International Journal of Mathematical, Engineering and Management Sciences
Vol. 7, No. 5, 681-696, 2022
https://doi.org/10.33889/IJMEMS.2022.7.5.045

A Cost Effective and Energy Efficient Algorithm for Cloud Computing

Priyanka Vashisht
Department of Computer Science and Engineering,
Amity University, Gurugram, Haryana, India.
E-mail: priyanka.vashisht@gmail.com

Vijay Kumar
Department of Mathematics, Amity Institute of Applied Sciences,
Amity University, Noida, Uttar Pradesh, India.
Corresponding author: vijay_parashar@yahoo.com

(Received on February 18, 2022; Accepted on July 10, 2022)

Abstract

Cloud-Computing offers high performance solution to solve complex engineering and scientific tasks by deploying resources at geo-diverse locations. With the large-scale demand of scientific and engineering jobs, huge number of cloud data centres needs to be constructed to fulfil the requirement of the jobs. The extensive use of cloud data centres leads to increases in cost as well as energy consumption. In this paper, an agent based Cost-Effective Energy Efficient scheduling algorithm, namely, CEEE has been proposed. To establish the effectiveness of the proposed algorithm, simulation is performed on CloudSim environment. The proposed algorithm is compared with state of art scheduling algorithms namely, Randam and MaxUtil. The experimental results demonstrates that CEEE algorithm is capable of refining energy efficiency with reduced cost. The proposed algorithm outperforms the prevailing algorithms in terms of energy consumption, resource utilization, number of hosts in sleep mode and completion time.

Keywords- Cloud computing, Job scheduling, Energy efficiency, Resource utilization, VM consolidation.

1. Introduction

Cloud computing is the technique in which various resources such as computing, storage etc. can be provided through virtualization over the internet. The services provided by the cloud computing architecture is based on data centres. Data centres consists of number of dedicated serves, air conditioners, cables etc. which consumes large amount of energy and produces enormous volume of Carbon-di-oxide (CO2). These data centres are evolving perpetually to address the need of extensive data processing and its execution on servers which further leads to more energy consumption. As a result, optimization of energy utilization in cloud computing has become a significant issue these days due to its impact on performance, environment and economy (Bohrer et al., 2002; Kusic et al., 2002; Mastelic et al., 2014). The foremost challenge of cloud computing is an efficient use of energy; hence researchers have turned their focus towards more sustainable computing, namely, Green cloud computing.

In Green cloud computing, the service provider takes corrective measures to reduce the impact of cloud technology on environment. Its scope is not limited to key computing resources such as storage, processors and visualization facilities, but it also considers other facilities associated with computing resources including supplementary equipment such as space that computing resources occupy or cooling mechanism etc. (Lee et al., 2012). Advancement in hardware technology have optimized energy consumption up to a certain level. Though, the energy consumption pattern of computing and associated facilities is still very alarming for sustainable computing. If the resources are not utilized efficiently i.e., either they are over utilized or underutilized, then a huge volume of energy will be consumed. Hence researchers are providing various software solutions for optimization of energy including virtualization, scheduling and job consolidation. Energy consumption and utilization of resources are linearly associated and are tightly coupled in cloud environment. According to studies (Barroso and Holzle, 2007; Fan et al.,2007) the energy utilization of resources in idle state can be as much as 60% and during resource utilization it can be as low as 20%. Due to the disparities of resource utilization in the geo diverse cloud environment, scientist suggested an effective technique, namely, Virtual Machine (VM)/jobs consolidation which efficiently exploit resources and in turn decrease the consumption of energy. The VM/jobs consolidation problem is the method where sufficient number of resources are assigned to various tasks without violating the Service Level Agreement (SLA) constraints. The main purpose is to assign available resources to the user jobs that implicitly or explicitly decreases the energy consumption while Quality of Service (QoS) constraints are met. Job/VM consolidation is Non deterministic Polynomial Time (NP) complete problem due to large number of solution space. It can be reduced by applying numerous constraints on problem set but still resultant solution set is very huge (Stavrinides et al., 2019). This method is suggestively qualified by virtualization that concurrently run several jobs on a single physical machine.

In this work, the suggested CEEE method describes a two-step strategy for migrating virtual machines to other hosts or region in order to reduce energy consumption by dynamically adjusts the threshold value pair. The rest of this paper is organised as follows: Section 2 includes a quick discussion of relevant work. The third unit lays out a network architecture for calculating cloud energy use. In section 4, we introduce an adaptive threshold-based cost-effective energy efficient algorithm (CEEE) and show an application based on it. In section 5, results are discussed to check the effectiveness of our algorithm. The last section contains a conclusion as well as some future directions.

2. Related Work

Cloud computing is a new model for offering computer and storage resources as a service based on customer request (Vashisht et al., 2019; Vashisht et al., 2020; Vashisht and Kumar., 2020). The rising processing and storage requirement of cloud environment has resulted in a significant rise in energy usage (Gao et al., 2013). Computation resources primarily consumes energy along with other resources such as memory, storage, and networking. Effective resource administration and efficient resource utilisation are both critical in distributed environment. The efficiency of system can be improved by appropriately allocating requests to the resources while effective resource utilisation is achieved by minimising the quantity of resources used. The hardware, such as CPU, has a variety of modes, the most popular of which are active, sleep and idle, each of which consumes different amount of energy. Specifically, even when simply considering the processor in executing mode, the connection amongst processor utilisation and energy usage is not linear (Choi et al., 2016). While CPU utilisation with workload variance approaches 90%, power consumption rises steeply, and the CPU uses power even when it is idle. Lien et al. (2006) exhibits the association between energy consumption and resource utilization in detail; even if the CPU is only used 10% of the time, the power consumed is more than 70% of the peak power. It implies that by moving the load, we may shut down certain hosts and save power. When CPU utilisation exceeds 70%, however, the power consumption rate rapidly increases. Few researchers (Choi et al., 2016; Xu et al., 2012; Xiao et al., 2012; Gmach et al., 2008) suggests that resource integration can reduce energy consumption by relocating over-burdened hosts or shutting down the idle nodes.

The basic concept of the Minimization of Migrations (Beloglazov et al., 2012) strategy is as follows: first define two threshold values for VM migration, once the resource utilisation surpasses the high threshold value, then migrate the VM with the minimum CPU utilization. After pre-allocating the VM to the destination node, based on the findings, the subsequent step is to pick a node with the minimum amount of energy usage. A two-phase dynamic resource scheduling approach was suggested by Xu et al. (2012). Similarly, the strategy established two virtual machine migration thresholds. The first is to use a single exponential smoothing technique to anticipate virtual machine load. When the expected value exceeds the threshold range, the virtual machine is migrated using the smallest cosine distance between the maximum allowable resource on the destination host and the computational usage of the VM. The association between processor utilization and energy usage is not entirely linear; when the utilisation frequency surpasses a particular threshold value, there is a considerable rise in energy use. Choi et al. (2016) classifies tasks into two groups: computationally intensive and data intensive. Based on transmission costs, it selectively migrates computationally intensive activities to hosts with less computationally intensive tasks.

Luo et al. (2014) presented a dynamic resource movement approach. Researchers categorised resources into three states based on their usage: hot areas, warm places, and cold spots. Based on skewness policies, hot-spot resources are moved to balance load, while cold-spot resources are released to save energy. Gmach et al. (2008) used load log which modify the excess threshold, as well as combine MC, React, Hist and other overload resource applications to meet the goal of cluster load balancing. As previously noted, these studies do not appropriately respond to shifting resource loads and the ambiguity of resource requests, resulting in SLA breaches if more idle resources are closed. As a result, we must identify a viable adaptive threshold determination approach from a theoretical standpoint as soon as possible. The introduction and development of three-way decision theories provide a fresh viewpoint on the subject. The primary principle behind three-way decisions as an efficient solution to a problem is to split the full set into three independent pieces and take various actions for each portion in order to gain the highest benefit or the lowest cost.

The Adaptive Thresholds Migration (ATM) approach is presented in virtual machine migration by incorporating three-way decisions (Jiang et al., 2019). First, the suggested technique evaluates cluster load based on resource utilisation, and then it dynamically determines two thresholds, resulting in the VMs being classified into three categories using three-way decision principles, namely, normal nodes, idle nodes and overloaded nodes. Second, nodes in various areas behave in different ways. ATM selects the virtual machine to migrate and optimises the target host based on the cluster's resource utilisation and transmission overhead for idle and overloaded nodes. Experiments using CloudSim reveal that the ATM algorithm saves roughly 20% of electricity when compared to TCEA and MM.

Chen et al. (2021) created a novel system energy consumption model that accounts for the runtime, switching, and computation energy consumption of all involved servers (both cloud and edge) and IoT devices. Further, they utilised a Self-Adaptive Particle Swarm Optimization algorithm with Genetic Algorithm operators (SPSO-GA), a novel energy-efficient offloading approach is developed. With layer partition procedures, this innovative technique can efficiently make offloading decisions for DNN layers, reducing the encoding dimension and improving SPSO-GA execution time. The proposed technique can greatly reduce energy usage when compared to other traditional ways, according to simulation findings.

Khemili et al. (2022) proposed a new Fuzzy-FCA approach for VNF placement built on Formal Concept Analysis (FCA) and fuzzy logic in a mixed environment based on cloud data centres and Multiple Access Edge Computing (MEC) architecture to combine Virtual Network Function (VNF) groups into a minimum amount of VMs with approximation of the association relation to a measure of confidence under the context of possibility theory. In another approach, Hummaida et al. (2022) analyses VM response time and determines the CPU threshold at which response time exceeds a set SLA level, then utilises that threshold to migrate VMs. It utilises reinforcement learning (RL) which encourages high CPU usage and penalises those who fail to meet SLAs.

The proposed CEEE algorithm outlines a virtual machine migration strategy that employs a two-fold approach. To accomplish the goal of reducing energy consumption, CEEE dynamically regulates the threshold value pair by constantly assessing resource utilization and controlling it rationally by releasing an idle host with a load under Lower Threshold (LT). It also helps in migrating an extra load higher than Upper Threshold (UT) to attain the goal of reducing energy consumption. In the next section architecture of proposed CEEE algorithm is discussed.

3. Network Model

An agent-based cost-effective energy efficient (CEEE) algorithm has been proposed which uses Fat-tree architecture and Pictorial representation of this architecture is depicted in Figure 1. The consideration of this architecture is done due of its performance in terms of throughput and average network delay (Bilal et al., 2013). Here, the infrastructure follows switch-centric network topology which is modelled as collection of small units called region. Each region is a set of hosts $H = \{h_1, h_2, \dots, h_i\}$, where H signifies set of physical hosts. For each host h_i a finite set of virtual machines $VM_i = \{v_{i1}, v_{i2}, \dots, v_{ij}\}$ exists, having resources $R_i = \{r_{i1}, r_{i2,} \dots, r_{il}\}$, $R \in H$. Number of hosts in the region is same as number of switches. The switches are methodically placed in two successive layers. The lower layer switches are connected to i /2 hosts of lower layer which are further connected to remaining i /2 switches in every region.

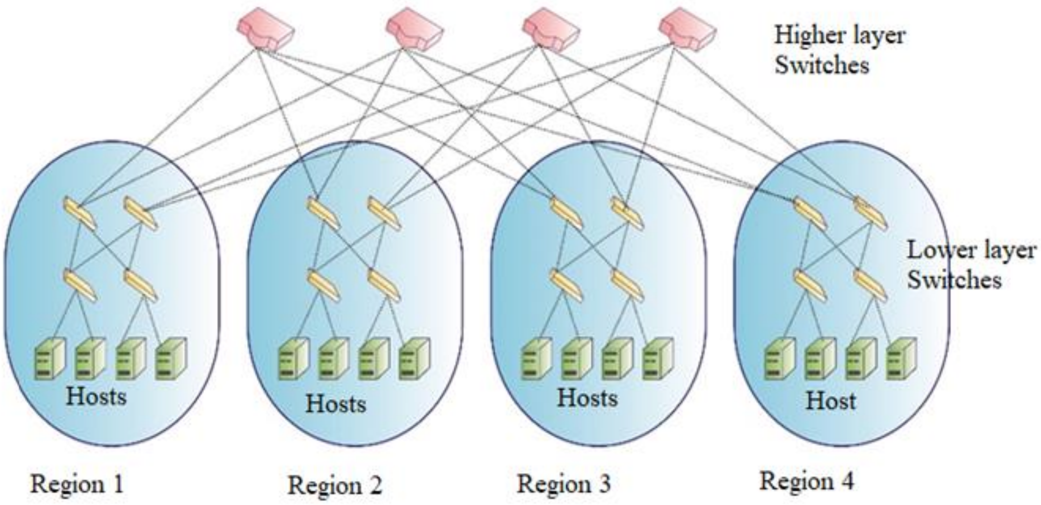

Figure 1. The fat-tree architecture.

A host can be in three states: *active*, *idle* and *passive* as shown in Figure 2.

- In *active state*, all the host within a region are in working condition and executing the job using allocated resources to fulfil the request made by user.
- In *idle state*, the job is not executing but is waiting for some I/O operation to happen.
- In *passive state*, all the host are in sleep mode i.e. servers are not executing any user requests and resource utilization is 0%.

Each host in the region contains agent which maintain information of resources such as storage, resource availability, bandwidth and computational unit. An agent is an intelligent entity which communicates with various components of the region. Each agent in a host is connected to its neighbouring ***k*** hosts through a static topology so that computational usage can be shared whenever required. The information gathered by the agent helps it to define when it should allocate the virtual machine (VM) to the user request.

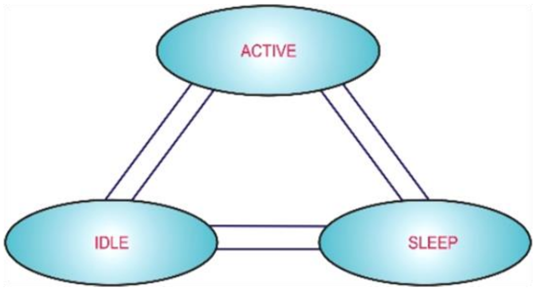

Figure 2. Various states of hosts.

Every host contains a database which maintains the information of resources used to fulfil the requirement of previous requests. When a new request arrives in the region, the agent on the host examines the record database for the request's resource requirements. It then sends the request's required information to the Resource Analyst (RA). The job of resource analyst is to evaluate type and number of resources required to process the user request based on the historical data. Afterwards, the Resource Manager (RM) determines whether to process the job inside a host or move it to a neighbouring region based on the availability of resources in the region.

If the requirement of the user request is met by the existing capacity of the region, then the Resource Broker (RB) will allocate the best VM out of available VMs to the user request. For provisioning of user requests, a dynamic pool of VMs is maintained by resource broker. After allocation of VMs to the required job, the resource monitor will keep regular watch on the changing resource requirement of the job. Each time the need of job changes, the entire procedure of VM provisioning is recycled. The components and working of a region are depicted in Figure 3.

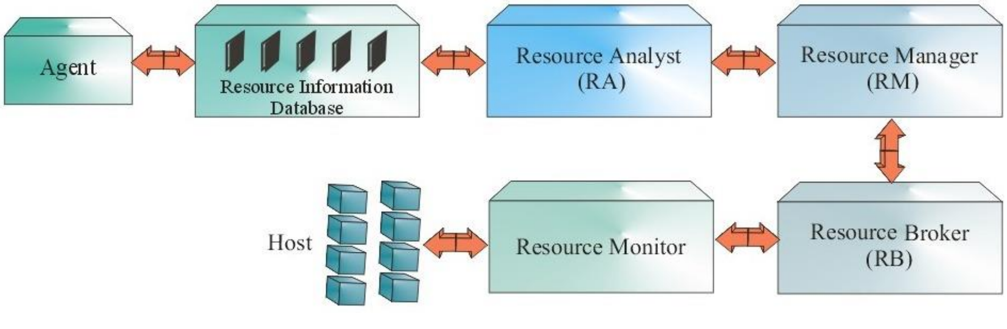

Figure 3. Detailed working structure of region.

4. Proposed Scheduling Algorithm

CEEE algorithm was proposed to provide a cost-effective solution for energy consumption and follows two-step technique. This approach works on two levels – regional level and global level. At regional level CEEE consolidate the jobs on limited number of VMs hence conserving the energy locally. At global level, the VM are migrated to other region having the less workload and sending more and more region to sleep mode. Detailed description of CEEE is provided in subsequent section.

4.1 Intra- Region Energy Optimization using CEEE

A host h_i comprises of VM v_{ij} having the computing resource c_{ij} and memory resource m_{ij}. The resource r_{ij} on jth $VM\ (v_{ij})$ of ith host h_i can be mathematically represented as:

$$r_{ij} = \sum_{i=1}^{k}\sum_{j=1}^{n} c_{ij} + \sum_{i=1}^{k}\sum_{j=1}^{n} m_{ij} \tag{1}$$

Where, n represents the number of VM on host h_i. VM consist of two components – memory & computing capacity, where amount of memory is measured in GB and amount of computing capacity is measured in MIPS. The host can execute certain instructions per seconds and is measured as millions of instructions per second (MIPS). Hence, equation (1) can be re-written as:

$$r_{ij} = \sum_{i=1}^{k}\sum_{j=1}^{n} MIPS_{ij} + \sum_{i=1}^{k}\sum_{j=1}^{n} GB_{ij} \tag{2}$$

Initially when the jobs/requests arrive at a region, an agent located on the host machine will check the record information of similar job from the history logs and share the information with RA to know the resource requirement of the job. RA will evaluate the current resource occupancy ratio of all the host within the region. This resource occupancy matrix displays the number of engaged jobs and the maximum jobs that each resource can handle. Hence, the resource occupancy (φ_i) of host h_i is calculated as follows:

$$\varphi_i = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n} r_{ij_{utlization}}}{\sum_{i=1}^{k}\sum_{j=1}^{n} r_{ij_{allocated}}} \tag{3}$$

where, $r_{ij_{utlization}}$ is total number of computing and memory resources being utilized by jth $VM\ v_{ij}$ of ith host h_i at time τ and $r_{ij_{allocated}}$ is total number of computing and memory resources that are allocated to the host h_i . The occupancy ratio of all the host will be shared with the RM of the region. The RM will choose the host with minimum resource occupancy i.e. $min(\varphi_1, \varphi_2, \ldots\ldots \varphi_i)$. The resources will be allocated to the request by the resource broker. Due to the dynamic nature of cloud environment the jobs on a particular host keep on changing which is being tracked by the resource monitor. While monitoring the request, if at any instance τ the occupancy of the host goes down the lower threshold (LT) value or marks the upper threshold (UT) value, RA will indicate the Manager to migrate its VM to neighbouring host.

Each *active* host can be in following states: *underloaded, balanced* and *overloaded.*

- *Underloaded:* where either the processor is idle for maximum duration of time or the memory is utilization is very less.
- *Balanced:* where the computing and storage utilization is less than UT and more than LT.
- *Overloaded:* where the computing and storage utilization is more than UT.

The upper threshold (UT) and lower threshold (LT) value of host h_i is calculated as averages of the maximum and the minimum of the resource utilization over the process history, which signifies the current workload on host h_i.

$$UT_i = \frac{\left\{max \sum_{i=1}^{k} \sum_{j=1}^{n} (r_{ij_{utlization}})\right\}}{R} \tag{4}$$

$$LT_i = \frac{\left\{min \sum_{i=1}^{k} \sum_{j=1}^{n} (r_{ij_{utlization}})\right\}}{R} \tag{5}$$

If the value of resource occupancy (φ_i) of a host falls below the LT, VM is migrated to neighbouring host. The host with $min(\varphi_1, \varphi_2, \ldots\ldots \varphi_i)$ will be selected on which job can be positioned. The host with threshold value less than LT, will be put in sleep mode after migrating all the jobs to neighbouring host. Moreover, if at any instance of time if the resource occupancy of host exceeds upper threshold, the VM can be released and migrated to other neighbours in the region. The jobs are consolidated to few active hosts which effectively manages the resources in long and short terms in cloud environment. The migration cost within the region is negligible as bandwidth within the cluster is very high. The migration of VM on another host onto the neighbouring host in a region and making more and more host into sleep mode will help our strategy to establish the energy efficient model at regional level. This can be extended further at global level by migrating the jobs across various regions or inter-region migration is possible. . If the overall upper and lower threshold value of the region/cluster (set of hosts) is not in the limits then jobs are migrated to another region which is discussed in next section.

4.2 Inter-Region Energy Optimization using CEEE

The selection of host on another region is based on multiple constraints and is a multi-objective optimization problem. Here region having the cost of transferring VM to another cluster with the least energy consumption should be selected. The energy optimization of a region can be computed as:

$$\varepsilon_a = \left(\left(1 - \frac{\sum_{i=1}^{m} \varphi_{ij}}{m}\right) * \ h_{idle} + (k-m) * h_{sleep}\right) \tag{6}$$

where, ε_a is the total energy saved in the region using CEEE, $\frac{\sum_{i=1}^{m} \varphi_{ij}}{m}$ average resource utilization of all active hosts. h_{idle} is the number of host presently in idle state. k is total number of host in a region. h_{sleep} is number of hosts in sleeping mode right now. $\left(1 - \frac{\sum_{i=1}^{m} \varphi_{ij}}{m}\right) * \ h_{idle}$ is the amount of energy saved when the host are in idle state, while $(k-m) * h_{sleep}$ is the energy saved by the host in sleep mode. The total consumption of power between two regions of the data centre network should be minimised subject to following constraints:

$$BW_{ig} \leq BW_{available} \tag{7}$$

Here, BW_{ig} is the bandwidth required by the VM for allocation / migration.

$$BW_{available} = BW_b - \ BW_{utlized} \tag{8}$$

where, BW_b is bandwidth of slowest link between two regions and $BW_{utlized}$ utilized bandwidth between the link at any given τ.

$$\textstyle\sum_{i=1}^{k} \sum_{j=1}^{n} c_{ij} \leq c_i \tag{9}$$

$$\textstyle\sum_{i=1}^{k} \sum_{j=1}^{n} m_{ij} \leq m_i \tag{10}$$

Constraint (7) ensures that available bandwidth is bandwidth required during migration of VM. In the similar manner, it is guaranteed in constraint (9) that sufficient computing power is offered for the incoming VM. The total memory consumption of all the active VMs must be less than the total available memory of the host. Moreover, it is ensured that the size of VM which needs to be migrated should be less than the existing available memory of the host and is shown as a limitation in constraint (10). Also, the computing capacity of the host must be less than the overall computing capacity used by all the running VMs. The total cost (ℵ) of transfer can be depicted as:

$$\aleph = \ \alpha * \ \varepsilon_a + \beta * \ \gamma \tag{11}$$

where, α, β are weighting factor, such that $\alpha + \beta = 1$ and γ is propagation delay. γ is directly dependant on the bandwidth between the hosts. The region with lowest cost will be selected for migrating the jobs.

The proposed energy model is developed considering that usage of computing resource has a linear association with energy consumption. In other words, resource utilization is appropriate parameter to quantify the energy consumption for the task. Whenever the user request arrives in the region, the resource broker will allocate free VM to the job. During the peak hours, requirement of resources is more than the off-peak hours. However, there will be high energy consumption if the hosts continue to offer same number of resources during the off-peak hours. Hence, the resource broker will migrate the VMs to limited active hosts and put rest of the hosts in sleep mode. Pseudocode of Inter-region and Intra-region VM Scheduling algorithm is shown in Figure 4 and Figure 5 respectively.

Algorithm 1 ***Intra-Region VM/ Job Migration***

```
INPUT:
$k$ : Number of neighbours        $h_i$: host in the region
$\emptyset$ : Occupancy ratio
OUTPUT: Energy preserving inter region migration
Calculate $\emptyset_i$
if $\emptyset_i < LT$ then
    for $i = 1$ to k do
        Find the minimum $\emptyset_i$
    end for
    if $(\emptyset_t + \emptyset_i) < UT$ then
        Set $\emptyset_t = \emptyset_t + \emptyset_i$
        Retain $h_i$ in sleep mode
    end if
end if
```

Figure 4. Intra-region VM scheduling algorithm.

Algorithm 2 ***Inter-Region VM/ Job Migration***

```
INPUT: Utilization information of each host
OUTPUT: Energy preserving intra cluster migration
for each host $h_i$ in the region do
  Calculate workload $\frac{\sum_{i=1}^{m} \varphi_{ij}}{m}$
      Calculate Energy based on workload $\varepsilon_a$
  if $(BW_{ig} \leq BW_{available})$ && $(\sum_{i=1}^{k} \sum_{j=1}^{n} c_{ij} \leq c_i)$ &&$(\sum_{i=1}^{k} \sum_{j=1}^{n} m_{ij} \leq m_i)$ &&$(v_{ij} \leq m_{i_{available}})$ then
      Calculate cost
  Select $Region$ with minimum cost
      for each host $h_k$in the region do
          Find highest $\emptyset_k$ such that $\emptyset_i + \emptyset_j <= UT$
      end for
      Migrate VMs from host $h_i$ to host $h_j$
          Keep host $h_i$ in sleep mode
      end if
end for
```

Figure 5. Inter-region VM scheduling algorithm.

5. Simulation Environment and Performance Metrics

The experimental setup was done using the java-based simulator, namely, CloudSim (Buyya et al., 2009). The configuration used for this problem is shown in Table 1.

Table 1. Parameter setup.

Parameters	**Values**
Number of Host	1000
Processor	Quad-core, Single Core
Number of regions	5
RAM	8 GB
Number of assigned requests	100-12000
Processing Capacity (MIPS)	2500-10000
Hard Drive	1 TB

For the current scenario, total number of hosts are 1000. The host can have either quad-core or single core processor with the processing capacity varies between 2500-10000 MIPS. RAM assigned to each host is 8 GB, and the size of hard drive is 1 TB. The hosts are connected to each other within a region having infinite bandwidth whereas, bandwidth of intra-region channel is 1 Gbit/s. There are four types of VM configuration with computing capacity of 500,1000, 1500, 2500 MIPS. The VMs of each type is generated randomly on various hosts. During our investigation, service requests arrive using a Poisson distribution. Several experiments are carried out under synthetic workload. The workload includes requests arriving pattern, which varies between 100 to 12000.

The proposed CEEE method is evaluated using a variety of metrics, including energy consumption, number of servers in sleep mode, resource usage, and completion time. These measures were chosen since they have been extensively embraced and utilised in several methods. The findings are compared to the outcomes of two other algorithms, namely, Random and MaxUtil. In Random algorithm when a host is overloaded or underloaded, migration of VM/Job to different nodes are done in random pattern. Whereas, in MaxUtil algorithm the average utilisation during the current job's processing time is the most important component of MaxUtil cost function. It has a dual benefit in terms of consolidation density and energy savings. MaxUtil, in other words, tends to make a resource more useful.

5.1 Performance Metrics for the Experiment

The performance of the proposed CEEE algorithm is evaluated using the following performance metrics:

- Energy Consumption
- Number of hosts in sleep mode
- Resource Utilization
- Completion Time

5.1.1 Energy Consumption

Two strategies are adopted to evaluate the energy consumption metrics which is discussed in this section.

(i) Comparing with Static Threshold

The static threshold values are compared with adaptive threshold generated using proposed CEEE algorithm. Table 2 shows the static threshold settings, whereas Figure 6 shows the simulation results. Table 2 reflects multiple cases based on certain static threshold values such as in case 1 the lower threshold value is set to 30 and upper threshold value is set to 70 and is represented as T1. Similarly other cases T2, T3, T4, T5, T6 have fixed threshold values. The results in Figure 6 depicts that the behavioural pattern of energy consumption with static threshold is unpredictable with varying VMs ranging from 100-7000. For example, with case 3 (T3), when number of VM are low its energy consumption is also low but it increases exponentially when load is increased on the host machine and then after some variations the energy consumption increases gradually. In case 1 (T1), energy consumption is initially high but as the load increases consumption starts decreasing and then eventually increases again. Figure 6 reveals that growth rate of the adaptive CEEE method is modest with shifting workload as compared to the static threshold, making it more adaptable to dynamic nature of the cloud environment.

Table 2. Static setup inputs.

Case	Lower Threshold	Upper threshold
T1	30	70
T2	40	80
T3	40	70
T4	50	80
T5	50	90
T6	60	90

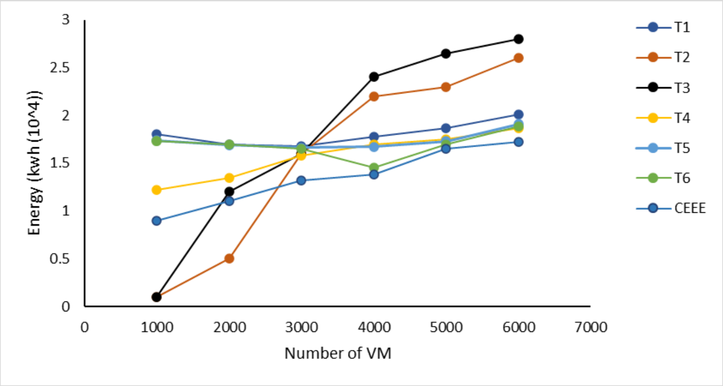

Figure 6. Static threshold vs adaptive threshold.

(ii) Varying Resource Utilization Pattern

The performance of the CEEE algorithm is judged in the second scenario using varied resource patterns. A comparison of CEEE with two additional algorithms, MaxUtil and Random, is also performed. Three possible workload patterns: low, medium, and high request/jobs are considered for experimental results. The number of jobs is created using random uniform distribution ranges from 100 to 1000 for low workload, 1000 to 6000 for medium workload, and 5000 to 12000 for high density workload. A Gaussian random number generator is used to produce the workload pattern. Table 3 summarises the workload categories of this complete simulation. Figure 7, Figure 8 and Figure 9 exhibits the results for energy consumption with various workload categories or number of jobs, which clearly indicates, CEEE and MaxUtil approaches has competent energy-saving capabilities when compared to Random algorithm. CEEE and MaxUtil outperformed Random approach by an average of 17.5% and 13.2%, respectively, regardless of migration adoption. While energy savings with lower number of jobs for Random algorithms is interesting as it is showing similar results when compared with CEEE and MaxUtil depicted in Figure 7. The workloads with higher number of jobs are better suited for task consolidation approaches such as MaxUtil and CEEE as shown in Figure 9. The experimental results shows that the advantage of migration was not immediately evident when the workload pattern has less jobs, as illustrated in Figure 7. This is mostly due to the fact that migrated activities often have short remaining processing periods, and these jobs are more likely to obstruct the consolidation of new arriving tasks, resulting in higher energy consumption than in the absence of migration.

Table 3. Workload categories for different VM.

Usage Pattern	**Workload**
Low	100-1000
Medium	1000-5000
High	5000-12000

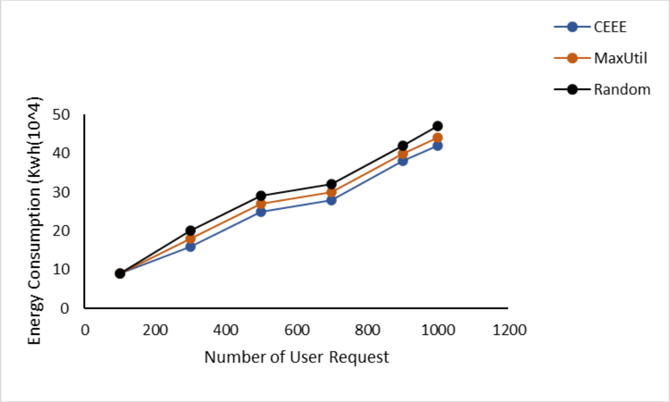

Figure 7. Energy saving with less jobs.

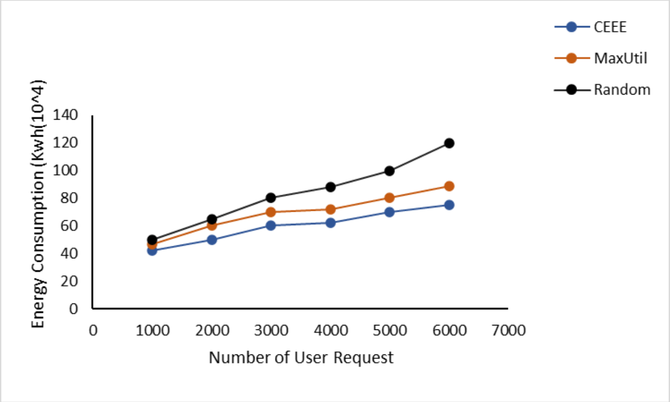

Figure 8. Energy saving with medium jobs.

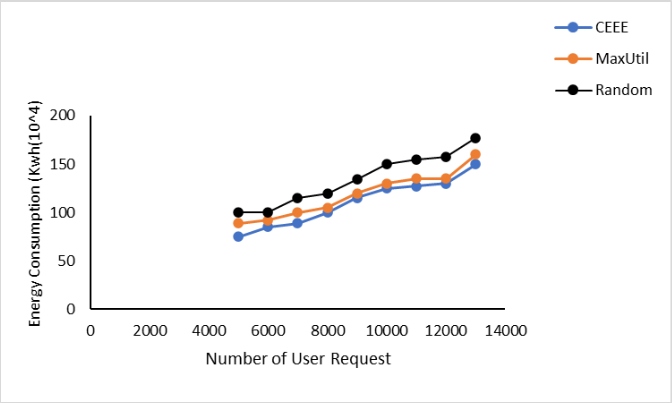

Figure 9. Energy saving with high job.

5.1.2 Number of Hosts in Sleep Mode

In Figure 10 the energy utilization is optimized by reducing the workload on a host. The workload is dependent on the total quantity of resources being used by Virtual machines. It is the ratio of resources occupied by entire set of VMs to the total number of resources in the region in percentage. As the jobs in the region declines, the resource requirement of VMs also decreases and hence VM with low utilization goes to sleep mode after migrating its jobs to another host. Since inter and intra region VM/ job migration is suggested in proposed algorithm, as a result CEEE algorithms is able to regulate the consumption of energy with varying workload. Moreover, VM/Job migration is considered at two levels: during the time of serving job and also at time of its placement on the host. Therefore, when the resource requirement increases or decreases at various time interval, CEEE decides when to send the host to sleep mode or send the wakeup call when required. The experimental results demonstrates that approximately 60% of host which are substantially large proportion can be in sleep mode using CEEE. While in MaxUtil algorithm approximately 50% and in Random approach only 40% of host can enter in sleep mode when the resource utilization is less than 50%. As the number of requests grows, more resources are needed to service them, resulting in fewer hosts available for sleep mode. Figure 10 illustrates that as resource usage approaches 95%. CEEE has the maximum number of hosts in sleep mode, which is marginally higher than MaxUtil and significantly more than Random algorithms.

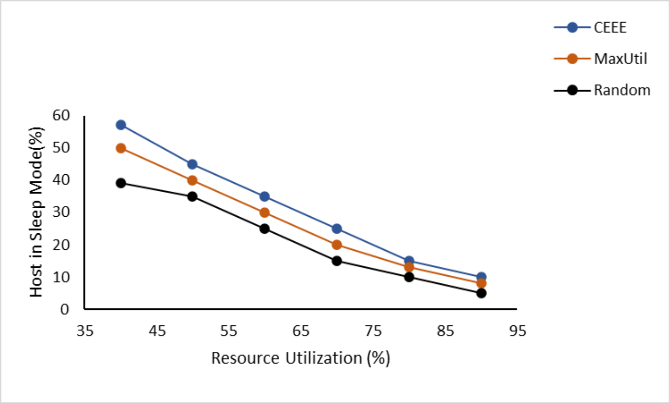

Figure 10. Percentage of hosts in sleep mode.

5.1.3 Resource Utilization

Resources in a diverse environment like cloud computing has multiple configurations, which is a disadvantage as resource availability is not assured. However, the benefit of the such environment is that lots of resources are used for completion of the task, which means maximum resources are used in this environment as compared to the homogeneous environment. In our experiment it is observed that as the workload in the system increases the resource utilization also increases as displayed in Figure 11. CEEE has utilized 98% of resources when the number of jobs is 11000 whereas MaxUtil has achieved approximately 96% accuracy when workload is 11000. This indicates that CEEE has effectively used the resources for the completion of the jobs. In terms of resource utilisation, the results showed that CEEE performed 15 percent better than Random and 2.5 percent better than MaxUtil.

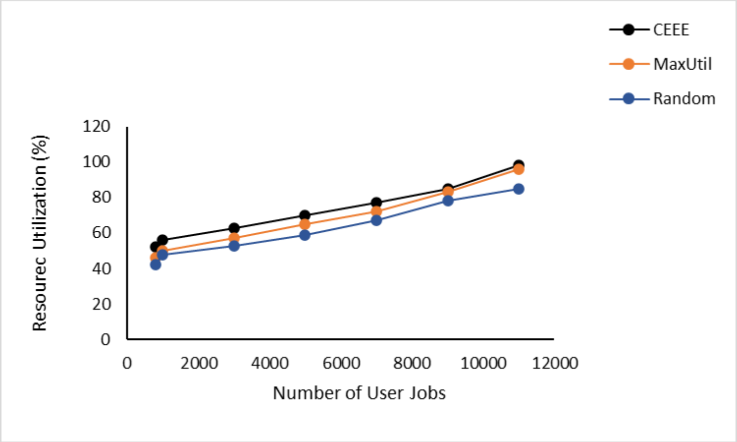

Figure 11. Resource utilization.

5.1.4 Completion Time

The amount of time necessary to accomplish the task is defined as completion time. The job is a collection of instructions that can be handled by a physical machine, which has a rate of handling the instructions known as the MIPS rating (Million Instructions per Second). Completion Time refers to the total time necessary to complete the execution of a jobs as well as the time spent by the job waiting for some resource. It is calculated as follows:

$$CT = \sum_{i=1}^{k}(T_i + W_i) \quad (12)$$

where, CT is completion time of the allocated job, W_i is the waiting time of job for availability of resources, T_i is execution time of the allocated job. Execution Time T_i denotes the total time it takes for machines to complete a task. The graph in Figure 12, suggests that the completion time of jobs is less in CEEE as compared to MaxUtil and Random Algorithm. The overall completion time MaxUtil with varying job pattern is less than the random algorithm. The Random algorithm time is marginal to the completion time of MaxUtil and CEEE when the number of jobs is between 1000-3000.

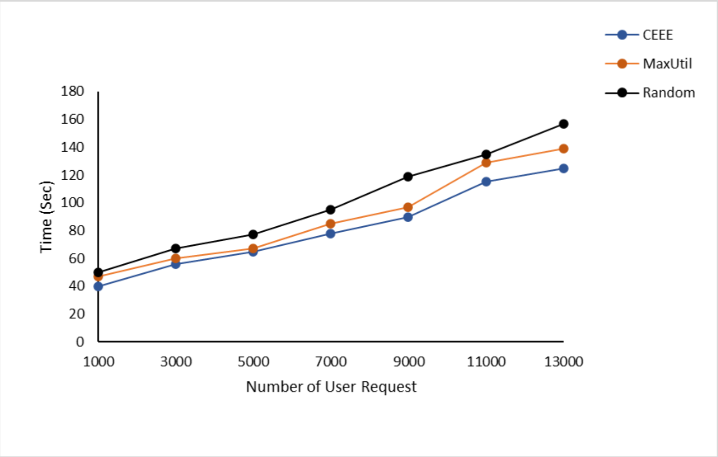

Figure 12. Completion time with varying jobs.

6. Conclusion and Future Work

In this paper, an agent-based cost-effective energy efficient algorithm (CEEE) has been proposed for saving the energy consumption of the network by consolidation of the jobs. CEEE is a two- fold algorithm, first it consolidates the jobs within the region by migrating the jobs to the neighbouring nodes. Secondly, it migrates the VM and jobs to another region when the workload on a specific region is low. The energy is conserved by reducing the number of active resources and putting the VMs in sleep mode in a systematic manner. The cost parameter of CEEE is developed based on the average resource utilization and the communication cost. The goal of cost estimation is to intensify the consolidation density of the resource. The density consolidation has its advantage as it primarily reduces energy consumption. An additional benefit of cost parameter is that it indirectly reduces the amount of active VMs since it tends to increase the employment of a smaller number of VMs.

In Future, to avoid performance deterioration, the proposed algorithm can be made more scalable by adding or removing virtual computers as needed. Additional resources (such as storage servers) can be considered to see how they affect energy use. To assure QoS for each VM, predefined memory requirement is considered in CEEE with limited number of VMs. For future prospects, it is likely to calculate the optimal number of VMs in a region, so as to increase the total processing requirement of the request. Moreover, other resources such as communication (network I/O) resources and secondary storage can also be considered in this decision making. Moreover, different methods should be provisioned for cooperation and consistency between different VM copies and failure recovery.

Conflict of Interest

Authors declare that there is no conflict of interest to declare for this publication.

Acknowledgements

The authors would like to express their sincere thanks to the editor and anonymous reviews for their time and valuable suggestions.

References

Barroso, L.A., & Hölzle, U. (2007). The case for energy-proportional computing. *Computer*, *40*(12), 33-37. https://doi.org/10.1016/j.future.2011.04.017.

Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, *28*(5), 755-768. https://doi.org/10.1016/j.future.2011.04.017.

Bilal, K., Khan, S.U., Zhang, L., Li, H., Hayat, K., Madani, S.A., Allah, N.M., Wang, L., Chen, D., Khan, M.I., Xu, C.-Z., & Zomaya, A.Y. (2013). Quantitative comparisons of the state-of-the-art data center architectures. *Concurrency and Computation: Practice and Experience*, *25*(12), 1771-1783.

Bohrer, P., Elnozahy, E.N., Keller, T., Kistler, M., Lefurgy, C., McDowell, C., & Rajamony, R. (2002). The case for power management in web servers. *Power Aware Computing* (pp. 261-289). Springer, Boston, MA.

Buyya, R., Calheiros, R.N., & Beloglazov, A. (2009). Cloudsim: A framework for modeling and simulation of cloud computing infrastructures and services. *The Cloud Computing and Distributed Systems (CLOUDS) Laboratory*. [Online]. [Accessed 18 May 2018].

Chen, X., Zhang, J., Lin, B., Chen, Z., Wolter, K., & Min, G. (2021). Energy-efficient offloading for DNN-based smart IoT systems in cloud-edge environments. *IEEE Transactions on Parallel and Distributed Systems*, *33*(3), 683-697. doi:10.1109/TPDS.2021.3100298.

Choi, H., Lim, J., Yu, H., & Lee, E. (2016). Task classification-based energy-aware consolidation in clouds. *Scientific Programming*, *2016*(1), 1-13. doi:https://doi.org/10.1155/2016/6208358.

Fan, X., Weber, W.D., & Barroso, L.A. (2007). Power provisioning for a warehouse-sized computer. *34th Annual International Symposium on Computer Architecture. 35*, pp. 13-23. New York, NY, United States: Association for Computing Machinery. doi:10.1145/1273440.1250665.

Gao, Y., Guan, H., Qi, Z., Wang, B., & Liu, L. (2013). Quality of service aware power management for virtualized data centers. *Journal of Systems Architecture*, *59*(4-5), 245-259. doi:https://doi.org/10.1016/j.sysarc.2013.03.007.

Gmach, D., Rolia, J., Cherkasova, L., Belrose, G., Turicchi, T., & Kemper, A. (2008). An integrated approach to resource pool management: Policies, efficiency and quality metrics. *International Conference on Dependable Systems and Networks With FTCS and DCC (DSN* (pp. 326-335). Anchorage, AK, USA: IEEE. doi: 10.1109/DSN.2008.4630101.

Hummaida, A.R., Paton, N.W., & Sakellariou, R. (2022). Dynamic threshold setting for VM migration. In *European Conference on Service-Oriented and Cloud Computing* (pp. 31-46). Springer, Cham. doi: https://doi.org/10.1007/978-3-031-04718-3_2.

Jiang, C., Wu, J., & Li, Z. (2019). Adaptive thresholds determination for saving cloud energy using three-way decisions. *Cluster Computing*, *24*(7), 8475-8482. https://doi.org/10.1007/s10586-018-1879-7.

Khemili, W., Hajlaoui, J.E., & Omri, M.N. (2022). Energy aware fuzzy approach for placement and consolidation in cloud data centers. *Journal of Parallel and Distributed Computing*, *161*, 130-142 doi: https://doi.org/10.1016/j.jpdc.2021.12.001.

Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N., & Jiang, G. (2009). Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing*, *12*(1), 1-5. doi:10.1109/ICAC.2008.31.

Lee, Y.C., & Zomaya, A.Y. (2012). Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, *60*(2), 268-280. doi:10.1007/s11227-010-0421-3.

Lien, C.H., Liu, M.F., Bai, Y.W., Lin, C.H., & Lin, M.B. (2006). Measurement by the software design for the power consumption of streaming media servers. *IEEE Instrumentation and Measurement Technology Conference Proceedings* (pp. 1597-1602). Sorrento, Italy: IEEE. doi:10.1109/IMTC.2006.328685.

Luo, L., Wu, W.J., & Zhang, F. (2014). Energy modeling based on cloud data center. *Journal of Software, 25*(7), 1371-1387. doi: 10.13328/j.cnki.jos.004604.

Mastelic, T., Oleksiak, A., Claussen, H., Brandic, I., Pierson, J.M., & Vasilakos, A.V. (2014). Cloud computing: survey on energy efficiency. *47*, 1-36. doi:10.1145/2656204.

Stavrinides, G.L., & Karatza, H.D. (2019). An energy-efficient, QoS-aware and cost-effective scheduling approach for real-time workflow applications in cloud computing systems utilizing DVFS and approximate computations. *Future Generation Computer Systems, 96*, 216-226.

Vashisht, P., & Kumar, V. (2020). Agent based optimized réplica management in data grids. *Investigación Operacional, 41*(2), 232-249.

Vashisht, P., Kumar, V., Kumar, R., & Sharma, A. (2019). Optimizing replica creation using agents in data grids. *In 2019 Amity International Conference on Artificial Intelligence (AICAI)* (pp. 542-547). Dubai: IEEE. doi:10.1109/AICAI.2019.8701244.

Vashisht, P., Kumar, V., Kumar, R., & Sharma, A. (2020). Optimization of replica consistency and conflict resolution in data grid environment. *International Journal of Mathematical, Engineering and Management Sciences, 4*(6), 1420-1433. doi:10.33889/IJMEMS.2019.4.6-112.