Journal of Computer Allied Intelligence
ISSN: 2584-2676
Vol.02(06), December 2024, pp.19-50

Review Article

Comparative Analysis of Load Balancing and Service Broker Algorithms in Cloud Computing

Tejinder Sharma[1,*] and Narinder Sharma[2]
[1,2] Department of Computer Science & Engineering, Amritsar Group of Colleges, Amritsar-143001, Punjab, India.
[*]Corresponding Author: Tejinder Sharma. Email: tejinder.acet@gmail.com

Received: 25/10/2024; Accepted: 20/12/2024.
DOI: https://doi.org/10.69996/jcai.2024027

Abstract: Cloud computing is a phenomenon which is growing exponentially for the enhanced use of network services, where, the proficiency of one node can be utilized by another node as per requirement of the end users. To manage these types of requirements in the precise and accurate way, various load balancing and service brokering techniques/algorithms have been proposed by the developers/researchers. The Techniques/algorithms discussed in this manuscript ensures the uniform distribution of the load on individual node along with optimal resource utilization and faster response time. This Manuscript demonstrates the comparative analysis of the various existing load balancing algorithms and service broker policies based on the available state-of -art literature along with the required performance metrics and challenges faced by researchers while proposing the new algorithms.

Keywords: Cloud computing, load balancing, service broker algorithm, VM, resource utilization.

1 Introduction

When preparing their papers for submission, authors are expected to adhere to this template that is available in Microsoft Word. Both the review and the typesetting process will be sped up as a result. In the present technological era, the cloud computing has embellished a huge demand with the revolutionary changing in the internet technologies. It is quite cumbersome for the users to elect an appropriate service provider. In the same time, users deal with distinct types of instances, interfaces and price management. Keeping these under consideration, cloud brokering and load balancing came into the existence. A cloud broker manages the scheduling management as well as uniform interface among various cloud service providers, as per load balancing is concerned It necessitates the implementation of efficient load balancing techniques to ensure the timely and optimal execution of assigned tasks within the specified timeframe [38]. Load balancing is stereotypically attained through the employment of Load Balancers (LB), which flawlessly forward incoming pleas without the client's cognizance. These LBs employ distinct scheduling algorithms to evaluate pre-decided parameters like current load and server availability for determining the most suitable server for managing each request. By sending requests to the elected server, LBs optimize the resource usage and improve overall system performance, as delineated in Figure 1. For making the final decision, the LB retrieves information about the server’s status including the health and workload for verifying its capcity to cater that request. The issues of load imbalance represent a multifaceted challenge with multiple constraints, which can significantly impair the performance and efficiency of computing resources if not effectively managed [2]. Various broker and scheduling algorithms have been defined by researchers to regulate the most appropriate backend server for handling incoming requests. Random selection or round-robin allocation algorithm come in the basic category. Based on the originator of the load balancing process, these algorithms can be categorized into distinct ways, including Receiver-Initiated, Sender-Initiated, Symmetric, Static, and Dynamic techniques, each offering unique strategies for optimizing system performance. Primarily used simulation tools for analyses of the scheduling and broker algorithms are CloudAnalyst and CloudSim. For providing the solutions to afore-said, a new simulator CloudSim has been discussed by the researcher [113] which enables to the new users to analyze and simulate the execution of their designed algorithms. The CloudSim exhibits the features like easy to use, capability to describe with large degree of flexibility and configurability, Graphical output, Repeatability, ease of extension in connection to it simulator also explains the Power Consumption, Virtual Machine Allotment, Network behavior, Cloud federation etc.

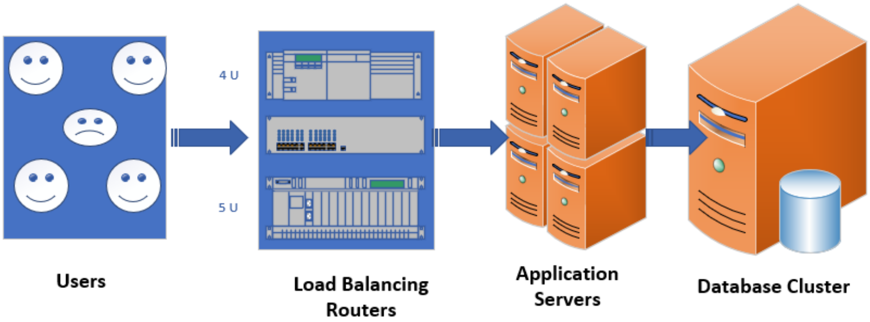

Figure 1: Schematics of highly available computers with load balancing

This manuscript comprises of six different sections and further some sections contain sub-sections. Section 1 demonstrates the introduction, section 2 describes the required metrics for cloud computing. Section-3 primarily propounds the Existing Load Balancing and Service Brokering Algorithms, further it is divided into two sub-sections like Load Balancing Algorithms and Service Brokering Algorithms for better understanding. Section 4 expounds the related literature survey along with comparative analysis of the existing techniques/algorithms. Section 5 is divided into subsections for better understanding and reports the challenges faced by various researchers in proposing optimal algorithms for load balancing and service brokering. Section No. 6 is the conclusion, which clearly defines the purpose of the comparative analysis conducted based on the existing state-of-the-art literature.

2 Metrics

For getting the desired output, it is always good to understand that how the load can be uniformly distributed to all the virtual Machines (VMs) available in the data center for that researchers must know about the in depth knowledge of the metrices used in the service brokering and load balancing algorithms as well as optimal management of these parameters. The most common used metrics are given as:

- **Throughput:** This metric is directly related to performance of the system, if throughput rate is high, then system performance will be better and vice versa. So, it is explained as the rate at which the submitted requests are processed with respect to time.
- **Response Time:** The time required to respond to the submitted request.
- **Makespan:** The time required to allocate resources to the desired users.
- **Migration Time:** It is the time required to shift a request transitioning from an

overloaded VM to an underutilized one. Lesser value of Migration time will lead to the high performance of the system.

- **Scalability:** It is actually the competence of the node to manage consistent load balancing even if required number of nodes shoots up.
- **Resource Use:** It is a vital metrices, if the Resource use will be more than overall cost of the system will get reduced. So, it can be defined as optimal use of the available resources for reducing cost, energy utilization and carbon emission.
- **Failure Resilience:** It is basically the ability of the load balancing method to function uniformly even in case of failure of any VM or link.
- **Degree of Imbalance:** It mainly occurs because of difference in the performance of VMs
- **Power Saving:** This metric describes the level of strength and power consumed by the virtual machine (VM) after the load balancing (LB) procedure is executed. An efficient LB method reduces the power and energy usage in a VM.

3 Existing Load Balancing and Service Brokering Algorithms

This section will provide an in-depth explanation of the current techniques/methods for load balancing, as well as a comprehensive overview of Service Broker Algorithms.

3.1 Load Balancing Algorithms

A load balancing algorithm is basically a set of rules used to allocate workloads across multiple computing resources like servers, networks, or clusters, to optimize resource use, improve response times, and ensure reliability. These algorithms help to balance the load by distributing incoming traffic or tasks evenly, preventing any single resource from becoming overwhelmed, and these algorithms have been classified in the following sub-sections.

3.1.1 Round Robin Algorithm (RR)

This algorithm is also said as FCFS Scheduling and the simplest one Algorithm available among available algorithms which is based on the concept of slices or time quantum as shown in figure 2. In this method, the time is being classified into number of slices which is being allocated to all the nodes available in the data centres for executing the operations. All resources are assigned on the basis of predefined time quantum intervals, confirming unbiassed sharing and effective usage across all processes. If the time quantum is very less then this scheduling is said to be Processor Sharing Algorithm with large number of context switches. It usually elects the load purely on random basis and also confront with heavily loaded and lightly loaded nodes. Although this method is quite simple yet additional load is required to addon the task scheduler to choose the quantum's size and it has more waiting time (average), low throughput, more turnaround time and higher context switches [37].

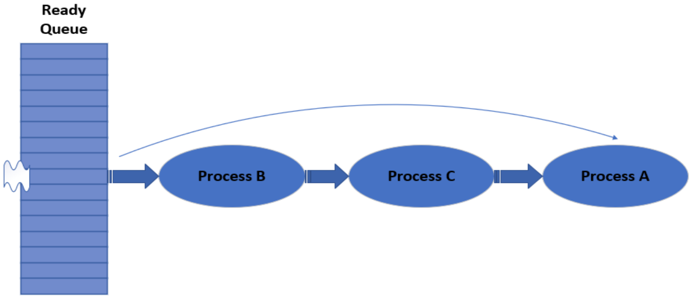

Figure 2: Round Robin Algorithm

3.1.2 Equally Spread Current Execution Algorithm (ESCE)

In this method (portrayed in figure 3), all the VMs available in the data center receive identical load, an index table is being managed by Load Balancer along with request count presently allocated to the VMs. If the data center receives any request in regard to allocation of VM, firstly, it checks the index table for the status of available VMs, if more than one VM is being divulged, then it identifies the first VM for handling the client request, in the mean while, load balancer also sends the ID of VM to the data center. After that, data center interconnects the plea to the identified VM (by ID) and updates its index table by reducing the alloted count of the VMs and this process is being repeated for the next request [119] [53].

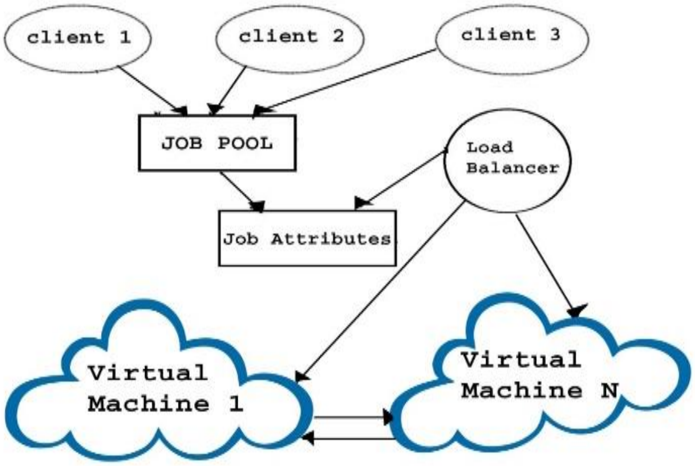

Figure 3: Equal Spread Current Execution

3.1.3 Weighted Round Robin Algorithm (WRR)

It is quite similar to the traditional method except weights have been assigned to each node. These weights have been allotted by the researcher [76] which is elected on the basis of the VMs capacity. This method is beneficial for computing the waiting time but major shortcoming is that it doesn’t distinct lengths of jobs to allot the appropriate VM.

3.1.4 Throttled Load balancing Algorithm (TLB)

In this method, the Load Balancer (LB) keeps an index table that monitors the status of VMs, differentiating them as either Available or Busy, depicted as figure 4. Firstly, server requests to data center for finding the required VM in order to perform the assigned task. After that, Data Center (DC) inquiries from Load Balancer for the allotment of the respective VM. At the same time, LB sequentially parses the index table starting from top until the accessibility of the respective VM. If the VM is being divulged, then, DC transfers this request to the identified VM and, simultaneously, DC also acknowledges to the Load Balancer (LB) and upgardes its index table respectively. During Scanning, If the VM)is not located, the LB get backs a value of 1 to the data center (DC), which then, places the request in its queue for furthermore processing As soon as VM accomplishes an assigned job, a request will be sent to DC, which will be further imparted to LB for de- allocation of the same VM (already shared ID) [1] [79].

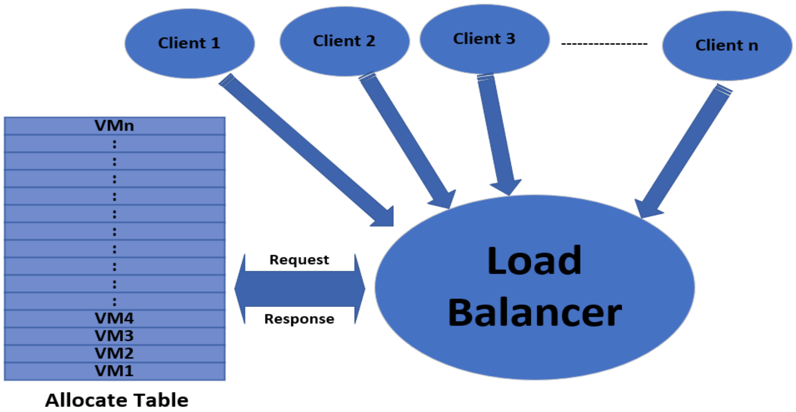

Figure 4: Throttled Load Balancing Algorithm

3.1.5 Minimum Execution Time: Heuristic Technique

This heuristic technique, also acknowledged as Limited Best Assignment, is pertinent for both static and dynamic strategies. It is precisely designed to optimize the distribution of jobs to the VMs [43].

3.1.6 Min-Min Algorithm

In this algorithm, the cloud environment chooses the task with the lowest size and the VM with the minimum capacity. Once the task is assigned to a specified VM, it is removed from the cavalcade, allowing the system to concentrate on the allocation of the remaining unassigned tasks [72].

3.1.7 Max-Min Algorithm

This technique closely resembles to the approach used in the Min-Min algorithm; however, it prioritizes the selection of the job with the largest size while opting for the VM of lowest capacity. Once the job is assigned, it is removed from the queue, and facilitates the allotment of the remaining jobs to the cloud resources [20].

3.1.8 Genetic Algorithm (GA)

This algorithm operates on the principles of individual chromosomes and populations. Load balancing metrics including makespan, fairness and throughput are utilized as fitness values for both chromosomes and the population. Further, these values are also optimized. During every iteration, the GA performs processes of selection, crossover, and mutation. Notably, various scientists have defined the chromosome size as the total count of tasks available in the CC environment [5].

3.1.9 Tabu Search

This technique is employed to determine the solution space beyond local optimality. This technique also controls adaptive memory to facilitate a more flexible search [33].

3.1.10 A-star Search

This algorithm is a graphical search method which amalgamate the strengths of both breadth-first and depth-first search algorithms. Within this framework, dual lists are being maintained: one serves as a priority queue for the tasks, while the other pertains to the execution capacities of all VMs [57].

3.1.11 Switching Algorithm

In a cloud computing environment, the switching algorithm is employed for handling jobs and VMs which is significantly good for the fault tolerance capabilities [10].

3.1.12 Central Load Balancer

In this method, each user's request is directed to the Data Centre Controller (DCC) which enquiries from the CLB for the allotment of different requests. The CLB also maintains a comprehensive table which includes various IDs, states, and priority levels of VMs. It finds the VM which has the highest priority and also checks the same for its availability. If the VM is available, then CLB returns its VM ID to the DCC, otherwise, CLB selects the next highest priority VM. Ultimately, the DCC assigns the job to the VM ID provided by the CLB [96].

3.1.13 Virtual machine-assign load balancer algorithm

This algorithm maintains a table of VMs that contains attributes like assign/index and their current load. It thoroughly scans for the least-loaded VM. If a VM is accessible and has not been engaged by a prior job assignment, then algorithm allocates the job along with its ID to that VM and also notifies the same to the data center. If the VM is unavailable, then, algorithm pursues to find the next least-loaded VM and allocates the task or job to it. This process continues iteratively until all the received jobs have been effectively allocated [116].

3.2 Brokering Algorithm

The service broker policies basically act as an interface between the DCs and clients/users. It chooses the data center which will deliver the service pleas from the user base. So, the Service Broker Policies (SBP) manage the load between data centers and the user bases which may lead to impact on cost as well on response time The Service Brokering policies are classified in the following sub-sections.

3.2.1 Service Proximity Based Routing or closet Data Centre Policy

It is simplest among the available policies. It maintains an Index Table of all the Data Centres. Whenever a request is received from the user, it queries from the Service Proximity Service Broker for the destination DCC which regains the region of the user who made the plea and enquiries for the region nearness list for that particular region. The Service Proximity Service Broker preferences the first DC listed in the closest region based on the proximity hierarchy. In case, more than one DC has been found in this region, then anyone may be elected randomly.

3.2.2 Performance Optimised Routing

In this policy, Best Response Time Service Broker is being used instead of Service Proximity Service Broker. In the similar way of previous service, it also maintains an index for Data centres and adopts the same way instead of identifying the nearest data centre in terms of latency. Rest of the process is same and there are 50% probability to elect the data centre which has least response time [81].

3.2.3 Dynamic Service Broker Policy

This algorithm maintains two lists simultaneously, one all data centers and other related to best response time of individual data center. When the user sends the request, it asks for the Dynamic Service Broker for the destination means DCC which uses the Service Proximity Service Broker with Best Response Time. After that, it updates the best response time records in case obtained response time is better than previous one. if the current response time is more than the superlative response time, at the same moment, the Dynamic Service Broker gives directions to the Data Center Controller to increase the number of VMs. If the current response time is less, than Dynamic Service Broker conveys to the DCC to decrease the number of VMs [59].

3.2.4 Service Broker policy for election of best Data Centre

The proposed service broker algorithm is used for optimizing response time and the cost along with process to elect the best Data Centre as a function of the efficiency and cost ratio. This policy explains that if the data center is available with the minimum ratio, then it will be selected otherwise request will be migrated to the nearest data center [14].

4 Related Literature Survey with Comparative Analysis

In the present scenario, there is huge demand of Load Balancing and Service Broker Algorithms besides various researchers/scientists have carried out their work in this direction. The related work in this direction has been discussed in the following subsections.

Chhabra et al. (2006) [17] have explained about the multi-processor system, where, the possibility of anyone of the processor may be remain being inactive, on the other hand, other processors may have multiple tasks. When such type of situation arises in the system load then the performance can be enhanced by migrating the tasks from the processors with the heavy loads to the processors with the light loads. Additionally, he classified load balancing algorithms into two different types named as static and dynamic. Static Load Balancing (SLB) algorithms make decisions related to task assignments for processors as well as transfer delays at compile time. In contrast, Dynamic Load Balancing (DLB) algorithms are flexible for varying conditions and make decisions at runtime. The Switching Algorithm (SA), portrayed by Shao et al. (2014) [95], describes the facilitates related to the reallocation of the tasks to attain balanced load. In another research effort, AlShawi et al. (2012) [9] illustrated a hybrid technique which is a concoction of fuzzy methods with the A-star algorithm for improving network longevity. Furthermore, Tsai et al. (2014) [95] expounded a parallel variant of the TS algorithm employed to a master-slave model. Larumbe et al. (2013) [68] has propounded an improved TS technique for the optimization of the arrangement of cloud DCs across various locations, majorly focusing on enhancing network performance, decreasing CO2 emissions, and improving resource utilization costs within the CC environment. For supporting the principles of this research, the proposed technique is being applied to the networks consisting of 500 nodes and 1,000 DCs. Dastjerdi A.V et al. (2014) [25] have described that Cloud computing focusses to power the next generation DCs and authorizes application service providers to lease DC capabilities for employing applications depending on user Quality of Service (QoS) needs.

Cloud applications have distinct requirements for composition, configuration and deployment. The authors [99] have stated that hardware technology and network bandwidth is emerging rapidly. So, in order to use the computing resources on the network to execute cumbersome tasks that need large-scale computation, the elected nodes must be taken care of. Nodes must be precisely chosen as per the requirement of the assigned job. Further, they proposed Load Balancing Min-Min algorithms for improving efficiency and maintaining the load balancing. Randles et al. (2010) [87] have explained that anticipated uptake of Cloud computing, will exhibit benefits in terms of flexibility, cost and availability for service users due to virtualization. Afore-said advantages are predictable for Cloud the services including the increase of Cloud's customer base. There are many technical issues in service-oriented architecture and IoS-type applications like high availability, fault tolerance, and scalability but the most critical issue is the effective load balancing techniques. This paper portrays the tri-distributed solutions for load balancing by Honeybee Foraging Behavior, Biased Random Sampling and Active Clustering approaches. Genetic Algorithm based approachhave been anticipated for minimising the makespan [24]. The population is determined with the use of binary strings, and the chromosomes depicted a random single-point crossover and a mutation probability of 0.05. Authors Li et al. (2014)[69] have discussed the Max-Min algorithm Heuristic technique, which defines a job status table for assessing the real-time load of VMS, along with the projected completion times of assigned jobs. This paper is focused to enhance makespan and balance the load of the DC by carefully assigning jobs [73].

The vital disadvantage of proposed technique is that it does not pay attention to the machine ready time and also indicates various changes in the load across the virtual machines. The researchers [70] explained the techniques In parallel processing within CC, it is mandatory to device an effective procedure for resource allotment and task arrangement. Parallelly, usage of a resource allocation technique that provisions preemptive job execution can vitally improve the overall usage of cloud resources, enhancing performance and efficiency. In this manuscript, the author explained an adaptive resource allocation algorithm for CC systems. The algorithm dynamically regulates the resource allocation based on real-time updates of the authentic task execution. Buyya R. (2009)[15] has explained that advancements in cloud computing have opened advanced options for Internet application developers. Previously, their vital focus was around the positioning and hosting of applications. The hosting and distribution of the resources have become economical as well as easily accessible through scalable, pay-per-use, and adaptable infrastructure services provided by cloud sellers. There are numerous cloud providers which offer distinct pricing models in distinct geographic location, the application developers are encountering issues for electing suitable providers and DC locations.

Though, there is a scarcity of tools to help developers in finding the needs of large-scale cloud applications, especially related to topographical allocation of resources and workloads for computing. To discourse this gap, the Cloud Analyst has been propounded by authors. Cloud Analyst is a tool intended to simulate large-scale cloud applications to explore and examine the behavior of allotted resources and workload of the users. Cloud Analyst provides the valuable insights for improving the distribution of applications across infrastructures (cloud) and using value-added services for the developers. Naser et al., (2012)[80] premeditated various types of load balancing algorithms including Throttled algorithm which treats the VMs with distinct two values, and these values can be sent to the remote VMs as well to intended VMs. Authors has made some modifications in the existing Throttled algorithm for improving the performance in regard to Fault tolerant, Process migration, and Overload Rejection. Gopalakrishnan et al. (2014)[35] have described that Resource adjudication and allotment are condemnatory management challenges in CC, as IT services must be rigged based on contribution models personalized to the clients' computing needs. Guaranteeing optimal usage of these resources limns a paramount challenge, and negligence to do so may lead to degradation of performance CC System. The author highlights the dynamic distribution and effective usage of resources within cloud architecture, stressing its significance for preserving system performance and fulfilling client needs. Tian W. et al. (2011) [106] have introduced a new algorithm abbreviated as DAIRS and named as Dynamic and Integrated Resource Scheduling algorithm.

Usually the basic load-balance algorithms consider only one parameter like the load of CPU in physical servers, but DAIRS integrates network bandwidth and Memory. The parameter Minimum Compilation Time (MCT), limned by Kim et al. (2013)[60], emphasizes on optimizing both ready-to-execute and expected execution time for attaining effective load balancing, allocating tasks to the core which defines the shortest completion time. Soni et al. (2014) [102] defined a CLB for optimal response times during load balancing within the CC environment. Similarly, Haidri et al. (2014) [39] have proposed a heuristic approach based load-balanced scheduling model for utilizing the CLB for the optimal execution of allocated Tasks. Furthermore, Rana et al. (2014) [86] have adorned different soft computing techniques, including GAs, Artificial Bee Colony, Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO), which are extremely impressive for load balancing in CC environments. Naha et al. (2016) [78] have stated three distinct cloud brokering algorithms like Cost Aware Brokering, Load Aware Brokering and Load Aware over Cost Brokering along with load balancing algorithm. Authors have reported simulation results and propounded that proposed algorithm reduces the cost as well as also improves the performance parameters like average response time, minimum response time, minimum and maximum DC processing time etc. Chaczko et al. (2011) [18] have described that accessibility of cloud systems is one of the main concerns of cloud computing. Authors also stated that the load balancing technique is being employed across various DCs to confirm the network availability by reducing the use of hardware, software failures and alleviating recourse restrictions. Ahmed et al. (2012) [7] have laid stress for appropriately managing the resources offered by the service provider, it is required to balance the load of the tasks that are being submitted to the cloud service provider.

Numerous algorithms have been developed for addressing this task. In this manuscript, a comprehensive juxtaposed analysis of distinct kinds of load balancing policies by using the Cloud Analyst tool has been discussed. Kaur J. et al. (2017) [51] have presented scheduling algorithms which are useful for maintaining the load balancing and offers enhanced strategies through well-organized job scheduling and improved resource allocation techniques. The load may be considered as memory capacity, CPU load, network or delay load. Katyal M. et al. (2014)[49] have explained that load balancing is realized across various DCs for enhancing network availability with least dependence on computer hardware as well as reducing software failures, and alleviating resource constraints. Kansal et al. (2012) [48] have revealed that load balancing encounters a significant challenge in CC, as it is necessary to disseminate dynamic workloads across multiple nodes by preventing any single node from overburdening. It is also useful for optimal usage of resources as well as enhancing the overall performance of the system. Haryani N. et al. (2014) [41] limelight the particular performance issues related with business applications, and which can adversely effective for the organization's for improving overall performance. Recent research specifies that organizations may incur extensive revenue losses due to the delays for surpass recognized performance baselines. The proposed work highlights the numerous problems in the IT industries such as delay or response time i.e. overall response time with the data center processing time. Shahid et al. (2020) [94] were major objective is to discuss the existing Load Balancing techniques, its parameters such as power savings, throughput, overhead, migration time, fault tolerance, resource sharing, scalability, response time etc. along with problems faced by researchers while implementing the algorithms. Authors have reported that Conventional Load Balancing algorithms cannot improve the performance until FT efficiency metrics are being incorporated. James et al. (2012) [44] have analyzed distinct load balancing algorithms for the virtual machines. Secondly, author has developed a novel VM load balancing algorithm for an IaaS platform in which environment it is simulated, and used for CC. Weighted Active Monitoring Load Balancing (WAMLB) Algorithm' uses Cloud Sim to disseminate the load requests uniformly among the accessible VMs, thus, allocating weights to these factors which may lead for enhancing the performance related to data processing efficiency and response time. The author (Mondal B. et al., 2012) [77] have explained Cloud Computing as a platform and Infrastructure applications. CC platform vigorously facilitates, configures/reconfigures the servers as per requirement.

The servers within the cloud may comprise of either physical machines (PMs) or virtual machines (VMs) disseminated across the network. In this paper, they have proposed load balancing methods based on a soft computing technique. A local optimization technique Stochastic Hill climbing is proposed for allocation of incoming tasks to the VMs. Proposed algorithms' performance is being analyzed in terms of quality and quantity both. Kaur S. et al. (2018)[52] have explained the appropriate Load Balancing leads to minimize the scalability, resource intake, enabling, averting bottlenecks, employing fail-over etc. Authors have explained the formula for balancing the burden among virtual products. Jain R. et al. (2017) [42] have limned a service broker (cloud-based) algorithm which explains the intercession to check suitable service providers for appropriate trade-off between performance and price. Authors have anticipated various accessible service broker algorithms and distinct cloud deployment models for the minimal cost and enhanced performance at the same time. Sharma et al. (2018)[97] have demonstrated Cloud computing is growing rapidly with global resource scheduling, reliability, fault tolerance and load balancing. cloud security can be a hurdle in the way to adopt these services. Kaur D. et al. (2019) [50] have propounded shared resources and data to the computers and additional devices as per requirement in cloud computing. Task scheduling is being vitally performed, controls the assigned tasks by keeping the balance between the performance and quality of service parameters. John et al. (2011) [45] have investigated and analyzed CIM (Cloud Inventory Manager) and CPM (Cloud Power Manager) in connection with CSM (Cloud Service Broker) and reported that CSB keeps the track of inventory information along with the hardware resources. The NIST broker tracks and keeps the record of use of cloud services including its performance. Jrad et al. (2012) [46] have expounded a broker which is capable to confirm the client service needs in connection to functional/non-functional SLA parameters. Tordsson et al. (2012) [107] have limned a cloud broker which is accountable for monitoring and ideal placement as well as organisation of virtual resources. Kertesz et al. (2014) [54] have anticipated an employment of SLA based SSV architecture for distinct cloud environments.

He has classified the services as Metabroker and Broker. Broker interacts with the virtual and physical resources, whereas, Meta-broker deals with the supervision of the specific user. Kessaci et al. (2013)[56] have presented the reduction in overall cost as well as response time required for the placement of VM request using Genetic Algorithm. Radi et al. (2021) [84] have projected an improved and economical service broker policy based on VIKOR. Ge Yaozhong et al. (2023) [31] discussed Memory Sharing Control Algorithms for the live migration of VMs to manage memory overload in Physical Machines. The authors implemented this approach by swapping inactive memory pages to access remote memory resources. The reported performance metrics included a throughput of 929 Mbps and a latency of 1.3 μs. Lan Wenjing et al. (2018) [67] proposed a load balancing strategy algorithm to achieve uniform load distribution by transferring load from heavily-loaded controllers to lightly-loaded ones. They optimized load information notification frequency with an adaptive load collection algorithm and achieved a maximum throughput of 17,000 packets per second (pps). Afzal Shahbaz and Kavitha G. (2019) [4] have explored the affairs of load imbalance signifies a multilayered challenge with multiple restrictions, which can meaningfully impair the performance and effectiveness of resources if not efficiently managed. Rajkumar S. and Katiravan Jeevaa (2023) [85] portrayed an efficient Genetic Algorithm (GA) developed for managing the load in hybrid cloud environments. Authors' research concentrated on robotising VM services through load balancing for enhancing performance, including VM placement, resource usage. The proposed algorithm described a high accuracy (Approx. 95%) in satisfying overall capacity necessities. Laalaoui Yacine and Al-Omari Jehad (2018) [66] introduced algorithms, namely Direct Move Heuristic (DMH) and Iterative Direct Move Heuristic (IDMH), for reallocating VMs in IaaS CC platforms. They conducted two experimental studies to validate their approach.

The initial study evaluated small-scale problem instances, demonstrating the applicability of their model and assessing its efficiency. The second study focused on larger problem instances to evaluate the scalability performance of the IDMH heuristic. The results exhibited strong manageability performance, particularly handling problem instances up to 800 VMs effectively. Liu Zhiyu et al. (2021) [71] expounded a port-based forwarding load balancing scheduling (PFLBS) method for DCNs. Authors projected on various vital innovations: Firstly, they described a port-based source-routing addressing scheme, clarifying switch issues and reducing the requirement for operations (table-lookup). After that, they designed an operative routing mechanism rooted on addressing scheme to recognize many available paths for movements. Thirdly, they suggested an effective methodology for dynamically scheduling huge flows related to current link usage ratios. This method is especially good for cloud DCNs and edge computing environments, focusing to reduce switch complexity and overall network power usage. Zhang Yanfeng and Wang Jiawei (2024) [123] limned an Enhanced Whale Optimization Algorithm (EWOA) by devising Lévy flight with WOA. This amalgamation of Lévy flight focusses to explore the forage space of WOA. EWOA has described good performance over other methods, mainly in resource consumption, energy utilization, and processing cost. Mao Li et al. (2023) [74] limned an Enhanced Whale Optimization Algorithm (EWOA) by devising Lévy flight with WOA. This amalgamation of Lévy flight focusses to explore the forage space of WOA. EWOA has described good performance over other methods, mainly in resource consumption, energy utilization, and processing cost. Sasikala P. (2012) [93] has highlighted the emerging paradigm of cloud computing as a beneficial tool for e-Governance. The study discusses how cloud computing standards and architectures can help in developing effective e-Governance strategies to achieve e-Government. Despite the slow adoption of Information Technology by governments, the implementation of e-services can provide cost-effective solutions that enhance government productivity and drive economic growth. Chen Ming and Huang Haifeng (2018) [21] have detailed a cross task load balancing strategy known as CTLB (Cross Task Load Balancing). In CTLB, resource overheads for each task are re-estimated at the start of each task, and resources are re-allocated based on the number of tasks. This strategy leverages the "time continuity" feature, making it a dynamic load balancing approach. Singh Neelam et al. (2023) [101] have proposed an innovative scheduling mechanism for receptacles in big data applications, based on Docker Swarm and Microservice architecture.

This mechanism utilizes Docker Swarm to manage workloads and service discovery efficiently. The results show that by employing microservices in containerized environments, the cumulative workloads of big data applications can be efficiently accomplished, and load balancing is efficiently achieved using Docker Swarm. Qian Jiarui et al. (2023) [83] have proposed a load-balancing Docker scheduling mechanism based on OpenStack. This mechanism employs a specific limitation strategy for container resources and a centralized scheduling strategy. It creates unique weights for containers via filtering stage, a weight adaptation stage, and a non-uniform memory access (NUMA) lean stage. Experimental findings demonstrate a significant reduction in resource load unevenness by 57.35% and 59.00% on average (within a node), and a reduction in average imbalance between nodes by 53.53% and 50.90%. Zhang Zhen et al. (2024)[122] have introduced Graph Neural Network-enhanced Elite Particle Swarm Optimization, termed GraphEPSO. In this method, a Directed Acyclic Graph (DAG) is constructed to model complex tasks, with a Graph Neural Network (GNN) used to encrypt useful data about task sets and unevenly distributed resources. Subtasks/independent tasks are treated as fundamental task units, and physical or virtual devices are considered resource units. The experimental results reported by authors exhibited the supremacy of GraphEPSO juxtaposed to existing baseline methods across the evaluated metrics. Harrath Youssef et al. (2019) [40] have given a solution to address the NP-hard problem of minimizing makespan and reducing the total execution cost of tasks using a multi-objective genetic algorithm. Their approach introduces novel crossover and mutation operators tailored for this purpose. Experimental results demonstrate the effectiveness of the genetic algorithm in achieving efficient solutions in terms of makespan across problem instances of varying sizes, comparing favorably to established lower bounds. Zhao Dongfang et al. (2018)[124] have proposed advancements in cloud services by enhancing next-generation container schedulers to prioritize application performance. Their method introduces a novel approach that consolidates the trade-off between load balancing and application performance, which is efficiently addressed using statistical methods. Ahmad Saima Gulzar et al. (2023)[6] have introduced an algorithm focused on Cost Optimization based on Task Deadline, prioritizing cost efficiency without sacrificing response time.

In their approach, they treat task deadlines as restrictions and chosen the most suitable DC for task accomplishment. The algorithm is designed for efficient runtime decision-making with low complexity. Their experimental results demonstrate an average cost reduction of 35% while ensuring that response times are maintained. Rostami Mohammad et al. (2024)[90] conducted an extensive review on Quality of Service (QoS)-aware load balancing methods within SDN-based IoT networks. According to their findings, comprehensive research encompassing all QoS aspects in load balancing is currently lacking in this field. They emphasize the importance of QoS performance parameters like availability, fault tolerance, and reliability. The authors provide a detailed discussion and comparison of various load balancing techniques, offering an overview of the latest techniques for future research in this area. Albdour Layla (2017)[8] has drawn a comparison that primarily focuses on the distribution of processing power and workload among virtual machines. The paper utilizes the Cloudsim simulation tool to evaluate distinct scenarios, specifically analyzing the metrics of makespan, average turnaround time, bandwidth utilization, and CPU utilization. Kherbache Vincent et al. (2017)[58] have introduced mVM, an innovative and extensile migration scheduler designed to offer schedules with nominal completion time. mVM optimizes the migration process by parallelizing and sequential zing migrations based on network topology as well as memory workload. Realized as a plugin for BtrPlace, it leverages its library to manage temporal/energy concerns. The authors report that mVM has reduced each migration period by an average of 20.4% and the completion time by 28.1%. Ashawa Moses et al. (2022)[11] have proposed an LSTM algorithm that provides an intuitive dynamic resource apportionment system. This system analyzes the heuristics of application resource usage for determining the optimal additional resources needed for each application. The authors also investigated Long-Short Term Memory (LSTM) and Monte Carlo Tree Search, comparing their efficiencies. They found that the proposed approach effectively resolves issues related to consistent traffic patterns, improving the accuracy rate by 10-15%. Zhou Jincheng et al. (2023)[125] have conducted a comparative analysis of various metaheuristic load balancing algorithms, focusing on metrices i.e; resource utilization, data center processing time, makespan time, flow time, degree of imbalance and response time. Subramanian Thiruselvan and Savarimuthu Nickolas (2016)[103] have have projected an innovative cloud brokering architecture which offers an optimum disposition plan for placing resources across different clouds. Proposed Model emphases to elect the best cloud services at minimal costs, assuming distinct attributes described in the SMI. The proposed cloud brokering architecture is modeled with mixed-integer programming formulation and also utilizes the Benders decomposition algorithm for efficient problem-solving.

Gond Sunita et al. (2019) [34] have concentrated on resource allocation in the cloud using the Teacher Learning Based Optimization (TLBO) approach. TLBO, a genetic algorithm, seeks the optimal placement of processes. Key information for TLBO analysis includes the count of machines, memory requirements, and processing times. The TLBO output serves as training input for an Error Back Propagation Neural Network, enhancing the quality of job sequencing. Results demonstrate that the proposed model significantly improves evaluation parameters across different scales compared to existing approaches. Yu Lei et al. (2016) [117] introduced a stochastic load balancing scheme focused at providing probabilistic guarantees against resource overloading through VM migration. Authors' approach minimizes total migration overhead and assesses migration costs independently of network topology considerations. Saravanan G. et al. (2023)[92] have introduced the Improved Wild Horse Optimization (IWHO) algorithm, designed to address challenges such as prolonged scheduling times, high-cost consumption, and heavy virtual machine loads. Their approach, known as IWHOLF-TSC, integrates the Horse Optimization (WHO) algorithm with Levy flight theory (LF). This hybridization constructs a multi-objective fitness function aimed at minimizing Makespan and maximizing resource utilization. Xiao Lei et al. (2023) [115] conducted an extensive review highlighting the diverse applications of cloud computing particularly in the industries related to the sports. Their focus spans athlete performance tracking, operations management, fan engagement, event hosting and sports marketing. This comprehensive analysis aims to enhance understanding of the current landscape and stimulate continued research and innovation in leveraging cloud computing applications within the sports industry.

Sahana Sudipta et al. (2020) [91] have explained an efficient load balancing technique utilizing the weighted Round-Robin algorithm, designed to distribute client requirements among multiple servers with minimal response time. In light of these advancements, a cloud-based dynamic load balancer addresses the challenges of load balancing within cloud infrastructure. Sulimani Hamza et al. (2024) [104] introduced the Hybrid Offloading (HybOff) algorithm, which advances load balancing and resource usage in fog networks by integrating clustering theory. Proposed approach aims to streamline and cost-effectively optimize offloading processes for IoT applications. The Experimental findings using the iFogSim simulation tool reported HybOff's effectiveness in reducing offloading messages and distances, minimizing decision-offloading consequences, and enhancing load balancing by 97%, State-of-the-Art (SoA) and Proof-of-Concept (PoA) by significant margins. Moreover, it boosts system usage by an average of 50% and improves system performance 1.6 times and 1.4 times more than SoA and PoA. Yunlong Fan and Jie Luo (2024) [118] have introduced strategies integrating AI, game theory, and blockchain to promote economic sustainability within the cloud ecosystem. Their work illuminates how these technologies can collectively address issues in stimulating cloud services, thereby enhancing the efficiency and sustainability of cloud computing.

The authors have discussed multi-criteria decision-making technique for the optimized results. It has been perceived from the reported results like response time, processing time and low cost that low cost is suitable for the cloud environment.

Table 1: Comparison of Load Balancing and Service Broker Algorithms

Reference	Algorithm	Criterion	Metrics Evaluated	Advantages	Disadvantages
Gujral et al. (2017) [37]	Round-Robin	Same Time Quantum allocated to each Node	Average Response and Data Processing Time	Uniformity in load balancing with minimum complexity	Increased context switching may result in reduced throughput.
Zamri, A. H. et al. (2023) [119]	Equally Spread Current Execution Algorithm	Equal Load assigned to all VMs	Throughput and optimal overall response time (8.631s)	Ease of implementation	There may be an added computational overhead associated with recurrently parsing the queue
Moly et al. (2019) [76]	Weighted Round Robin Algorithm	Equal Load assigned to all VMs with weights	Average Waiting and Turnaround time	Distribute the load as per assigned weights	To assume servers are similar to manage the load
Narale S.	Throttled	Search an	DC transfer	Uniformity or	Computational

A.et al. (2018) [79]	Load balancing	appropriate VM to perform the recommended job	cost, total VM cost, DC processing time and decreased response time	Consistency in load distribution across each node. to minimise DC transfer cost, total VM cost, DC processing time and reduce response time	overhead required for identifying a suitable VM
Madni S. H. H. Et al. (2017) [72]	Less Execution Time: Heuristic Technique	Allocating the tasks to the VM using the Least Execution Method	Completion and Execution Time is being reduced	Applicable for static and dynamic approaches	Prolonged starvation
Chen H. et al. (2013) [20]	Max-Min	Task with large size and minimum VM capacity	Makespan and Resource Utilisation	Appropriate for compact distributed systems	Jobs with least completion time may starve
Aggarwal M. et al. (2016) [5]	Genetic	Chromosomes and population play a vital role in enabling the processes of selection, crossover and mutation	Overall Response Time	Improved performance and Efficiency	More time-intensive and complex
Glover and Laguna (2013) [33]	Tabu Search: adaptive memory programming	Meta-heuristic approach	Computational Cost	Better Performance	Requires more time for accomplishment of allocated tasks
Khaleel M. et al. (2013) [57]	A-Star Search	Hybrid approach amalgamating breadth-first and depth-first search is employed, with	Bandwidth and Data Packet Size	Enhanced Performance	More overheads are required

		two distinct sheets created: one for prioritizing tasks and other for managing processing capacity			
Aruna M. et al. (2019) [10]	Switching Central Load Balancer	Switching of Jobs	Response Time	Good Efficiency	There is a significant likelihood that allocated tasks may not be accomplished within the specified time
Sharma T. et al. (2013) [96]	Central Load balancer	Heterogenous physical server	Response Time	Less Response Time	Less Dynamic
Xu M. et al. (2016) [116]	Virtual Machine Assigned Load Balancer	Least loaded VM	Resource Utilisation, Scalability and Response Time	Better response time and fairness	For mixed loads (Static and Dynamic)
Payaswini P. et al. (2021) [81]	Performance Optimized Routing	Maintains an index for Data Centre (DC)	Processing Time and Resource Utilisation	Identify nearest DC based on Latency	Least response time
Khodar A. et al. (2020) [59]	Dynamic Service Broker	Maintains two lists one for data centers and other for best response time	Total Response and Execution Time	Response time is better	Sometimes results are not good as it gets overburden by managing two lists parallelly
Shao S. et al. (2014) [95]	Random Switching Traffic Scheduling	Optimal balanced Meter Data Collection Tree	Packet Loss and Average end-to-end delay	Packet loss ratio of the burst data and release is being reduced	Failures of single or some more smart meters may affect the performance of algorithm
Ge Yaozhong	Memory Sharing Control	memory allocation based on NVMe over fabric NoF	Throughput (929 MBps) and latency	Manage memory overload	Live migration can use a significant

et al. (2023) [31]	Algorithms		(1.3 μs)	without interrupting or suspending active applications	amount of network bandwidth, which may impact other network-dependent services and applications
Lan Wenjing et al. (2018) [67]	dynamic / adaptive load balancing strategy (load informing strategy)	Hierarchical control plane for distributed controllers in SDN (via switch migration)	Throughput (17000 pps)	Dynamically transfer the load from the overburdened controller to the less burdened and maintains the uniformity	Shifting load requires transferring data and control information between controllers, potentially leading to increased network overhead and latency.
Rajkumar S. and Katiravan Jeevaa (2023) [85]	Intelligent Genetic Algorithm	automating VM services through runtime resource provisioning	Accuracy rate (95%) and Optimal Resource Utilisation	High Accuracy rate	Increased management complexity of VMs and the entire cloud infrastructure
Laalaoui Yacine and Al-Omari Jehad (2018) [66]	Direct Move Heuristic (DMH) and Iterative Direct Move Heuristic (IDMH)	Reallocating VMs in IaaS CC environment	success rate and solution quality	Better performance in terms of solution, success rate, quality and scalability	Operational Disruption and Complexity
Liu Zhiyu et al. (2021) [71]	Port-based forwarding load-balancing scheduling algorithm	efficiently distributing traffic among network ports to optimize throughput and minimize	Throughput and Completion Time	Reduces flow completion time and enhances average throughput	Latency and Complexity

		congestion			
Zhang Yanfeng and Wang Jiawei (2024) [122]	Enhanced Whale Optimization Algorithm	Lévy flight strategy with the standard Whale Optimization (nature-inspired metaheuristic optimization technique) Algorithm inspired by social behavior of humpback whales during bubble-net feeding	Resource Usage, Energy Consumption and Execution Cost	Exhibits optimal Resource Utilization and Energy Consumption	Scalability Issues
Chen Ming and Huang Haifeng (2018) [21]	cross task load balancing strategy	time continuity	resource allocation and task segmentation implementation	Enhanced Resource Allocation and Reliability	Complex Implementation, Overhead Cost and Latency issues
Ahmad Saima Gulzar et al. (2023) [6]	cloud/fog environment based on Task Deadline	cost is being optimised without any compromise for response time	Execution Cost and Response Time	Reduces the cost by 35% on average while managing response time	Complexity in Deadline Management
Gond Sunita et al. (2019) [34]	Teacher Learning Based Optimization (TLBO) Approach	Genetic Algorithm	Makespan, Total Flow Time and Total Time Variance	Reduced Makespan by 8.7715% and Total Flow time by 2.55%	Complexity in Parameter Tuning and slower convergence rates
Yu Lei et al. (2016) [117]	VM migration based load balancing approach	stochastic load balancing by considering VMs as random variables	total migration overhead	To minimize migration costs for load balancing, it is essential to take into account the network topology and enhance the system's resilience against	Determining the destination of VM migrations requires consideration of the bandwidth available on the paths between the destinations and other VMs utilized by the application.

				potential hotspots, thereby improving its worst-case performance.	
Saravanan G. et al. (2023) [92]	Improved Wild Horse Optimization (IWHO) algorithm	Concoction Horse Optimization (WHO) algorithm with Levy flight theory (LF)	Makespan and resource Utilisation	minimizing Makespan and maximizing resource utilization	Complexity, Scalability and Parameter Tuning
Sahana Sudipta et al. (2020) [91]	weighted Round-Robin algorithm	Weight Assignment, Round Robin Selection and Weighted Distribution	throughput, efficiency and response time	throughput, efficiency and response time which regulates the degree of performance have exhibited high degree of precision and accuracy	Complexity of Weight Assignment, Inability to Handle Bursty Traffic and Dependency on Initial Configuration
Sulimani Hamza et al. (2024) [104]	Hybrid Offloading (HybOff) algorithm	Task Offloading Strategy along with Dynamic Decision Making	offloading messages and distances	enhancing load balancing by 97% it boosts system usage by 50% and improves system performance 1.6 times (SoA) and 1.4 (PoA)	Resource Intensive, Dependency on Simulation Tools, Implementation and Integration Issues
Ezugwu Absalom E. et al. (2013) [29]	Mapping algorithm for Virtual Machines	Principles of set theoretic	waiting time, context switching, response time and turnaround time	Robotically adjust the allocation of resources between VMs and physical hosts	Efficiently managing VM allocation with minimal PMs necessitates advanced tools for monitoring

				Cloud users can efficiently run their VMs with a limited number of PMs	and managing resources to optimize performance and utilization effectively.
Kumar M. et al. (2018) [61]	Conventional Non Classical	Heuristic: Classical, Deterministic	Makespan (decreased) and Task Acceptance ratio increased	Capable of managing substantial workloads within specified deadlines Improved Elasticity	Tasks after deadline are declined for execution. Experiments are conducted exclusively using a space-sharing policy only
Xiao Z. et al. (2017) [114]	Fairness Aware Algorithm	Based on non-cooperative game theory	Expected Response Time and Fairness Index	Expected Response time reduced	Task Execution time is very high
Ashouraei M. (2018) [12]	hybrid Ant colony-honey method and a Round-Robin (RR) based load balancing Algorithm	Optimization: nature Inspired	Energy usage and Migration Rate	Optimal use of resources. Minimal energy consumption	Limited scalability and fault tolerance
Adhikari M. (2018) [3]	Heuristic-based load-balancing algorithm	Optimization: Linear programming Based Technique	Makespan, Scheduled Length Ratio, Waiting Time, VM utilization and CPU utilisation	Good makespan and Resource utilisation	The quality of service is degraded
Kumar M. (2017) [62]	Dynamic Load Balancing Algorithm	Non Classical, Deterministic	Makespan (912 for 50 No. of Tasks) and utilization ratio	Good makespan and Resource utilisation	Limited fault tolerance and energy efficiency
Tang L. (2017) [105]	Bacteria Foraging Optimization	Optimization	Time v/s Run Time Number	Minimal VM downtime, execution and	Limited Scalability, Throughput and

	and Lamarack Evolutionary Theory			transfer time	Resource Utilization
Vanitha M. (2017) [111]	dynamic well-organized load balancing (DWOLB) algorithm	Genetic Algorithm : Nature Inspired	Response Time and Makespan	Decrease in response time and makespan	Limited Scalability, Throughput and Resource Utilization
Tripathi A.M. (2018) [108]	Active Monitoring	Heuristic	Response Time and Overhead	Decrease in response time and Overhead. Efficient Resource Utilisation	Limited Throughput High Service Level Value
Mathur S. (2017) [75]	ASA Max-Min algorithm	maximizing fairness and minimizing the disparity in load distribution	Throughput and Scalability	High Throughput and Scalability Limited Fault Tolerance and Overhead	Limited Resource Utilisation and High Makespan
Singh A. N. (2018) [100]	Weighted Active Monitoring Load Balancing	VM selected on the bases of their weights for execution of the task	Average Overall response time (217.18ms), Overhead, Markspan and Resource Utilisation	Reduced overhead and makespan Maximizing resource utilization	Limited Throughput
Abdelaziz Kella (2014)	Stable Matching Algorithm	Using the Coase theorem to ascertain the optimal count of VMs for migration to minimize costs	Response Time and Energy usage	Enhance datacenter energy efficiency	Executing live migration and powering off idle PMs carries inherent operational risks
Remesh Babu et	Enhanced bee	Nature Inspired: Honey bee	Resource Consumption	Minimize resource	Limited Scalability

al. (2016) [89]	colony– based load balancing	Method for minimizing resource usage and response time	and Response Time	consumption and response time, Limited number of Task Migrations	Complexity
Devi and Uthariaraj et al. (2016) [26]	Weighted round-robin technique	assigns weights to servers based on their processing capacities or other relevant metrics	Response and Execution Time	Improved response time of assigned tasks	Homogeneous environment execution
Keshvadi and Faghih (2016) [55]	Multiagent-based load-balancing architecture	Maximizing resources using various agents	Response Time and Resource Utilisation	Reduced response time, improved makespan and resource utilisation	DcM agents need a parent message to initiate their self-destruction process, without a built-in timer for automatic self-destruction.
Elmougy, S. et al. (2017) [28]	Hybrid task scheduling technique	Integration of on unswerving job and round robin with dynamic task quantum	Waiting, Turnaround and Response Time	Minimizing task starvation and waiting times enhances overall response and turnaround times, improving overall system efficiency	Task quantum is less effective
Sharma, T. and Bedi, R. P. (2024) [98]	Pragmatic Load Balancing Algorithm	contains index table for the VMs status and task assigned to VM based on their current status	Overall Response Time, Data Processing Time and Cost	Minimal ORT (207.77ms), DPT (62.88ms) and Cost ($3442.79) (For CCS with 40 VMs)	Limited Adaptability and Scalability Challenges

5 Challenges

Load balancing and Service Broker scheduling has to be vitally taken care of, any mismanagement or problem may lead to drastic change in the chosen conditions. There is also a challenge of optimal allocations of the VMs. Although cloud computing propositions significant prospects to the IT industry, the technology is still in continuous development and faces many unresolved issues (Zhang, Q. et al., 2010) [120]. As the adoption of local cloud computing architectures rises, organizations are increasingly recognizing the issue of power waste caused by underutilized resources [30]. So, these may be considered as paramount part of cloud computing and many of the researchers have reported the problems in the below given sub-sections.

5.1 Nodes Distribution

The data centers are present in the distinct geographical areas where nodes are scattered in these data centers. Due to this unevenly scattering of the node may impact the performance of the scheduled algorithms [27].

5.2 Scalability of Load Balancer

As requirement of cloud services are different for distinct end users, so it should always keep in mind while designing the required algorithms that it should be scalable and easily and quickly balance the load in the data center.

5.3 Failure of Master Node

All the decisions have to be carried out by the Master Node, any failure of this node may lead to disturbance of the whole system.

5.4 Algorithm Complexity

For Accomplishing the particular tasks, the algorithm may be developed in such a simple way that it will be easily implemented. So, it’s always a need of the hour to design simple and easy to implement algorithm.

5.5 Migration of Virtual Machine

In case the system get overburdened means addition VM’s are required to be assigned for smooth functioning of the assigned tasks for that VMs are required to be relocated or migrated. Migration techniques are required to resolve the above issues but sometimes these techniques do not hold the required results [23].

5.6 Automatic Service Provisioning

It is the fundamental feature which has potential to attain and release the resources on-demand. CSP aims to allot and withdraw these resources to meet service level objectives (SLO) while minimalizing operational costs. However, achieving this balance is complex. Specifically, translating high-level SLOs like QoS requirements, into low-level resource needs like CPU and memory is challenging. While automated service rigging is not a current issue, dynamic resource rigging for internet applications has been comprehensively explored by the researchers [110] [121].

5.7 Sever Consolidation

Server consolidation is a powerful strategy for maximizing resource utilization and minimizing energy consumption in a cloud computing environment (Chekuri, C. et al., 2004) [19].

5.8 Energy Management

Enhancing energy efficiency is a critical concern in CC. It is projected that powering and cooling account for 53% of the total expenses of DCs [16].

5.9 Traffic Management and Analysis

ISPs face several challenges in extending existing traffic measurement and analysis methods to data centers. Firstly, the density of links in data centers is much more in ISPs, presenting a worst-case situation for these methods. After that, while most existing techniques can calculate traffic matrices between a few hundred end users, a modular DC can contain numerous servers, complicating the measurement and analysis process [36].

5.10 Data Security

Data security includes various measures and practices aimed at protecting data from unauthorized access, breaches, and other threats while it is stored, processed, or transmitted in a cloud environment. Ensuring data security in the cloud, managed by the infrastructure provider, involves maintaining confidentiality and auditability [47] .

5.11 Software framework

Cloud computing offers a robust platform for hosting large-scale and data exhaustive applications. These applications often utilize frameworks like Hadoop for flexibilty and fault-tolerant data processing. Whereas, the performance of a MapReduce job is extensively reliant on the specific type of application [47].

5.12 Novel Cloud Architecture

Presently, most commercial clouds are realized in big, centralized DCs. While this design provides economies of scale and high manageableness, it also has cons, including high energy costs and significant preliminary investments for constructing the DCs.

5.13 Heterogenous Nodes

In the early stages of cloud load balancing research, the emphasis was principally on homogeneous nodes. However, in cloud computing, user requirements evolve dynamically, requiring the processing of jobs/tasks on non-uniformly distributed nodes to optimize resource usage and reducing response times. Consequently, developing effective load-balancing approaches tailored for assorted environments poses a significant challenge for scientists (Kumar, P. et al., 2019) [63].

5.14 Distinct Faults

Failure typically refers to a situation that results in unexpected behavior or output. In Cloud Computing, besides Master Node failures, various other types of faults can occur at different levels, such as Servers [112], Services [32], and Networks [82]. These faults can lead to different classifications of failures.

Hardware Failure: The crashing of physical devices.

VM Failure: The crashing of logical components residing on VM.

Job Failure: Multiple tasks crashing due to logical dependency.

Task Failure: When one task crashes but the other tasks continue to function.

5.15 Spatial distribution of the cloud nodes

Some methods are developed specifically for closely positioned nodes where communication delays are immaterial. However, designing an efficient load balancing algorithm for spatially distributed nodes remains a challenge [13].

5.16 Administrative functionalities

Although numerous cloud services are accessible, the primary steps include network transformation, infrastructure management, dynamic-resource allocation and dynamic-scaling functionality, which are essential for many organizations. There is significant potential to enhance the robustness and load balancing (LB) functionality provided so far [88].

5.17 Interoperability

The unified platform should consolidate resources seamlessly from other platforms, a concept known as interoperability. Achieving this via web services is increasingly feasible, though designing these services remains complex [65].

5.18 Portability

Applications operating on one cloud platform should seamlessly transition to another cloud platform without requiring layout or programming changes. However, achieving portability is challenging due to each cloud provider using different standard languages for their frameworks [123].

5.19 Handling Data

Cloud computing (CC) has tackled the challenges posed by traditional storage devices, which were costly in terms of resources and equipment. The cloud enables users to store data assertedly, eliminating control issues. As data storage requirements grow, repetition of stored data becomes essential for improved approachability and data continuity [63].

Many of the researchers have tried their best to address the different kind of challenges such as, Ezugwu Absalom E. et al. (2013) [29] addressed the challenges of VM allocation by developing a set-theoretic based mapping algorithm. Using a Virtual Computing Laboratory framework model within a private cloud, the authors extended the open-source IaaS solution Eucalyptus. Their findings indicate that cloud users can effectively utilize VMs with a limited number of physical machines, resulting in efficient resource utilization. Domanal et al. (2013)[27] delineated a VM assignment algorithm which efficiently removes both underutilization and overutilization of VMs. Authors expounded technique also reports the inadequacies allied with the Active VM algorithm. Church, K. et al. (2008) [22] have explained that smaller data centers are more advantageous than larger ones because they require less power consumption, occupy less space, and need limited cooling systems, all of which contribute to cost reduction. Kumar and Kushwaha (2019) [64] have defined fault tolerance as the system's capability to perform its envisioned functions even in the existence of faults. Similarly, many others have addressed the issues of Real Time Decision Making, Complexity in Multitenancy Environments, Security and Private Concerns, QoS guarantee, adaptability to changing conditions, Fault Tolerance, Optimization Resource Utilization etc. This also requires the system to detect the existence of errors and initiate corrective actions to ensure the expected results are not compromised.

6 Conclusion

This manuscript describes a comprehensive comparative analysis of several existing load balancing and service broker methods/algorithms, highlighting the required performance parameters and associated challenges. It propounds that different algorithms exhibit unique characteristics; for instance, some emphasis on reducing the makespan of the network, while others focus to implement allocated jobs within a stipulated period of time. Additionally, the manuscript systematically juxtaposes the advantages and disadvantages of distinct load balancing and service broker algorithm.

This study will assist researchers in identifying key research challenges within the field of load balancing, offering a comprehensive summary of available findings.

This study will assist researchers in identifying key research challenges within the field of load balancing, offering a comprehensive summary of available findings.

Acknowledgment: Not Applicable.
Funding Statement: The author(s) received no specific funding for this study.
Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. R. Abdul, “Service broker based on cloud service escription language,” *IEEE 15th International symposium on parallel and distributed computing*, pp.196-201, 2016.
2. A. A. Adewojo and J. M. Bass, “A novel weight-assignment load balancing algorithm for cloud applications,” *SN Computer Science*, vol.4, no.3, pp. 270, 2023.
3. M. Adhikari and T. Amgoth, “Heuristic-based load-balancing algorithm for IaaS cloud,” *Futur Gener Comput Syst*, vol.81, pp.156–165,2018.
4. S. Afzal and G. Kavitha, “Load balancing in cloud computing–A hierarchical taxonomical classification,” *Journal of Cloud Computing*, vol.8, no.1, pp.1-24, 2019.
5. M. Aggarwal et al. “A genetic algorithm inspired task scheduling in cloud computing,” *International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, 2016.
6. S. G. Ahmad, T. Iqbal, E. U. Munir and N. Ramzan, “Cost optimization in cloud environment based on task deadline,” *Journal of Cloud Computing*, vol.12, no.1, pp.9, 2023.
7. T. Ahmed and Y. Singh, “Analytic Study of Load Balancing Techniques Using Tool Cloud Analyst,” *International Journal of Engineering Research and Applications*, vol.2, no.2, pp.1027-1030, 2012.
8. L. Albdour, “Comparative study for different provisioning policies for load balancing in cloudsim,” *International Journal of Cloud Applications and Computing (IJCAC*), vol.7, no.3, pp.76-86, 2017.
9. I. S. AlShawi, L. Yan, W. Pan and B. Luo, “Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm,” *IET Conference on Wireless Sensor Systems (WSS 2012)*, London, vol.1-6, 2021.
10. M. Aruna et al. “Load balancing in cloud environment with switching mechanism and token-based algorithm,” *International Journal of Public Sector Performance Management*, vol.5, no.2, pp.123-133,2019.
11. M. Ashawa, O. Douglas, J. Osamor and R. Jackie, “RETRACTED ARTICLE: Improving cloud efficiency through optimized resource allocation technique for load balancing using LSTM machine learning algorithm,” *Journal of Cloud Computing*, vol.11, no.1, pp.87, 2022.
12. M. Ashouraei SN. Khezr and R. Benlamri, NJ. Navimipour, “A new SLA-aware load balancing method in the cloud using an improved parallel task scheduling algorithm,” *In: 2018 IEEE 6th international conference on future internet of things and cloud (FiCloud)*, pp 71–76, 2018.
13. K. G. Bakde and B. M. Patil, “Survey of techniques and challenges for load balancing in public cloud,” *International Journal of Technical Research and Applications*, vol.4, no.2, pp.279-290, 2016.
14. Z. Benlalia, “A New service broker algorithm optimizing the cost and response time for the cloud computing,” *International Symposium on Machine Learning and Big Data Analytics for Cyber Security and Privacy*, vol.151, pp.992-997, 2019.
15. R. Buyya, “CloudAnalyst: A CloudSim-based tool for modelling and analysis of large scale cloud computing environments,” *Distrib. Comput. Proj. Csse Dept. Univ. Melb*, pp.433-659, 2009.
16. T. Chaabouni and M. Khemakhem,“Energy management strategy in cloud computing: a perspective study,” *The Journal of Supercomputing*, vol.74, pp.6569-6597, 2018.

17. G. S. Chabbra, “Qualitative Parametric Comparison of Load Balancing Algorithms in distributed Computing Environment,” *IEEE 14th International Conference on Advanced Computing and Communication*, Surathkal, pp.58-61, 2006.
18. Z. Chaczko, “Availability and Load Balancing in Cloud Computing. International Conference on Computer and Software Modeling,” *Singapore*, vol.14, pp.134-140, 2011.
19. C. Chekuri and S. Khanna, “On multidimensional packing problems,” *SIAM journal on computing*, vol.33, no.4, pp.837-851, 2004.
20. H. Chen, F. Wang, N. Helian and G. Akanmu, “User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing,” *In 2013 national conference on parallel computing technologies (PARCOMPTECH)*, pp. 1-8, 2013.
21. M. Chen and H. Huang, “Cross-Task Dynamic Load Balancing Strategy,” *In 2018 IEEE 3rd Optoelectronics Global Conference (OGC)*, pp. 39-42, 2018.
22. K. Church, A. G. Greenberg and J. R. Hamilton, “On Delivering Embarrassingly Distributed Cloud Services,” *In HotNets*, pp. 55-60, 2008.
23. C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach and A. Warfield, “Live migration of virtual machines,” *In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, vol.2, pp. 273-286, 2005.
24. K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal and Dam, “A genetic algorithm (ga) based load balancing strategy for cloud computing,” *Procedia Technology*, vol.10, pp.340-347, 2013.
25. A. V. Dastjerdi and R. Buyya “Compatibility-aware cloud service composition under fuzzy preferences of users,” *IEEE Transactions on cloud computing*, pp.1-13, 2014.
26. D. C. Devi and V. R. Uthariaraj, “Load balancing in cloud computing environment using improved weighted round robin algorithm for nonpreemptive dependent tasks,” *The scientific world journal*, vol.2016, no.1, pp.3896065, 2016.
27. S. G. Domanal, “Load Balancing in Cloud Computing using Modified Throttled Algorithm,” *IEEE International Conference: Cloud Computing in Emerging Markets (CCEM)*, pp.1-5, 2013.
28. S. Elmougy, S. Sarhan and M. Joundy, “A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique,” *Journal of Cloud computing*, vol.6, pp.1-12, 2017.
29. A. E. Ezugwu, S. M. Buhari and S. B. Junaidu, “Virtual machine allocation in cloud computing environment,” *International Journal of Cloud Applications and Computing (IJCAC)*, vol.3, no.2, pp.47-60, 2013.
30. J. M. Galloway, “Power aware load balancing for cloud computing,” *Proceedings of the world congress on engineering and computer science*, vol.1, pp.19–21, 2011.
31. Y. Ge, Y. C. Tian, Z. G. Yu and W. Zhang, “Memory sharing for handling memory overload on physical machines in cloud data centers,” *Journal of Cloud Computing*, vol.12, no.1, pp.27, 2023.
32. P. Gill, N. Jain and N. Nagappan, “Understanding network failures in data centers: measurement, analysis, and implications,” *In Proceedings of the ACM SIGCOMM 2011 Conference*, pp. 350-361, 2011.
33. F. Glover and M. Laguna, “Tabu Search,” *Springer New York*, pp.3261-3362, 2013.
34. S. Gond and S. Singh, “Dynamic load balancing using hybrid approach,” *International Journal of Cloud Applications and Computing (IJCAC)*, vol.9, no.3, pp.75-88, 2019.
35. T. R. Gopalakrishnan, et al. “A QoS-based routing approach using genetic algorithms for bandwidth maximisation in networks,” *International Journal of Artificial Intelligence and Soft Computing*, vol.4, no.1, pp.80–94, 2014.
36. A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, C., Lahiri, et al., “VL2: A scalable and flexible data center network,” *In Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pp. 51-62, 2009.

37. R. K. Gujral, et al. “Critical analysis of load balancing strategies for cloud environment,” *International Journal of Communication Networks and Distributed Systems*, vol.18, no.3/4, pp.213, 2017.
38. Y. Gupta, “Novel distributed load balancing algorithms in cloud storage,” *Expert Systems with Applications*, vol.186, pp.115713, 2021.
39. N. Haryani et al. “Dynamic Method for Load Balancing in Cloud Computing,” *IOSR Journal of Computer Engineering, International Conference on Signal Propagation and Computer Technology (ICSPCT)*, vol.16, no.4, pp.23-28, 2014.
40. Y. Harrath and R. Bahlool, “Multi-objective genetic algorithm for tasks allocation in cloud computing,” *International Journal of Cloud Applications and Computing (IJCAC)*, vol.9, no.3, pp.37-57, 2019.
41. R. A. Haidri, C. P. Katti and P. C. Saxena, “A load balancing strategy for Cloud Computing environment,” *In 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014)*, pp.636-641, 2014.
42. R. Jain et al. “a Review on Service Broker Algorithm in Cloud Computing,” *International Journal of Computer Applications*, vol,159, no.3, 2017.
43. R. Jain, N. Sharma and T. Sharma, “Enhancement in performance of service broker algorithm using fuzzy rules,” *In 2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp.922-925, 2018.
44. D. J. James, “Efficient VM Load Balancing Algorithm for a Cloud Computing Environment,” *International Journal on Computer Science and Engineering*, vol.4, no.9, pp.1658-1663, 2012.
45. R. B. John, “NIST Cloud Computing reference Architecture,” *IEEE World Congress, Washington DC*, pp.594-600, 2011.
46. F. Jrad, J. Tao and A. Streit, “Simulation-based evaluation of an intercloud service broker,” *Cloud Computing*, pp.140-145, 2012.
47. K. Kambatla, A. Pathak and H. Pucha, “Towards Optimizing Hadoop Provisioning in the Cloud,” *HotCloud*, vol.9, no.12, pp.28-30, 2009.
48. N. J. Kansal and I. Chana, “Existing Load Balancing Techniques in Cloud Computing: A Systematic Re-View,” *Journal of Information Systems and Communication*, vol.3, no.1, pp.87-91, 2012.
49. M. Katyal and A. Mishra, “A comparative study of load balancing algorithms in cloud computing environment,” arXiv preprint arXiv:1403, pp.6918, 2014.
50. D. Kaur, et al. “Scheduling Algorithms in Cloud Computing”, *International Journal of Computer Applications*, vol.178, no.9, 2019.
51. J. Kaur, “Various Load Balancing Algorithms for Cloud Computing,” *World Wide Journal of Multidisciplinary Research and Development*, vol.3, no.5, pp.60-63, 2017.
52. S. Kaur, “Efficient load balancing using improved central load balancing technique,” *2nd International Conference on Inventive Systems and Control*, Coimbatore, India, 2018.
53. A. Kella and G. Belalem, “A stable matching algorithm for VM migration to improve energy consumption and QOS in cloud infrastructures,” *International Journal of Cloud Applications and Computing (IJCAC)*, vol.4, no.2, pp.15-33, 2014.
54. A. Kertesz et al. “An interoperable and self adaptive approach for SLA based service virtualization in heterogenous cloud environment,” *Future Generation Computation System*, vol.32, pp.54-68, 2014.
55. S. Keshvadi and B. Faghih, “A multi-agent based load balancing system in IaaS cloud environment,” *International Robotics & Automation Journal*, vol.1, no.1, pp.1-6, 2016.
56. Y. Kessaci et al. “A pareto-based genetic algorithm for optimized assignment of VM requests on a cloud brokering environment,” *IEEE Congress on Evolutionary Computation*, Cancun, pp.2496-2503, 2013.
57. M. Khaleel et al. “Finding a STAR in a vehicular cloud,” *IEEE Intelligent Transportation Systems Magazine Published by Institute of Electrical and Electronics Engineers*, vol.5, no.2, pp.55-68, 2013.

58. V. Kherbache, E. Madelaine and F. Hermenier, “Scheduling live migration of virtual machines,” *IEEE transactions on cloud computing*, vol.8, no.1, pp.282-296, 2017.
59. A. Khodar, “Evaluation and Analysis of Service Broker Algorithms in Cloud Analyst,” *IEEE conference of Russian Young researchers in Electric and Electronics Engineering*, St. Petersburg and Moscow, Russia, pp.351-355, 2020.
60. S. I. Kim, H. T. Kim, G. S. Kang and J. K. Kim, “Using dvfs and task scheduling algorithms for a hard real-time heterogeneous multicore processor environment,” *Proceedings of the workshop on Energy efficient high performance parallel and distributed computing, ACM*, pp.23-30, 2013.
61. M. Kumar, K. Dubey and S. C. Sharma, “Elastic and flexible deadline constraint load balancing algorithm for cloud computing,” *Proced Comp Sci*, no.125, pp.717–724, 2018.
62. M. Kumar and S. C. Sharma, “Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing,” *Proced Comp Sci* vol.115, no.C, pp.322–329, 2017.
63. P. Kumar and R. Kumar, “Issues and challenges of load balancing techniques in cloud computing: A survey,” *ACM computing surveys (CSUR)*, vol.51, no.6, pp.1-35, 2019.
64. S. Kumar and A. S. Kushwaha, “Future of fault tolerance in cloud computing,” *Think India Journal*, vol.22, no.17, pp.359-363, 2019.
65. M. H. Kuo, “Opportunities and challenges of cloud computing to improve health care services,” *Journal of medical Internet research*, vol.13, no.3, pp.e1867, 2011.
66. Y. Laalaoui and J. Al-Omari, “A planning approach for reassigning virtual machines in IaaS clouds,” *IEEE Transactions on Cloud Computing*, vol.8, no.3, pp.685-697, 2018.
67. W. Lan, F. Li, X. Liu and Y. Qiu, “A dynamic load balancing mechanism for distributed controllers in software-defined networking,” *In 2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp.259-262, 2018.
68. F. Larumbe and B. Sanso, “A tabu search algorithm for the location of data centers and software components in green cloud computing networks,” *IEEE Transactions on Cloud Computing*, vol.1, no.1, pp.22-35, 2013.
69. X. Li, Y. Mao, X. Xiao and Y. Zhuang, “An improved max-min task-schedulng algorithm for elastic cloud,” *IEEE International Symposium on Computer, Consumer and Control (IS3C)*, pp.340-343, 2014.
70. F. Lin and H. Ying, “Modeling and control of fuzzy discrete event systems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol.32, no.4, pp.408-415, 2002.
71. Z. Liu, A. Zhao and M. Liang, “A port-based forwarding load-balancing scheduling approach for cloud datacenter networks,” *Journal of Cloud Computing*, vol.10, no.1, pp.13, 2021.
72. S. H. H. Madni et al. “Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment,” *PLOS ONE Journal*, 2017.
73. M. Maheswaran, S. Ali, H. J. Siegal, D. Hensgen and R. F. Freund, “Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems,” *Eighth Heterogeneous Computing Workshop*, Proceedings, San Juan, pp.30–44, 1999.
74. L. Mao, R. Chen, H. Cheng, W. Lin, B. Liu and J. Z. Wang, “A resource scheduling method for cloud data centers based on thermal management,” *Journal of Cloud Computing*, vol.12, no.1, pp.84, 2023.
75. S. Mathur, A. A. Larji and A. Goyal, “Static load balancing using ASA max-min algorithm,” *Int J Res Appl Sci Eng Techno*. 2017.
76. Moly, “Load Balancing approach and algorithm in cloud computing environment,” *American Journal of Engineering Research*, vol.8, no.4, pp.99-105, 2019.
77. B. Mondal, K. Dasgupta and P. Dutta, “Load balancing in cloud computing using stochastic hill climbing-a soft computing approach,” *Procedia Technology*, no.4, pp.783-789, 2012.
78. R. K. Naha and M. Othman, “Cost-aware service brokering and performance sentient load balancing

algorithms in the cloud," *Journal of Network and Computer Applications*, vol.75, pp.47–57,2016.

79. S. A. Narale and P. K. Butey, "Throttled load balancing scheduling policy assist to reduce grand total cost and data center processing time in cloud environment using cloud analyst," *In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp.1464-1467, 2018.
80. A. Naser and J. Joshi, "An Efficient Load Balancing Algorithm for virtualized Cloud Data Centers," *Recent Advances in Electrical and Computer Engineering*, vol.2, no.7, pp.65-71, 2012.
81. P. Payaswini, "Comparative study on load balancing and service broker algorithms in Cloud computing using cloud analyst tool," *International Journal of Next-Generation Computing*, vol.12, no.1, pp.49-61, 2021.
82. S. Prathiba and S. Sowvarnica, "Survey of failures and fault tolerance in cloud," *In 2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, pp.169-172, 2017.
83. J. Qian, Y. Wang, X. Wang, P. Zhang and X. Wang, "Load balancing scheduling mechanism for OpenStack and Docker integration," *Journal of Cloud Computing*, vol.12, no.1, pp.67, 2023.
84. M. Radi, "Efficient and Cost effective Service Broker policy based on user priority in VIKOR for Cloud Computing," *The Scientific Journal of King Faisal University*, vol.23, no.1, pp.1-8, 2021.
85. S. Rajkumar and J. Katiravan, "Virtualized intelligent genetic load balancer for federated hybrid cloud environment using deep belief network classifier," *Journal of Cloud Computing*, vol.12, no.1, pp.138, 2023.
86. M. Rana, S. Bilgaiyan and U. Ka, "A study on load balancing in cloud computing environment using evolutionary and swarm based algorithms," *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kanyakumari, India, pp.245-250, 2014.
87. M. Randles, D. Lamb and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," *IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, pp.551-556,2010.
88. A. Rashid and A. Chaturvedi, "Cloud computing characteristics and services: a brief review," *International Journal of Computer Sciences and Engineering*, vol.7, no.2, pp.421-426, 2019.
89. K. R. Remesh Babu and P. Samuel, "Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud," *In Innovations in Bio-Inspired Computing and Applications: Proceedings of the 6th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2015)*, Kochi, India during, pp.67-78, 2016.
90. M. Rostami and S. Goli-Bidgoli, "An overview of QoS-aware load balancing techniques in SDN-based IoT networks," *Journal of Cloud Computing*, vol.13, no.1, pp.89, 2024.
91. S. Sahana, T. Mukherjee and D. Sarddar, "A conceptual framework towards implementing a cloud-based dynamic load balancer using a weighted round-robin algorithm," *International Journal of Cloud Applications and Computing (IJCAC)*, vol.10, no.2, pp.22-35, 2020.
92. G. Saravanan, S. Neelakandan, P. Ezhumalai and S. Maurya, 'Improved wild horse optimization with levy flight algorithm for effective task scheduling in cloud computing," *Journal of Cloud Computing*, vol.12, no.1, pp.24, 2023.
93. P. Sasikala, "Cloud computing and e-governance: Advances, opportunities and challenges," *International Journal of Cloud Applications and Computing (IJCAC)*, vol.2, no.4, pp.32-52, 2012.
94. M. A. Shahid, N. Islam, M. M. Alam, M. M., Su'ud and S. Musa, "A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach," *IEEE Access*, no.8, pp.130500-130526, 2020.
95. S. Shao, S. Guo, X. Qiu and L. Meng, "A random switching traffic scheduling algorithm in wireless smart grid communication network," *23rd International Conference on Computer Communication and Networks (ICCCN)*, pp.1-6, 2014.
96. T. Sharma et al. "Proposed Efficient and Enhanced Algorithm in Cloud Computing," *International*

Journal of Engineering Research & Technology, vol.2(2), no.2278-0181, pp.1-6, 2013.

97. T. Sharma et al. “Proposed hybrid RSA algorithm for cloud computing,” *2nd International Conference on Inventive Systems and Control, Coimbatore*, India, pp.60-64,2018.
98. T. Sharma and R.P. Bedi, “Design and Development of Pragmatic Load Balancing Algorithm for Cloud Environment,” *Wireless Personal Communications*, pp.1-21, 2024.
99. W. Shu-Ching, “Towards a Load Balancing in a Three-level Cloud Computing Network,” *Proc. 3rd International Conference on Computer Science and Information Technology (ICCSIT)*, pp.108- 113, 2010.
100. A. N. Singh and S. Prakash, “WAMLB: weighted active monitoring load balancing in cloud computing,” *In: Big data analytics*, Springer, Singapore, pp.677–685, 2018.
101. N. Singh, Y. Hamid, S. Juneja, G. Srivastava, G., Dhiman, T. R. Gadekallu and M. A. Shah, “Load balancing and service discovery using Docker Swarm for microservice based big data applications,” *Journal of Cloud Computing*, vol.12, no.1, pp.4, 2023.
102. G. Soni et al. “A novel approach for load balancing in cloud data center,” *IEEE International Advance Computing Conference (IACC)*, 2014.
103. T. Subramanian and N. Savarimuthu, “Application based brokering algorithm for optimal resource provisioning in multiple heterogeneous clouds,” *Vietnam Journal of Computer Science*, no.3, pp.57-70, 2016.
104. H. Sulimani, R. Sulimani, F. Ramezani, M. Naderpour, H. Huo, T. Jan and M. Prasad, “HybOff: a Hybrid Offloading approach to improve load balancing in fog environments,” *Journal of Cloud Computing*, vol.13, no.1, pp.113, 2024.
105. 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,” *Knowl-Based Syst*, vol.138, pp.91–104, 2017.
106. W. Y. Tian, Zhao, Y. Zhong, M. Xu and C. Jing, “A dynamic and integrated load-balancing scheduling algorithm for Cloud datacenters,” *IEEE International Conference on Cloud Computing and Intelligence Systems*, pp.311-315,2011.
107. J. Tordsson et al. “Cloud Brokering Mechanisms for optimized placement of virtual machines across multiple providers,” *Future Generation Computation System*, vol.28, no.2, pp.358-367, 2012.
108. A. M. Tripathi and S. Singh, “PMAMA: priority-based modified active monitoring load balancing algorithm in cloud computing,” *J Adv Res Dynam Cont Syst*, pp.809–823, 2018.
109. C. W. Tsai and J. J. Rodrigues, “Metaheuristic scheduling for cloud: A survey,” *IEEE Systems Journal*, vol.8, no.1, pp.279-291, 2014.
110. B. Urgaonkar, P. Shenoy, A. Chandra and P. Goyal, “Dynamic provisioning of multi-tier internet applications,” *In Second International Conference on Autonomic Computing (ICAC'05)*. IEEE. pp.217-228, 2005.
111. M. Vanitha and P. Marikkannu, “Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines,” *Comp Elec Eng*, no.57, pp.199–208, 2017.
112. Vishwanath, Kashi and Nachiappan Nagappan, “Characterizing cloud computing hardware reliability,” *In Proceedings of the 1st ACM symposium on Cloud computing*,ACM. pp.193-204, 2010.
113. B. Wickremasingha et al. “Clouad Analyst: A cloud-sim based visual modeler for analyzing cloud computing environments and applications,” *Proceedings – International Conference on Advanced Information Networking and Applications*, Perth, 2010.
114. Z. Xiao, Z. Tong, K. Li and K. Li, “Learning non-cooperative game for load balancing under self-interested distributed environment,” *Appl Soft Comput*, no.52, pp.376–386, 2017.
115. L. Xiao, Y. Cao, Y. Gai, J. Liu, P. Zhong and M. M. Moghimi, “Review on the application of cloud computing in the sports industry,” *Journal of Cloud Computing*, vol.12. no.1, pp.152, 2023.

116. M. Xu et al. "A Survey on Load Balancing Algorithms for VM Placement in Cloud Computing," *Wiley Inter Science: Concurrency and Computation: Practice and* Experience, pp.1-22, 2016.
117. L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang, and Y. Pan, "Stochastic load balancing for virtual resource management in datacenters," *IEEE Transactions on Cloud Computing*, vol.8, no.2, pp.459-472, 2016.
118. F. Yunlong and L. Jie, "Incentive approaches for cloud computing: challenges and solutions," *Journal of Engineering and Applied Science*, vol.71, no.1, pp.51,2024.
119. A. H. Zamri, N. S. M. Pakhrudin, S. Saaidin and M. Kassim, "Equally Spread Current Execution Load Modelling with Optimize Response Time Brokerage Policy for Cloud Computing," *International Journal of Advanced Computer Science and Applications*, vol.14, no.2, 2023.
120. Q. Zhang, L. Cheng and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, no.1, pp.7-18, 2010.
121. Q. Zhang, L. Cherkasova and E. Smirni, "A regression-based analytic model for dynamic resource provisioning of multi-tier applications," *In Fourth International Conference on Autonomic Computing (ICAC'07),* pp.27-27, 2007.
122. Y. Zhang and J. Wang, "Enhanced Whale Optimization Algorithm for task scheduling in cloud computing environments," *Journal of Engineering and Applied Science*, vol.71, no.1, pp.121, 2024.
123. Z. Zhang, C. Xu, S. Xu, L. Huang and J. Zhang, "Towards optimized scheduling and allocation of heterogeneous resource via graph-enhanced EPSO algorithm," *Journal of Cloud Computing*, vol.13, no.1, pp.108, 2024.
124. D. Zhao, M. Mohamed and H. Ludwig, "Locality-aware scheduling for containers in cloud computing," *IEEE Transactions on cloud computing*, vol.8, no.2, pp.635-646, 2018.
125. 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,* vol.12, no.1, pp.85, 2023.