A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment

Mansi Kaushik
Department of Computer Science and Applications
Kurukshetra University
Kurukshetra
dcsa2123mansikaushik@kuk.ac.in

Rajender Nath
Professor, Department of Computer Science and Applications
Kurukshetra University
Kurukshetra
rnath@kuk.ac.in

***Abstract*— Load balancing is an important component in cloud computing that ensures of optimal resource distribution, improved system performance, and high availability. As cloud environments experience fluctuating user demands and dynamic workloads, effective load balancing techniques become essential to maintaining environment's scalability, reliability, and energy efficiency. This paper presents a comparative study of load balancing algorithms- static, dynamic, and hybrid. Forty-Nine research papers focused on load balancing are collected from various resources systematically analyzed and evaluated these algorithms based on key parameters- Load Distribution, Resource Allocation Efficiency, Adaptability, Scalability, Fault Tolerance, Overhead Costs, Energy Efficiency. This paper also highlighting researchers' key points, methodologies, and evaluation criteria given by them over the past decade. It has been found that Hybrid algorithms are the best choice for systems. Adaptability, fault tolerance, and energy efficiency of hybrid algorithms consistently rank high across these metrics, making them ideal for modern, distributed systems with dynamic workloads.**

I. Introduction

Load Balancing is the process of distributing workloads evenly across multiple computing resources (e.g., servers, networks, or virtual machines) to ensure optimal resource utilization, reduce latency, improve system performance, and avoid overloading a single resource. Load balancing can be implemented in hardware or software and is widely used in high-availability systems, data centers, and cloud environments. Without effective load balancing, cloud systems can experience bottlenecks, increased latency, degraded performance, and even system failures.

As cloud computing infrastructures evolve, the need for robust, scalable, and dynamic load balancing solutions has grown significantly. Over the past decade, many algorithms have been proposed to address these challenges. To guide future research and development, many research papers have been published, analyzing and comparing these algorithms across various criteria such as efficiency, scalability, energy consumption, and fault tolerance. Load balancing algorithms are broadly classified into three categories – Static, Dynamic, Hybrid

(a) Static- These algorithms distribute tasks using predefined rules that do not consider real-time server load. For Instance- Round Robin, Weighted Round Robin are static load balancing algorithms.

(b) Dynamic- These algorithms adjust requests distribution in real-time based on server performance and workload. For Instance - Least Connections, GA-Based Dynamic Load Balancing, Dynamic Load Balancing Ant Colony Optimization (DLBACO), Decentralized Dynamic are dynamic algorithms.

(c) Hybrid- Hybrid algorithms combine static and dynamic techniques for enhanced performance. For Instance- GA-Based Hybrid Load Balancing, Firefly and Improved PSO, Hybrid BFOA- PSO are hybrid algorithms.

II. Related Work

An integrated framework was proposed [1] for evaluating the performance & trustworthiness of various load balancing algorithms. A macro-level study of load balancing in cloud computing [2] was carried based on operational principles, and performance metrics such as latency reduction, energy efficiency, and real-time resource management. A Review of static, dynamic, and AI-based approaches. Authors discussed [17] emerging trends like serverless and edge computing, emphasizing adaptive algorithms for heterogeneous workloads. Authors recommended [30] hybrid approaches combining static and dynamic methods. Suggested machine learning for predictive load management. Enhancing Quality of Service (QoS) through load balancing. Authors highlighted [31] the necessity of dynamic approaches and fault-tolerance mechanisms for handling fluctuating workloads. Authors categorized [27] algorithms into traditional, heuristic, and AI-based models. Emphasized real-time adaptability and future applications in edge computing. Authors focused [37] on Challenges like server congestion and resource wastage. Advocated integrating predictive mechanisms with existing models for fault-tolerant and self-adaptive systems. The paper emphasizes the need for fault-tolerant and self-adaptive models. Authors focused [39] on Enhancing cloud performance and user satisfaction. They suggested AI-based predictive models to improve traditional algorithms. Authors focused [40] on highlighted the flexibility of software solutions and advocated for AI-driven load balancing in future deployments.

The comparative study of load balancing algorithms emphasized the role of load balancing in cloud computing to ensure efficient resource utilization, reduced latency, and enhanced system performance. Various load-balancing algorithms offer unique advantages depending on workload characteristics and cloud environments.

Inspired by the review of existing load-balancing algorithms and techniques, a novel comparative analysis of load-balancing is presented in this paper. Our analysis evaluates these techniques based on critical parameters, including Load Distribution, Resource Allocation Efficiency, Adaptability, Scalability, Fault Tolerance, Overhead Costs, and Energy Efficiency. This highlights the importance of using flexible and hybrid approaches that integrate real-time performance data and predictive models to improve resource allocation while reducing costs and saving energy.

III. COMPARATIVE STUDY OF LOAD BALANCING ALGORITHMS

Over the past couple of decades, much research has been done in the field of Cloud Computing and Load Balancing. For this Literature downloaded Research papers from different trusted sites such as IEEE, ScienceDirect, ResearchGate, Springer. Later, in this paper discussing the comparative analysis of collected Research papers from 2013 to 2024 on Load balancing algorithm paper focusing on key parameters.

This paper intends to compare different load balancing algorithms in different categories. For the study 49 research paper published from 2013 to 2024 relating to load balancing in cloud computing environment were collected and these were categorized based on type of load balancing algorithm viz are static, dynamic, hybrid. Key parameters - Load Distribution, Resource Allocation Efficiency, Adaptability, Scalability, Fault Tolerance, Overhead Costs, Energy Efficiency were identified, and algorithms were category wise on these parameters. We aim to examine how the scope and focus of these papers have shifted over the time.

These papers are classified into following categories:

(a) *Static Load Balancing Algorithms*

(b) *Dynamic Load Balancing Algorithms*

(c) *Hybrid Load Balancing Algorithms*

IV. PARAMETERS DEFINITIONS & THEIR VALUE

To evaluate the load balancing algorithms Eight parameters are identified which are explained below:

(a) *Algorithm Used by Different Authors:* Round Robin (RR), Weighted Round Robin (WRR), Least Connections (LC), First Come First Serve (FCFS), Dynamic Weighted Least Connections, Dynamic Load Balancing Ant Colony Optimization (DLBACO), Hybrid BFOA-PSO, Bacterial Foraging and PSO, Firefly and Improved PSO, Hybrid PSO-SQP

(b) *Load Distribution:* Load Distribution refers to the process of distributing tasks or workloads across multiple resources (e.g., servers). The load is maintained at each resource in such a manner that neither it gets overloaded nor idle during the execution.

Nominal values for Load Distribution are taken as: Low, Moderate, High

(c) *Resource Allocation Efficiency:* Resource Allocation Efficiency measures how effectively system resources (e.g., CPU, memory, bandwidth) are utilized to handle workloads, minimizing idle or overloaded resources.

Nominal values for Resource Allocation Efficiency are taken as: Low, Moderate, High

(d) *Adaptability:* Adaptability refers to the ability of an algorithm to adjust to changing conditions, such as fluctuations in workload or resource availability.

Nominal values for Adaptability are taken as: Low, Moderate, High

(e) *Scalability:* Scalability refers to a system's ability to handle increased load by adding resources. As your user base grows or traffic spikes, you want your system to maintain its performance without downtime or degradation. Scalability ensures that the system can grow and continue functioning efficiently.

Nominal values for Scalability are taken as: Low, Moderate, High

(f) *Fault Tolerance:* Fault Tolerance is ability of a system to continue operating correctly even if one or more components fail.

Nominal values for Fault Tolerance are taken as: Low, Moderate, High

(g) *Overhead Costs:* Overhead Costs are additional computational resources or time required to manage the load balancing process, including monitoring, decision-making, and task migration.

Nominal values for Overhead Costs are taken as: Low, Moderate, High

(h) *Energy Efficiency:* Energy Efficiency measures how effectively a system uses energy resources to perform tasks, minimizing energy consumption while maintaining performance.

Nominal values for Energy Efficiency are taken as: Low, Moderate, High

V. EVALUATION OF STATIC ALGORITHMS

Static Algorithms proposed by the authors [12,19,24,32,42,45] are evaluated based on the parameters defined in the forgoing paragraph. As per criteria defined these algorithms are evaluated as shown in Table 1. This evaluation helps in understanding the strengths and weaknesses of these algorithms.

It is observed from Table 1, that the algorithms such as Round Robin (RR), Weighted Round Robin (WRR), and Least Connections (LC) generally show low to moderate load distribution. This is due to their inability to adapt dynamically to workload variations, leading to uneven task allocations. Resource allocation efficiency for static algorithms shows low to moderate. The static nature limits the ability to optimize resources dynamically, which result in under-utilization or overburdening of certain nodes. Adaptability of static algorithms demonstrated as low. They lack mechanisms to respond to real-time workload changes or dynamic environments, which is a significant limitation. These algorithms demonstrated low to moderate scalability. As the system grows, the performance of static algorithms degraded due to their inability to adjust to larger workloads. Fault tolerance is consistently low in static algorithms. Since these methods operate on predefined rules, they fail to handle system failures or node crashes effectively. Static algorithms experienced low to moderate overhead costs. This is one of static algorithm advantages, as these algorithm do not require complex computations or continuous monitoring. Energy efficiency is generally low to moderate in static algorithms. Without dynamic load adjustments, energy consumption cannot be optimized effectively, leading to inefficiencies in resource utilization.

TABLE I. COMPARATIVE ANALYSIS OF STATIC ALGORITHM

Authors & Year	Algorithm Discussed	Load Distribution	Resource Allocation Efficiency	Adaptability	Scalability	Fault Tolerance	Overhead Costs	Energy Efficiency
(Shafiq et al., 2021)	RR, WRR, LC	Moderate	Moderate	Low	Moderate	Low	Moderate	Moderate
(Shah & Farik, 2019)	RR, LC	Moderate	Moderate	Low	Moderate	Low	Moderate	Low
(Hota et al., 2018)	RR, FCFS, LC	Moderate	Moderate	Low	Low	Low	Moderate	Low
(Hamadah, 2017)	RR, LC	Low	Low	Low	Moderate	Low	Low	Moderate
(Kashyap & Viradiya, 2014)	RR, LC, WRR	Moderate	Moderate	Low	Low	Low	Moderate	Moderate
(Sidhu & Kinger, 2013)	RR, LC	Low	Moderate	Low	Moderate	Low	Moderate	Low

VI. EVALUATION OF DYNAMIC ALGORITHMS

Dynamic Algorithms proposed by authors [2,9,18,21,22,29,33,38] are evaluated based on the parameters defined in the forgoing paragraph. As per criteria defined theses algorithms are evaluated as shown in table 2. This evaluation helps in understanding the strengths, weaknesses of these algorithms.

It is observed from Table 2. that Dynamic algorithms generally perform better in load distribution compared to static algorithms. It achieved low to high load distribution. Dynamic algorithms excel in resource allocation efficiency, with most techniques achieving moderate to high efficiency. This is due to their ability to adaptively allocate resources based on real-time system conditions, enhancing system performance and utilization. Adaptability is a strong characteristic of dynamic algorithms, with demonstrating moderate to high adaptability. These algorithms can respond to fluctuating workloads, making them suitable for environments with unpredictable changes. Dynamic Algorithms showed a moderate to high level of scalability, making them well-suited for large-scale systems. Dynamic algorithms offered moderate to high fault tolerance. Their real-time adjustments contribute to maintaining system stability. The overhead costs of dynamic algorithms vary between low to moderate. Dynamic algorithms achieved moderate to high energy efficiency.

TABLE II. COMPARATIVE ANALYSIS OF DYANMIC ALGORITHMS

Authors & Year	Algorithm Discussed	Load Distribution	Resource Allocation Efficiency	Adaptability	Scalability	Fault Tolerance	Overhead Costs	Energy Efficiency
(Darakhshan Syed et al., 2024)	GA, PSO	Moderate	High	High	High	Moderate	Low	Moderate
(Vahid Mohammadian et al., 2022)	Replication, Checkpointing, Predictive techniques	High	High	Moderate	Moderate	High	High	Moderate
(Raghav & Vyas, 2019)	Dynamic Weighted Least Connections	Moderate	Moderate	High	High	Moderate	Moderate	Moderate
(Arvinda Kushwaha et al., 2019)	Dynamic Load Balancing Ant Colony Optimization (DLBACO)	High	High	High	Moderate	Moderate	Moderate	High
(Najme Mansouri et al., 2019)	PSO	Moderate	High	High	Moderate	Moderate	Moderate	High
(Joshi & Kumari, 2017)	PSO	Moderate	High	High	High	Moderate	Moderate	High
(A.S. Milani et al., 2016)	Decentralized Dynamic	Moderate	Moderate	High	High	High	Moderate	High
(K. Karthika & R.B.K. Kanakamb al, 2015)	Dynamic Weighted Least Connections	Low	Low	Moderate	Low	Moderate	Low	Moderate

VII. EVALUATION OF HYBRID ALGORITHMS

Hybrid Algorithms proposed by authors [5,11,14,15,20,27,36,48,49,] are evaluated based on the parameters defined in the forgoing paragraph. As per criteria defined theses algorithms are evaluated as shown in table 3. This evaluation helps to understand the strengths and weaknesses of these algorithms.

It is observed from Table 3 that Load distribution is consistently high, demonstrating an efficient and equally distribution of computational tasks across resources, reducing bottlenecks and improving system performance. Resource Allocation efficiency is consistently rated as high, suggested effective resource utilization and management within these algorithms, making them suitable for systems. Hybrid algorithms demonstrated moderate to high adaptability, showcasing their capacity to adjust to dynamic and complex scenarios effectively. Scalability demonstrated as moderate to high across hybrid algorithms, demonstrating their ability to handle increasing workloads or system expansion effectively. Fault tolerance ranges from moderate to high, indicating that these algorithms handle system failures reasonably well. Overhead costs vary between low to moderate, with most hybrid algorithms keeping additional computational overhead minimal. Energy efficiency is rated between moderate to high, indicating that these algorithms are generally energy-conscious, balancing performance with reduced energy consumption.

TABLE III. COMPARATIVE ANALYSIS OF HYBRID ALGORITHMS

Authors & Year	Algorithm Discussed	Load Distribution	Resource Allocation Efficiency	Adaptability	Scalability	Fault Tolerance	Overhead Costs	Energy Efficiency
Murugandam et al. (2023)	HHO-ACO	High	High	High	High	High	Low	High
Princess and Radhamani (2021)	Hybrid Meta-Heuristic Algorithm	High	High	High	High	Moderate	Moderate	High
Rani (2020)	Hybrid Task Scheduling	High	High	High	High	High	Low	High
Jena et al. (2020)	Meta-Heuristic Hybrid Algorithm	High	High	High	High	High	Low	High
Golchi et al. (2019)	Firefly and Improved PSO	High	High	High	High	Moderate	Moderate	High
Yazdi et al. (2017)	NS-HS-DE	High	High	High	High	High	Low	High
Domanal and Reddy (2015)	Hybrid Scheduling Algorithm	High	High	High	High	Moderate	Low	Moderate
Wei et al. (2013)	Mathematical Scheduling Models	High	High	Moderate	High	Moderate	Moderate	Moderate
Mohanty et al. (2013)	Hybrid BFOA-PSO	Moderate	High	High	Moderate	High	Low	High

VIII. FUTURE RESEARCH DIRECTIONS

The reviewed research papers on load balancing algorithms in cloud computing reveal several significant trends over the past decade. These trends highlight the evolution of techniques and emerging research directions driven by technological advancements and the growing complexity of cloud environments.

(a) *Shift Toward Hybrid Approaches:* Hybrid algorithms effectively combine the strengths of static and dynamic approaches. These algorithms demonstrate higher adaptability, fault tolerance, and energy efficiency, making them suitable for dynamic and heterogeneous cloud systems.

(b) *Increased Focus on Energy Efficiency:* With sustainability becoming a critical concern, researchers are emphasizing energy-efficient load balancing algorithms. Approaches leveraging hybrid models and real-time adjustments have proven effective in reducing energy consumption while maintaining system performance.

(c) *Rise of Metaheuristic Algorithms:* Researchers are exploring hybridization of metaheuristics to address specific challenges in load balancing, such as fault tolerance and real-time workload fluctuations.

(d) *Parameter-Specific Optimization:* Further research, can be taken toward parameter-specific evaluations of load balancing algorithms, focusing on metrics like response time, throughput, resource utilization, and overhead costs.

IX. CONCLUSION & FUTURE SCOPE

The findings reveal that static algorithms, have been found simple and low in computational overhead, but struggle with adaptability and scalability and hence making them less suitable for dynamic cloud environments. Dynamic algorithms excel in real-time adaptability and resource utilization, offering better fault tolerance and energy efficiency but often at the cost of higher overhead. Hybrid algorithms emerge as a balanced solution, combining the strengths of static and dynamic approaches. They demonstrate high performance across most parameters, making them ideal for complex, large-scale cloud systems.

The future of load balancing is moving towards to more efficient solutions. Energy-efficient algorithms are gaining importance to support sustainability while maintaining system performance. Metaheuristic algorithms are helping solve challenges like handling sudden workload changes. Researchers can also be focused on optimizing specific factors like response time, resource use, and costs, ensuring future algorithms meet various needs.

REFERENCES

[1] G. Ajay Kumar Dubey, Vimal Mishra, "Analysis of performance and trust in load balancing algorithm in cloud computing environment", International Journal of Advanced Intelligence Paradigms, Vol. 27, No. 2, pp 91-103,2024

[2] Darakhshan Syed, Ghulam Muhammad and Safdar Rizvi, "Systematic Review: Load Balancing in Cloud Computing by Using Metaheuristic Based Dynamic Algorithm" Intelligent Automation & Soft Computing, Vol.39, No.(3), pp 437-476, 2024

[3] Medha Kirti, Ashish Kumar Maurya, "Fault-tolerance approaches for distributed and cloud computing environments: A systematic review, taxonomy and future directions," Concurrency and Computation: Practice and Experience, 2024.

[4] Zohu, Lilhore, Tao Hai, "Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing," Journal of Cloud Computing, Springer, 2023.

[5] Sumathi Muruganandam, Vijayaraj Natarajan, Raja Soosaimarian Peter, & Rajkamal Murugesan (2023). HHO-ACO hybridized load balancing technique in cloud computing.

[6] Ranjit, Anjali, "Load balancing techniques in cloud platform: A systematic study," Proceedings of the International Conference on Cloud Computing and Big Data, 2023.

[7] Teoh et al., "Load balancing in predictive maintenance for Industry 4.0," Journal of Industrial and Production Engineering, Vol. 10, No. 3, pp. 120-135, 2023.

[8] Raghunadha Reddi Dornala, "An Advanced Multi-Model Cloud Services using Load Balancing Algorithms," Proceedings of the 2023 IEEE International Conference on Innovations in Computing and Renewable Energy (ICIRCA), pp. 579-586, 2023.

[9] Vahid Mohammadian, Nima Jafari Navimipour, Mehdi Hosseinzadeh, and Aso Darwesh, "Fault-Tolerant Load Balancing in Cloud Computing: A Systematic Literature Review," IEEE Access, Vol. 10, pp. 15087-15102, 2022.

[10] Thakur, Kaur, "Load balancing techniques for edge computing environments," ScienceDirect Journal of Computing Technologies, Vol. 12, No. 4, pp. 205-220, 2022.

[11] Poornima Princess GA, Radhamani AS . A hybrid meta-heuristic for optimal load balancing in cloud computing. Journal of Grid Computing, 19, 1–22,2021.

[12] Shafiq, Jhanji, Abdullah, "Load balancing techniques in cloud computing environment: A review," ScienceDirect Journal of Cloud Computing, 2021.

[13] A. Kumar, D.P. Chawla, "A Systematic Literature Review on Load Balancing Algorithms of Virtual Machines in a Cloud Computing Environment," SSRN, 2020.

[14] Sunita Rani PK . An efficient and scalable hybrid task scheduling approach for cloud environment. International Journal of Information Technology, 12, 1451–1457,2020.

[15] Jena, U.K., Das, P.K., & Kabat, M.R. Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. Journal of King Saud University – Computer and Information Sciences,2020.

[16] S.K. Mishra, B. Sahoo, P.P. Parida, "Load balancing in cloud computing: A big picture," Journal of King Saud University - Computer and Information Sciences, Vol. 34, No. 1, pp. 134-145, 2020. DOI: 10.1016/j.jksuci.2018.01.003.

[17] R.G.P.R.S., S. Duraisamy, "Survey on Load Balancing in Cloud Computing," Global Journal of Computer Science Technology B: Cloud and Distributed Systems, Vol. 19, No. 2, 2019. DOI: 10.22214/ijraset.2019.4479.

[18] Raghav, Y.Y., Vyas, V., "A comparative analysis of different load balancing algorithms on different parameters in cloud computing,"3rd International Conference on Recent Developments in Control, Automation and Power Engineering (RDCAPE), pp. 628–634, IEEE, 2019. DOI: 10.1109/RDCAPE47089.2019.8979122

[19] Shah, Farik, "Challenges in static load balancing algorithms," ResearchGate - Static Load Balancing Algorithms in Cloud Computing: Challenges & Solutions, 2019.

[20] Golchi, M.M., Saraeian, S., Heydari, M. (2019). A hybrid of firefly and improved particle swarm optimization algorithms for load balancing in cloud environments: Performance evaluation. Computer Networks, 162, 106860.

[21] Arvinda Kushwaha, Mohd Amjad, Arvind Kumar, "Dynamic Load Balancing Ant Colony Optimization (DLBACO) Algorithm for Task Scheduling in Cloud Environment," 2019.

[22] Najme Mansouri, Behnam Mohammad Hasani Zade, Mohammad Masoud Javidi, "Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory," Computers & Industrial Engineering, Vol. 127, pp. 95-108, 2019. DOI: 10.1016/j.cie.2019.03.006.

[23] Yadav, Jay Shankar Prasad, "A Review on Load Balancing Algorithms in Cloud Computing Environment," International Journal of Computer Sciences and Engineering (IJCSE), 2018.

[24] Hota, A., Mohapatra, S., Mohanty, S., "Survey of Different Load Balancing Approach-Based Algorithms in Cloud Computing: A Comprehensive Review," SpringerLink, 2018.

[25] Ali Akbar Neghabi, Nima Jafari Navimipour, Mehdi Hosseinzadeh, Ali Rezaee, "Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature," 2018.

[26] A. Thakur, M.S. Goraya, "A taxonomic survey on load balancing in cloud," ScienceDirect Journal of Computer Applications, 2017.

[27] J. Yazdi, Y. H. Choi, J. H. Kim (2017). Non-dominated sorting harmony search differential evolution (ns-hs-de): A hybrid algorithm for multi-objective design of water distribution networks. Water, 9(8), 587.

[28] M. Alam, Z. Ahmad Khan, "Issues and challenges of load balancing algorithm in cloud computing environment," Indian Journal of Science and Technology, Vol. 10, No. 25, pp. 1– 12, 2017. DOI: 10.17485/ijst/2017/v10i25/105688.

[29] S. Joshi, U. Kumari, "A Comprehensive Analysis of Existing Load Balancing Algorithms in Cloud Network," ResearchGate - Load Balancing Algorithms in Cloud Computing, 2017.

[30] A.B. Singh, S. Bhat, R. Raju, R. D'Souza (2017). Survey on Various Load Balancing Techniques in Cloud Computing.

[31] E. Jafarnejad Ghomi, A. Masoud Rahmani, N. Nasih Qader (2017). Load-balancing algorithms in cloud computing: A survey.

[32] S. Hamadah, 2017 - A survey, a comprehensive study of static, dynamic and hybrid load balancing algorithms Microsoft Word - 6.Load_Balancing _survey_FinalCopy.doc (zuj.edu.jo)

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

[34] K. Elngomi, Zahra Mohammed Khanfar (2016). A Comparative Study of Load Balancing Algorithms: A Review Paper.

[35] Goyal, S., Verma, M.K. (2016). Load Balancing Techniques in Cloud Computing Environment–A Review.

[36] Domanal, S.G., & Reddy, G.R.M. (2015). Load Balancing in Cloud Environment using a Novel Hybrid Scheduling Algorithm. IEEE International Conference on Cloud Computing in Emerging Markets, 37–42, 2015.

[37] D. Gabi, A. Samad, A. Zainal (2015). Systematic Review on Existing Load Balancing Techniques in Cloud Computing.

[38] K. Karthika, R.B.K. Kanakambal (2015). Load balancing algorithm review's in cloud environment servers in data centers.

[39] Aslam, S., Shah, M.A. (2015). Load Balancing Algorithms in Cloud Computing: A Survey of Modern Techniques. IEEE Conference Publication.

[40] M. Kemal, M. Prathap (2014). Load balancing techniques: Major challenge in Cloud Computing - a systematic review. IEEE Conference Publication.

[41] R. Azizi, H. Shoja (2014). A comparative survey on load balancing algorithms in cloud computing. IEEE Conference Publication.

[42] Dharmesh Kashyap, Jaydeep Viradiya (2014). A Survey Of Various Load Balancing Algorithms In Cloud Computing.

[43] Tushar Desai, Jignesh Prajapati (2013). A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing.

[44] Mayanka Katyal, Atul Mishra (2013). A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment.

[45] Amandeep Kaur Sidhu, Supriya Kinger (2013). Analysis of Load Balancing Techniques in Cloud Computing.

[46] Suriya Begum, Dr. Prashanth C.S.R (2013). Review of Load Balancing in Cloud Computing.

[47] Shanti Swaroop Moharana, Rajadeepan D. Ramesh & Digamber Powar . Analysis of Load Balancers in Cloud Computing. International Journal of Computer Science and Engineering (IJCSE), 2(2), 101–108, 2013.

[48] Mohanty, B., Panda, S., & Hotab, P. K. Hybrid BFOA–PSO algorithm for automatic generation control of linear and nonlinear interconnected power systems. Applied Soft Computing, 13, 4718–4730, 2013.