九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Innovative Analysis of Workload Balancing Algorithms for Fog-Cloud Networks: A Contemporary Perspective

Sohal, Asha
DCSA, Kurukshetra University

Kait, Ramesh
DCSA, Kurukshetra University

https://doi.org/10.5109/7160920

出版情報：Evergreen. 10 (4), pp.2353-2364, 2023-12. 九州大学グリーンテクノロジー研究教育センター
バージョン：
権利関係：Creative Commons Attribution 4.0 International

Innovative Analysis of Workload Balancing Algorithms for Fog-Cloud Networks: A Contemporary Perspective

Asha Sohal[1,*], Ramesh Kait[2]
[1]DCSA, Kurukshetra University, Kurukshetra, Haryana
[2]DCSA, Kurukshetra University, Kurukshetra, Haryana

*Author to whom correspondence should be addressed:
E-mail: asha.sohal24@gmail.com

(Received July 31, 2023; Revised October 3, 2023; accepted October 6, 2023).

Abstract: Cloud technology is increasingly widespread as thriving businesses and research organizations seek to leverage its on-demand access, service models, and deployment patterns. However, further enhancements are still required to achieve optimal performance within this field. One challenging aspect is load balancing, specifically the equilibrium distribution of workload across virtual machines, which is a computationally complex problem. The present work extensively reviewed the state-of-the-art methods for load balancing in the cloud, encompassing traditional techniques, heuristics, meta-heuristics, and hybrid approaches. This paper provides a comprehensive historical assessment and comparative analysis of the prominent literature on load balancing, serving as a valuable resource for researchers aiming to develop new and effective load-balancing algorithms in the domain of fog-cloud networks. In this study, a thorough historical evaluation and comparative research of load-balancing literature can provide important insights into the development, efficacy, and adaptability of load-balancing strategies in a variety of scenarios. There are many fascinating directions that researchers in this area might go in, addressing both the past developments and the upcoming difficulties of load balancing in the dynamic environment of computing and networking.

Keywords: Carbon Emission; Meta-heuristic; Workload Balancing Algorithms; OoS; Fog-Cloud Networks

1. Introduction

The concept of several types of fog-cloud computing is transformed into one that includes numerous cloud services that are provided as utilities, such as energy, water, and telephone. Various forms of distributed computing paradigms, including mainframe, cluster, and grid computing, have emerged to fulfill the vision of utility computing.

Due to the widespread use of distributed computing, an organization must store a massive amount of data and efficiently retrieve it. Hence, there is a need for a computing system that not only provides services but also caters to the diverse requirements of customers across various domains simultaneously. In terms of utility, the term used to describe this concept is known as cloud computing and the gear can send real-time data to the cloud, allowing printing processes to be remotely monitored1). According to this concept, users can use cloud services as a "pay-per-use" method is available for the customers. This approach is backed by many data centers that combine and effectively employ resources by utilizing virtualization breakthroughs. In a cloud computing model, customers subscribe to the desired services and enter into a Service Level Agreement (SLA) with the cloud vendor. The SLA specifies the Quality of Service (QoS) and establishes other conditions governing the provision of the service. Aprilliani2) clearly expresses that the co-precipitation process was used to create a carbon-Fe3O4 composite, resulting in AC-M (Activated Carbon-Mixture), which are the parts of OoS in load balancing. To improve the load balancing process, this type of composite material could be employed in hardware components like sensors, transmitters, or network equipment and the energy efficiency of network components could be improved by using co-precipitated materials with features. The load balancing approach may be affected by reduced energy use because it will make resource allocation more effective. The popularity of fog-cloud applications among customers leads to a significant increase in demand for resources such as cores, storage systems, requirements in hardware, requirements in software, high bandwidth, and many more. Also, this results in a decrease in the makespan in the cloud, which increases efficiency in resource utilization. In order to maintain a strong service resource starvation should be kept to a minimum and use IoT & AI domains in hybrid methods as this may result in overheads3). In accordance with some policies and scheduling algorithms, tasks are distributed to VMs. To solve these problems in this situation, more effective and viable load balancing algorithms should be developed. The structure of the study is organized as follows Section 2 illustrates fundamental concepts related to workload balancing within fog-cloud networks. Section 3 covers the historical analysis of load balancing as per studies. Section 4 presents load balancing algorithms taxonomy which is used in cloud computing, focusing on each category, and conducting an analysis. Section 5 provides a summary of existing studies in the context of the suggested approach. Section 6 outlines the findings derived from the survey. Section 7 discusses future directions and offers concluding thoughts. The novelty of this analysis lies in its forward-looking, cutting-edge approach to a critical aspect of fog-cloud computing, which is increasingly important in the modern computing landscape. By putting these tactics into practice, Researchers can increase the uniqueness and applicability of your study in comparison to previous efforts, significantly advancing the topic of workload balancing in fog-cloud networks. This study gives practical insights for those wishing to install or enhance workload balancing in fog-cloud networks and illustrates the relevance and application of load balancing algorithms in the real world.

2. Balance of Loads Required

The portion of various tasks that are assigned to the VM in a cloud computing environment is referred to as the load. The cloud computing system can classify loads as underloaded, overloaded, or balanced. Load balancing algorithms aim to enhance system throughput by transparently transferring workloads from heavily loaded nodes to less burdened nodes through cloud migration. This approach ensures a more balanced distribution of system loads. A crucial component of cloud assignment planning is the load balancing of jobs that may be reliant on or independent of virtual machines (VMs)4).

2.1 Balancing loads QoS Measures

In order to assess the efficiency of various load balancing methods, several fundamental load balancing measures must be used. The following list of performance measures in the cloud computing environment, also known as QoS (Quality of Service) metrics have an impact on load balancing.

Throughput: The number of processes or user requests (tasks) that a virtual machine can execute successfully in each amount of time. A high throughput translates into better performance.

Response Time: This is the amount of time it takes for a task to start responding once it has been sent to a virtual machine. So, this time should be shorter to attain a larger performance. The ability of a system to provide continuous and consistent service even when one or more arbitrary nodes fail is known as fault tolerance.

Migration time: The duration required to migrate a virtual machine or task from one physical computer to another is known as migration time. This migration can occur between hosts or even across data centers. Minimizing this time is crucial to achieving effective load balancing outcomes. The degree of imbalance among virtual machines is quantified and evaluated.

Power Consumption: This refers to the amount of energy consumed by the equipment used in cloud computing or by a particular data center.

Carbon Emission: This refers to the amount of energy consumed by the equipment used in cloud computing or by a particular data center.

Carbon Emission: The carbon emissions generated by an electric service provider have a detrimental impact on the environment, necessitating their reduction or control. Resource utilization refers to the effective utilization of system resources such as CPU, memory, storage, and networking. In the event of a system failure, the workload is shifted to another virtual machine to enhance system reliability.

Bandwidth (BW): This regulates the simultaneous flow of incoming and outgoing traffic between internet agents and the local network. Managing the traffic disparity through the network is crucial.

These are the primary metrics utilized for evaluating load balancing, although additional factors such as BW, Overhead, Cost, Accuracy, Predictability, Thrashing, Associated Overhead, Reliability, ANOVA5) and Associated Cost may also be considered based on specific research requirements.

3. Overview of Surveys (OoS)

Every selected publication in this study utilized the keyword "Load Balancing in Cloud Computing" either in the title or as a keyword. Various reputable journals such as Elsevier, IEEE, Springer, Wiley, and other international journals were included. Numerous publications have been dedicated to exploring this topic within the realm of cloud computing. These publications are categorized into two main groups: survey/review articles and experimental-based articles. This section focuses specifically on survey/review articles, which are further analyzed, based on certain parameters, and summarized in Table 1.

Table 2 contains a comparison of our survey's results with those of previous surveys in terms of various factors.

4. Cutting-Edge Algorithms and Analysis

The load balancing algorithms mentioned in the literature were classified based on one of the following criteria:

Dynamic and Static

Depending on the initiator of the process and the system's current condition.

Table 1. Overview of Literature Review

Year	Ref	Author(s)	Remarks
2015	6)	Mala Kalra et al.	Performance and effectiveness of metaheuristic algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Least Common Ancestors (LCA), and BAT are investigated and compared in the context of cloud and grid computing
	7)	Sulthan Rafii et al.	Analyzed the effect of carbon dioxide (CO_2) caused by variation in Ant-Based Phylogenetic Reconstruction (ABPR) on growth rate in HS-9.
2016	8)	Jiangtao Zhang et al.	Articles and algorithms are inspected and viewed to achieve the goals of resource providing. Algorithm techniques are categorized and methodically examined. Addressed the problems and shortcomings of conventional methods.
	9)	Alireza Sadeghi Milani et al.	Proposed a thorough literature review of the Dynamic and Hybrid load balancing strategies currently in use. Outlined the characteristics of several load balancing systems, including benefits and drawbacks. Various cloud metrics are used to perform detailed classification. Addressed the difficulties and unresolved problems posed by these algorithms.
2017	10)	Einollah Jafarnejad Ghomi et al.	Analyzed and provided a modern classification of load balancing and work scheduling methods. Analyzed and assessed seven different kinds of load balancing algorithms, then condensed the results into Quality of Services (QoS) measures. Gave insight into unresolved problems and suggested directions for further investigation.
	11)	Minxian Xu et al.	Identified the difficulties and examined the load-balancing methods in use to assign VMs to hosts in IaaS. Algorithms that have been surveyed are categorized using classification. A comprehensive and comparative analysis of previous load balancing techniques was presented. Gave the authors a new perspective on potential future improvements.
	12)	Avnish Thakur et al.	Presented a classification of taxa that balances their load based on statistics and inspiration from nature. Presented a flowchart for each algorithm. State-of-the-art algorithms were examined, analyzed, and summarized in a tabular format. Pie charts were utilized to present the metrics used in the various articles. Talked about the difficulties and unresolved problems, as well as potential solutions.
2018	13)	Sambit Kumar Mishra et al.	Outlined a taxonomy analysis of static and dynamic load balancing algorithms. Outlined and presented a thorough approach to load balancing techniques. Sorted the algorithms into groups according to the related performance measures. In order to evaluate the performance in terms of Makespan and Energy usage, graphs were employed to simulate the behavior of several heuristic algorithms.
2019	14)	AR. Arunarani et al.	Discussed a thorough study on work scheduling and the corresponding measures. Addressed the different scheduling-related problems and obstacles to be solved. To determine the value of scheduling characteristics, many scheduling methods are explored. A systematic review of the literature based on three metrics: methods, applications, and parameter based. Recognized the challenges for cloud computing-related future research.
2020	15)	Devaraj, A. F. S. et al.	Suggested Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization (FIMPSO) model demonstrated efficient operation, maximum CPU utilization, proper resource usage, and job response times.
	16)	Azizi, S. et al.	Presented a greedy randomized VM placement (GRVMP) algorithm with multiple performance matrices for use in real-world production environments (Amazon EC2).

2021	17)	Tian, W.et al.	Divides a VM request into a number of sequential sub-requests, each of which is treated as a regular VM request and has its own start time, finish time, and capacity requirement. By accomplishing the predetermined load balancing objective, it becomes possible to proactively establish a threshold for each virtual machine (VM) request on every physical machine. This, in turn, instructs the scheduler to make necessary preparations prior to VM migration.
	18)	Saxena, D.et al.	By doing experiments on the Google Cluster dataset, Planet Lab, and Bit brains VM traces, three real-world workload datasets are used to evaluate the proposed framework.
2022	19)	M. A. Habib et al.	Investigated two game structures photovoltaic (PV) and carbon footprint (CF) power system for analysis in data.
	20)	Manoj Kumar Gupta et al.	Discussed about Carbon fiber reinforced polymers (CFRPs) and Carbon Fibers. The study investigates the detailed study of automotive and aerospace.
2023	21)	Joshila Grace LK et al.	Focused on Swarm intelligence methods are a type of microbe algorithm that have long been demonstrated to be quite successful.
	22)	Sharma A. et al.	'Ant colony optimization- Breadth first computation-Minkowski Static' (ABMS) approach and 'Ant colony optimization-Breadth first computation-Minkowski Dynamic' (ABMD) technique have been simulated.

Table 2. Comparison of the Current and Previous Surveys

Ref	**[6]**	**[7]**	**[8]**	**[9]**	**[10]**	**[11]**	**[12]**	**[13]**	**[14]**	**[15]**	**[16]**	**[17]**	**[18]**	**[19]**	**[20]**	**Current Survey (2023)**
Year	**2015**		**2016**		**2017**			**2018**	**2019**	**2020**		**2021**		**2022**		
Comparative Evaluation	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Cutting-Edge	√	×	×	√	√	×	√	√	√	√	√	√	√	√	√	√
Visual Presentation	×	√	×	√	√	×	√	√	√	√	√	√	√	√	√	√
Classification System	√	×	×	×	√	√	√	√	×	√	√	√	√	√	√	√
Flowchart Representation	×	×	×	×	×	×	√	×	×	×	×	√	√	√	√	√
Survey of Survey (SoS)	×	×	×	×	×	×	×	×	×	×	×	×	×	√	×	√

Statistics-based and nature-inspired approaches

In this section, we have categorized the load balancing algorithms into four main groups based on the types of algorithms utilized for this purpose, ranging from classic approaches to hybrid heuristics, as depicted in Fig. 1.

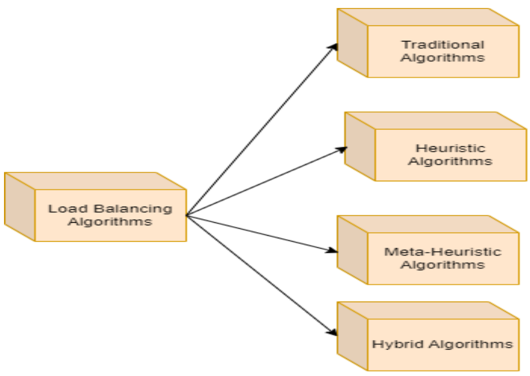

Fig. 1: Categorization of Load Balancing Techniques

The subsequent subsections provide a brief overview of each algorithm within these groups. We have reviewed and evaluated the various algorithms used and implemented by researchers in each category based on specific criteria. The load balancing algorithm is categorized as follows:

4.1 Traditional Algorithm

The traditional algorithm encompasses well-known CPU scheduling techniques. CPU scheduling allows one process to run while others wait in a queue, efficiently utilizing the CPU. The operating system (OS) selects a process from the ready queue and allocates CPU time for execution. In distributed computing systems, various load balancing scheduling algorithms are available. Conventional algorithms can be categorized as preemptive or non-preemptive. Preemptive algorithms involve interrupting an ongoing execution to prioritize and complete a higher priority task before resuming the interrupted process. Priority, either internal or external, influences preemptive scheduling[21)]. Examples of such algorithms include Round Robin and Priority-based scheduling. Non-preemptive scheduling algorithms, on the other hand, do not consider priority and allow the task that completes first to utilize the resources and SJF (Shortest Job First). Typically, a ready queue is maintained as a linked list, where tasks await execution based on their arrival time.

4.2 Heuristic Algorithms

Heuristic algorithms are employed when classical or traditional methods are slow or fail to provide exact solutions. Heuristics serve as optimization strategies to solve problems more efficiently and are often referred to as approximation algorithms. These techniques aim to generate solutions within an acceptable time frame, even if they may not be the optimal ones. Heuristic algorithms utilize informed guesses to determine potential solutions, either independently or in conjunction with other optimization techniques. Static heuristics are employed when the duration of a task is known in advance, while dynamic heuristics handle dynamically arriving tasks. This section focuses on heuristic algorithms such as Min-min, Max-min, RASA (a hybrid approach), MOF (Bio-MOF) in CO[222)] and Improved Maxmin. The algorithms, research areas, tools, and prospects are discussed. Although significant contributions have been made to the development of load balancing algorithms, there is still room for improvement.

4.3 Meta-heuristic Algorithm

Metaheuristics originate from the fields of artificial intelligence and operations research[23).] Traditional heuristic techniques often produce a limited number of alternative solutions and struggle to approach optimality. To overcome the limitations of heuristics, metaheuristics were developed as iterative improvement techniques by taking the study of BG, SOFC[24–25)] and reduction in CO2 into consideration. Metaheuristics combine higher-level approximate approaches to guide local improvement processes effectively and efficiently explore the search space using Hybrid Taguchi–PCA-GRA[26).] In the words of Voss[27),] a meta-heuristic is "an iterative master process that guides and modifies the operations of subordinate heuristics to develop high-quality solutions effectively. Iterations may involve manipulating single or multiple related solutions. The subordinate heuristics can be basic local searches, high-level or low-level operations, or construction techniques." The three primary operators in this framework are transition, evaluation, and decision, which are employed to search for potential solutions[28).] Perturbative and constructive transitions are the two popular techniques used in combinatorial problems [29).] As presented in Fig. 2, metaheuristic algorithms employed in load balancing can be classified into different categories. These categories include Swarm-based Algorithms[31)], Single Solution approaches[30)], Population-based Algorithms, and Local Search techniques[30)].

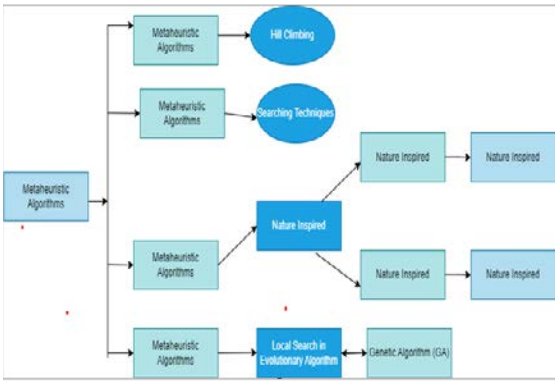

Fig. 2: Taxonomy of Meta-heuristic Techniques

In this study, the algorithms have been categorized and grouped based on their search strategy and solution-oriented features to study the importance of workload balancing techniques by taking the data from articles published from 2012 to 2023. Examples of such algorithms encompass Simulated Annealing[31)], Tabu search[31)], Ant colony optimization[32-33)], Artificial Bee Colony/Honeybee[34-35)], Particle Swarm Optimization[36-37)], Artificial Bee Colony/Honeybee[37)], and Intelligent Water Drop (IWD)[38)].

4.4 Hybrid Algorithm

In this subsection, we have presented a general overview of hybridization. Subsequently, we have examined and analyzed several popular hybrid load balancing strategies. Furthermore, a comprehensive classification, depicted in Fig. 3 below, is presented, highlighting the characteristics of hybrid approaches, and showcasing various hybridization techniques. Hybrid methods have gained prominence by combining different classes of metaheuristic algorithms to leverage their advantages while mitigating their limitations. This synergy is believed to be beneficial for hybrids39). The integration of various mechanisms within this system has proven to be effective, leading to its widespread adoption in the optimization field. As per the previous studies, the exploration capabilities of population-based approaches are significant and have been divided into three sections based on the search space, as depicted in Fig. 3.

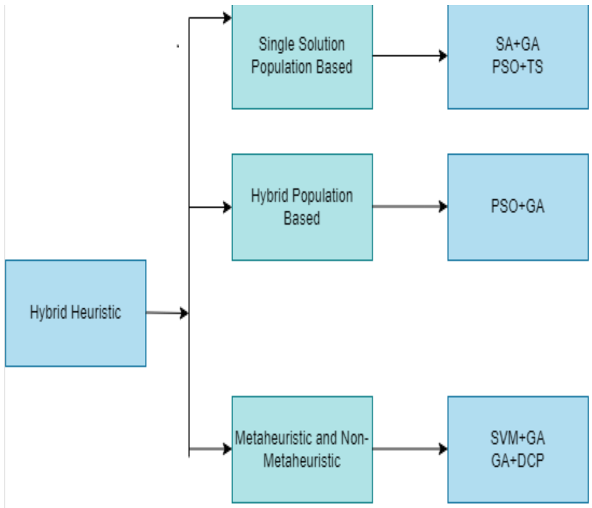

Fig. 3: Taxonomy of Hybrid Heuristic Algorithms

5. Summary of Existing Studies

In this research, Authors have distributed the results in two categories:

5.1 Classification based on parameters

Table 3 given above emphasizes QoS characteristics of the metaheuristic algorithms used in various papers in order to evaluate the efficacy of suggested load balancing approaches.

As shown in Fig. 4, we also used a graph to represent this analysis.

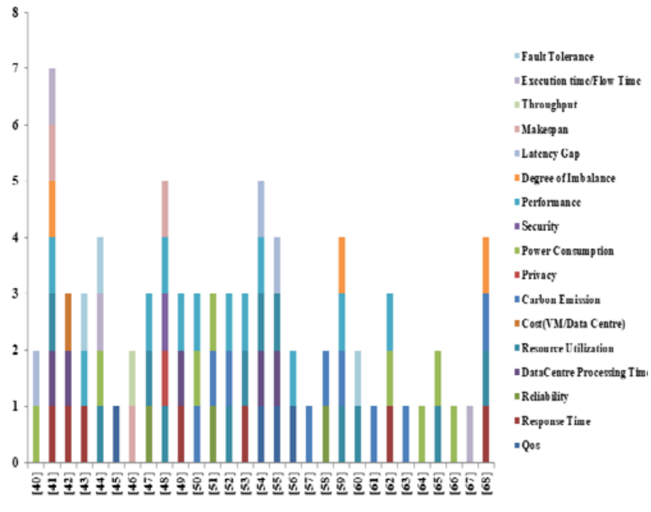

Fig. 4: Graphical Quality of services (QoS) Metrics Display

5.2 Technique-based categorization

In this specific category, our attention was exclusively directed towards metaheuristic techniques, as illustrated in Table 4. This is clear that a substantial portion of these techniques were utilized between 2019 and 2023 to address workload balancing challenges in fog-cloud environments.

6. Discussion

Our survey commenced by examining the available literature, as presented in Table 1. Subsequently, in Table 2, we compared our survey with an existing study in the field. Finally, we conducted an extensive analysis of load balancing methods and their corresponding parameters, which is summarized in our overall analysis and survey. Heuristic methods are created to tackle complicated issues with little effort. When the search space expands in proportion to the magnitude of the problem, it cannot be solved. The literature makes it clear that heuristic approaches are unable to identify the nearly ideal answer in an acceptable amount of time. Additionally, this approach is useless for handling complicated multimodal and combinatorial issues. Due to its advantages over heuristics, meta-heuristic algorithms have been used by many academics to overcome the limitations of heuristic techniques. These methods can be used in numerous series of problems because they are not problem specific. To prevent becoming trapped in local optima, they might merge with other systems. Due to exploration and exploitation, these algorithms can quickly identify close to ideal solutions. These can be used to solve multimodal and combinatorial problems. Although the literature demonstrates that these algorithms perform better, due to some inherent drawbacks of algorithms, they do not ensure the presence of an optimal solution

Table 3. Parameter-Driven Categorization

References	Qos	Reliability	Resource Utilization	Carbon Emission	Power Consumption	Performance	Degree of Imbalance	Fault Tolerance	Makespan	Execution time/Flow Time	Response Time	Throughput	Latency Gap	Datacenters Processing Time	Cost (VM/Data Centre)	Security	Privacy
40	×	×	×	×	√	×	×	×	×	×	×	×	√	×	×	×	×
41	×	×	√	×	×	√	√	×	√	√	√	×	×	√	×	×	×
42	×	×	×	×	×	×	×	×	×	×	√	×	×	√	√	×	×
43	×	×	×	×	×	√	√	√	×	×	√	×	×	×	×	×	×
44	×	×	√	×	√	×	×	√	×	√	×	×	×	×	×	×	×
45	√	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
46	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	√	√
47	×	√	√	×	×	√	×	×	×	×	×	×	×	×	×	×	×
48	×	×	√	×	×	√	×	×	√	×	×	×	×	×	×	√	√
49	×	×	×		×	√	×	×	×	×	√	×	×	√	×	×	×
50	×	×	×	√	√	√	×	×	×	×	×	×	×	×	×	×	×
51	×	√	×	√	√	×	×	×	×	×	×	×	×	×	×	×	×
52	×	×	√	√	×	√	×	×	×	×	×	×	×	×	×	×	×
53	×	×	√	×	×	√	×	×	×		√	×	×	×	×	×	×
54	√	×	√	×	×	√	×	×	×	×	×		√	√	×	×	×
55	√	×	√	×	×	×	×	×	×	×	×	×	√	√	×	×	×
56	√	×	×	×	×	√	×	×	×	×	×	×	×	×	×	×	×
57	×	×	×	√	×	×	×	×	×	×	×	×	×	×	×	×	×

58	×	√		√	×	×	×	×	×	×	×	×	×	×	×	×	×
59	×	×	√	√	×	√	√	×	×	×	×	×	×	×	×	×	×
60	×	×	√	√	×	×	×	√	×	×	×	×	×	×	×	×	×
61	×	×	×	√	×	×	×	×	×	×	×	×	×	×	×	×	×
62	×	×	×	×	√	√	×	×	×		√	×	×	×	×	×	×
63	×	×	×	√		×	×	×	×	×	×	×	×	×	×	×	×
64	×	×	×	×	√	×	×	×	×	×	×	×	×	×	×	×	×
65	×	×	√	×	√	×	×	×	×	×	×	×	×	×	×	×	×
66	×	×	×	×	√	×	×	×	×	×	×	×	×	×	×	×	×
67	×	×	×	×	×	×	×	×	×	√	×	×	×	×	×	×	×
68	×	×	√	√	×	×	√	×	×		√	×	×	×	×	×	×

Table 4. Divisions in workload balancing techniques based on Parameters (Year 2012 to 2023)

Workload Balancing Techniques	**SHC**	**SA**	**GA**	**PSO**	**ACO**	**HONEYBEE**	**IWD**	**BFO**	**LCO**	**FA**	**CS**	**GSA**	**SO**
Year 2012 to 2015				Ref [48]									
Year 2016 to 2019	Ref [40]	Ref [41][42][43]	Ref [44][45][46]	Ref [49][50][51]	Ref [53] [54] [55] [56]	Ref [59]			Ref [63]	Ref [64][65]	Ref [66]	Ref [67]	Ref [68]
Year 2020 to 2023			Ref [47]	Ref [52]	Ref [57] [58]	Ref [60]	Ref [61]	Ref [62]					

Binary PSO, for instance, has a low rate of convergence, and local optima can be a problem for traditional PSO. GSA is known for its extensive computational time, while GA often faces challenges such as early convergence and unpredictable outcomes. GA also involves complex parameter settings for crossover, selection, and encoding strategies. The quality of the initial population can significantly impact the solution quality, and techniques like local search can be employed in algorithms such as PSO and GA to find suitable starting populations. Researchers have explored modifying transition operators in metaheuristic algorithms to improve result quality, for example, by focusing on pheromone updates in ACO. Hybrid approaches, combining metaheuristic and heuristic algorithms or multiple metaheuristic algorithms, aim to leverage the strengths of each algorithm and compensate for their limitations. This integration allows for the exploration of diverse solutions and helps achieve optimal performance and solution quality. Hybridization techniques69) have been applied in different contexts, such as combining GA with SA or PSO with BF and TS to explore advantageous solutions in specific local areas. In earlier studies, researchers have been using different models like ACO, CS, and PSO by using hybridization methods[70] to overcome the problem of local optima. Furthermore, some other researchers worked on other combinations including Genetic Algorithm with fuzzy theory and ACO with network theory. The "power of two choices" approach in ACO[71], XCS and BCM-XCS[72], MtLDF[73], the taxonomy of fog[74], and Firefly algorithm[75] has been shown to outperform other algorithms, as reported in the literature.

7. Summary and Future Perspective

In the resource pool-constrained architecture of cloud computing, it is essential to evenly allocate the workloads among the different fog-cloud nodes (VMs). This ensures efficient resource utilization while considering various parameters such as power usage, carbon emissions, and other quality of service (QoS) standards at data centers. To comprehensively address this scenario, authors have conducted an Overview of Surveys (OoS) to provide a specialized division of workload-balancing algorithms. In this study, authors have established a hierarchy that clarifies the existing algorithms of workload balancing. The performance and outcomes of each algorithm type are thoroughly examined and summarized in tables. By addressing the identified challenges and optimizing the load distribution across multiple VMs, it is possible to enhance these aspects. There is significant room for improvement in the identified algorithms, making further research in this area a top priority. In this paper, a significant advancement in the topic is the creation of a hierarchy for workload balancing algorithms in fog-cloud networks. In the end, it fosters a better understanding of workload balancing difficulties and solutions within the context of fog-cloud computing by helping to organize, categorize, and clarify the available knowledge.

References

1) S. A. Khan, R. Arora, H. Kumar, and P. K. Arora, "A Perspective on Advances in Cloud-based Additive Manufacturing," Evergreen, 9(3) 861-869 (2022). doi.org/10.5109/4843119
2) M. N. Aprilliani, B. Rusdiarso, and W. Trisunaryanti, "Synthesis of Activated Carbon-Fe_3O_4 Composite as Gasoline Adsorbent in Gasoline-Water Mixture and Its Adsorption Kinetics Study," Evergreen, 9(3) 701-710 (2022). doi.org/10.5109/4843101
3) M. S. Sumathi, V. Jain, G. K. Kumar, and Z. Z. Khan, "Using Artificial Intelligence (AI) and Internet of Things (IoT) for Improving Network Security by Hybrid Cryptography Approach," Evergreen, 10(2) 1133-1139 (2023). doi.org/10.5109/6793674
4) S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A big picture," Journal of King Saud University, Computer and Information Sciences,32(2) 149-158 (2018). doi.org/10.1016/j.jksuci.2018.01.003.
5) R. Zulkafli, N. K. Othman, N. Yaakob, F. K. Sahrani, and M. S. H. Al-Furjan, "Electrochemical Studies of Microbiologically Influenced Corrosion on API 5L X65 by Sulfate-Reducing Bacteria in CO_2 Environments," Evergreen 10(1) 601-607 (2023). doi.org/10.5109/6782167
6) M. Kalra, and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," Egyptian informatics journal, 16(3), 275-295, (2015).
7) S. R. Ardiansyah, W. Sjamsuridzal, W. Wardhana, and N. B. Prihantini, "Analysis of carbon dioxide solubility increasement caused by baffle diameter variation in airlift photobioreactor to growth rate of synechococcus hs-9 biomass," Evergreen 8(4) 879-884 (2021). doi.org/10.5109/4742136
8) J. Zhang, H. Huang, and X. Wang, "Resource provision algorithms in cloud computing: A survey," Journal of Network and Computer Applications, 64 23-42 (2016). doi.org/10.1016/j.jnca.2015.12.018.
9) A. S. Milani, and N. J. Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends," Journal of Network and Computer Applications, 71(2) 86-98 (2016) .doi.org/10.1016/j.jnca.2016.06.003.
10) E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: a survey," Journal of Network and Computer Applications, 88(2) 50-71 (2017). doi.org/10.1016/j.jnca.2017.04.007.

11) M. Xu, W. Tian, and R. Buyya, "A survey on load balancing algorithms for virtual machines placement in cloud computing," Concurrency and Computation: Practice and Experience, 29(12) 123-128 (2017). doi.org/10.1002/cpe.4123.
12) A. Thakur, and M. S. Goraya, "A taxonomic survey on load balancing in cloud," Journal of Network and Computer Applications, 98(5) 43-57 (2017) .doi.org/10.1016/j.jnca.2017.08.020.
13) M. Kumar, K. Dubey, and S. C. Sharma, "Elastic and flexible deadline constraint load Balancing algorithm for Cloud Computing," *Procedia Computer Science*, **125**(1) 717-724 (2018). doi.org/10.1016/j.procs.2017.12.092.
14) A. R. Arunarani, D. Manjula, and V. Sugumaran, "Task scheduling techniques in cloud computing: A literature survey," Future Generation Computer Systems, 91(3) 407- 415 (2019). doi.org/10.1016/j.future.2018.09.014.
15) A. F. S., Devaraj, Elhoseny, M., Dhanasekaran, S., Lydia, E. L., and Shankar, "K. 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 (2020). doi.org/10.1016/j.jpdc.2020.03.022.
16) S. Azizi, M. J. Abawajy, and R. Buyya, Grvmp," A greedy randomized algorithm for virtual machine placement in cloud data centers," IEEE systems journal, 15(2) 2571-2582 (2020). doi.org/10.1109/JSYST.2020.3002721.
17) W. Tian, M. Xu, G. Zhou, K. Wu, C. Xu, and R. Buyya, "Prepartition: Load Balancing Approach for Virtual Machine Reservations in a Cloud Data Center," 19(10) (2021). doi.org/10.48550/arXiv.2110.09913.
18) D. Saxena, A. K. Singh, and R. Buyya, "OP-MLB: an online VM prediction-based multi-objective load balancing framework for resource management at cloud data center," IEEE Transactions on Cloud Computing, 10(4) 2804-2816 (2021). doi.org/10.1109/TCC.2021.3059096.
19) M. A. Habib, K. A. Kabir, and J. Tanimoto, "Evolutionary game analysis for sustainable environment under two power generation systems," Evergreen 9(2) 326-344 (2022). doi.org/10.5109/4793672
20) M. K. Gupta, V. Singhal, and N. S. Rajput, "Applications and Challenges of Carbon-fibres reinforced Composites: A Review," Evergreen 9(3) 682-693 (2022). doi.org/10.5109/4843099
21) LK, J. G., Maneengam, A., KV, P. K., and J., Alanya-Beltran, "Design and Implementation of Machine Learning Modelling through Adaptive Hybrid Swarm Optimization Techniques for Machine Management," Evergreen 10(2) 1120-1126 (2023). doi.org/10.5109/6793672
22) A. Sharma, S., Sharma, and D. Gupta, Ant Colony Optimization Based Routing Strategies for Internet of Things, Evergreen 10(2) 998-1009 (2023). doi.org/10.5109/6793654
23) C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," Applied Soft Computing, 11(6) 4135-4151 (2011). doi.org/10.1016/j.asoc.2011.02.032.
24) N. M. A. Lestari, "Reduction of co_2 emission by integrated biomass gasification-solid oxide fuel cell combined with heat recovery and in-situ co_2 utilization," Evergreen 6(3) 254-261 (2019). doi.org/10.5109/2349302
25) B. Shahriari, A. Hassanpoor, A. Navehebrahim, and S. Jafarinia, "A systematic review of green human resource management," Evergreen 6(2) 177-189 (2019). doi.org/10.5109/2328408.
26) A. Kumar, A. K. Chanda, and S. Angra, "Optimization of stiffness properties of composite sandwich using hybrid taguchi-gra-pca," Evergreen 8(2) 310-317 (2021). doi.org/10.5109/4480708
27) C. W. Tsai, and J. J. Rodrigues, "Metaheuristic scheduling for cloud: A survey," IEEE Systems Journal,8(1) 279-291 (2014). doi.org/10.1109/JSYST.2013.2256731.
28) C. W. Tsai, W. C. Huang, M. H. Chiang, M. C. Chiang, and C. S. Yang, "A hyper-heuristic scheduling algorithm for cloud," IEEE Transactions on Cloud Computing, 2(2) 236-250 (2014) doi.org/10.1109/TCC.2014.2315797.
29) Z. Beheshti, and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithms," Int. J. Adv. Soft Comput. Appl, 5(1) 1-35 (2013).
30) H. H. Hoos and T. St¨utzle, "Stochastic Local Search: Foundations & Applications," Amsterdam, The Netherlands: Elsevier, 2004 https://books.google.co.in/books?hl=en&lr=&id=DyzhHMOhI.
31) K. Amine, "Multiobjective simulated annealing: Principles and algorithm variants," Advances in Operations Research, 19(1) 1-13 2019 doi.org/10.1155/2019/8134674.
32) A. Akhtar, "Evolution of Ant Colony Optimization Algorithm--A Brief Literature Review arXiv preprint arXiv:1908.08007", 10(2) 1-11 (2019). doi.org/10.48550/arXiv.1908.08007.
33) W. Deng, J. Xu, and H. Zhao, "An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," IEEE access, 7(1) 20281-20292 (2019). doi.org/10.1109/ACCESS.2019.2897580.
34) C. A. Oi, "Honeybee queen mandibular pheromone fails to regulate ovary activation in the common wasp," Journal of Comparative Physiology A, 208(2) 297-302 (2022).

doi.org/10.1007/s00359-021-01531-0.

35) A. Leska, A. Nowak, J. Szulc, I. Motyl, and K. H. Czarnecka-Chrebelska, "Antagonistic Activity of Potentially Probiotic Lactic Acid Bacteria against Honeybee," Apis mellifera L. Pathogens. Pathogens, 11(11) 1367 (2022). doi.org/10.3390/pathogens11111367.
36) A. Sohal, and R. Kait, "Analytical Workload Allocation for Optimized Power Consumption and Delay in Fog-Cloud Networks Using Particle Swarm Optimization Algorithm," IREMOS 16(1) 5-12 (2023). doi.org/10.15866/iremos.v16i1.23282.
37) V. Kayvanfar, M. Zandieh, and E. Teymourian, "An intelligent water drop algorithm to identical parallel machine scheduling with controllable processing times: a just-in-time approach," Computational and Applied Mathematics, 36(1) 159-184 (2017). doi.org/10.1007/s40314-015-0218-3.
38) S. Nonita, P. A. Xalikovich, C. R. Kumar, M. Rakhra, I. A. Samori, Y. M. Maquera, and J. L. A. Gonzáles, "Intelligent Water Drops Algorithm-Based Aggregation in Heterogeneous Wireless Sensor Network," Journal of Sensors, 2022. doi.org/10.1155/2022/6099330.
39) F. Al-Tam, and N. Correia, "On load balancing via switch migration in software-defined networking," IEEE Access, 7(2) 95998-96010 (2019). doi.org/ 10.1109/ACCESS.2019.2929651.
40) S. S. Tripathy, K. Mishra, D. S. Roy, K. Yadav, A. Alferaidi, W. Viriyasitavat,... and R. K. Barik, "State-of-the-art load balancing algorithms for mist-fog-cloud assisted paradigm: A review and future directions," Archives of Computational Methods in Engineering, 30(2) 2725–2760 (2023). doi.org/10.1007/s11831-023-09885-1.
41) J. Zhou, U. K. Lilhore, T. Hai, S. Simaiya, D. N. A. Jawawi, D. Alsekait,... and M. Hamdi, Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing. Journal of Cloud Computing, 12(1) 1-21 (2023). doi.org/10.1186/s13677-023-00453-3.
42) M. A. Shahid, M. M. Alam, and M. M. Su'ud, "Performance Evaluation of Load-Balancing Algorithms with Different Service Broker Policies for Cloud Computing. Applied Sciences," 13(3) 1586 (2023). doi.org/10.3390/app13031586.
43) B. Mondal, and A. Choudhury, "Simulated annealing (SA) based load balancing strategy for cloud computing," International Journal of Computer Science and Information Technologies, 6(4) 3307-3312 (2015).
44) M. I. Khaleel, "Efficient job scheduling paradigm based on hybrid sparrow search algorithm and differential evolution optimization for heterogeneous cloud computing platforms," Internet of Things, 22 100697 (2023). doi.org/10.1016/j.iot.2023.100697.
45) N. K. Gupta, A. Walia, and A. Sharma, "GP-MSJF: An improved load balancing generalized priority-based modified SJF scheduling in cloud computing," In Advances in Information Communication Technology and Computing: Proceedings of AICTC, 589-597 (2021). doi.org/10.1007/978-981-19-0619-0_51.
46) A. S. Shaiden, S. Islam, and K. Subramaniam, "Android based Digital Steganography Application using LSB and PSNR Algorithm in Mobile Environment," Evergreen 8(2) 421-427 (2021). doi.org/10.5109/4480724
47) M. Vanitha, and P. Marikkannu, "Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines," Computers & Electrical Engineering, 57(4) 199-208 (2017). doi.org/10.1016/j.compeleceng.2016.11.001.
48) G. S. Sriram, "Challenges of cloud compute load balancing algorithms," International Research Journal of Modernization in Engineering Technology and Science, 4(1) 1186-1190 (2022).
49) A. Saoud, and A. Recioui, "Hybrid algorithm for cloud-fog system based load balancing in smart grids," Bulletin of Electrical Engineering and Informatics, 11(1) 477-487 (2022). doi.org/10.11591/eei.v11i1.3450.
50) F. Ramezani, J. Lu, and F. K. Hussain, "Task-based system load balancing in cloud computing using particle swarm optimization," International journal of parallel programming, 42(5) 739-754 (2014). doi.org/10.1007/s10766-013-0275-4.
51) A. I. Awad, N. A. El-Hefnawy, and H.M. Abdel_kader, "Enhanced particle swarm optimization for task scheduling in cloud computing environments," Procedia Computer Science, 65(1) 920-929 (2015). doi.org/10.1016/j.procs.2015.09.064.
52) F. Ebadifard, and S. M. Babamir, "A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment," Concurrency and Computation: Practice and Experience, 30(12) e4368 (2018). doi.org/10.1002/cpe.4368.
53) R. P. S. Bedi, "insight on the comparative analysis of load balancing and service broker algorithms in cloud computing," Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology, 54(3) 105-116 (2022).
54) P. Kumar, and R. Kumar, "Issues and challenges of load balancing techniques in cloud computing: A survey," ACM Computing Surveys (CSUR), 51(6) 1-35 (2019). doi.org/10.1145/3281010.
55) V. Priya, C. S. Kumar, and R. Kannan, "Resource scheduling algorithm with load balancing for cloud service provisioning," Applied Soft Computing, 76 416-424 (2019). doi.org/10.1016/j.asoc.2018.12.021.
56) M. Padmavathi, and S. M. Basha, "Dynamic and

elasticity ACO load balancing algorithm for cloud computing," *In Intelligent Computing and Control Systems (ICICCS), 2017 International Conference*, 77-81 (2017).

57) A. Gupta, and R. Garg, "Load balancing based task scheduling with ACO in cloud computing," *In Computer and Applications (ICCA), 2017 International Conference on*, 174-179 (2017). doi.org/10.1109/COMAPP.2017.8079781.

58) P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments,," Applied Soft Computing, 13(5) 2292-2303 (2013). doi.org/10.1016/j.asoc.2013.01.025.

59) D. P. Mahato, and R. S. Singh, "Load balanced transaction scheduling using Honey Bee Optimization considering performability in on-demand computing system," Concurrency and Computation: Practice and Experience, 29(21) e4253 (2017). doi.org/10.1002/cpe.4253.

60) S. Elsherbiny, E. Eldaydamony, M. Alrahmawy, and A. E. Reyad, "An extended Intelligent Water Drops algorithm for workflow scheduling in cloud computing environment," Egyptian informatics journal, 19(1) 33-55 (2018). doi.org/10.1016/j.eij.2017.07.001.

61) L. Tang, Z. Li, P. Ren, J. Pan, Z. Lu, J. Su, and Z. Meng, "Online and offline based load balance algorithm in cloud computing," Knowledge-Based Systems, 138(2) 91-104 (2017). doi.org/10.1016/j.knosys.2017.09.040.

62) K. T. Zingre, X. Yang, and M. P. Wan, "Performance analysis of cool roof, green roof and thermal insulation on a concrete flat roof in tropical climate," Evergreen 2(2) 24-33 (2015). doi.org/10.5109/1544078.

63) N. Susila, S. Chandramathi, and R. Kishore, "A fuzzy-based firefly algorithm for dynamic load balancing in cloud computing environment," Journal of Emerging Technologies in Web Intelligence, 6(4) 435-440 (2014). doi.org/10.4304/JETWI.6.4.435-440.

64) A. P. Florence, and V. Shanthi, "A load balancing model using firefly algorithm in cloud computing," Journal of Computer Science, 10(7) 1156-1165 (2014). doi.org/10.3844/jcssp.2014.1156.1165.

65) M. Yakhchi, S. M. Ghafari, S. Yakhchi, M. Fazeli, and A. Patooghi, "Proposing a load balancing method based on Cuckoo Optimization Algorithm for energy management in cloud computing infrastructures," *In Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 6th International Conference on*, 1-5 (2015). doi.org/10.1109/ICMSAO.2015.7152209.

66) D. Chaudhary, and B. Kumar, "Cloudy GSA for load scheduling in cloud computing," Applied Soft Computing, 71 861- 871 (2018). doi.org/10.1016/j.asoc.2019.105627.

67) M. Abdullahi, and M. A. Ngadi, "Symbiotic Organism Search optimization based task scheduling in cloud computing environment," Future Generation Computer Systems, 56 640-650 (2016). doi.org/10.1016/j.future.2015.08.006.

68) B. Yang, J. Wang, X. Zhang, T. Yu, W. Yao, H. Shu... and L. Sun, "Comprehensive overview of meta-heuristic algorithm applications on PV cell parameter identification," Energy Conversion and Management, 208 112595 (2020). doi.org/10.1016/j.enconman.2020.112595.

69) M. N. Omidvar, X. Li, and X. Yao, "A review of population-based metaheuristics for large-scale black-box global optimization—Part II," IEEE Transactions on Evolutionary Computation, 26(5) 823-843 (2021). doi.org/10.1109/TEVC.2021.3130835.

70) P. Kumari, and S. K. Sahana, "Swarm based hybrid ACO-PSO meta-heuristic (HAPM) for QoS multicast routing optimization in MANETs," Wireless Personal Communications, 93(2) 1-23 (2022). doi.org/10.1016/j.micpro.2022.104574.

71) M., Sumathi, N., Vijayaraj, S. P., Raja, and M. Rajkamal, "HHO-ACO hybridized load balancing technique in cloud computing." International Journal of Information Technology, 15(3), 1357-1365 (2023). https://doi.org/10.1007/s41870-023-01159-0.

72) M. Abbasi, M. Yaghoobikia, M. Rafiee, A. Jolfaei, and M. R. Khosravi, "Efficient resource management and workload allocation in fog–cloud computing paradigm in IoT using learning classifier systems," Computer communications, 153(1) 217-228 (2020). doi.org/10.1016/j.comcom.2020.02.017.

73) E. C. P. Neto, G. Callou, and F. Aires, "An algorithm to optimise the load distribution of fog environments. In 2017 IEEE International Conference on Systems," Man, and Cybernetics (SMC), 10(1) 1292-1297 (2017). doi.org/10.14529/jsfi230101.

74) R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," Internet of Everything: Algorithms, Methodologies, Technologies, and Perspectives, 5(5) 103-130 (2018). doi.org/10.1007/978-981-10-5861-5_5.

75) M. Adhikari, and H. Gianey, "Energy efficient offloading strategy in fog-cloud environment for IoT applications," Internet of Things, 6(2) 100053. (2019). doi.org/10.1016/j.iot.2019.100