energies

MDPI

Article

Applications of Virtual Machine Using Multi-Objective Optimization Scheduling Algorithm for Improving CPU Utilization and Energy Efficiency in Cloud Computing

Rajkumar Choudhary and Suresh Perinpanayagam *

Integrated Vehicle Health Management Centre, Cranfield University, Cranfield MK43 0AL, UK
* Correspondence: suresh.nayagam@cranfield.ac.uk

Abstract: Financial costs and energy savings are considered to be more critical on average for computationally intensive workflows, as such workflows which generally require extended execution times, and thus, require efficient energy consumption and entail a high financial cost. Through the effective utilization of scheduled gaps, the total execution time in a workflow can be decreased by placing uncompleted tasks in the gaps through approximate computations. In the current research, a novel approach based on multi-objective optimization is utilized with CloudSim as the underlying simulator in order to evaluate the VM (virtual machine) allocation performance. In this study, we determine the energy consumption, CPU utilization, and number of executed instructions in each scheduling interval for complex VM scheduling solutions to improve the energy efficiency and reduce the execution time. Finally, based on the simulation results and analyses, all of the tested parameters are simulated and evaluated with a proper validation in CloudSim. Based on the results, multi-objective PSO (particle swarm optimization) optimization can achieve better and more efficient effects for different parameters than multi-objective GA (genetic algorithm) optimization can.

Keywords: CloudSim; multi optimization technique; virtual machine; host machine; genetic algorithm; particle swarm optimization; cloud computing

check for updates

Citation: Choudhary, R.; Perinpanayagam, S. Applications of Virtual Machine Using Multi-Objective Optimization Scheduling Algorithm for Improving CPU Utilization and Energy Efficiency in Cloud Computing. *Energies* **2022**, *15*, 9164. https://doi.org/10.3390/en15239164

Academic Editors: Marcin Sosnowski, Jaroslaw Krzywanski, Karolina Grabowska, Dorian Skrobek, Ghulam Moeen Uddin, Yunfei Gao, Anna Zylka, Anna Kulakowska and Bachil El Fil

Received: 24 October 2022
Accepted: 30 November 2022
Published: 2 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The cloud computing (CC) concept has been utilized in various fields to enable the network access to various computing resources, including services, applications, storage, servers, and networks. CC can be divided into three sub-categories based on the architecture structure: IaaS (infrastructure as a service), PaaS (platform as a service), and SaaS (software as a service). IaaS is considered to be fundamental for the service models developed using this concept [1]. Recently, most of the studies have focused on the development of multi-objective meta-heuristic algorithms such as ACO (ant colony optimization), GA, SA (simulated annealing), and PSO to resolve the workflow scheduling issues and obtain efficient responses. Reducing the makespan and the execution costs are normally considered to be conflicting targets in energy consumption, and most of the existing research using the EMO (evolutionary multi-objective optimization) algorithm [2] has focused on developing Pareto-optimal solutions, especially in CC workflow applications.

Moreover, CC applications are normally implemented in a VC (virtual cluster) environment that executes multiple VMs possessing PM (physical machine) resources on demand. Furthermore, the energy consumption, especially in data centers, increases cumulatively along with the utilization of the deployed applications. In this study, we define the total electricity consumption utilized by the data centers as 8% of the entire electrical consumption that is utilized in the world. However, experimenting with CC applications related to energy efficiency is also significant for data centers. Moreover, VC allocation, defines the allotment of every virtual machine of the virtual cluster to the proper PM. Adequately assigning VMs to PMs will facilitate energy consumption minimization and enable the PMs to be grouped together. Nevertheless, this kind of evaluation will increase the availability of the VC [3].

Even though various methods have been developed to save the energy and costs, enhance the resource usage, and guarantee QoS (quality of service) in CC, the studies have infrequently examined the rapid and optimal allocation of cloud resources to meet the rising demands. Different methods such as the task relaxation degree have been developed based on task execution time and deadline time, and the relaxation degree of the task should be less than the allocated threshold value based on the emergent task. Hence, these emergent tasks are processed based on existing or new VMs using randomness-aware scheduling techniques. However, these techniques have only been adapted to priority levels set based on the task, and they do not include diverse and heterogeneous emergent resource demands [4].

Metaheuristic-based techniques are utilized to develop effective scheduling solutions, especially for CC tasks that are executed independently under certain deadline constraints. The existing methods integrate PSO and CRO (Chemical Reaction Optimization) to create a CR–PSO method [5] that is able to enhance the PSO optimization standards and increase the QoS parameter performance. These existing methods and mathematical models help to resolve the multi-objective optimization issues such as energy consumption, average execution time, makespan time, and computational costs.

It remains necessary to enhance the adaptive management capabilities of the VM placement in CC with multi-objective optimization and an adaptive management technique for VM placement in CC in accordance with the PSO. The development of the PSO global dynamic objective function with the VM global optimal solution in CC is initiated via the deconvolution method, and the PSO optimum position is checked in a 2D space. Issues related to the multi-objective optimization of the adaptive management in the VM placement are now considered to be PSO issues that must be resolved to realize VM placement, multi-objective optimization, and adaptive management in CC [6].

Recently, the utilization of data centers has consistently increased, yet limited resources have significantly hindered the enhancement of the service quality. The energy consumption has also rapidly increased in the CDCs (cloud data centers), creating certain limitations for cloud providers. Additionally, an average of 30% of the servers in the operating environment are considered to be idle VMs. For this reason, enormous schemes for resource management have been developed at CDCs. Normally, these problems raised in the CC environment can be resolved by deploying a VM. Nevertheless, the VM's service scale must be broadened due to the VM's booting overheads. When it comes to data centers, migration technologies enable effective resource scaling. These methods can be migrated effectively and rapidly to enable dynamic load balancing, automatic failover, and zero downtime [7].

The requirement for improved energy efficiency and performance in real-time applications running on virtual machines in cloud data centers is the impetus behind this line of study. The objective is to design deterministic algorithms and adaptive heuristics that are capable of dynamic energy-aware scheduling with the end aim of maximizing both the energy efficiency and performance effectiveness. The development of dynamic energy-aware scheduling for deterministic algorithms is, initially, the primary focus of the project. The goal is to reduce the amount of energy that is used while maintaining the level of performance that is necessary for the real-time activity.

2. Literature Survey

In [8], a Pareto-based MOVMrB (Multi-Objective VM reBalance) solution was utilized to simultaneously reduce the disequilibrium of the intra-HM (host machine) and inter-HM loads. A summary of the existing methods for load rebalancing was also provided. This method was considered to be one of the most effective solutions for enhancing the intra-HM and inter-HM loads by applying an MOO (multi-objective strategy) to solve the VMrB (VM rebalance) issues. The load balance was also used as a main component in CC to enhance the distributed system performance by assigning the workload in a cooperating host set. This method was found to minimize the VMrB I/O complexity. Nevertheless, when it comes to real-world scenarios, the HM balance is disrupted because of the removal of the VMs. Hence, it is vital to allocate the VMs as the VMrB.

In [9], an enhanced ch-PICEA-g multi-objective co-evolutionary algorithm was used to resolve the problem of complexity and enhance repetitions to achieve a global optimum. This heuristic algorithm is considered to be effective, wherein the tent maps and logistics are considered to be chaotic systems, and they are employed with originating values to solve the permutation convergence in the early stages of the population. Further, in [9], an enhanced fitness function was employed to enhance the original PICEA-g performance. The functionality of the technique was evaluated using an extensive experimentation process, and the hybrid technique achieved better results based on various performance metrics.

In [10], a hybrid method with an enhanced MPSO (multi-objective particle swarm optimization) known as FIKPSO was utilized to solve the load-balancing issues and facilitate the prediction of enhanced responses. The computing properties and workloads were split in a CC environment that allowed the enterprise to handle the workload demands. Further, this method employed the FF (FireFly) algorithm to reduce the search space alongside the IMPSO method to predict the enhanced response. Normally, the IMPSO technique picks the GBest (Global Best) particle with a small point distance that denotes a line. This method obtained an average load to improve the essential evaluation factors such as the task response time and the resource usage. The method achieved good results with a makespan of 148, a throughput of 72%, a reliability of 675, a memory utilization of 93%, a response time of 13.58 ms, and a CPU usage of 98%.

In [11], a hybrid algorithm integrating optimization methods including PSO and CS (Cuckoo Search) was utilized to resolve the CC scheduling problems. This study focused on various aspects, and it minimized the deadline violation rate and makespan cost. The CloudSim toolkit was utilized to evaluate the method's performance. This technique obtained better simulation results and reduced the deadline violation rate and makespan cost compared to other methods such as FCFS, PBACO, MIN-MIN, and ACO.

In [12], a novel method based on the MBO-VM Monarch Butterfly Optimization algorithm was utilized for a new VM placement. This method was developed to increase the packaging efficiency and minimize the number of active physical servers. Nevertheless, the data center resources were not utilized properly, which impacted the energy efficiency; this factor should be more strongly considered in the VM placement strategy. Moreover, the CloudSim toolkit was used to validate the MBO-VM technique's efficiency in real-world cloud and synthetic workloads. The simulation results demonstrated that MBO-VM can achieve efficient results with known proper VM placement methods. Ultimately, this method minimized the active server count and leveraged the package efficiency.

In [13], multi-objective-based integrated task scheduling algorithm in a cloud environment was proposed to solve NP-hard issues. Obtaining optimal solutions during the task-scheduling process is generally complex when one is faced with multiple conflicting ideas. This study focused on resolving the issues that persist for task scheduling in CC. Another goal was to provide an optimal solution that concentrates on various terms such as load balancing, execution cost, and time.

In [14], the various energy-saving issues that occur in data centers were analyzed utilizing VM placements. Further, live migration was used to evaluate the network link loads. The issues were articulated as multi-objective integer linear programs, and they were resolved using CPLEX to reduce the server energy consumption and VM time migration. The heuristic method was utilized to resolve the issues raised regarding CPLEX's high computational costs and to reduce the network migration. Heuristics were also estimated based on the energy consumption and performance evaluation utilizing a real data center testbed with Hadoop HiBench benchmarks. Ultimately, this method was found to save up to 30% of the energy consumption in the servers.

In [15], a multi-objective PSO algorithm was utilized to minimize the energy consumption in the VMs and minimize the loss of the linked resources. During the request of a single user, the VMs are focused on energy consumption minimization, eventually leading to a decrease in their request reliability. The research in [15] concentrated only on the single-point failures that increase the request reliability. The simulation results showed that this concept can reliably fulfill the tenant request and efficiently manage the link loss, thereby minimizing energy consumption in the data center.

The necessity to use correct power models is the most typical obstacle that people face. It is difficult to know how to arrange the activities in order to reduce the amount of power that is used when the correct power models are not available. The creation of novel heuristics presents another obstacle that is to be overcome. At the moment, just a few heuristics have been presented for consideration [16]. Additional effort needs to be put into the creation of new heuristics as well as the optimization of those that already exist. In spite of these obstacles, dynamic energy-conscious scheduling is a potential method for cutting down on the amount of power that is used by real-time systems.

After the power models have been built, they may be used to plan jobs in order to make the most efficient use of the available power. The overall power consumption of the system need to be reduced as much as possible, and the job schedule ought to reflect that objective [17]. Real-time systems might benefit from a technology called dynamic energy-conscious scheduling, which has the potential to save power usage. Despite this, there are still a great number of obstacles that need to be overcome before its use can become more widespread.

In [18], a new MOGA (multi-objective genetic algorithm) was utilized to schedule the workflow in a CC environment. This method considers the conflicting focus of stakeholders in the cloud to achieve optimization, and it provides a result for makespan minimization in addition to solving deadline constraints. Further, this method was found to provide energy-efficient results utilizing dynamic voltage frequency scaling.

3. Research Methodology

In the present research, a novel approach based on multi-objective optimization is proposed with CloudSim as the underlying simulator to evaluate the VM allocation performance. Here, we computed the energy consumption, CPU utilization, and the number of executed instructions in each of the scheduling intervals for complex VM scheduling solutions to increase their energy efficiency and reduce their execution time. To achieve high-performance cloud computing, energy-aware scheduling has become an efficient approach. The energy-aware approach is usually performed by scheduling tasks with the aim of reducing the execution time and the consumption of energy [19]. The quality of the service in the cloud can be considered to be a multi-objective optimization problem where various objectives must be considered, including the computing power, memory utilization, and bandwidth. Considering these objectives, we propose an optimal approach to achieve an efficient energy-aware task-scheduling concept.

A VM (virtual machine) is a software-based computer that is normally executed using physical resources [20]. The operating systems and applications can be developed and executed on a VM with a connection to a physical computer. Therefore, optimization simply involves identifying an effective solution from a set of solutions. Multi-objective optimization is a novel technique used to analyze and optimize several conflicting objectives of an issue, and it could be utilized for this purpose. The current research focuses on various problems such as the VM host placement based on the RAM and CPU requirements. The VMP (virtual machine placement) issues are considered from the requirements for VM decision making to host allocation. The main focus of this study is to evaluate the performance and reduce the energy consumption and execution time using the Multi-Objective Particle Swarm Optimization (MO-PSO) method. Then, the results are compared with other methods to evaluate the performance.

The current research was executed utilizing the CloudSim simulation environment (http://www.cloudbus.org/cloudsim/; accessed on 11 November 2022) with the initial CloudSim configuration represented in Table 1. During the initialization of the CloudSim configuration, various parameters related to the data center configuration were used, including the number of data centers (2), architecture (×86), Operating System (OS) (Linux), time zone (10.0), cost process (3.0), memory cost (0.05), storage cost (0.001), and bandwidth cost (0.1). Next, during the host configuration, the parameters included the number of hosts (1), storage (100,000 MB), Host_mips (1000), Host RAM (65,535 MB), and host bandwidth (100,000 Gbps). For the VM configuration, the experimental parameters included the number of VMs (10), VM_Image_Size (10,000 MB), VM_RAM (2048 MB), VM_MIPS (250), VM_BANDWIDTH (1000 Gbps), VM_PES (1), and VMM_NAME (Xen).

Table 1. Initial CloudSim configuration.

Data Center Configuration	
Name of Attribute	**Value**
Number of data centers	2
Architecture	×86
OS	Linux
Time-Zone	10.0
Cost—process	3.0
Cost—Memory	0.05
Cost—Storage	0.001
Cost—Bandwidth	0.1
Host Configuration	
Number of Hosts	1
Storage	100,000 MB
Host_mips	1000
Host RAM	65,536 MB
Host Bandwidth	100,000 Gbps
VM Configuration	
No of VMs	10
VM Image Size	10,000 MB
VM_RAM	2048 MB
VM_MIPS	250
VM_Bandwidth	1000 Gbps
VM_PES	1
VMM_NAME	Xen

CloudSim is a toolkit with which users can test a technique's performance before utilizing it in real systems. Normally, CloudSim models and simulates CC environments where the researchers can design and develop a VM for a data center that has different hosts. CloudSim can enable the simulation of VM scheduling in two ways: VM-level simulation or host-level simulation. The hosts in the existing research are distributed in the VMs based on the power of the cores, which is distributed according to processing of the tasks. The simulation environment is executed using different selection methods for choosing the utilized PMs but not for the execution of VMs. MO-PSO used the following software parameters for the experimentation process:

- Operating System: Windows 10 (64 bit);
- IDE Tool: Netbeans 12.5;
- Install netbeans version 12.5;
- Add required cloudsim-package for implementation;
- Deploy the java projects;
- Execute Java.

In the CloudSim simulation environment, virtual machine migration is a vital process that is executed using optimized allocation with the package taken from CloudSim. In the present research, the VM migrations were developed and analyzed based on the following steps:

Step 1—Detection of over-used PMs;
Step 2—VMs are chosen to migrate, especially for the machines in the previous step;
Step 3—Chosen VMs are assigned to the new PMs;
Step 4—Detection of underutilized PMs;
Step 5—VMs are chosen to migrate from the underutilized machines;
Step 6—The chosen VMs are assigned to the new PMs.

3.1. Multi-Optimization Techniques in a Cloud Environment

Firstly, the particular task is carried out with the proper analysis and finalized collected resources. Then, the CC service provider is allocated based on the requests or full tasks. Here, multiple parameters are considered with the objective and multi-optimization-based techniques that have been deployed to solve MOOP (multi-objective optimization problems) with the objective of minimizing both the makespan and energy under the constraints such as physical machine selection for a task with the task unit's maximum processing time, maximum communication time, and energy consumption for communication, load distribution, etc. Particle Swarm Optimization is also used to aid in formulating a solution with the best local and global search capabilities and the fastest convergence. The parameters in our simulation (i.e., the number of PMs, number of task units, number of tasks per task unit, size of task, bandwidth, PM ratings, energy consumption, etc.) are similar to those used in a real-time cloud environment, and they remained fixed. Moreover, several simulation runs of the energy-efficient task-scheduling process were executed. Once the simulation was complete, the energy consumption and makespan for different tasks units were recorded with different PMs under different types of machine heterogeneity. Finally, the proposed technique was compared based on various estimated performance attributes, such as the converge ratio, the distance-based distribution, the maximum spread, the hyper area, and the hyper volume ratio to verify the efficiency of the optimization.

To evaluate the performance of the proposed algorithm, the Genetic Algorithm concept was used in the current research. During this process, the fitness value of every chromosome is specified in the population. The process is repeated until the new population reproduces. During processing, it is necessary to choose two individual chromosomes taken from a population based on the roulette wheel selection method. Then, the chosen parents cross over under a crossover probability to develop a new child. The defined children will be copies of the parents. The mutations of every new offspring based on the mutation probability at every position of the chromosome are noted in the process. Finally, the new child is placed into a reproduced population, and the best individual outcome is stored.

3.2. Proposed Multi-Optimization PSO Algorithm

In this study, the Multi-Optimization PSO algorithm was used based on certain parameters. Normally, PSO includes a swarm of particles that defines a solution for given issues. In this study, the particles were defined as a set of VMs assigned to a specific task. Each particle represents the swarm behavior with various characteristics: the x position defines the location that is suggested, and the velocity denoted as v indicates the speed, while pbest and gbest are used to determine the solution in the population. Every particle position at any time is denoted by pbest, while the other positions in the global problem space are denoted as gbest. The workflow of the proposed technique is depicted in Figure 1.

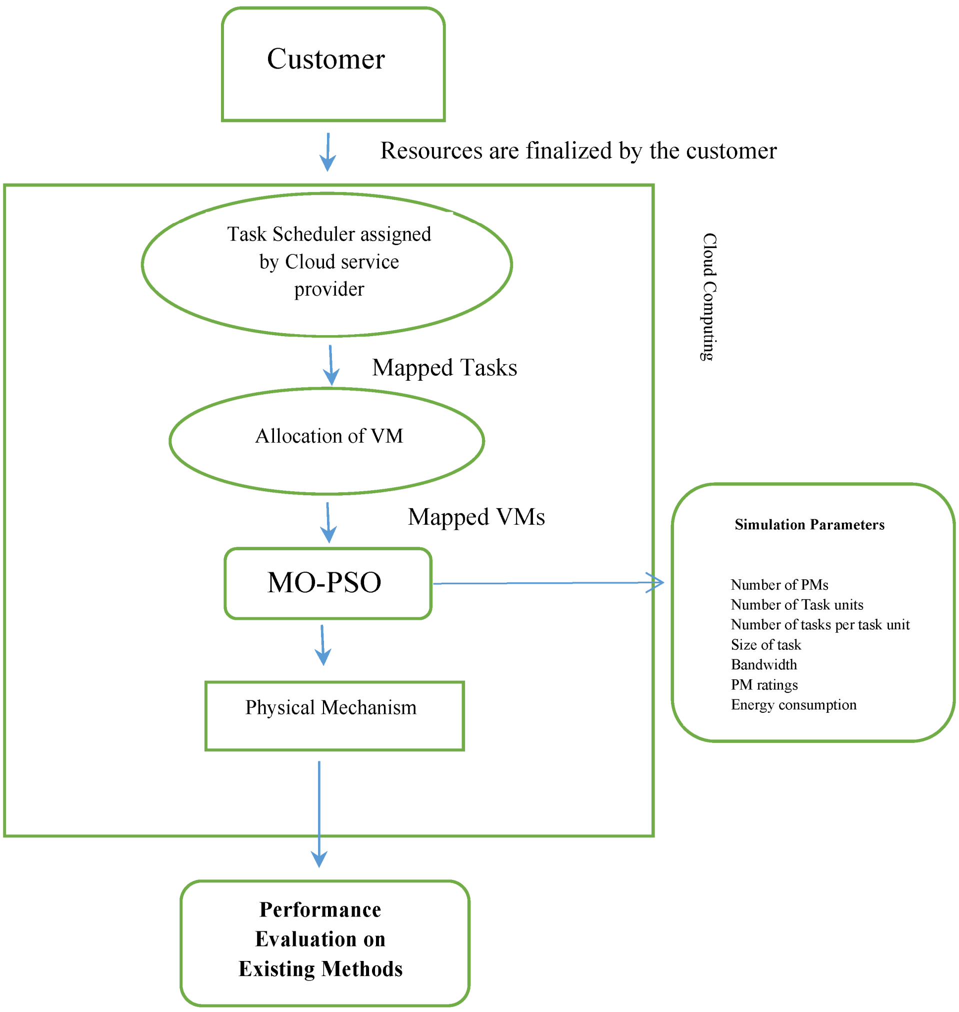

Figure 1. Workflow of multi optimization PSO algorithm.

The PSO utilizes adaptive movement that transfers a particle position in every iteration. The mathematical formulation used in this research is as follows:

$$xj(t) = xj\ (t1-1) + vj\ (t1) \tag{1}$$

where $xj(t1)$ indicates the particle's current position j at every iteration $t1$, $xj(t1-1)$ represents the particle's current position i at every $t1-i$ iteration, and $vj(t1)$ represents the particle velocity i at the $t1$ iteration.

The velocity of every particle (j) at a certain time ($t1$) is denoted as follows:

$$vj(t1) = w1 \times vj(t1-1) + r1 \times C1 \times (pbestt - xj) + r2 \times C2 \times (gbestt - xj\ (t1)) \tag{2}$$

where $xj(t1)$ is the particle j current position at every iteration $t1$, $vj(t1)$ is the particle velocity (j) at the $t1$ iteration, $vj(t1-1)$ is the particle velocity (j) at every iteration $t1-1$, pbestt is the best position for particle j, gbestt is the best value position for every particle (j) presented as a $w1$ inertial weight in the population, $r1$ and $r2$ are random numbers that range between 0 and 1, and $C1$ and $C2$ are the acceleration coefficients.

As shown in Algorithm 1, we can use a set number of iterations of the above code to run dynamically and see how the particles explore, and we can stop it when we cannot see any update to the best solution in a number of iterations. In particular, it starts by initializing the swarm, position, velocity, and pbest values. From lines 3 to 9 (main loop), it iterates through all of the particles to discover the best solutions. The COMPUTEFITNESS function is used to compute the fitness function for each particle based on the particular objective. The EVALUATE function is invoked to compare the personal best for each particle with the fitness value and return the best value (pbest). The UPDATE function is used to update the position and velocity in each iteration. In the end, for each particle, the pbest with the new best value is returned.

Algorithm 1. PSO Algorithm

```
procedure_P PSO ()
INITIALIZE(Swarm (s), Velocity (v), Position (p), Pbestt)
While (Stop S criteria, if it is not satisfied) do
For p ε P do // particles iteration
F = COMPUTERFITNESS (Swarm (p)); // Fitness computation
Pbestt = EVALUATE (F)
UPDATE (v, p) // here use the equations 1, 2
End—for
End—while
Return—Pbestt
End procedure
```

4. Experimental Results and Analysis

The analysis of the simulation results and performance evaluation of the proposed novel technique are elaborated below.

4.1. Energy-Aware CloudSim with Optimization

In the experimentation stage, it was first necessary to install the tools for the execution process, such as the operating system (Windows 10 (64 bit)), the IDE Tool (Netbeans 12.5), the CloudSim package, and Java (for deployment and execution). In the current research, the simulation process was conducted using a novel multi-objective approach with the integration of Particle Swarm Optimization, and a scheduling algorithm with the interval was used. We used CloudSim as the underlying simulator to evaluate the performance of the VM allocation.

The process was completed based on CPU utilization, energy consumption, and the number of executed instructions in each scheduling interval for difficult VM scheduling solutions. The simulation process uses a minimal amount of energy and a minimal execution time. Figure 2 presents the CloudSim configuration with the various attributes. These attributes and values help to configure and run the simulation process. The configuration settings were then processed with the following settings: the Scheduling Interval (10), host number (3), host start_up delay (5), host shut_down delay (3), host start_up power (5), host shut_down power (3), number of VMs (4), VM PES (4), no. of cloudlets (8), Cloudlet_PES (2), Cloudlet_Length (50,000), host PES (8), energy consumption static (35), and energy consumption max (50).

CloudSim then indicates the initial configuration deployment including the aforementioned attributes necessary to execute the simulation process and thereby determine the efficiency of the proposed method.

4.2. Energy Consumption of the Host and Virtual Machine

The energy consumption of the Host and Virtual Machine was determined based on computations of the CPU utilization and energy consumption of the VM and HM with an interval of 10, as shown in Figure 3.

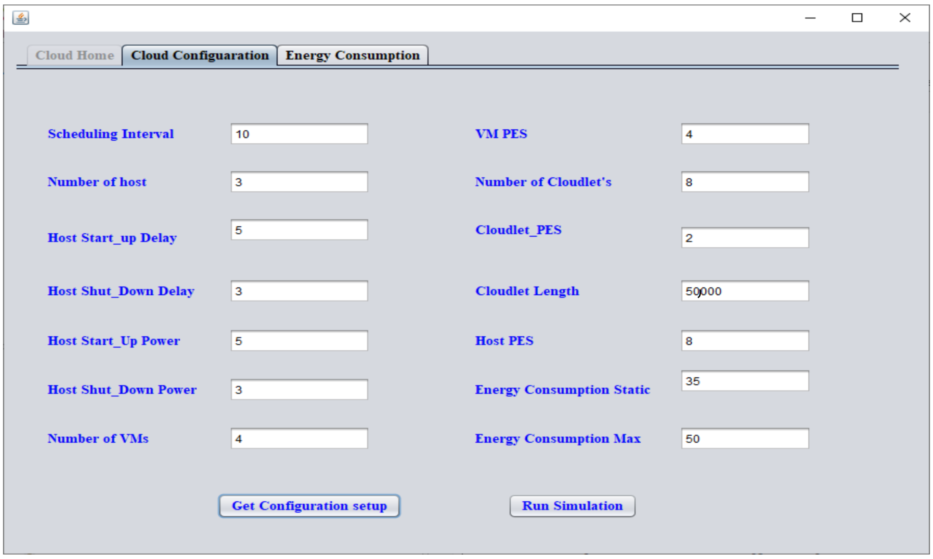

Figure 2. CloudSim Configuration.

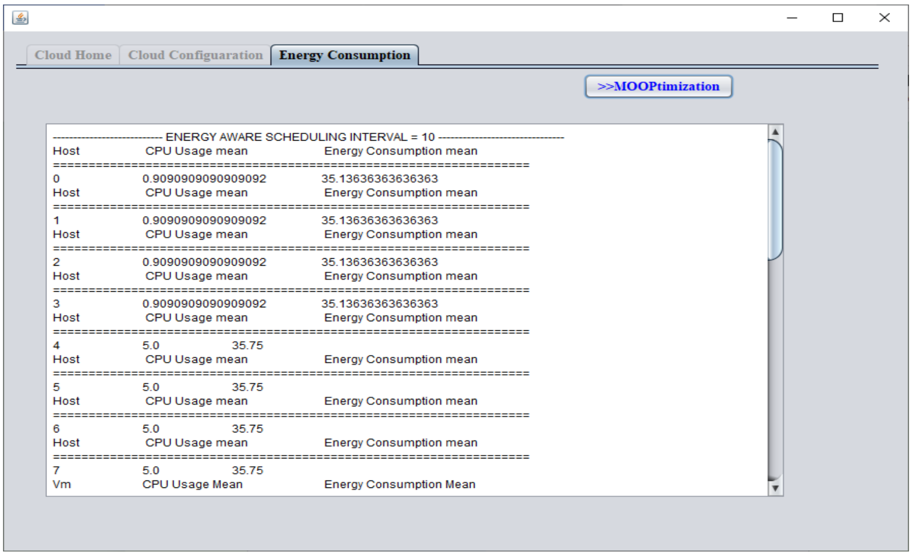

Figure 3. Energy consumption of HM and VM.

4.3. Multi-Objective and Genetic Optimization Technique

After the developed multi-objective and genetic optimization algorithm and energy consumption of the hosts and VM in the GUI window were computed, the enhanced results were displayed in the outcome window. The algorithm was categorized into initialization and looping. Once the initialization process was over, a feasible solution was generated randomly, and the fitness values were evaluated to determine the best solution. Next, the looping parts validated whether a certain terminal condition was met. During the continuous loop, the mutation, crossover, and selection operations were employed in a sequence. Ultimately, the best solution was chosen in the iteration process.

4.4. CPU Utilization

CPU utilization defines the usage of computers based on the processing resources and tasks managed by the CPU [21,22]. Normally, CPU utilization differs based on the computing tasks and work managed by the CPU. Certain tasks need more CPU time, while others need less time due to non-CPU resource needs. Figure 4 provides a graphical representation based on CPU usage and execution time for the host and VM. Table 2 indicates that the CPU usage of the host was 0.9 in 50 s and 0.5 in 100 s, while the CPU usage of the VM was 9.4 in 50 s and 23.2 in 100 s.

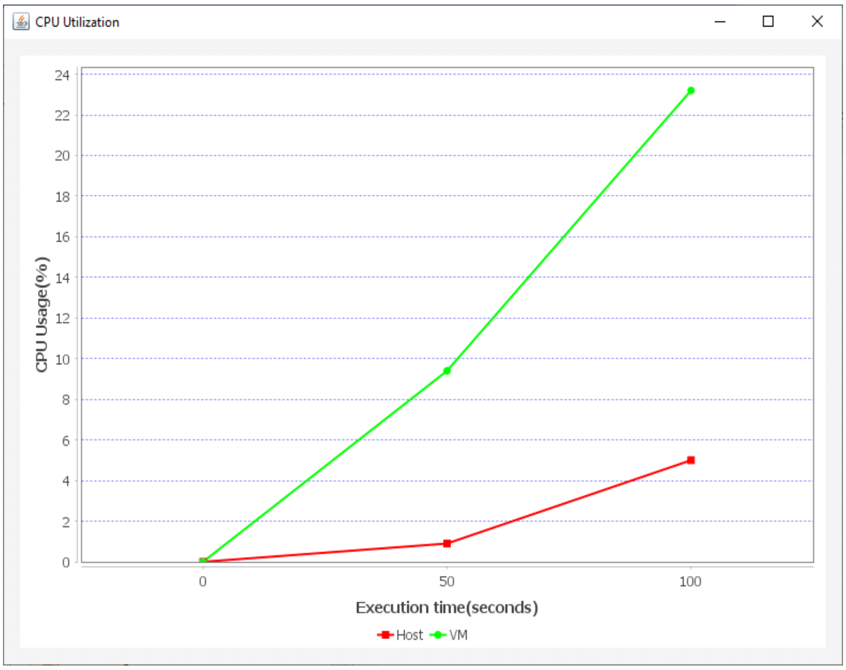

Figure 4. CPU utilization.

Table 2. CPU utilization values based on CPU usage and time.

Host		VM	
Time (s)	**CPU Usage (%)**	**Time (s)**	**CPU Usage (%)**
50	0.9	50	9.4
100	0.5	100	23.2

4.5. Energy Efficiency

The virtual machine and host efficiencies indicate that minimizing the physical servers via virtualization can cut the power and cooling costs while providing sufficient computing power with a minimum amount of space. In the present study, the energy consumption was found to decrease by around 50%. Figure 5 and Table 3 provide a graphical illustration and values, respectively, based on the energy efficiency and time. Furthermore, the VM vs. energy consumption were analyzed. Table 3 indicates that the energy efficiency of the host reached 35.13 KWh in 50 s and 35.75 KWh in 100 s. The energy efficiency of the VM was 18.20 KWh in 50 s and 17.5 KWh in 100 s.

Table 3. Energy efficiency values for host and VM.

Host		VM	
Time (s)	**Energy Efficiency (KWh)**	**Time (s)**	**Energy Efficiency (KWh)**
50	35.13636363636363	50	18.203125
100	35.75	100	17.5

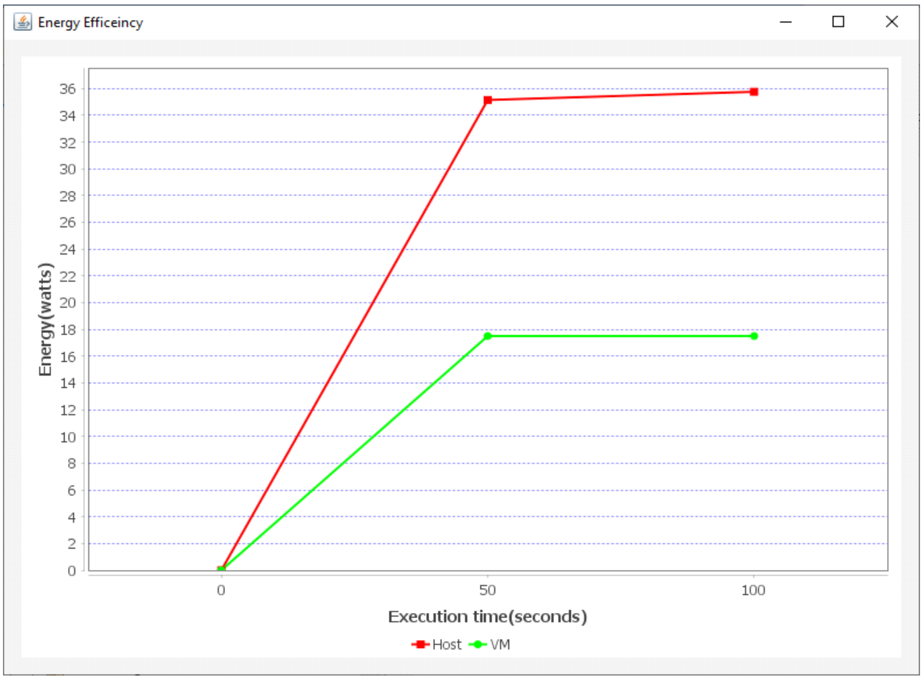

Figure 5. Energy efficiency.

4.6. Energy Consumption

The energy consumption when we were applying workloads on different VMs was measured for the multi-objective genetic algorithms (MOGA) and multi-objective particle swarm optimization (MOPSO). Recently, multi-objective stochastic optimization has been widely used to find the set of solutions with the best performances, which include MOGA and MOPSO. To update a population of solutions, an iterative process is executed in the MOGA and MOPSO algorithms. During each iteration, new trial solutions are generated and evaluated. Both of the algorithms use stochastic operations to produce new solutions with good existing solutions, but their details differ. The VM workloads were measured in seconds, and the energy consumption was calculated in kWh. Firstly, as outlined in Figure 6 and Table 4, the MOGA was evaluated with VM workloads of 25, 50, 75, 100, 125, and 150 s, and it achieved energy consumption values of 43.7, 71.74, 97.23, 113.13, 121.75, and 133.254 KWh, respectively. Secondly, the MOPSO was evaluated with VM workloads of 25, 50, 75, 100, 125, and 150 s, and it achieved energy consumption values of 41.203125, 65.5, 95.5, 96.5, 104.5, and 109.3 KWh, respectively.

Table 4. Energy consumption evaluation for MOGA and MOPSO.

MOGA		MOPSO	
VMs	**Energy Consumption (kWh)**	**VMs**	**Energy Consumption (kWh)**
25	43.7	25	41.203125
50	71.74	50	65.5
75	97.23	75	95.5
100	113.13	100	96.5
125	121.75	125	104.5
150	133.254	150	109.3

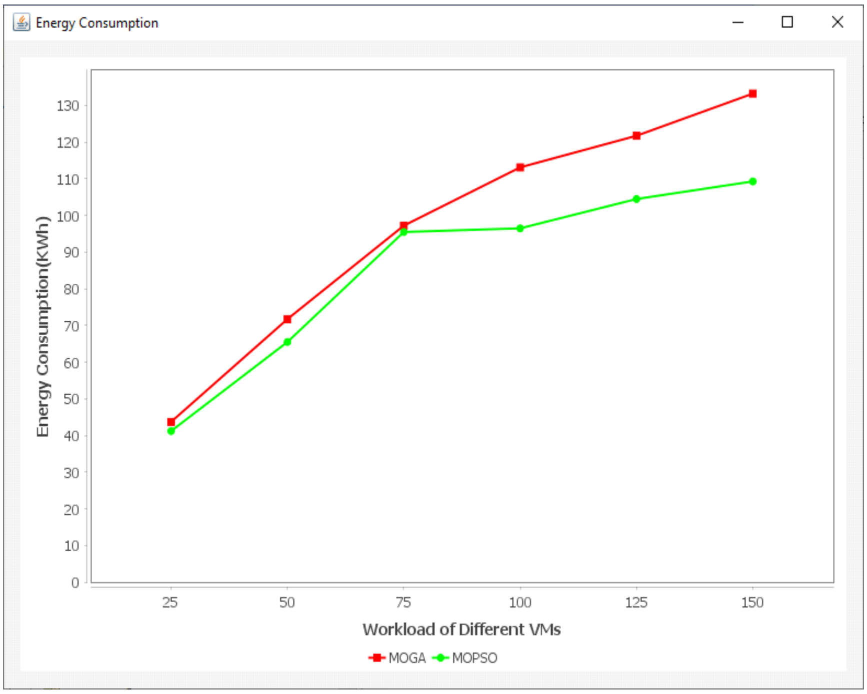

Figure 6. Energy consumption.

4.7. Physical Machine Shut Down

We measured the physical machine numbers that shut down when we were applying workloads on different VMs for MOGA and MOPSO. The VMs were also measured in seconds for the physical machine shut down number. Firstly, as outlined in Figure 7 and Table 5 the MOGA was evaluated with VM workloads of 25, 50, 75, 100, 125, and 150 s for the PM shutdown number with corresponding PM shutdown numbers of 150, 200, 300, 300, 200, and 200. Secondly, the MOPSO was evaluated with VM workloads of 25, 50, 75, 100, 125, and 150 s for corresponding PM shutdown numbers of 100, 175, 260, 140, 135, and 135.

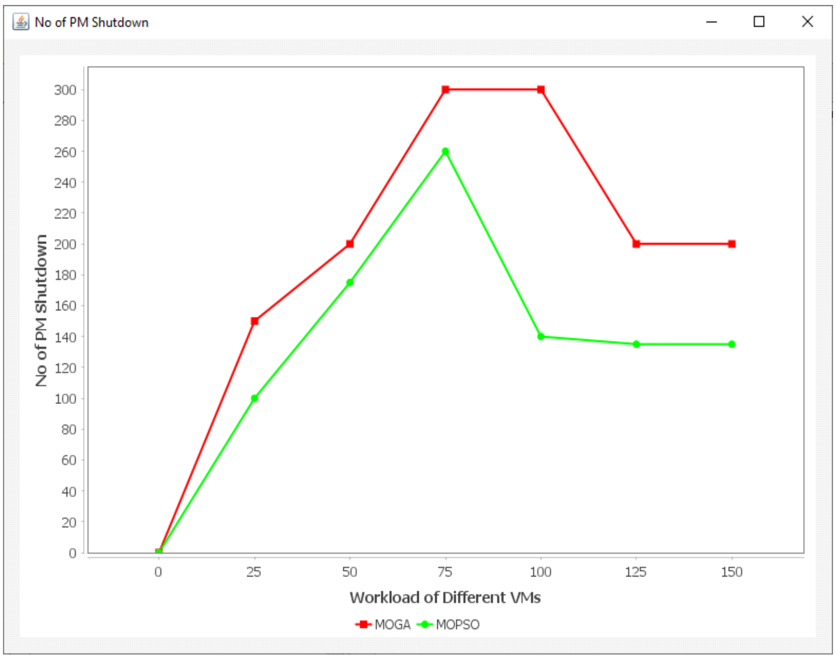

Figure 7. Physical machine shutdown.

Table 5. PM shutdown with VMs for MOGA and MOPSO.

MOGA		MOPSO	
VMs	**No of PMs Shutdown**	**VMs**	**No of PMs Shutdown**
25	150	25	100
50	200	50	175
75	300	75	260
100	300	100	140
125	200	125	135
150	200	150	135

Based on the above analysis, all of the parameters including CPU utilization, energy consumption, and scheduling were simulated and evaluated with proper validation in CloudSim. Based on the achieved results, the multi-objective PSO optimization achieved better and more efficient results under different parameters compared to the multi-objective GA optimization. Moreover, integrating condition-based maintenance (CBM) in the Cloud with the proposed MOPSO technique in real-time will help in monitoring and enhancing the performance of the IoT technologies. This technique will prevent performance degradation without other predictive maintenance tasks by improving the thermography analysis. Hence, utilizing CBM applications with the proposed novel technique will decrease the maintenance costs and energy requirements. CBM application economics, however, must be further studied in practice.

5. Conclusions and Future Work

Cloud data centers are massive, leading to a high level of energy consumption and greater task execution times. Therefore, it is necessary for the users to transfer data frequently and the for data centers to regularly utilize VM scaling to enhance the system resource efficiency. In the current research, a novel approach based on multi-objective optimization, MOPSO, was utilized with CloudSim as the underlying simulator to evaluate the VM allocation performance. Here, we computed the energy consumption, the CPU utilization, and the number of executed instructions in each of the scheduling intervals for complex VM scheduling solutions that increase the energy efficiency and reduce the execution time. The final results demonstrate that multi-objective PSO optimization has achieved better and more efficient results using different parameters when it was compared to those of multi-objective GA optimization. In future, we plan to apply the current technique to front-to-end simulations and validate the final results on a real machine.

Author Contributions: Conceptualization, R.C. and S.P.; methodology, R.C.; software, R.C.; Validation, R.C. and S.P.; writing—original draft preparation, R.C.; supervision, S.P.; writing—review and editing, R.C. and S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Dörterler, S.; Dörterler, M.; Ozdemir, S. Multi-objective virtual machine placement optimization for cloud computing. In Proceedings of the 2017 International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, Morocco, 16–18 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
2. Mohammadzadeh, A.; Masdari, M.; Gharehchopogh, F.S. Energy and cost-aware workflow scheduling in cloud computing data centers using a multi-objective optimization algorithm. *J. Netw. Syst. Manag.* **2021**, *29*, 31. [CrossRef]
3. Liu, X.; Cheng, B.; Wang, S. Availability-aware and energy-efficient virtual cluster allocation based on multi-objective optimization in cloud datacenters. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 972–985. [CrossRef]

4. Chen, J.; Du, T.; Xiao, G. A multi-objective optimization for resource allocation of emergent demands in cloud computing. *J. Cloud Comput.* **2021**, *10*, 20. [CrossRef]
5. Dubey, K.; Sharma, S.C. A novel multi-objective CR–PSO task scheduling algorithm with deadline constraint in cloud computing. *Sustain. Comput. Inform. Syst.* **2021**, *32*, 100605. [CrossRef]
6. Li, S.; Pan, X. Adaptive management and multi-objective optimization of virtual machine in cloud computing based on particle swarm optimization. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 102. [CrossRef]
7. Son, A.Y.; Goh, S.; Huh, E.N. Multi-objective optimization method for resource scaling in cloud computing. In Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, Langkawi, Malaysia, 5–7 January 2018; pp. 1–6.
8. Li, R.; Zheng, Q.; Li, X.; Yan, Z. Multi-objective optimization for rebalancing virtual machine placement. *Future Gener. Comput. Syst.* **2020**, *105*, 824–842. [CrossRef]
9. Paknejad, P.; Khorsand, R.; Ramezanpour, M. Chaotic improved PICEA-g-based multi-objective optimization for workflow scheduling in cloud environment. *Future Gener. Comput. Syst.* **2021**, *117*, 12–28. [CrossRef]
10. Devaraj, A.F.S.; Elhoseny, M.; Dhanasekaran, S.; Lydia, E.L.; Shankar, K. Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments. *J. Parallel Distrib. Comput.* **2020**, *142*, 36–45. [CrossRef]
11. Prem Jacob, T.; Pradeep, K. A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization. *Wirel. Pers. Commun.* **2019**, *109*, 315–331. [CrossRef]
12. Ghetas, M. A multi-objective Monarch Butterfly Algorithm for virtual machine placement in cloud computing. *Neural Comput. Appl.* **2021**, *33*, 11011–11025. [CrossRef]
13. Langhnoja, H.K.; Joshiyara, H.A. Multi-objective based integrated task scheduling in cloud computing. In Proceedings of the 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 12–14 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1306–1311.
14. Al-Tarazi, M.; Chang, J.M. Network-aware energy saving multi-objective optimization in virtualized data centers. *Clust. Comput.* **2019**, *22*, 635–647. [CrossRef]
15. Luo, J.; Song, W.; Yin, L. Reliable virtual machine placement based on multi-objective optimization with traffic-aware algorithm in industrial cloud. *IEEE Access* **2018**, *6*, 23043–23052. [CrossRef]
16. Li, Y.; Chen, M.; Dai, W.; Qiu, M. Energy optimization with dynamic task scheduling mobile cloud computing. *IEEE Syst. J.* **2017**, *11*, 96–105. [CrossRef]
17. Shaw, R.; Howley, E.; Barrett, E. Applying Reinforcement Learning towards automating energy efficient virtual machine consolidation in cloud data centers. *Inf. Syst.* **2022**, *107*, 101722. [CrossRef]
18. Rehman, A.; Hussain, S.S.; ur Rehman, Z.; Zia, S.; Shamshirband, S. Multi-objective approach of energy efficient workflow scheduling in cloud environments. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4949. [CrossRef]
19. Abualigah, L.; Diabat, A. A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. *Clust. Comput.* **2021**, *24*, 205–223. [CrossRef]
20. Guo, X. Multi-objective task scheduling optimization in cloud computing based on fuzzy self-defense algorithm. *Alex. Eng. J.* **2021**, *60*, 5603–5609. [CrossRef]
21. Hosseinzadeh, M.; Ghafour, M.Y.; Hama, H.K.; Vo, B.; Khoshnevis, A. Multi-objective task and workflow scheduling approaches in cloud computing: A comprehensive review. *J. Grid Comput.* **2020**, *18*, 327–356. [CrossRef]
22. Duan, H.; Chen, C.; Min, G.; Wu, Y. Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. *Future Gener. Comput. Syst.* **2017**, *74*, 142–150. [CrossRef]