Static and Dynamic Load Balancing Algorithms in Cloud Computing: A Comparative Survey

[1]Ajay Singh Thakur
[1]Assistant Professor
[1]Department of Computer Science,
[1]School of Study Computer Application, Bastar University, Jagdalpur, Chhattisgarh, Pin- 494001.

Abstract: Cloud computing is the most reliable, competent and useful in the world of current technology. Cloud computing is a technique that consists of several Web-enabled software and services. Resource programming and resource allocation in cloud computing is the vast area of research today. Therefore, the efficient algorithm for scheduling and allocating resources in the cloud will result in more elastic and efficient cloud computing services. The cloud implementation uses several steps from maintenance to deployment, etc. In this document, we conduct a survey of different load balancing algorithms and classify them based on their dynamics and find out the challenges facing these algorithms. So, we compare these algorithms with an approach to the Markov hidden model technique and with this technique, we will try to clarify the load balancing scenario of the cloud service when programming resources. Markov's hidden model in cloud resource scheduling can help predict the next workload that will arrive on the server, under which the system will allocate resources to a given job. This will contribute to better CPU utilization as well as a faster execution environment in cloud computing.

Index Terms **- Cloud Computing; Load balancing; Proactive technique; Resource allocation; Resource Scheduling.**

I. INTRODUCTION

The cloud is a web-based service. When it comes to cloud computing, it allows access to system shares and high-level services on demand over the Internet. Many researchers and industries have shown their great interest in the field of cloud computing. Cloud computing has taken an important step in the IT sector. Cloud computing is the vast concept. Cloud computing provides a means of accessing applications as utilities on the Internet. The whole internet can be considered as the cloud of many resources. Cloud computing provides on-demand services, which is the main profitable area, as it is independent of the device and location. Cloud computing also has extensive network access worldwide, meaning you can access anything anywhere on any device that supports the cloud. The cloud is a large data or storage center and can be mainly of three types, namely public, private or hybrid. Google, Microsoft, Amazon and many large industries use the cloud. Cloud services are popular due to efficient processing and storage technologies: cloud servers are made up of a large number of resources, as they always include numerous services to scale capacity and maintain its performance. In many cases, one server has to use the services and resources of another server and for each resource they have to pay to that server. On the other hand, many resources are not being used properly; therefore these unused resources have yet to be paid for. The main objective is to study different techniques to manage these resources in the cloud so that the programming and allocation of resources are efficient. Therefore, the main job is programming resources and allocating resources.

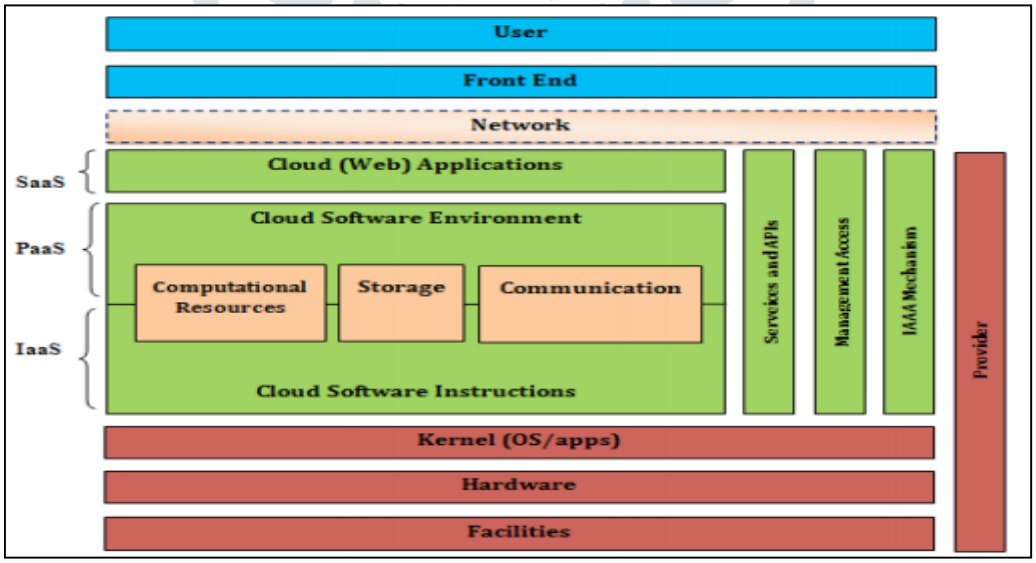

Figure 1: Cloud Computing Architecture

The cloud is mainly composed of three implementation models. They are IaaS, PaaS and SaaS. These three models implement the cloud as a complete infrastructure. Infrastructure as a service (IaaS) tends to provide users with access to cloud computing resources, such as servers, storage, networks, etc. Platform as a Service (PaaS) offers a field in which applications can be developed and managed. Software as a Service (SaaS) provides access to those applications and software that have been developed and managed in PaaS. The cloud can be of many types. The main types of clouds are private, managed by a single organization, managed by the same organization or by a third, second public cloud, which is an open cloud and the third is a hybrid cloud which is a mixture of both public and private cloud . There is also a commercial cloud, but that's not the point of view at the time. Load balancing plays a very important role in the field of cloud computing. The cloud workload needs to be

balanced to provide QoS (quality of service) to users and even providers. Therefore, the main thing is to find an effective strategy for distributing the workload between different processors. Here, in cloud computing, load balancing can help reduce runtime, wait times, performance, response times and many other areas of cloud computing. Therefore, you need to make sure that all processors in the system do the same job and all jobs. It must be finished at the same time.

There are several algorithms defined to balance the workload in cloud computing. The load balancing algorithm works very easily. View heavy processors that contain many tasks / jobs and then transfer these tasks to low-load processors to speed up execution and reduce waiting times. The main reason for the load balancing algorithm is to make each processor equally busy and to make sure each processor finishes running at about the same time. A distributed system contains many processors that work together and independently, regardless of whether they are connected to each other. Each processor has a certain amount of work to do at any given time based on its ability to work. The work here must be distributed among all processors based on their processing capacity and processing speed, so as to minimize execution and waiting times. Therefore, the load balancing algorithm should be equally effective in producing the best possible results.

Here, resource scheduling and resource allocation is the main area of cloud computing load balancing management. Resource scheduling is a process that creates a schedule that specifies when and where and on what resource a job should be run, including how long the task should run. This is necessary because each server has limited resources to perform the required work and a series of activities for these resources, so these activities must be scheduled. Considering that, when we talk about resource allocation, to allocate resources to activities, it is necessary to program which program resources, as well as plan the activities that reach these resources.

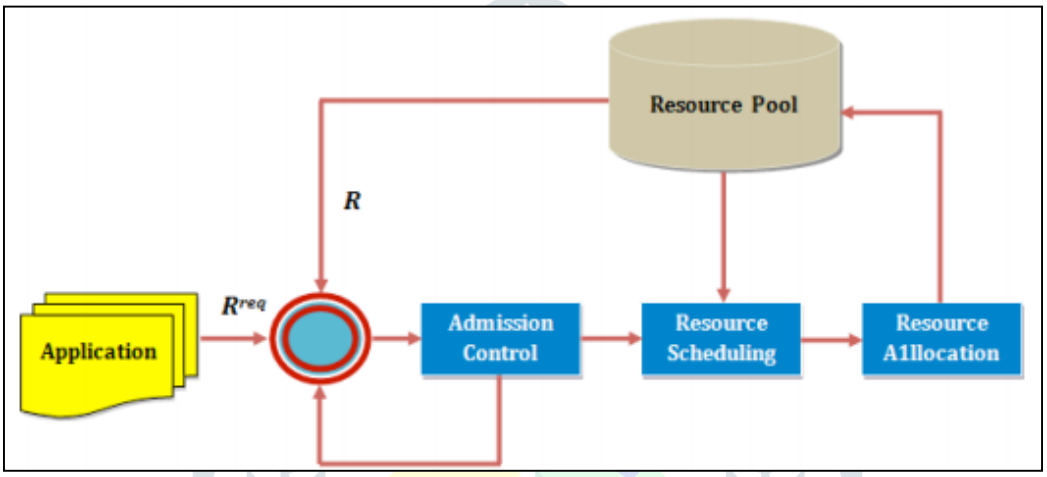

Figure 2: Resource Allocation & Scheduling

It is necessary to check the availability of processors so that future activities can be optimally allocated to the best resources. The burden must also be balanced between the same type of resources. Creating a scalable cloud resource management system that meets all these requirements is very difficult. There are several resource programming algorithms defined in different ways to find the best and optimal load balancing in cloud computing. The main reason for these algorithms must meet all these requirements, including costs, time and CPU usage.

II. LITERATURE SURVEY

The ***S. Santra and K. Mali [1]*** proposed a Round Robin approach in a circular fashion and attempted to clarify a better load balancing scenario in cloud computing. Their work translates into a reliable and fast working environment of the activity assigned by the user. He also created an effective communication framework between the virtual machine and the broker. The whole framework was created in CloudSim and Java. In addition to the Round Robin programming algorithm, they also used the FCFS programming policy for better results. The main objective of the document was the load balancing of virtual machines in cloud computing using the Round Robin technique implemented in CloudSim.

O. Kaneria and R.K. Banyal [2] described several stages of cloud implementation, from start to implementation, use and maintenance, so no. The document focused on various aspects of the cloud, such as security, speed, privacy, etc. They modified the base load balancing algorithms and made effective use of cloud resources. Here, they worked on CloudSim and modified the algorithms present by default in CloudSi, that is Round robin and strangled. The primary goal of the document was to distribute the business to their respective data centers and then allocate resources. He also searched for free hosts in the data center and, if he finds the free host, allocates a process to that host and if there are no free hosts, checks the host with the maximum number of processors and then assigns the activity to that 'host. This translates into an effective distribution of activities to data centers.

Nikravesh et al. [3] proposed an approach to address cost compensation in cloud computing. They proposed an automatic scaling system based on Markov's hidden model that addresses resource allocation problems. They had experimented with their model in the Amazon EC2 infrastructure. This automatic scaling system demonstrates that Markov's hidden model works better with 97% accuracy than other algorithms. They experimented to measure accuracy in measuring different scale situations, such as CPU usage, performance and response times. A more detailed experiment is needed to fully confirm the assessment.

S. Joshi and U. Kumari [4] analyzed the main challenges and problems of load balancing in cloud computing. According to the author, the cloud is emerging rapidly and for this reason a large number of users are attracted to cloud computing. Here, the author discussed several challenges and problems, such as cloud service models, virtualization, load balancing, as well as many different load balancing algorithms, such as round robin, max-min, min-min, central manager, threshold, honey, etc. to. The author described the pros and cons of all these algorithms and compared them. The main objective of this work was to evaluate load balancing efficient in terms of performance, use, stability and response times by comparing the various existing load balancing algorithms.

H. Mehta et al. [5] explained the main problem of scheduling resources in cloud computing, as they directly affect system performance in the cloud. According to the document, monitoring and forecasting of resources were the keys to obtaining the use of high-performance resources. The author used the hidden Markov model to monitor the resources available in the cloud. The resources were therefore classified in terms of lower, medium and heavy load categories. After monitoring the resources using the hidden Markov model, the appropriate resource scheduling algorithm was used to assign the activities to those resources. After monitoring, the author applied several algorithms such as FCFS, Min-min, Max-min, etc. and then compared the results. According to the usage model, different algorithms can be used for different situations.

K. A. Nuaimi et al. [6] described several load balancing challenges in cloud computing, such as the spatial distribution of cloud nodes, storage / replication, the complexity of the algorithm and the point of failure. The author also looked at several algorithms based on resource programming policy, such as VM mapping, Ant colony, etc. Furthermore, the comparison of different algorithms has been described in terms of replication, speed, heterogeneity, network overload, fault tolerance, etc.

H. Shoja et al. [7] described several load balancing algorithms and compared them to each other in terms of parameters such as response time and data processing time, etc. The response time of these two algorithms was the same, i.e. there was no effect on the data center request time after changing the algorithms. While the cost calculated for the virtual machine per hour was not the same. The accelerated algorithm reduced the cost effect, therefore the accelerated programming algorithm proved more efficient in terms of costs for load balancing.

F. Alam et al. [8] described the need for efficiency in cloud computing. According to the author, the Round Robin load balancing algorithm (RLBA) was the algorithm most used for its simplicity, but it is still not efficient enough. Therefore, the author introduced two new approaches, namely Adaptive RLBA and Predictive RLBA and therefore validated the effectiveness of both algorithms through the simulation results. The comparison parameters used here were the correlation of the server load and the change in load. The RLBA predictive was a supervised learning model called the SVM (support vector machine). Both modified algorithms performed better than the previous algorithm in a uniform and uneven web traffic area.

S. Aslam and M.A. Shah [9] provided a brief survey on load balancing algorithms. They described the structured and complete description of research on various algorithms in cloud computing. The author assessed the performance of static and dynamic load balancing algorithms such as Ro und Robin, opportunistic load balancing and LB min-min, Max-min and min-min which were listed in the category of load balancing algorithms Static and ant cologne, honeybee, strangled and cardboard, which resides in the category of dynamic load balancing algorithms. The author compared these algorithms based on impartiality, response times, performance, fault tolerance, performance, speed and complexity.

R. Buyya et al. [10] CloudSim proposed: an extensible simulation toolkit that allows you to model and simulate cloud computing environments. The cloudim toolkit presented by the author supports the modeling and creation of one or more virtual machines, jobs and their assignments. It also allowed for simulation of multiple data centers. It has helped increase the reliability and automatic scaling of applications. It was designed to manage the complexities derived from simulation environments.

Sharma and S.K. Peddoju [11] briefly explained load balancing based on response time in cloud computing. The author has described several load balancing issues, such as balancing a server after a server has been overloaded and constant queries, etc., which have increased computing costs and bandwidth consumption. In relation to these problems, the author introduced an algorithm that adopts a preventive load balancing approach when considering the response time of each request. Resource allocation was decided based on response time. This proposed model was dynamic in nature. This algorithm has saved unnecessary communication bandwidth between load balancing and virtual machines, therefore it only takes into account the response time that is readily available. The algorithm proved to be good and efficient in terms of costs and response times.

R. Kong [13] introduced an economic approach to resource programming in cloud computing. The document stated that programming balanced load resources also entails a reduction in overall costs. This new modified profitable algorithm proposed here minimizes the total cost of resources. The author designed a balanced load planner and classified resources into two types, that is, based on reserves and demand. The author also implemented the existing algorithm and then compared it to the new modified algorithm. The experiment showed that the Profitable Resource Scheduling Algorithm (ESRB) improved performance in terms of total resource cost.

H. Chen et al. [14] studied the problem of monitoring and forecasting resources in cloud computing environments. The author had implemented an adaptive resource monitoring framework for cloud computing and introduced a resource prediction mechanism based on automatic vector regression (VAR) through the correlation between various resources. The experiment effectively monitored the use of resources in cloud computing. The forecasting mechanism proposed here was more effective than the existing forecasting models. The author took multiple causes to achieve effective monitoring. The model performed better than linear prediction. The temporal complexity of the structure remained a difficult problem.

R. Kapur [18] described numerous inspiring algorithms of the nature of the NIA and their comparisons depend on different parameters and their respective areas of application. Nature-inspired algorithms have been classified as bio-inspired algorithms, swarm intelligence-based algorithms, non-SI-based algorithms, physics and chemistry-based algorithms, and other algorithms. The author also explained ISAs in terms of ant colonies optimization algorithms, particle swarm optimization algorithms, firefly algorithms, bee colonies optimization and simulated annealing. All algorithms originate in several phenomena based on swarm intelligence. According to the author, these ISAs can be used in combination with other algorithms to improve their QoS.

Z. He and X. Wang [19] proposed a new activity planning model to solve the load imbalance between virtual machines in cloud computing. In this model, the author optimizes the task execution times, including the task execution times and the use of system resources. Based on this work, a PSO (Particle Swarm Optimization) based algorithm was proposed. The author has improved the standard PSO to increase the performance of virtual machines. The results have been much better than the standard PSO algorithm which has an invariable inertial weight. Virtualization and distributed technology have also been developed by improving the PSO algorithm.

L. Singh and S. Singh [20] proposed a genetic algorithm that programmed workflow applications in an unreliable cloud environment. The proposed genetic algorithm has reduced time to a minimum as well as reducing the failure rate and expanding workflow applications. The proposed genetic algorithm allocates resources for workflow applications that were reliable and the execution cost was less than the budget. The author also compared the proposed genetic algorithm with max-min and min-min programming algorithms in an unreliable cloud environment. The result shows that the proposed genetic algorithm has minimized the execution time and failure rate of workflow applications compared to both algorithms.

H. Chen et al. [21] introduced an improved load balancing algorithm based on the Min-Min algorithm, which increased resource utilization as well as reducing production. The author called the improved LBIMM algorithm which means an improved load balanced Min-Min algorithm. The improved algorithm was simulated using the Mat lab toolbox. The proposed algorithm has shown that the simulation results lead to a significant increase in performance and achieve an improvement of over 20% in the rate of use of resources.

III. COMPARATIVE ANALYSIS

Based on this survey, the results are described below in the form of a table in which the different algorithms are classified according to their types and experiments. Different algorithms work differently to balance the workload in cloud computing. The survey describes several load balancing algorithms based on their VM allocation dynamics, i.e. the algorithm described is static or dynamic or both. The main parameters centered here are the response time and the execution time. The main objective of the research is to study the functioning of different static and dynamic algorithms in approximately the same parameters that can be compared with a new predictive approach to load balancing, i.e. the hidden Markov model. Here, in addition to the dynamics and parameters used, the set of tools in which the algorithm has worked is also described.

Table: 1 Comparison of Static and Dynamic Algorithms in Cloud Computing

#	Reference	Algorithm	VM Allocation	Parameter Used	Toolkit Utelized
1.	S. Santra and K. Mali [1]	Round Robin	Static	Execution Time and Resource Scheduling	CloudSim
2.	O. Kaneria and R.K. Banyal [2]	Modified RR Algo.	Static	Resource Allocation & Utelization	CloudSim
3.	A.Y. Nikravesh et. Al. [3]	Hidden Markov Model	Dynamic	CPU Utilization, Throughput & Resource Time	Amazon EC2 & TPC-W infrastructure
4.	H. Mehta et al. [5]	Resources monitored using HMM & diff. dynamic algorithms	Dynamic	Execution Time	Resource Classifiers & Open Source Tools
5.	H. Shoja et. al. [7]	RR & Throttled Scheduling	Static & Dynamic	Response Time	Cloud Analyst
6.	F. Alam et. Al [8]	Adaptive & Predictive RR	Static	Performance Optimization	-
7.	R. Buyya et. Al. [10]	Space-shared Scheduling Algo.	Static/Dynamic	Execution Time	CloudSim
8.	A Sharma et. Al. [11]	Threshold Algo.	Static	Response Time, Threshold & Prediction Time	-
9.	R. Kapur [13]	CERS Algo.	Static	Execution Time & Cost	Amazon EC2 & Implemented in PHP
10.	H. Chen et. Al. [14]	Resource Prediction Mechanism using	Dynamic	Resource Monitoring &	Adaptive Resource

		VAR		Prediction	Monitoring Framework
11.	R. Kapur [18]	Nature Inspired Algorithms (ACO, PSO, BCO & SF)	Static and Dynamic	Resource Optimization	-
12.	Z. Lui et. Al. [19]	PSO based Algo.	Dynamic	Execution Time & Resource Utilization	MATLAB
13.	L.Singh et. Al. [20]	Genetic Algo.	Dynamic	Execution Time & Makespan	CloudSim
14.	H. Chen. et. Al. [21]	User Priority Guided Min-Min Algorithm	Static	Resource Utilization	MATLAB

IV. Conclusion

Cloud computing is a very fast emerging area, as it provides a quantity or services at any time and in any place, which attracts a number of users day by day. Therefore, much better resource management projects are needed to provide a reliable and fast execution environment for cloud computing. In this document, we examine the best load balancing approach to cloud computing and find HMM that can help balance workload in a much better way than other algorithms. As is known, Markov's hidden model is a predictive technique and can predict workload in a much better way than other algorithms so that resources can be programmed and allocated more accurately. In this survey on load balancing algorithms, HMM also proved to be better than neural networks, genetic algorithm and SVM (Support Vector Machine). Therefore, a number of investigations can be conducted into the use of HMM in the field of resource planning in cloud computing.

References

[1] S. Santra and K. Mali, "A New Approach to Survey on Load Balancing in VM in Cloud Computing: Using CloudSim", International Conference on Computer, Communication and Control, pp. 1-5, IEEE 2015.
[2] Ojasvee Kaneria and R K Banyal, "Analysis and Improvement of Load Balancing in Cloud Computing", International Conference on ICT in Business Industry & Government (ICTBIG), pp. 1-5, IEEE, 2016.
[3] A. Y. Nikravesh, Samuel A. Ajila, Chung-Horng Lung, "Cloud Resource Auto-scaling System Based on Hidden Markov Model (HMM)", International Conference on Semantic Computing, pp. 124-127, IEEE 2014.
[4] S. Joshi and U. Kumari, "Load balancing in cloud computing: Challenges and issues", 2nd International Conference on Contemporary Computing and Informatics (IC3I), pp. 120-125, IEEE 2016.
[5] H. Mehta, V. K. Prasad and M. Bhavsar, "Efficient Resource Scheduling in Cloud Computing", International Journal of Advanced Research in Computer Science vol. 8, No. 3, March-April 2017.
[6] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi and Jameela Al-Jaroodi, "Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", Second Symposium on Network and Cloud Computing and Applications, pp. 137-142, IEEE 2012.
[7] Hamid Shoja, Hossein Nahid and Reza Azizi, "A comparative survey on load balancing algorithms in cloud computing", Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1-5, IEEE 2014.
[8] F. Alam, V. Thayananthan and I. Katib, "Analysis of round-robin load balancing algorithm with adaptive and predictive approaches", UKACC 11th International Conference on Control (CONTROL), pp. 1-7, IEEE 2016.
[9] Sidra Aslam and Munam Ali Shah, "Load balancing algorithms in cloud computing: A survey of modern techniques", National Software Engineering Conference (NSEC), pp. 30-35, IEEE 2015.
[10] Rajkumar Buyya, R. Ranjan and Rodrigo N. Calheiros, "Modelling and simulation of scalable cloud computing environments and the CloudSim Toolkit: Challenges and opportunities", International Conference on High Performance Computing & Simulation, pp. 1-11, IEEE 2009.
[11] Agraj Sharma and S.K. Peddoju, "Response time based load balancing in cloud computing", International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 1287-1293, IEEE 2014.
[12] S. A. Ajila and Akindele A. Bankole, "Cloud Client Prediction Models Using Machine Learning Techniques", 37th Annual Computer Software and Applications Conference, pp. 134-142, IEEE 2013.
[13] Ritu Kapur, "A cost effective approach for resource scheduling in cloud computing", International Conference on Computer, Communication and Control (IC4), pp. 1-6, IEEE 2015.
[14] H. Chen, X. Fu, Z. Tang and X. Zhu, "Resource monitoring and prediction in cloud computing environments", 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence, pp. 288-292, IEEE 2015.
[15] S. Farrag, S. Abbas Mahmoud, El Sayed and M. EL-Horbaty, "Intelligent cloud algorithms for load balancing problems: A survey", Seventh International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 210-216, IEEE 2015.
[16] Jaimeel M Shah, K. Kotecha, Sharnil Pandya, D. B. Choksi and Narayan Joshi, "Load balancing in cloud computing: Methodological survey on different types of algorithm", International Conference on Trends in Electronics and Informatics (ICEI), pp.100-107, IEEE 2017.
[17] Igor N. Ivanisenko and Tamara A. Radivilova, "Survey of major load balancing algorithms in distributed system", Information Technologies in Innovation Business Conference (ITIB), pp. 89-92, IEEE 2015.
[18] Ritu Kapur, "Review of nature inspired algorithms in cloud computing", International Conference on Computing, Communication & Automation, pp. 589-594, IEEE 2015.
[19] Z. Lui and Xiaoli Wang, "A PSO- based algorithm for load balancing in Virtual Machines of cloud computing environment", International Conference in Swarm Intelligence, pp. 142-147, Springer 2012.
[20] L. Singh and Sarbjeet Singh, "A Genetic Algorithm for Scheduling Workflow Applications in Unreliable Cloud Environment", International Conference on Security in Computer Networks and Distributed Systems, pp. 139-150, Springer 2014.
[21] H. Chen, F. Wang, Na Helian and G. Akanmu, "User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing", National Conference on Parallel Computing Technologies (PARCOMPTECH), pp. 1-8, IEEE 2013.