www.revistageintec.net
ISSN: 2237-0722
revista GEINTEC Gestão, Inovação e Tecnologias

Performance Improvement in Cloud Computing Environment by Load Balancing-A Comprehensive Review

K. Ramya[1]; Dr.A. Senthilselvi[2]
[1]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.
[1]ramyak2@srmist.edu.in
[2]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.
[2]senthila3@srmist.edu.in

Abstract
With the advent of cloud computing, the affinity between business and technology had increased manifold, allowing users to access IT resources at their convenience through the pay-per-use scheme. With such huge demand surging day to day, the cloud environment must cater to the user requirements flawlessly and also should be rewarding to the providers of cloud service. To maintain its high level of efficiency, there are several challenges that the cloud environment should tackle. One amongst those challenges is the balancing of load. It is one of the primary features of cloud computing that focuses on avoiding the overloading of nodes where there may be idle nodes or nodes with lesser load present at the same juncture. By keeping an effective check on the load several the Quality of Service (QoS) parameters including response time, throughput, resource utilization, energy consumption, cost etc., can be improved, adding to better performance of the entire cloud environment. Even distribution of load among datacenters will contribute to optimal energy consumption and keeps a check on carbon emissions. In this paper we have presented a methodical review on literature pertaining to load balancing strategies that had been proposed in the cloud environment. We had made in-depth analyses of available load balancing techniques and had come up with their advantages, limitations along with the challenges to be addressed by researchers for developing efficient load balancing strategies in the near future. We had also suggested prospective insights about the aspects in load balancing that could be applied in the cloud environment.

Key-words: Cloud Computing, Load Balancing, Quality of Service, Resource Utilization and Datacenters.

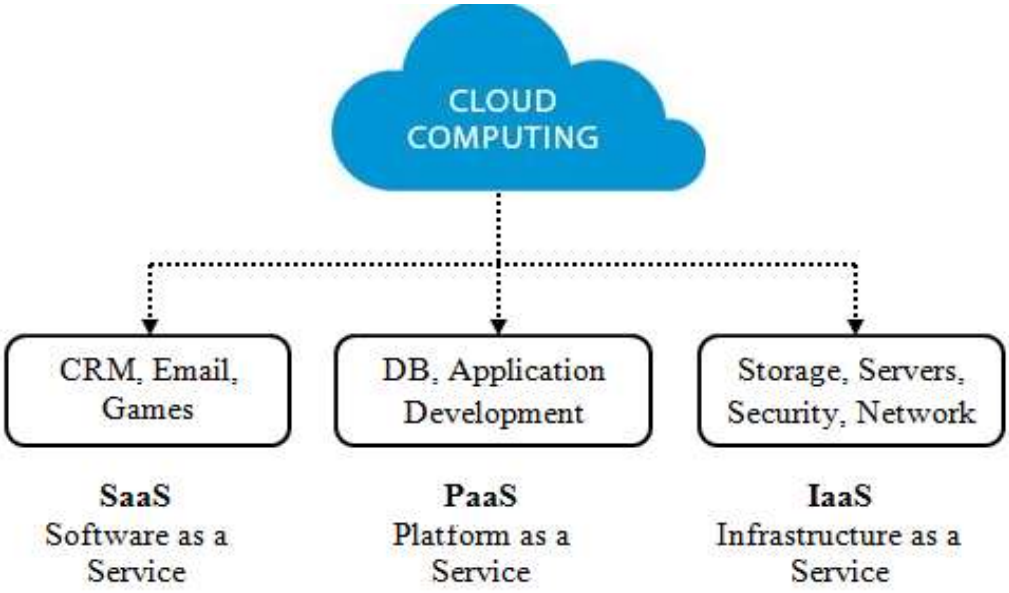
1. Introduction

Cloud technology had increased the ease of use phenomenon for the clients to execute their application. The cloud resources are widely distributed that has to be efficiently utilized for providing

services to clients [1-3]. Clients can place and realize their service requirements through a variety of devices like the laptops, mobile phones, PDAs, tablets, desktop computers and so on. The resources available at the cloud include hardware, software, application development platforms, testing tools for applications yet to be launched and much more. These vast resources should be appropriately mapped to service requirements placed from the client side. Figure 1 shows the resources and the type of servicing at the cloud could be categorized under Infrastructure as a service (IaaS) cloud, Software as a service (SaaS) and platform as a service (PaaS) [4-8]. The cloud computing model follows the pay-as-you-use feature where the clients are billed based on their respective usage. Apart from technology giants like Amazon, Microsoft, Google, SAP, Oracle, VMware, Sales force, IBM etc., there are many start-ups that had ventured into cloud computing technology. The National Institute of Standards and Technology (NIST), in its definition for cloud computing had outlined four deployment models namely public, private, community and hybrid cloud under cloud computing models [9-12].

There are a handful of challenges faced by the cloud computing technology that includes security, load balancing, scalability, availability of services, and efficient management of QoS parameters, energy consumption, data lock-in and reliable performance. Of those mentioned, balancing the load is one of the primary issues that need to be efficiently addressed [13-18]. The available resources need to be optimally managed by assigning or reassigning the load amongst them for maximizing the throughput, performance, resource utilization and at the same time minimizing the cost, response time and energy expended.

Figure 1- Services of Cloud Computing



Received: 28.02.2021 – Accepted: 06.04.2021

Load balancing ensures that no machine is overloaded with tasks while still there are many machines that are under-loaded. A distributed system like cloud needs such load balancing techniques to improve its performance. It either balances or improves or reduces various QoS parameters resulting in improvement of cloud performance. Load balancing is an optimization strategy where scheduling of task comes under the NP hard problem. Most of the load balancing strategies proposed in literature had primarily focused on task scheduling, allocation of tasks, scheduling and allocation of resources and overall management of available resources in an optimal manner [19-20]. The literature proposed lack in-depth analysis and do not present a comprehensive picture about factors that lead to situation where load gets unbalanced. Earlier surveys on load balancing techniques had not provided any systematic classification of methods and techniques. Hence this survey on load balancing is aimed at providing a comprehensive knowledge to future researchers that could be immense help for them in designing innovative algorithms and strategies pertaining to load balancing in cloud. An in-depth analysis of recent literature and various state of the art mechanisms had been studied and we had presented in this survey, the existing techniques applied in load balancing, their features and thrown light on their pros and cons.

The remaining of the article had been structured as: In Section 2 we had presented a comprehensive review on the models, metrics, policies and categorization of load balancing algorithms. Review of existing load balancing algorithms and their classification is presented in Section 3. Section 4 presents the conclusion of our survey where we had provided future topics of interests related to load balancing.

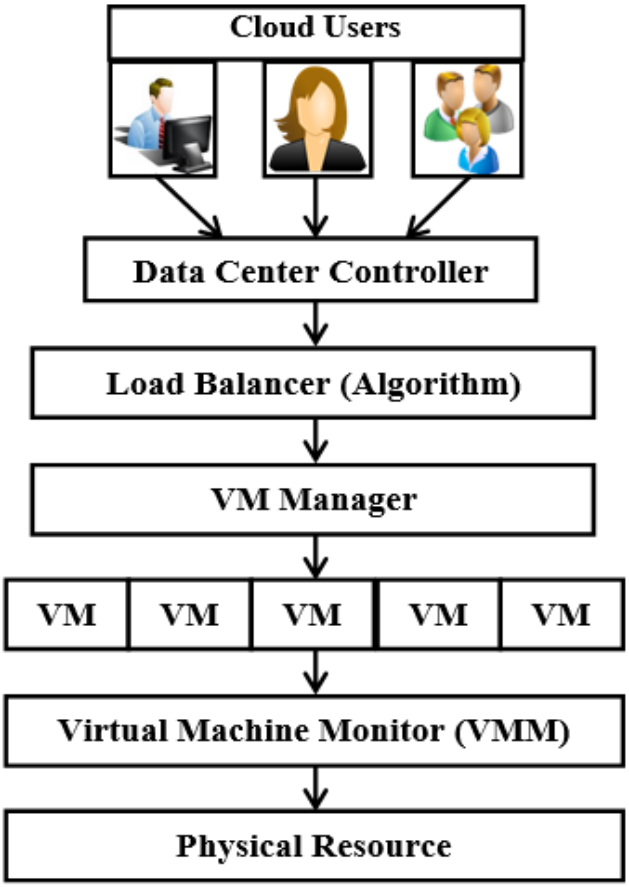
2. Load Balancing Model, Challenges, Classifications and Metrics

2.1. Load Balancing Model

Gupta et al. [21] had presented a model for load balancing approach that had been presented in the figure 2 below. It could be noticed that once the Load Balancer is supplied with requests from users, load balancing algorithms are executed for evenly distributing the load among the Virtual Machines (VMs). The decision to select the appropriate virtual machine to which the next request needs to be assigned is done by the Load Balancer module. The tasks that are received are managed by the Data Center Controller. Virtual machines (VMs) are controlled and managed by the VM manager. Virtualization is the predominant strategy applied in the cloud. The primary focus of virtualization is to share the expensive hardware resources amidst the available VMs. A Virtual

Machine is actually a software emulation of a computer system upon which the operating system and applications can be executed. User requests are processed at the VMs. Worldwide users submit their requests to the cloud in a random manner. These requests should be delivered to the VMs to get processed. Assignment of tasks too is a significant issue in cloud computing environment. A hypervisor or the Virtual Machine Manager (VMM) had been assigned the responsibility of creation and management of VMs. The VMM performs four operations namely multiplexing, storage, resumption and life migration. When a few VMs are still idle or under-loaded, increasing the load on a few of them by continuously assigning tasks to them is unfair and may result in deterioration of QoS parameters. Deterioration of QoS make the users unsatisfied may even make them to ignore such service providers in future. The VMs needs to be efficiently managed and assigned with manageable load.

Figure 2: The Cloud Load Balancing Model



As a pre-requisite, the challenges and other issues that impact the performance of load balancing algorithms in cloud need to be discussed. These challenges should be addressed by the

researchers' before-hand while coming up with an optimal solution to the load balancing issue in cloud. The summary of these challenges had been presented below:

2.2. Load Balancing Metrics

Load balancing deals with the distribution the dynamic workload equally among all available virtual machines. If handled efficiently load balancing guarantees to achieve better resource utilization, thereby increasing the user satisfaction. Also the management of resources is done in an optimal way such each and every resource gets a fair treatment. Ultimate result is minimization of resource consumption, increasing the scalability aspect of cloud operation, evading out formation of bottle-necks, appropriate provisioning etc. The primary quality metrics applied for balancing of load are described below:

Response time: Gives an account of the time taken for a system to respond.

Scalability: It is a qualitative feature of the load balancing algorithm that highlights the capability to achieve a balanced load with limited optimal number of resources or machines.

Resource utilization: It is the measure of optimal utilization of the available system resources. An efficient load balancing algorithm should strive to provide maximum resource utilization.

Throughput: It represents the measure of quantity of data sent or received by any specific node. It refers to the count of number of nodes whose state gets transitioned to 'complete state' in a specified time interval. To achieve better cloud performance, the throughput factor should be maximized.

Migration time: It is the time expended in shifting a task from one machine to another machine. This migration time measure needs to be minimal for achieving better performance of the cloud system.

Make span: It is an account of total time taken to complete the submitted tasks or the measure of time taken to allocate the resources to the user tasks.

Fault tolerance: It is the capability of the system to overcome any arbitrary failure resulting due to unavoidable problems in the network and provide overall uniform performance

Degree of imbalance: It is the measure of imbalance that is assessed for any virtual machine, caused due to improper allocation of load.

Performance: It is the qualitative measure of how effectively the system performs after carrying out the load balancing activity.

The performance of the entire system gets highly improved once the aforesaid parameters are satisfied in an optimal manner.

2.3. Load Balancing Challenges

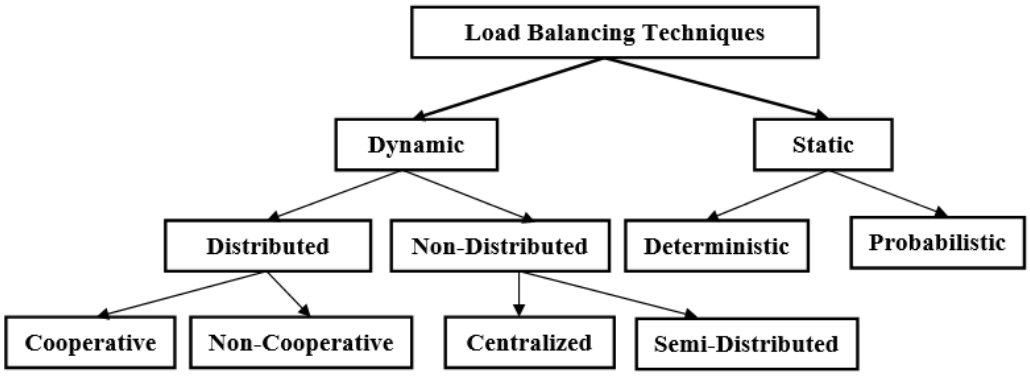
The intrinsic challenges in cloud environment should be appropriately addressed to realize the actual benefits of load balancing algorithms. These challenges are herewith enlisted:

1. Geographical/spatial distributions of nodes: Designing a load balancing algorithm for a cloud system whose resources are spatially or geographically spread out is the toughest task. The spatial distance between the nodes is inversely proportional to the speed of data transfer, thus having an adverse impact on the throughput factor.
2. Algorithmic Complexity: Complexity and performance are inversely proportional. The design of the load balancing algorithm should be less complicated. Increased complexity increases the delay which further worsens the performance.
3. Point of failure: The design should prevent the presence of single point of failure in the algorithm.
4. Static load balancing algorithm: Static load balancing algorithms could only be applied for uniform loads. They are not capable to adapt for dynamic load.

2.4. Load Balancing Classifications

The figure 3 given below presents the general classification of load balancing techniques based on the strategy they adopt.

Figure 3- Classification of Load Balancing Techniques



Received: 28.02.2021 – Accepted: 06.04.2021

3. Review of Load Balancing Techniques

Load in the cloud environment could be balanced by installing virtual machines. Any active virtual machine from a physical host can be transferred to another host. Ramezani et al. [22] analyzed the Particle Swarm Optimization algorithm and had proposed a task based load balancing algorithm. The authors had not suggested transferring the entire load from an overloaded virtual machine and instead the extra load alone could be migrated to another virtual machine.

Dhinesh et al. [23] had proposed an algorithm for balancing load across the virtual machines in the cloud. The authors had got inspiration for their algorithm from the behavior of honey bees. The authors had compared the tasks which are supposed to be balanced as the bees, the virtual machines were compared to that of sources of food and the bees were destined towards the machines with lesser loads. The process of loading a task to any specific virtual machine is analogous to the process of a bee that looks out for a food source. If the virtual machines get overloaded, forthcoming tasks were directed towards under-loaded virtual machines. Once a task gets completed on any VM, the remaining tasks are updated accordingly.

Ramesh et al. [24] had analyzed the honey bee characteristics, to be specific, their food searching behavior and had compared it with the assignment of load across virtual machines in cloud. The additional tasks from an overloaded virtual machine could be transferred to any suitable virtual machine that had lesser load comparatively. In their proposed approach, the authors had compared the diverted tasks from any over loaded machine to that of the honey bees and the lightly loaded virtual machines are compared to sources of food. The authors had proposed their strategy in four steps. Those four steps are: calculation of the current load on any virtual machine, load balancing and arriving to a decision regarding the scheduling of tasks, grouping of virtual machines and finally the scheduling of tasks. Experimental results accounted for reduction of make span and virtual machine migrations. Also there were improvements in the QoS delivered to the customers.

Yakhchiet al. [25] analyzed the behavior of the cuckoo bird and had proposed the Cuckoo Optimization Algorithm (COA) for balancing load in cloud computing. The technique proposed by the authors accounted for energy savings. The authors had applied the minimum migration time policy for choosing virtual machines that were lightly loaded whenever any virtual gets overloaded. Simulation results had shown that the proposed algorithm accounted for reduced energy consumption. On the flip side, the proposed algorithm caused SLA violations.

Keshvadi et al. [26] had proposed a multi-agent load balancing strategy to be applied for an IaaS cloud environment. The proposed strategy integrated both the receiver initiated as well as sender

initiated approaches for balancing the load in IaaS environment. This approach yielded minimum task waiting time and was compliant to the Service Level Agreement (SLA) made. The virtual machine manager agent (VMM) tracked the load on any VM by maintaining a note of CPU, memory and bandwidth utilized by any virtual machine while servicing any received task. A table is built and maintained by the VMM agent for storing the state of each virtual machine. The datacenter manager agent (DM), executes the information policy at each datacenter based on the monitored information at the VMM.

Jena et al. [27] had proposed an innovative technique named QMPSO, for dynamic balancing of the load among virtual machines in the cloud. The QMPSO is a hybridized technique obtained by integrating the Particle Swarm Optimization and Q-learning algorithms. Hybridization results in improvement in performance of virtual machines by dynamically balancing their load, maximizing their throughput and obtaining a balance among the tasks by considering their priorities and optimizing their waiting times.

Semmoud et al. [28] had proposed an innovative algorithm for balancing load that was centered on adaptive starvation. The proposed algorithm focused on balancing the load among server machines and accomplished minimized response time, maximized the rate of utilization of servers, reduction of migration costs and accounted for system stability. Experimental results obtained had shown that the proposed algorithm produced notable performance gains and considerably reduced the number of migrations.

Neelima et al. [29] had the Adaptive Dragonfly algorithm (ADA) for balancing the load and scheduling the tasks in the cloud environment. The proposed algorithm is a hybrid product, designed by integrating the dragon fly and firefly algorithms. The proposed ADA algorithm had employed a multi-objective function for attaining better performance. The multi-objective function included three parameters namely make span, cost and load.

Shadab et al. [30] had proposed the QPSL Queuing Model for load balancing in the cloud computing environment. The proposed model was based on M/M/k queue by emulating the FIFO operating standard. The proposed model assessed the service rates and waiting times by utilizing the Exponential distribution and Poisson distribution respectively. Experimental results obtained had shown that the proposed QPSL model fared well with respect to service rate and response time.

Devaraj et al. [31] had proposed the Firefly Improved Multi-Objective Particle Swarm Optimization algorithm (FIMPSO) for balancing the load in the cloud that was designed by hybridizing the Firefly algorithm and Improved Multi-Objective Particle Swarm Optimization

(IMPSO) algorithm. The concept to minimize the search space and to obtain an improved response had been replicated from Firefly algorithm and IMPSO algorithm respectively. The proposed algorithm generated an average load effectively and accounted for the improvement of resource utilization rate and task response time. Experimental results showed that the proposed FIMPSO yielded maximum CPU utilization and memory utilization rates, throughput and make span.

Hung et al. [32] had analyzed the Max-Min scheduling algorithm and had come up with its improvement called the MMSIA algorithm. In the improved MMSIA algorithm, the completion time of requests were faster as the authors had made use of the machine learning strategy called "learned learning" and by forming clusters based on the request sizes and clusters of virtual machines based on the rate of utilization of each virtual machine. Then the algorithm selected the largest request cluster and assigned it to the virtual machine that had a least utilization rate. Such selection and assigning strategy was repeated whenever the request list had become empty. Improvement was achieved in the completion time of tasks using the proposed MMSIA algorithm.

Amrita et al. [33] had proposed the Multi-agent Deep Reinforcement Learning-Dynamic Resource Allocation (MADRL-DRA) algorithm for deploying in the Local User Agent (LUA). The proposed algorithm could be applied for predicting the environmental set up required for user tasks and accordingly allocated them to appropriate virtual machines by considering the task priority. Subsequently the virtual machines were assessed for their load. The proposed approach increased the throughput and accounted for reduction in the response time with respect to the allocation of resources to tasks. The Dynamic Optimal Load-Aware Service Broker (DOLASB) was deployed at the Global User Agent (GUA) for effective scheduling of tasks. The availability of cloud brokers (CBs) was made known using which the required services were provided to user tasks. The proposed (MADRL-DRA) yielded better results with respect to execution time, waiting time, throughput, resource utilization, energy consumption and make span.

Ashok et al. [34] had integrated the Crow Search algorithm, Dragon Fly algorithm and Fractional Calculus and proposed a hybridized algorithm named Crow search with the integrated Fractional Dragonfly Algorithm (C-FDLA) for load balancing in the cloud environment. Crow Search algorithm, Dragon Fly algorithm and Fractional Calculus were integrated to design the hybrid C-FDLA algorithm. The authors had designed a multi-objective model based on selection probabilities and a frequency scaling method based on machine capacity for assessing the data length for tasks. Different type of cloud scenarios were developed for analyzing the performance of the

proposed C-FDLA and it produced a considerable improvement in performance by minimized the load.

Table 1- Summary of Literature Review

Author	Algorithm	Parameters Considered	Simulation Tool
Ramezani et al[22]	TBSLB-PSO	Load Balancing, Execution Time, Task Transfer Time	Cloudsim
Dhinesh et al[23]	HBB-LB	Load Balancing, Waiting time, Throughput	Cloudsim
Remesh et al[24]	BCOA	Load Balancing, Makespan, VM Migration	Cloudsim
Yakhchi et al[25]	COA-MMT	Load Balancing, Migration Time	Cloudsim
Sina et al[26]	MALBA	Load-Balance, Response Time, Makespan	Cloudsim
Jena et al[27]	MPSO, QMPSO	Load Balancing, Waiting time, Throughput	Cloudsim
Abderraziq et al[28]	DLBA	Load Balancing, Response Time, Utilization Rate, Cost, Stability	Cloudsim
Neelima et al[29]	ADFOA	Load Balancing, Execution Cost, Execution Time	CloudSim
Shadab et al[30]	QPSL	Load Balancing, Availability, Response Time	Matlab
Francis et al[31]	FIMPSO	Load Balancing, Resource Utilization, Response Time, Throughput, Makespan	Matlab
Hung et al[32]	MMSIA	Load Balancing, Completion Time	CloudSim
Amrita et al[33]	MADRL-DRA, DOLASB, BD-MIP	Makespan, Waiting Time, Execution Time, Energy Efficiency, Throughput, Resource Utilization	CloudSim
Ashok Kumar et al [34]	C-FDLA	Load Balancing, Migration Cost	JAVA
Jean et al[35]	IBPSO-LBS	Load Balancing, Completion Time, Cost	CloudSim
Mohanty et al[36]	MPSO	Load Balancing, Makespan, Resource Utilization	CloudSim

Jean et al. [35] had proposed an efficient binary form of Particle Swarm Optimization algorithm that could be applied for task scheduling and balancing the load in cloud computing environment. The proposed algorithm accounted for low time complexity and was highly economical. The authors had defined an objective function that calculated the difference in the maximum completion time between heterogeneous virtual machines based on the constraints defined for updation and optimization. Their proposed algorithm achieved considerable performance related to scheduling of tasks and load balancing.

Mohanty et al. [36] had analyzed the Particle Swarm Optimization algorithm and designed a meta-heuristic load balancing algorithm named Modified PSO algorithm. Benefits of the original PSO algorithm were replicated in the design of the MPSO. The proposed algorithm accounted for

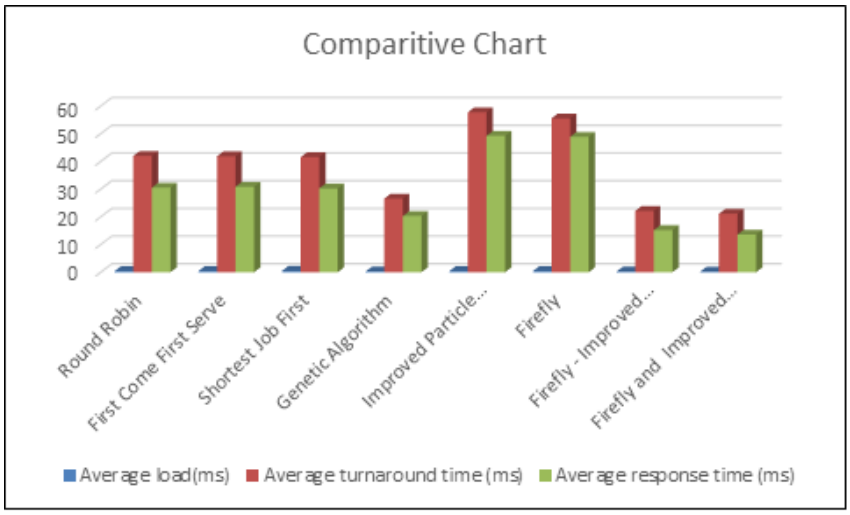
minimization of task overhead and maximization of resource utilization rate. The authors had analyzed the efficiency of their proposed algorithm under different cloud configurations by varying the count of virtual machines and cloudlets.

Table 2 explains the Comparison between average load, average turnaround time and average response time for different scheduling methods. Figure 4 explains the performance of different scheduling method.

Table 2- Average Load, Average Turnaround Time and Average Response Time for different Scheduling Methods. [31]

Methods	**Average load(ms)**	**Average turnaround time (ms)**	**Average response time (ms)**
Round Robin	0.430	41.98	30.50
First Come First Serve	0.460	41.87	30.84
Shortest Job First	0.495	41.56	30.24
Genetic Algorithm	0.310	26.57	20.30
Improved Particle Swarm Optimization	0.457	57.74	49.23
Firefly	0.470	55.54	48.87
Firefly-Improved Particle Swarm Optimization	0.259	22.13	15.21
Firefly and Improved Multi-Objective Particle Swarm Optimization	0.247	21.09	13.58

Figure 4- Performance Comparison



4. Conclusion

One of the major challenges being faced by the cloud environment is to provide an evenly balanced workload for the cloud nodes. In this article we had presented a survey of literatures pertaining to the load balancing phenomenon in cloud. Several metrics that hold key to load balancing had been listed and discussed which will be of immense help to future researchers in the load balancing arena. In this survey we had presented the key ideas, algorithms, metrics, advantages, disadvantages that various authors had considered in their respective proposals. Presently research on load balancing had been focusing upon two key QoS criterions namely energy consumption and carbon emission. As a part of future research guideline for researchers, we suggest them to study and analyze the recently proposed approaches in load balancing domain and evaluate them using any simulation toolkit and subsequently compare their performance by considering new QoS metrics.

References

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems, 25*(6), 599-616.

Ergu, D., Kou, G., Peng, Y., Shi, Y., & Shi, Y. (2013). The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment. *The Journal of Supercomputing, 64*(3), 835-848.

Zhang, Q., Cheng, L., & Boutaba, R., (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications, 1*(1), 7-18.

Milani, A.S., & Navimipour, N.J. (2016). Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends. *Journal of Network and Computer Applications, 71*, 86-98.

Albert, P., & Nanjappan, M. (2020). An Efficient Kernel FCM and Artificial Fish Swarm Optimization Based Optimal Resource Allocation in Cloud. *Journal of Circuits, Systems and Computers*. https://doi.org/10.1142/S0218126620502539

Ghomi, E.J., Rahmani, A.M., & Qader, N.N. (2017). Load-balancing algorithms in cloud computing: A survey. *Journal of Network and Computer Applications, 88*, 50-71.

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications, 1*(1), 7-18.

Gobalakrishnan, N., & Arun, C. (2018). A new multi-objective optimal programming model for task scheduling using genetic gray wolf optimization in cloud computing. *The Computer Journal, 61*(10), 1523-1536.

Manikandan, N., & Pravin, A. (2019). Hybrid resource allocation and task scheduling scheme in cloud computing using optimal clustering techniques. *International Journal of Services Operations and Informatics, 10*(2), 104-121.

Pradeep, K., & Jacob, T.P. (2018). CGSA scheduler: A multi-objective-based hybrid approach for task scheduling in cloud environment. *Information Security Journal: A Global Perspective, 27*(2), 77-91.

Pradeep, K., & Jacob, T.P. (2018). A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. *Wireless Personal Communications, 101*(4), 2287-2311.

Natesan, G., & Chokkalingam, A. (2019). Optimal task scheduling in the cloud environment using a mean grey wolf optimization algorithm. *International Journal of Technology, 10*(1), 126-136.

Nanjappan, M., & Albert, P. (2019). Hybrid-based novel approach for resource scheduling using MCFCM and PSO in cloud computing environment. *Concurrency and Computation: Practice and Experience*, e5517.

Natesan, G., & Chokkalingam, A. (2019). Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. *ICT Express, 5*(2), 110-114.

Krishnadoss, P., & Jacob, P. (2019). OLOA: Based task scheduling in heterogeneous clouds. *International Journal of Intelligent Engineering and Systems, 12*(1), 114-122.

Manikandan, N., & Pravin, A. (2018). An efficient improved weighted round Robin load balancing algorithm in cloud computing. *International Journal of Engineering and Technology (UAE), 7*(3.1), 110-117.

Manikandan, N., & Pravin, A. (2019). LGSA: Hybrid task scheduling in multi objective functionality in cloud computing environment. *3D Research, 10*(2), 1-16.

https://doi.org/10.1007/s13319-019-0222-2

Natesan, G., & Chokkalingam, A. (2020). An improved grey wolf optimization algorithm based task scheduling in cloud computing environment. *International Arab Journal of Information Technology*, 17(1), 73-81.

Natesan, G., & Chokkalingam, A. (2020). Multi-Objective Task Scheduling Using Hybrid Whale Genetic Optimization Algorithm in Heterogeneous Computing Environment. *Wireless Personal Communications, 110*(4), 1887-1913.

Jacob, T.P., & Pradeep, K. (2019). A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization. *Wireless Personal Communications, 109*(1), 315-331.

Gupta, H., & Sahu, K. (2014). Honey bee behavior based load balancing of tasks in cloud computing. *International journal of Science and Research, 3*(6).

Ramezani, F., Lu, J., & Hussain, F.K. (2014). Task-based system load balancing in cloud computing using particle swarm optimization. *International journal of parallel programming, 42*(5), 739-754.

LD, D.B., & Krishna, P.V. (2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied soft computing, 13*(5), 2292-2303.

Babu, K.R., & Samuel, P. (2016). Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud. *In Innovations in bio-inspired computing and applications,* 67-78.

Yakhchi, M., Ghafari, S.M., Yakhchi, S., Fazeli, M., & Patooghi, A. (2015). Proposing a load balancing method based on Cuckoo Optimization Algorithm for energy management in cloud computing infrastructures. *In 2015 6th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO),* 1-5.

Keshvadi, S., & Faghih, B. (2016). A multi-agent based load balancing system in IaaS cloud environment. *International Robotics & Automation Journal, 1*(1), 1-6.

Jena, U.K., Das, P.K., & Kabat, M.R. (2020). Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. *Journal of King Saud University-Computer and Information Sciences, 1,* 514-521.

Semmoud, A., Hakem, M., Benmammar, B., & Charr, J.C. (2020). Load balancing in cloud computing environments based on adaptive starvation threshold. *Concurrency and Computation: Practice and Experience, 32*(11), e5652.

Neelima, P., & Reddy, A.R.M. (2020). An efficient load balancing system using adaptive dragonfly algorithm in cloud computing. *Cluster Computing,* 1-9.

Siddiqui, S., Darbari, M., & Yagyasen, D. (2020). An QPSL Queuing Model for Load Balancing in Cloud Computing. *International Journal of e-Collaboration (IJeC), 16*(3), 33-48.

Devaraj, A.F.S., Elhoseny, M., Dhanasekaran, S., Lydia, E.L., & Shankar, K. (2020). Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments. *Journal of Parallel and Distributed Computing, 142,* 36-45.

Hung, T.C., Hieu, L.N., Hy, P.T., & Phi, N.X. (2019). MMSIA: improved max-min scheduling algorithm for load balancing on cloud computing. *In Proceedings of the 3rd International Conference on Machine Learning and Soft Computing,* 60-64.

Jyoti, A., & Shrimali, M. (2020). Dynamic provisioning of resources based on load balancing and service broker policy in cloud computing. *Cluster Computing, 23*(1), 377-395.

Kumar, C.A., & Vimala, R. (2019). C-FDLA: Crow search with integrated fractional dragonfly algorithm for load balancing in cloud computing environments. *Journal of Circuits, Systems and Computers, 28*(07), 1950115.

Mapetu, J.P.B., Chen, Z., & Kong, L. (2019). Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing. *Applied Intelligence, 49*(9), 3308-3330.

Mohanty, S., Patra, P.K., Ray, M., & Mohapatra, S. (2018). A novel meta-heuristic approach for load balancing in cloud computing. *International Journal of Knowledge-Based Organization, 8*(1), 2018.