JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ANALYSIS OF LOAD BALANCING ALGORITHMS IN THE CLOUD COMPUTING ENVIRONMENT: A REVIEW

Santosh Kumar Maurya[1] , Dr. Garmia sinha[2]
[1]Research Scholar IIMT University Meerut (India) santoshcse200@gmail.com
[2]Professor Department of Computer Science IIMT University Meerut (India) mailatgarima@yahoo.co.in

Abstract:

Cloud computing is a cutting-edge worldview that allows for the provision of many forms of assistance over the Internet. Task scheduling and load balancing is an important and vital component of cloud computing, and load imbalance refers to the situation in which some nodes get overloaded while the remaining nodes are idle or have just minor work to perform. By attempting to enhance the Quality of Service (QoS) measurement metrics, such as the minimum makespan, load balancing is attempting to improve the usage of cloud resources. Cloud computing infrastructure scheduling involves a number of difficult challenges such as computing time, budgeting, load balancing and other factors. Load balancing is one of the most difficult and time-consuming concerns for Cloud platforms to address. Load balancing techniques are used primarily to improve resource usage and throughput by distributing workload over multiple servers. In this study, we evaluate the existing methods for job scheduling and load-balancing algorithms, as well as provide information and insight into the identification of unresolved concerns and guidelines for future research, as well as recommendations for further research.

Keywords: Cloud computing, load balancing, and task scheduling, virtual machine, makespan time

I. INTRODUCTION

Cloud computing is an on-demand services in the computer field to provide the computing environment through the Internet to the clients at any time and on the utility basis. Cloud computing is the extended concept of distributed and parallel computing. Scalability, flexibility, on-demand network access and virtualization are the main features of the cloud environment and all these features are available through Internet. Virtualization is a key concept in cloud computing; it provides a huge execution/computing environment to the cloud users. IaaS(Infrastructure as a service) , PaaS (platform as a service), and SaaS (software as a service) are three main types of basic services provided by the cloud service providers. Users of the cloud uses the services in utility form (i.e. pay per use basis) and cloud providers offer service level agreement (SLA) and satisfactory level of services to the users. The Cloud computing environment has suffered from many issues, including service availability, security, proper load balancing, resource scheduling, data center energy consumption, and scaling. In the cloud environment balancing the load on the different virtual machines is an important concern. Function of the task scheduler has to assign job/task to virtual machine from available pool of resources and load balancer continuously observing load of each virtual machine, if any virtual machine are under loaded or overloaded then it apply load balancing algorithm to balance the load on each virtual machine and in order to minimize makespan time and increase throughput. The purpose of the load balancing algorithms is to use computer resources properly. So that all virtual machine/nodes can be used as much as possible. It is not that some resources are being used very much and some are not being used. In a cloud computing environment, arrived job/task consists of many interdependent tasks/job and they may run the independent task/job in multiple node/VMs or in the same node/VMs multiple cores. The available heterogeneous computing VMs are handled by assigning the task to appropriate VMs by static or dynamic scheduling to make the cloud computing environment efficient.

Minimum makespan and enhance the utilization are the main QoS parameter for literature review.

II. SCHEDULING AND LOAD BALANCING

Figure1 illustrates the task matching using virtual machines (VMs) and the load balancing concept. The cloud task manager accepts the request from the client, verifies the task's properties, such as the presence of dependent and independent tasks, and then passes the request to the scheduler for processing and scheduling. The scheduler makes use of the scheduling algorithm to identify which virtual machine is the best appropriate for a given task and to assign tasks to it. The scheduler assigns run-time appearance jobs to the most appropriate virtual machines (VMs) based on whatever virtual machine is the least used at the time of the appearance. Task migration happens in real time depending on the needs of the load balancer, allowing an overloaded VM to move its duties to an idle VM or a VM with a lower burden. Task migration is enabled by the load balancer. Whenever it detects an idle VM or a VM with the least amount of load based on the resources' current status information. Resource monitor collect information about the capabilities of each virtual machine, the current load on the machine, and the number of tasks in the execution/waiting queue. The process requirement, which includes the duration of the tasks to be performed, is provided by the user, and the scheduler utilises this information in order to make operational choices about how to accomplish the tasks.

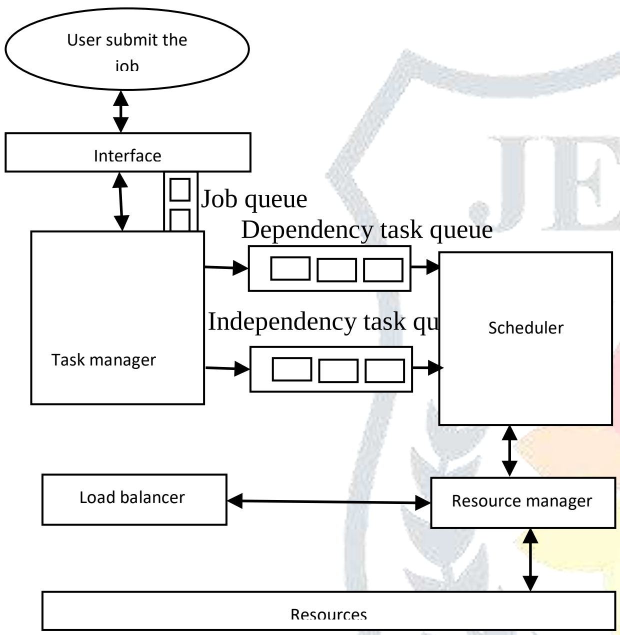

Figure 1: Scheduling and load balancing design [1]

2.1 Load balancing parameters

Here, we will examine load balancing parameters in cloud computing. Inside this previously, there are numerous load-balancing techniques presented by academics/researchers [1,2,3,4,5]. For the implementation of load-balancing algorithms, the literary fiction on load balancing offered the following measures, which we summarise as follows:

- Throughput: The throughput of a virtual machine would be the amount of jobs completed in a unit of time by the virtual machine itself. The presence of a high throughput means the system is functioning well. The completion time of a machine is inversely related to the throughput of the system under consideration.
- Response time: It is the total amount of time spent transmitting, waiting, and servicing. As a consequence, the machine's effectiveness is inversely proportional to its response time. A shorter makespan is achieved by a quicker response time.
- Makespan: It is the amount of time necessary to accomplish all processes submitted to the system. The system's makespan is the maximum amount of time required for the host to run across the data centre. The optimal makespan results in proper load balancing of the system.
- Scalability: It is a characteristic of a systems or model that refers to the system's ability to function in unexpected circumstances. It refers to a balanced system's capacity to continue functioning when the number or quantity of jobs or workload increases. Resources will be adjusted on a regular basis in a flexible cloud system.
- Fault tolerance: The fault-tolerant method has been one of the system's capacities for maintaining service in the event of the failure of one or more system components. Additionally, it eliminates obstacles resulting from logical inconsistencies.
  The number of failure sites may be used to quantify fault tolerance (i.e., single point failure or multipoint failure). Service providers may need extra resources or virtual machines to handle certain cloud system issues. While this procedure adds to the consumer's costs, the final result is a fault-free system.
- Migration time: When a job or virtual machine is moved from a resource to another, the time required is known as the transfer time.The task migration might take place across VMs on the same host or between VMs on a different host.

The degree of imbalance is a statistic that evaluates the amount of imbalance between VMs.
Performance: It evaluates the efficiency of the system after it has performed a load-balancing algorithm.

- Energy consumption: It calculates the overall amount of energy consumed by all of the nodes. Load balancing helps prevent overheating by spreading the load over all nodes, resulting in a reduction in energy consumption.
- Carbon emission: It determines the total amount of carbon dioxide emitted by all sources of energy. Load balancing is critical in reducing this metric since it transfers loads from under loaded terminals and causes them to shut down.

2.2 Issues in load balancing algorithms

[6] When working in the cloud, there are a number of challenges to overcome. These include load balancing, data loss, and cloud security. The topics covered will be specifically related to load balancing and will include topics such as how to map incoming requests (tasks) to virtual memory in such a way that each task can complete its execution in the shortest amount of time possible. Each load balancing algorithm has a different goal, which varies depending on the objective function used in the calculation. Several algorithms, for example, are designed to maximise throughput and average resource utilisation ratio, whereas others are designed to minimize response times, makespan times, and execution times. Neither a conventional algorithm nor a meta heuristic is capable of achieving all of these objectives at the same time. It is also important to note that the performance of the load balancing method is dependent on the geographical distribution of the nodes, their security, and the cost of communication between them. When nodes are distributed evenly across a defined area, it is possible to measure fault tolerance and efficiency in a relatively short period of time. A delay may occur if nodes are located in different geographical locations.

Nowadays, static or dynamic load-balancing algorithms are used to distribute the load across a network or systems. They are not dependent on the current state of the system (node); rather, those who have been based on prior awareness (state) about just the node's capacity, memory size, and user requirements. In contrast to dynamic algorithms, which are not state-dependent, the principal disadvantage of this type of algorithm is that computer resources and tasks fail quickly. However, dynamic algorithms are, by their very nature, more complex than static algorithms. [7] aside from that, there are some flaws in the load balancing methodology. For example, it is extremely difficult to analyze the level of every node and task. Whereas server queue size is really a good indicator of weight, measuring it is time-consuming and requires special equipment. Nobody knows when or who will migrate a task, so it is impossible to predict. The task approach to immigration determines how tasks are moved from one place to the next.

- Rules on data gathering: Determine what information on the workload should be gathered, when the information should be collected, and how the information should be acquired.
- Aggravating rules: Define the duration of time and during that task balancing will be implemented and maintained.
- Picking rules: Determine which jobs from an overloaded virtual machine/node should be moved to a virtual machine/node that is less overburdened.
- Place rules: Identify a suitable virtual machine to which we may move the task so that it can be completed.
- Migration rules: This function to determine whether or not such a node is in a state that is appropriate for task delegation.
- Node Information policy: Following a given time interval, this policy decides the condition of each node. The use of elastic computing can be beneficial when the load on a virtual computer exceeds its capability (horizontal or vertical scalability)

III. CONDUCT A SURVEY OF CURRENT LOAD BALANCING ALGORITHMS/MECHANISMS.

Over the last two decades, several cloud load balancing algorithms/technologies have been suggested and implemented; some are static [2, 8, 9, 10, 11], and others are dynamic [12, 13, 14]. Static algorithms are a subcategory of load balancing algorithms that need previous knowledge of the node's characteristics, the job or task specifications, and the availability of cloud resources. Only when workload fluctuation is minimal does the static method optimise the QoS metric. Dynamic algorithms take the current load on each computer unit (also called a node) in the system into account. This technique continually monitors virtual computers in cloud data centres.

[1] Presented a weighted round robin method that takes into account the power of each node/VM and the task size of each arrived work while distributing jobs to the most suited nodes/VMs. IWRR algorithms employ three separate stages to address the three distinct circumstances that occur throughout the task's life cycle. The static scheduler method prioritises initial task delivery by equitably assigning job requests across participating VMs depending on the VMs' node/strength configuration and the length of the requested job. The dynamic scheduler takes into account the load on all configured nodes/VMs and determines the approximate completion time of the present load in conjunction with the estimated completion time of the arrived task in each of the participating nodes/VMs. Following that, the shortest probable completion time for this particular work was calculated in one of the nodes/VMs using the aforementioned computations, and the job was allocated to this VM. The response time is the key criterion for this algorithm's quality of service (QoS).

[2] Presented an enhanced min-min (PA-LBIMM) method that reduces makespan time and maximises resource efficiency in comparison to the min-min algorithm. He utilised the min-min method to initialise the virtual machine. In the second phase, he picks

the shortest job possible from the heavily loaded computer and estimates the time required to complete it on the other virtual machine. Then, the Min-Min algorithm calculates the time required to complete the task in comparison to this minimal completion time. If the value is less than; the task is allocated to the virtual machine that generated it.

[11] The authors utilized a credit-based scheduling method in a cloud computing environment to improve resource usage and shorten the makespan in this study. He gave credit for each assignment based on its duration and priority. The resources will be allocated the tasks with the greatest credit initially. He tests the algorithm using the CloudSim simulator and compares the results to those of other algorithms based on job duration and priority, claiming that his methods outperform the competition. When the number of tasks rises, the algorithm does not produce a better outcome..

[15] Introduced a heuristic-based approach for cloud computing's diverse environment. Both tasks and virtual machines are ordered in highest to lowest of length and capacity in this method. He assigned the work to VMs using FCFS. He set a criterion for virtual machines: if they are working at less than 25% of capacity, they are termed underloaded; if they are operating at more than 85% of capacity, they are deemed overloaded. He also conducts a comparison of the suggested load balancing algorithm with the FCFS, Min-Min, and SJF algorithms. The simulation is repeated 100 times in order to determine the makespan, and he wins the competition by achieving superior outcomes. As a result, it offers adequate load balancing and used less time. CloudSim was used to simulate the situation.

[17] For the goal of maximizing resource utilization in a cloud context, the authors of this study developed a genetic algorithm-based load balancing approach for Cloud Computing. According to the findings of the investigation, the suggested load balancing approach not only beats a few current strategies, but it also assures that the customer task fulfils the needed Quality of Service (QoS) standards. While it has been assumed that all tasks have the same priority, it is possible that this is not the case; this may be accommodated in the job unit vector, which is then taken into consideration in the fitness function, if necessary. Moreover, although a relatively basic GA technique was utilized in this study, modifications in the crossover and selection procedures might be applied in future studies to get more efficient and tailored findings.

[18] That seeks to schedule tasks and jobs in a way that takes the least amount of time and money. In order to carry out and simulate the tasks assignment method as well as distributed work scheduling, CloudSim1 is being used. The outcomes are compared to the results of the Activity-Based Costing algorithm (ABC). The conclusion is that the scheduling algorithm that was used is superior than the activity-based costing approach that was used.

[19] They created a load balancing approach for cloud computing environments that is based on the forage for food strategy used by honey bees, which they found to be effective. This method not just to balances the load, but it also takes into consideration the priority of jobs that have been withdrawn from excessively laden Virtual Machines. The jobs that have been withdrawn from these virtual computers are referred to as honey bees, which are responsible for updating global information. Additionally, the relative priority of the jobs is taken into consideration by this method. Inspire by honey bee activity, load balancing aims to boost overall processing throughput, whereas priority-based balancing seeks to reduce the amount of time a task spends waiting in the queue of a virtual machine. Because of this, the reaction time of virtual machines has been shortened.

[21] This paper proposes a Priority Based Job Scheduling strategy based on IBA and the EASY backfilling algorithm to address the issues that have been identified with the existing backfilling and job scheduling systems. According to simulation findings, when compared to other algorithms, the suggested approach increased the usage of available resources. This study primarily describes a job scheduling algorithm based on priority, as well as the IBA and EASY backfilling strategies for cloud computing environments. It combines the advantages of the priority and backfilling strategies, and as a result, it improves the work scheduling efficiency of the system, as well as the pace at which the system's resources are utilized.

[23] The modified throttled method works in a similar way to the Throttled algorithm in that it keeps an index table of virtual machines as well as their states. An attempt has been made to increase response time while also maximizing the utilization of the virtual machines that are now accessible. A cost-effective technique to handling the load on servers is shown in this work, which takes into account both the availability of VMs for a particular request and consistent load sharing across the VMs for the number of requests serviced. The effort was aimed at developing an efficient system for load balancing, as demonstrated by its two distinct aims. One is the response time necessary to service the requests, and the other is the distribution of the load across the existing virtual machines ("VMs"). Comparing the suggested method to current Round-Robin and Throttled algorithms, the reaction time of the proposed algorithm has significantly improved. When compared to existing techniques, the proposed algorithm distributes load roughly uniformly among VMs while providing faster response times.

[24] The goal of ant colony optimization is to find the most efficient route between a food source and an ant colony based on the behaviour of the ants. The goal of this strategy is to distribute work load across the nodes in an efficient manner. When the request is initiated, the ant begins its journey towards to the source of food, starting at the head of the colony. He developed a modified technique to ant colony optimization, which has been applied to cloud or grid network systems, with the primary aim of node load balancing as the major purpose of the approach. Using the pseudocode supplied in the appendix, it is easy to see how ants are

generated and their functions as a consequence of the pheromone trails they follow and the nodes they meet. With this updated method, the original strategy, in which each ant produces its personal distinctive result set, which is subsequently integrated into a full solution, surpasses the original approach. As an alternative to continually maintaining their own data set, the ants keep updating an only one result set. Instead of being compiled only once in a time, the solution set is progressively built upon and continually improved upon in this manner. The second advantage of this strategy is that each ant's work is specialised rather than being general, and the task is dependent on the type of initial node that is encountered, whether it is overloaded or underloaded. The task of each ant is specific rather than being generic.

[25] It is a static load balancing method (Opportunistic Load Balancing Algorithm), which means that it does not take into account the present workload of any of the systems. As a consequence, it maintains each node occupied by allocating incomplete jobs to available nodes in a random fashion. As a result, whenever it came to load balancing, the algorithm does not work well at all. It is unable to compute the implementation time of the node, resulting in a decrease in the processing task's performance. Furthermore, when nodes are not in use, the cloud system will encounter bottlenecks.

[26] This is a dynamic load balancing algorithm that determines which virtual machines are least loaded and assigns them a load. Load balancing controllers keep track of all servers and requests in the server's index table. As a result, when a new request is received, the data centre anticipates that the index table will identify the servers that are least loaded or idle. That is, the algorithm assigns load to servers in a first come, first served fashion. The task is identified by its server-id, and when the server receives a load, its state is updated in the index table. Similarly, when a task is completed, the data centre and controllers receive the information, which reduces the index table's server state. When an internet user submits a request, the load balancer recursively scans the index table and assigns processes appropriately.

[27] Alternatives that have been suggested in cloud computing, the Whale's approach is used for load balancing purposes. The first thing that is accomplished in this study is to illustrate how cloud computing handles balanced load situations. Second, a whale optimization method is used to improve population variety. This algorithm depends on the Sine and Cosine Algorithm for local optimization and the Leapfrog Algorithm for individual screening, and it is based on the Sine and Cosine Algorithm for individual screening. Next, a simulation was performed to demonstrate that the algorithm suggested in this study is superior to earlier algorithms in terms of task performance, fairness, and energy saving, as well as overall performance (such as the ant colony algorithm, particle swarm algorithm, and whale optimization algorithm).

[28] In this work, a novel Spider Monkeys method for load balancing is suggested, which is named Spider Monkey Optimization Inspired Load Balancing (SMO-LB). This algorithm is based on replicating the foraging behavior of Spider Monkeys, and it is described in detail. Its goal is to distribute the workload among virtual machines in order to improve performance by decreasing the makespan and reaction time.

Table 1 An Overview of the Load Balancing Algorithms that has been reviewed, organized by Response Time and Makespan Time.

Reference	Strength	Future scope	Tool	Compared with
[1]	Response time is the main QoS metrics for this algorithm. It is more suitable for heterogeneous resources with heterogeneous/homogeneous jobs.	It does not maintain the execution state of the job during migration. It does not support the distributed computing in multiple PEs(processing elements) in the participating heterogeneous VMs.	CloudSim	Round robin, weighted round robin
[2]	He asserts that experimental data indicate that task completion times are shorter when compared to Min-Min. In most circumstances, a 20% improvement in the average resource utilisation ratio.	The interdependence of each activity was not taken into consideration while creating the paper deadline. The geographical placement of tasks and resources was not taken into consideration.	Matlab	Min-Min algorithm
[11]	Makespan time of the machine is lesser when compared with Min-	By taking the task's deadline into consideration, the proposed plan	CloudSim	Min_Min and priority based

	Min	may be improved.		other algorithms
[15]	The proposed approach uses traditional techniques for load balancing and horizontal expansion of the datacenter in a cloud environment. This method adopted a task migration strategy rather than a virtual machine migration strategy. Not only does the proposed approach reduce makespan time, but it also reduces the likelihood of a virtual machine being overloaded or underloaded.	He did not take into account the task's priority, timeliness, or other quality of service (QoS) criteria in this study.	CloudSim	FCFS,Min-Min,SJF
[17]	Using this approach, you can make optimal use of the resources available in your cloud environment. According to the findings, the suggested method for load balancing not only surpasses a few current solutions, but it also ensures that the qualities of service (QoS) requirements of the customer task are met.	variation of the crossover and selection strategies could be applied as a future work for getting more efficient and tuned results	Cloud Analyst	Stochastic Hill Climbing (SHC).,FCFS,RR
[18]	This improved ABC algorithm only takes the initial research on task scheduling in Cloud platform. However many issues remain open.	The optimization of this technique should focus on concurrent job scheduling rather than independent task scheduling in a cloud context.	CloudSim	Activity based costing
[19]	This algorithm not only balances the load, but also takes into consideration the priorities of tasks that have been removed from heavily loaded Virtual Machines.	This approach not only balances the load, but also takes the priority of work removed from heavily loaded Virtual Machines into account.	CloudSim	honey bee behavior inspired load balancing, Dynamic Load Balancing (DLB)
[21]	proposed algorithm enhanced the resource utilization.	fairness for low priority jobs when high priority jobs keep on coming.	CloudSim	FCFS,EASY,BA
[23]	Proposed algorithm distributes load nearly uniform among VMs, with improved response time compared to existing algorithms	As a future scope the present work need to be focused on changing the data structures used for maintaining the index.	Cloud Analyst	Round Robin and Throttled Algorithm
[24]	The solution set is gradually built on and continuously improved upon rather than being compiled only once in a while.	This technique does not consider the fault tolerance issues. Researchers can proceed to include the fault tolerance issues in their		Only claim his algorithm is better than normal ACO.

		future researches.		
[25]	Opportunistic load balancing algorithm attempts to keep each node busy	It fails to calculate the node's implementation time		
[26]	The proposed method EDLBHA overcomes the load balancing problem for cloud computing. Simulation results clearly shows that our proposed methodology EDLBHA, having outstanding results in terms of efficient energy utilization, waiting time, response time and turnaround time.	The capacity of datacenters can be increased or decreased depends on the request of the users.	CloudSim	Round Robin, Honey bee Method
[27]	Simulation experiments show that the algorithm in this paper has good effects in terms of task execution time, fairness and energy consumption.		CloudSim	ant colony algorithm, particle swarm algorithm,
[28]	The developed method not only handling the issue of load balancing but also takes into consideration the capability and accessibility of the resource through the proposed grouping strategies of tasks and virtual machines.	This technique does not consider the fault tolerance issues. Researchers can proceed to include the fault tolerance issues in their future researches.	CloudSim	Round-Robin and Throttled

IV. CONCLUSION

To provide the quickest connectivity for all nodes or devices that utilize cloud computing services, the finest load balancing methods must be used. We have examined various load balancing strategies in the Cloud context in this article. We discussed the methods that have already been proposed by several scholars. Additionally, we uncovered flaws in existing static load balancing techniques. Additionally, the various load balancing methods are compared in terms of response time and makespan time.

REFERENCES

[1] D. Chitra Devi and V. Rhymend Uthariaraj , “Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks”, Volume 2016 |Article ID 3896065 | https://doi.org/10.1155/2016/3896065

[2] Huankai Chen, F. Wang, N. Helian and G. Akanmu, "User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing," 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH), 2013

[3] Kumar, P., & Kumar, R. (2019). Issues and challenges of load balancing techniques in cloud computing: a survey. ACM Computing Surveys (CSUR), 51(6),1-35.

[4] Hota, Arunima & Mohapatra, Subasish & Mohanty, Subhadarshini. (2019). Survey of Different Load Balancing Approach-Based Algorithms in Cloud Computing: A Comprehensive Review. 10.1007/978-981-10-8055-5_10.

[5] Sambit Kumar Mishra, Bibhudatta Sahoo, Priti Paramita Parida,Load balancing in cloud computing: A big picture,Journal of King Saud University - Computer and Information Sciences,Volume 32, Issue 2,2020,Pages 149-158

[6] KUMAR, MOHIT. (2013). A Technique to reduce the Economic Denial of Sustainability (EDoS) Attack in Cloud.

[7] KUMAR, MOHIT. (2018). Load balancing algorithm to minimize the makespan time in cloud environment.

[8] Atul Vikas Lakra, Dharmendra Kumar Yadav, Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput optimization,Procedia Computer Science,Volume 48,2015.

[9] Li, Ji & Feng, Longhua & Fang, Shenglong. (2014). An Greedy-Based Job Scheduling Algorithm in Cloud Computing. Journal of Software. 9. 10.4304/jsw.9.4.921-925.

[10] P. Samal,P. Mishra, "Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing", International Journal of computer science and Information Technologies, Vol.4, No.3, pp.416-419, 2013.

[11] Thomas, A., Krishnalal, G., Raj, V. P. J. Credit based scheduling algorithm in cloud computing environment. Procedia Computer Science, 2015, 46: 913–920.

[12] Espadas J., Molina A., et al. A tenant-based resource allocation model for scaling;software-as-a-service applications over cloud computing infrastructures. Future Generation Computer Systems, 2013, 29(1): 273–286.

[13] Kumar, Mohit and Sharma, S. C. Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment. Computers & Electrical Engineering, 2017.

[14] Somasundaram, Thamarai Selvi & Kannan, Govindarajan. (2014). CLOUDRB: A framework for scheduling and managing High-Performance Computing (HPC) applications in science cloud. Future Generation Computer Systems. 34. 47 - 65. 10.1016/j.future.2013.12.024.

[15] KUMAR, MOHIT & Sharma, Subhash. (2017). Dynamic load balancing algorithm to minimize the makespan time and utilize the resources effectively in cloud environment. International Journal of Computers and Applications.

[16] Omara, Fatma & Abdelkader, Doaa. (2012). Dynamic Task Scheduling Algorithm with Load Balancing for Heterogeneous Computing System. The Egyptian Informatics Journal. 13. 135-145. 10.1016/j.eij.2012.04.001.

[17] Dasgupta, Kousik & Mandal, Brototi & Dutta, Paramartha & Mandal, Jyotsna & Dam, Santanu. (2013). A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing. Procedia Technology. 10. 340–347. 10.1016/j.protcy.2013.12.369.

[18] Selvarani, Shaila & Sadhasivam, Sudha. (2010). Improved cost-based algorithm for task scheduling in cloud computing. Computational Intelligence and Computing Research (ICCIC). 10.1109/ICCIC.2010.5705847.

[19] Dhinesh Babu L.D., P. Venkata Krishna,Honey bee behavior inspired load balancing of tasks in cloud computing environments,Applied Soft Computing,Volume 13, Issue 5,2013,

[20] Wong, Adam & Goscinski, Andrzej. (2007). Evaluating the EASY-backfill job scheduling of static workloads on clusters. 64 - 73. 10.1109/CLUSTR.2007.4629218.

[21] Dubey, Kalka & KUMAR, MOHIT & Chandra, Mayank. (2015). A Priority Based Job Scheduling Algorithm Using IBA and EASY Algorithm for Cloud Metaschedular.

[22] Mishra, Ratan & Jaiswal, Anant. (2012). Ant colony Optimization: A Solution of Load balancing in Cloud. International journal of Web & Semantic Technology. 3. 10.5121/ijwest.2012.3203.

[23] S. G. Domanal and G. R. M. Reddy, "Load Balancing in Cloud Computing using Modified Throttled Algorithm," 2013 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2013, pp. 1-5, doi: 10.1109/CCEM.2013.6684434.

[24] K. Nishant et al., "Load Balancing of Nodes in Cloud Using Ant Colony Optimization," 2012 UKSim 14th International Conference on Computer Modelling and Simulation, 2012, pp. 3-8, doi: 10.1109/UKSim.2012.11.

[25] "Cloud Load Balancing Image,".[Online].

[26] Rai, S., Sagar, N., & Sahu, R. (2017). An Efficient Distributed Dynamic Load Balancing Method based on Hybrid Approach in Cloud Computing. International Journal Of Computer Applications, 169(9), 16-21. doi: 10.5120/ijca2017914876

[27] Xiaoxia Li. (2021). A Study into Load Balancing in Cloud Computing Based on Whale Optimization Algorithm. CONVERTER, 2021(7), 180-189. Retrieved from http://www.converter-magazine.info/index.php/converter/article/view/487

[28] Alshattnawi, Sawsan, and AL-Marie Mohammad. "Spider Monkey Optimization Algorithm for Load Balancing in Cloud Computing Environments."

[29] Sinha, G., & Sinha, D.K. (2020). Enhanced Weighted Round Robin Algorithm to Balance the Load for Effective Utilization of Resource in Cloud Environment. EAI Endorsed Trans. Cloud Syst., 6, e4.