Journal of Information Systems and Telecommunication
Vol.12, No.4, October-December 2024, 280-290
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

An Energy-Aware Approach to Virtual Machine Consolidation Using Classification and the Dragonfly Algorithm in Cloud Data Centers

Nastaran Evaznia[1*], Reza Ebrahimi[1] , Davoud Bahrepour [1,2]

[1].Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.
[2].Department of Cybersecurity and Cyberspace, Intelligent Financial Innovation Research Center, Mashhad Branch, Islamic Azad University, Mashhad, Iran.

Received: 16 Sep 2024/ Revised: 01 Dec 2024/ Accepted: 11 Feb 2025

Abstract
Nowadays, reducing energy consumption in cloud computing is of great interest due to the high operational costs and its impact on climate change. The consolidation solution is an effective method for minimizing the number of physical machines (PMs) and reducing energy consumption. The virtual machine consolidation process encounters the challenge of reducing energy consumption while effectively managing resource allocation. The aim of this paper is to address these challenges through the classification of PMs and the use of the dragonfly algorithm. The quartile parameter is utilized to classify PMs into three categories: underloaded, medium load, and overloaded. First, we identified the overloaded PMs in the overloaded category. Then, we presented a solution to select virtual machines from an overloaded PM based on resource usage. Additionally, the Dragonfly algorithm is utilized to select destinations for hosting migrant virtual machines in the medium load category. Furthermore, we identified underloaded PMs in the underloaded categories using this algorithm. The proposed solution is evaluated using the CloudSim toolkit and tested with workloads consisting of over a thousand data points from virtual machines based on PlanetLab data. The results from the simulation experiments indicate that the proposed solution, while avoiding SLA violations and minimizing additional migrations, has significantly reduced energy consumption.

Keywords: Cloud Computing; Consolidation; Quartile Parameter; Dragonfly Algorithm; SLA Violations; Migrations; Energy Consumption.

1- Introduction

Data centers in cloud computing [1-4] are physical spaces where hosts and necessary equipment are stored and accessed via the Internet to provide better services. Increasing energy consumption in cloud infrastructure can lead to higher carbon dioxide emissions and elevated operating costs [5-7]. With the advancement of virtualization, reducing energy consumption has become an important and challenging issue in the design of new systems [8-12]. To address this, a key strategy for reducing the number of active hosts is the virtual machine consolidation process, which enhances resource utilization by strategically migrating virtual machines(VMs) [13, 14]. This approach not only minimizes energy consumption but also strives to prevent violations of Service Level Agreements (SLAs) to the greatest extent possible [14-17].

Despite the significant benefits of this solution, inefficient consolidation can lead to increased costs. Research has shown that a server consumes about 70% of its energy when idle [13]. As a result, if underloaded and overloaded servers are not properly identified, they can contribute to increased migration, energy consumption, and violations of quality of service. Therefore, in this paper, we aim to answer the following question: How can the virtual machine consolidation process be conducted in a way that improves resource management and minimizes costs, while considering energy costs and migration?
Today, meta-heuristic algorithms are recognized as effective methods for solving complex problems and optimizing various fields. Their ability to explore large solution spaces and find optimal outcomes makes them invaluable for addressing diverse challenges [18]. Therefore, this paper presents a combined approach that integrates physical machines (PMs) classification with the Dragonfly meta-heuristic algorithm. In this approach, PMs are first classified using quartile criteria and categorized

✉ **Nastaran Evaznia**
Nastaran_avaznia@yahoo.com; nastaran.evaznia@iau.ir

into underloaded, medium load, and overloaded groups. A quartile divides a dataset into four equal parts, each representing a specific percentage of the data [19]. The proposed classification is based on server CPU usage, as research indicates that CPU performance significantly impacts energy consumption [13]. Consequently, this classification can effectively reduce energy consumption by accurately identifying underloaded, medium load, and overloaded PMs, while also preventing violations of quality of service standards. Therefore, in the proposed solution, we first identify the overloaded PMs within the overloaded category. Next, one or more VMs from these PMs need to be migrated to alleviate the overload condition. A multi-criteria solution based on RAM and CPU usage is proposed to determine which VMs should be migrated, thereby minimizing unnecessary migrations by selecting VMs appropriately. Additionally, a multi-criteria Dragonfly algorithm is utilized for the underloaded and medium load categories to identify the most suitable hosts. The improved Dragonfly algorithm, considering a multi-criteria fitness function, targets hosts with lower energy consumption and greater available resources to meet energy reduction goals.
Thus, the main innovations of this paper are summarized as follows:

1. Providing a solution to classify PMs based on quartile parameters.
2. Selecting migrating virtual machines from overloaded hosts based on multiple criteria to avoid excessive migrations.
3. Identifying underloaded and medium-load PMs using the improved Dragonfly algorithm, employing a multi-criteria fitness function to reduce the number of active servers and overall energy consumption.

The structure of the paper is organized as follows: In Section Two, we review and critique related works. Next, the proposed method is introduced. In Section Four, we analyze the proposed method. Finally, in Section Five, we present the conclusion.

2- Related Works

Mustafa et al. [20] proposed an energy-optimal and SLA-aware method in the consolidation process. To achieve this, two consolidation methods are presented to select the destination for hosting migrating VMs based on the Best Fit Decreasing (BFD) method. Simulation results demonstrate improvements in energy efficiency and a reduction in SLA violations. To reduce energy and improve SLA, Dabhi and Thakor [21] addressed the destination selection mechanism for the migration VM allocation. In this framework, the performance of the destination physical machine's processor is evaluated, and hosts with an average load are selected. Furthermore, the results demonstrate the performance improvements of the proposed approach. Researcher in [22] has presented a virtual machine consolidation algorithm aimed at optimizing the use of VMs to influence the balance between energy consumption and quality of service. This algorithm selects VMs for consolidation based on resource usage. For migration, it employs criteria such as the distance between hosts and the fulfillment of quality of service requirements. The simulation results indicate a better balance between energy consumption and service quality compared to other methods. Khalid et al. [23] focus on energy optimization through virtual machine consolidation. For VM consolidation, they employ mechanisms based on dynamic thresholds and adaptive migration of VMs. The proposed algorithm seeks to balance energy efficiency and performance by identifying overused hosts and relocating VMs to underutilized hosts. The simulation results of this paper demonstrate a reduction in energy consumption while maintaining high-quality services for users in the cloud infrastructure. Ali et al. [24] emphasize the importance of addressing energy consumption and security issues in cloud computing. To achieve this, they utilize particle swarm optimization (PSO) and colony optimization (CO) techniques. The simulation results indicate a reduction in costs and an increase in efficiency. Shaw et al. [24] present a virtual machine consolidation method using a reinforcement learning algorithm. In this paper, the reinforcement learning algorithm for the consolidation problem represents resource capacity to optimize the distribution of VMs, thereby improving resource management. The experimental results show that avoiding violations of the service level agreement enhances energy efficiency. Researchers have introduced the Modified Bird Feeding Algorithm (ModAFBA) in [25] as a solution for the VM consolidation process, aiming to enhance resource management and efficiency in cloud infrastructure. The simulation results reveal a reduction in energy consumption and the number of migrations, while preventing violations of quality of service. Patel and Bhadka [26] present two computational frameworks for allocation and migration. In this structure, a placement technique is employed to find the best location for each request based on the typical data center configuration of servers. Additionally, a list of VMs is calculated using a power model for migration, targeting those that consume more power. Furthermore, the destination is selected using Dolphin optimization, considering the server with the maximum workload. The experimental results indicate a reduction in energy consumption and the amount of migration. Manikandan and Janani [27] propose a solution that combines hybrid fuzzy and k-means clustering with black widow method optimization and fish swarm optimization for efficient resource allocation. The results of the tests demonstrate a reduction in costs and energy consumption. A summary of the studied methods is presented in Table 1.

Table 1: Summary searches in the area of cloud computing focus on energy awareness

Method	*overloaded PMs*	*VM selection*	*Destination selection*	*Underloaded PM*
Mustafa et al. [20]	×	×	✓	×
Dabhi and Thakor [21]	✓	×	✓	✓
Kumaran et al. [22]	×	✓	×	×
Khalid et al. [23]	✓	✓	×	✓
Ali et al. [24]	×	×	✓	×
Shaw and Barrett [28]	✓	×	✓	×
Alsadie and Alsulami [25]	×	✓	✓	×
Patel and Bhadka [26]	✓	✓	✓	×
Manikandan and Janani [27]	×	×	✓	×
Proposed Method	✓	✓	✓	✓

Based on Table 1, the papers are categorized according to the steps of consolidation, clarifying which steps each paper focuses on. The proposed method, which classifies PMs and appropriately categorizes them while utilizing the multi-criteria Dragonfly algorithm, offers an effective solution for optimal resource management at each consolidation stage.

3- Proposed Method

The consolidation strategy occurs in four stages. The first step is to identify the overloaded PMs. Next, if a PM is overloaded, one or more VMs must be migrated from that host to avoid SLA violations. In the third stage, the strategy focuses on finding the destination to host the migrating virtual machines. Finally, it identifies underloaded PMs to shut down [15, 17]. In the proposed method, the quartile parameter is used to categorize PMs within the cloud infrastructure, and the Dragonfly algorithm is employed to enhance mapping and reduce the number of PMs. First, the PMs are sorted by CPU usage, as CPU usage affects the energy consumption of PMs [13, 22]. Then, the first (Q1), second (Q2), and third (Q3) quartiles are calculated based on this. The PMs are divided into three categories: underloaded, medium load, and overloaded PMs based on the quartile parameter. Fig. 1 illustrates this classification of PMs.

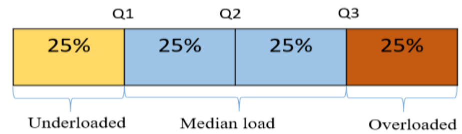

Fig. 1 Classification of PMs

According to Fig. 1, a PM whose CPU usage is less than Q1 is categorized as underloaded, while a PM whose CPU usage is in the range of Q2 is categorized as medium load. A PM with CPU usage greater than Q3 is classified as overloaded. First, we identify the overloaded PMs.

3-1- Identification of Overloaded Physical Machines

According to the consolidation steps, the first step in the consolidation phase is to identify overloaded PMs. Researchers in [29] proposed four suitable approaches to find dynamic thresholds for detecting overloaded PMs. Compared to external models, the Median Absolute Deviation (MAD) method is robust. In this phase, the MAD is used to identify overloaded PMs among those in the overloaded category. Eq. (1) provides this metric.

$$\text{If } (PM_i^{CPU} > \text{Q3}) \quad i=1, 2,\dots, N \qquad \{ T_u = 1 - s.\ \text{OC\_MAD} \qquad (1)$$

In Eq. (1), PM_i^{CPU} is CPU usage of PM_i, T_u is the upper threshold, $s \in R^+$, and OC_MAD is MAD in the overloaded category. N is the number of PMs.

MAD parameter uses previous knowledge to generate a new threshold value. To obtain a MAD value, it is necessary to use univariate data $X_1, X_2, \dots, X_n$. Eq. (2) expresses this criterion [20, 29].

$$\text{MAD} = \text{median}(|X_i - median_j(X_j)|) \qquad (2)$$

Hence, if the host's CPU usage in the overloaded category is greater than T_u, that host is considered to be overloaded.

In the event that the PM is overloaded, it would be necessary to migrate several VMs from that PM to prevent service quality violations. It is assumed that cloud centers include an N number of PMs and a V number of VMs. In the next step, we check what virtual machine to choose for migration from the overloaded PM.

3-2- Selection of Migrant Virtual Machines

In the previous works [20, 29], the Minimum Migration Time (MMT) policy is used to choose a VM to migrate from a host that was overloaded. The virtual machine has been selected for migration under this policy due to its reduced memory usage. In addition, merely one criterion is considered. In this policy, the amount of CPU used by the VM, which might be influential in overloading and increasing the energy consumption of the physical machine, has not been taken into account. In the proposed solution, the minimum memory maximum processor method has been presented, which aims to combine the MMT policy and the use of the virtual machine processor by considering several criteria. The purpose of presenting the desired method is to choose a virtual machine that has the lowest migration time compared to other virtual machines and uses more processors than the other virtual machines; therefore, we can reduce the number of additional migrations with this selection. Eq. (3) provides this criterion.

$$\upsilon \in \mathrm{V_i} | \forall \alpha \in \mathrm{V_i}. \frac{\mathrm{RAM_u}(\upsilon)}{\mathrm{UtilizationOfCpu_u}(\upsilon)} \leq \frac{\mathrm{RAM_u}(\alpha)}{\mathrm{UtilizationOfCpu_u}(\alpha)} \quad (3)$$

According to Eq. (3), $\mathrm{RAM_u}$ (α) is the recently used amount of RAM by virtual machine α. $\mathrm{UtilizationOfCpu_u}(\alpha)$ is the value of the recently used processor by virtual machine α. $\mathrm{V_i}$ is a set of VMs that have been recently allocated to the $\mathrm{host_i}$. In the proposed solution, a virtual machine with a lower ratio than that of other VMs is selected as the designated virtual machine for migration from the overloaded host. After the migration, this criterion is applied again to select the next virtual machine if the overloaded host's performance remains above the threshold.

3-3- Selection of the Destination Host

The next step in the consolidation phase, after selecting the migration VMs, is to choose a destination for hosting these migrated virtual machines.

After identifying the overloaded hosts and migrating the necessary virtual machines, the migrated virtual machines should be transferred to a destination with the required capacity and cost-effective efficiency. To select the destination for hosting the migrating virtual machines, the PM is chosen based on the proposed Dragonfly algorithm from the PMs in the medium load category. PMs in the medium load category have CPU utilization greater than Q1 and less than Q3 according to Eq. (4).

If (PM_i^{CPU}>Q1 and PM_i^{CPU}<Q3) (4)
i=1, 2, ...,N
⇨ *find PMs based on Dragonfly algorithm*

In Eq. (4), *Q1* is the first quarter, and *Q3* is the third quarter. PM_i^{CU} shows the CPU usage of the PM_i.

The reason for choosing this category is that it mitigates the risk of overloading the PMs in the future while also avoiding classification in the underloaded category, which could prevent the shutdown of that host later. Furthermore, in the proposed Dragonfly algorithm, a multi-criteria fitness function is employed to identify the best host for allocation within this category.

In general, due to its high speed, accuracy, and capabilities, the Dragonfly algorithm [30] has been utilized alongside a multi-criteria fitness function. As a result, the steps of the proposed method are as follows:

Step 1: The Dragonfly population and food sources, along with their characteristics, are quantified. In modeling the proposed solution using the Dragonfly algorithm, dragonflies represent VMs that search for prey (PMs). PMs possess characteristics such as processors, memory, network, and bandwidth. Consequently, these resources are considered within a broad set of constraints and can be represented by three parameters: CPU, RAM, and bandwidth (BW). Therefore, if PM_i represents the i-th physical machine, the available capacity of this PM (AC (PM_i)) is expressed in Eq. (5).

$$\mathrm{AC}\ (\mathrm{PM_i}) = \{\mathrm{CPU_i}, \mathrm{RAM_i}, \mathrm{BW_i}\} \quad (5)$$

Step 2: In this step, the fitness function for all PMs (resources) is calculated. By defining a multi-criteria fitness function and evaluating several criteria, we aim to select a host for allocation that possesses the necessary resources for hosting while avoiding the risk of overloading and violating the quality of service. Consequently, among the PMs, those that are not overloaded and meet the necessary resource requirements for hosting the virtual machine can be selected based on the proposed function. The limit is calculated using Eq. (6) and Eq. (7), respectively.

$$\mathrm{RR}(VM_i)<\mathrm{AC}\ (\mathrm{PM_i}) \qquad i=1, 2,\ldots,N \quad (6)$$

In Eq. (6), the $\mathrm{RR}(VM_i)$ indicates the resources required by the virtual machine, while AC ($\mathrm{PM_i}$) represents the available capacity of the i-th PM to prevent overloading. Consequently, the fitness function (FF) is calculated using Eq. (7).

$$\mathrm{FF} = \mathrm{MIN}\left(\frac{Energy\ (PM_i)}{\mathrm{AC}(PM_i)}\right) \quad (7)$$

According to the FF in Eq. (7), Energy (PM_i) represents the amount of energy consumed by PMi, and AC (PMi) is the capacity of the available resources of the PM. Based on the fitness function, the lower the energy consumption ratio of the host and the capacity of the available resources (RAM, CPU, and BW), the more suitable the position would be for the prey in the proposed Dragonfly algorithm (i.e., more suitable in the proposed host method). Therefore, in this structure, the host where the fitness function criterion is minimized compared to other PMs (positions) would be the most suitable host for the destination selection process for migrant dragonflies (migrant virtual machines).

Step 3: Update the optimal position (based on the fitness function). The position of the prey (PM) is updated for all migrant dragonflies (migrant virtual machines).

Step 4: Check the termination conditions. If the immigrant Dragonfly (representing the immigrant virtual machine) is not yet finished, repeat the process from the second step to select the host. Otherwise, if all dragonflies have been mapped, the termination condition is satisfied.

The pseudocode at this stage of the consolidation process is shown in Fig. 2.

```
Input: pmList, vmList     Output: allocated Host
For each VM in vmList do
   Allocated Host ← NULL;
   MinFitness ← MAX;
   For host in HostList do
      If utilization of cpu(PM_i)>Q1 and utilization of cpu(PM_i)<Q3
            Energy (PM_i) ← energy (host, VM);
            FF ← Energy (PM_i) / AC (PMi)
            If  fitness < MinFitness then
               MinFitness ← fitness;
               AllocatedHost ← host;
         end if
      end if
   end for
end for
return allocated Host;
```

Fig. 2 Destination PMs selection pseudocode

3-4- Identification of the Underloaded Physical Machines

To identify underloaded PMs, among the PMs whose CPU usage is less than the first quartile (Q1), the PM with the lowest fitness function metric in the Dragonfly algorithm is considered an underloaded PM for shutdown. Eq. (8) provides this feature. In Eq. (8), PM_i^{CU} is CPU usage of PM_i, and N is the number of PMs.

$$\text{If } (PM_i^{CPU} < \text{Q1}) \quad i=1, 2, \ldots, N \tag{8}$$

⇨ *find PMs based on fitness function in Dragonfly algorithm*

The dynamic VM consolidation process periodically migrates and reallocates VMs to PMs [31]. Therefore, the proposed method periodically addresses the issue of migrating VMs and reallocating them to medium load PMs, as well as putting underloaded PMs to sleep.

In the following, the flowchart diagram to express the proposed method is presented in Fig. 3.

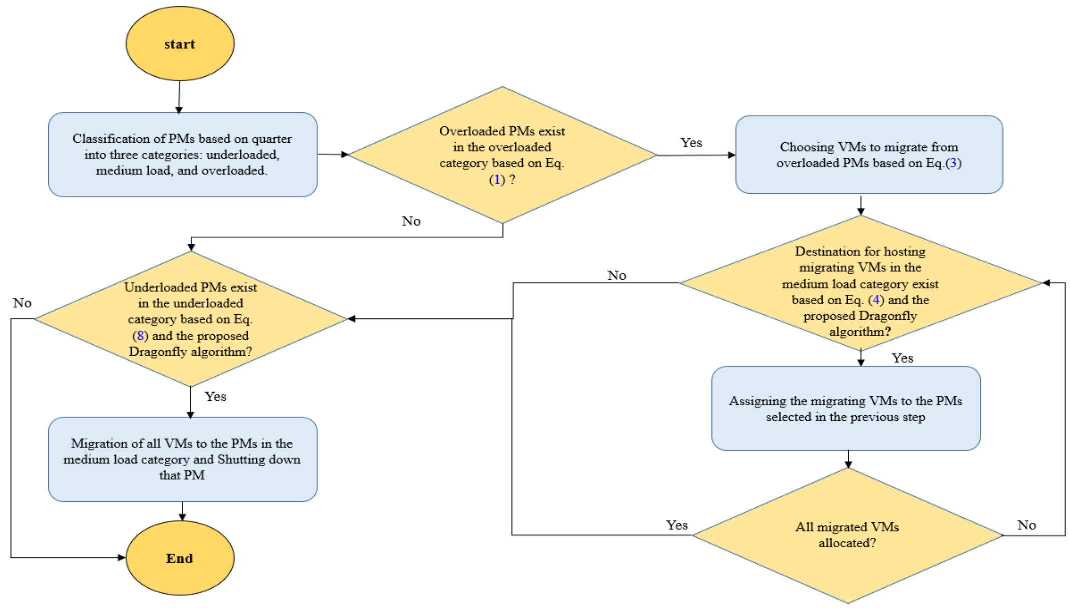

Fig. 3 Flowchart of the proposed method

According to Fig. 3, the steps of the proposed method include classifying PMs based on quartiles and applying the dragonfly algorithm in the consolidation process. First, the PMs are divided into three categories according to the quartile parameter. In the first step of consolidation, in the overloaded category (hosts whose CPU usage exceeds Q3), overloaded hosts are identified based on MAD [20]. If an overloaded host is found, it selects VMs to migrate from that host based on Eq. (3). After selecting the migrating VMs, the destination host is chosen from the medium load category using the Dragonfly algorithm. Based on the provided fitness function, the host with the minimum fitness value is selected for allocation. If no overloaded PMs are identified the method then looks for underloaded PMs in the underloaded category (hosts whose CPU utilization is below Q1 and have the minimum fitness function value in the Dragonfly algorithm for this category). This process continues until all VMs are allocated.

4- Analysis

Evaluating the proposed method in large-scale cloud data center infrastructures and real-world environments can be challenging. Therefore, to ensure the repeatability of the experiment, the CloudSim toolkit [32] has been chosen as the simulation platform. To assess the proposed method, we used real-world workloads for a more accurate evaluation. The workload utilized is based on actual data generated by PlanetLab. Using this tool, a data center with 800 heterogeneous physical nodes was simulated. In this test environment, half of the servers are HP ProLiant ML110 G4, while the other half are HP ProLiant ML110 G5. The specifications of these servers are provided in Table 2, according to the results of the SPECpower benchmark[1].

Table 2: Specifications of physical machines

Physical machine	*Bandwidth (Gbit/s)*	*The number of cores*	*RAM (MB)*	*CPU (MIPS)*
HPProLiant ML110 G4	1	2	4096	1860
HP ProLiant ML110 G5	1	2	4096	2660

Each category of PMs in Table 2 has different processing speeds, memory capacities, number of cores, and bandwidths. Additionally, the specifications of the VMs used are based on real Amazon EC2 examples[2]. In this architecture, all VMs in the dataset are single-core.

[1] http://www.spec.org/power_ssj2008

[2] http://aws.amazon.com/ec2/instance-types/

The specifications and features of the virtual machines used for evaluation are provided in Table 3.

Table 3: Specifications of virtual machines

virtual machine	***Micro instance***	***Small instance***	***Extra-large instance***
CPU (MIPS)	500	1000	2000
RAM(MB)	613	1700	3750
Bandwidth (Gbit/s)	1	1	1
Size (GB)	2.5	2.5	2.5

In Table 3, the virtual machines (VMs) differ in terms of processing speed and memory capacity. The workload in the presented method is based on real data obtained from PlanetLab over ten different days. This data reflects CPU usage by more than 1,000 VMs simultaneously from servers located in over 500 locations. For this purpose, ten days were randomly selected from the workflow data collected between March and April 2011 [33]. The characteristics of the dataset used to evaluate the results are shown in Table 4 [33].

Table 4: Specifications of PlanetLab data

Date	***Number of VMs***
03/03/2011	1052
06/03/2011	898
09/03/2011	1061
22/03/2011	1516
25/03/2011	1078
03/04/2011	1463
09/04/2011	1358
11/04/2011	1233
12/04/2011	1054
20/04/2011	1033

The proposed algorithm and the comparison method were coded using NetBeans software and CloudSim version 3, and executed on a 64-bit system with 8 GB of RAM.

4-1- The results of the simulation

The criteria and the parameters considered for evaluating the proposed method are energy efficiency, migrations, and service level agreement violation (SLAV). The energy consumption parameter is based on processor efficiency [20]. Given that processor efficiency changes over time, the energy criterion is defined as a function of time according to the processor's efficiency, as expressed in Eq. (9) [20, 27].

$$E_i = \int_{t0}^{t1} P\big(u(t_i)\big)dt \tag{9}$$

According to Eq. (9), E_i, the total amount of energy used by the *i*-th physical machine, is calculated as the integral of energy efficiency over a period from t_0 to t_1. $u(t_i)$ represents the utilization rate of the *i*-th physical machine's processor as a function of time.

Additionally, the SLAV parameter, which is entirely unfavorable in cloud infrastructure, contributes to increased costs. This criterion depends on two main factors: the state of hosts being overloaded and the occurrence of additional migrations. Specifically, these factors are represented by SLAV Time per Active Host (SLATAH) and Performance Degradation due to Migration (PDM). Consequently, these criteria are examined in Eq. (10) and Eq. (11) [20, 27].

$$\text{SLATAH} = \frac{1}{M}\sum_{j=1}^{M}\frac{T_{si}}{T_{ai}} \tag{10}$$

Let *M* be the number of hosts, and T_{si} represent the total time that the *i*-th host experiences 100% utilization, which results in a SLAV. Furthermore, *Tai* estimates the total time of the *i*-th PM in an active state. In the following section, the parameter PDM is detailed in Eq. (11) [20, 27].

$$\text{PDM}=\frac{1}{N}\sum_{i=1}^{N}\frac{C_{dj}}{C_{rj}} \tag{11}$$

based on Eq. (11), *N* indicates the number of v VMs, C_{dj} estimates the efficiency violation of the *j*-th VM caused by the migration, while C_{rj} represents the total capacity required by the *j*-th VM during its execution. Considering the equal importance of these two criteria in service quality violations, a combined criterion that accounts for both parameters is utilized for the SLAV measurement. This parameter is presented in Eq. (12) [20, 27].

$$\text{SLAV} = \text{SLATAH} * \text{PDM} \tag{12}$$

The works considered for comparison are the Energy and SLA-Aware VM Placement (ESVMP) [21] and the Black-widow and Fish Swarm Optimization (BWFSO) [27]. These papers were chosen due to the compatibility of their methods with the simulation environment and their utilization of meta-heuristic algorithms. Fig. 4 displays the results of the energy consumption for the proposed method alongside the compared methods, based on PlanetLab data.

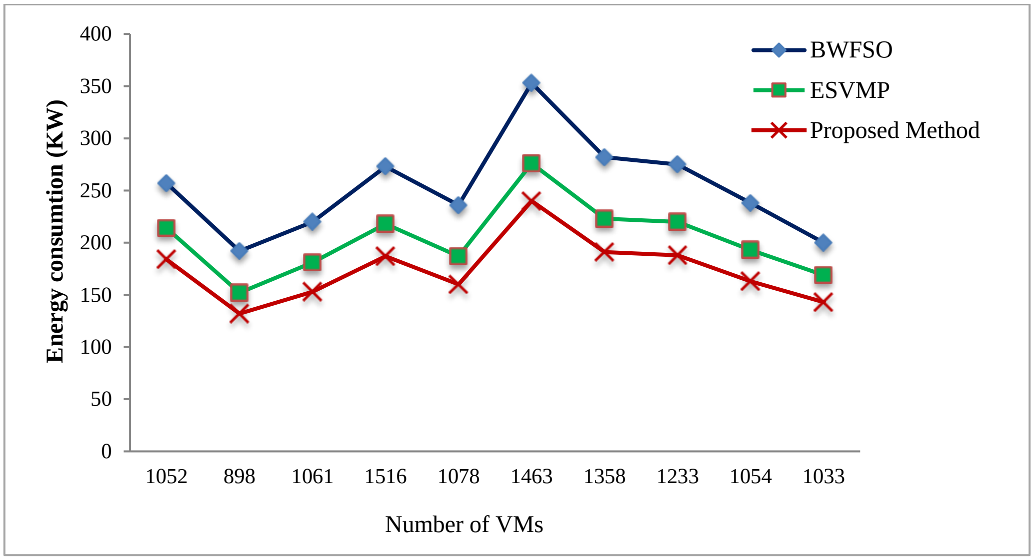

Fig. 4 Comparison of Energy Consumption

In Fig. 4, the amount of energy consumed in kilowatts at different times is displayed on the vertical axis, utilizing PlanetLab data. As illustrated, the energy consumption of the proposed method is significantly lower than that of the compared methods. This reduction is due to the classification of PMs based on the quartile parameter and the use of the Dragonfly algorithm with a multi-criteria objective function during the consolidation stages. Our solution effectively improves and reduces costs, including energy consumption, by optimizing resource management. Specifically, energy consumption is reduced by 14% compared to ESVMP and 31% compared to BWFSO. Fig. 5 presents the number of migrations for the proposed method compared to other methods.

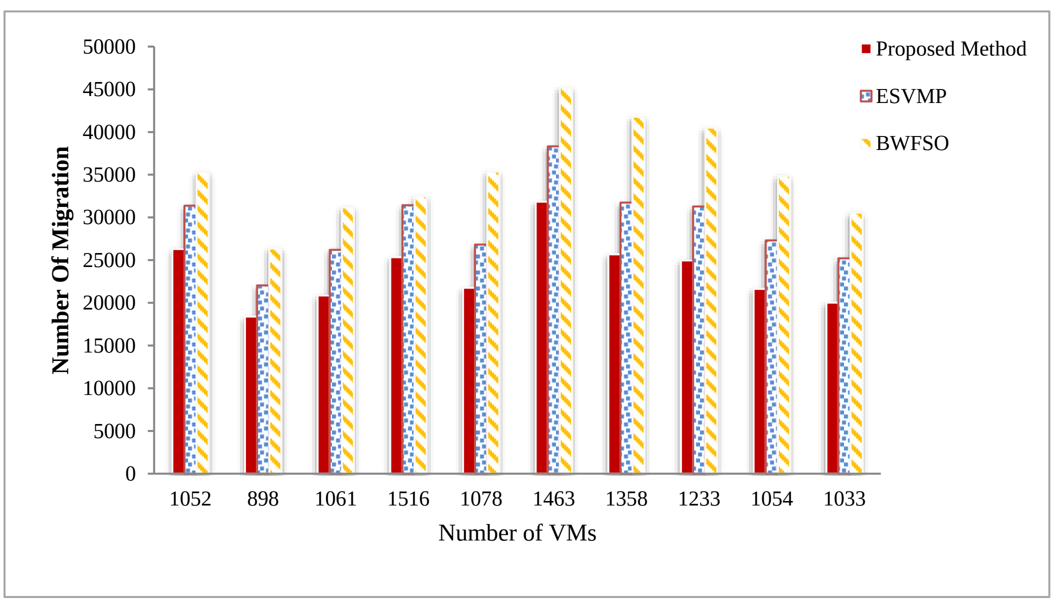

Fig. 5 Comparison of the Number of Migrations.

In Fig. 5, the number of migrations is plotted on the vertical axis using Planet Lab data. As shown, the proposed method results in fewer migrations compared to the baseline papers, with improvements of 19% compared to ESVMP and 33% compared to BWFSO. Fig. 6 compares the SLAV of the proposed method with ESVMP and BWFSO.

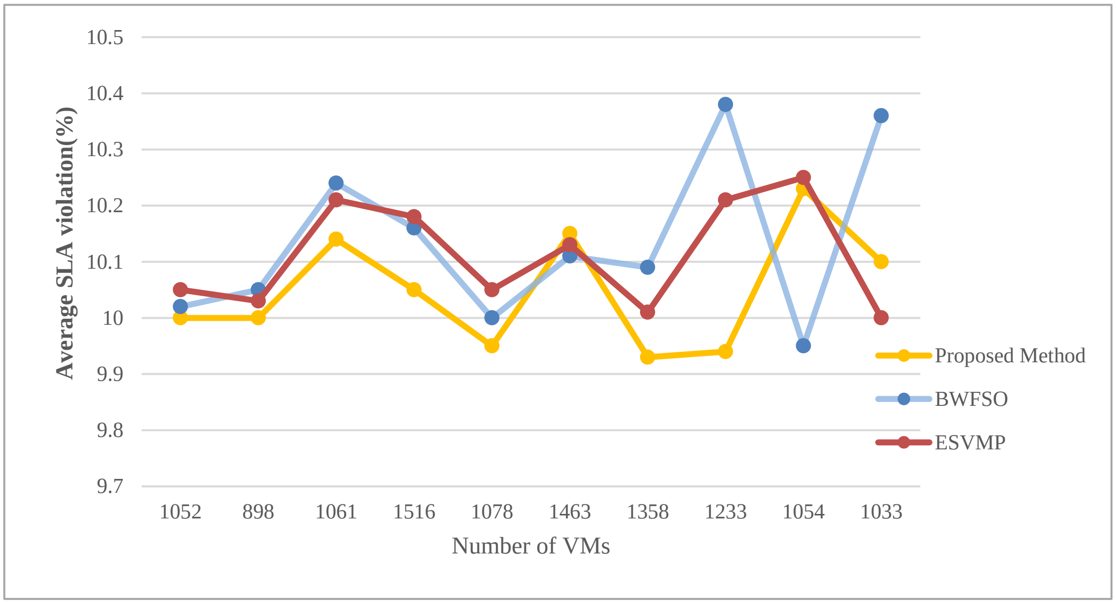

Fig.6 Comparison of the Percentage of SLAV.

In Fig. 6, the average SLAV violation is demonstrated. The results show that, by considering multiple criteria and categorization, the proposed method performs better by reducing energy consumption while avoiding violations of the service level agreement. Additionally, it has improved by 1% compared to ESVMP and 2% compared to BWFSO. The graphs indicate that increasing the number of VMs leads to higher energy consumption and more migrations. However, our proposed method, which categorizes PMs appropriately and utilizes the multi-criteria Dragonfly algorithm, demonstrates improved performance in reducing both energy consumption and the number of migrations, while also preventing an increase in SLAV.

5- Conclusions

In recent years, the growing demand for cloud services has made energy consumption optimization a critical issue. High energy usage in data centers negatively impacts operational costs and the environment. To maximize the benefits of cloud services and reduce expenses, it is essential to minimize energy consumption while adhering to Service Level Agreements. The process of virtual machine consolidation can effectively optimize energy consumption by reducing the number of active physical machines (PMs) and shutting down idle servers. However, improper consolidation can increase energy usage and negatively affect service quality. To address these challenges, this paper proposes a hybrid solution that combines PMs classification with a meta-heuristic algorithm to optimize energy consumption and manage resources effectively. PMs are categorized based on processor utilization using the quartile parameter, as optimal processor utilization is essential for minimizing energy consumption. By accurately identifying PMs within the appropriate categories, we can achieve improved energy efficiency and more effective resource management. Additionally, by identifying migrating virtual machines based on several criteria, we can prevent unnecessary migrations that increase costs. Furthermore, the use of the Dragonfly algorithm with a multi-criteria fitness function based on energy consumption and available resources helps us find suitable destinations for hosting migrating virtual machines. Finally, we identify underloaded PMs in the underloaded category using the proposed Dragonfly algorithm and take steps to shut them down, thereby reducing energy consumption. The performance of the proposed method has been evaluated using real workloads in the CloudSim simulator. The simulation results demonstrate that, compared to the first and second papers, energy consumption decreased by 14%

relative to ESVMP and by 31% compared to BWFSO. Additionally, the total number of migrations was reduced by 19% compared to ESVMP and by 33% compared to BWFSO, while the SLAV was decreased by 1% and 2% respectively. For future work, it is recommended to incorporate fog computing into the proposed method to further reduce latency. Moreover, focusing on the healthcare sector and integrating this approach could effectively lower user costs.

Abbreviations

SLA	Service Level Agreement
PMs	Physical Machines
SLAV	Service Level Agreement Violation
BFD	Best Fit Decreasing
ModAFBA	Modified Feeding Birds Algorithm
PM	Physical Machine
MAD	Medium Absolute Deviation
VMs	Virtual Machines
MMT	Minimum Migration Time
SLATAH	SLAV time per active host
PDM	Performance Degradation due to Migration
ESVMP	Energy and SLA-aware VM Placement
BWFSO	Black-widow and Fish Swarm Optimization
K-means	K-means refers to data classification with the aim of partitioning n data into k clusters.

References

[1] T. Alam, "Cloud Computing and Its Role in the Information Technology," SSRN Electron. J., vol. 1, no. 2, 2020 , pp. 108–115, doi: 10.2139/ssrn.3639063.

[2] M. Yenugula, S. Sahoo, and S. Goswami, "Cloud computing for sustainable development: An analysis of environmental, economic and social benefits," J. Futur. Sustain., vol. 4, no. 1, 2024, pp. 59–66, doi: 10.5267/j.jfs.2024.1.005.

[3] M. M. Sadeeq, N. M. Abdulkareem, S. R. M. Zeebaree, D. M. Ahmed, A. S. Sami, and R. R. Zebari, "IoT and Cloud computing issues, challenges and opportunities: A review," Qubahan Acad. J., vol. 1, no. 2, 2021, pp. 1–7, doi: https://doi.org/10.48161/qaj.v1n2a36.

[4] S. S. Fateminasab, S. Memarian, S. R. K. Tabbakh, and M. C. Romero-Ternero, "A Review on Open Data Storage and Retrieval Techniques in Blockchain-based Applications," in 2024 10th International Conference on Web Research (ICWR), IEEE, 2024, pp. 297–302, doi: 10.1109/ICWR61162.2024.10533356.

[5] W. Yao, Z. Wang, Y. Hou, X. Zhu, X. Li, and Y. Xia, "An energy-efficient load balance strategy based on virtual machine consolidation in cloud environment," Futur. Gener. Comput. Syst., vol. 146, 2023, pp. 222–233, doi: 10.1016/j.future.2023.04.014.

[6] F. Tashtarian, M. F. Zhani, B. Fatemipour, and D. Yazdani, "CoDeC: A cost-effective and delay-aware SFC deployment," IEEE Trans. Netw. Serv. Manag., vol. 17, no. 2, 2019, pp. 793–806, doi: 10.1109/TNSM.2019.2949753.

[7] S. Shahryari, F. Tashtarian, and S.-A. Hosseini-Seno, "CoPaM: Cost-aware VM Placement and Migration for Mobile services in Multi-Cloudlet environment: An SDN-based approach," Comput. Commun., vol. 191, 2022, pp. 257–273, doi: 10.1016/j.comcom.2022.05.005.

[8] R. Shaw, E. Howley, and E. Barrett, "An energy efficient anti-correlated virtual machine placement algorithm using resource usage predictions," Simul. Model. Pract. Theory, vol. 93, 2019, pp. 322–342, doi: 10.1016/j.simpat.2018.09.019.

[9] C. Thiam and F. Thiam, "Energy efficient cloud data center using dynamic virtual machine consolidation algorithm," in Business Information Systems: 22nd International Conference, BIS 2019, Seville, Spain, June 26–28, 2019, Proceedings, Part I 22, Springer, 2019, pp. 514–525, doi: 10.1007/978-3-030-20485.

[10] D. Bahrepour, N. Evaznia, and T. Khodabakhshi, "A New Resource Allocation Method Based on PSO in Cloud Computing," Int. J. Web Res., vol. 7, no. 2, 2024 , pp. 13–21, doi: 10.22133/ijwr.2024.457539.1216.

[11] N. Evaznia and R. Ebrahimi, "Providing a Solution for Optimal Management of Resources using the Multi-objective Crow Search Algorithm in Cloud Data Centers," in 2023 9th International Conference on Web Research (ICWR), IEEE, 2023, pp. 179–184, doi: 10.1109/ICWR57742.2023.10139192.

[12] S. S. F. Nasab, T. Z. Marjaneh, and D. Bahrepour, "Energy Efficiency and Establishing Service Level Agreement using Fuzzification of Virtual Machine Selection Policies for Migrating in Cloud Computing," in 2023 9th International Conference on Web Research (ICWR), IEEE, 2023, pp.201–207, doi: 10.1109/ICWR57742.2023.10138982.

[13] A. Varasteh, F. Tashtarian, and M. Goudarzi, "On reliability-aware server consolidation in cloud datacenters," in 2017 16th International Symposium on Parallel and Distributed Computing (ISPDC), IEEE, 2017, pp. 95–101, doi: 10.1109/ISPDC.2017.26.

[14] L. Helali and M. N. Omri, "A survey of data center consolidation in cloud computing systems," Comput. Sci. Rev., vol. 39, 2021, p. 100366, doi: 10.1016/j.cosrev.2021.100366.

[15] R. Zolfaghari and A. M. Rahmani, "Virtual machine consolidation in cloud computing systems: Challenges and future trends," Wirel. Pers. Commun., vol. 115, no. 3, 2020, pp. 2289–2326, doi: 10.1007/s11277-020-07682-8.

[16] M.-H. Malekloo, N. Kara, and M. El Barachi, "An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments," Sustain. Comput. Informatics Syst., vol. 17, 2018, pp. 9–24, doi: 10.1016/j.suscom.2018.02.001.

[17] U. Arshad, M. Aleem, G. Srivastava, and J. C.-W. Lin, "Utilizing power consumption and SLA violations using dynamic VM consolidation in cloud data centers," Renew. Sustain. Energy Rev., vol. 167, 2022, p. 112782, doi: 10.1016/j.rser.2022.112782.

[18] H. F. Farimani, D. Bahrepour, S. R. K. Tabbakh, and R. Ghaemi, "A new meta-heuristic algorithm: Artificial Yellow Ground Squirrel (YGSA)," 2022, doi: 10.21203/rs.3.rs-1909482/v1.

[19] H. F. Farimani, D. Bahrepour, and S. R. K. Tabbakh, "Reallocation of virtual machines to cloud data centers to reduce service level agreement violation and energy consumption using the FMT method," J. Inf. Syst. Telecommun., vol. 4, no. 28, 2020, p. 316.

[20] S. Mustafa, K. Bilal, S. U. R. Malik, and S. A. Madani, "SLA-aware energy efficient resource management for cloud environments," IEEE Access, vol. 6, 2018, pp. 15004–15020, doi: 10.1109/ACCESS.2018.2808320.

[21] D. Dabhi and D. Thakor, "Energy and SLA-Aware VM Placement Policy for VM Consolidation Process in Cloud Data Centers," in Sustainable Technology and Advanced Computing in Electrical Engineering: Proceedings of ICSTACE 2021, Springer, 2022, pp. 351–365, doi: 0.1007/978-981-19-4364-5_26.

[22] K. M, "Energy-Aware Virtual Machine Consolidation Algorithm for Enhanced QoS in Data Centers," Int. Sci. J. Eng. Manag., vol. 03, no. 04, 2024, pp. 1–9, doi: 10.55041/ISJEM01696.

[23] U. Khalid, S. Ahmad, B. Chang, M. Nisar, J. Cha, and E. Munir, Energy Optimization in Cloud Computing Environmentthrough Virtual Machine Consolidation. 2023. doi: 10.21203/rs.3.rs-3284176/v1.

[24] A. Ali and T. T. Tin, "Unleashing the Power of Consolidate Cloud Computing: Secure and Energy-Efficient Virtual Machines at Your Service," 2023, doi: 10.21203/rs.3.rs-3133236/v1.

[25] D. Alsadie and M. Alsulami, "Efficient Resource Management in Cloud Environments: A Modified Feeding Birds Algorithm for VM Consolidation," Mathematics, vol. 12, no. 12, 2024, p. 1845, doi: 10.3390/math12121845.

[26] R. P. Patel and H. B. Bhadka, "Energy-Aware VMs Consolidation Computing Frameworks' of Data Center in Cloud Computing Environment," J. Sci. Technol., vol. 7, no. 1, 2022, pp. 82–91.

[27] N. Manikandan, P. Divya, and S. Janani, "BWFSO: hybrid Black-widow and Fish swarm optimization Algorithm for resource allocation and task scheduling in cloud computing," Mater. Today Proc., vol. 62, 2022, pp. 4903–4908, doi: 10.1016/j.matpr.2022.03.535.

[28] R. Shaw, E. Howley, and E. Barrett, "Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers," Inf. Syst., vol. 107, 2022, p. 101722, doi: 10.1016/j.is.2021.101722.

[29] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurr. Comput. Pract. Exp., vol. 24, no. 13, 2012, pp. 1397–1420, doi: 10.1002/cpe.1867.

[30] Ç. İ. Acı and H. Gülcan, "A modified dragonfly optimization algorithm for single-and multiobjective problems using Brownian motion," Comput. Intell. Neurosci., vol. 2019, no. 1, 2019, p. 6871298, doi: 10.1155/2019/6871298.

[31] H. Xiao, Z. Hu, and K. Li, "Multi-objective VM consolidation based on thresholds and ant colony system in cloud computing," IEEE Access, vol. 7, 2019, pp. 53441–53453, doi: 10.1109/ACCESS.2019.2912722.

[32] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Softw. Pract. Exp., vol. 41, no. 1, 2011, pp. 23–50, doi: 10.1002/spe.995.

[33] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," ACM SIGOPS Oper. Syst. Rev., vol. 40, no. 1, 2006, pp. 65–74, doi: 10.1145/1113361.1113374.