International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

An In-Depth Analysis of VM Allocation and Load Balancing Parallel Strategies in Cloud Computing: A Systematic Review

Saurabh Kumar[1], Amandeep Kaur[2]

Submitted: 02/03/2024 **Revised**: 15/04/2024 **Accepted**: 23/04/2024

Abstract: The adoption of innovative technology, particularly cloud computing, has attracted users and information technology companies due to its numerous advantages, such as cost savings, remote work capabilities, automatic syncing, data backups, and easy accessibility. However, the increasing demands of end users and the need to expand data centers have led to challenges for Cloud Service Providers (CSPs), including high energy consumption and operational costs, resulting in a rise in carbon dioxide emissions. To address these environmental concerns, the concept of Green Cloud Computing has emerged, aiming to create an eco-friendly computing environment. In addition, symmetrical strategies have also been employed to reduce energy consumption, operational costs, and CO2 emissions in Cloud Data Centers (CDCs) through resource allocation, virtualization, and Virtual Machine (VM) migration. Virtualization technology in cloud computing offers cost-effective deployment of virtual resources. The use of symmetrical strategies ensures energy efficiency and load balancing among Physical Machines (PMs), allowing customers to access and configure cloud resources based on a pay-per-use model. However, the presence of heterogeneous servers and dynamic resource usage within VMs in CDCs can lead to resource imbalances, resulting in performance degradation and violation of Service Level Agreements (SLAs). To achieve effective scheduling and address these issues, load balancing algorithms have been developed to support elastic scheduling, which is a complex problem to solve. This paper provides insights into energy consumption in CDCs, the relationship between VMs and PMs, CDC terminology, centralized and distributed management, and existing research in the field. Additionally, it presents load balancing algorithms aimed at mitigating resource imbalances in CDCs. In summary, the objective of this work is to provide the nuts and bolt understanding and knowledge of the potential algorithms of VM allocation and load balancing through comparatively analyzing of different approaches in the context of Green Cloud Computing and load balancing in CDCs.

Keywords: *Cloud Data Centre; Cloud Computing; Energy Consumption; Virtualization*

1. Introduction

Cloud Computing (CC) has been extensively embraced across various domains due to its popularity and cost-effectiveness. Virtual Machine (VM) technology facilitates efficient resource allocation, ensuring security through techniques like blockchain and cryptography. CC models offer on-demand services, allowing users to dynamically adjust their leased resources, characterizing it as "elastic computing." Cloud Service Providers (CSPs) offer basic computing and storage resources collectively in Cloud Data Centers (CDCs), sparing organizations the high costs of purchasing and maintaining physical infrastructure [1]. However, the rapid growth of CDCs has led to increased energy consumption and carbon dioxide (CO2) emissions. Studies indicate a significant rise in global CDC energy consumption, contributing to a substantial portion of worldwide electricity consumption. Green Cloud Computing aims to mitigate this environmental impact by using renewable energy sources and promoting sustainable energy use with minimal carbon emissions. Despite its potential, there's room for improvement in Green cloud computing, necessitating further research and development. This research aims to address these gaps by examining existing work in the field. It emphasizes the importance of reducing CO2 emissions and energy consumption in CDCs through innovative technologies and practices [2]. Fig. 1 illustrates the concept of CO2 emissions and energy consumption in a Google search, highlighting the significance of addressing these environmental concerns in the context of cloud computing. One approach, Virtual Machine Consolidation (VMC), enhances resource utilization and energy efficiency by reallocating VMs across Physical Machines (PMs) [3,4].

VMC involves sharing hardware among VMs monitored by a Virtual Machine Monitor (VMM) [5]. This method remaps residual workloads to less active PMs, putting idle PMs into sleep mode to conserve energy, reducing overall energy consumption in Data Center Consolidation [6]. Migration shifts VMs from underutilized to highly utilized PMs, reducing the number of required PMs. Symmetrical to cloud scheduling architectures, this consolidation aligns with elasticity principles, saving energy by transitioning inactive PMs to sleep mode when no VMs are allocated. Fewer active servers lead to decreased energy consumption, as reflected in Equation (1) computing resource utilization by VMs to PMs ratio.

[1*] *Research Scholar Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India*
[2] *Professor Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India*
** Corresponding Author Email: weth.smarttech@gmail.com*

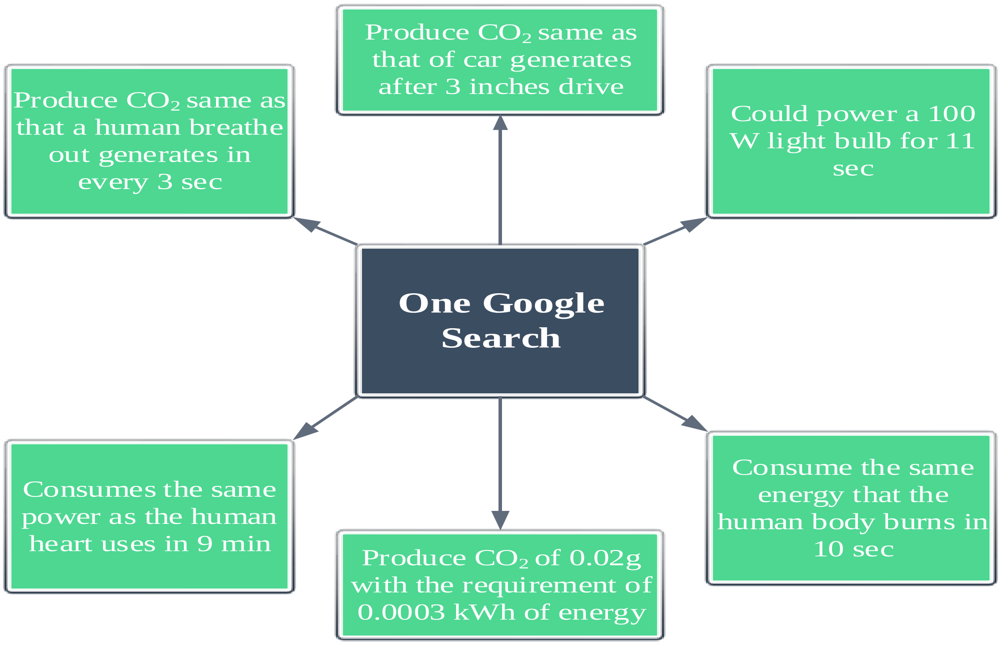

One Google Search
Produce CO_2 same as that a human breathe out generates in every 3 sec
Produce CO_2 same as that of car generates after 3 inches drive
Could power a 100 W light bulb for 11 sec
Consumes the same power as the human heart uses in 9 min
Produce CO_2 of 0.02g with the requirement of 0.0003 kWh of energy
Consume the same energy that the human body burns in 10 sec

Fig. 1. Energy Consumption of Machines in a Single Search of Google [3]

In case of an increase in resource utilization ratio, the overall resource utilization ratio of CDC has been increasing, which is given by equation (2).

$$RU_{ri} = \frac{Amount\ of\ resource\ utilized\ by\ r_i}{Total\ amount\ of\ resource\ of\ r_i} \quad (1)$$

$$RU_{CDC} = \frac{1}{N}\sum_{i=1}^{N} RU_{ri} \quad (2)$$

RU is defined as the resource utilization rate RU_{CDC} is the resource utilization rate by the CDC centers.

1.1. Relation between VM and PM in CDC

In CDC, the distribution of workload from the heavily loaded machines to the ideal one is done by the mechanism of load balancing. This mechanism aimed to balance the load so that the resources can be utilized appropriately with better user satisfaction. By assigning VMs to the right hosts, CDC aims to accomplish load balancing and effective VM scheduling. This allocation ensures a balanced utilization of all resources across all hosts, thereby optimizing the overall system performance. But in order to make the greatest use of the resources at hand and minimize resource waste, the right Load Balancing (LB) models and algorithms are employed. These benefits achieve failover, and scalability, and avoid bottlenecks and over-configuration with the reduction of response time. The relationship of applications, VMs, and hosts in a CDC [7]. The bottom-tier host is a representation of the physical resources, including CPU, RAM, and disc use. A server virtualization platform like XEN virtualizes and manages these resources, allowing hosts to run multiple VMs. Each VM executes applications with predefined dependencies. Multiple VMs are allocated to each host, and each VM may have multiple installed applications. The VM manager unifies the load balancing algorithm. This study examined various LB algorithms [8].

1.2. Organization of the Paper

The paper is structured in the following manner: Background research along with terminology used in CDC is presented. In section 3, LB Algorithm is explained in cloud. Section 4 represents the scheduling metric evaluated while balancing workload in the cloud. This section contains the symmetrical analysis of the parameters that contribute to the evaluation of cloud scheduling and allocation architectures. In section 5, the Performance evaluation platform for VM load balancing like a realistic platform, and simulation toolkit is presented. In section 6 conclusions drawn after studying multiple papers is presented followed by contributions and scope of the paper.

2. **Related Work**

Before discussing state-of-art, we discussed basic terms used for VM load balancing algorithms. The terminology use in CDC is described as follows:

- Virtualization Technology: This technology ameliorates the capabilities of existing infrastructure and resources and provides CDC with the opportunity to host multiple VMs on a common infrastructure. In recent years, technologies such as VMware and Xen have been widely used to integrate the hardware infrastructure of enterprise data centers.
- Virtual Machine Migration (VMM): Specifically,

the term VMs live migration has been used from a user's perspective, the VM seems to remain responsive throughout the migration process. In contrast to traditional migration, live migration delivers several advantages including energy consumption minimization and load balancing. Voorsluys et al. have evaluated the various applications operated in Xen VM to determine the outcome of VM live migration and showed that the produced overhead due to migration is minimized up to the desired level [1].

- Virtual Machine Consolidation: According to VM resource requirements, VM consolidation can also be employed in cloud computing. By moving the current VM from underutilized resources to another one, energy usage is decreased.

2.1. Centralized and Distributed Management

Mostly, centralized and distributed schedulers are the two most commonly used load schedulers in which load balancing is implemented. These services allow cloud users to access the services or the resources such as storage, processing, and many more offered by the CSPs.

Centralized: In this approach, the entire available resources are delivered to the cloud user by a single IT business unit [2]. This approach provides better data quality, due to the centralized features provided by the centralized schedular over the entire IT infrastructure. Ni et al. demonstrated a VM mapping policy to balance the load. It utilized the available resources that are being consumed by the active VM. For load-balancing, a self-adaptive weight scheme has been used. Also, to resolve the load balancing problem, the probability method was used [3]. Tordsson et al. (2012) modeled a new method in which the optimization of VM placement has been performed as per the user’s specifications by managing the infrastructure [4]. Wang et al. (2013) have used static and dynamic load balancing approaches. An encoded rule expansion algorithm has been used for rule transformation with very little effect on the classification results [5]. Song et al. (2015) have presented a federal-based migration approach for HLA to manage the load on CDC. VM has been utilized as a container for the federal. The effectiveness of the HLA system has been boosted by adopting a migration plan, according to measurements of computation and communication costs [6]. The scenario of supporting LB algorithm in CDC. Gao et al. (2016) have presented a solution for the newly designed resource management approach using a centralized controller. This approach failed to balance higher data traffic in CDC. To overcome this, an NP-complete method has been modelled to address the load balancing problem. An f-approximation point, the biggest number in terms of potential in the planned network, has been created using this method. Using this scheme, multiple controllers can work together and hence increase the efficiency with a balanced load in CDC [7]. In order to facilitate centralized data management in the cloud and inside an integrated system notably within the educational system, Machado et al. (2017) conducted a survey of several solutions [8]. Cui et al. (2018) have devised a method to address the issue of SDNs numbers through reaction time regulation. Here, the variation in the features of cloud nodes concerning the response time has been observed for different controllers [9]. Tarahomi and Mohammad (2019) have focused on balancing the load by analyzing the dependencies of nodes on the present behavior of the system. An integrated approach has been presented by which the VMs are placed in a well-managed way, hence contributing to saving energy [10]. Kumar et al. have developed a solution to balance the load using a dragonfly-based approach. For the purpose of selecting the activities that have to be reassigned in VM to balance the load, the approach has leveraged a few PM and VM parameters [11]. RM et al. balance the load using the optimization technique such as Firefly in which the behavior of the flies has been related to the proposed technique to obtain the outcome [12].

All the concerned authors of the fertility illustrated a total of four issues to be resolved under best utilization policies.

- VM Placement over a PM
- Selection of the hotspot PM
- Hotspot PM is used to select the VMs.
- The select VMs further placed to the target PM allocated from the hotspot PM.

The VM placement architecture follows the minimum power consumption policy before the allocation of any VM [13]. To do so, the VM allocation policy MBFD which is inspired by BFD is applied universally. The MBFD algorithm is utilized to assess the availability of PMs in a given list and determine if they have adequate resources to accommodate a VM. The algorithm checks the status of each PM and ensures that sufficient resources are available before making the allocation decision. If the resource is available to the physical machine and overloaded, then that machine may lead to higher energy consumption. In terms of mathematical illustrations, the MBFD algorithm can be re-written as follows.

Algorithm 1: MBFD

```
Input: VML, PML // VML and PML are the VM and PM list
$\mathrm{VMLs{=}Sort(VML.VM_{cpu_{utilization}},'d')}$ // Sort the VMs as per their CPU utilization in descending order.//
```

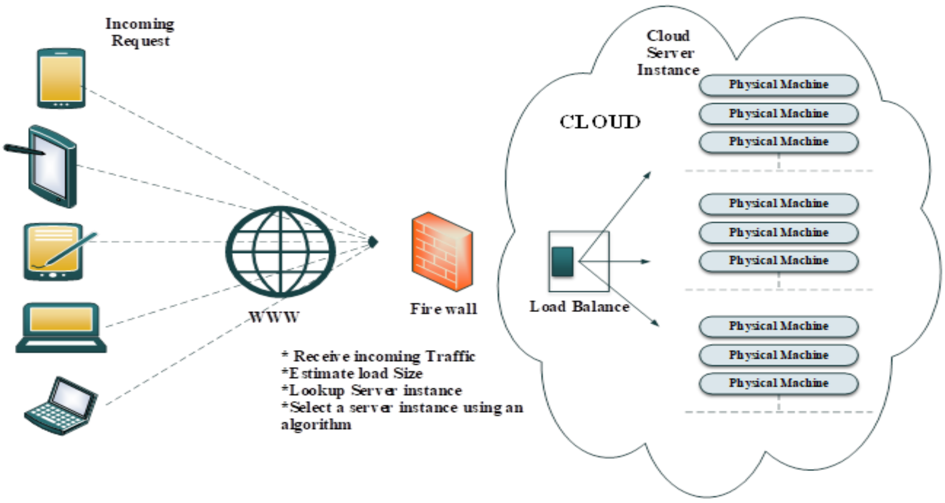
```
For i = 1: k  // where k represents the overall number of VMs in VMLs.
    For j = 1: p // where p is the number of PMs in the PML as a whole.
        $\min_{power} = \max_{power} \forall VMLs$  // Let any power be the minimum power consumption
        $\min_{index} PML = \max_{index} PML$  // Let the last index be the PM
        $If_1( PML_j.resources > VMLs_i.resoruce_{demand})$
            $PC_{current} = Compute.PC(VMLs_i, PML_j)$//Compute power consumption of the allocation//
            $If_2\ PC_{current} > \min_{power} \forall VMLs$
                $\min_{power} \forall VMLs = PC_{current}$  // Replace the minimum power consumption //
                $\min_{index} PML = j$  // Replace the index of allocation
            $End\ If_2$
        $End\ If_1$
    $End\ For_2$
Allocate i to j
$End\ For_1$
```

Decentralized: As the name suggests, a decentralized schedular shares the infrastructure between multiple nodes, all of which are served equally as per the user's requirements. Permanent individuals can host their cloud servers if cloud users share computing resources. The users can rent their resources and generate additional revenue. The advantage is that there is no point in failures and the ability to expand is achieved by finding more people to join. The working scenario for a decentralized cloud system. It is seen that different VMs has been allocated to the host machine or Physical Machine through the migration process in which VM has been migrated based on the resource utilization. Shah et al. have explored the utilization of decentralized schemes for load distribution in a grid environment. The researchers have concluded that their work performed by them provides better results with an adaptive solution [14]. Choudhary et al. (2012) have presented multiple workflow techniques in a decentralized cloud computing environment. The designed model described the information in contrast to the processors that consist of slot information, exchange of information, and workload-related knowledge. To access these jobs task model is used that processes these jobs in the queue [15]. Sangwan et al. (2016) have contributed to enhancing the problem of scheduling a task in a distributed environment. The tasks have been allocated to an appropriate user as per their demand and hence utilized the jobs appropriately. This all has been performed using a load balancing scheme. Here, the work has been performed in five different steps namely; workflow submission, pre-processing, task scheduling, task completion notification, and finally achieving the best solution [16]. Centelles et al. (2020) have presented a decentralized mechanism for each device attached to the cloud network. Monitoring workload of the system can be monitored using the CPU utilization and RAM usage and hence minimizing the energy consumption in the CC [17]. Wang et al. (2021) introduced a security-oriented retrieval technology aimed at preserving data with minimal resource consumption and wastage [18].

3. Allocation Strategies for Elastic Scheduling

This section highlights the information about the elastic scheduling used in a cloud computing system. Let us first understand what elasticity is attained using and how one can achieve it in a cloud environment. The load balancer is a device whose duty is to distribute workload among backend servers. The real example of this is website servers, which obtained millions of requests from outside users to access multimedia data and the need of all users is that data should be of better quality and available to the users in minimum time.

Fig. 2 represents a typical LB structure for a cloud environment. In this structure, the role of the load balancer is as follows:



* Receive incoming Traffic
*Estimate load Size
*Lookup Server instance
*Select a server instance using an algorithm

Fig. 2. LB Structure in Cloud [19]

- Collect information from incoming requests received from end-users.
- Determine the request size and establish a request queue according to workload
- Using the server monitor tool, the current load status of the cloud server is checked periodically.
- Load balancing schemes are used for the selection of appropriate servers.

Every second, routing of high data traffic is performed by a complex computing network. The load balancer distributes this high amount of data across the servers that are accessible so that solutions can be offered to the end user without affecting the SLA. Further, load balancing was handled using the dedicated servers. Instead providing availability to the user's load balancer has the following disadvantages:

- Helps to control and track traffic.
- Improve resource utilization
- Balance network load according to node function
- Improve resource availability
- Reduce oversubscription of infrastructure

The role of the load balancer is to create a queue as per the user’s requests and then allocate appropriate VMs to the user. The information related to the allocated VMs must be recorded, and hence the idle VM is known in advance.

A dynamic load balancer is developed by Mohrana et al., (2013) which is known as an “Active load balancer” [19]. Using this scheme, those VMs having little load, the new incoming request can be assigned to that one. The authors have developed an improved throttling load balancing algorithm. This technique has been used to allocate requests to available VMs based on response and processing times. Using this approach, the status of every VM has been recorded. CDC will request permission from the throttling VM load balancer to determine how and when the VM can be allotted to the user based on the user's request [20].

A small number of the algorithms used in the cloud environment also made use of optimization techniques to distribute the load throughout the cloud system. The survey can be found in the section that follows.

3.1. Elasticity using an Optimization Approach

Several academics have employed a small number of optimization techniques to optimize load in a cloud context. The ABC algorithm was used by Nakrani and Craig (2004) as an optimization method for load optimization. In scenarios where the end-user's requirements for web services frequently change, this approach proves advantageous as it dynamically adjusts the load distribution of the web server. In this context, the server is conceptualized as a virtual server that maintains a queue of service requests. By evaluating performance parameters such as CPU time, the server calculates a "profit" indicator to determine the service it provides to the application. An "advertising board" indicates whether an idle server requires any services. The overall profit is determined by assessing various metrics. In this analogy, bees represent servers, and free servers act akin to waiting bees seeking nectar [21]. Nishant et al. (2012) employed the ACO approach as a LB algorithm. This method uses pheromone trails to simulate ant movement. They take the method of representing virtual machines (VMs) as nodes and software modules as ants. In the RLBN, a single node is initially designated as the head node. The artificial ants travel at random and make an effort to get to the head node. Throughout this process, a table is maintained to record information regarding the load on each node in the network. The purpose of the ants is to search for the least loaded node and allocate incoming requests to evenly distribute the load across the network [22]. Kumar and Mandeep (2015) have tried to balance load using a Genetic Algorithm (GA) as a nature-inspired algorithm. Using GA in the cloud for load balancing helps to migrate VMs to PMs using the fitness function of GA. After deployment of load on the node, the load has been evaluated and a better solution is determined that can deploy VM with minimum migration. Also, to enhance the performance of GA, ACO has been integrated and authors have claimed that better results have been obtained with the hybridization approach [23]. Esa et al. (2016) have used Firefly as an optimization algorithm to increase the speed of job execution in the cloud. Initially, n number of jobs is created and the resources like job length and speed of resources have been determined. Also, the fitness function is used to generate population, and based on that best jobs have been scheduled with minimum job execution time [24]. The "Taguchi method" is a load balancing strategy that has been enhanced, according to Ragmani et al. (2018). Response speed and lowest cost have been considered while evaluating the factors that have a stronger impact on cloud systems. The proposed ACO algorithm performs 11% and 38% better than the round-robin algorithm when it comes to response time and processing time analysis, respectively [25]. A highly efficient VM allocation strategy with the least amount of power and bandwidth usage was put out by Xing et al. in 2022. The resource consumption by developing a new solution using the ACO ensures obtaining the best solution [26].

Ebadifard and Babmir (2018) introduced a static task scheduling algorithm that incorporates the use of the PSO algorithm to enhance its performance. In this algorithm, the tasks and virtual machines (VMs) are metaphorically linked to food sources and bees, respectively. Assigning tasks to VMs is akin to bees exploring and collecting food from various sources. When a VM becomes overloaded, it is comparable to a bee encountering an empty food source. Thus, to ensure balance, tasks are transferred from overloaded VMs to those with lighter loads [27]. It's easy to compare the process of deleting work from an overloaded virtual machine (VM) and locating a suitable, lightly laden VM to bees foraging for food by dropping tasks. The standard deviation of the system load is calculated in order to assess if the load balancing in the cloud environment is insufficient. With a shown boost of 22% in performance and a 33% decrease in makespan, the proposed approach using PSO surpassed the current round-robin (RR) approach. PSO, is a reliable optimization method that relies on the movement and positioning of particles. The velocity of particles is adjusted according to their positions, with particles closer to the optimal solution or food source considered as optimal, while those farther away are deemed suboptimal. The algorithmic process for the PSO approach is written below.

Algorithm 3: PSO based Load balancing

```
1   Start
2   Input: Set of VM={$VM_1$,$VM_2$,……$VM_j$}, Jobs=
    J={$J_1$, $J_1$,…………..$J_i$}
3   Output: To find the best position jobs on the VM
4   Initialize particle dimension, particle position, and
    velocity.
5   For each particle balance particle position using the
    load balancing algorithm
6   For all particles compute fitness value using
7   $Fv = \frac{makespan}{Average\ Utilization}$
8   If (fitness value (Fv  <= pbest)
9   Set obtained value as updated pbest value
10  Select particle position as gbest
11  Chose the best particle as gbest
12  Calculate velocity and update particle positions
13  Repeat steps from 6 to 14 if the criteria for maximum
    iteration are not satisfied
14  Return: best position of job on VM
15  End – Function
```

It is important to emphasize that when selecting these articles, we considered the editorial and publisher. In other words, the selection process of the literature is concentrated on the articles of related publishers. The number of articles published in IEEE explored including (conferences, journals, Magazines, early access articles, courses, and books for load balancing in cloud and virtualizations is shown in Fig. 3, and Fig. 4 respectively.

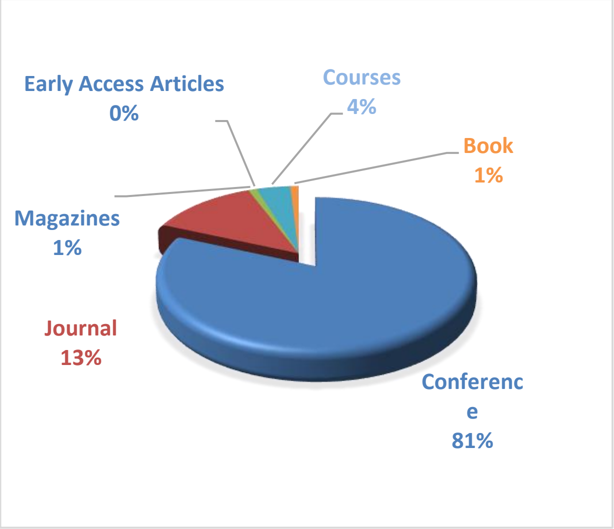


Fig. 3. Load Balancing in Cloud Computing IEEE Papers

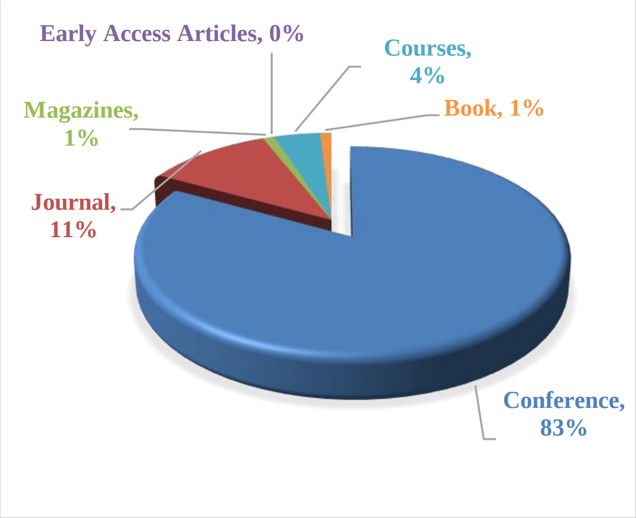


Fig. 4. Virtualization in Cloud Computing IEEE Papers

4. Comparison of Load Balancing Scheduling Metrics

The effectiveness of the LB algorithm for VM load balancing in the cloud system may be evaluated using several factors. To increase performance metrics, the parameters can be optimized by combining many approaches to obtain the best values. The parameters are symmetrical and depending on the object under evaluation. Here, we will be discussed different parameters. These performance metrics have been utilized to determine the effectiveness of the machines and their usage.

- Load variance: This metric is used to measure the deviation of the measured value from the average utilization. This metric has been evaluated by several researchers including Ni et. al., (2011) because it is easy to compute [3]. However, in some of the articles, the researchers are focused to evaluate time constraints instead of measuring load variance.

- Makespan: It is employed to gauge how long hosts need to process information before a project can be completed. It is one of the crucial factors considered while assessing the effectiveness of the scheduling algorithm. The small value of makespan represents the smaller load on the server, and minimizing makespan is the main motive of any of the scheduling algorithms.
- The quantity of hosts is overloaded: This parameter is used to know the status of overloaded hosts in the cloud system. The threshold value has been used to identify the overloaded hosts. The load balancing algorithm's primary objective is to lower the likelihood of overloaded hosts in the system. This parameter provides a clear picture of how the workload is distributed among the hosts.
- Location of all VMs as a percentage (%): This parameter is utilized to know about the percentage distribution of workload in CDC. The percentage has been calculated by knowing the minimum and maximum value of the current VMs that are allocated to PM for task execution in CDC. But, this parameter analyses load by considering only the allocated VMs without considering the resources used. Therefore, if the system is heterogeneous, the concept of VM workload balancing remains open to be discussed.
- Throughput: It measures the speed at which the host can process requests because an unbalanced load affects the performance of the system. Therefore, higher throughput should be accompanied by a better system under dynamic load. When service response time is the main concern, this indication is especially pertinent. This statistic is often considered in combination with other indications rather than by itself in load balancing algorithms. For instance, it is necessary to take throughput measurements along with migration count measurements.
- The number of migrations: This indicator serves as a supplementary metric that reflects overall performance and is evaluated in conjunction with other parameters. Even if more migrations could be needed to create a balanced load in the cloud system, this could negatively impact the overall performance of the system. As such, this metric is not evaluated in isolation, but rather is utilised to evaluate the impact of load balancing.
- SLA violations: This parameter is also used to represent the performance of the cloud server. SLA violation can be defined as the VM's inability to obtain sufficient resources from the host (for example, CPU). Too many violations of SLA indicate a poor balance between hosts. Therefore, this parameter should be minimized.

The parameters evaluated by several researchers are presented in Table 1.

Table 1. Data summary of granite rock burst experimental data

Parameters	Technique used
Utilization standard deviation	Self-adaptive weighted and Lightest memory first
Host load	Central Load Balancing Policy for Virtual Machines
Makespan	Using the GA and the Load Balanced Min-Min (ILBMM) method. Load balancing approach. a conventional-based approach that utilized the concept of task migration instead of VM migration.
Overloaded hosts	VM based federate migration scheme
Location of all VMs as a percentage (%)	Scheduling Interoperability
Throughput	Central Load Balancing Policy for Virtual Machines
Number of migrations	ACO approach
SLA Violation PDR, Latency, and Packets dropped Learning Rate	Ant Colony System Superframe Interleaving Cognitive Intelligence

5. Performance Evaluation Platform for VM Load Balancing

This section described real-world platforms and simulation toolkits used for virtual machine load balancing performance determination.

5.1. Realistic Platforms

Experimenting in a real environment is more convincing, and a few real platforms used for performance testing of cloud system is described below.

- OpenNebula: The infrastructure of OpenNebula is managed using a control server, the so-called frontend, which can run on Linux or OS X. The exchange of data between the control machine and the nodes of the cloud cluster takes place via the SSH protocol. OpenNebula uses MySQL or SQLite database to store parameters. Implemented disk image management, hot plugging, template repository, management of the entire VM life cycle (creation, cloning, and so on) and accounts (user, group, roles). The disk imaging subsystem supports multiple SAN and NAS storage. Access to images from any cluster node is organized using SSH, NFS, SFTP, HTTP, GlusterFS, Luster, and iSCSI / LVM protocols. Virtual networks are created in Virtual Network Manager, which provides the desired level of abstraction and isolation. By leveraging this testbed, it becomes feasible to implement a novel load balancing algorithm that incorporates the integration of platforms like KVM and OpenNebula [26].

- ElasticHosts: It is a global IaaS CSP, which contains different geographical distributions, and they provide instant and flexible computing power for easy-to-use cloud servers [27].
- EC2: Amazon EC2 is a pay-per-use platform that provides services to its users from the EC2 cloud. This platform facilitates users with services like storage, and processing along with web services. Amazon EC2 is a virtual computing environment that allows users to access a range of applications with different operating systems through web service interfaces [28].

5.2. Simulation Toolkits

For impulsive network systems and the laboratory resources such as hosts, it becomes essential to design simulation tools for the simulation of large-scale data. Researchers need a visual demonstration and execution of an application in the clouds for better study on massive amounts of data. The description of cloud data like the status of workload on the particular application, end-user information, location of users, number of users, available resources, etc. are included by the cloud center. The simulation process included load balancing algorithms for easy implementation and to check the workload status on a cloud server. The multi-layered architecture of clouds is shown in Fig 10. The main layer provides management of applications, VM hosts, and dynamic system conditions. After the expansion of the core VM function, the CSP also learns the effectiveness of various layers in the architecture. The top “user layer”, represents the general entries of users and by expanding the structures on this layer, the developer allows the application to create requests in different approaches and configurations [29] in the research.

- CloudSim: Several elements of cloud-based systems may be modelled and assessed by academics and developers using the popular cloud computing environment simulator CloudSim. It offers a scalable and adaptable platform for assessing resource provisioning, scheduling policies, energy-efficient algorithms, and overall system performance. By utilizing CloudSim, users can replicate intricate cloud infrastructures, simulate the behavior of cloud applications, and analyze the effectiveness and efficiency of diverse cloud solutions. Its modular design and wide range of features make it an invaluable resource for comprehending and enhancing cloud computing systems [30].
- CloudSched: Using the CloudSched platform, users can assess and contrast various scheduling algorithms in cloud infrastructure (IaaS). It gives users access to information on host and server loads, enabling them to evaluate the efficiency and efficacy of different scheduling techniques. The simulation kit helps the designers to find a required solution while implementing different scheduling algorithms to the cloud structure.
- FlexCloud: It is a dynamic simulator by which, end users simulate the data initialization process in CDC, like allocation of VM requests, and evaluation of performance for different scheduling algorithms. The structure of FlexCloud consists of four levels named resource levels, Scheduler level, Broker level, and client level. The top level that is entitled “client level” is located at the top of the surface. The requested task is allocated to the workload schedular for managing the workload. After that, the scheduled data is passed to the bottom layer “resource level”. Once all steps are completed, the scheduled sequence is forwarded to the lower level for subsequent processing [31]. The data is processed, and upon completion, an acknowledgment is transmitted to the end-user through the client level.

Python: As per the requirement of modern-day developments, python has emerged as a novel development platform for complex problems [32]. The platform provides a lot of open-source dependencies to cope with the algorithm architectures like PSO, ABC, Neural Networks, etc.

6. **Conclusion**

The paper examines the importance of achieving balanced load distribution in cloud systems to ensure peak efficiency and symmetry. It highlights the significance of appropriate scheduling and resource allocation for evaluating cloud computing system performance. Over the past decade, research efforts have aimed to enhance cloud system performance, particularly in the areas of load balancing and virtual machine allocation. The study explores various factors influencing the performance of virtualization systems and CDC under dynamic workload conditions. It also discusses load balancing methods developed for CC systems, focusing on algorithms that parallelly balance load by allocating virtual machines to suitable hosts. The paper provides a comprehensive presentation and discussion of these algorithms, aiming to offer a deep understanding of current techniques and potential future developments in the field. Through trials on simulation tools, the study reveals that meta-heuristics techniques yield superior performance metrics compared to traditional approaches. It suggests employing metaheuristics approaches for validating real-time platforms and promoting research in real cloud environments. The paper notes that VM load balancing algorithms are multi-objective functions, aiming to minimize costs while enhancing various metrics, posing a challenge for diverse optimization schemes and ensuring their reliability in future research. Modern VM load balancing algorithms are dynamic, making static VM load distribution in the cloud unsuitable. Thus, the paper emphasizes the need for investigating more adaptable VM load balancing techniques for future research. With heterogeneous VMs being used in current environments and resource availability not limiting, the ability to apply future VM load balancing methods to diverse VM sources is desirable. Overall, the paper provides insights into the evolving landscape of cloud system performance improvement, underscoring the importance of dynamic load balancing and adaptable VM allocation techniques in addressing the challenges of modern cloud computing environments.

7. **Contributions and Scope**

The paper reviews many articles across various sources, investigating Virtualization and LB algorithms' strengths, weaknesses, and challenges. It focuses on concerns like placing virtual machines on physical machines (PM), selecting hotspot PMs, and optimizing VM placement. The proposed methodology models VMLB algorithm in a distributed environment, emphasizing elastic resource allocation and PSO-based LB technique for performance optimization. It compares LB algorithms' effects on metrics like load variance, makespan, and SLA violations using simulation toolkits like CloudSim. The paper underscores the importance of effective load balancing for VM allocation and presents a compilation of relevant papers within the IEEE domain. Future challenges involve exploring and validating optimization schemes for dynamic load balancing, as static distribution may not suit the evolving cloud environment, necessitating adaptive algorithms.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. Shao, Y. Yang, S. Juneja, T. GSeetharam, "IoT data visualization for business intelligence in corporate finance," Information Processing & Management, vol. 59, p. 102736, 2022.

[2] J. Ni, Y. Huang, Z. Luan, J. Zhang, D. Qian, "Virtual machine mapping policy based on load balancing in private cloud environment," in International Conference on Cloud and Service Computing, 2011, pp. 292–295.

[3] J. Tordsson, R.S. Montero, R. Moreno-Vozmediano, I.M. Llorente, "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers," Future Generation Computer Systems, vol. 28, pp. 358–367, 2012.

[4] P.C. Wang, "Scalable packet classification for datacenter networks," IEEE Journal on Selected Areas in Communications, vol. 32, pp. 124–137, 2014.

[5] X. Song, Y. Ma, D. Teng, "A load balancing scheme using federate migration based on virtual machines for cloud simulations," Mathematical Problems in Engineering, vol. 2015, pp. 1–11, 2015.

[6] X. Gao, L. Kong, W. Li, W. Liang, Y. Chen, G. Chen, "Traffic load balancing schemes for devolved controllers in mega data centers," IEEE Transactions on Parallel and Distributed Systems, vol. 28, pp. 572–585, 2016.

[7] L.M. de Almeida Machado, F.J.L. Rita, C.H. da Silva Santos, "Mobile and cloud based systems proposal for a centralized management of educational institutions," Independent Journal of Management & Production, vol. 8, pp. 271–286, 2017.

[8] J. Cui, Q. Lu, H. Zhong, M. Tian, L. Liu, "A Load-Balancing Mechanism for Distributed SDN Control Plane Using Response Time," IEEE Transactions on Network and Service Management, vol. 15, pp. 1197–1206, 2018.

[9] M. Tarahomi, M. Izadi, "A hybrid algorithm to reduce energy consumption management in cloud data centers," International Journal of Electrical and Computer Engineering, vol. 9, p. 554, 2019.

[10] M. Kumar, S.C. Sharma, "Dynamic load balancing algorithm to minimize the makespan time and utilize the resources effectively in cloud environment," International Journal of Computers and Applications, vol. 42, pp. 108–117, 2020.

[11] S.P. RM, S. Bhattacharya, P.K.R. Maddikunta, S.R.K. Somayaji, K. Lakshmanna, R. Kaluri, A. Hussien, T.R.

Gadekallu, "Load balancing of energy cloud using wind driven and firefly algorithms in internet of everything," Journal of Parallel and Distributed Computing, vol. 142, pp. 16–26, 2020.

[12] R. Shah, B. Veeravalli, M. Misra, "On the design of adaptive and decentralized load balancing algorithms with load estimation for computational grid environments," IEEE Transactions on Parallel and Distributed Systems, vol. 18, pp. 1675–1686, 2007.

[13] V. Choudhary, S. Kacker, T. Choudhury, V. Vashisht, "An Approach to Improve Task Scheduling in a Decentralized Cloud Computing Environment," International Journal of Computer Technology & Applications, vol. 3, pp. 312–316, 2012.

[14] A. Sangwan, G. Kumar, S. Gupta, et al., "To convalesce task scheduling in a decentralized cloud computing environment," Review of Computer Engineering Research, vol. 3, pp. 25–34, 2016.

[15] R.P. Centelles, M. Selimi, F. Freitag, L. Navarro, "Redemon: Resilient decentralized monitoring system for edge infrastructures," in: 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), pp. 91–100, 2020.

[16] T. Wang, Q. Yang, X. Shen, T.R. Gadekallu, W. Wang, K. Dev, "A privacy-enhanced retrieval technology for the cloud-assisted internet of things," IEEE Transactions on Industrial Informatics, vol. 18, pp. 4981–4989, 2021.

[17] S.S. Moharana, R.D. Ramesh, D. Powar, "Analysis of Load Balancers in Cloud Computing," International Journal of Computer Science and Engineering, vol. 2, pp. 101–108, 2013.

[18] S. Nakrani, C. Tovey, "On honey bees and dynamic server allocation in internet hosting centers," Adaptive Behavior, vol. 12, pp. 223–240, 2004.

[19] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K.P. Singh, R. Rastogi, et al., "Load balancing of nodes in cloud using ant colony optimization," in: 2012 UKSim 14th International Conference on Computer Modelling and Simulation, pp. 3–8, 2012.

[20] P. Kumar, E.M. Kaur, "Load balancing in cloud using ACO and genetic algorithm," International Journal of Scientific Research Engineering & Technology (IJSRET), vol. 4, pp. 724-730, 2015.

[21] D.I. Esa, A. Yousif, "Scheduling jobs on cloud computing using firefly algorithm," vol. 9, pp. 149-158, 2016.

[22] A. Ragmani, A. El Omri, N. Abghour, K. Moussaid, M. Rida, "A performed load balancing algorithm for public Cloud computing using ant colony optimization," Recent Patents on Computer Science, vol. 11, pp. 179–195, 2018.

[23] H. Xing, J. Zhu, R. Qu, P. Dai, S. Luo, M.A. Iqbal, "An ACO for energy-efficient and traffic-aware virtual machine placement in cloud computing," Swarm and Evolutionary Computation, vol. 68, p. 101012, 2022.

[24] G.N. Nguyen, N.H. Le Viet, M. Elhoseny, K. Shankar, B.B. Gupta, A.A. Abd El-Latif, "Secure blockchain enabled Cyber–physical systems in healthcare using deep belief network with ResNet model," Journal of parallel and distributed computing, vol. 153, pp. 150-160, 2021.

[25] OpenNebula, "OpenNebula – Open Source Cloud & Edge Computing Platform," OpenNebula, 2016.

[26] R. Dowsley, A. Michalas, M. Nagel, N. Paladi, "A survey on design and implementation of protected searchable data in the cloud," Computer Science Review, vol. 26, pp. 17-30, 2017.

[27] Elastichosts, "Elastichosts," 2022, Available: https://www.crunchbase.com/organization/elastichosts . Accessed May 4, 2023.

[28] A.A. Khan and M. Zakarya, "Energy, performance and cost efficient cloud datacentres: A survey," Computer Science Review, vol. 40, p.100390, 2021.

[29] W. Tian, M. Xu, A. Chen, G. Li, X. Wang, Y. Chen, "Open-source simulators for cloud computing: Comparative study and challenging issues," Simulation Modelling Practice and Theory, vol. 58, pp. 239–254, 2015.

[30] T. Le, "A survey of live virtual machine migration techniques," Computer Science Review, vol. 38, p.100304, 2020.

[31] M. Xu, G. Li, W. Yang, W. Tian, "Flexcloud: A flexible and extendible simulator for performance evaluation of virtual machine allocation," in: 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), pp. 649–655, 2015.

[32] Z.J.K. Abadi, N. Mansouri, M. Khalouie, "Task scheduling in fog environment—Challenges, tools & methodologies: A review," Computer Science Review, vol. 48, p.100550, 2023.