International Journal of Basic and Applied Sciences, 14 (3) (2025) 385-395

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/60apby38
Research paper

Effective Load Balancing on Servers Using Virtual Machines for Cloud Computing Environment Using Green Computing and The Skewness Algorithm

T. Swathi [1] *, S. Saravanan [2], Namra Joshi [3], Mohammad Arif [4], Mahesh Babu Kota [5], Indhumathi R [6], Elangovan B [7]

[1] *Associate Professor, Department of Computer Science and Engineering, G. Pulla Reddy Engineering College, Kurnool, India*
[2] *Associate Professor, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, Tamilnadu, India – 600062.*
[3] *Assistant Professor, Department of Electrical Engineering, SVKM's Institute of Technology, Dhule, Maharashtra, India.*
[4] *Professor, Department of Computer Science and Engineering, Alliance University, Bengaluru, India – 562106.*
[5] *Department of Electronics and Communication Engineering, Aditya University, Surampalem, Kakinada, India.*
[6] *Assistant Professor, Department of Electrical and Electronics Engineering, M.Kumarasamy College of Engineering, Thalavapalayam, Karur, Tamilnadu, India.*
[7] *Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh - 522302.*
**Corresponding author E-mail: swathitudi@gmail.com*

Received: April 15, 2025, Accepted: July 19, 2025, Published: July 27, 2025

Abstract

Efficient resource management in cloud data centers is crucial for optimizing resource utilization while maintaining performance and energy efficiency. This study presents the design and implementation of an automated dynamic resource management system that balances overload avoidance and green computing. Overload avoidance is achieved by dynamically adjusting physical machine (PM) resources to meet virtual machine (VM) demands, preventing performance degradation due to excessive resource contention. Simultaneously, energy efficiency is ensured by minimizing the number of active PMs, turning off unused ones to conserve power. To enhance resource allocation, the concept of skewness is introduced to measure the uneven utilization of server resources. By reducing skewness, overall server efficiency is improved while addressing multi-dimensional resource constraints. The proposed system is implemented and evaluated using the CLOUDSIM 2.0 simulation platform with NetBeans, demonstrating its effectiveness in optimizing resource allocation, preventing overload, and reducing energy consumption in cloud environments.

Keywords: *Cloud Computing; VM Migration; CLOUD SIM; NetBeans; Skewness Algorithm; Automated Resource Management System.*

1. Introduction

Nowadays, more and more areas of IT applications require computing power provided by a computing infrastructure, be it in the fields of big data, machine learning, high-performance computing, or computer applications and social networks. To meet these needs, companies are turning to cloud computing, a way to outsource their IT services. The cloud offers a pooling of IT resources and thus reduces human, hardware, and software costs thanks to economies of scale. A cloud provider has a data center where it deploys its many physical machines (PMs) and can offer to provide its resources to customers. In the case of the infrastructure as a Service model, the customer rents data center resources in the form of virtual machines (VMs). This model relies on virtualization as the main component making it possible to divide a PM, called a host, into several guest VMs. Virtualization is made possible by a hypervisor, which provides a software abstraction to partition a PM, isolate VMs from each other, and provide VMs with access to the hardware of the PM. Supplier of cloud can then allocate to the customer, in the form of a VM, a workspace of pre-defined size, while guaranteeing the quality of service and performance. In cloud platforms, resources are managed dynamically by software thanks to the virtualization layer that is interposed between the physical resources of the host servers and the runtime environments of the users, i.e., the VMs consuming the virtualized resources presented to them by the hypervisor. VM users overestimate their resource requirements [1], and the optimization of resource utilization rates by cloud infrastructure operators relies on over-commitment of resources [2]. Virtualization enables several mechanisms for optimizing resource exploitation, particularly the automatic migration (AM) of VMs, which is a backbone mechanism of dynamic resource management systems (RMSs).

Migrations consume infrastructure resources both at the level of the host ceding the VM and at the destination host of the VM [3] and at the level of network bandwidth to transmit the current state of the devices. Migrated VMs; these migrations are therefore costly for the infrastructure. In addition, migrated VMs - not necessarily those manipulated to cause migrations - may experience performance degradation due to the migration process [4]. Much work has focused on optimizing, evaluating, or even modeling the dynamic migration of VMs, this has never been considered from a security perspective. In the work carried out, AM as an attack vector is exploited to maliciously consume infrastructure resources and degrade the performance of hosted VMs. Beyond this attack, that an 'attack is called ' untimely migrations of VMs' demonstrated in this work, the objective is to highlight the vulnerability of dynamic RMSs with respect to consumption profiles in malicious resources. The reasoning is based on two observations. On the one hand, these systems react by nature to the resource consumption of VMs that are possibly under the control of attackers. On the other hand, multi-tenancy creates an interdependence between the VMs [5], which, in the case of dynamic resource management, causes the VMs co-resident with the attacking ones to suffer performance degradation when they are migrated. There is a policy concern about enhancing mapping to ensure that the resource needs of VMs are fulfilled while minimizing the number of PMs utilized. The resource needs of VMs stem from their operation across many different applications, making it difficult as the workload fluctuates over time. The primary two disadvantages are addressed through overload prevention and eco-friendly computing. This study presents the design and functioning of an automated resource management (ARM) system that strikes a suitable balance between the two aspects mentioned above. The first issue is addressed by ensuring the PM meets the resource needs of all VMs operating on it. Otherwise, the PM will become overloaded and diminish the efficiency of its VMs. The issue is resolved by decreasing the quantity of PMs utilized until every VM satisfies the criteria. Therefore, the inactive PMs are maintained in an off state to conserve energy. Consequently, a resource allocation system is created that successfully prevents system overloads while minimizing the number of servers utilized. The concept of skewness is presented to assess the random utilization of a server. Minimizing the skewness enhances the overall usability of the servers when faced with the risk of multi-dimensional resource limitations.

2. Literature review

Migration can also be used for the purpose of minimizing the amount of unused resources on hosts in a cluster, i.e., to maximize host consolidation. In this case, migration is used to optimize the placement of VMs on the different hosts making up a cluster to maximize the resource utilization rates of each host to minimize the number of hosts constituting a cluster.

A Microsoft study finds that the cost of physical resources accounts for 45% of the total cost of data centers, and the cost of energy accounts for 15% of the total cost [6]. The energy cost is therefore not negligible, which reinforces the interest in the consolidation of the hosts.

In [7], Ferreto et al. reduce the problem of placing VMs on hosts to the problem of the multi-dimensional backpack with the additional constraint of minimizing the number of running hosts in the cluster, several heuristics to solve this problem have been proposed. The need to redistribute the VMs on the hosts is re-evaluated periodically while taking care to minimize the number of VM migrations to limit the performance degradations induced by these migrations.

Verma et al. offer pMapper [8], a VM placement system that considers the optimization of energy consumption, the performance constraints of VMs execution, or the constraints related to the integration of the cost of the migration of VMs in the migration decision process. pMapper collects the execution performance of all VMs present in a cluster as well as their energy consumption. The intelligence of pMapper lies in the aggregation of recommendations dictated by energy, performance, and migration managers to find a compromise and propose an allocation of resources as close as possible to the needs of VMs while seeking to guarantee good performance. execution of VMs and minimizing the energy consumption of clusters.

Entropy [9] is a dynamic placement system for VMs on a cluster of hosts, offered by the Ecole des Mines de Nantes. Entropy aims to minimize the number of hosts in use in a cluster and uses constraint programming in two main phases. The first, based on constraints, considers the capacity of the hosts, the capacity of the VMs (CPU and memory), calculates a placement that implies a minimum number of hosts, and draws up a reconfiguration plan for the distribution of the VMs, which makes it possible to meet these constraints. The second phase considers the cost and feasibility of the migrations proposed in the first phase and seeks to reduce the number of migrations. Entropy is built in two layers, a central Entropy server and Entropy instances running inside the VMs, which allows Entropy to scale the number of instances according to the number of hosts and VMs in the cluster.

Xu et al. [10], propose an algorithm responsible for determining a distribution of workloads on different VMs and then distributing these VMs on the hosts using the migration of VMs with three objectives, i) minimizing the quantities of unused resources on the hosts, ii) minimization of energy consumption and, iii) minimization of thermal consumption, ie temperature peaks generated by server activity. They use the GGA (Grouping Genetic Algorithm) algorithm [11], which allows for grouping problems with different constraints into several groups. The proposed algorithm deals only with the initial placement of VMs and does not consider the revision of the placement based on the resource consumption of the VMs.

Dong et al [12] reduce the problem of placing VMs on hosts in a cluster to two known problems, which are the backpack problem and the quadratic assignment problem. The authors propose an algorithm that considers not only the constraint of minimizing the number of hosts constituting a cluster to minimize power consumption, but also minimizing the network bandwidth consumed within a data center. This algorithm deals only with the initial placement of VMs and does not react to the variation in the load of the VMs to dynamically change the distribution of VMs on the hosts.

Chen et al. (2022) [23] investigated the effect of data complexity on load forecasting performance using a long short-term memory (LSTM) model. They stated that inconsistent data negatively affects the accuracy and generalizability of the model. By applying normalization techniques such as logarithmic and power transformation, the study showed significant improvements in prediction performance. This highlights the importance of addressing data sparsity in time series models and has important implications for other fields, including cloud resource forecasting, where data distribution also affects prediction accuracy.

Existing studies provide valuable strategies for energy-efficient VM deployment and load balancing in data centers, but they also have some limitations. Many studies (e.g., those by Xu et al. and Dong et al.) focus only on the initial setup of VMs and cannot dynamically adapt to workload changes, limiting their real-time effectiveness. Although methods such as Entropy and pMapper consider multi-objective optimization, they often result in high migration costs, which degrade performance. In addition, the studies on energy consumption and performance protocols by Ferreto et al. and Verma et al. do not fully address data-driven prediction problems, and therefore cannot proactively manage resource allocation. In addition, Chen et al. (2022) pointed out that data complexity and inconsistency affect prediction accuracy, but most VM control models still rely on static or rule-based solutions rather than combining time series prediction models that meet dynamic requirements. In addition, thermal impact and network bandwidth are usually considered separately rather than as a single

optimization framework. Therefore, current systems still lack a comprehensive solution that combines predictive analysis, energy awareness, dynamic migration control, and data normalization.
To overcome these challenges, a more dynamic and flexible approach to resource management in virtual cloud environments is required to enable intelligent VM migration, accurate workload prediction, and energy consumption.

3. Dynamic and sustainable resource management in virtualized cloud environments

3.1. Migration for load balancing

Load balancing between the hosts composing a cluster aims to optimize the placement of VMs on the different hosts to minimize the risk of contention on each of the hosts while maximizing the resource utilization rates, avoiding overloading certain hosts in the presence of very light hosts. This approach, which aims to minimize the dispersion of the load between the hosts, assumes knowledge at the cluster level of the distribution of the loads between the hosts to judge the migrations of VMs to be operated.
To demonstrate the attack by unwanted migrations of VMs, the commercial VMware DRS [13] algorithm is used, which primarily pursues an objective of load balancing within the cluster. DRS is selected for two main reasons. First, DRS is the state-of-the-art dynamic resource management algorithm often used for the evaluation of new algorithm proposals. In addition, the use of a commercialized and available algorithm facilitates the reproduction by the community of the results that are obtained in the experiments. According to the state of the art, there are no dynamic management algorithms for open resources, and the versions of algorithms published in articles such as those described in the previous section [14-16] are often partial.

3.2. Cloud computing

Cloud computing delivers on-demand services over the Internet, providing computing power, storage, and software access based on a pay-as-you-go model. Service Level Agreements (SLAs) define usage terms, including performance expectations like availability, execution time, and response thresholds.
Key characteristics of cloud computing include on-demand self-service, resource sharing, scalability, global accessibility, and measurable billing. It supports four deployment models—private (exclusive use), community (shared among specific users), public (open to all), and hybrid (combining private and public). Cloud services are categorized into three models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).
IaaS provides virtual or physical infrastructure, such as computing power, memory, storage, and networking, offering users full control over configurations. Leading providers include Amazon EC2, Google Compute Engine, and private cloud solutions like OpenStack and VMware vCenter.
PaaS offers a managed development environment, enabling application deployment without infrastructure management. Examples include Google App Engine and Microsoft Azure, which provide frameworks and APIs for streamlined development.
SaaS delivers fully managed applications accessible via the Internet, eliminating installation and maintenance efforts. Common examples include Google Apps and Salesforce. This work focuses on enhancements at the IaaS level.

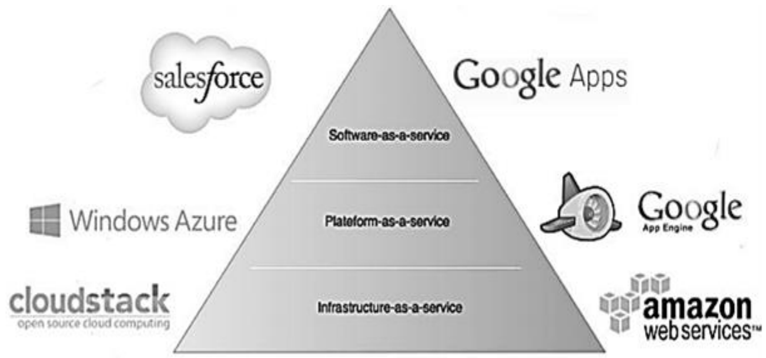

Fig. 1: Service Models of Cloud Computing.

Cloud computing infrastructure is implemented through various models based on architecture, data hosting, and user requirements. The Public Cloud provides shared resources managed by third-party providers, ensuring scalability and cost efficiency. The Private Cloud offers dedicated infrastructure for a single organization, enhancing security and control. The Community Cloud serves specific groups with common needs, facilitating collaboration. The Hybrid Cloud integrates public and private cloud features, optimizing flexibility and resource management.

3.3. Virtualization, cloud computing technology

Virtualization is the technology that can be described as a key technology on which cloud computing is based. All applications deployed in the Cloud are executed within platform's VMs, commonly known as VMs. Virtualization allows the separation of the hardware and software layers. Within a VM, there is no knowledge of the underlying architecture, the hardware architecture on which it runs. The hypervisor acts as a gateway between the physical hardware layer and VMs. Traditional IT infrastructure runs multiple applications and dedicates a physical server to each task. This approach is expensive and may not be the best use of resources since the task does not consume the full capacity of the resources 100% of the time. Indeed, The Green Grid shows in its study that servers are underused. The average CPU utilization rate is 5% with a maximum of 10%. In this study, 85% of servers have a CPU utilization rate of 6% or less. This same study shows that, depending on the capacities and uses of the physical servers, a consolidation of 10:1 is possible, without CPU overhead. That is, one physical server can support the load of 10 other servers. This consolidation is possible using virtualization. Another, more recent study shows that a third of the servers are unused. This percentage potentially represents 10 million servers worldwide, carrying virtualization or not. Linked to this underutilization of physical servers, energy cost can also become a huge issue in large data centers.

Servers are unused but still consume power, increasing energy waste. The various studies, therefore, highlight the imperfect operation of data centers. Virtualization makes it possible to pool several physical systems on a single machine, in the form of VMs. It then reduces the number of physical servers in the data centers and helps reduce the cost of wasting resources. In the next section, virtualization, the core technology of Cloud Computing, is focused [17].

3.4. Green computing

Data centers are undoubtedly the fastest-growing category in ICT (Information and Communication Technologies) in terms of energy consumption and emissions. According to an IBM study, 90% of the world's data has been produced in the last 2 years. This whole volume, therefore, logically needs its physical base, and so data centers have become the critical nerve center of the ICT world, and their number is constantly increasing. They form a key part of today's Internet infrastructure. It is important to note that they not only form the infrastructure that is used to transmit data, but also the hubs where most of this data is stored. Massive development and exponential growth are also associated with a sharp increase in increasingly powerful devices. This development is mainly due to the impact of the expansion of the Internet, a change in approach to its use, as well as the development of various Internet services and applications that have accelerated this development. This changes the whole functioning of the ICT industry and the distribution of ICT. Functions became solutions and services, and decentralized became centralized. The huge increase in the number of data centers is directly proportional to energy consumption. Given the frequency and volume they represent, you should look for energy savings on this side as well. The energy consumption of the data center can be divided into the consumption of ICT and the consumption of supporting technologies. Among ICT, the consumption of servers, data devices, communication infrastructure, workstations, and accessories such as monitors, switches, etc, is counted. The supporting technologies of the data center are components of the distribution, including electricity backup, batteries, generators, switchboard transformers, as well as other components necessary for operation, such as lighting, security systems, monitoring, etc. A significant part of the cost consists of cooling components such as cooling sources, cooling units, air conditioning (CRAC, from English: Computer Room Air Condition), ventilation, and expansion units. (9) As data centers consume a significant portion of IT budgets, there is a constant trend of cost reductions. Rising energy prices, continuous development, as well as pressures to reduce environmental pressures are also pushing CIOs to various conceptual changes [18].

3.5. Cloud-oriented green computing (COGC)

COGC deals with a processing infrastructure that integrates the flexible nature, service quality, and energy reduction. The energy crisis is triggering green computing, and green computing needs mechanisms to reconfigure energy efficiency. It is essential to use the computer resources in an efficient, effective, and economical manner and to reduce power consumption. Virtualization, product recycling, telecommunication, and energy management are the different approaches to green IT. Virtualization is an energy-efficient approach that allows significant improvements in the energy efficiency of cloud providers by improving the economies associated with multiple companies sharing the same corporate infrastructure [19]. By integrating unused servers in the form of multiple VMs that share the same physical server (PS) in high usage, companies can achieve greater savings in terms of capacity and energy. COGC is a blooming technology that provides a sustainable environment. Cloud computing can provide better energy savings by the utilization of larger partitioned virtualized data centers, but the increased Internet traffic and big data will reduce the energy savings further, and it becomes a critical problem along with the carbon emissions from ICT, which affect the global climate. For the above critical problems, green cloud computing provides solutions by saving energy and reducing operating costs. Because of Adaptable provisions, sub-lease, server employment, and data center efficiency [20-22], companies can reduce carbon emissions by progressing their applications to the cloud. But the main challenge to the cloud is efficient resource utilization and reduction of power consumption, and it was met by the design of the COGC system for the data center. Four major entities are involved in green-cloud computing infrastructure: consumers, green resource allocators, VMs, and PMs (see Figure 2). Figure 3 shows the COGC system.

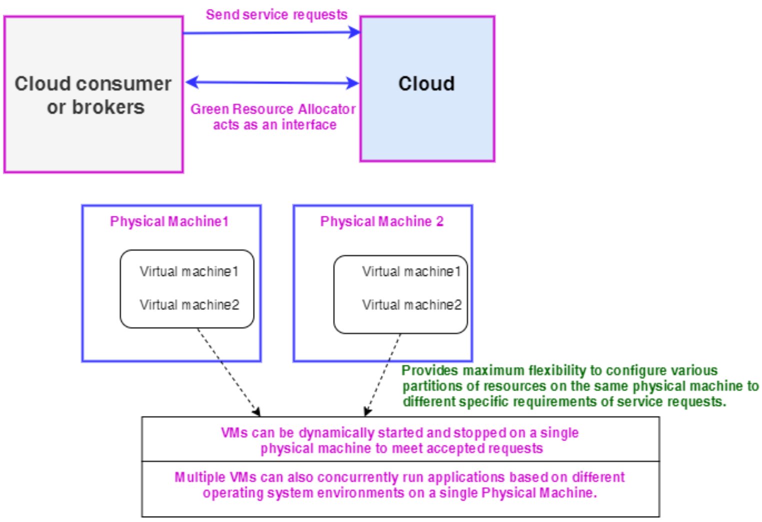

Fig. 2: Entities Involved in Green-Cloud Computing Infrastructure.

This paper aims to develop Cloud-Oriented Green Computing (COGC) architecture for dynamic resource allocation. It includes a review of existing cloud computing architectures and the proposed COGC framework.

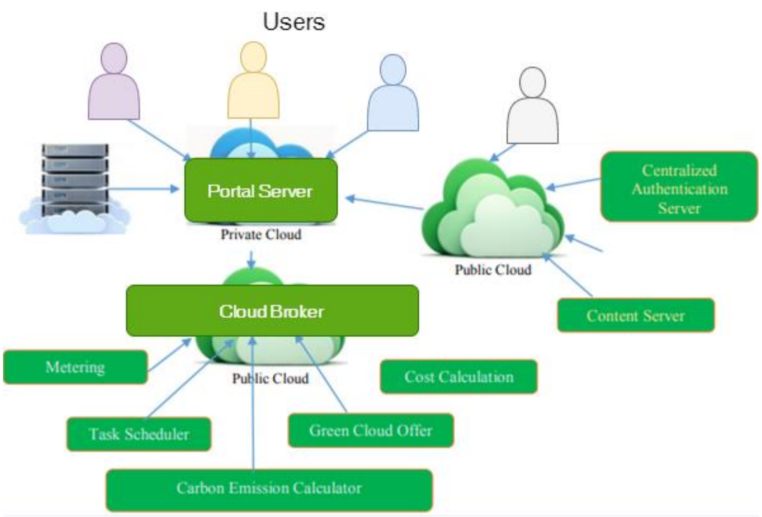

Fig. 3: COGC System.

4. System overview

Figure 4 illustrates the architecture of an automated Dynamic Resource Allocation (DRA) system in a cloud computing environment. It comprises three Physical Machines (PMs), each hosting three Virtual Machines (VMs). These VMs are linked to a VM scheduler, which connects to the internet for dynamic resource allocation to clients. The scheduler periodically evaluates resource demands, server capacity, load distribution, and VM placement across PMs to optimize resource utilization. The proposed system is detailed below.

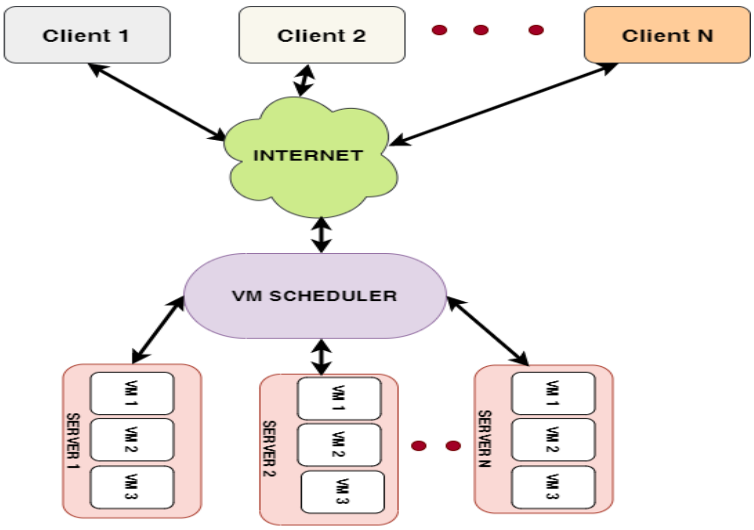

Fig. 4: Architecture of the Proposed Method.

4.1. Skewness algorithm

The proposed method evaluates resource utilization inconsistencies on a server. Let n be the number of resources, and r_j represent the utilization factor of the j^{th} resource. The resource skewness of the t^{th} server is given by equation 1:

$$S(t) = \sqrt{\sum_{j=1}^{n}(\frac{r_j}{\bar{r}} - 1)^2} \quad (1)$$

Where $\bar{r}$ denotes the mean utilization across all nnn resources on the server, given by equation 2:

$$\bar{r} = \frac{1}{n}\sum_{j=1}^{n} r_j \quad (2)$$

Since not all resources impact performance equally, only critical ones are considered. Minimizing skewness optimizes overall resource usage by balancing workloads effectively [23].
For a cloud environment with n PMs and m VMs, resource consideration remains constant, allowing efficient estimation. The skewness mechanism consists of three components: load forecasting, hotspot mitigation, and the Cloud-Oriented Green Computing (COGC) approach. Load forecasting periodically assesses allocated resources based on predicted VM demands.
COGC includes modules such as the VM scheduler, predictor, hotspot solver, coldspot solver, and migration list. The VM scheduler periodically gathers VM demand history, PM capacity, and load data before forwarding requests to the predictor. The predictor analyzes historical data to estimate present and future resource needs. A PM's load is determined by aggregating the resource utilization of its VMs. The hotspot solver detects overloaded PMs by comparing their utilization against a threshold. If a PM exceeds the threshold, VM migration is triggered. The temperature of a hotspot is given by equation 3:

$$T(t) = \sum_{r \in S}(r - ht)^2 \quad (3)$$

Where S represents loaded resources, and ht is the hot threshold. The skewness mechanism further identifies HotSpot, WarmSpot, and ColdSpot states. When a WarmSpot's average PM usage is below the green computing threshold, some PMs are shut down to conserve energy. A ColdSpot occurs when all server resources fall below both warm and cold thresholds, making it a suitable candidate for VM migration. Once all VMs are migrated, the server can be shut down to optimize energy efficiency.
The system aims to eliminate or reduce hotspots. VM migration decisions prioritize PM temperature. During low loads, COGC reduces active PMs while maintaining performance. Shutdowns occur when overall resource utilization drops below the green computing threshold. PMs are sorted by ascending memory usage to determine which to deactivate. The total memory of all VMs in a ColdSpot is considered before shutdown, ensuring efficient resource allocation and energy savings.

4.2. Predicting future resource needs

The focus remains on web applications, with one approach involving VM-level application statistics, such as analyzing pending request logs. However, this method is not always feasible. Instead, predictions rely on the historical external behavior of VMs. The system employs an exponentially weighted moving average (EMA) using the TCP/IP model, represented by equation 4,

$$EMA(J) = \propto * E(J) + (1 - \propto) * O(J), 0 \leq a \leq 1 \quad (4)$$

where $E(J)$ and $O(J)$ denote the estimated and actual load at time t for the J^{th} iteration, respectively. The parameter α balances stability and response accuracy. This equation is used to predict the load on a DNS server.

4.3. Cache server implementation

As a modification to this work, a cache is created for the work requested by the user, which will be saved for a limited time. If another user requests the same server for the Cloud Service Provider (CSP) server, the server will first check the cache memory, thus reducing the working process time. If the user request arrives, the server will immediately provide the data to the user if the requested data is in cache memory; otherwise, the server executes the task of the user request by transferring it to RAM. The data flow diagram is shown in Figure 5, and the use case diagram is shown in Figure 6. And the Sequence diagram of Figure 7 shows the flow of logic within the system. CSP provides an interface with the users.

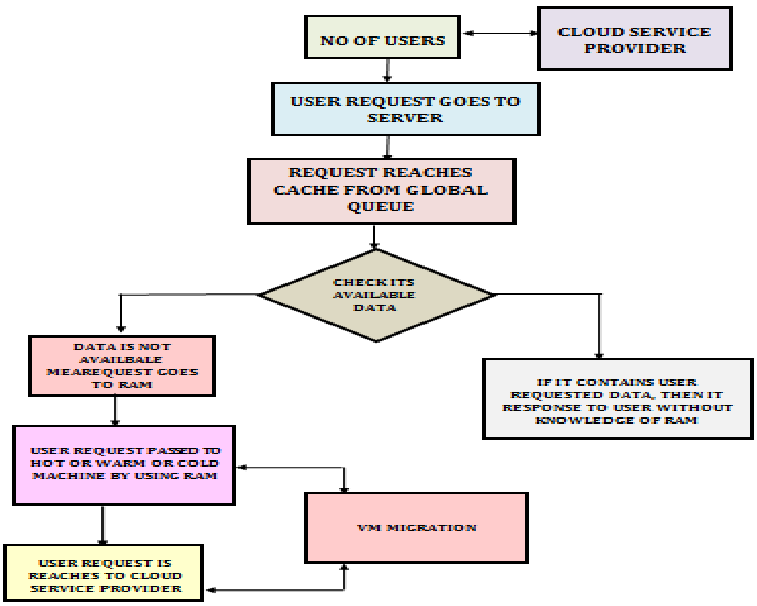

Fig. 5: Data Flow Diagram of VM Migration Implementing Cache Memory.

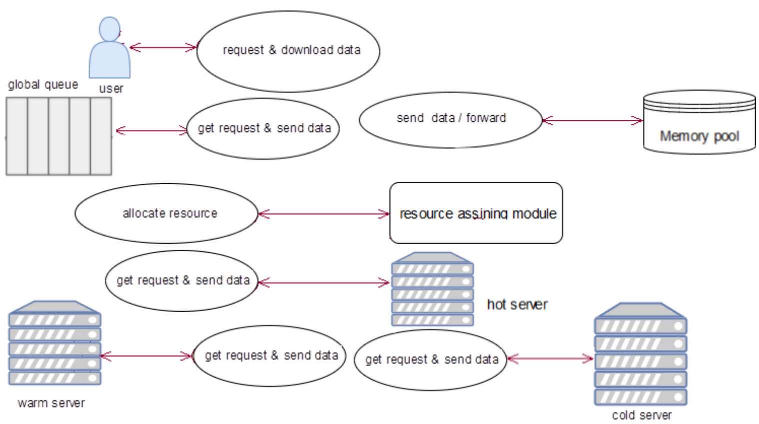

Fig. 6: Use Case Diagram of the Proposed System.

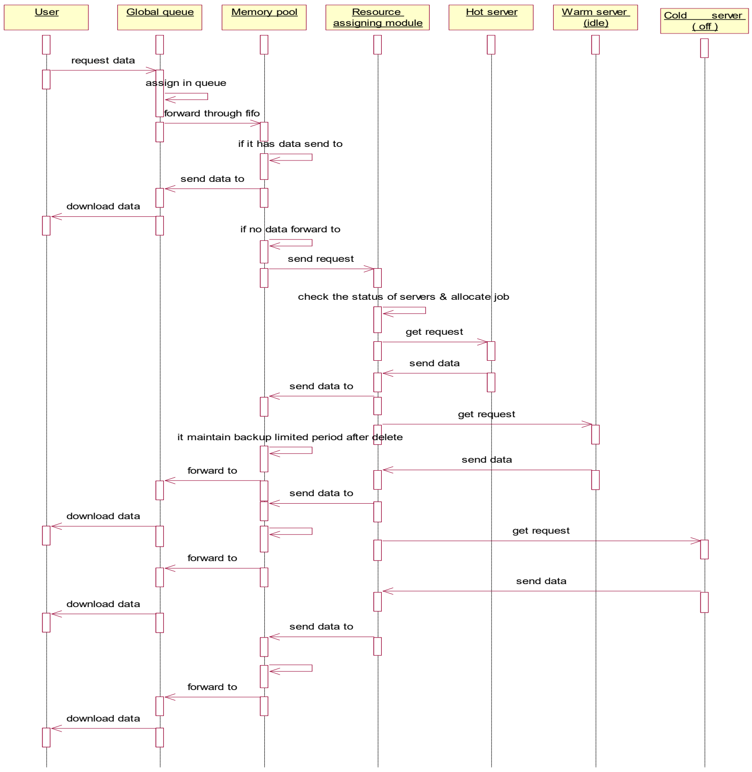

Fig. 7: Sequence Diagram of the Proposed System.

The system specifications which show the hardware and software configuration are shown below in Figure 8.

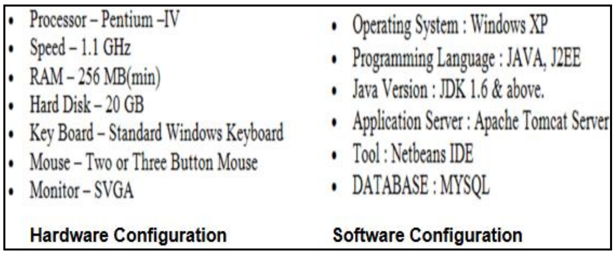

Fig. 8: System Specifications.

5. Experimental results

Initially, the Java data server is opened in the cmd window (Figure 8). Then the Member is successfully registered and logged in as shown in Figure 10. The three VMs, namely VM1, VM2, and VM3, are created for every PM. Here, three PMs, namely PM1, PM2, and PM3, are created in the experimental study. Then the source window opens, and the available file format shows all the files in different formats. In this experiment, the PDF files are selected, and the available PDF files appear in the files list. The required files are selected from the files list for downloading. At that time, VM1 in PM1 is processing 1000000.pdf, VM2 in PM1 is processing 1100000.pdf, and VM3 in PM1 is processing 11111111.pdf. VM1 in PM2 is processing 1200000.pdf, VM2 in PM2 is processing 22222222.pdf, and VM3 in PM2 is processing 333333333.pdf. VM1 in PM3 is processing 444444.pdf, VM2 in PM3 is processing 44444444444.pdf, and VM3 in PM3 is processing 5555555.pdf. Once 1000000.pdf is downloaded, 5555555.pdf is processed by VM1 in PM1, and VM3 in PM3 is turned off. Once 1100000.pdf is downloaded, 77777777.pdf is processed by VM2 in PM1. Then 444444.pdf is downloaded, and now VM1 in PM3 is also turned off. Then 333333333.pdf is downloaded, and now VM3 in PM2 is also turned off. Then, after 5555555.pdf is downloaded, 666666.pdf from the files list is processed by VM1 in PM1. After downloading 1200000.pdf, VM1 in PM2 is turned off. After downloading 11111111.pdf, VM3 in PM1 is turned off. After downloading 44444444444.pdf, VM2 in PM3 is turned off. After downloading 22222222.pdf, VM2 in PM2 is turned off. After downloading 777777777.pdf, VM2 in PM1 is turned off. After downloading 666666.pdf, VM1 in PM1 is turned off. Thus, the Job is first allocated to PM1, next PM2, and PM3. Hence, VM1 is a hot server, VM2 is a warm server, and PM3 is a cold server, respectively. Thus, all PDF files are downloaded and stored in the temp folder. And the files are uploaded to the cloud, and they can also be available in a cloud platform. Files processed by VMs and their migration with file names are shown in Figure 9.

File No.	File Name	Size of the file	PM1			PM2			PM3			File Downloaded
			VM1 H	VM2 W	VM3 C	VM1 H	VM2 W	VM3 C	VM1 H	VM2 W	VM3 C	
1	1000000.pdf	1.01 MB	1	2	3	4	5	6	7	8	9	1 (1000000.pdf)
2	1100000.pdf	1.00 MB	9	2	3	4	5	6	7	8	-	2 (1100000.pdf)
3	11111111.pdf	1.08 MB	9	10	3	4	5	6	7	8	-	7 (444444.pdf)
4	1200000.pdf	1.15 MB	9	10	3	4	5	6	-	8	-	6 (333333333.pdf)
5	22222222.pdf	1.07 MB	9	10	3	4	5	-	-	8	-	9 (5555555.pdf)
6	333333333.pdf	1.29 MB	11	10	3	4	5	-	-	8	-	4 (1200000.pdf)
7	444444.pdf	1.45 MB	11	10	3	-	5	-	-	8	-	3 (11111111.pdf)
8	44444444444.pdf	1.34 MB	11	10	-	-	5	-	-	8	-	8 (44444444444.pdf)
9	5555555.pdf	1.83 MB	11	10	-	-	5	-	-	-	-	5 (22222222.pdf)
10	777777777.pdf	1.47 MB	11	10	-	-	-	-	-	-	-	10 (777777777.pdf)
11	666666.pdf	1.20 MB	11	-	-	-	-	-	-	-	-	11 (666666.pdf)
			-	-	-	-	-	-	-	-	-	

Fig. 9: Files Processed by VMS and Its Migration with File Name.

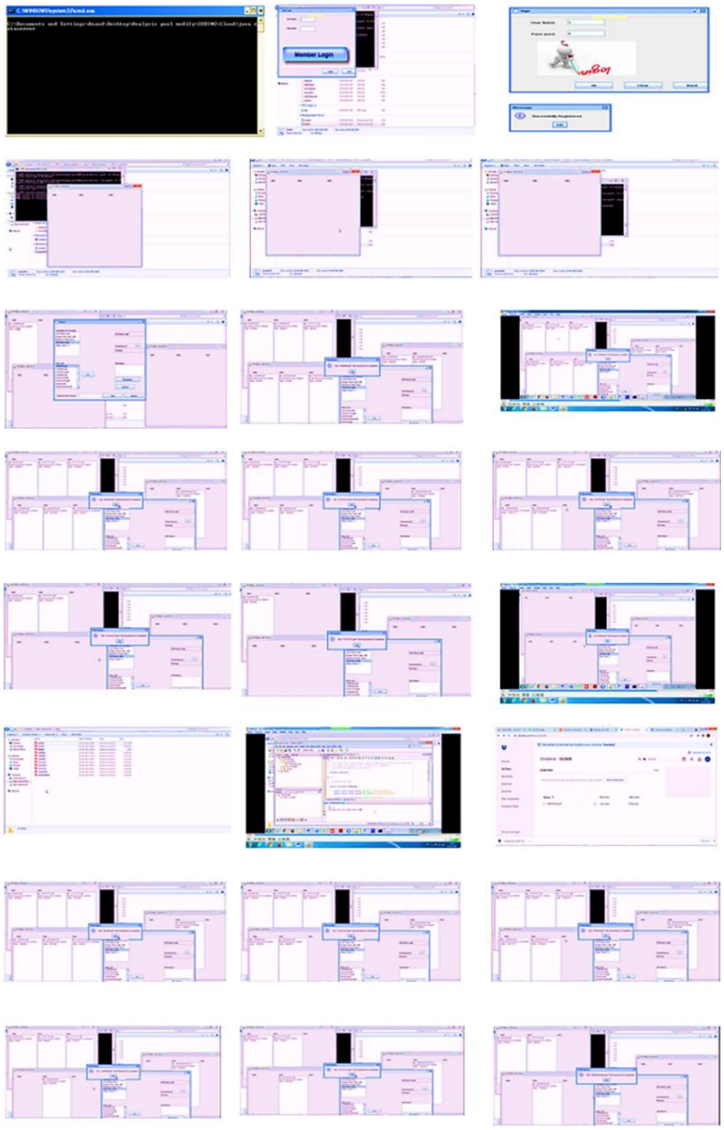

Fig. 10: VM Migration Patterns.

5.1. Test results and analysis

An important analysis of sophisticated research accepted in the literature to solve the migration problem for achieving energy efficiency at the data center level is presented. However, the energy efficiency can be achieved by the employment program considering the availability of multi-dimensional resources, ensuring unrestricted resource allocation, and reducing the number of VM migrations required for improvement. In this test, it is seen how the algorithm handles a combination of CPU, network-intensive workloads, and memory. The resource allocation level of the three servers, 1, 2, and 3, has 500 KB of total memory, and the resource utility memory for servers 1, 2, and 3 is 120 KB, 170 KB, and 80 KB, respectively (Table 1).

Table 1: Resource Allocation Status

Server Name	Total Memory	Resource Used
Server1	500 kB	120kB
Server2	500 kB	170kB
Server3	500 kB	80kB

In a cloud computing environment, resource demands fluctuate dynamically. Virtualization helps data centers optimize resource utilization while minimizing power consumption. By enabling multiple operating systems and applications to run on fewer servers, virtualization reduces both power usage and the physical footprint of data centers. This approach enhances performance by balancing workloads, optimizing resource allocation, and improving server utilization—potentially increasing usage rates from an initial 10-15% to nearly 80%.
The system is tested in a cloud setup using Java, with a user-friendly interface that allows customers to submit job requests based on specific configurations. Users can define the number of servers, virtual machines (VMs), MIPS, and RAM. The test environment includes both homogeneous and heterogeneous data center configurations, evaluating parameters such as resource utilization, power consumption, and load balancing. VM configurations use MIPS values of 200, 300, 400, and 500, with RAM options of 256MB and 512MB. Workloads vary across different applications, ranging from 100 to 500 tasks. Figure 9 illustrates the performance comparison between the proposed algorithm and the Hypergraph Total Variation (HTV) algorithm, demonstrating that the proposed approach improves response time and achieves approximately 13.6% performance enhancement.

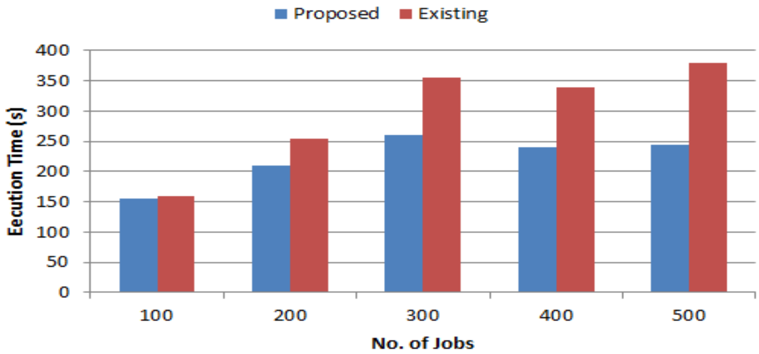

Fig. 9: Comparison of Performances of the Proposed and HTV Algorithm.

Further testing is conducted using various data center configurations with different server setups. Figure 10 presents the average resource utilization across these configurations. The results indicate that the proposed algorithm achieves higher resource efficiency compared to existing methods.

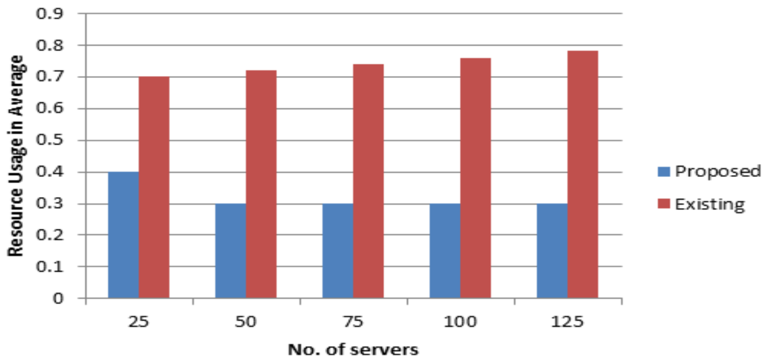

Fig. 10: Average Resource Usages for Multiple Data Center Configurations for Different Servers.

The proposed system improves resource utilization by approximately 22.8% compared to the existing system. Figure 11 illustrates the energy consumption comparison, where the proposed method demonstrates a 10.27% reduction in energy usage across various workloads and data center configurations.

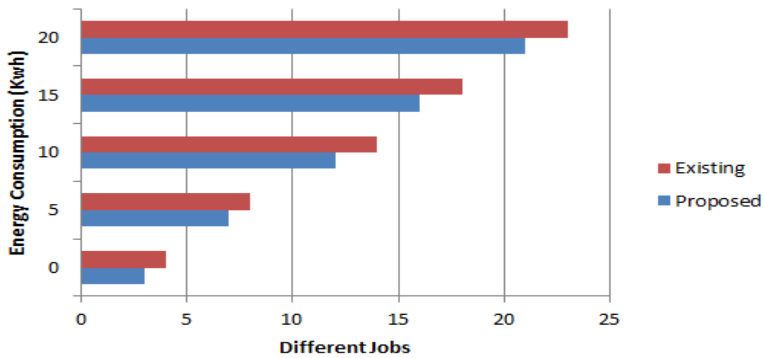

Fig. 11: Comparison of Resource Utilization of the Proposed and HTV Algorithm.

Load balancing capacity for homogeneous servers is measured based on their usage. Resource utilization (U) is comparable in terms of very less ($U \leq 0.2$), less ($U > 0.2$), medium ($U \leq 0.3$), and high values $U \leq 0.6$. Load balance is classified as good if loads on servers with minimal differences from average loads (less than 1%), while medium loads may not be the same on servers, but within allowable tolerances of 1 to 10% of average loads, and low if all servers have a random load. Table 2 shows the load balance and utilization of both systems.

Table 2: Comparison of Load Balance and Utilization of the Proposed and HTV Algorithm

Datacenter Configurations	Proposed	Existing	Proposed	Existing
Job 1	Medium	High	Medium	Medium
Job 2	Medium	High	Good	Good
Job 3	Medium	High	Medium	Good
Job 4	Medium	High	Medium	Good
Job 5	Medium	High	Medium	Good

This system achieves higher utilization (Table 3). The ability to balance loads under different jobs is a medium to existing method, however, which is good in the proposed approach, thereby it is consistent in different workloads and configurations. The servers are set up with 1000MIPS and 2000MIPS and RAM of 1024MB and 2048MB respectively in Legacy Servers. The number of VMs and their configurations and workload are set to be homogeneous for similar server systems. Figure 12 illustrates that the response time of this system will be significantly better than the existing HTV and thus a reduction of about 16.89% in response time is achieved.

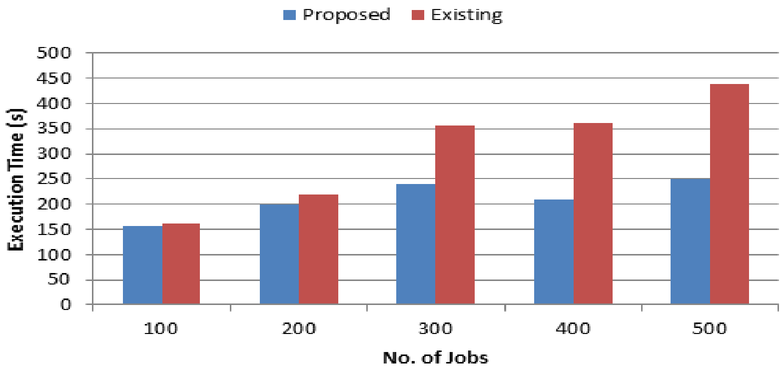

Fig. 12: Comparison of the Response Time of the Proposed and HTV Algorithm.

The resource usage for different data center configurations is shown in Figure 13, which clearly shows significant differences in usage of approximately 27.92% more than the existing one, thereby revealing its performance.

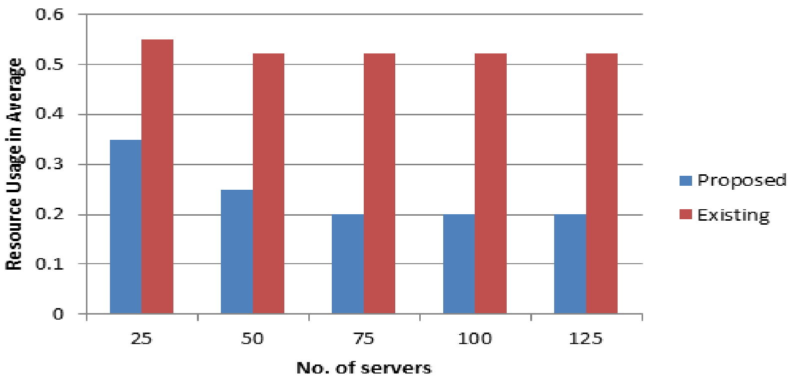

Fig. 13: Average Resource Usages for Multiple Data Center Configurations for Different Servers.

The energy consumption of heterogeneous servers in the data center is shown in Figure 14. This demonstrates the dominance of the proposed mechanism in relation to the existing one. It can achieve a reduction of about 6.91% in energy consumption.

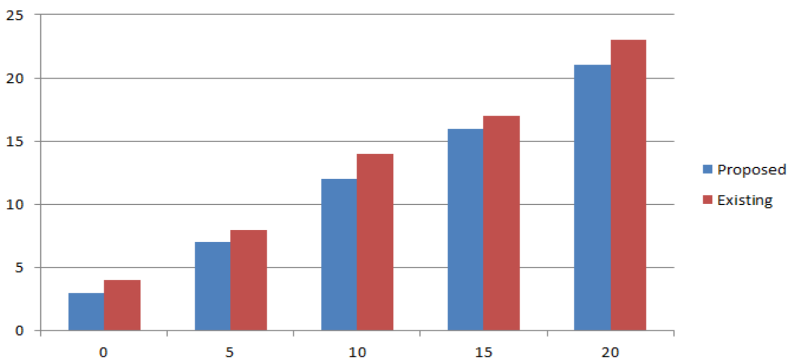

Fig. 14: Energy Consumption for Different Jobs in Heterogeneous Servers.

Table 4 shows the comparison of the proposed policy with the existing policy in terms of resource utilization and load balancing capabilities. It can be seen (Table 3) that the proposed scheme demonstrates the dominance of the existing mechanism in both resource utilization and load balancing capacity.

Table 3: Resource Utilization and Load Balance Comparison for Data Centers with Heterogeneous Server Configurations

Datacenter Configurations	Proposed	Existing	Proposed	Existing
Job 1	Medium	Medium	Medium	Medium
Job 2	Medium	Medium	Good	Good
Job 3	Low	Medium	Medium	Good
Job 4	Low	Medium	Medium	Good
Job 5	Low	Medium	Medium	Good

The proposed dynamic resource allocation (DRA) system is very practical in public, private, and hybrid cloud models due to its intelligent VM scheduling, resource allocation, and computation mechanisms. In public clouds, the system helps improve scalability and cost-effectiveness by dynamically allocating resources and reducing unnecessary processing through encryption. In private clouds, the system saves energy and ensures optimal performance by identifying unused physical machines (PMs) and decoupling them. In hybrid cloud environments, the system supports workload migration between different types of clouds based on real-time demand and resource availability. In addition, reducing the carbon footprint of the system helps improve resource efficiency in data centers, reduces unnecessary VM migrations, and significantly reduces energy consumption - saving 10.27% in one case and 6.91% in another. In summary, the system supports sustainable cloud computing by providing intelligent workload balancing, minimizing idle equipment usage, and managing environmentally friendly resources.

6. Conclusions

The proposed Cloud-Oriented Green Computing (COGC) system effectively enhances resource allocation and energy efficiency in cloud computing environments. By dynamically managing virtual machines and optimizing server utilization, the system achieves a significant reduction in hotspots while maintaining workload balance. Experimental results demonstrate that the proposed system improves resource utilization by 22.8% and reduces energy consumption by 10.27% compared to existing methods. Additionally, the use of predictive models and virtualization ensures efficient load distribution, minimizing power wastage. Overall, the COGC approach provides a scalable and energy-efficient solution for cloud data centers, contributing to sustainable computing practices.

References

[1] Poonam R. Maskare, & Sulke, S. R. (2014). E-learning using cloud computing. *International Journal of Computer Science and Mobile Computing (IJCSMC)*, 3(5), 1281–1287.

[2] Jella, K., & Kishore, B. (2015). A study on dynamic resource allocation using virtual machines for IaaS. *International Journal of Computer Engineering Research Trends*, 2(11), 761–764.

[3] Nathan, S., Bellur, U., & Kulkarni, P. (2016). On selecting the right optimizations for virtual machine migration. *ACM SIGPLAN Notices*, 51(7), 37–49. https://doi.org/10.1145/3007611.2892247.

[4] Piao, G., Oh, Y., Sung, B., & Park, C. (2014). Efficient pre-copy live migration with memory compaction and adaptive VM downtime control. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing* (pp. 85–90). IEEE. https://doi.org/10.1109/BDCloud.2014.57.

[5] Beloglazov, A., & Buyya, R. (2014). OpenStack Neat: A framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds. *Concurrency and Computation: Practice and Experience*, 32–36. https://doi.org/10.1002/cpe.3314.

[6] Greenberg, A., Hamilton, J., Maltz, D. A., & Patel, P. (2008). The cost of a cloud: Research problems in data center networks. *SIGCOMM Computer Communication Review*, 39(1), 68–73. https://doi.org/10.1145/1496091.1496103.

[7] Ferreto, T. C., Netto, M. A. S., Calheiros, R. N., & De Rose, C. A. F. (2011). Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems*, 27(8), 1027–1034. https://doi.org/10.1016/j.future.2011.04.016.

[8] Verma, A., Ahuja, P., & Neogi, A. (2008). pMapper: Power and migration cost aware application placement in virtualized systems. In *Middleware '08: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware* (pp. 243–264). Springer. https://doi.org/10.1007/978-3-540-89856-6_13.

[9] Hermenier, F., Lorca, X., Menaud, J. M., Muller, G., & Lawall, J. (2009). Entropy: A consolidation manager for clusters. In *VEE '09: Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments* (pp. 41–50). ACM. https://doi.org/10.1145/1508293.1508300.

[10] Xu, J., & Fortes, J. A. B. (2010). Multi-objective virtual machine placement in virtualized data center environments. In *2010 IEEE/ACM International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing (GREENCOM-CPSCOM)* (pp. 179–188). IEEE. https://doi.org/10.1109/GreenCom-CPSCom.2010.137.

[11] Falkenauer, E. (1996). A hybrid grouping genetic algorithm for bin packing. *Journal Unknown*, [Details not provided]. https://doi.org/10.1007/BF00226291.

[12] Dong, J., Jin, X., Wang, H., Li, Y., Zhang, P., & Cheng, S. (2013). Energy-saving virtual machine placement in cloud data centers. In *2013 IEEE/ACM 13th International Symposium on Cluster, Cloud and Grid Computing (CCGrid)* (pp. 618–624). https://doi.org/10.1109/CCGrid.2013.107.

[13] Gulati, A., Holler, A., Ji, M., Shanmuganathan, G., Waldspurger, C., & Zhu, X. (2012). VMware distributed resource management: Design, implementation, and lessons learned. *Journal Unknown*, [Details not provided].

[14] Wood, T., Shenoy, P., Venkataramani, A., & Yousif, M. (2007). Black-box and gray-box strategies for virtual machine migration. In *NSDI'07: Proceedings of the 4th USENIX conference on Networked Systems Design and Implementation* (p. 17). USENIX Association.

[15] Singh, A., Korupolu, M., & Mohapatra, D. (2008). Server-storage virtualization: Integration and load balancing in data centers. In *2008 International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2008)* (pp. 1–12). https://doi.org/10.1109/SC.2008.5222625.

[16] Arzuaga, E., & Kaeli, D. R. (2010). Quantifying load imbalance on virtualized enterprise servers. In *WOSP/SIPEW '10: Proceedings of the First Joint WOSP/SIPEW International Conference on Performance Engineering* (pp. 235–242). ACM. https://doi.org/10.1145/1712605.1712641.

[17] Liu, Z., Zhao, A., & Liang, M. (2021). A port-based forwarding load-balancing scheduling approach for cloud datacenter networks. *Journal of Cloud Computing*, 10, Article 13. https://doi.org/10.1186/s13677-021-00226-w.

[18] Shabeer, H. A., Ramaswamy, P. P., Zubar, H. A., & Banu, R. S. D. W. (2020). Editorial: Green Cloud Computing and Communication. *Mobile Networks and Applications*, 25, 1287–1289. https://doi.org/10.1007/s11036-020-01553-z.

[19] Awate, A., Nikhale, S., & Chunarkar, P. (2013). A path for horizing your innovative work environmental sustainability and the green cloud computing. *International Journal of Pure and Applied Research in Engineering and Technology (IJPRET)*, 1(8), 537–545.

[20] Kamble, R. S., & Nikam, D. A. (2013). Green cloud computing: New approach of energy consumption. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 3(2).

[21] Kaur, G., & Kumar, P. (2013). Compositional framework of green cloud. *International Journal of Emerging Trends in Engineering and Development*, 1(3). https://doi.org/10.1109/AICERA.2012.6306716.

[22] Kumara, M., & Sharma, S. C. (2017). Dynamic load balancing algorithm for balancing the workload among virtual machines in cloud computing. *Procedia Computer Science*, 115, 322–329. https://doi.org/10.1016/j.procs.2017.09.141.

[23] Chen, H., Bu, Y., Zong, K., Huang, L., & Hao, W. (2022). The effect of data skewness on the LSTM-based mooring load prediction model. *Journal of Marine Science and Engineering*, 10, Article 1931. https://doi.org/10.3390/jmse10121931.