IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

COMPARATIVE ANALYSIS OF VARIOUS SCHEDULING ALGORITHM IN CLOUD COMPUTING

[1]Priyal Ghetiya [2] Dhaval Nimavat

[1]Student of Master of Technology (M.Tech), Computer Engineering, R.K.University, Rajkot, India
[2]Assistant Professor, Faculty of Computer Science, R.K.University, Rajkot, India
R.K. University, Rajkot, India

Abstract: Today, cloud computing is an established technology that allows businesses and individuals to store data easily, retrieve it quickly, and access it from any location. Currently, both industry and academia are adopting this technology. Load balancing is crucial for ensuring the efficient operation of different components in the cloud computing environment. When users request specific resources from the cloud, the system supplies them. An effective load balancing strategy is crucial for preventing server overload and ensuring that requested resources are delivered as soon as possible. In this research article, we have presented a critical analysis of various existing cloud load balancing and scheduling algorithms. Different simulation tools are available in the market for simulating cloud load balancing and scheduling algorithms. Additionally, the article discusses different simulation methods and technologies employed by different algorithms. Overall, the research article explains how different scheduling and load balancing algorithms can help in resolving the problem of load distribution in cloud computing systems.

Index Terms **– Scheduling, Load Balancing, Virtual Machine, Cloud Computing, Cloud Analyst, CloudSim.**

I. INTRODUCTION

During the last several decades, rapid improvements in computing power, storage, and networking technologies opens up many new opportunities for businesses and organizations to grow more and more and have allowed organizations or individuals to generate necessary information, process, and share a large amount of information using the internet in dramatically new ways [1]. As a result, there is a need for powerful computing applications to arise to effectively handle user requests and to ensure the smooth flow of the system which in turn creates the demand for even more effective and powerful computing infrastructures as compared to previously available infrastructure used for conventional systems.

To compete with this infrastructure demand, researchers and system designers are constantly looking for new applications, algorithms, and computing infrastructures to process large collections of data and react to user requests rapidly. System designers/researchers make this feasible with today's computing system. It is now possible to assemble a large system by integrating many small inexpensive, powerful components networked together to provide less expensive storage and computing infrastructure. Such a computing system tends to be less expensive than that of a single, faster machine with the same capabilities.

To meet users' demand for specific resources proper load balancing strategy must be needed to be implemented in a cloud system to ensure that each user request is efficiently allocated to virtual machines and that leads to minimum load on the cloud system. Various scheduling and load balancing algorithms have been proposed to overcome resource allocation problems in cloud infrastructure. Depending on network requirements and by considering scalability factors of cloud systems appropriate load balancing strategy is chosen to overcome load distribution problems in cloud computing systems.

In a cloud computing environment load balancing is defined as distributing a load of system to individual nodes (Virtual machines) of the cloud computing system to achieve higher throughput and enhance the response time of the cloud system.

Depending on the load distribution scheme load balancing algorithms are classified into two different classes: 1) Static load balancing algorithms: Requires priory knowledge of resources or applications. Static load balancing algorithm doesn't consider current load or current allocation to VM while loading distribution. 2) Dynamic load balancing algorithms: This doesn't require prior knowledge of the system. While load distribution dynamic load balancer considers the current load on the VMs.

The load balancer in cloud computing is an active component that is associated with the task of shifting load to the various processing elements of VMs. The figure shows the working of the load balancer. Load balancer maps the incoming requests to the backend cloud server. The cloud server is a back-end processing element of cloud computing that accept user requests performs operations and replies to the user request.

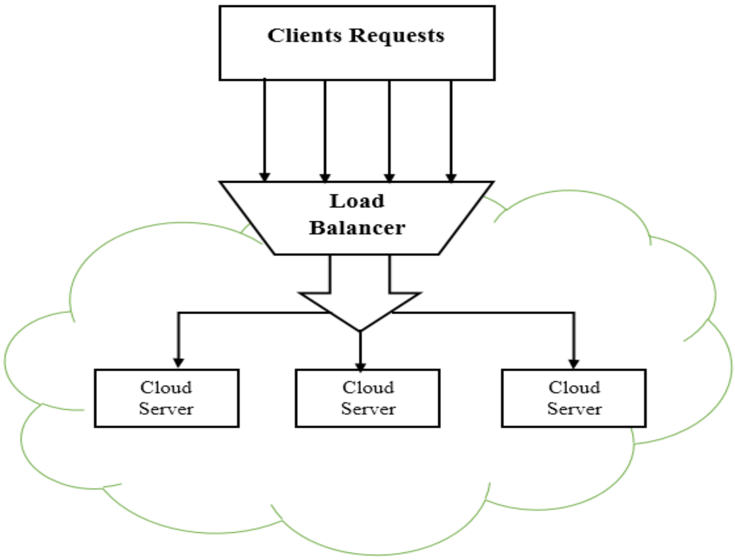

Figure 1.1: Cloud load balancer

II. RESEARCH BACKGROUND

(Rai, 2020) presented a cloud load balancing algorithm based on VM MIPS and the processing capacity of PEs. The algorithm takes scheduling decisions based on virtual machine MIPS and bandwidth. The author introduced virtual parameters such as Virtual Machine Capacity and Virtual Total Load. VM capacity is compared with a dynamic threshold and scheduling decisions will be taken [1].

(Kumar, 2018) proposed adaptive load balancing algorithm which is the modified version of the multi-time load balancing algorithm [2]. The major objective of the research work is to maximize throughput and minimize the response time of the cloud computing system. The algorithm is implemented using the Amazon EC2 instances on the Linux platform with the help of a shell script. The author also presented steps required to execute the load balancer algorithm using Amazon EC2. Algorithm execution results show that the proposed algorithm gives 25% less response time compared to the existing round-robin algorithm. Also, the data center processing time is 22.35% lesser than that of the round-robin algorithm.

(Narale, 2018) presented detailed performance analysis of the throttled load balancing algorithm. The throttled load balancing algorithm comes inbuilt with the CloudAnalyst package. CloudAnalyst is a cloud simulation tool based on the CloudSim library. The tool provides a GUI interface to users that make the configuration of cloud computing components easier. The author used 8 different simulation configurations to test the performance of the throttled load balancing algorithm. The author presented performance analysis of throttled load balancing algorithm by considering various parameters such as datacenter processing time, throughput, data transfer cost, and total cost [3].

(Mishra, 2020) proposed a priority-aware load balancing algorithm. The algorithm is implemented using the cloudAnalyst tool. Depending on the task's priority credit score will be calculated and a request will be allocated to the available VM. Simulation results show that the proposed system gives a better response time than that of the existing round-robin scheduler [4].

(Joshi, 2018) presented threshold-based load balancing algorithm. The algorithm is implemented using the OpenStack cloud platform on CentOS Linux 7 operating system. Virtual machine overloading and underloading are determined using the static threshold value. The execution result shows that the presented algorithm solves the VM underloading and overloading problem and ensures even request distribution among all available virtual machines [5].

(Malhotra, 2018) presented dynamic load balancing algorithm that considers load on VMs while allocating user requests. The load on VMs is calculated using RAM and MIPS requirements. The author used the cloudsim tool to simulate the load balancing algorithm. The research article presents the analysis of load on VM and execution time after simulation operation. The author used three different simulation configurations to simulate the proposed algorithm: 5VM, 10 VM, and 20 VM [6].

(Sahu, 2019) presented a novel scheduling algorithm that reduces the energy consumption of the cloud components. The author also presented working of different scheduling and load balancing algorithms such as throttled load balancing, Active monitoring load balancer, and round-robin load balancer. Performance evaluation of the proposed algorithm is performed using the cloudSim simulation tool. Energy consumptions of five different scheduling algorithm (Max-min, Min-min, min-max, MCT, and MET) is presented in this research article[8].

The followings are some of the major task scheduling algorithms.

First come first serve (FCFS): The FCFS is a very basic task scheduling algorithm that schedules users' tasks depending on their arrival sequence. The task which arrives first will be scheduled first. So basically, the first task will receive cloud resources first and then after the second task and so on. The disadvantage with this approach is that sometimes priority and large needs to wait for a long duration [9]

Round Robin: The round-robin algorithm is like that of the FCFS algorithm. The difference only lies in their allocation pattern. The RR algorithm allocates user requests to VM based on their arrival sequence but for the next request allocation, the schedule starts searching for available VM from the next index to that of the previously allocated VM. Whereas the FCFS algorithm always starts searching from the first index [9].

.

Max-min algorithm: The algorithm computes expected completion time and expected execution time to allocate cloud resources. Based on both of these parameters, the user request will be allocated to VM having maximum completion time first. The intention here is to allocate resources to larger tasks first. So that larger tasks can be completed earlier, and smaller tasks can be scheduled letters [10].

Min-min algorithm: The algorithm uses the concept of a minimum completion time algorithm. The algorithm allocates user requests to VM having minimum expected completion time. The algorithm is a static scheduling algorithm. Computing expected completion time for every task require information on task characteristics. This algorithm is already implemented and integrated into the Workflowsim simulation tool. Users can check the performance of this algorithm using different input files [10].

Minimum completion time algorithm: The algorithm is similar to that of the Max-min algorithm and Min-min algorithm. The algorithm employs the approach of expected minimum completion time to allocate user requests to cloud resources [11].

SJF algorithm: As the name suggests itself, the algorithm allocates user requests to VM based on their task length. Task length can be computed in MI (Millions of Instructions). Task having smaller task lengths will be scheduled earlier. This algorithm gives more importance to shorter tasks [11].

LJF Algorithm: The algorithm is like that of the SJF algorithm. Here the algorithm allocates requests to longer tasks first. In both algorithm SJF and LJF, details about task length are required before execution. So, both of these algorithms are static task scheduling algorithms. The algorithm gives more importance to larger tasks [12].

Hybrid SJF-LJF Algorithm: It is a combination of SJF and LJF algorithms. The algorithm first sorts all tasks based on the SJF concept. After sorting the algorithm selects tasks alternatively from the first and last index. Already selected tasks will be removed from the list. With this concept, importance is given to both short and long tasks. Although the algorithm requires details of task length before execution [12].

III. Comparative Analysis

After conducting detailed research on existing scheduling and load balancing algorithms comparative analysis is presented in the table below. The table presents various tools and parameters considered for performance evaluation with advantages and challenges faced by different scheduling policies.

Table 3.1: Comparative analysis of existing scheduling algorithms

Article	Tools / Technologies	Parameters	Pros	Cons
(Rai, 2020)	CloudSim	Start time and finish time, makespan	Request allocation is based on VM MIPS and the processing capacities of PEs.	A comparison of VM load share is not presented. Very few user requests are used for performance testing (10 only).
(Kumar, 2018)	Amazon EC2	Response time, throughput, standard deviation	Real-time cloud load balancing implementation. Faster than the existing round-robin scheduler. Large input request dataset – 100000 requests.	Only two VMs are used.
(Narale, 2018)	CloudAnalyst	Datacenter processing time, data transfer cost, and total cost.	8 different cases are used to test the performance of the algorithm.	Characteristics of VMs and userbase are not defined in the research article.
(Mishra, 2020)	CloudAnalyst	Execution time	All simulation configuration parameters are presented. All evaluation parameters are clearly defined.	Other parameters from cloudAnalyst simulation results are ignored such as cost, data center processing time. Ambiguity in the

				presentation of experimental setup and simulation.
(Joshi, 2018)	Openstack on Sent OC Linux 7	Load on VM	Real-time load balancing is performed. The system overcomes the overloading and underloading of VMs.	A static threshold value is used to check the underloading and overloading of VMs.
(Malhotra, 2018)	CloudSim	Load on VM	Consider load on VMs while allocating user requests. Load is measured using RAM and MIPS requirements.	Bandwidth and CPU utilizations are not considered while calculating VM load. Simulation results are not compared with any existing algorithms.
(Tang, 2018)	CloudSim	Resource utilization and traffic volume	Compared results with existing min-min scheduling algorithm.	Only 100 tasks were used for performance evaluation.
(Sahu, 2019)	CloudSim	Energy consumption and Execution time	Comparisons with 5 different algorithms are done.	Simulation parameters are not defined in the research article.
(Tyagi, 2019)	CloudAnalyst	Response time, processing time, cost	Evaluation based on multiple parameters is done.	Performance analysis and actual simulation results were not presented.
(Alworafi, 2019)	CloudSim	Makespan, throughput, Response time, Execution time	Multiple comparison parameters are considered. Analysis of four different algorithms is presented: HSLJF, SJF, LJF, and Round Robin	Not considering processing capabilities of VM. Not considering the load on VMs.
(Swarnakar, 2020)	CloudAnalyst	Response time, Makespan	Simulation characteristics were presented clearly. Snapshots of userbase and datacenter configuration were added.	CloudAnalyst presents a detailed simulation report. Not all simulation parameters were presented.
(Mishra, 2020)	CloudAnalyst	Response time, Execution time	Implementation and setup instructions were presented. Snapshots of userbase and datacenter configuration were added.	CloudAnalyst output report not added.
(Zedan, 2021)	CloudAnalyst	Response time	Implementation is already available in the cloudAnalyst tool. Simple algorithm and consider the load on VMs.	The processing capabilities of VMs and hosts are not considered.
(Singh, 2018)	CloudAnalyst	Response time	The algorithm is simple and easy to understand. Userbase specifications were given.	Other parameters can also be considered such as processing time and cost.
(Mondal, 2021)	No simulation tool used	No parameter	Double threshold value-based approach. Policies for static and dynamic load balancing were presented.	No simulation and performance evaluation is done.
(Seth, 2019)	Cloudsim	Makespan	Dynamic scheduling algorithm. Cloudlet and simulation configurations are presented.	Comparison with other parameters can also be presented.
(Zhang, 2018)	CloudSim	Response time, Cost	Both dependent and independent tasks were considered. Different comparisons were presented.	Simulation characteristics not presented.

IV. Performance Evaluation

It is hard to test the performance of this application using real testbeds because some of the elements cannot be controlled and predicted by system designers/developers. Simulators like WorkflowSim, CloudSim, cloudAnalyst, etc. are used for controlling the performance of this application by providing repeatability to experiment cloud resource allocation. The performance evaluation of all these algorithms can be carried out in the future with any of the cloud simulation tools. TABLE I shows the characteristics of the resource used. There is one data center that contains 25 hosts in it. These configurations can be considered ideal to measure the performance of the scheduling algorithm [12].

Table 4.1: Simulation Characteristics

Datacenter	
Number of Datacenters	1
Number of hosts	25
VM (Virtual Machine)	
Number of VMs	100
MIPS of PE per VM	1000-3000 MIPS
VM memory	512
Bandwidth	1000
Type of manager	Space shared

V. Future Work

The research article presents a comparative analysis of various existing scheduling and load balancing algorithms. Many cloud simulation tools are available in the market that enables users to simulate real-world cloud scenarios such as energy-efficient scheduling, load balancing, virtual machine migration, etc. In the future analysis of various cloud simulation tools can be done to present the most suitable tool for various simulation applications. Advantages and disadvantages of these tools can be presented that helps researchers in sleeting the most appropriate tool for their research and development work. Some GUI tools such as CloudAnalyst, Cloud Report, etc. are designed with separate GUI interface that makes it easier for researchers to create the specific experimental configuration. While some simulation tools don't have any GUI interface which makes it difficult for researchers to create experimental configuration as it is required to interact with the programming interface to change configuration every time. So, working on all such tools and how to perform simulation using these tools can also be presented as future work.

VI. Conclusion

An increase in the number of users in the system raised many challenges to cloud service providers. Scheduling all user requests at a rapid rate and ensuring service level agreement and QoS availability becomes the most challenging task. Scheduling is not only about mapping user requests to resources but also comprises managing resource utilization, energy consumption, load balancing, data center processing, and many more. In this research article, a comparative analysis of various scheduling and load balancing algorithms is presented. Multiple simulation tools can be used to simulate real-world cloud computing applications. The research article also highlights simulation tools used with the pros and cons of each scheduling approach. Also working of all scheduling algorithms is presented here with parameters considered for simulation and performance evaluation. Scheduling algorithms not only schedule resources but also help in minimizing energy consumption by providing efficient distribution of incoming tasks to available virtual machines.

References

[1] Rai, H., Ojha, S. K., & Nazarov, A. (2020, December). Cloud Load Balancing Algorithm. In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 861-865). IEEE.

[2] Kumar, P., Bundele, M., & Somwansi, D. (2018, November). An adaptive approach for load balancing in cloud computing using MTB load balancing. In 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE) (pp. 1-5). IEEE.

[3] Narale, S. A., & Butey, P. K. (2018, April). Throttled load balancing scheduling policy assist to reduce grand total cost and data center processing time in cloud environment using cloud analyst. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 1464-1467). IEEE.

[4] Mishra, A., & Tiwari, D. (2020, December). A Proficient Load Balancing Using Priority Algorithm In Cloud Computing. In 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT) (pp. 1-6). IEEE..

[5] Joshi, N., Kotecha, K., Choksi, D. B., & Pandya, S. (2018, January). Implementation of Novel Load Balancing Technique in Cloud Computing Environmen. In 2018 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5). IEEE.

[6] Malhotra, D. (2018, December). LD_ASG: Load Balancing Algorithm in Cloud Computing. In 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 387-392). IEEE.

[7] Tang, L. (2018, August). Load Balancing Optimization in Cloud Computing Based on Task Scheduling. In 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS) (pp. 116-120). IEEE.

[8]. Sahu, S., & Pandey, M. (2019, July). Efficient load Balancing algorithm analysis in Cloud Computing. In 2019 International Conference on Communication and Electronics Systems (ICCES) (pp. 779-783). IEEE.

[9] Agarwal, M., & Srivastava, G. M. S. (2016, April). A genetic algorithm inspired task scheduling in cloud computing. In 2016 International Conference on Computing, Communication and Automation (ICCCA) (pp. 364-367). IEEE.

[10] Mao, Y., Chen, X., & Li, X. (2014). Max–min task scheduling algorithm for load balance in cloud computing. In Proceedings of International Conference on Computer Science and Information Technology (pp. 457-465). Springer, New Delhi.

[11] Mehdi, N. A., Mamat, A., Amer, A., & Abdul-Mehdi, Z. T. (2011, December). Minimum completion time for power-aware scheduling in cloud computing. In 2011 Developments in E-systems Engineering (pp. 484-489). IEEE..

[12] Tyagi, N., Rana, A., & Kansal, V. (2019, February). Creating elasticity with enhanced weighted optimization load balancing algorithm in cloud computing. In 2019 Amity International Conference on Artificial Intelligence (AICAI) (pp. 600-604). IEEE.

[13], Swarnakar, S., Kumar, R., Krishn, S., & Banerjee, C. (2020, September). Improved Dynamic Load Balancing Approach in Cloud Computing. In 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE) (pp. 195-199). IEEE.

[14] Mishra, A., & Tiwari, D. (2020, December). A Proficient Load Balancing Using Priority Algorithm In Cloud Computing. In 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT) (pp. 1-6). IEEE.

[15] Alworafi, M. A., Dhari, A., El-Booz, S. A., Nasr, A. A., Arpitha, A., & Mallappa, S. (2019). An enhanced task scheduling in cloud computing based on hybrid approach. In Data Analytics and Learning (pp. 11-25). Springer, Singapore.

[16] Zedan, M., Attiya, G., & El-Fishawy, N. (2021, July). Load balancing Based Active Monitoring Load Balancer In Cloud Computing. In 2021 International Conference on Electronic Engineering (ICEEM) (pp. 1-6). IEEE.

[17] Singh, A. N., & Prakash, S. (2018). WAMLB: weighted active monitoring load balancing in cloud computing. In Big data analytics (pp. 677-685). Springer, Singapore..

[18] Mondal, A. S., Mukhopadhyay, S., Mondal, K. C., & Chattopadhyay, S. (2021). A Double Threshold-Based Power-Aware Honey Bee Cloud Load Balancing Algorithm. SN Computer Science, 2(5), 1-16.

[19] Seth, S., & Singh, N. (2019). Dynamic heterogeneous shortest job first (DHSJF): a task scheduling approach for heterogeneous cloud computing systems. International Journal of Information Technology, 11(4), 653-657.

[20] Yadav, M., & Prasad, J. S. (2019, April). An Enhanced Genetic Virtual Machine Load Balancing Algorithm for Data Center. In International Conference on Advances in Computing and Data Sciences (pp. 244-253). Springer, Singapore.

[21] Mulla, B., Krishna, C. R., & Tickoo, R. K. (2019, August). Virtual Machine Allocation in Heterogeneous Cloud for Load Balancing Based on Virtual Machine Classification. In International Conference on Inventive Computation Technologies (pp. 331-341). Springer, Cham.

[22] Zhang, Y., Cheng, X., Chen, L., & Shen, H. (2018). Energy-efficient tasks scheduling heuristics with multi-constraints in virtualized clouds. Journal of Grid Computing, 16(3), 459-475.