MSLB: Multi-level scheduling for achieving load balancing in Cloud Environment

Aliva Priyadarshini
P.G Department of Computer Science Application
Utkal University, Bhubaneswar, Odisha, India
aliva2020@gmail.com

Sateesh Kumar Pradhan
P.G Department of Computer Science Application
Utkal University, Bhubaneswar, Odisha, India
sateesh1960@gmail.com

Samaleswari Prasad Nayak*
Department of Computer Science & Engineering
Silicon Institute of Technology, Bhubaneswar, Odisha, India
samaleswari.nayak@silicon.ac.in

Suchismita Rout
School of Computer Engineering,
KIIT University, Bhubaneswar, Odisha, India
suchismita.rout28@gmail.com

Abstract:

Cloud computing offers an efficient solution to the users for the storage, computation, and many such services. With the incoming user requests, cloud system is assigned with load. This makes the system overloaded, underloaded and balanced system. The situations of overloaded and underloaded cloud system may invite different problems like power consumption, device failure, etc. So, load balancing is an essential system needs for a healthy and robust system. It becomes a significant aspect of task scheduling in cloud computing. There are various factors for considering load of the system such as memory load, network load, computation load etc. Various researches proposed different solutions to balance the load in cloud infrastructure. Through this article an optimal load balancing mechanism MSLB is proposed with solutions. A brief explanation of different parameters is also provided by comparing existing solutions with proposed one. An innovative solution to balance the load of the virtual machines is described considering various user bases. Simulation results are obtained using Cloud Analytics and the results are presented to support that.

Keywords: Load Balancing, Clod Computing, Performance metrics, Taxonomy, Scheduling

***e*ISSN**1303-5150

DOINumber:10.14704/nq.2022.20.8.NQ44828 **NeuroQuantology 2022; 20(8): 8029-8040**

1. Introduction: In the field of networking, cloud computing has achieved tremendous progress due to improvement of communication technology, rapid use of internet for accessing the resources and technical ability to solve large scale complex problems. Cloud computing is also coined as pay-as-you-go. The consumer can access all the resources (e.g. network, storage, services, server etc.) over the internet and the bill is generated as per the usage, The user is free from all the over burden of creating the infrastructure, periodically updating the software, handling the storage mechanism and many more. The cloud service provider (CSP) outsourced all the resources through the internet. The Service Level Agreement (SLA) is set between the cloud provider and cloud user. Both the provider and user adhere to the rules of this SLA. Many big Software companies such as Google, Amazon, IBM, HP, Apple, Microsoft etc. are providing cloud computing as part of utility computing. The cloud computing is providing three service models such as Infrastructure as a service (Iaas), Platform as a service (Paas), and Software as a service (Saas). It also has four deployment models such as private, public, hybrid and community [1].

The user can avail all the services at any time provided good internet connection must be there. Due to virtualization concept in cloud, the resources are seemed to be unlimited. The CSP provided all the services to users on rented basis. This service provision is done at a very complex mode. The CSP has taken into account all the available cloud resources efficiently to satisfy the incoming demand of the cloud users. The role of the CSP is even more complex with the exponential rise in the demand of the cloud user. Therefore, load balancing in cloud is a very important issue which needs to be addressed competently. Many researchers paying attention to find the efficient solution to provide a balanced system. The solution has to balance the trade-off between financial benefits and user satisfaction through load balancing [2]. This ensures the system load must be distributed in a fairly manner to provide high resource utilization and better response time.

Static and dynamic are the two versions of load balancing algorithm depending on the types of systems. Static version is more suitable for stable environment with homogeneous system. The dynamic version is more adaptable in homogeneous and heterogeneous system. The static system has fewer burdens as compared to dynamic system.

In cloud computing terminology, the task allocation to different VM is known as load. Load distribution is formulated as distribution of different task into physical machines (PM), which again assigned to different VM on respective PM. The migration of VM from one PM to another PM is done to improve the resource utilization of the data center. In case of overloaded PM, the VM associated with this PM is migrated to another PM. In case of overloaded VM, the task migration from overloaded VM to less overloaded VM is done. The task migration is an important feature in load balancing in cloud computing [3].

In this paper, a new efficient strategy to improve the load balancing in cloud computing is evaluated. We also present the cloud computing architecture to illustrate the cloud system. A taxonomy explaining the load balancing in cloud computing is present. Various performance parameters are explained to compare with different researches on load balancing in cloud computing.

The remaining of this paper is organized as follows. The cloud computing architecture is

*e*ISSN1303-5150 www.neuroquantology.com

explained in section 2. Section 3 gives idea about various performance metrics used in load balancing algorithm. The classification of load balancing algorithm is given in section 4. The proposed algorithm and its simulation behavior are illustrated in section 5 and 6 respectively. Finally, the paper is concluded in section 7.

2. Cloud Computing Architecture:

Cloud computing architecture is different from traditional architecture in term of its properties like scalability, economics controlling, and virtualization [4]. Many researchers have followed the single host architecture in cloud environment as shown figure 1. The architecture consists of three layers. The hardware layer consists of main memory, processor, network bandwidth, secondary storage. The VMM (Virtual Machine Monitor) or hypervisor to act as boundary between guest OS and VMs. Multiple operating systems can run on a single hardware platform concurrently. Each guest operating system can run different number of heterogeneous applications {VM_1, VM_2, VM_3, …, VM_n}.

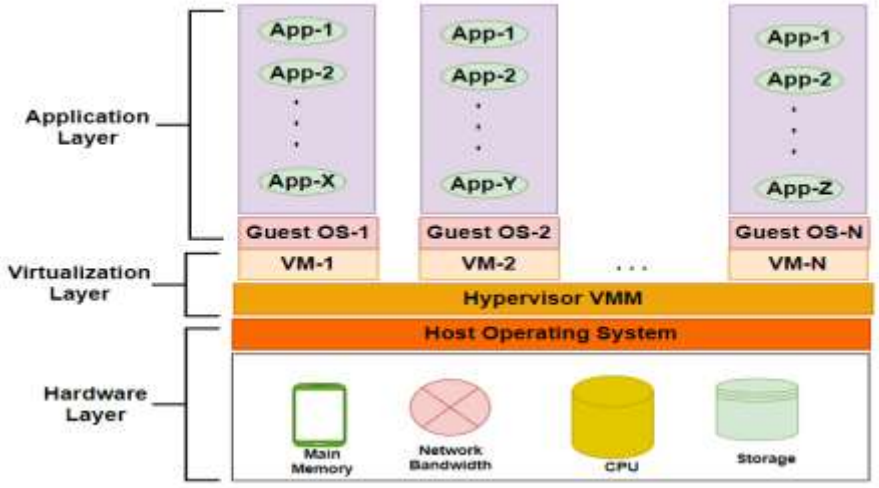


Figure 1 Single Host Architecture

A cloud data center consists of finite number of hosts. These hosts are heterogeneous in nature. Each host is recognized by host identification number, processing speed in terms Million instructions per second. Every host is assigned with number of VMs. A VM has some attributes to complete the assigned task. Each VM can complete one task at a time. The major challenge in cloud computing is to mapping the load of each user to the central load balancer or scheduler with the cloud resources. After each incoming request, the task is assigned to the load balancer. The load balancer distributes each task with the VM to complete it within the

time period, Sometimes the available VMs are not enough to complete all incoming request, in such cases the task has to wait if SLA permits. After any of the resources are free, it can be utilized to execute the pending request [5].

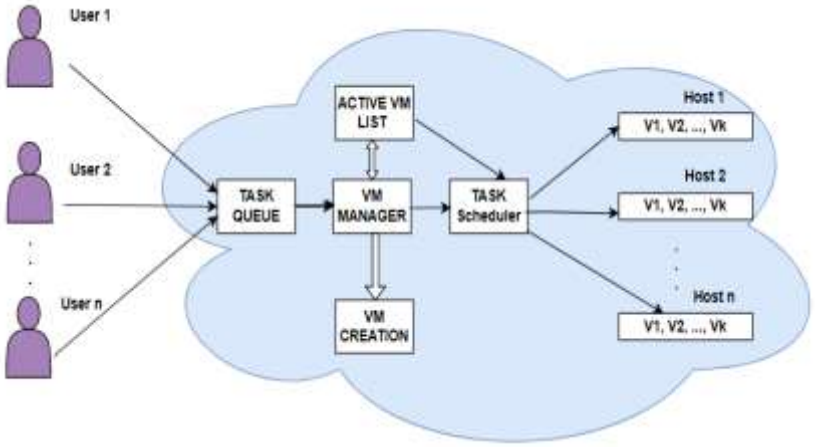


Figure 2 Scheduling Model in the cloud data centre

The scheduling model is shown in Figure 2. A large number of heterogeneous tasks are coming to the cloud data centre with various requirement. The n number of tasks are coming {T1, T2, T3, …, Tn} to the task queue. The VM manager figures out the number of active VM in the list to complete the tasks with their current available resources and task queue length of all the host. If the number of availed resources is not enough to execute the task, then the VM manager creates the new VM with active number of resources. The task scheduler allocates all the task to the VM, based on their load requirement. It does the job of load balancer to balance the incoming requests with the number of VMs [6].

3. Performance metrics used in Load Balancing Algorithm:

Load balancing is a proven solution in cloud computing to improve the system stability. While balancing the load, we are not only considering the single objective, but also, we focus on other performance metrics such as energy savings, makespan, throughput, system accuracy etc [4]. Some of these parameters are explained here.

Makespan: It is the total time taken by the system to complete all the task. It is the maximum time that the host will take to execute the task. In a balanced system, the makespan is more optimized.

Throughput: The number of user requests executed per unt time is called throughput. The system with high throughput is considered as good system. The throughput is inversely proportional to the makespan.

Thrashing: When the majority of the resources are exhausted or limited to execute the assigned number of incoming requests coming to the system, then thrashing occurs. In cloud environment, if the system is not properly balanced, then the VMs are spending more time in migration. To maintain a balanced system proper scheduler is required.

Reliability: This will make the system more stable. During the task execution, if any failure occurs, then the task must be transferred to the other VMs, to maintain stability of the system.

Accuracy: It determines how perfectly the system can compute the task. System with high

*e*ISSN1303-5150 www.neuroquantology.com

accuracy has more demand. The accuracy of a system degrades the makespan of the system.

Scalability: It will indicate how efficiently the system can handle in unexpected conditions. In a balanced system, when the number of workloads is increased, then the rescaling of the resources will occur.

Associated Overhead: The overhead cost associated with the execution of the algorithms is known as associate overhead. A balanced system is associated with less overhead.

Energy consumption: The total energy consumption is calculated based upon the devices that are connected to the cloud system. Local server, networking nodes and personal systems are the major sources of energy consumption. Different solutions are proposed such as to minimize energy consumption in hardware, efficient energy conserving algorithm, power minimization in the server, and power utilization of wired and wireless network.

Response Time: - It is the amount of time taken by a task to execute in the system. The sum of time related to waiting, transmission and service in the system. The optimal response time is able to give better makespan time.

4. Classification of load balancing algorithm:

The classification of load balancing algorithm is based on the current state of the VM. Based on this it can be dynamic or static. In the former the load information is available before the allocation. Whereas in the later the VMs are available without any load information. Based on the available resources and incoming task request, the load balancer achieved the high throughput system and user satisfaction. Resource utilization is the major role in balancing the load in cloud computing [7].

Load balancing is broadly classified as two categories as given in figure 3.

Static Strategies: These statics follow two assumptions. Those are: the initial task arrival and availability of physical resources must be known in the beginning. The resource list is updated after that. Example- MCT, MET, OLB, GA, Min-max, min-min etc.

Dynamic Strategies: This is followed by the dynamic load distribution among the physical machines during the run-time. The virtual machines are created based upon the input tasks. The dynamic based algorithms are further classified as Off-line mode and On-line mode. In Off-line mode the tasks are allocated in some predefined moments. Example- Max-min, min-min. In On-line or immediate mode, the user task after entering into a scheduler is mapped onto a computing mode. Example- MCT, MET, OLB.

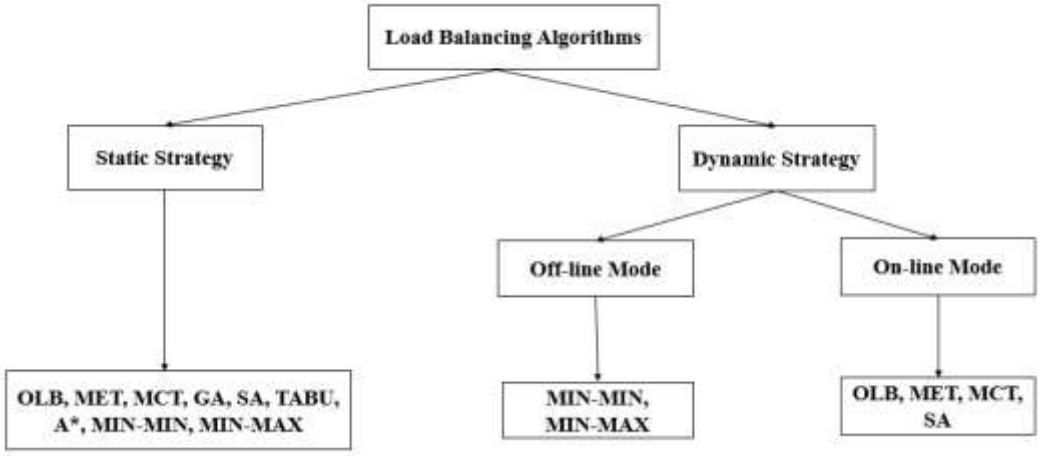


Figure 3: Classification of Load Balancing Algorithms

OLB (Opportunistic Load Balancing): It allocates the task arbitrarily and check for the next machines. In online mode task is allocated based on the execution time. In OLB scheduling algorithm the task is divided into three level sub tasks such as service manager, request manager and Service node. This will help to assign and solve the work in least time [8].

MET (Minimum Execution Time): This algorithm follows both static and dynamic strategy. Each task in the scheduler is assigned as per its minimum execution time. The advantage is that the system ca completes the task in very less period of time. The major problem with this algorithm is that it is not considering the machine ready time and shows lots of variance in the load of the system [9].

MCT (Minimum Compilation Time): It follows both static and dynamic strategy. It is preferable in the condition to balance the system in the ready-to-execute time and expected execution time. MCT will perform after allocation of task to a machine with appropriate core selection [10].

Min-Min: In the cloud environment the task with smallest size which can be executed with VM with minimum resource. After its completion the task is removed to give a chance to other unallocated tasks. To improve the load balancing the total tasks are separated into group A and B. Group A contains higher priority tasks and group B contains low priority tasks. The final schedule is prepared by the load balancing algorithm [11, 12].

Max-min: This is similar to Min-min algorithm. Max is related to the size of task and min refers to the minimum processing power of resources. After the resource is allocated, the task can be removed from the queue and resource is allocated to another task [12]. This algorithm is more suitable for small-scale distributed system.

Genetic algorithm: GA is based on the process of selection, crossover, and mutation in each iteration. The base is on the population and individual chromosomes. The number of tasks in the system is considered as the chromosome. GA based load balancing algorithm is used for lessening the makespan. The population is encoded with a binary string and chromosomes experience with a random single point crossover with a probability of 0.5 [13].

Simulated annealing: This is very useful method to solve unconstrained and bound constrained problem. In each iteration a new point is generated based on the probability distribution.

*e*ISSN1303-5150 www.neuroquantology.com

Instead, local minima it is able to provide good solutions by searching globally. It can avail resources for a number of assigned tasks [14].

Tabu search: It uses adaptive memory to perform more elastic search behavior. In [15] different TS heuristics are used for placing different cloud data centers in different locations. The major objectives are to enhance network performances, CO_2 emissions, and resource utilization cost.

A-star Search: This algorithm is applicable in Graphic searching algorithm. By combining the gain of both Depth-first search and Breadth-first search the result is A star search. It has two lists. The first one refers to the priority of the tasks and second one for processing capacity of all VMs. In [16], the network lifetime is enhanced by combining the fuzzy method and A-star method.

Switching Algorithm: It used for migration of tasks in cloud environment. The fault tolerant property is achieved using this property. Authors Shao et.al.[17] have proposed a method for switching the task by balancing the load of the network.

5. Proposed Model:

In cloud computing environment, the random arrival of tasks with random utilization of CPU can load specific resources heavily, while some are less loaded. Hence load balancing is a major challenging issue in cloud computing while distributing the wok load. It is widely accepted that load balancing is to distribute workload across multiple computers or other resources over the network links to achieve optimal resource utilization, maximize throughput, minimum response time, and avoid overload. The objective of the proposed model MSLB is same as balancing of workloads among clouds. We have used the concept of scheduling for performance enhancement, at the same time not only the tasks are scheduled but with a new thought of scheduling the virtual machines.

Initially the incoming tasks are submitted to a list and are sorted on the basis of their priorities. It has been categorized into three different categories w.r.to their priorities as high, low and medium. Using this process all the jobs are assigned to different queues for parallel execution to make the model faster. The jobs are sorted and assigned to their appropriate queue. In addition, calculated each queue execution time at a complete by sorting the queue from low to high.

The virtual machine is a software implementation of computer that operating systems and applications can run on. During the next phase all the virtual machines are also schedules by using their remaining task. All the virtual machines were sorted using their capacity from least to most. The queues are also assigned with virtual machines for implementation of multithreading. The Capacity of virtual machines are divided into low, high, medium as previous step.

Assigning of tasks to different virtual machines is a crucial part for load balancing in cloud computing. To address the problem, we have proposed a model representing the procedure of easier assignment of tasks to the VMs. Each queue of job is assigned to each queue of VM prior to the measures to priorities and loads. After completion of all preliminary assignments the execution of virtual machine takes place by balancing the loads effectively.

Algorithm-1 (Initialization of Virtual Machines)

1. Initialize the threshold value of each VM
2. for i =1 to n

a. L = L + Load(VM[i])
b. P = P + LoadCapacity[i]
c. c = c + Capacity[i]

3. T_{VM} = Avg(L*p*c)
4. for i=1 to n

a. if Load[i] < T_{VM}

i. Status[i] = “Underload”

b. else if Load[i] > T_{VM}

eISSN1303-5150

i. Status[i] = "Overload"
c. else
i. Status[i] = "Balanced"
5. return

Through this proposed algorithm Virtual Machine scheduling is targeted. Three different types of conditions are assumed for separations of load like underload, overload, and balanced. Here load is considered as the key performance indicator of the tasks. The average execution time of all VMS can define by load, load capacity and capacity of VMs. But all these calculations are done based on the threshold value of each VMs. If load is less than the average load then underload and if load is greater than average then overload else assumed as balanced VM.

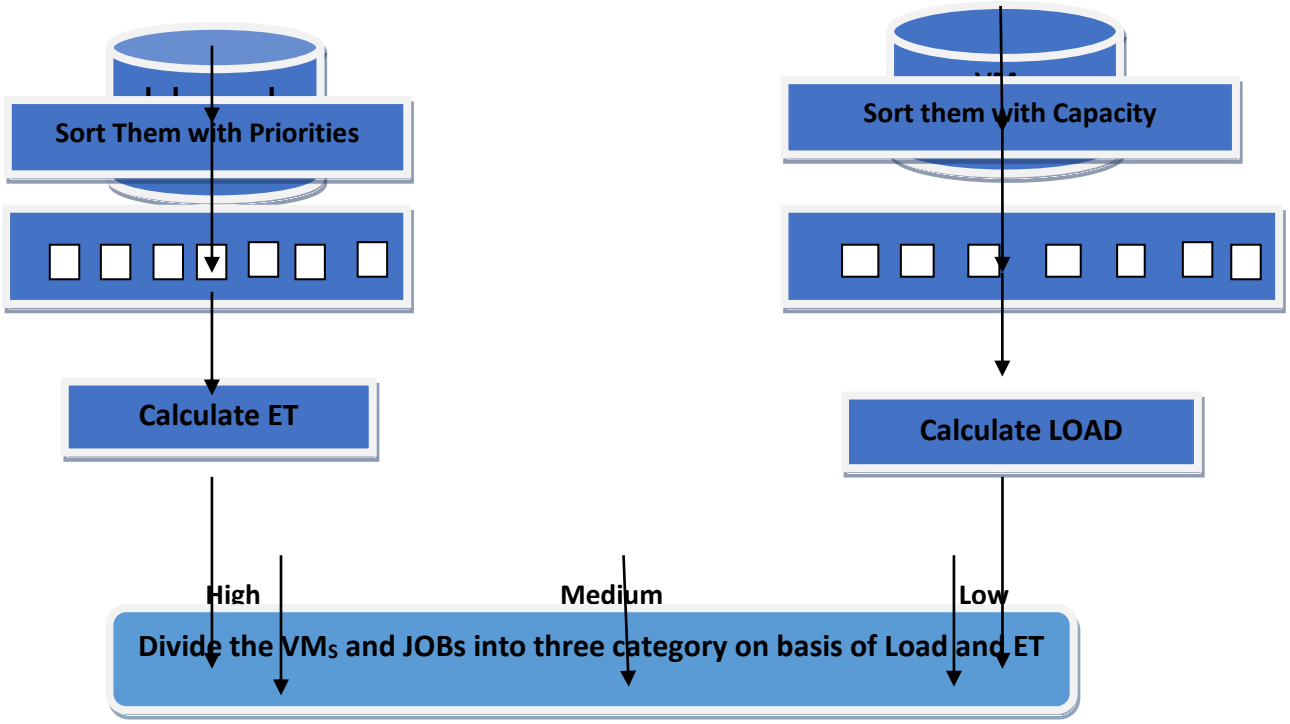
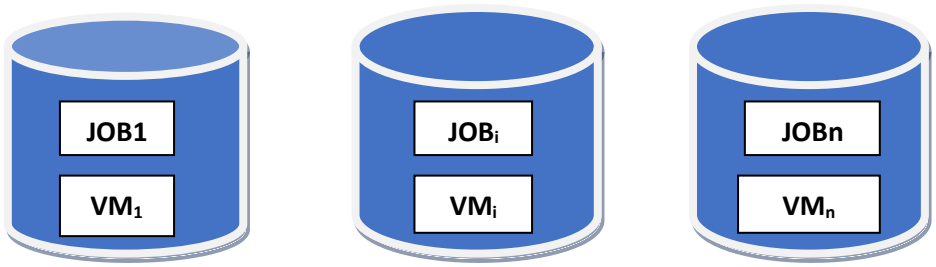


Figure 3: Proposed working model of MSLB



eISSN1303-5150

Algorithm-2 (Computation of Virtual Machines)

1. for i = 1 to n
a. $TJ[i] = N_{EJ} + N_{WJ} + N_{PJ}$
b. PT[i] = TJ[i] – TA[i]
2. for i =1 to n-1
a. for j = i+1 to n
b. if PT[j] > PT[j+1]
i. JSL[k] = PT[j]
ii. K = k+1
c. else if PT[j] = = PT[j+1]
i. if PJ[j] < PJ[j+1]
1. JSL[k] = PT[j]
2. k = k+1
3. Find the virtual machine ratio based on job and number of VMs
4. For i =1 to n
a. VMR[i] = TJ[i] / totalVM
5. Read request from User
6. Allocate the request on the basis of JSL status
7. Return

To assign appropriate VMs to respective tasks, first step is to accept the request from user base. Sequentially mapping of request from job submitted list and their execution time is done. Here a ratio of job and number of VMs are calculated and assignments of jobs are carried out based upon that calculated value. If any job can't be assigned to the VM, then next scheduled VM will be assigned automatically.

6. Experimental Results:

To justify the correctness of the proposed model the model has been compared with existing solutions. Different user bases have been selected and the load balancing mechanisms are run using cloud analytics tool. The proposed model MSLB has been compared with weighted round robin (WRR) and round robin (RR) with the same user base and same load on cloud platform. The effectiveness of MSLB is found to be better than the rest two methods. The performance of MSLB has increased around 8% compared to WRR and 9% than RR related to the complete execution time. Similarly considering the average response time MSLB provides better result than WRR and RR by 7%. The user bases are selected from different geographical locations to satisfy the uniqueness of the model. The response of the model is also satisfactory while increasing the load size compared to other existing models.

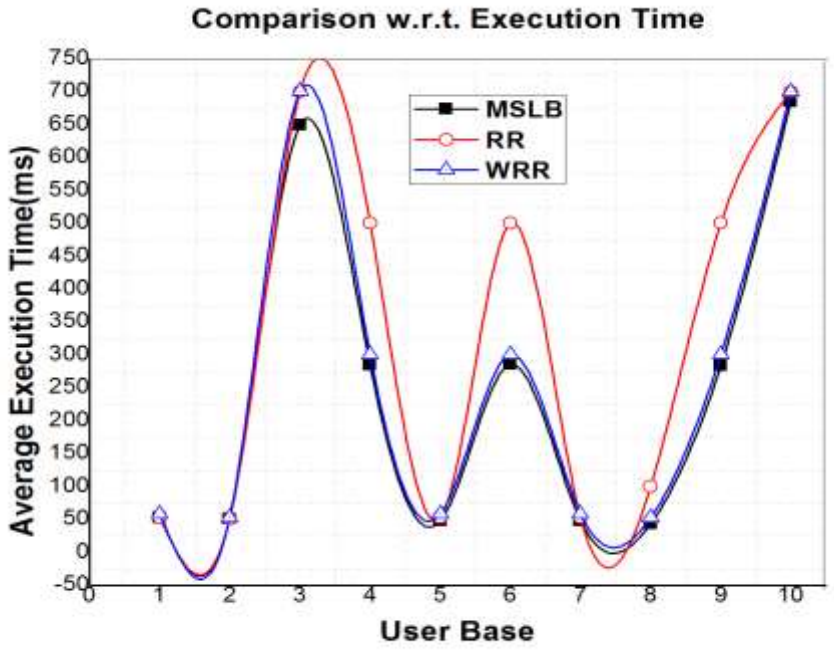


Figure 5: Comparison between different models w.r.to average execution time

7. Conclusions:

Working in the equation of "Pay-as-you-go", cloud computing is a proven solution for giving many computing and storage services to its distance customers. Load balancing is the major issue in scheduling all incoming load to the network. Many researchers provided efficient solutions to overcome this problem. Through this article we have provided the taxonomy of such solutions provided by many researchers. Along with that we have also proposed the system architecture of cloud infrastructure through MSLB. Better solution has been proposed through MSLB model to balance the load in the cloud infrastructure by considering the performance metrics. The simulation result of the MSLB model is proved to be better than the existing approaches.

*e*ISSN1303-5150

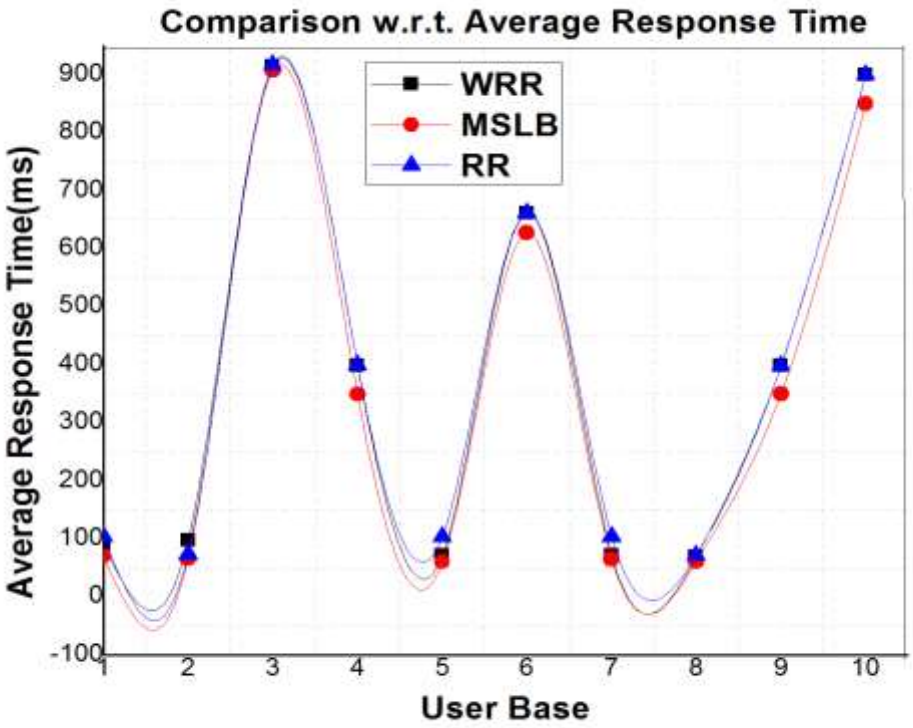


Figure 6: Comparison between different models w.r.to average response time

References:

[1] Marinescu, Dan C. *Cloud computing: theory and practice*. Morgan Kaufmann, 2022.

[2] Afzal, Shahbaz, and G. Kavitha. "Load balancing in cloud computing–A hierarchical taxonomical classification." *Journal of Cloud Computing* 8.1 (2019): 1-24.

[3] Sefati, SeyedSalar, Maryamsadat Mousavinasab, and Roya Zareh Farkhady. "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation." *The Journal of Supercomputing* 78.1 (2022): 18-42.

[4] Mishra, Sambit Kumar, Bibhudatta Sahoo, and Priti Paramita Parida. "Load balancing in cloud computing: a big picture." *Journal of King Saud University-Computer and Information Sciences* 32.2 (2020): 149-158.

[5] Ullah, Arif, and Nazri Mohd Nawi. "Enhancing the dynamic load balancing technique for cloud computing using HBATAABC algorithm." *International Journal of Modeling, Simulation, and Scientific Computing* 11.05 (2020): 2050041.

[6] Pradhan, Arabinda, and Sukant Kishoro Bisoy. "A novel load balancing technique for cloud computing platform based on PSO." *Journal of King Saud University-Computer and Information Sciences* (2020).

[7] Ghomi, Einollah Jafarnejad, Amir Masoud Rahmani, and Nooruldeen Nasih Qader. "Load-balancing algorithms in cloud computing: A survey." *Journal of Network and Computer Applications* 88 (2017): 50-71.

[8] Rewehel, Ekram M., Mostafa-Sami M. Mostafa, and Mohamed Osman Ragaie. "New subtask load balancing algorithm based on olb and lbmm scheduling algorithms in cloud." *Proceedings of the 2014 International Conference on Computer Network and Information Science*. 2014, pp-9-14.

[9] Armstrong, R., Hensgen, D., & Kidd, T. (1998, March). The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions. In *Proceedings Seventh Heterogeneous Computing Workshop (HCW'98)* (pp. 79-87). IEEE.

[10] Kim, S. I., Kim, H. T., Kang, G. S., & Kim, J. K. (2013, June). Using DVFS and task scheduling algorithms for a hard real-time heterogeneous multicore processor environment. In *Proceedings of the 2013 workshop on Energy efficient high performance parallel and distributed computing* (pp. 23-30).

[11] Chen, H., Wang, F., Helian, N., & Akanmu, G. (2013, February). User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. In *2013 national conference on parallel computing technologies (PARCOMPTECH)* (pp. 1-8). IEEE.

[12] Kokilavani, T., & Amalarethinam, D. G. (2011). Load balanced min-min algorithm for static meta-task scheduling in grid computing. *International Journal of Computer Applications*, *20*(2), 43-49. [13] A genetic algorithm (ga) based load balancing strategy for cloud computing

[14] Henderson, D., Jacobson, S. H., & Johnson, A. W. (2003). The theory and practice of simulated annealing. In *Handbook of metaheuristics* (pp. 287-319). Springer, Boston, MA.

[15] Larumbe, F., & Sanso, B. (2013). A tabu search algorithm for the location of data centers and software components in green cloud computing networks. *IEEE Transactions on cloud computing*, *1*(1), 22-35.

[16] AlShawi, I. S., Yan, L., Pan, W., & Luo, B. (2012). Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm. *IEEE Sensors journal*, *12*(10), 3010-3018.

[17] Shao, S., Guo, S., Qiu, X., & Meng, L. (2014, August). A random switching traffic scheduling algorithm in wireless smart grid communication network. In *2014 23rd International Conference on Computer Communication and Networks*

***e*ISSN**1303-5150