figure out are not increasing. So it is not easy to maintain the balance between user's request and the response provided by the cloud figure out. With the help of the Research work, we developed an efficient load collate algorithm. We reached at the result through these phases, Study of existing load collate, Propose a load collate algorithm based on the result of the above study, Execute the algorithm on Clouds and the last compare the result of the proposed Algorithm with the existing Round Robin algorithm to on Predefined parameters to authenticate our work. In future, we will try to test the proposed algorithm on other simulator and high power hardware to get more accurate results.

## ACKNOWLEDGMENT

Authors would like to pay their sincere thanks to Dr. Mohd. Shahid Hussain, Assistant Professor, Ibri College of Applied Sciences, Ministry of Higher Education, Oman, Muzzamil Hasan Assistant Professor, CSE department, M.M.M University of Technology, Gorakhpur, India and Dr. Faiyaz Ahamad, Associate Professor, CSE department, Integral University, Lucknow, India for their continuous guidance in completing the research work done.

## REFERENCES

1. Rajat1, Dr. Sanjeev Kumar, "cloud based load balancing architecture: a study", *IJCSE March 2017-Sept*, vol. 8(2), 2017, pp.112-116.
2. Mamta Mittal, Lalit Mohan Goyal, Jasleen Kaur Sethi and D Jude Hemanth, "Monitoring the Impact of Economic Crisis on Crime in India Using Machine Learning", *Computational Economics*, Springer, 2018, pp. 1-19.
3. Aanjeey Mani Tripathi\*, Sarvpal Singh. "A literature review on algorithms for the load balancing in cloud environments and their future trends", *Computer Modelling & New Technologies 2017*, vol. 21(1), 2017, pp 64-73,
4. Dolly Sharma, Shailendra Singh and Mamta Mittal, "Models in Grid Computing: A Review", *Recent Patents on Engineering*, 2018 Vol 13 (2), pp. 94-100
5. Kumar Mishra, "needs, objective and major challenges in cloud-A Systematic review", *International Journal of Computer Applications*, December 2015, Vol. 131(18).
6. Shreedhar g. Domanal, G. Ram Mohana reddy. "Optimal Load Balancing in cloud computing by efficient utilization of virtual machine". 978-1-4799-36309/14@ IEEE 2014.
7. Ritu Garg, Mamta Mittal, Le Hoang Son, "Reliability and Energy Efficient Workflow Scheduling in Cloud Environment", *Cluster Computing*, 2019, pp. 1-15
8. Mamta Mittal and S.C Pandey "The Rudiments of Energy Conservation and IoT", *Energy Conservation for IoT Devices*, Springer, Singapore, 2019, 1- 17.
9. Ponam Kumari, mohit Saxena, "A round robin based load approach Scalable demands and maximized resource availability", *International Journal of engineering and Computer science*, ISSN: 2319-7242, Val. 5(8), 2016 .
10. Geethu Gopinath PP<sup>(1)</sup>, Shriram K.Vasudevan, "An in-depth analysis and study Load technique in the Cloud environment". *ELSEVIER* 2015.

11. Prashant Maheta, "utilizing round robin Concept for load balancing algorithm at virtual machine level in cloud environment", *conference paper in International Journal of Computer Application*, May 2014, Doi: 10.5120/16332-5612.
12. Divya Chaudhry, Rajinder Singh chiller. "A New Load technology in cloud environment" *International journal of computer application*, vol. 69(23), 2013.
13. Hamid Shoja, Hossein Nahid, Reza Azizi, "A Comparative Survey on Load Balancing Algorithms in Cloud", *5th ICCNT 2014*, 2014, pp.11- 13, IEEE - 33044
14. Supreeth S 1, Shobha Biradar 2, "Scheduling Virtual Machines for Load balancing in Cloud Platform", *International Journal of Science and Research (IJSR)*, India Online ISSN: 2319-7064, vol. 2(6), 2013.
15. Mamta Mittal, Balas, V.E., Goyal, L.M. and Kumar, R, *Big Data Processing using Spark in Cloud*, 43, (Singapore, Springer Nature Pte Ltd.), 2018.
16. NikhatAzhar, Mohd Haroon, "A Novel Method for Load Balancing In Cloud Computing: Round Robin with Floyd-Warshall Algorithm". *International Journal of Science and Research(IJSR)*, vol. 8(2), 2019.
17. Akinniyi Ojo\*, Ngok-Wa Ma\*, Isaac Woungang, "Modified Floyd-Warshall Algorithm for Equal Cost Multipath in Software-Defined Data Center", *IEEE ICC 2015-Workshop on advance in software-defined and context-aware cognitive networks 2015 (IEEE SCAN-2015)*.
18. Rajkumar Buyya, Rajiv Ranjan and Rodrigo N.Calheiros, "Modeling and Simulation of Scalable Cloud Environments and the Clouds Toolkit: Challenges and Opportunities". *published online 24 August in Wiley Online Library*, vol. 41(1), 2011, pp. 23-50.
19. Komal Mahajan\*, Ansuyia Makroo\* and Deepak Dahiya \*, "Round Robin with Server Affinity: A VM Load balancing Algorithm for Cloud-Based Infrastructure". *J Inf Process Syst*, vol.9(3)

## AUTHORS' PROFILES



**NikhatAzhar:** NikhatAzhar born in Lucknow, India on 19<sup>th</sup> March 1985 has achieved her B.E. Degree in Information Technology from Maharana Pratap College of Technology, Gwalior (M.P.) India in the year 2008. She is currently pursuing her M.Tech Degree in Computer Science & Engineering at Integral University, Lucknow (U.P.) India.

Her first paper was published in Second Conference on Advancement in Computer Engineering and Information Technology (ACEIT-2018) with the title "Review of Different Approaches and Analysis of Big Data Security Issues" organized by Integral University, Lucknow, India publisher International Journal Engineering and Management Research. Her second paper (review paper) was published in International Journal of Science and Research (ISSN: 2319-7064) with the title "A Novel Method for Load Balancing In Cloud Computing: Round Robin with Floyd-Warshall Algorithm" in February 2019.



**Mohd. Haroon:** Mohd. Haron born in Basti, India on 20<sup>th</sup> August 1979 has achieved his B.Tech Degree in Computer Science from Dr. Bhim Rao Ambedkar University, Agra (U.P.) India in the year 2003 and M.Tech Degree From Allahabad Agriculture Institute Deemed University Allahabad (U.P.) India in the year 2006. He Did his Ph.D from Teethankar Mahaveer University, Moradabad in 2016. He is currently Working as Associate Professor in Computer Science & Engineering Department at Integral University, Lucknow (U.P.) India. His approximately 20 international paper and 15 national research papers are published. His first paper was published International journal of computer science engineering and information technology research (ISSN: 2249-6831) with the title "Analysis of a Dynamic Load Balancing in Multiprocessor System" in March 2013. His one paper

published in 2015 2<sup>nd</sup> international Conference on Computing for Sustainable Global Development (INDIACom), publisher IEEE in May with the title "Interest Attentive Dynamic Load Balancing in Distributed System" in May 2015. His paper with the title "Performance Enhancement in Grid System" in International Journal of Advanced Research in Computer Science (ISSN: 0976-5697) in April 2018. His research interest include Artificial Intelligence and Machine Learning.

Table I: Comparison Table for Load Collate Algorithms on Pre-defined Parameters.

Algorithm Parameters	Round - Robin	Min - Min	Min - Max	Round- Robin with Server Affinity	Round- Robin with Warshal Algorith m
Type	Static	Static	Static	Dynamic	Dynamic
Performance	Medium	Low	Low	High	High
Throughput	Low	Low	Medium	Medium	Medium
Resource utilization	Proper	Proper	Proper	Proper	Proper
Scalability	Low	Low	Low	Medium	High
Fault- Tolerant	Low	Low	Low	High	High

Table II: The VM Configuration Used for Simulation.

Parameters	Values
Image size of VM	10000
The Memory of VM	512 MB
The Bandwidth of VM	1000

Table III: The Datacenter Configuration Used for Simulation.

Parameters	Values
Architecture	X86
OS	Linux
VMM	Xen
Number of machines	4
Memory per machines	2048 MB
Storage per machines	100000
Available BW per machine	10000
No. of processor per machine	2
Processor's speed	100 MIPS
VM Shearing policy	Time Shared
Cloudlet length	300

Table IV: Represents the Results for the Round Robin with Warshall Algorithm.

No. of processin g (no. of rounds)	When no application is Running			When some Applicatio ns are running
	First( 6:10 am)	Second (11 am)	Third(2 am)	Fourth (3:30 pm)
1 <sup>st</sup>	374	343	227	407
2 <sup>nd</sup>	377	286	239	379
3 <sup>rd</sup>	235	245	231	253
4 <sup>th</sup>	227	283	353	228

Table V: the test results for the Round Robin and the RoundRobin with warshall load collate algorithm

Used Algorithms		No. of Requests(per second)					
		5	10	15	20	25	30
WLB	No.of Response	1200	875	737	598	489	375
RRLB	per Milisecond	3000	2870	2467	2190	2000	1831
RRWB		3580	3009	2690	2309	2190	2100