Memory Priority Scheduling Algorithm for Cloud Data Center Based on Machine Learning Dynamic Clustering Algorithm

Bin Liang and Di Wu

***Abstract*—As the cloud data center (CDC) landscape continues to broaden, CDC resource utilization as the benchmark for assessing scheduling methodologies. Concurrently enhancing both CPU and memory utilization stands as a paramount priority. However, prevalent algorithms tend to solely prioritize CPU utilization while neglecting memory efficiency, ultimately escalating energy expenditure. This article initiates by conducting a comprehensive examination of memory utilization repercussions on CDCs. Subsequently, it facilitates the dynamic clustering of physical machines and virtual machine deployments, ensuring a balanced utilization profile. Furthermore, it introduces the memory priority scheduling algorithm for CDC based on machine learning dynamic clustering algorithm (PMPD). Comparative evaluations against other algorithms underscore the prowess of PMPD in concurrently optimizing CPU and memory utilization, thereby minimizing the number of active PMs and diminishing energy consumption of CDCs.**

***Index Terms*—Cloud data center (CDC), clustering algorithm, machine learning, virtual machine (VM).**

I. Introduction

AMONG the diverse strategies aimed at enhancing utilization and mitigating energy consumption, optimizing scheduling algorithms stands as the most straightforward and potent approach. Cloud service providers leverage these algorithms to elevate utilization, thereby decreasing the number of operational physical machines (PMs) and ultimately reducing the energy consumption of cloud data center (CDCs). While existing scheduling algorithms have demonstrated performance improvements [1], [2], [3], [4]. First, while PM specifications in idealized CDCs may be uniform, it vary significantly due to temporal disparities in reality, introducing complexity into scheduling algorithms. Second, the diverse nature of cloud user services necessitates scheduling algorithms that can adapt to a wide range of VM applications. Finally, the ever-growing scale of CDCs poses a significant challenge to the performance of scheduling algorithms.

To tackle the development hurdles of CDCs, this article embarks on a comprehensive analysis of the ramifications of memory utilization within these systems. Subsequently, it achieves a dynamic clustering of PM and VM allocations, guided by the principle of utilization equilibrium. Introducing the PMPD algorithm, this article concludes by deploying it on a platform for rigorous validation.

The cornerstone contributions of this article encompass.

1) A thorough analysis of how CPU and memory utilization influence the efficacy of PMs.
2) A dynamic clustering rule for PM, leveraging utilization balance in conjunction with machine learning-based clustering techniques.
3) The establishment of VM deployment rule for VM deployment.
4) The formulation and validation of the PMPD algorithm.

The rest of this article is organized as follows. Section II delves into the existing research landscape of CDCs. Section III outlines the clustering of PMs and the deployment for VMs. Section IV introduces the PMPD algorithm. Section V presents the verification process and comparative analysis of the PMPD algorithm. Finally, Section VI concludes this article.

II. Existing Research

This article is divided mainly into VM deployment, optimizing the performance of CDCs and reducing the energy consumption of CDCs.

A. Related Research on VM Deployment

Zhou et al. [5] introduced an innovative mechanism, AFED-EF, which is adaptive and energy-conscious for VM allocation and deployment in IoT applications. This algorithm excels in managing load variations and demonstrates superior performance in VM allocation and arrangement processes. Belgacem et al. [6] employs a machine learning approach aimed at minimizing both the frequency of VM migrations and energy consumption. The novel virtual machine learning migration strategy focuses on refining the VM migration process and selection criteria. Zhou et al. [7] introduces two innovative adaptive algorithms that are sensitive to energy consumption, designed to optimize energy efficiency and reduce the rate of service level

Received 8 September 2024; revised 13 November 2024; accepted 6 January 2025. Date of publication 22 January 2025; date of current version 4 April 2025. This work was supported by the Shaanxi Natural Science Foundation Basic Research Program under Grant 2023-JC-YB-601. Paper no. TII-24-4562. *(Corresponding author: Di Wu.)*

Bin Liang is with Xi'an Shiyou University, Xi'an 710065, China (e-mail: liangbinhehe@stu.xjtu.edu.cn).

Di Wu is with the Haojing College of Shaanxi University of Science and Technology, Xi'an 712046, China (e-mail: 130708@xsyu.edu.cn).

Digital Object Identifier 10.1109/TII.2025.3528574

agreement violations in CDCs. Chen et al. [8] developed a power and thermal-conscious VM Dynamic Scheduling Framework tailored for CDCs. This framework adapts VM consolidation approaches dynamically through real-time surveillance of server temperatures and resource utilization, while accounting for thermal recycling dynamics within the data centers confines. Pandey and Rawat [9] propose an innovative hybrid methodology called ARIMA-RM. The primary goal of this advanced ARIMA-RM model is to refine resource allocation and scheduling strategies, ultimately leading to superior performance. Zou et al. [10] carried out an exhaustive exploration of the core procedures involved in VM consolidation.

B. Related Research on Optimizing CDC Performance

Khabbaz and Assi [11] endeavor to enhance the quality of service metrics within data centers, adhering to the aforementioned benchmarks. Yang et al. [12] present a novel algorithm for task allocation in CDCs, utilizing the Stoer–Wagner binary tree approach to expedite computational processes. Choudhary et al. [13] devise a metaheuristic-driven workflow scheduling algorithm, prioritizing the reduction of manufacturing time and costs. Ren et al. [14] propose a method known as VM placement based on an enhanced sparrow search algorithm, which incorporates Levy flight disturbance to enhance the diversity of resource allocation during the search process. Metwally et al. [15] propose a distributed framework for managing geographically dispersed data centers, logically organized into regions. Yao et al. [16] introduce a classification algorithm specifically designed for PMs experiencing load anomalies, considering both current and predicted loads to reduce VM migrations. Wang et al. [17] maximize the profitability of multiservice providers by integrating network resource scheduling in cloud radio access networks with computing resource allocation in mobile edge computing, offering a unified approach to balancing power consumption and performance. Li et al. [18] present a two-phase approach to minimize service costs while ensuring task security and timely completion for cloud users. Tong et al. [19] introduce deep Q-learning task scheduling, a cutting-edge AI-driven algorithm that merges the advantages of Q-learning and deep neural networks.

C. Related Research on Reducing the Energy Consumption of CDCs

Ding et al. [20] introduce an innovative dynamic VM integration system that utilizes adaptive performance and power ratio sensing. This framework effectively addresses the balance between performance enhancement and energy efficiency, leveraging predictions of heterogeneous host resource utilization. Ghasemi et al. [21] present MRRL, a novel approach that initially clusters VMs using the k-means algorithm, followed by the deployment of a tailored reinforcement learning strategy with multiple reward signals for optimized placement. Liang et al. [22] delve into the influence of variations in cloud task parameters on VM execution time. By leveraging these differences, they propose a revised mapping sequence for cloud tasks, aiming to expedite VM execution and minimize power consumption of PMs. Luo et al. [23] strives to redefine virtual machine migration as a partitioning challenge and presents a holistic framework that skillfully assesses workload conditions and accurately pinpoints the optimal migration destination, thereby reducing the costs linked to virtual machine migration. Lin et al. [24] develop a comprehensive thermal model that incorporates CPU and server thermal dynamics to accurately capture non-uniform and dynamic thermal conditions. Ma et al. [25] propose a scheduling policy that integrates synchronous and asynchronous vacation periods, formulated as a two-dimensional Markov stochastic model encompassing multiple servers. Liang et al. [26] prioritize memory utilization when mapping cloud tasks and deploying VMs. This algorithm enhances the utilization of PMs and reduces the energy consumption of CDCs.

TABLE I
MODEL PARAMETERS

parameter	*meaning*
pm$_i$.nc	Number of CPU of PM$_i$
pm$_i$.nm	Memory size of PM$_i$
pm$_i$.uc	CPU utilization of PM$_i$
pm$_i$.um	Memory utilization of PM$_i$
pm$_i$.et	Execution time of PM$_i$
pm$_i$.lt	Light load time of PM$_i$
pm$_i$.rt	Remaining time of PM$_i$
pm$_i$.pc	Power consumption of PM$_i$
vm$_j$.nc	Number of CPU of VM$_j$
vm$_j$.nm	Memory size of VM$_j$
vm$_j$.nt	Required time of VM$_j$

TABLE II
PARAMETERS OF THE PMS

PM	nc	nm	uc	um	rt
pm$_1$	18	72 GB	0.50	0.25	5000
pm$_2$	18	72 GB	0.50	0.50	3000

Traditionally, research endeavors have predominantly centered enhancing performance metrics or minimizing energy expenditure, yet these two objectives frequently exhibit contrasting trajectories. To bridge this gap, the present article delves into the intricate interplay between memory utilization and CDCs. Leveraging insights into balanced utilization, this work formulates a dynamic clustering rule for PMs and deployment rule for VM. Our research mainly addresses three issues related to CDC. First, simultaneously improve the CPU and memory utilization of PMs in CDC, and reduce the uneven utilization of PMs. Second, avoid prolonging runtime of PM due to new VM deployment. Finally, by reducing the energy consumption of PMs, the energy consumption of CDC can be lowered. Ultimately, the introduction of the PMPD algorithm represents a novel approach to addressing these intertwined challenges.

III. PM CLUSTERING AND VM DEPLOYMENT

A. PM Memory Utilization

The parameters that are used in this article and their meanings are given in Table I.

In reference to the specifications of both PMs and rented VMs within Amazon's CDCs. Assume the deployment parameters for these PMs and VMs are given in Tables II and III, respectively.

TABLE III
PARAMETERS OF THE VMS

VM	nc	nm	nt
vm_1	8	16 GB	720
vm_2	4	16 GB	1440
vm_3	4	32 GB	1440

TABLE IV
PM PARAMETERS AFTER FF ALGORITHM DEPLOYMENT

PM	nc	nm	uc	um	rt
pm_1	18	72 GB	0.94	0.47	5000
pm_2	18	72 GB	0.72	0.72	3000

TABLE V
PM PARAMETERS AFTER CONSIDERING THE UTILIZATION BALANCE

PM	nc	nm	uc	um	rt
pm_1	18	72 GB	0.94	0.92	5000
pm_2	18	72 GB	0.94	0.72	3000

When employing the first fit (FF) algorithm for VM allocation, a scenario arises where after deploying vm_1 and vm_2, insufficient capacity remains on both PMs to accommodate vm_3. Consequently, the current utilization status of the PMs as given in Table IV. It reveals that while both PMs have residual memory capacity, pm_1 is notably underutilized in terms of memory. To rectify this imbalance and optimize resource usage, the scheduling algorithm should prioritize placing VMs with lower ratio of CPU count to memory capacity requirements onto PMs. Among the available VMs, vm_3 stands out with the ratio of CPU count to memory capacity of 1:8, making it an ideal candidate for pm_1, given its low memory utilization. Additionally, to further enhance the overall utilization of PMs, the algorithm adjusts the placement of vm_1 and vm_2, relocating vm_1 to pm_2 and vm_2 to pm_1. The ultimate deployment configuration as given in Table V. It not only successfully accommodates all three VMs simultaneously but also significantly improves the CPU and memory utilization rates of both PMs. This approach ensures optimal resource allocation and higher operational efficiency within the CDC environment.

B. Dynamic Clustering Rule of PMs

To enhance the performance efficiency of PMs, this article adopts a machine learning-inspired clustering methodology to devise a dynamic clustering rule for PMs. This approach groups PMs with similar balanced utilization into distinct clusters, thereby facilitating optimized VM deployment.

Rule 1. Dynamic Clustering Rule: Which employs an unsupervised learning algorithm to cluster active PMs based on their CPU and memory balanced utilization. The clustering configuration is updated in real-time, accommodating changes incurred by each new VM deployment.

To elucidate the clustering process, we consider a hypothetical scenario involving six PMs, modeled after popular processor types of Amazon: Intel Xeon e5-2686 V4, AWS Graviton2, and Intel Xeon p-8175m. The specifications of these PMs are given in Table VI. For the purpose of this example, we set the number of cluster centers to three. Due to the 2:1 ratio of CPU to memory utilization in pm_1 and pm_5, they belong to the same cluster center. The CPU to memory utilization ratio of pm_2, pm_3, and pm_4 is 1:1, so they are clustered together. Similarly, due to the CPU to memory utilization ratio of 1:2 for pm_6, it belongs to a separate cluster center. This ensures that PMs with the same ratio of CPU to memory utilization are in the same cluster center. By adapting the K-means clustering strategy, we iteratively refine the clustering, with the final cluster assignments given in Table VII. This process ensures that VMs are strategically allocated to PM clusters, promoting balanced resource utilization and enhanced overall performance.

TABLE VI
PM PARAMETERS FOR CLUSTERING

PM	nc	nm	uc	um	rt
pm_1	18	72 GB	0.50	0.25	5000
pm_2	18	72 GB	0.50	0.50	3000
pm_3	64	256 GB	0.25	0.25	2000
pm_4	64	256 GB	0.75	0.75	3000
pm_5	16	64 GB	0.50	0.25	2000
pm_6	16	64 GB	0.25	0.50	2000

TABLE VII
CLUSTER CENTERS OF K-MEANS

cluster center 1	cluster center 2	cluster center 3
pm_1	pm_2	pm_6
pm_5	pm_3	
	pm_4	

TABLE VIII
VM CLASSIFICATION RESULT

VM	cluster center
vm_1	3
vm_2	2
vm_3	1

C. VM Deployment Rule

The VM provisioning process comprises two pivotal stages. First, categorizing the VM into its corresponding cluster center, and subsequently selecting an eligible PM for deployment. This initial categorization ensures optimal PM utilization, thereby minimizing the number of active PMs. The subsequent step focuses on minimizing energy expenditure by the PMs.

Rule 2. VM Deployment Rule: Upon deploying a new VM instance, it is assigned to a cluster center based on its ratio of CPU count to memory capacity. Consequently, the VM is restricted to deploying on PMs within that cluster center. If multiple PMs within the hub have sufficient resources, the one that consumes the least energy, determined by its proximity to completing the deployment within the required timeframe, is chosen.

Utilizing the VM specifications from Table III and clustering outcomes from Table VII as illustrative examples, Table VIII gives the final VM classification. Due to the ratio of CPU count to memory capacity in VM_1 being 1:2, this ratio has exceeded

TABLE IX
PM PARAMETERS AFTER VM DEPLOYMENT

PM	nc	nm	uc	um	rt
pm_1	18	72 GB	0.50	0.25	5000
pm_2	18	72 GB	0.50	0.50	3000
pm_3	64	256 GB	0.31	0.31	2000
pm_4	64	256 GB	0.75	0.75	3000
pm_5	16	64 GB	0.75	0.75	2000
pm_6	16	64 GB	0.75	0.75	2000

the normal 1:4 ratio for PM. Therefore, in order to correct the CPU to memory utilization of PM, vm_1 will be classified into cluster center 3. At the same time, the ratio of CPU count to memory capacity for VM_2 has reached 1:4, which is in line with the conventional ratio for PM. Therefore, it is classified as cluster center 2. Using the same approach, in order to adjust the CPU to memory utilization ratio of PMs, VM_3 will be classified as cluster center 1. The final classification results are given in Table VIII. This ensures that no matter which PM the VM is deployed on in the cluster center, it will not exacerbate the imbalance in CPU and memory utilization of that PM, but will instead correct it. After completing the classification of VMs, each VM will search for a suitable PM within its cluster center to complete deployment. The basis that needs to be referred to the required time of the VM. The VM deployment rule will search for a PM with remaining time similar to the required time of the VM to complete the deployment, so as not to prolong the execution time of the host PM due to the deployment of the VM, thereby avoiding unnecessary energy consumption. In this example, since the cluster center 3 where vm_1 is located only has pm_6, it can only be deployed on pm_6. However, the cluster center 2 where VM_2 is located has three PMs, and the remaining time of PM_3 is 2000, which is closest to required time of VM_2. Therefore, VM_2 will be deployed on PM_3. The same VM_3 is located in cluster center 3, which has two PMs. The remaining time of PM_5 is 2000, which is closest to the required time of VM_3. Therefore, VM_3 will be deployed on PM_5. The parameters of PMs after completion are given in Table IX. In addition, when the VM deployed on the PM stops working, the PM enters sleep mode to reduce its energy consumption. Building upon these insights, we shall introduce the PMPD algorithm subsequently.

IV. PMPD ALGORITHM

Taking into account the preceding examination of PM memory usage, this article introduces a dynamic clustering approach for PMs, leveraging a machine learning-based clustering algorithm. Subsequently, guided by the established VM deployment rule, the categorization and placement of VMs are executed. Based on this comprehensive analysis and stipulated rules, we present the PMPD algorithm, formally outlined as Algorithm 1. This algorithm represents a novel contribution aimed at optimizing VM allocation and PM utilization.

To commence, a suitable cluster center is identified for the impending VM deployment (line 2). Following this, within the designated cluster center, a PM is meticulously chosen based on the VM deployment rule (lines 3–8). Upon successful VM deployment, the pertinent PM parameters undergo an update process to reflect the new configuration (lines 9 and 10). The flowchart for the PMPD algorithm in Fig. 1.

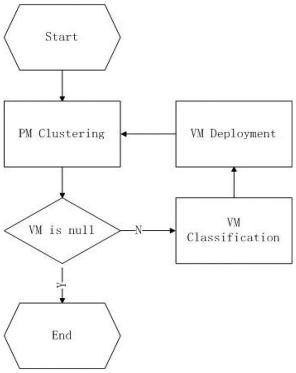

Fig. 1. Flowchart for the PMPD algorithm.

Algorithm 1: (PMPD).

```
Input: Deployed VM: VM = {$vm_1, vm_2, ..., vm_m$},
       Activated PM: PM = {$pm_1, pm_2, ..., pm_n$},
Output: Activated PM: PM = {$pm_1, pm_2, ..., pm_n$}
1  while VM not null do
2    Nocluster = FINDPM($vm_i$,PM)
3    while PM not null do
4      if(Nocluster = = True)
5        if($((1 - pm_j.uc) * pm_j.nc >= vm_i.nc)\&\&((1 - pm_j.um) * pm_j.nm >= vm_i.nm)$)
6          Min(abs($pm_j.rt - vm_i.nt$))
7          DeployedNo = j
8        end if
9      end if
10   end while
11    $pm_{DeployedNo}.uc = pm_{DeployedNo}.uc + (vm_i.nc/pm_{DeployedNo}.nc)$
12    $pm_{DeployedNo}.um = pm_{DeployedNo}.um + (vm_i.nm/pm_{DeployedNo}.nm)$
13   end while
```

On the one hand, the PMPD algorithm utilize dynamic clustering rule and VM deployment rule to complete PM clustering and VM deployment, thereby simultaneously improving the CPU and memory balanced utilization of PM in CDCs. Ultimately, it reduces the PM energy consumption of CDC, thereby improving its energy efficiency. On the other hand, it is commonly thought that the total energy consumption of CDC primarily consists of the energy used by PMs and the energy used for cooling. Since PMs contribute the most significant proportion to the overall energy consumption, while other forms of energy usage, like cooling, exhibit a consistent and direct correlation with PM energy consumption, minimizing PM energy usage can consequently lead to decreased cooling energy consumption. By examining the aforementioned two perspectives, it becomes evident that the PMPD algorithm effectively diminishes both PM

TABLE X
TEST PARAMETERS

parameter	value
Number of VMs	{10000, 20000, 30000, 40000, 50000}
CPUs of the VMs	{1, 2, 4}
Minimum PM utilization thresholds	{0.50, 0.60, 0.70}

and cooling energy consumption, thereby enhancing the energy efficiency of CDC.

V. EXPERIMENTAL VERIFICATION

This section delves into assessing the efficacy of the PMPD algorithm and compares it with the Best-First (BF), WS [27], TPO [28], and DE-ERPSO algorithm [24]. We analyze the effects of the PMPD algorithm. Initially, using the Alibaba Cluster Data from 2018, we evaluated the performance of the PMPD algorithm within the OpenStack environment. Nevertheless, because of the restricted size of our custom-built Infrastructure as a Service (IaaS) cloud platform, leading to insignificant consequences from the PMPD algorithm. Additionally, to overcome the limitations of the simulation setup, we integrated a simulation validation tool within CloudSim. Across three varied PM configurations, each distinguished by its CPU count, embodying the inherent heterogeneity of CDCs. The evaluation encompasses metrics such as number of VMs, VM capacity, and minimum PM utilization thresholds. Furthermore, this section dissects the light load time of PMs, average CPU and memory utilization, and CDC energy consumption across the PMPD algorithm. Their definitions are shown in formulas 1–4. Additional experimental parameters are given in Table X

$$\mathrm{T} = \sum_{i=1}^{m} pm_i.lt. \tag{1}$$

Among them, T represents the total light load time of PMs. The pm_i.lt represents the light load time of PM_i

$$\mathrm{UC} = \frac{\sum_{i=1}^{m} pm_i.uc}{m}. \tag{2}$$

Among them, UC represents the average CPU utilization of PMs. The pm_i.uc represents the CPU utilization of PM_i

$$\mathrm{UM} = \frac{\sum_{i=1}^{m} pm_i.um}{m}. \tag{3}$$

Among them, UM represents the average memory utilization of PMs. The pm_i.um represents the memory utilization of PM_i

$$\mathrm{E} = \sum_{i=1}^{m} pm_i.et \times pm_i.pc. \tag{4}$$

Among them, E represents the total energy consumption of PMs. The pm_i.et represents the execution time of PM_i. The pm_i.pc represents the power consumption of PM_i.

A. Number of VMs

First, the number of VMs determines the business capability and concurrency of cloud service providers.

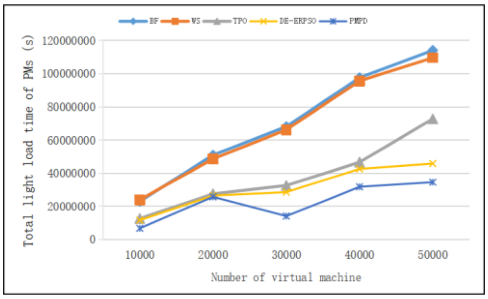

Fig. 2. Light load PM duration based on the number of VMs.

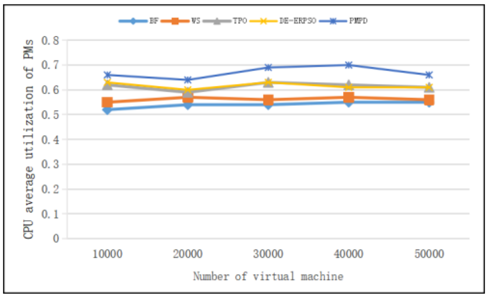

Fig. 3. Average CPU utilization based on the number of VMs.

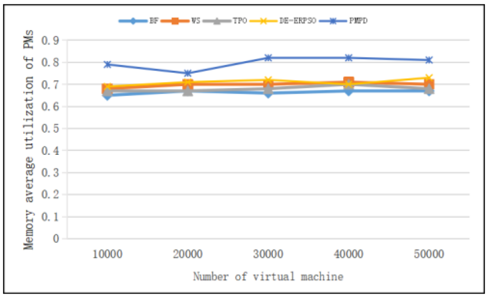

Fig. 4. Average memory utilization based on the number of VMs.

As depicted in Fig. 2, the PMPD algorithm exhibits a noteworthy decrease in light load PM processing module duration by 67%, 66%, 38%, and 29% respectively in comparison to the FF, WS, TPS, and DE-ERPSO algorithms. This substantial improvement stems primarily from the PMPD meticulous consideration of balancing CPU and memory utilization across individual PMs during cluster center selection, thereby enhancing overall resource efficiency and curtailing idle time of PM. Subsequently, Fig. 3 illustrates the enhanced average CPU utilization achieved by the PMPD algorithm, surpassing the FF, WS, TPS, and DE-ERPSO methods by 24%, 19%, 9%, and 8%, respectively, while maintaining a commendable 67% average CPU utilization rate. Moreover, Fig. 4 shows the average PM memory utilization figures, with the PMPD algorithm boasting an impressive 80% utilization. These favorable outcomes are attributed to the successful clustering of active VMs, ensuring that operational PMs maintain peak CPU and memory efficiency levels. Finally, Fig. 5 highlights a 20% reduction in energy consumption at the

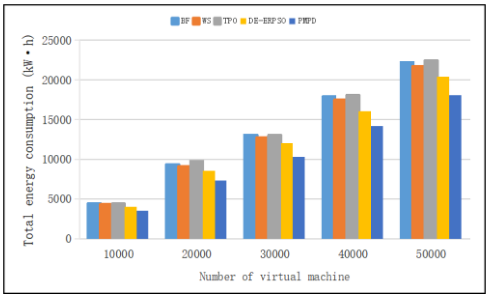

Fig. 5. CDC energy consumption based on the number of VMs.

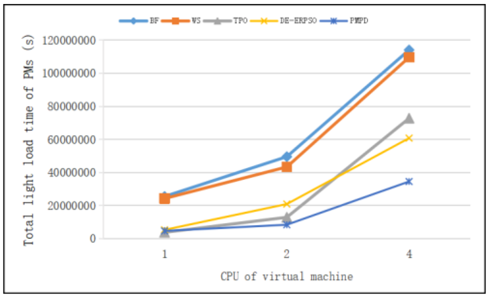

Fig. 6. Light load PM duration based on the capacities of VMs.

CDC level for the PMPD algorithm compared to its counterparts, FF, WS, TPS, and DE-ERPSO. This energy savings is primarily facilitated by the algorithm preference for deploying VMs on PMs that minimize energy expenditure.

B. VM Capacity

The varying demands of cloud users are mirrored in the diverse capacity configurations of VMs. The intricacies inherent in scheduling algorithms stem from the multifaceted requirements posed by these users. This experimental setup aligns closely with the authentic operational landscape of CDCs, ensuring a high degree of realism and practical applicability.

As shown in Fig. 6, the PMPD algorithm significantly decreases the light load PM processing module time by 78%, 76%, 16%, and 38%, respectively, in comparison to the FF, WS, TPS, and DE-ERPSO algorithms. This trend becomes even more pronounced as the VM capacity varies. Ultimately, the PMPD algorithm optimizes the utilization balance across PMs. Figs. 7 and 8 depict the average CPU and memory utilization achieved by the PMPD algorithm, showcasing average improvement of 15%. This concurrent enhancement in both CPU and memory utilization underscores positive impact on overall average utilization of the PMPD algorithm, particularly in terms of memory utilization, highlighting its ability to achieve high memory efficiency. As depicted in Fig. 9, the PMPD algorithm reduces energy consumption in the CDC by 18%, 17%, 20%, and 14%, respectively, compared to the FF, WS, TPS, and DE-ERPSO algorithms. This notable reduction is primarily attributed to the improved utilization stemming from the PM clustering rule employed by the algorithm.

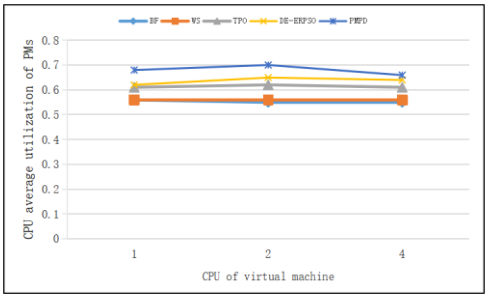

Fig. 7. Average CPU utilization based on the capacities of VMs.

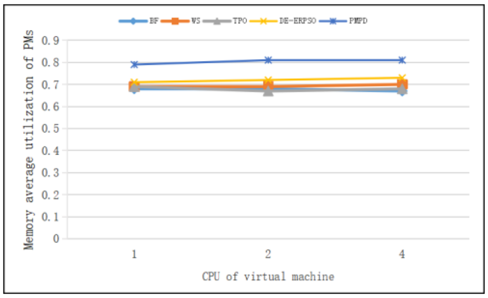

Fig. 8. Average memory utilization based on the capacities of VMs.

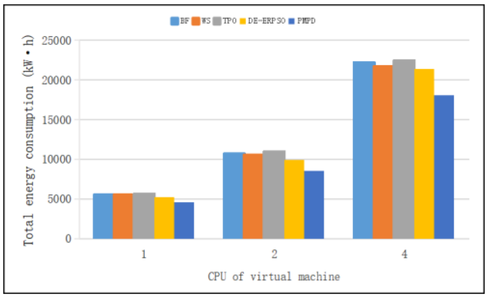

Fig. 9. CDC energy consumption based on the capacities of VMs.

C. Minimum PM Utilization Thresholds

The number of VMs that can be allocated to PMs is dictated by the minimum PM utilization thresholds set for this PMs. Additionally, these thresholds correspond to a specific range of energy consumption for the PMs. Consequently, conducting an in-depth analysis of these minimum utilization thresholds holds significant importance within the experimental framework.

As shown in Fig. 10, the PMPD algorithm achieves a substantial decrease in the idle period of processing PMs by 72%, 70%, 46%, and 54%, respectively, when compared to the FF, WS, TPS, and DE-ERPSO algorithms. This reduction translates to an enhanced utilization of PMs and a diminished requirement for active PMs. Figs. 11 and 12 show the average CPU and memory utilization achieved by the PMPD algorithm, displaying average improvement of 17%. Furthermore, as depicted in Fig. 13, the energy consumption of the CDC utilizing the PMPD algorithm is reduced by approximately 19% compared to the other four

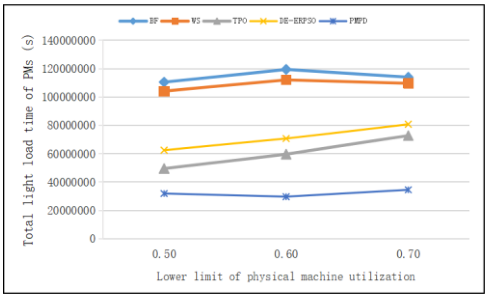

Fig. 10. Light load PM duration based on the minimum PM utilization thresholds.

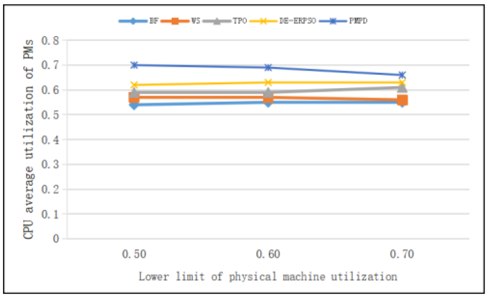

Fig. 11. Average CPU utilization based on the minimum PM utilization thresholds.

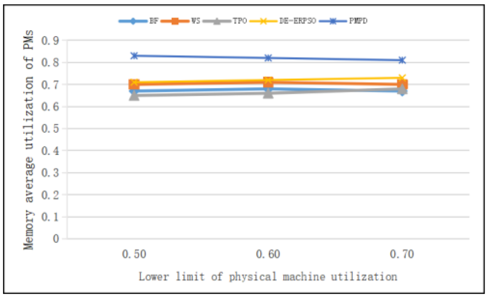

Fig. 12. Average memory utilization based on the minimum PM utilization thresholds.

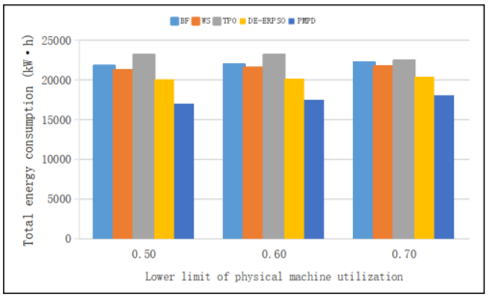

Fig. 13. CDC energy consumption based on the minimum PM utilization thresholds.

algorithms. This improvement is primarily attributed to the decrease in energy usage and the number of activated PMs, underscoring the algorithm efficiency and effectiveness.

VI. Conclusion

This article commences with a comprehensive analysis of the influence of memory utilization. Subsequently, the process of dynamically PM clustering processing and deploying VMs is executed, guided by the principle of utilization balance. Following this, the PMPD algorithm was introduced. The proposed algorithm was then implemented on a testing platform for validation. The empirical outcomes demonstrate that the PMPD algorithm boasts high CPU and memory utilization, while significantly curtailing the duration of light load conditions. Ultimately, this translates to a substantial reduction in the energy consumption of the CDC.

There are two main directions for future work. Firstly, analyze the supply mode of clean energy such as wind and solar energy for CDC energy consumption, and study the impact of periodic clean energy on CDC efficiency and energy consumption. Secondly, further analyze the effectiveness of other machine learning clustering and classification algorithms. Improve the deployment process of VM based on the characteristics of different algorithms.

References

[1] B. Guindani, D. Ardagna, A. Guglielmi, R. Rocco, and G. Palermo, "Integrating Bayesian optimization and machine learning for the optimal configuration of cloud systems," *IEEE Trans. Cloud Comput.*, vol. 12, no. 1, pp. 277–294, Jan./Mar. 2024.

[2] D. Fernando, J. Terner, P. Yang, and K. Gopalan, "V-recover: Virtual machine recovery when live migration fails," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 3289–3300, Jul./Sep. 2023.

[3] R. M. Haris, K. M. Khan, A. Nhlabatsi, and M. Barhamgi, "A machine learning-based optimization approach for pre-copy live virtual machine migration," *Cluster Comput.*, vol. 27, pp. 1293–1312, 2024.

[4] S. Vila, F. Guirado, and J. L. Lérida, "Cloud computing virtual machine consolidation based on stock trading forecast techniques," *Future Gener. Comput. Syst.*, vol. 145, pp. 321–336, 2023.

[5] Z. Zhou, M. Shojafar, M. Alazab, J. Abawajy, and F. Li, "AFED-EF: An energy-efficient VM allocation algorithm for IoT applications in a cloud data center," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 658–669, Jun. 2021.

[6] A. Belgacem, S. Mahmoudi, and M. A. Ferrag, "A machine learning model for improving virtual machine migration in cloud computing," *J. Supercomputing*, vol. 79, pp. 9486–9508, 2023.

[7] Z. Zhou et al., "Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms," *Future Gener. Comput. Syst.*, vol. 86, pp. 836–850, 2018.

[8] R. Chen, B. Liu, W. Lin, J. Lin, H. Cheng, and K. Li, "Power and thermal-aware virtual machine scheduling optimization in cloud data center," *Future Gener. Comput. Syst.*, vol. 145, pp. 578–589, 2023.

[9] M. C. Pandey and P. S. Rawat, "Virtual machine provisioning within data center host machines using ensemble model in cloud computing environment," *SN Comput. Sci.*, vol. 5, pp. 690–713, 2024.

[10] J. Zou, K. Wang, K. Zhang, and M. Kassim, "Perspective of virtual machine consolidation in cloud computing: A systematic survey," *Telecommun. Syst.*, vol. 87, pp. 257–285, 2024.

[11] M. Khabbaz and C. M. Assi, "Modelling and analysis of A novel deadline-aware scheduling scheme for cloud computing data centers," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 141–155, Jan./Mar. 2018.

[12] T. Yang, X. Han, H. Li, W. Li, and A. Y. Zomaya, "Parallel scientific power calculations in cloud data center based on decomposition-coordination directed acyclic graph," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 2491–2502, Jul./Sep. 2023.

[13] A. Choudhary, I. Gupta, V. Singh, and P. K. Jana, "A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing," *Future Gener. Comput. Syst.*, vol. 83, pp. 14–26, 2018.

[14] Q. Ren, B. Zhuge, Z. Zhang, L. Dong, and X. Jiang, "Improved sparrow algorithm based virtual machine placement," *Cluster Comput.*, vol. 27, pp. 6511–6525, 2024.

[15] K. Metwally, A. Jarray, and A. Karmouch, "A distributed auction-based framework for scalable IaaS provisioning in geo-data centers," *IEEE Trans. Cloud Comput.*, vol. 8, no. 3, pp. 647–659, Jul./Sep. 2020.

[16] 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," *Future Gener. Comput. Syst.*, vol. 146, pp. 222–233, 2023.

[17] X. Wang et al., "Dynamic resource scheduling in mobile edge cloud with cloud radio access network," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 11, pp. 2429–2445, Nov. 2018.

[18] L. Li, C. Zhou, P. Cong, Y. Shen, J. Zhou, and T. Wei, "Makespan and security-aware workflow scheduling for cloud service cost minimization," *IEEE Trans. Cloud Comput.*, vol. 12, no. 2, pp. 609–624, Apr./Jun. 2024.

[19] Z. Tong, H. Chen, X. Deng, K. Li, and K. Li, "A scheduling scheme in the cloud computing environment using deep Q-learning," *Inf. Sci.*, vol. 512, pp. 1170–1191, 2020.

[20] W. Ding, F. Luo, L. Han, C. Gu, H. Lu, and J. Fuentes, "Adaptive virtual machine consolidation framework based on performance-to-power ratio in cloud data centers," *Future Gener. Comput. Syst.*, vol. 111, pp. 254–270, 2020.

[21] A. Ghasemi, A. Toroghi Haghighat, and A. Keshavarzi, "Enhancing virtual machine placement efficiency in cloud data centers: A hybrid approach using multi-objective reinforcement learning and clustering strategies," *Computing*, vol. 106, pp. 2897–2922, 2024.

[22] B. Liang, X. Dong, Y. Wang, and X. Zhang, "A low-power task scheduling algorithm for heterogeneous cloud computing," *J. Supercomputing*, vol. 76, pp. 7290–7314, 2020.

[23] L. Luo, S. Wei, H. Tang, and C. Wang, "An effective partition-based framework for virtual machine migration in cloud services," *Cluster Comput.*, vol. 27, pp. 12899–12917, 2024.

[24] J. Lin, W. Lin, W. Wu, W. Lin, and K. Li, "Energy-aware virtual machine placement based on a holistic thermal model for cloud data centers," *Future Gener. Comput. Syst.*, vol. 161, pp. 302–314, 2024.

[25] Z. Ma, S. Guo, and R. Wang, "The virtual machines scheduling strategy based on M/M/c queueing model with vacation," *Future Gener. Comput. Syst.*, vol. 138, pp. 43–51, 2023.

[26] B. Liang, X. Dong, Y. Wang, and X. Zhang, "Memory-aware resource management algorithm for low-energy cloud data centers," *Future Gener. Comput. Syst.*, vol. 113, pp. 329–342, 2020.

[27] K. M. Tarplee, A. A. Maciejewski, and H. J. Siegel, "Robust performance-based resource provisioning using a steady-State model for multi-objective stochastic programming," *IEEE Trans. Cloud Comput.*, vol. 7, no. 4, pp. 1068–1081, Oct./Dec. 2019.

[28] F. López-Pires, B. Barán, L. Benítez, S. Zalimben, and A. Amarilla, "Virtual machine placement for elastic infrastructures in overbooked cloud computing datacenters under uncertainty," *Future Gener. Comput. Syst.*, vol. 79, pp. 830–848, 2018.

Bin Liang received the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, Xi'an , China in 2020.

He is currently a Teacher with the School of Computer Science, Xi'an Shiyou University, Xi'an, China. His research interests focus on cloud computing, energy-aware scheduling, cloud data center scheduling and blockchain model.

Di Wu received the master's degree in electronic information from Northwest Normal University, Lanzhou¸ China in 2023.

He is a Teacher with Haojing College, Shaanxi University of Science and Technology, Xi'an, China. His research interests include control principles, signal transmission and processing.