

# Bayes Theorem Based Virtual Machine Scheduling for Optimal Energy Consumption

R. Swathy\* and B. Vinayagasundaram

Computer Center, Madras Institute of Technology, Anna University, Chennai, 600044, India

\*Corresponding Author: R. Swathy. Email: swathy.balamurugan@mitindia.edu

Received: 17 September 2021; Accepted: 29 October 2021

**Abstract:** This paper proposes an algorithm for scheduling Virtual Machines (VM) with energy saving strategies in the physical servers of cloud data centers. Energy saving strategy along with a solution for productive resource utilization for VM deployment in cloud data centers is modeled by a combination of “Virtual Machine Scheduling using Bayes Theorem” algorithm (VMSBT) and Virtual Machine Migration (VMMIG) algorithm. It is shown that the overall data center’s consumption of energy is minimized with a combination of VMSBT algorithm and Virtual Machine Migration (VMMIG) algorithm. Virtual machine migration between the active physical servers in the data center is carried out at periodical intervals as and when a physical server is identified to be under-utilized. In VM scheduling, the optimal data centers are clustered using Bayes Theorem and VMs are scheduled to appropriate data center using the selection policy that identifies the cluster with lesser energy consumption. Clustering using Bayes rule minimizes the number of server choices for the selection policy. Application of Bayes theorem in clustering has enabled the proposed VMSBT algorithm to schedule the virtual machines on to the physical server with minimal execution time. The proposed algorithm is compared with other energy aware VM allocations algorithms viz. “Ant-Colony” optimization-based (ACO) allocation scheme and “min-min” scheduling algorithm. The experimental simulation results prove that the proposed combination of ‘VMSBT’ and ‘VMMIG’ algorithm outperforms other two strategies and is highly effective in scheduling VMs with reduced energy consumption by utilizing the existing resources productively and by minimizing the number of active servers at any given point of time.

**Keywords:** Energy saving strategy; VM scheduling; VM migration; Bayes theorem; resource utilization

## 1 Introduction

With the ever-increasing mobile devices and smart phones connected to the internet, Cloud computing has become the most flamboyant technology adopted by the industry in recent times. In the context of computational efficiency, cloud computing provides limitless services, processing power, memory, and storage at very affordable cost. As cloud-based solutions are increasing significantly, the need to utilize



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the data centers hosted in the cloud optimally also increases. Virtualization techniques and scheduling strategies are used to choose the apt physical servers in the data centers to host the virtual machines (VMs). The virtual machines with varying CPU cores, physical memory and storage requirements are created as per the request and hosted on a physical server in the data centers. The users will host their application on these virtual machines in the cloud [1,2]. Virtualization enables many virtual machines to be executed and share hardware resources on the same physical server. VM consolidation allows us to allocate the maximum number of VMs in a smaller number of physical servers [3]. The available computing power in a cloud is limited and any arbitrary usage of the resources will lead to poor performance in terms of scheduling and energy utilization. Identifying an optimal strategy to allocate VMs and still ensuring an increased utilization of physical servers in a data center is a challenging problem.

Energy efficiency for data centers hosted in the cloud is becoming more critical as the industry usage increases exponentially. Cloud data centers are observed to gobble energy when resources are indefinitely switched on even if they are not in use. About 70% of peak power is depleted by idle servers [4]. The idle server's power wastage is known to be a significant cause of over-utilization of energy there by increasing the overall operational and service cost. Moreover, usage of inefficient algorithms results in the allocation of a greater number of VMs to a single physical server than its capacity to handle exacerbating resource competition among the VMs and increasing the response time multiple times. Introducing energy aware scheduling with enhanced resource management strategy would significantly improve the computational efficiency of the cloud data centers and the associated cost of cloud-based solutions.

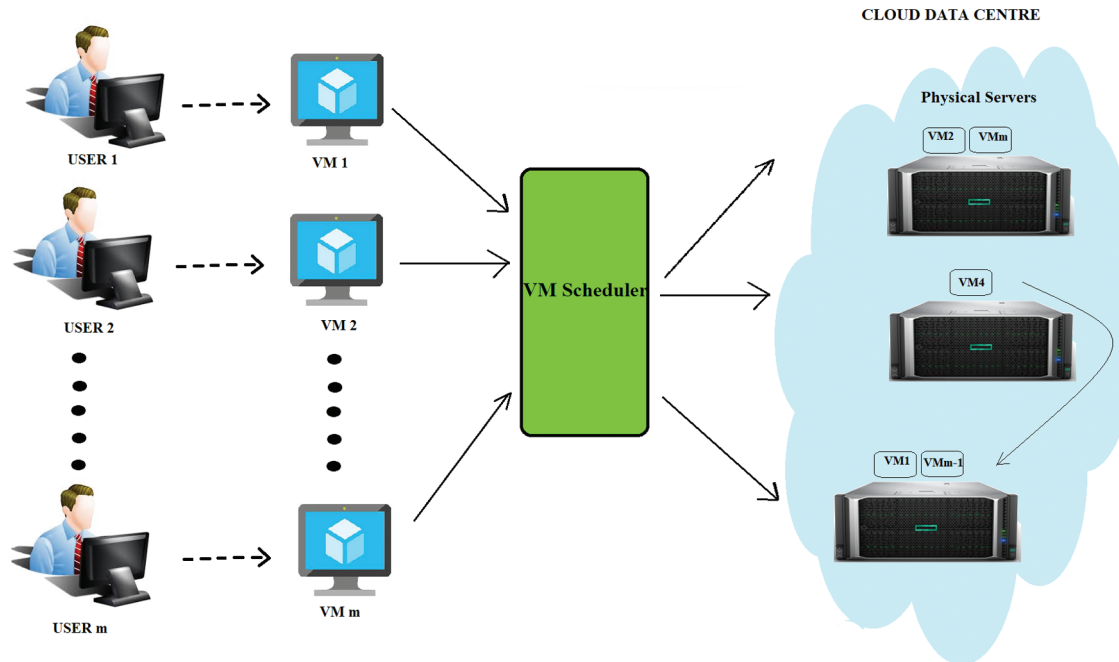
This work focuses on the solution to reduce the excessive energy utilized by the physical servers in the cloud data center by combining an energy aware VM scheduling algorithm and a VM migration procedure that is shown in Fig. 1. The proposed solution effectively brings down the energy consumed by under-utilized server by re-allocating the VMs to the physical servers that are hosting VMs much less than their maximum handling capacity. To optimize the power utilization of data centers the scheduling algorithms must be able to shut down idle servers and migrate VMs to under-utilized physical. The objective of this study and experiments revolves around improving the energy efficiency of physical servers by optimally assigning them the requested virtual machines and consolidating the virtual machines across several physical servers to maximize the resource utilization ratio. Our work proposes an energy aware VM scheduling with Bayes Theorem (VMSBT) algorithm and VM Migration algorithm (VMMIG). The VMSBT algorithm is used to strategically schedule VMs to the appropriate physical machine without over-loading it and the VMMIG algorithm reduces the number of active servers to service the given VM load as other VMs depart from its service. This scheduling algorithm combined with migration algorithm helps to reduce overall energy consumption in the data centers and improves the utilization of resources.

The proposed solution for optimal use of resource utilization and reduction of overall energy consumption of data centers consists of the following steps: (1) Clustering of optimal server set for VM allocation using Bayes theorem, (2) Allocation of VM without overloading the server using load consumption degree and energy consumption function and (3) Relocation of VMs from under-utilized server to other servers. Simulation of the proposed solution indicates that VMSBT has considerable energy saving capability compared to the Ant-Colony optimization-based scheduling and Min-Min scheduling procedures. The efficiency of the VMSBT and VMMIG algorithms is leveraged by the execution time taken for scheduling the VM to the appropriate server and relocating the VM from lightly loaded server to other active servers.

The analysis of the simulation results of the proposed solution demonstrates the following advantages:

- i) The clustering of servers using Bayes rule reduces the number of server choices for VM allocation.
- ii) Scheduling algorithm ensures that no physical servers are overloaded in terms of resources and energy consumption.

- iii) The migration strategy reduces the active physical server count by shutting down the idle and under-utilized servers.



**Figure 1:** VM scheduling in cloud data center

The contribution of the work is organized as follows. Section 2 summarizes all related works for energy efficient virtual machine placement. Section 3 describes identification of optimal physical servers for hosting virtual machines with Bayes theorem and discusses the strategy for Virtual Machine Migration in cloud data centers. Section 4 introduces the Virtual Machine Scheduling using Bayes Theorem (VMSBT) algorithm and Virtual Machine MIGration (VMMIG) algorithm. Section 5 presents the findings of the study of different methodologies for research. Finally, Section 6 presents the conclusion and potential scope of expansion of the current work.

## 2 Related Works

The principle of cloud computing scheduling refers to the technique of mapping a collection of jobs to a set of VMs or allocating VMs to operate on the available set of physical servers to meet the demands of users' [5]. In cloud computing, various scheduling techniques are used to improve system efficiency, load balancing, optimal resource utilization, energy conservation, decrease operational costs, and minimize the total processing time. The scheduler should therefore consider the VMs, and the constraints of user's requests together achieve efficient matching between jobs and resources [6]. Computing resources in a cloud environment is scheduled using two methods [7]: VM based scheduling and Host based scheduling. In the VM based scheduling, tasks are routed to the allocated VMs using a task/job scheduler for execution and this is called Task Scheduling. In the host-based scheduling, a VM scheduler is used at the host to assign the VMs into physical servers and this approach is termed VM Scheduling. The current research paper focuses on VM scheduling in cloud environments between various servers. In [8], the author proposes a solution that consolidates the tasks and the applications hosted on the data center so that they are moved to a limited number of physical servers and turn-off the unused servers to save

energy. Here the consolidation is accomplished at the task level whereas in the proposed research work consolidation is accomplished at the virtual machine (VM) level.

Reference [9] considers CPU availability and proposes a migration solution for dynamic reallocation of virtual machines. Here, the solution proposes an upper and lower threshold that determines the utilization rate of the processors in all the VMs. If the host's processor utilization rate reaches the upper threshold, some of the VMs are relocated and all the hosted VMs should be relocated if processor utilization rate is lesser than the lower threshold. Feller et al. [10] proposes ACO algorithm to reduce the number of physical servers to support the incoming requests. However, the experimental results show high computational cost as this approach consolidates all VMs in a single physical server only. In the proposed work, the consolidation of VMs is made based on various resource capacities *i.e.*, CPU, memory, and disk space.

Gao et al. [11] using Virtual Machine Placement with Ant Colony System (VMPACS) has aimed to reduce both consumption of energy and the waste of resources. Ant-colony based algorithm which achieves the pareto set has been proposed in this approach. Ibrahim et al. [12] proposes an adaptive genetic algorithm to improve both energy efficiency and the response time of the request. Raju et al. [13] proposed energy-aware multi objective chiropteran algorithm (EAMOCA), a hybrid cloud algorithm that aims at minimizing processing time and energy consumption while optimizing resource utilization. Most of the VM scheduling problems are solved by heuristic approaches like genetic algorithm, annealing algorithm, and ant-colony algorithm. There are several works that have used Meta-heuristics [14] to implement scheduling solutions that have very quick execution time. These Meta-heuristics approach uses strategic guess and repeated calculation to identify the best route among large solution space. Nimrod algorithm in Grid computing [15], genetic algorithms [16], ant colony optimization (ACO) and particle swarm optimization techniques (PSO) [17,18] are observed to be the optimal strategies for solving job shop scheduling problems.

Huge volume of data streaming from social networking services and IoT devices are hugely dependent on cloud computing platforms for processing and analytics. Large numbers of virtual machines are generated [19] in a cloud data center to analyze this data and several Physical Machines (PMs) are needed to manage it. The low density of VM deployed in traditional scheduling approaches is observed to waste significant PM resources. Energy efficiency of the whole cloud environment is significantly reduced by this under-utilization of physical resources. It is therefore important for VM scheduling to retain only active PMs by increasing the density of VMs allocated to PMs. Some of recent works in VM Scheduling with energy efficiency in cloud data centers have been given in following [Tab. 1](#).

**Table 1:** Energy efficient VM scheduling in cloud data center

Algorithm	Benefits	Parameter	Simulation environment
Ant colony optimization with particle swarm (ACOPS) [20]	Keeps the load balanced and converge fast	Makespan requests rejected execution time	C-simulated cloud environment with star topology
Order exchange and migration for ACS (OEMACS) [21]	Significant energy saving.	Average utilization of servers	C++ simulation
Knapsack algorithm [22]	Increased resource utilization	Resource utilization rate energy consumption rate	Cloudsim simulator

(Continued)

**Table 1 (continued)**

Algorithm	Benefits	Parameter	Simulation environment
LB-Min migration [23]	Reduced energy consumption with limited migrations	Lower bound for energy consumption Turning OFF PMs when IDLE Turning OFF PMs when IDLE with migration	Customized two-node cloud.
EASE framework [24]	Saves server power consumption and job completion time	Energy efficiency Energy proportionality	Custom simulation on CentOS 7.2 based VM
Genetic algorithm with tabu search algorithm (GATA) [25]	Maximize load balancing and achieve energy efficiency	Execution time, crossover rate, mutation rate, Chromosome length and Length of tabu table	Python

In this paper, a solution based on Bayes theorem for clustering physical servers is designed. VMs are mapped to appropriate physical servers (PS) using the parameters load consumption degree and energy consumption function. The allocation of VMs to PMs is accomplished by the VMSBT algorithm and migration of VMs is executed by VMMIG algorithm. The following section gives the detailed strategies used in VMSBT and VMMIG algorithms.

### 3 Proposed Work

#### 3.1 Problem Formulation

The scheduling problem of VMs to physical servers in cloud environment is formulated as follows: Consider there are ‘n’ physical servers for hosting ‘m’ virtual machines. PS denotes the set of ‘n’ physical server available at time ‘t’ and this is given as  $PS = \{S_1, S_2, S_3, S_4, \dots, S_n\}$ . Let VM be the set with ‘m’ number of virtual machines arriving at time t, that need to be hosted on the physical servers. The virtual machines are given as  $VM = \{V_1, V_2, V_3, V_4, \dots, V_m\}$ . Several research work on VM scheduling has considered only two computing resources *i.e.*, CPU and Primary memory or CPU and Secondary memory. In contrast to those approaches, the proposed scheduling solution utilizes multiple physical resources for hosting the virtual machines. The computing resources considered for hosting the virtual machines on a physical server are CPU, disk, and memory. Let  $S_i$  and  $V_j$  be the individual resource component of physical server and virtual machine. Each  $S_i$  in PS and  $V_j$  in VM comprises three resource components: CPU, disk, and memory.  $S_i$  is formed with three resource components  $\{S_c, S_d, S_m\}$ , similarly  $V_j$  is formed with resources  $\{V_c, V_d, V_m\}$ .  $\{S_c, S_d, S_m\}$  indicates the currently available CPU, disk, and memory resource of physical server  $P_i$  for hosting the virtual machine  $V_i$  which is requesting for resources  $\{V_c, V_d, V_m\}$ .

These individual resource components of PS and VM are combined into single component and a new set PS’ and VM’ are formed. The resources for a physical server  $S_i$  are combined as shown in Eq. (1); here  $\alpha, \beta$

and  $\gamma$  are the weight values determined by back propagation learning in a neural network. Similarly, for  $V_j$  the resources are grouped as in Eq. (2).

$$S_i = \alpha S_c^i + \beta S_d^i + \gamma S_m^i \quad \text{where } \alpha + \beta + \gamma = 1 \quad (1)$$

$$V_j = \alpha V_c^i + \beta V_d^i + \gamma V_m^i \quad \text{where } \alpha + \beta + \gamma = 1 \quad (2)$$

As a first step the favorable physical servers are clustered for virtual machine allocation using Bayes law. The Bayes law specifies how the probability of occurrence of an event has an impact due to a hypothetical new event given the condition that new event is expected to turn out to be true. Bayes rule is given as the probability of server  $S_i$  being selected to deploy the virtual machine  $V_j$ . This is termed as the posterior probability and is given as  $P(S_i/V_j)$ , the server  $S_i$  executing the virtual machine  $V_j$ .  $S_i$  denotes the physical server and  $V_j$  denotes the virtual machine. The deployment of virtual machines on different physical servers is carried out based on the resources requested by the virtual machines and the load available on the physical servers. The Bayes law for clustering the favorable physical server is given in Eq. (3).

$$P(S_i|V_j) = \frac{P(V_j|S_i) \times P(S_i)}{P(V_j)} \quad (3)$$

Thus, the posterior probability  $P(S_i/V_j)$  is dependent on the prior probability  $P(V_j/S_i)$ , the probability of  $V_j$  deployed on server  $S_i$ ,  $P(S_i)$ , denotes the availability of physical servers and  $P(V_j)$ , denotes the virtual machine's probability. The prior probability is determined by Eq. (5).

$$P(V_j|S_i) = 1 - \frac{\text{Resource requested by VM}}{\text{Resource available at PS}} \quad (4)$$

$$P(V_j|S_i) = 1 - \frac{V_j}{S_i} \quad (5)$$

The probability of the number of physical servers available, is given as  $P(S_i)$  in Eq. (6). 'n' is the number of physical servers available.

$$P(S_i) = \frac{1}{n} \quad (6)$$

The probability of virtual machine  $P(V_j)$  is given as in Eq. (7). Certain physical servers cannot accommodate the virtual machines due to inadequate resource availability and a threshold is set for every physical server to identify the appropriate servers.

$$P(V_j) = \sum_{i=1}^n P(V_j|S_i) \times P(S_i) \quad (7)$$

The physical servers whose posterior probability is greater than the given threshold is clustered into a new set. Here, Bayes theorem helps to eliminate the set of physical servers that cannot accommodate the virtual hosts and cluster only the most appropriate servers. In this way Bayes law minimizes the number of server choices in VM scheduling. Following the clustering process, the virtual machines are deployed among the clustered servers based on the load consumption degree and the energy consumption degree as explained below.

### 3.2 Load Consumption Degree

The load consumption degree of the physical server depends on the utilization of CPU ( $uc_i$ ), disk ( $ud_i$ ) and memory ( $um_i$ ), if the virtual machine  $V_j$  is allocated to the physical server and the threshold value. A



maximum CPU ( $T_c$ ), disk ( $T_d$ ) and memory ( $T_m$ ) threshold is fixed for every physical server in the data center. These thresholds help to prevent over utilization of any physical server. The load consumption degree also helps to place the virtual machine on the optimal physical server correctly. The load consumption degree for the server is given by Eq. (8).

$$LC_i = \sqrt{(uc_i - T_c)^2 + (ud_i - T_d)^2 + (um_i - T_m)^2} \quad (8)$$

In the above equation utilization of CPU, disk and memory is determined using the Eqs. (9)–(11). In Eq. (12),  $B_{ij}$  represents binary value, if virtual machine  $j$  is placed on physical server  $i$ , the binary value is set as 1 else its set to 0.

$$sc_i = \sum_{j=1}^m B_{ij}c_j \text{ and } uc_i = \alpha(sc_i + V_c) \quad (9)$$

$$sd_i = \sum_{j=1}^m B_{ij}d_j \text{ and } ud_i = \beta(sd_i + V_d) \quad (10)$$

$$sm_i = \sum_{j=1}^m B_{ij}m_j \text{ and } um_i = \gamma(sm_i + V_m) \quad (11)$$

$$B_{ij} \rightarrow \begin{cases} 1, & \text{if } j\text{th VM placed on } i\text{th PM} \\ 0, & \text{no VM placed otherwise} \end{cases} \quad (12)$$

### 3.3 Energy Consumption Degree

Calculation of the energy used by  $i^{\text{th}}$  physical host is calculated as given in Eq. (13)

$$EC_i = ec_i^{\text{idle}} + (ec_i^{\text{max}} - ec_i^{\text{idle}})UR_i \quad (13)$$

where load utilization  $LU_i$  is shown in Eq. (14)

$$LU_i = uc_i + ud_i + um_i \quad (14)$$

And the utilization rate is given in Eq. (15)

$$UR_i = \frac{LU_i}{S_i} \times 100 \quad (15)$$

The batch of virtual machine requests are optimally allotted to the physical server using the VMSBT algorithm. The VM requests are placed with specific CPU, disk, and memory resource requirements. Similarly, the server has CPU, disk, and memory resources available. Those resources are consolidated using Eq. (13) for both virtual machines and physical servers. Then, Bayes rule is applied for every virtual machine and for every physical server. Based on the posterior probability obtained using Bayes rule, the physical servers with sufficient resources to accommodate virtual machines are clustered as optimal physical servers. The load consumption degree and energy consumption degree for optimal servers are calculated using Eqs. (14) and (15) to determine if the virtual machine can be hosted on those specific physical servers. Finally, the virtual machine is deployed on to the server with minimal load consumption degree and with minimal energy consumption degree. The above process gets repeated for all the virtual machines until they get allotted to a physical server.

## 4 Algorithms

### 4.1 VMSBT Algorithm and Description

---

**Algorithm:** VMSBT

---

Input: N-PS with available resource  $\{S_c, S_d, S_m\}$ , M-VM with requested resource  $\{V_c, V_d, V_m\}$

1. PS\_OPT =  $\varphi$
2. for i in PS
  - 2.1. Calculate  $S_i = \alpha S_c^i + \beta S_d^i + \gamma S_m^i$
3. End for
4. For j in VM
  - 4.1. Calculate  $V_j = \alpha V_c^j + \beta V_d^j + \gamma V_m^j$
5. End for
6. For j in VM
  - 6.1. For i in PS
    - 6.1.1. Calculate  $P(S_i|V_j) = \frac{P(V_j|S_i) \times P(S_i)}{P(V_j)}$
    - 6.1.2. If  $P(S_i|V_j) > threshold$ 
      - 6.1.2.1. Add PS<sub>i</sub> to OPT\_PS<sub>i</sub>
    - 6.1.3. End if
  - 6.2. End for
  - 6.3. For i in OPT\_PS<sub>i</sub>
    - 6.3.1. (Assumption what if VM<sub>j</sub> gets deployed in PS<sub>i</sub>)
    - 6.3.2. Calculate:
    - 6.3.3.  $sc_i = \sum_{j=1}^m B_{ij}c_j$  and  $uc_i = \alpha(sc_i + V_c)$
    - 6.3.4.  $sd_i = \sum_{j=1}^m B_{ij}d_j$  and  $ud_i = \beta(sd_i + V_d)$
    - 6.3.5.  $sm_i = \sum_{j=1}^m B_{ij}m_j$  and  $um_i = \gamma(sm_i + V_m)$
    - 6.3.6.  $LU_i = uc_i + ud_i + um_i$
    - 6.3.7.  $UR_i = \frac{LU_i}{S_i} \times 100$
    - 6.3.8.  $LC_i = \sqrt{(uc_i - T_c)^2 + (ud_i - T_d)^2 + (um_i - T_m)^2}$
    - 6.3.9.  $EC_i = ec_i^{idle} + (ec_i^{max} - ec_i^{idle})UR_i$
  - 6.4. End for
  - 6.5. Find MIN LC<sub>i</sub> in OPT\_PS<sub>i</sub>

---

(Continued)



---

**Algorithm (continued)**

---

- 6.6. If  $EC_i < ET$  for MIN  $LC_i$ 
    - 6.6.1. Allocate:  $PS_i \leftarrow VM_j$
    - 6.6.2. Update:  $PS_i$  with  $C_i \leftarrow C_i + V_j$
    - 6.6.3. Break
  - 6.7. Else
    - 6.7.1. Find Next MIN  $LC_i$  and repeat step 6.6
  - 6.8. End if
  7. End for
- 

Certain virtual machines hosted on a physical server remains idle after completing the jobs assigned to it by the client applications. These physical servers where a few virtual machines remain idle among the set of virtual machines hosted, goes to an under-utilized state and at the same time consume power considerably. The collective power consumption of many such under-utilized physical servers in the data center is significantly high and is one of the main causes for higher power utilization of the data centers. To solve this problem, the VMMIG algorithm is executed at periodic intervals which will identify the under-utilized physical servers whose resource utilization is below 20% of the available resource and relocate the virtual machines to another physical server.

**4.2 VMMIG Algorithm and Description**

---

VMMIG Algorithm

---

1. For  $i$  in PS
    - 1.1. If  $C_i \leq ST$ 
      - 1.1.1. Select  $V_j$  and  $P_i$
      - 1.1.2. Add  $VMMIG \leftarrow V_j$
  2. Add POOF  $\leftarrow P_i$ 
    - 2.1. End if
  3. End for
  4. VMSBT (N-POOF Server, VMMIG)
  5. Mark all servers in POOF as SHUTDOWN
- 

**4.3 Complexity Analysis of VMSBT Algorithm**

Time complexity analysis is done based on the number of inputs provided. Let us assume there are 'n' number of VM requested submitted for scheduling and 'm' be the number of physical servers available for VM allocation. Worst-case behavior of the algorithm depends on the number of jobs submitted and the number of iterations done. The combined resource components for 'm' number physical server are computed and the combined resource components for 'n' number of virtual machines. Then the probability to cluster the eligible physical hosts, and for computing resource utilization and energy

consumption for the eligible physical host executes  $c$  instructions for  $m$  times for all ‘ $n$ ’ number of VMs. Hence  $f(n)$  of the algorithm is given  $n + m + n*(2*c*m)$  where 2 and  $c$  are constants. By dropping down the terms which makes negligible changes as  $n$  and  $m$  grows larger, the function is kept as  $f(n) = n*m$ . Hence the worst-case behavior is given as  $O(f(n))$  which is  $O(n*m)$  and is quadratic.

## 5 Results and Discussion

### 5.1 Simulation Environment

All the experiments for the proposed work were implemented using a Cloudsim simulator running on a stand-alone PC with i7 Extreme processor, 64 GB RAM and 2 TB SSD storage capacities. The proposed VMSBT algorithm, ACO and min-min algorithms were all simulated in the Cloudsim environment, executed with various test parameters and results have been documented. Around 800 VMs requests are created with varying CPU, disk, and memory requirements. These VM requests are sent in batches with each batch containing 100 to 800 VM requests. The proposed solution has been simulated in the Cloudsim environment through several iterations with each iteration having a minimum of 10 batch requests. Each physical server is simulated to have 12000MIPS of CPU usage and 2 TB of primary memory. The data center of the cloud environment was deployed with 1000 servers with CPU capacity varying up to 24 cores, RAM up to 100 GB and hard disk size up to 2 TB as shown in [Tab. 2](#). Energy range of the server is set between the range [1,4] KW. Each server is fixed with the constraint, that the server resource utilization does not exceed 85% to prevent overloading the server. A subset of the experimental results of different parameters like execution time, resource utilization, energy consumption, no of active physical servers with and without migrations are shown in [Tabs. 3 and 4](#).

**Table 2:** Sample VM request

VM request	No of CPU Cores	Disk capacity in GB	Memory in GB
VM1	2	420	4
VM2	4	1200	16
VM3	2	240	2
VM4	8	1200	16
VM5	1	180	2
VM6	16	600	8

**Table 3:** Execution time, resource utilization, energy consumed, and number of active physical servers

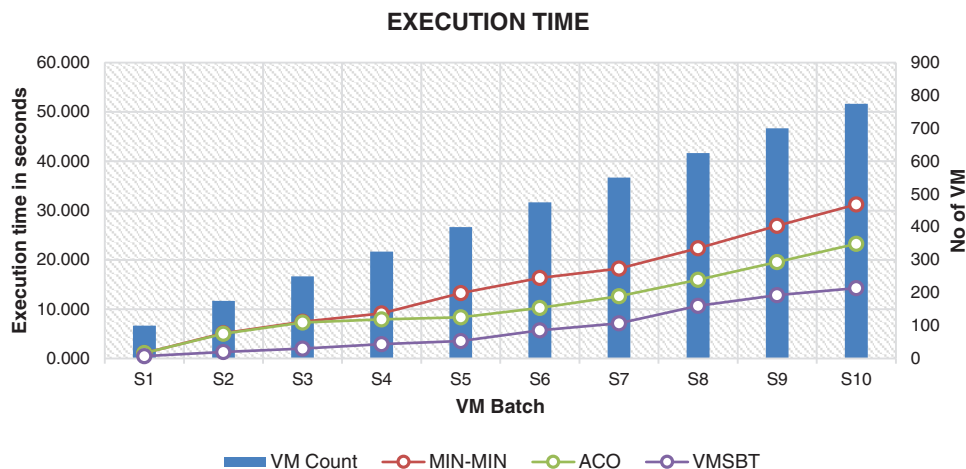
VM batch	No of VM	No of VM migrated	Execution time in seconds			Resource utilization in percentage			Energy consumed in KW/H			No of active physical servers		
			MIN-MIN	ACO	VMSBT	MIN-MIN	ACO	VMSBT	MIN-MIN	ACO	VMSBT	MIN-MIN	ACO	VMSBT
S1	100	25	1.113	1.104	0.451	23.6	24.1	23.5	2.72	2.55	2.21	16	15	13
S2	175	35	5.142	5.015	1.329	31.9	32.5	32.3	2.89	2.72	2.89	17	16	17
S3	250	50	7.473	7.306	2.003	37.5	37.9	38.4	3.91	4.42	3.91	23	26	23
S4	325	70	9.125	7.912	2.888	39.2	40.1	43.5	5.27	6.63	4.76	31	39	28
S5	400	85	13.256	8.357	3.536	43.5	45.2	51.8	7.14	8.16	5.61	42	48	33

**Table 4:** Number of active physical servers and energy consumption with and without migration

VM REQUEST BATCHES	No of VMs	No of active physical servers		Energy consumption in KW/H	
		VMSBT-with migration	VMSBT-without migration	Energy consumed-with migration	Energy consumed-without migration
S1	100	13	15	2.21	2.55
S2	175	17	19	2.89	3.23
S3	250	23	26	3.91	4.42
S4	325	28	32	4.76	5.44
S5	400	33	39	5.61	6.63

### 5.2 Execution Time of the Scheduling Algorithm

Comparison of execution times of VMSBT scheduling, Ant-Colony based scheduling and min-min scheduling algorithms showed that VMSBT scheduling algorithms had minimum execution time with an average of 6 s compared to the other two scheduling algorithms. VMSBT scheduling algorithm reduces the number of searches to identify the physical server to host by clustering out the optimal physical server using Bayes rule. The algorithm ensures that only the servers with sufficient computing resources are aggregated using Bayes rule and the virtual machines are mapped to optimal servers based on resource utilization and energy consumption. The results plotted in Fig. 2 shows the execution time for all the three scheduling algorithms.

**Figure 2:** Execution time in seconds

### 5.3 Productive Resource Utilization

In VMSBT scheduling procedure, the correct virtual machines are assigned to the optimal physical servers using the load consumption formula given in Eq. (8). It is observed highest resource utilization of VMSBT is found to be increased by 91% compared to ACO-based scheduling which is 67% and min-min scheduling which is 69%. The highest resource utilization shows that VMSBT optimally uses the resources in the physical servers where the VMs are allocated. Fig. 3 below shows the resource utilization trend among the three algorithms for 10 different batch requests with the number of VMs in each batch ranging from 100 to 800.

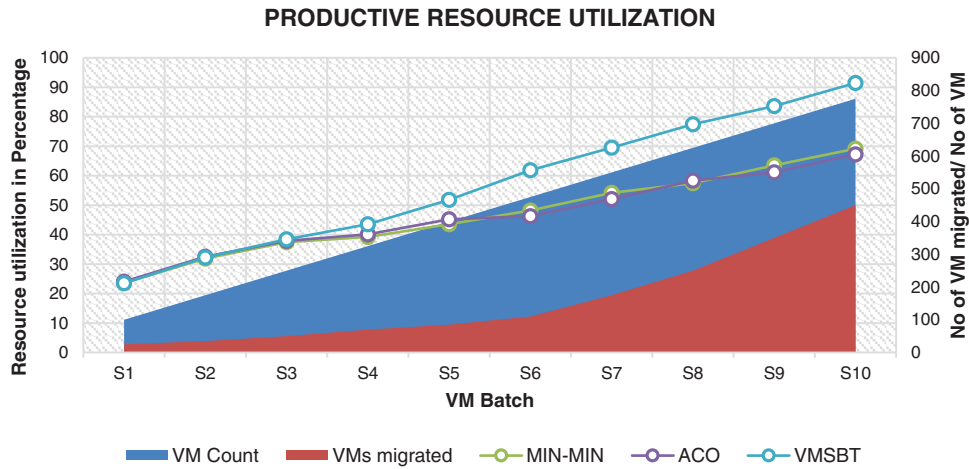


Figure 3: Productive resource utilization

5.4 Amount of Energy Consumed

The VMSBT and VMMIG algorithm has been proposed to reduce energy consumption and save energy in the data center as one of its core objectives. In the simulation environment a physical server running for 60 m is set to consume 0.17 KW of energy and the energy consumption for all three algorithms were recorded. The results show that the VMSBT algorithm with VMMIG on an average has consumed 6 KW of energy while the MIN-MIN and ACO has used 8 and 9.2 KW of energy for the same count of virtual machine requests. For every physical machine an upper threshold limit and lower threshold limit for resource utilization has been setup to 35% and 85% respectively. The upper threshold limit ensures that no physical server is being overloaded. The lower resource threshold limit is used to identify virtual machines for migration. This process of migration is carried out by the VMMIG algorithm and optimal placement of the virtual machine on physical server is carried out by the VMSBT algorithm. After the migration the physical server is shut down and as a result this reduces the energy consumption of the server. The Amount of Energy Consumed plotted as a graph in Fig. 4.

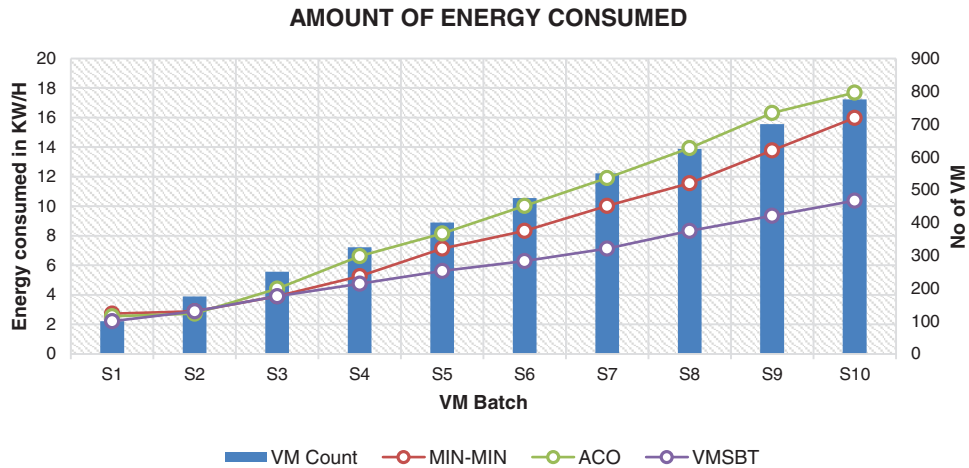


Figure 4: Amount of energy consumed in KW/H

### 5.5 Number of Active Physical Servers by VMSBT, ACO Based and Min-Min Scheduling Algorithm

This measure indicates the total number of physical servers utilized for placing the virtual servers requested in each batch. The result shown in Fig. 5 indicate that VMSBT algorithm outperforms ACO based and min-min scheduling algorithm. The simulation environment has been set with 120 physical servers and VM requests were submitted in batches ranging from 100 to 800 VMs per batch. ACO based and min-min scheduling algorithm often uses all available physical servers while VMSBT scheduling algorithm manages to use only 50% of the available physical servers even in the peak load condition where around 775 virtual machines were requested.

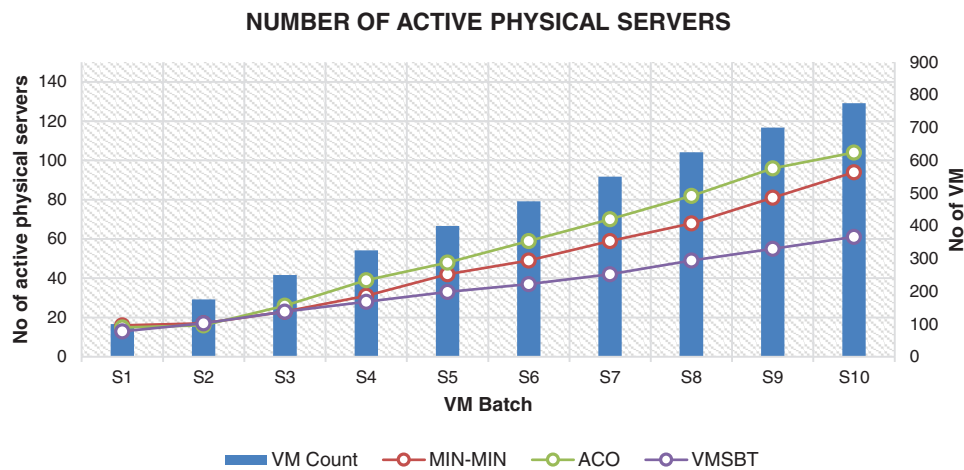


Figure 5: Number of active physical servers

### 5.6 Amount of Active Physical Servers Using VMSBT Algorithm with and Without VMMIG Algorithm

Fig. 6 shows the number of active physical servers used when scheduling is done by VMSBT with and without migration algorithm. When VMSBT scheduling algorithm is combined with VMMIG algorithm, it increases the number of unused physical servers by 14% to 25% compared to running the VMSBT algorithm without VMMIG migration algorithm. Also, the increase in the number of unused servers reduces the energy consumption and saves power as explained in the above results section. The results tabulated below shows that the VMSBT algorithm when used with VMMIG algorithms has resulted in claiming 22% of underutilized servers and thereby reducing the overall energy consumption of the data center as well.

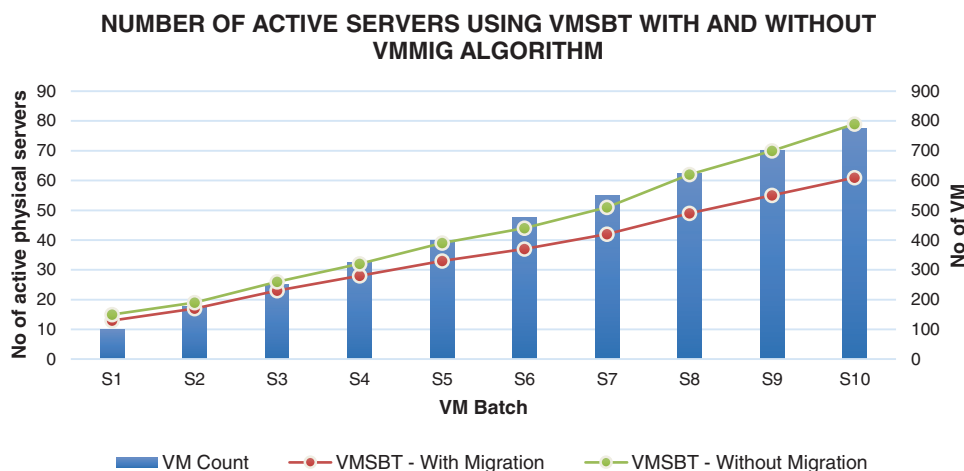
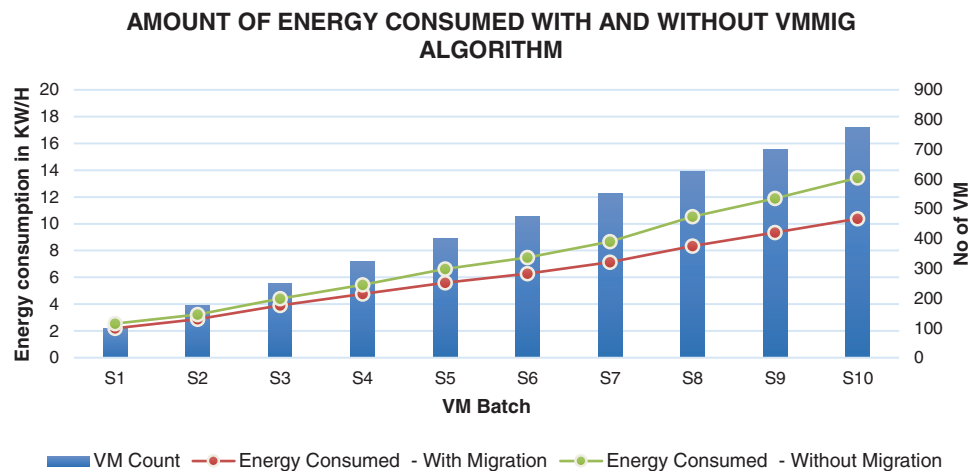


Figure 6: Number of active physical servers with and without migration

### 5.7 Amount of Energy Consumed by VMSBT Algorithm with and Without VMMIG Algorithm

VMSBT algorithm optimally places all virtual machines on to appropriate physical servers by migrating some of the VMs to underutilized physical servers in the data center. During this process, it ensures that no server gets overloaded and also the VMs are placed on the server with sufficient number of resources. Over the period, few virtual machines leave the server as it finishes its work. As a result, some of the physical server functions with its resources are underutilized. Hence, the VMs running on those physical servers are consolidated and migrated to other servers using VMMIG algorithm. Several iterations of the experiment and the results indicate that the power consumption is reduced by 10% to 25% when VMSBT is used with the VMMIG algorithm which is illustrated as graph in Fig. 7.



**Figure 7:** Amount of energy consumed with and without migration

## 6 Conclusions and Future Work

This work proposed an optimal solution called VMSBT scheduling algorithm to deploy virtual machines on an appropriate physical server. VMSBT can also save energy while scheduling VM requests which is supported with the VMMIG algorithm. Many historic works have considered only two types of resources like memory and disk, in contrast to the current scheduling process which has considered three different computing resources like CPU, disk and memory. The physical server's existing workload is analyzed before placing the virtual machine which prevents overloading of the servers. The optimal physical servers for VM scheduling is identified with Bayes theorem. This method of clustering out the optimal servers helps to reduce the computation time as well as prevents scheduling virtual machines to an overloaded server. Virtual machine is mapped to its appropriate server based on the load availability and energy consumption model.

Simulation results has shown that VMSBT algorithm works better compared to traditional ACO and MIN-MIN scheduling algorithms. VMSBT always ensures that the physical machines are not being overloaded while mapping the virtual machines, improves the utilization rate of the resources and saves energy with the support of VMMIG algorithm. The results of the proposed algorithms reveal that the overall server utilization rate has improved by 50% and overall, the total energy saving has improved by 10% to 25%.

For future work, network related problems during migration like bandwidth, latency, congestion, error rate that can impact VMSBT and VMMIG algorithms can be studied in detail. After evaluation of algorithm's efficiency in the simulated environment including the above-mentioned network parameters, the solution can be taken to real time cloud environment.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] Z. G. Chen, Z. H. Zhan, H. H. Li, K. J. Du, J. H. Zhong *et al.*, “Deadline constrained cloud computing resources scheduling through an ant colony system approach,” in *Int. Conf. on Cloud Computing Research and Innovation (ICCCRI)*, IEEE, Singapore, pp. 112–119, 2015.
- [2] H. H. Li, Z. G. Chen, Z. H. Zhan, K. J. Du and J. Zhang, “Renumber coevolutionary multiswarm particle swarm optimization for multi-objective workflow scheduling on cloud computing environment,” in *Proc. of the Companion Publication of the 2015 Annual Conf. on Genetic and Evolutionary Computation*, Madrid, Spain, pp. 1419–1420, 2015.
- [3] C. Mastroianni, M. Meo and G. Papuzzo, “Probabilistic consolidation of virtual machines in self-organizing cloud data centers,” *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 215–228, 2013.
- [4] E. Naone, “Conjuring clouds,” *Technology Review*, vol. 112, no. 4, pp. 54–56, 2009.
- [5] V. Manglani, A. Jain and V. Prasad, “Task scheduling in cloud computing,” *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, pp. 821–825, 2017.
- [6] L. Liu and Z. Qiu, “A survey on virtual machine scheduling in cloud computing,” in *2nd IEEE Int. Conf. on Computer and Communications (ICCC)*, Chengdu, China, pp. 2717–2721, 2016.
- [7] E. Pacini, C. Mateos and C. G. Garino, “Multi-objective swarm intelligence schedulers for online scientific clouds,” *Computing*, vol. 98, no. 5, pp. 495–522, 2016.
- [8] S. Srikantaiah, A. Kansal and F. Zhao, “Energy aware consolidation for cloud computing,” in *Proc. of the 2008 Conf. on Power Aware Computing and Systems, Ser. HotPower ’08*, San Diego, California, pp. 10, 2008.
- [9] A. Beloglazov and R. Buyya, “Energy efficient resource management in virtualized cloud data centers,” in *Proc. of 10th IEEE/ACM Int. Conf. on Cluster, Cloud and Grid Computing, ser. CCGRID ’10*, pp. 826–831, 2010.
- [10] E. Feller, L. Rilling and C. Morin, “Energy-aware ant colony-based workload placement in clouds,” in *IEEE/ACM Int. Conf. of Grid Computing, IEEE*, Lyon, France, pp. 26–33, 2011.
- [11] Y. Gao, H. Guan, Z. Qi, Y. Hou and L. Liu, “A multi-objective ant colony system algorithm for virtual machine placement in cloud computing,” *Journal of Computer and System Sciences*, vol. 79, no. 8, pp. 1230–1242, 2013.
- [12] H. Ibrahim, R. O. Aburukba and K. El-Fakih, “An integer linear programming model and adaptive genetic algorithm approach to minimize energy consumption of cloud computing data centers,” *Computers & Electrical Engineering*, vol. 67, no. 8, pp. 551–565, 2018.
- [13] R. Raju, J. Amudhavel, N. Kannan and M. Monisha, “A bio inspired energy-aware multi objective chiropteran algorithm (EAMOCA) for hybrid cloud computing environment,” in *Int. Conf. of Green Computing Communication and Electrical Engineering (ICGCCEE)*, Coimbatore, India, pp. 1–5, 2014.
- [14] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison,” *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003.
- [15] R. Buyya, D. Abramson and J. Giddy, “Nimrod/g: An architecture for a resource management and scheduling system in a global computational grid,” in *Proc. of the Int. Conf./Exhibition on High Performance Computing in the Asia-Pacific Region*, Beijing, China, vol. 1, pp. 283–289, 2000.
- [16] J. Cao, D. Spooner, S. Jarvis, S. Saini and G. R. Nudd, “Agent-based grid load balancing using performance-driven task scheduling,” in *Proc. of the Int. Parallel and Distributed Processing Symp., IEEE*, Nice, France, pp. 10, 2003.
- [17] B. Liu, L. Wang and Y. H. Jin, “An effective PSO-based memetic algorithm for flow shop scheduling,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 1, pp. 18–27, 2007.
- [18] R. Poli, J. Kennedy and T. Blackwell, “Particle swarm optimization,” *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.



- [19] H. Mi, H. Wang, G. Yin, Y. Zhou, D. Shi *et al.*, “Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers,” in *IEEE Int. Conf. on Services Computing*, Miami, Florida, pp. 514–521, 2010.
- [20] K. M. Cho, P. W. Tsai, C. W. Tsai and C. S. Yang, “A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing,” *Neural Computing and Applications*, vol. 26, no. 6, pp. 1297–1309, 2015.
- [21] X. F. Liu, Z. H. Zhan, J. D. Deng, Y. Li, T. Gu *et al.*, “An energy efficient ant colony system for virtual machine placement in cloud computing,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 113–128, 2018.
- [22] S. Han, S. Min and H. Lee, “Energy efficient VM scheduling for big data processing in cloud computing environments,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 42, no. 4, pp. 1–10, 2019.
- [23] P. Kuang, W. Guo, X. Xu, H. Li, W. Tian *et al.*, “Analyzing energy-efficiency of two scheduling policies in compute-intensive applications on cloud,” *IEEE Access*, vol. 6, pp. 45515–45526, 2018.
- [24] Y. Qiu, C. Jiang, Y. Wang, D. Ou, Y. Li *et al.*, “Energy aware virtual machine scheduling in data centers,” *Energies*, vol. 12, no. 4, pp. 646, 2019.
- [25] D. M. Zhao, J. T. Zhou and K. Li, “An energy-aware algorithm for virtual machine placement in cloud computing,” *IEEE Access*, vol. 7, pp. 55659–55668, 2019.