Performance Enhancement in Clustering Cloud Workload Trace Big Data for Cloud Data Center

Suhad A. Yousif[1], Susan M. Al Naqshbandi[2], Suzan J. Obaiys[3], Hayder H. Maala[4], and Nur Amalina Binti Jamaludin[5(✉)]

[1] Computer Science Department, Al-Nahrain University, Baghdad, Iraq
Suhad.a.yousif@nahrainuniv.edu.iq

[2] Computer Engineering Department, Komar University of Science and Technology (KUST), Sulaymaniyah, Iraq
susan.alnaqshbandi@komar.edu.iq

[3] Department of Computer System and Technology, Faculty of Computer Science and Information Technology, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
suzan@um.edu.my

[4] University of Technology, Baghdad, Iraq
hayder.h.maala@uotechnology.edu.iq

[5] Department of Mathematics, Centre for Foundation Defence Studies, National Defence University of Malaysia, 57000 Kuala Lumpur, Malaysia
amalinajamaludin@upnm.edu.my

Abstract. Cloud computing has received considerable interest from research institutions, developers, and individuals in the last few years. The complexity and scale increase of cloud environments encourage realistic workload-representative benchmarks and stronger performance analysis. This paper tackles energy-consumption enhancement in cloud data centers by proposing an energy-efficient strategy that clusters user tasks and then serves them according to CPU demand. The study uses the Google cluster trace, recorded from approximately 12,500 machines, as a real cloud environment. A large-scale clustering process was performed over about 50 million task records using k-means executed in SQL Server. The clustering stage produced three workload groups—high, medium, and low CPU usage—and the mean value of each cluster was later used as a threshold for task allocation. Based on these thresholds, a CPU-Based Allocation (CPUBA) approach was implemented in CloudSim. Experimental results show that strategic workload distribution over data-center machines improves resource utilization and reduces power consumption.

Keywords: Google Cluster Trace · Clustering · Performance Enhancement · K-means · Cloud Computing · Big Data · CloudSim · Power Consumption · CPU Rate

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16758, pp. 270–283, 2027.
https://doi.org/10.1007/978-3-032-30497-1_17

1 Introduction

Cloud computing has become a core paradigm for delivering scalable, on-demand services through shared and virtualized infrastructures. Its growth has transformed data centers into large, highly dynamic computing environments that support big data processing, scientific applications, and AI-driven services. At the same time, this growth has intensified concerns about energy usage, operational cost, and sustainable resource management, making energy-efficient scheduling and allocation central research problems in modern cloud systems [1,2].

A substantial body of recent work has focused on reducing energy consumption in cloud data centers through workload-aware consolidation, predictive resource management, and environmentally sustainable scheduling. Panwar et al. [1] reviewed energy-utilization management strategies for cloud data centers and showed the breadth of heuristic, metaheuristic, machine-learning, and statistical approaches used to reduce energy demand. Wang et al. [2] proposed an energy-aware VM-consolidation strategy based on combined prediction to reduce energy consumption, VM migrations, and SLA violations in CloudSim-based evaluation. These studies confirm that energy-aware allocation remains an active and practical research direction.

Within this direction, Clustering techniques play a central role in workload characterization for cloud environments. While k-means clustering [3] is widely adopted due to its simplicity and scalability, density-based methods such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) provide an alternative approach by identifying clusters based on data density rather than distance to centroids [4,5]. DBSCAN is particularly effective in detecting arbitrarily shaped clusters and handling noise, making it suitable for complex and heterogeneous datasets. Clustering-based workload analysis has remained especially relevant because it offers a lightweight and scalable way to characterize large traces and support allocation decisions. In work directly related to the present study, Yousif and Al-Dulaimy [6] presented a clustering-based analysis of cloud workload traces to improve cloud data-center performance. Maala and Yousif [7] later examined Google cluster-trace data and used k-means-based clustering over large-scale workload records to derive CPU-oriented groupings that support more suitable task distribution. In a related big-data clustering contribution, Sagheer and Yousif [8] proposed a parallel clustering model based on Hadoop multi-node deployment and Apache Mahout, highlighting Suhad A. Yousif's continued research interest in scalable clustering and big-data analysis. Together, these studies provide a clear methodological foundation for integrating workload clustering with efficient cloud resource management.

More recent literature has expanded the discussion from workload characterization toward thermal-aware, learning-based, and sustainability-oriented cloud control. Alissa et al. [9] proposed a holistic thermal-aware workload and infrastructure-control framework for heterogeneous data centers using machine learning, showing that workload placement and cooling control should be considered jointly. Wang et al. [2] developed a reinforcement-learning framework for

environmentally sustainable federated-cloud task scheduling, explicitly addressing both energy consumption and carbon emissions. Feng and Ding [10] surveyed cloud workload prediction from an application-oriented perspective and highlighted its importance for cost reduction, performance assurance, and energy optimization. These developments show that current cloud scheduling research increasingly combines predictive analytics, learning-based control, and sustainability objectives.

Recent studies have also explored hybrid intelligent scheduling schemes. [11] proposed a hybrid cuckoo-search and transformer-based task-scheduling approach for cloud environments and reported improvements in energy-aware scheduling performance. Such methods are promising, but they are often computationally heavier and more difficult to deploy than clustering-based approaches, especially when the target environment involves very large traces and simulation-driven evaluation. This creates space for practical methods that retain low complexity while still improving placement quality and energy efficiency.

Motivated by this gap, the present paper adopts a clustering-driven strategy for task classification and allocation in cloud data centers. The proposed CPU-Based Allocation (CPUBA) approach uses thresholds derived from large-scale workload-trace analysis to classify tasks according to CPU demand and then allocate them to virtual machines in a manner that improves utilization and reduces unnecessary activation of physical machines. By building on prior clustering-based workload studies while addressing current energy-efficiency concerns, the paper positions its contribution between classical trace-driven scheduling and newer intelligent resource-management approaches.

For evaluation, the work relies on real Google cluster-trace characteristics together with CloudSim-based experimentation, aiming to demonstrate that clustering-informed allocation can still offer a practical and effective path toward energy-aware cloud resource management.

2 Proposed Methodology

Clustered cloud computing is used for providing an environment for services of various categories. There is room for much performance improvement in the field of clustered cloud computing. The indicators considered in this paper include the reduction of power consumption, network and storage usage, in addition to waiting and scheduling time.

To get a more accurate understanding of the scenarios in which cloud data centers operate, real data collected and offered by Google was used. The data was recorded from a Google cloud cluster of approximately 12,500 machines for 29 days. Data collection, cleaning, and normalization operations were performed after downloading the released files using the `gsutil` tool, which provides access to Google Cloud Storage and supports tasks such as downloading, editing, and deleting stored data [12].

After downloading the data, normalization operations were used to convert the raw dump files into readable data and then gather the required files into

one SQL dataset. The k-means clustering algorithm was then performed over approximately 50 million CPU-rate records to produce three thresholds by big-data clustering. These thresholds represent high-, medium-, and low-CPU usage tasks. The resulting groups are used later by the proposed allocation method. The overall workflow is shown in Fig. 1.

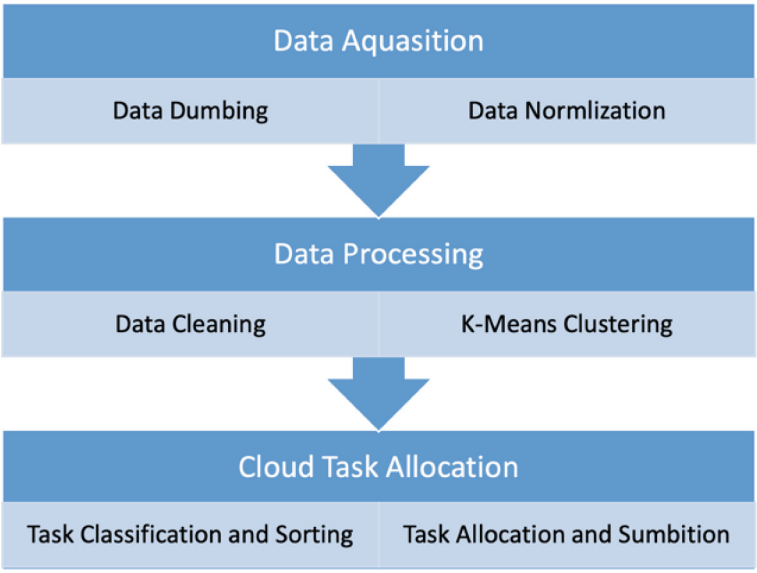

Fig. 1. Main stages of the proposed methodology.

Our previous work [7] describes the decompression workflow and the automatic import of 500 CSV files into SQL Server using a sequential C# implementation that preserves the original record ordering.

2.1 Modeling and Simulation with CloudSim

The objective of cloud computing is to power new generations of data centers that can deploy applications that fully suit the users' QoS requirements. Such applications require a wide range of specifications, including deployment, structure, and configuration. Problems that challenge cloud developers include quantifying resource-allocation policy performance and developing algorithms for scheduling, power consumption, heat dissipation, and system sizing.

CloudSim is an extensible simulation toolkit that enables the modeling and simulation of cloud-computing environments. In this work, CloudSim was used to model virtual machines, tasks, task-to-VM mapping, and the policies associated with allocation and scheduling [13,14]. It can simulate multiple data centers and also supports VM migration and the study of allocation policies.

Cloud computing delivers infrastructure, platform, and software services as subscription-based services, commonly represented as IaaS, PaaS, and SaaS. The cloud aims to power the next generation of data centers, providing network-accessible services that can be deployed on demand with a competitive cost that depends primarily on QoS requirements. This enables providers and customers to test deployments and evaluate strategies before deployment in real environments.

Simulation tools are especially useful because real infrastructures are costly and difficult to reproduce under identical conditions. CloudSim therefore offers a controlled environment in which developed services can be tested repeatedly and in which different allocation strategies can be compared under identical configurations.

2.2 CloudSim Architecture

CloudSim software components are shown in Fig. 2. SimJava forms the event-simulation core, GridSim provides grid-level abstractions, and CloudSim adds cloud-specific entities such as hosts, VMs, cloudlets, data centers, and brokers. This layered design enables researchers to focus on scheduling and resource-provisioning policies while avoiding low-level implementation overhead [13–15].

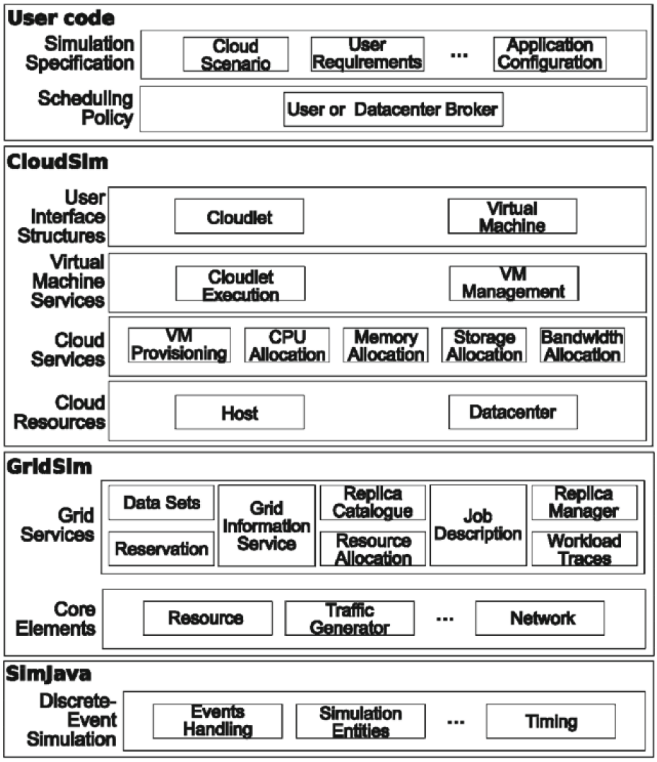

Fig. 2. Layered CloudSim architecture.

2.3 VM and Jobs Allocation

In cloud computing, tasks are submitted by many users and executed via heterogeneous VMs. Each task is mapped to a VM that suits the request, while VMs are hosted on physical machines. Such processes require effective resource allocation for bandwidth, memory, CPU, and storage. The problem of VM allocation and placement is NP-hard [16].

Based on the system model above, this work proposes an approach for energy efficiency in virtualized cloud data centers. The approach presents a classification model for jobs and a VM-placement strategy known as mixed-type placement, intending to reduce energy waste in the data center [17,18].

The proposed model assumes a fully virtualized cloud-computing environment where jobs are submitted together with fixed QoS requirements. Jobs are initially handled by a global scheduler and then forwarded to a specific data center for later VM allocation and PM placement. The workload is modeled as a bag of tasks (BoT), where tasks are autonomous and do not communicate with one another [18].

2.4 Strategic Allocation and Placement of Tasks

The proposed CPUBA approach follows the sequence below:

1. The user submits the job to the cloud host.
2. A preprocessing step identifies the task class according to its CPU requirement.
3. The task is stored with other tasks of similar CPU requirements.
4. The task CPU-integration process combines multiple jobs and forwards them to a suitable VM.

There are two main policies in cloud-computing environments for allocating tasks to VMs: space-shared and time-shared [15]. This work uses the space-shared policy. Available VM configurations are sorted in ascending order according to the number of cores, and each job is matched sequentially with the lowest suitable VM configuration.

The CPU architecture supports various frequencies, and each job is allocated to a VM with a specific frequency level. This makes it possible to scale CPU-core frequencies using DVFS. When a VM is allocated to a low-CPU job, the frequency can be minimized provided the QoS requirement is maintained. This contributes to lower power use.

The four possible combinations considered during the proposed VM allocation are:

1- Lowest low-CPU jobs with lowest mid-CPU jobs and highest high-CPU jobs.
2- Lowest low-CPU jobs with highest mid-CPU jobs and highest high-CPU jobs.
3- Highest low-CPU jobs with highest mid-CPU jobs and lowest high-CPU jobs.
4- Highest low-CPU jobs with lowest mid-CPU jobs and lowest high-CPU jobs.

Applying CPUBA enhances energy efficiency in two ways: it reduces the number of active PMs and improves the utilization of those PMs. The overall job life cycle is illustrated in Fig. 3.

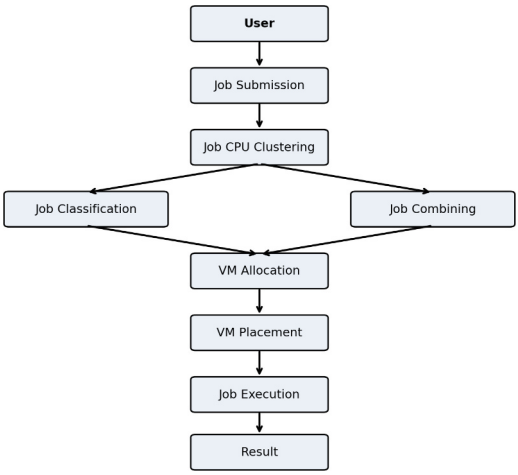

Fig. 3. User's job life cycle in the CPUBA approach.

3 Cloud Environments and Experiment Results

After obtaining the clustering results, the work moved to the next stage, which represents the implementation of the data-clustering outputs over a cloud-computing environment. The experimentation used CloudSim [14]. Two phases were considered: the first examined task allocation using default available methods; the second examined the proposed CPUBA allocation over the same environment, using the same data and variables (Tables 1 and 2).

Table 1. Features and variables of VMs/hosts in the experimental cloud environment.

Category	Feature/Variable	Value
VM	Number of Virtual Machines (VMs)	20
VM	Number of processing elements (PEs)	20
VM	RAM Size (MB)	512
VM	Storage Size (MB)	10000
VM	Bandwidth (MB)	1000

Table 2. Features and variables of cloudlets in the experimental environment.

Category	Feature/Variable	Value
Cloudlet	Number of Cloudlets	100
Cloudlet	Number of processing elements (PEs)	16
Cloudlet	Cloudlet length	Calculated thresholds
Cloudlet	File Size	1024
Cloudlet	Output Size	1024

3.1 Task Distribution over the Cloud Data Center

Experimentation was performed on ten hosts using the default CloudSim allocation policies and the proposed CPUBA approach. One hundred cloudlets were executed in both settings. Table 3 shows the distribution produced by the default space-shared/round-robin allocation, while Table 4 shows the result produced by CPUBA over the same cloud environment and the same cloudlet-length range (Figs. 4 and 5).

Table 3. Cloudlet distribution (CPU length/time) under default allocation.

Host	No. Tasks	Task CPU Length	Time
0	10	916176	922
1	10	852057	857
2	10	829579	835
3	10	905116	912
4	10	1046482	1052
5	10	950866	955
6	10	1063518	1071
7	10	842172	848
8	10	739412	745
9	10	951568	957
Total	100	9096946	9154

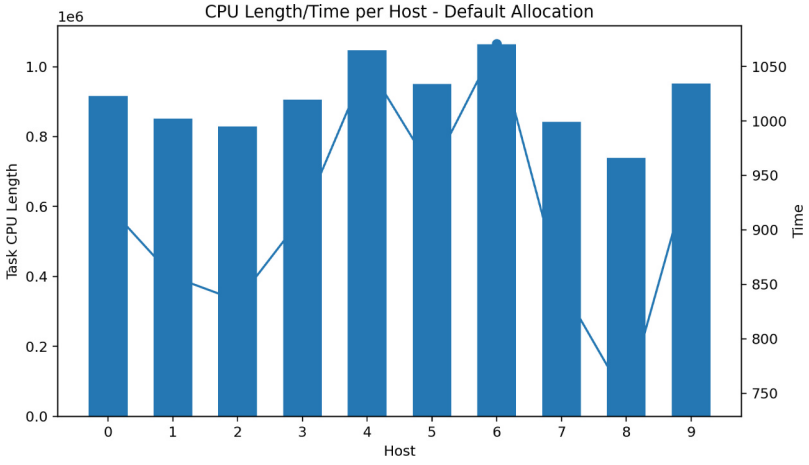

Fig. 4. CPU length/time per host for the default allocation.

Table 4. Cloudlet distribution (CPU length/time) under CPUBA.

Host	No. Tasks	Task CPU Length	Time
0	10	1040912	1047
1	10	806787	813
2	10	1056381	1061
3	10	811029	817
4	10	990039	994
5	10	697184	703
6	10	1017719	1024
7	10	788752	794
8	10	1079281	1086
9	10	808862	815
Total	100	9096946	9154

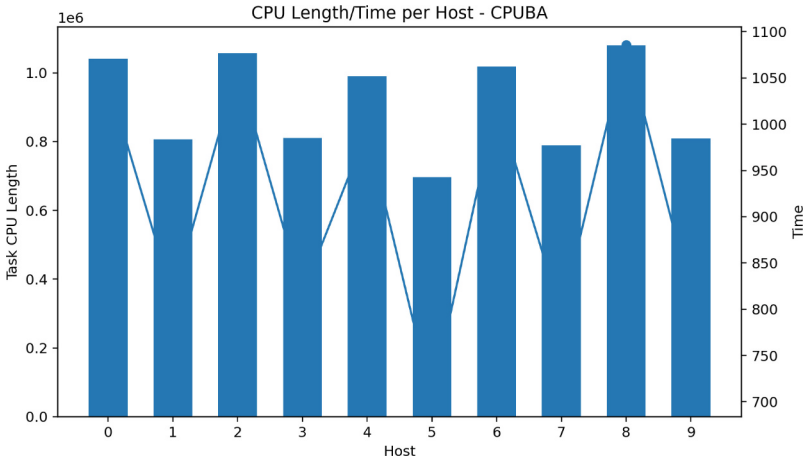

Fig. 5. CPU length/time per host for CPUBA.

3.2 Data Center Power Consumption Comparison

This section illustrates the power-consumption improvement obtained by using the proposed CPUBA approach. Both policies were applied to the same data center, using the same CPU cloudlet lengths and identical environment characteristics (Table 5).

Table 5. Datacenter total power consumption: default allocation vs. CPUBA.

Default allocation			CPUBA		
Host	Watt-Sec	KWh	Host	Watt-Sec	KWh
0	27375	0.00760	0	23026	0.00640
1	27312	0.00759	1	22308	0.00620
2	27250	0.00757	2	23075	0.00641
3	27472	0.00763	3	22321	0.00620
4	27903	0.00775	4	22881	0.00636
5	27617	0.00767	5	22010	0.00611
6	27811	0.00773	6	22962	0.00638
7	27297	0.00758	7	22249	0.00618
8	26832	0.00745	8	23141	0.00643
9	27600	0.00767	9	22305	0.00620
Total	274469	0.07624	Total	226278	0.06287

Figures 6, 7, 8 and 9 illustrate the power-consumption profiles of the datacenter hosts for the default policy and CPUBA.

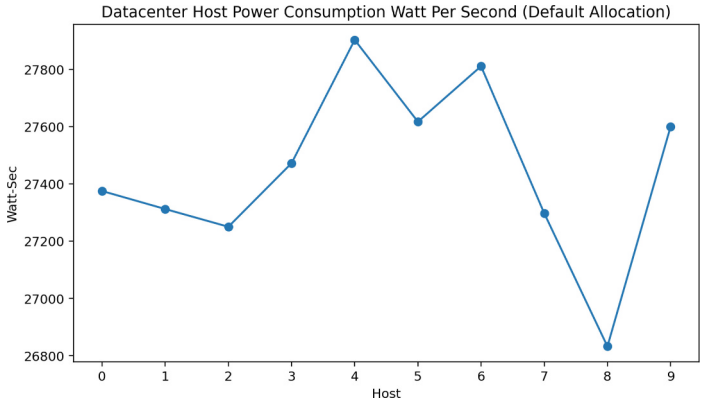

Fig. 6. Datacenter host power consumption in watt-seconds under default allocation.

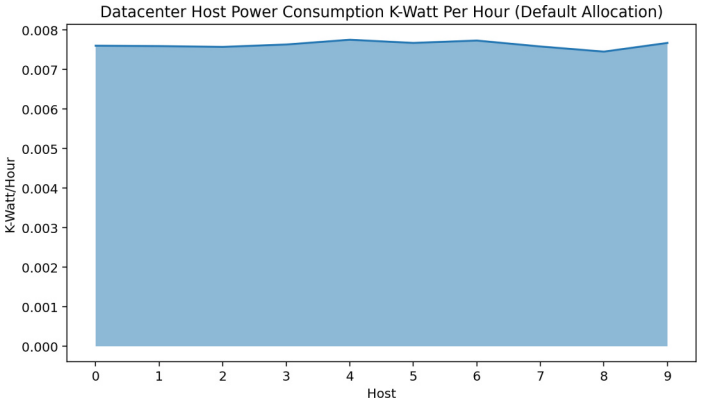

Fig. 7. Datacenter host power consumption in kilowatt-hours under default allocation.

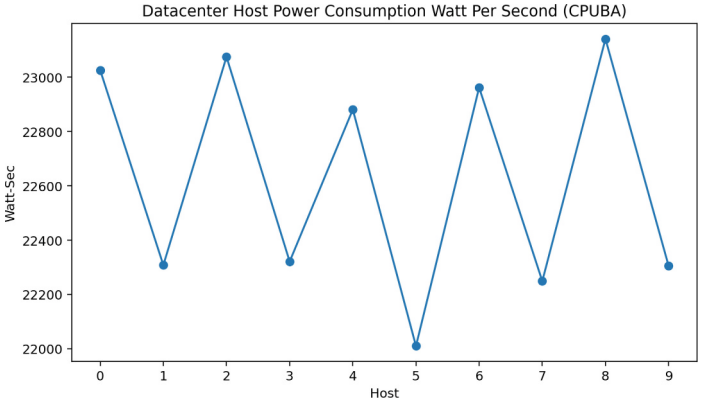

Fig. 8. Datacenter host power consumption in watt-seconds under CPUBA.

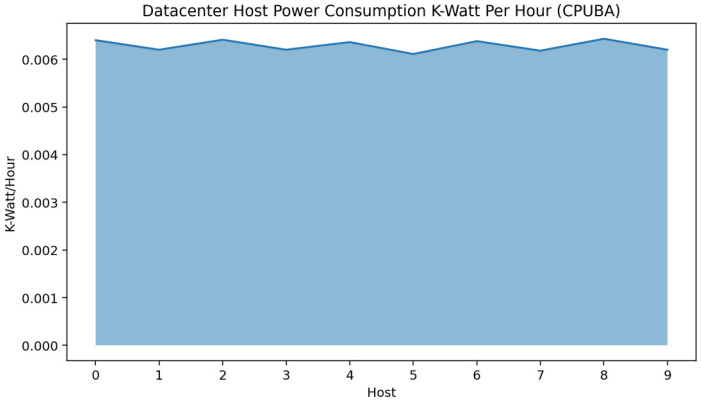

Fig. 9. Datacenter host power consumption in kilowatt-hours under CPUBA.

The main intention of this work is to develop an allocation approach for more efficient task allocation over cloud data-center hosts, PMs, and VMs. Power consumption improved in the same data center with identical characteristics and tasks. The improvement total and enhancement percentage are summarized in Table 6.

Table 6. Datacenter power-consumption improvement.

Result	Total Watts	Total K-Watts
CPUBA Result	226278	0.062
Original Result	274469	0.076
Improvement Total	48191	0.010
Improvement %	17.5	17.5

4 Conclusion

The analysis of Google trace data in this work targeted a large number of task-usage records representing the requirements of tasks operating in the Google cloud during the recording period. The analysis of 50 million task records produced CPU thresholds that were later used for cloud data-center allocation.

The study concludes that task allocation and distribution over suitable machine clusters has a significant impact on performance enhancement, especially when the number of clusters is substantial and the clustering stage is based on large-scale real workload traces. Based on the results, cloud machine-resource availability and workload behavior can be predicted more clearly. This supports the orientation of new tasks toward the most suitable machine group, which reduces scheduling time, waiting time, and unnecessary resource waste.

Cloud-environment performance enhancement is difficult to achieve by focusing on only one aspect, due to the variety of cloud challenges and performance factors. In this work, both the power factor and scheduling time were enhanced. The experimentation reveals the following points:

- Real cloud-data analysis provides a clearer presentation of cloud work patterns and performance factors.
- Task allocation and distribution over cloud data centers has a positive effect on cloud-environment performance.
- The requested CPU rate of each task can play a significant role in the task-allocation process.
- Pre-sorting tasks before VM allocation has a clear positive effect on task allocation, scheduling time, and processor power loss.

5 Future Work

Future work can analyze larger data volumes to find more workload groups and more accurate CPU thresholds by using big-data processing environments such as Hadoop ecosystems. Another promising direction is to extend the task-distribution model by considering additional task requirements such as RAM, in addition to CPU usage, in order to obtain a clearer representation of the resources needed by each task and later improve allocation efficiency in CloudSim or a real cloud environment.

Acknowledgements. This research was sponsored by the University of Malaya Translational Research Grant Scheme, TRG0019-2025 (UM.0002388/HRU.IP) and the Research Excellence Grant, UMREG 068 - 2024 (UM.0000285/HRU.RC).

Conflicts of Interest. The authors declare no conflict of interest.

References

1. Panwar, S.S., Rauthan, M.M.S., Barthwal, V.: A systematic review on effective energy utilization management strategies in cloud data centers. J. Cloud Comput. **11**, 95 (2022)
2. Wang, Z., et al.: Reinforcement learning based task scheduling for environmentally sustainable federated cloud computing. J. Cloud Comput. **12**(1), 174 (2023)
3. Azkeskin, S.A., Aladağ, Z.: Evaluating regional sustainable energy potential through hierarchical clustering and machine learning. Environ. Res. Commun. **7**(1), 015002 (2025)
4. Abdulhameed, T.Z., Yousif, S.A., Samawi, V.W., Al-Shaikhli, H.I.: SS-DBSCAN: semi-supervised density-based spatial clustering of applications with noise for meaningful clustering in diverse density data. IEEE Access **12**, 131507–131520 (2024)
5. Widjaja, A.E., Toer, G.A., et al.: Clustering digital governance adoption patterns in the metaverse using K-Means and DBSCAN algorithms. Int. J. Res. Metaverse **3**(1), 1–13 (2026)
6. Yousif, S.A., Al-Dulaimy, A.: Clustering cloud workload traces to improve the performance of cloud data centers. In: Proceedings of the World Congress on Engineering, vol. 1, pp. 7–10 (2017)
7. Maala, H.H., Yousif, S.A.: Cluster trace analysis for performance enhancement in cloud computing environments. J. Theoret. Appl. Inf. Technol. **97**(7) (2019)
8. Sagheer, N.S., Yousif, S.A.: A parallel clustering analysis based on Hadoop multi-node and Apache Mahout. Iraqi J. Sci. **62**(7) (2021)
9. Pulse Global: Big Data for development: challenges and opportunities (2012). Retrieved 12 Dec 2012
10. Feng, X., Siarry, P., Han, L., Yang, L.: Cloud Computing. Springer, Switzerland (2025)
11. Nandagopal, M., et al.: Enhancing energy efficiency in cloud computing through task scheduling with hybrid cuckoo search and transformer models. Discov. Comput. **28**(1), 199 (2025)
12. UN Global Pulse: Big data for development: Challenges & opportunities (2012)

13. Buyya, R., Ranjan, R., Calheiros, R.N.: Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities. In: 2009 International Conference on High Performance Computing & Simulation, pp. 1–11. IEEE (2009)
14. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Exp. **41**(1), 23–50 (2011)
15. Wickremasinghe, B., Calheiros, R.N., Buyya, R.: CloudAnalyst: a CloudSim-based visual modeller for analysing cloud computing environments and applications. In: 2010 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 446–452. IEEE (2010)
16. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. **1**(1), 7–18 (2010). https://doi.org/10.1007/s13174-010-0007-6
17. Aldulaimy, A., Zantout, R., Zekri, A., Itani, W.: Job classification in cloud computing: the classification effects on energy efficiency. In: 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), pp. 547–552. IEEE (2015)
18. Thai, L., Varghese, B., Barker, A.: A survey and taxonomy of resource optimisation for executing bag-of-task applications on public clouds. Futur. Gener. Comput. Syst. **82**, 1–11 (2018)