International Journal of Engineering &Technology, 7 (3.12) (2018) 1071 -1077

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Fibonacci Series based Virtual Machine Selection for Load Balancing in Cloud Computing

Japman Kaur Dhaliwal[1], Mohd Naseem[2*], Aadil Ahamad Lawaye[3], Ehtesham Husain Abbasi[4]

[1]Computer Science and Engineering Department, Thapar Institute of Engineering & Technology, Patiala, Punjab, India
[2, 3]Department of Computer Science, Baba Gulam Shah Badshah University Rajouri, Jammu & Kashmir, India
[4]Fellow Green Economics Institute, UK, & Department of Accounting, College of Business, Rabigh, King Abdulaziz University, Jeddah Saudi Arabia
**Corresponding author E-mail: [2]mohdnaseemshakeel@gmail.com*

Abstract

The rapid advancement of the internet has given birth to many technologies. Cloud computing is one of the most emerging technology which aim to process large scale data by using the computational capabilities of shared resources. It gives support to the distributed parallel processing. Using cloud computing, we can process data by paying according to its uses which eliminates the requirement of device by individual users. As cloud computing grows, more users get attracted towards it. However, providing an efficient execution time and load distribution is a major challenging issue in the distributed systems. In our approach, weighted round robin algorithm is used and benefits of Fibonacci sequence is combined which results in better execution time than static round robin. Relevant virtual machines are chosen and jobs are assigned to them. Also, number of resources being utilized concurrently is reduced, which leads to resource saving thereby reducing the cost. There is no need to deploy new resources as resources such as virtual machines are already available.

Keywords: *Cloud Computing, Load Balancing, Round Robin, Fibonacci sequence, Virtual Machine.*

1. Introduction

Cloud computing is relatively new technique through which various computing resources such as storage, processors, software applications etc. can be used through the internet. Cloud provider is responsible for managing the shared pool of the resources [1]. One of the fundamental characteristic of the cloud computing is its elasticity which means that it can enhance or reduce the computation according to the user requirements [2]. The second most important characteristic is scalability which indicates that it can balance frequently changed demand of storage and processing power of CPU and bandwidth etc. The main problem with the traditional computing is that their resources are scalable but they do not maintain elasticity. The second drawback of traditional computing is those users have to pay based upon requirement. Whereas, in cloud computing, Multiple number of user can use the resources and they have to pay according to consumption of the resources.

1.1 Resource Allocation:

Resources are allocated for flexibility of using them on the demand basis. The primary goal of resources allocation is to avoid the wastage of CPU speed, memory and other resources by keeping the record of overloaded node Resources are allocation using two levels mapping. At Initial level, mapping between the virtual machine and host is performed. Physical servers are also known as the hosts which contain the Virtual machine (VM). Mapping of VM to the host and the process are determined based upon the capacity and the availability of resources. At the second level, mapping of application to the virtual machine is performed.

For execution of any application, some power is required. So, one more responsibility of VM is to afford the power. Figure 1 depicts the cloud computing service models, IaaS (Infrastructure as a Service) includes the infrastructure such as physical or virtual machines, networking devices or local area network, storage systems etc. PaaS (Platform as a Service) provides the platform such as operating system, database, programming etc. SaaS (Software as a Service) or Application means software for the end user, which is ready to use is provided by this model. End user does not have to worry about to the installation, setup and execution of the program; it is already done by the service provider.

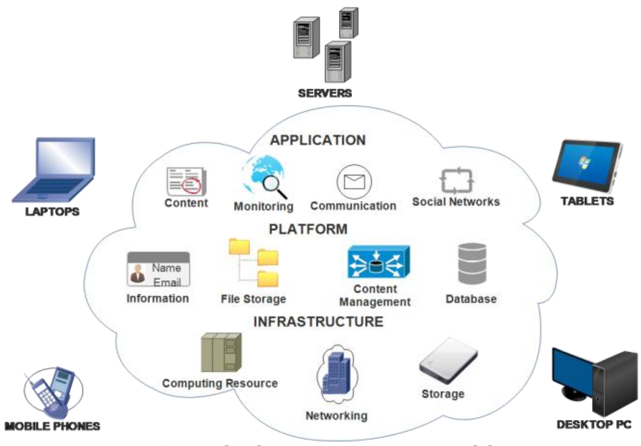

Fig. 1: Cloud Computing Service Model

1.2 Task Scheduling

Once, relevant resources are allocated to a particular task, task scheduling is performed which stipulate the way for resources allocation. Type and name of resources which can serve a particular requirement id decided using resource allocation while method of method of resource allocation is specified using task scheduling it tries to decide whether a resource are individually available or it is already shared. Multiprogramming can be achieved by using shared resources. Task scheduling can be categorized into two types: space shared and time shared. In space sharing mode, resources cannot be preempted while in the case of time sharing mode, resources can be pre-empted. While performing task scheduling, following four cases have to be considered:

I. The VMs and hosts both can be assigned according to time shared basis.

II. Only the VMs are assigned according to the time shared basis but the hosts are assigned on the space shared basis.

III. Only the VMs are assigned according to the space shared basis but the hosts are assigned on the time shared basis.

IV. Both the VMs and hosts are assigned according to the space shared basis.

1.3 Load Balancing

The main purpose of load balancing is to efficiently distribute the total load among the individual nodes so that resource can be effectively utilized and response time of a job can be reduced. Load balancing tries to eliminate any possible condition in which some of the nodes have to carry load that are beyond its capacity while some others nodes are still underutilized. Any algorithm for balancing the load must be dynamic in nature which must not consider the any previous state or behaviour of the system i.e. Load balancing algorithms are only depended upon the present state and behaviour of the system. The most essential part of any load balancing algorithm is to contemplate the following methodology: 1. how to estimate the load. 2. How to analyse of load? 3. Procedure to measure the stability of divergent system. 4. How to measure the performance of system? 5. How different nodes interact among themselves? 6. Selection of target nodes and many others [5]. The parameters of load can be any one or combination of CPU load, delay, network load or amount of memory used etc.

1.4 Purpose of Load Balancing

Load Balancing is the major challenge in any network or system because it adversely affects three main important parameters of the system, i.e., performance, functionality and cost in cloud [6]. The main purpose of resource management technique s is to effectively utilize all resources at minimum cost. It is an approach to evenly disseminate the workload among slug nodes in the network. For instance, we have six identical servers; A, B, C, D, E and F whose relative loads are 95%, 80%, 75%, 40%, 25 and 0%, respectively, of their capacity. Hence, each server on an average would have 52.5% of the total load. LB middleware is an influential technique that is used extensively to enhance the scalability and system throughput in a distributed environment. Why LB is a major concern in cloud when there are many scheduling techniques already exist? Answer is because of its elasticity. The resource provisioning is often provided by the independent companies. As a result it is a work of load balancer to make a decision as to which server component gives maximum profit among the listing of available server components after receiving a particular request [7].

1.5 Process of Load Balancing:

In cloud computing, load comprise of not only traffic generated by cloud provider but also CPU load, delay, network load and memory space of individual servers. A load balancing technique tries to ensure that each node in the network must have same amount of workload at any instant of time. This means that none of the nodes in the network is disproportionately used. Business of server is the main parameter based upon which load balancer scatter the load among different nodes in the network. If a load balancer is busy somewhere, then the client have to wait up to the time at which his process might get a chance to be processed, This process might be too tiring and frustrating for him. Load balancing process require various exchange of information such as waiting time of jobs in queue, processing power of CPU, arrival rate of different jobs etc. Failure of any of these information of the load balancers may leads to serious repercussion, losing the data is one of them.

1.6 Motivation

Cloud Computing is a big topic of research. Now, cloud can provide connection less or connection oriented service to the Internet. After studying the available research in details, we have analyzed several challenges in the cloud computing. Effective load balancing is one the major challenge among others. A lot of load balancing algorithms have been propose in literature but their main concern is to maximize the throughput. In our paper, we have concentrated on enhancement of the system performance by the effective utilization of the VMs. So, we have proposed a new load balancing technique known as improved weighted round robin load balancing algorithm.

The rest of the paper is organized as follows: In Section 2, a brief survey of the existing load balancing algorithm and comparison among them has been presented. In Section 3, we have presented the proposed algorithm. The proposed load balancing algorithm has been simulated and results have been evaluated in Section 4. Eventually, Section 5 concludes the proposed work.

2. Related Work

A number of load balancing algorithms have been already proposed in the cloud computing using simulation software such as Cloud Analyst & CloudSim. A short survey of existing load balancing algorithms is given as follows:

Dharmesh Kashyap and Jaydeep Viradiya in [9] has proposed a throttled Algorithm which maintains a table of virtual machines (VM) along with their current states whether VM are used by someone or it is available for allocation. The services of VM are requested by any of the client/server machines. The groups of VM are managed by data centre. The main job of the data centre is to allocation of the VM by load balancer. The responsibility of load balancer is to maintain a stack of available VM so that it can be given to on demand. If load balancer is able be find the free VM then only it loads data centre. If pertinent VM is not available, which is required by a client, then load balancer returns -1 to the data centre. The requests of VM are maintained in a queue by the data centre. When the allocated task to a particular VM, it must broadcast an acknowledgement message to the data centre. Now it is responsibility of data centre to de-allocate this particular VM so that it would be able to serve some other request.

K Dhakad in [10] has proposed a Equally Spread Current Execution (ESCE) algorithm which tries to disseminate the execution of load on several VMs. It manages a table of VMs in conjunction with currently allocated number of request to a particular VM. If data centre request for execution, then this algorithm inspect the maintained table to find the particular VM which currently have least load. If number of VMs is found, then first VM in the table is preferred and designate for execution of the given request. Now, index table is updated to reflect the increased number of allocation on the selected VM. When the selected VM will accomplish the allocated request, then index table will be updated by decreasing the total count of allocated request for a particular VM by one.

This algorithm has the disadvantage of additional overhead of computation for repeatedly searching the index table.

D Saranya et al. in [11] have proposed an Active Monitoring Load Balancing (AMLB) Algorithm which maintains a table of VMs in conjunction with the number of currently allocated requests to individual VMs. Whenever any process makes a request for allocation of a new VM; AMLB distinguished the least loaded VM from all available VMs. If more than one VM is found, the first pinpointed VMs is preferred. It returns the VM identity to the Data Centres which request the selected VM using its identification and inform about this new allocation to the Active VM Load Balancer. This algorithm gives importance only on the current load of VM. It does not consider processing power. Due to this reason, there is more probability of increase in the waiting time of some jobs which violates the requirement of QoS. [11]

S. Aslam and M. A. Shah in [8] have proposed straightforward and static scheduling technique known as Round Robin algorithm (RRA) for load balancing which is a most classical and still widely used. RRA works on the principle of time slices i.e. dividing time into number of intervals and these time intervals are allocated to different VM. RRA arbitrary selects any of the VMs and allocate the existing VMs on round robin basis on demand. The request for first VM from group is designated randomly. Then data centre assign this VM in a round robin order. RRA disburse requests of clients throughout the servers. The RRA assigns a request of client to the individual servers one by one. Once, all available servers are assigned at least one time, then only RRA return back to the starting of the list and again distribute the request from the first server. This algorithm can be easily implemented but main problem with this algorithm is that it is not able to disseminate the load efficiently in a scenario where number of request varies randomly. Because of this problem, number of researchers is trying to modify RRA so that it becomes more robust which can accommodate dynamic request of load.

L Datta in [12] hace proposed a Weighted round robin load balancing algorithm (WRRA) which assign the load to a server based upon weight. Hence, server having the highest weight is preferred for serving a request because it can handle more traffic as compared to other available servers. Although WRRA performs better as compared to RLBA, but there is no meaningful enhancement in load distribution. The main disadvantage of the RRA and WRRA is that both are static load balancing algorithms which do not scrutinize the current status of server during load distribution. Due to this reason, WRRA performance degrades for enormous versatile client. Author proposed a Dynamic load balancing algorithms (DLBA) which pay attention on the current status of server while disseminating the load among the servers. This is the main advantage of DLBA as compared to RRA and WRRA In [12], author has also shows that WRRA reduces the network cost.

G. Kanagaraj et al. in [13] have proposed a very popular Adaptive load balancing algorithms (ALBAs) which can dynamically change its behavior based upon current changing load. This dynamic behavior includes real time server statistics which can be loaded servers while they are running. But, the main disadvantage of ALVAs is that it requires too much computation as compared to the other existing algorithms.

Y Zhao et al. in [14] have proposed Prediction-based load balancing (PBLB) approaches which distribute the load based upon the prediction, it perform better as compared to RRA and WRRA load balancing algorithms. PBLB uses Support vector machine (SVM) machine learning model for prediction of coming load. But this algorithm suffers from higher computational overhead. S. G. Damanal and G. R. Reddy in [15] have proposed optimal VM Assign Load Balancing Algorithm for efficient utilization of virtual machines to distribute and assign the workload on the least loaded virtual machines [15].

The some other existing load balancing algorithms have been compared as follows:

1. Round Robin algorithm selects the first node in a arbitrary manner but after that it assigns request to all other servers in a circular manner. This algorithm suffers from uneven load means some nodes are overloaded while some other are underutilized. The main reason for overloaded node is that it is not possible to know the running time of any job prior to its execution.
2. Weighted Round Robin algorithm assigns a weight to each node. It depends on the weight for serving the requests. But precise prediction of execution time of a request is very difficult. Therefore this algorithm does not perform as expected.
3. Central Load Balancing Decision Model (CLBDM) improves the round robin load balancing algorithm. In this algorithm, a connection time between the user and the node is calculated and is compared with the threshold value. Because of dependency of round robin technique, number of connection may go beyond the threshold value.
4. Ant Colony Optimization [15] try to discover the under loaded node while searching a server. This algorithm works in a decentralized manner. Hence, the probability of single point of failure has been reduced. This algorithm can collect the information faster. But, large number of ants in the network may cause congestion. The second problem with this algorithm is that the status of the nodes is not considered once ant visits this particular node.
5. Load Balancing Min-Min (LBMM) [16] first execute with the smallest time. Due to these selection criteria, some jobs may experience starvation. Biased Random Sampling [17] balances the load throughout all system nodes using random sampling. But additional overhead is encountered due to the computation of the walk length.
6. Equally Spread Active Execution (ESAE) algorithm as the tasks are submitted, they are queued. If the task size and the size of the VM match, the job is assigned. This is done by the job scheduler based on the priority.
7. Throttled Algorithm try to distribute the workload on the virtual machines so that resource can be effectively utilized but it failed to disseminate the load uniformly because it overloads initial VMs while other VMs are underutilized [9].

The main contribution in this paper can be summarized as follow:

(1) We have explained the importance of resource allocation, task scheduling and load balancing in cloud computing

(2) We have presented existing load balancing algorithms available in related work section

(3) We have analysed the shortcomings of available load balancing algorithms in cloud computing.

(4) We have explained the importance of Febbincci sequence numbers for load balancing algorithms.

(5) We have demonstrated the working of our proposed load balancing algorithms using Fibonacci numbers.

(6) We have performed simulation of the proposed load balancing algorithms and have compared the result with existing algorithm to show the improvement in load balancing.

3. Proposed Algorithm

3.1 Improved Weighted Round Robin Algorithm:

The proposed algorithm is one of the most optimal algorithms and jobs are assigned based upon various factors such as processing power of individual VMs, load on the respective VMs and length of the incoming tasks along with their priority. This algorithm works on two phases. The first phase is static phase which try to utilize the processing power of individual VMs, number of tasks on individual VMs and time required for each task. Based upon these parameters, the proposed algorithm decides the allocation of request to appropriate VM. The second phase of this algorithm is dynamic phase which uses the load at each of the VMs and also the above stated information for the assignment of request to the most relevant VM.

For execution of loop, if any task requires more execution time as compared to initially estimated run time, then the proposed algorithm reorder the task according to the idle time slot that can be utilized for other VMs. This algorithm try to distribute the load from heavily loaded VMs. Whenever any of the VM finished its assign task, this algorithm tries to analyse unutilized/underutilized VMs using resource prober. If any unutilized VM is not identified, then no task rescheduling among the VMs has been performed using load balancer. If any unutilized/underutilized VM is identified, load balancer will try to reschedule the assigned task from the overloaded VM to the identified VM. Once, any assigned task to a particular VM is finished and then only resources of VM are reanalysed. This is due to the reason that analysis of resources independently at any instant will increases overhead on the VMs. This technique help in decreasing the number of task rescheduling among the VMs and it also reduces the number of resource probe among the VMs.

3.2 Fibonacci Sequence

Fibonacci is closely connected with nature, science and real life. Therefore, Fibonacci sequence is used in many fields for load balancing. This sequence begins with a_0=0, a_1=1 and next number in sequence is the summation of the previous two numbers. All number in the Fibonacci sequence can be calculated as [20-22]:

$$a_{n+1} = a_n + a_{n-1} \qquad (1)$$

Some of the real life applications of Fibonacci sequence for load balancing are as follow:

1. if we focus on the petals of the flowers, it has been observed that they follow Fibonacci numbers. Reason for this is that petals of flower try to reduce the overlapped area so that it can get more sunlight. On observation, the count of spirals is found to be 13 counter clockwise spirals and 21 clockwise spirals, which are Fibonacci numbers.
2. It has been observed that total numbers of branches in a tree are always Fibonacci number. Each existing branch of tree will obtain a new branch after a certain time. If the tree has only 1 branch at the initial stage, then it will have 1 branch in the next year and in next cycle, it will receive 3 branches. In every cycle, the total number branches of the tree are increased based upon Fibonacci sequence.
3. It has been observed that the family tree of the bees also follow the Fibonacci sequence. The number of child for n^{th} generation of a first male bee G_n equals to the a_n the n^{th} number in Fibonacci sequence.

3.3 Integration with Fibonacci Sequence in the Proposed algorithm:

To formulate how Fibonacci proved to be an advantage in our implementation, we take a real world example to make the process clear. We are considering that number of application required with the following load.

Port = 10kb
Photo = 1mb
Video = 10mb
Live Video = 100mb

Each algorithm handles this load differently and has been explained below:

Static Round Robin will allocate this load to various virtual machines randomly. This leads to overloading of some nodes and others are left underutilized. Load and execution time are not taken into account before distributing the load.

Weighted Round Robin has weights associated with the jobs. It allocated different weights to different jobs instead of giving equal share of processor. Here, fraction of processor time allocated to a particular job is considered as the weight of a job. This does balance load but may lead to exhaustion of resources very fast. For example, if we have 50 cloudlets (jobs), all of these will be utilized thereby rendering the system inefficient.

In the proposed improved weight round robin load balancing algorithm, we have considered an example which required only 35 virtual machines out of 50 cloudlets. So, equal load will be given i.e. 50/5=10.

Following the basic principle of Fibonacci, we will have the following sequence,

0, 1, 2, 3, 5, 8, 13, 21, 34

Further if we continue this sequence, the next number would be 55, which is out of range for our case and will not be considered for this example.

So only, 10 out of 35 virtual machines will be used to handle this load. Remaining 25 machines will be idle.

3.4 Major Advantages:

1. Out of 35 virtual machines, only 10 machines are in use. This increases the efficiency and throughput of the algorithm. Therefore, resources are saved this way.
2. Remaining 25 machines can be used for later processing when resources get exhausted. So there is no need to deploy new virtual machines, as we already have some machines which are not in use.

4. Simulation Result

We have simulated our proposed algorithm using cloudsim and the cloudsim based toolkit. Java programming has been used for the implementation of the application. We have assumed that the single data center consisting of number of VMs has been used for deployment of our application.

4.1 Interface for Balancing Load for Resource Saving

Step 1: Enter the number of cloudlets and number of virtual machines. Note: no. of cloudlets > no. of VM's ; if we take number of cloudlets less than VM's there would be no need for load balancing.

Here we take no. of cloudlets= 50

No. of VM's= 5 (shown in figure 2)

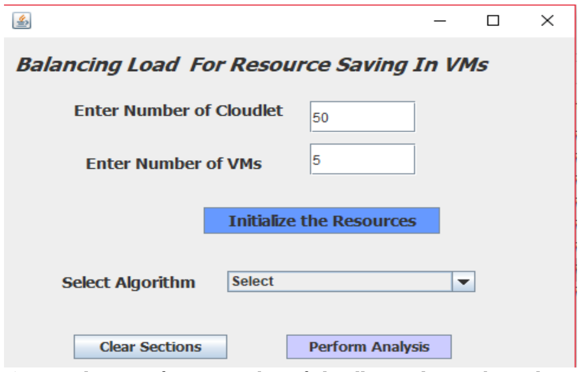

Fig. 2: Selection of input number of cloudlets and VM's by end user

Step 2: Initialize the resources so that simulator takes the input.

Step 3: Select algorithm i.e. Static Round Robin or Improved weighted round robin as shown in figure 3.

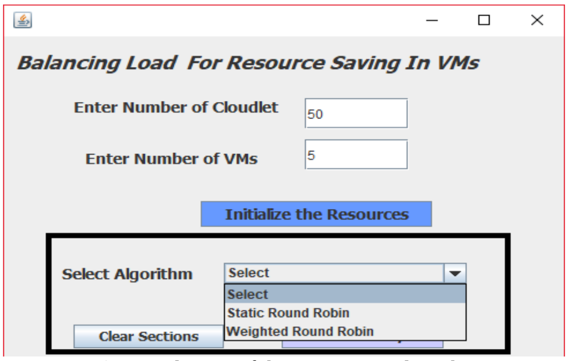

Fig. 3: Selection of the appropriate algorithm

4.2 Results using Static Round Robin

This is in continuation to the values taken before.
No. of cloudlets= 50 No. of VM's= 5
Static Round Robin algorithm is applied and the results are shown as follows.
Figure 4 depicts that five virtual machines (VM-0, 1, 2, 3, and 4) were taken as input and load has been assigned to each virtual machine using Static Round Robin algorithm. It has also been observed that load is not properly balanced between the machines, for example VM-2 has the minimum load and VM-5 has maximum load.
Figure 5 depicts execution time of each cloudlet.

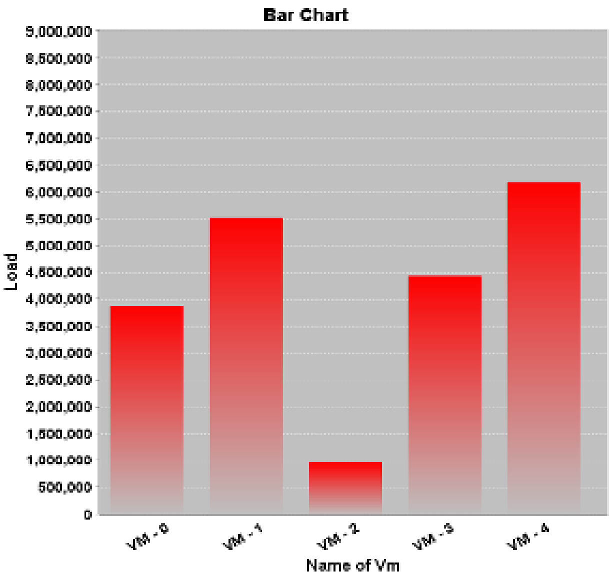

Fig. 4:VM's seclected (x-axis) and Load (y-axis) using Static Round Robin

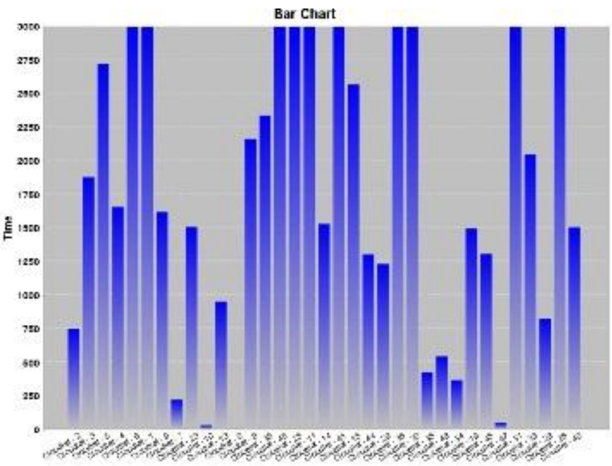

Fig. 5: Cloudlets (x-axis) and execution time (y-axis) using Static Round Robin

Relation between cloudlets in Static Round Robin algorithm:
5 VM's numbered as 0, 1,2,3,4
50 Cloudlets numbered as 0, 1, 2, 3…..49
Table 1 depicts the child-parent relation between 50 cloudlets in static round robin algorithm.

Table 1: Relation among Cloudlets in Static Round Robin

CLOUDLET	CHILDREN	PARENT
0	No Child	11
1	23,5	10
2	No Child	25
3	46,19,17,15	13
4	7	13
5	14,9	1
6	No Child	7
7	43,37,27,6	4
8	No Child	32
9	41	5
10	1	23
11	0	25
12	No Child	35
13	4,3	23
14	28	5
15	44,36	3
16	No Child	31
17	No Child	3
18	24	26
19	45	3
20	No Child	29
21	No Child	38
22	No Child	48
23	38,13,10	1
24	No Child	18
25	11,2	46
26	18	31
27	No Child	7
28	49	14
29	20	44
30	35	36
31	26,16	47
32	8	40
33	40	47
34	42	41
35	48,12	30
36	30	15
37	No Child	7
38	21	23
39	No Child	44
40	32	33
41	34	9
42	No Child	34
43	No Child	7
44	39,29	15
45	47	19
46	25	3
47	33,31	45
48	22	35
49	No Child	28

4.3 Results using Improved Weighted Round Robin

No. of cloudlets= 50 no. of VM's= 5
Improved Weighted Round Robin algorithm is applied and the results are shown as follows.
Figure 6 depicts that five virtual machines (VM-0, 1, 2, 3, and 4) were taken as input and load has been assigned to each virtual machine using Improved Weighted Round Robin algorithm. It has also been observed that load is properly balanced between the machines as compared to the previous algorithm. This graph shows clearly, that VM-3 and VM-4 are not utilized which leads to resource saving.
Figure 7 depicts execution time of each cloudlet. Execution time is better for weighted round robin compared to static.

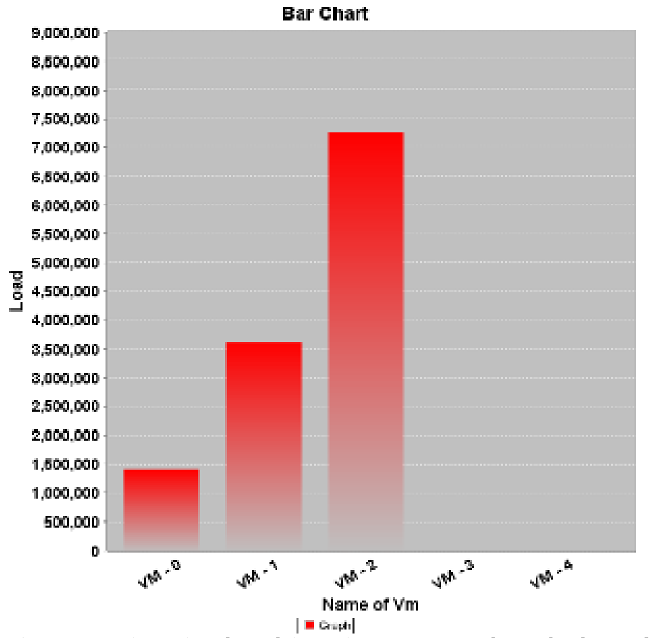

Fig. 6: VM's (x-axis) and Load (y-axis) using Improved Weighted Round Robin

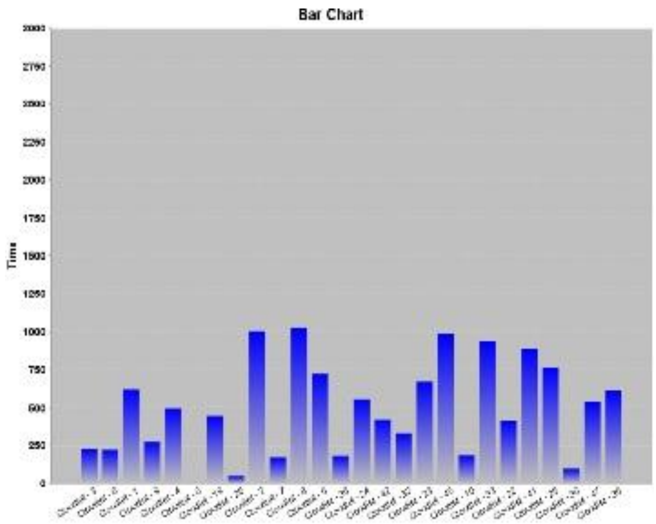

Fig.7: cloudlets (x-axis) and execution time (y-axis) using Improved Weighted Round Robin

Relation between cloudlets in Improved Weighted Round Robin algorithm:
5 VM's numbered as 0,1,2,3,4
50 Cloudlets numbered as 0,1,2,3…..49
Table 2 depicts the child-parent relation between 50 cloudlets in improved weighted round robin algorithm.

Table 2: Relation among Cloudlets in Improved Weighted Round Robin

CLOUDLET	CHILDREN	PARENT
0	36,19	2
1	40	19
2	0	26
3	26	47
4	9	24
5	No Child	47
6	No Child	42
7	No Child	15
8	No Child	22
9	No Child	4
10	33	45
11	No Child	35
12	No Child	16
13	No Child	41
14	No Child	34
15	17,7	39
16	28,12	28
17	No Child	15
18	43,20	20
19	1	0
20	18	18
21	No Child	30
22	41,37,8	33
23	45	32
24	42,4	36
25	35,30	41
26	38,2	3
27	No Child	30
28	48,16	16
29	34	43
30	47,27,21	25
31	No Child	39
32	23	42
33	22	10
34	39,14	29
35	11	25
36	44,24	0
37	No Child	22
38	No Child	26
39	46,31,15	34
40	No Child	1
41	25,13	22
42	32,6	24
43	29	18
44	No Child	36
45	10	23
46	No Child	39
47	5,3	30
48	49	28
49	No Child	48

Table 3 represents that execution time corresponding to a particular cloudlet is compared by taking some instances from the results above and proves that Improved Weighted Round Robin outperforms Static Round Robin.

Table 3: Comparison of instances of execution time

EXECUTION TIME COMPARISON		
CLOUDLET NO.	STATIC ROUND ROBIN	IMPROVED WEIGHTED RR
2	750	240
8	2150	1000
19	1400	400
23	1500	600
33	3000	1000

5. Conclusion

In this paper, we have analyzed the recent load balancing algorithms in the area of cloud computing. We have also discussed the drawbacks of existing load balancing algorithms. In this paper, we have proposed a more efficient and dynamic load balancing algorithm to improve the performance of the cloud computing technology. According to our proposed algorithm, the Improved Weighted Round Robin Load Balancing Algorithm, the cloudlets are assigned to virtual machines based on Fibonacci sequence. Using simulation results we have demonstrated that execution time of the proposed algorithm is sufficiently decreased as compared to the execution time of the static algorithm and load on each virtual machine is also reduced. Here, resources are not overloaded. Therefore, there is less load distribution on virtual machines. Hence, resources are saved because of the reason that all the virtual machines available are not utilized. There is no need to deploy new costly resources because we already have unutilized existing resources.

References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing", National Institute of Standards and Technology, Sept. 2011.

[2] Qaisar, E. J. "Introduction to cloud computing for developers: Key concepts, the players and their offerings." In Information Technology Professional Conference (TCF Pro IT), 2012 IEEE TCF (pp. 1-6). IEEE, 2013.

[3] Jadeja, Y., & Modi, K. "Cloud computing-concepts, architecture and challenges". In Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on (pp. 877-880). IEEE, 2012

[4] Avram M., "Advantages and challenges of adopting cloud computing from an enterprise perspective", in Elsevier proc. of 7th International Conference Interdisciplinary in Engineering (INTER-ENG), pp. 529-534, 2013.

[5] Seemakuthi S., "A Review on Various Scheduling Algorithms", International Journal of Scientific & Engineering Research, Vol. 6, No. 12, Pp. 769-779, 2015.

[6] Sommer M., "Predictive Load Balancing in Cloud Computing Environments Based on Ensemble Forecasting", Autonomic Computing (ICAC), IEEE International Conference, 2016.

[7] Belalem, G., Tayeb, F., & Zaoui, W. (2010). Approaches to improve the resources management in the simulator CloudSim. Information Computing and Applications, 189-196.

[8] Aslam S., "Load balancing algorithms in cloud computing: A survey of modern techniques", Software Engineering Conference (NSEC), 2015 National, February 2016

[9] Kashyap, Dharmesh, and Jaydeep Viradiya. "A survey of various load balancing algorithms in cloud computing." Int. J. Sci. Technol. Res 3, no. 11 (2014): 115-119.

[10] Dhakad K., "Performance Analysis Of Round Robin Scheduling Using Adaptive Approach Based On Smart Time Slice And Comparison With SRR", International Journal of Advances in Engineering & Technology, Vol. 3, No. 2, Pp. 333-339, May 2012.

[11] Saranya D., "Load Balancing Algorithms in Cloud Computing: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, No. 7, Pp. 1107-1111, 2015.

[12] Datta L., "Efficient Round Robin Scheduling Algorithm with Dynamic Time Slice", I.J. Education and Management Engineering, Vol. 2, Pp. 10-19, 2015

[13] Kanagaraj, Gowtham, Naveen Shanmugasundaram, and Sathish Prakash. "Adaptive Load Balancing Algorithm Using Service Queue." Jurnal. ICCSIT (2012).

[14] Zhao, Y., Chen, L., Li, Y., & Tian, W "Efficient task scheduling for Many Task Computing with resource attribute selection. China Communications, vol. 11(no. 12), pp. 125-140, 2014.

[15] S. G. Damanal and G. R. Reddy, "Optimal Load Balancing in Cloud Computing by Efficient Utilization of Virtual Machines", in IEEE proc. of 6th International Conference on Communication Systems and Networks (COMSNETS), Jan. 2014.

[16] Zuo, L., Shu, L., Dong, S., Zhu, C., & Hara, T. "A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing". IEEE Access, vol. 3, pp. 26872699, 2015.

[17] Teymoori, P., Sohraby, K., & Kim, K."A fair and efficient resource allocation scheme for multi-server distributed systems and networks". IEEE Transactions on Mobile Computing, vol. 15, no. 9, pp. 2137-2150, 2016

[18] Ariharan, V., & Manakattu, S. S. (2015, March). Neighbour Aware Random Sampling (NARS) algorithm for load balancing in Cloud computing. In Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on (pp. 1-5). IEEE.

[19] Calheiros, R. N., Ranjan, R., De Rose, C. A., & Buyya, R. (2009). Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services. arXiv preprint arXiv:0903.2525.

[20] Naseem M, Kumar C. Congestion-aware Fibonacci sequence based multipath load balancing routing protocol for MANETs. Wireless Personal Communications 2015; 84(4):2955–2974.

[21] Naseem, Mohd, and Chiranjeev Kumar. "QSLB: queue size based single path load balancing routing protocol for MANETs." International Journal of Ad Hoc and Ubiquitous Computing 24.1-2 (2017): 90-100.

[22] Naseem, Mohd, and Chiranjeev Kumar. "Queue-based multiple path load balancing routing protocol for MANETs." International Journal of Communication Systems 30, no. 6 (2017).