

Energy and Service Level Agreement Aware Resource Allocation Heuristics for Cloud Data Centers

K.Sutha¹ and Dr.G.M.Kadhar Nawaz²

¹Research Scholar, Computer Science, Research and Development Centre,
Bharathiar University, Coimbatore - 641046, Tamilnadu, India
Assistant Professor, Department of Computer Applications,
Dr.MGR Janaki College of Arts and Science for Women, Chennai - 600028
[e-mail: ksutha1986@gmail.com]

²Professor & Director, Department of Master of Computer Applications,
Sona College of Technology, Salem - 636005, Tamilnadu, India
[e-mail: nawazse@yahoo.co.in]

*Corresponding author: K.Sutha

*Received January 6, 2018; revised April 17, 2018; accepted May 21, 2018;
published November 30, 2018*

Abstract

Cloud computing offers a wide range of on-demand resources over the internet. Utility-based resource allocation in cloud data centers significantly increases the number of cloud users. Heavy usage of cloud data center encounters many problems such as sacrificing system performance, increasing operational cost and high-energy consumption. Therefore, the result of the system damages the environment extremely due to heavy carbon (CO₂) emission. However, dynamic allocation of energy-efficient resources in cloud data centers overcomes these problems. In this paper, we have proposed Energy and Service Level Agreement (SLA) Aware Resource Allocation Heuristic Algorithms. These algorithms are essential for reducing power consumption and SLA violation without diminishing the performance and Quality-of-Service (QoS) in cloud data centers. Our proposed model is organized as follows: a) SLA violation detection model is used to prevent Virtual Machines (VMs) from overloaded and underloaded host usage; b) for reducing power consumption of VMs, we have introduced Enhanced minPower and maxUtilization (EMPMU) VM migration policy; and c) efficient utilization of cloud resources and VM placement are achieved using SLA-aware Modified Best Fit Decreasing (MBFD) algorithm. We have validated our test results using CloudSim toolkit 3.0.3. Finally, experimental results have shown better resource utilization, reduced energy consumption and SLA violation in heterogeneous dynamic cloud environment.

Keywords: Cloud computing, Energy consumption, SLA violation, Resource utilization, Modified Best Fit Decreasing, Heuristic algorithm

1. Introduction

Cloud computing is a growing service-oriented paradigm that enables on-demand self services for competitive IT industries. It enumerates services like Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Such services are deployed over the internet and are utilized based on users' demands. The main objective of cloud computing is to focus on reducing operational cost and increasing productivity. Meanwhile, it shares pool of ubiquitous resources managed by minimum management efforts. While eliminating capital expenditure, Cloud Service Providers (CSP) are forced to ensure availability of high-speed resource sharing through on-demand users' requirement. Therefore, efficient utilization of cloud resources is the most significant research area in cloud computing. However, as the number of cloud users increases, obviously performance and quality of services are sacrificed.

Information and Communication Technology (ICT) is one of the highest energy consuming and CO₂ emitting industries. The total energy consumption of the United States in December 2017 was 97,827 Trillion Btu (British thermal unit) [1], which includes Residential 19,996TBtu, Commercial 18,067TBtu, Industrial 31,560TBtu and Transportation 28,199TBtu. Out of these four sectors, industries are the highest energy consumer with reducing Return on Investment (ROI). Fig. 1 illustrates the total energy consumption from 1950s to 2017 and the report shows that energy consumption by the industrial sector in 1950 was 16,241TBtu and 2017 was 31,560TBtu. The results concerning industrial sector clearly demonstrates that the values are doubles. Hence, cloud providers are forced to focus on energy consumption without compromising quality of services.

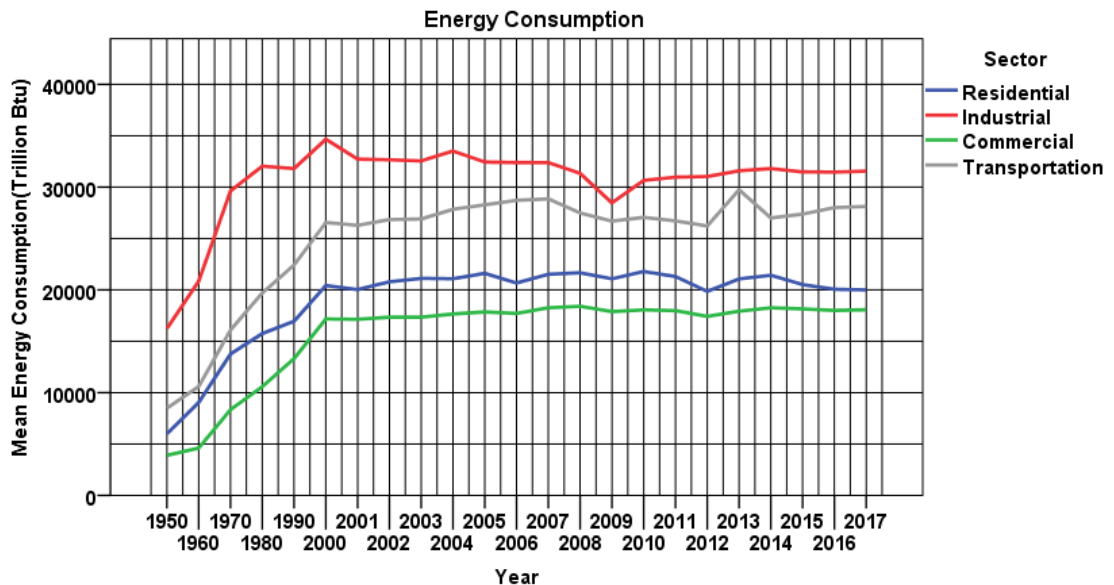


Fig. 1. Energy Consumption Report till December 2017

In June 2016, U.S Data Center Energy Usage Report [2] mentioned that the estimated total energy consumption of 2014 was 70 billion KWh and this is 1.8% of total power consumed by U.S. government. This report forecasts, it is likely to increase 73 billion KWh in 2020, which

is not favorable. In addition, Mike Ebbers et al. [3], IBM green datacenter journey has announced, only 45% of organizations maintain energy-efficient programs like energy star label for reducing CO₂ emission. Besides, each data center consumes 55% for power and cooling while 45% for IT load. Server hardware like power supply, memory, fan and drive consumes 70% of energy and 30% is consumed by processor. Resource usage of computing nodes utilizes 20%, while 80% of power is wasted in idle or sleep mode. Unpredicted load of cloud data centers consumes more energy. For that reason, heuristic-based energy and SLA-aware Modified Best Fit Decreasing algorithm is introduced for reducing energy consumption and SLA violations. The term heuristic is used to achieve immediate goals and helps to offer sufficient outputs with minimum duration. Many heuristic algorithms have been proposed for reducing energy consumption and increasing resource utilization. However, existing algorithms are typically focused on resource requirements of active servers in cloud data centers. In this paper, our SLA-aware MBFD algorithm supports dynamic allocation of resources to the VMs based on historical records. Hence, if the requested resources are allocated properly, energy consumption will be reduced radically. Otherwise, it will generate more complication to the cloud providers due to dynamic pricing schemes.

The paper also describes EMPMU VM migration policy for efficient utilization of cloud resources. This policy is used to minimize number of VM migration in cloud data centers by identifying overloaded and underloaded host usage. It also enables to achieve dynamic placement and migration of virtual machines in cloud environment as per the users' demands. Furthermore, it monitors the current CPU workload and then applies VM migrations for either overloaded or underloaded host. Energy-efficient resource allocation of cloud data centers focuses on maximizing CPU utilization, minimizing resource usage and power consumption. Therefore, SLA-aware MBFD and EMPMU heuristic algorithms are proposed to achieve efficient utilization of cloud resources, which will reduce energy consumption, avoid SLA violation and facilitate higher potential growth of cloud computing.

2. Related Work

Fei Cao et al. [4] have proposed an energy-efficient workflow scheduling based on Directed Acyclic Graph (DAG) model. In this paper, DVFS technique is adapted for resource provision and allocation of cloud data centers. In addition, authors have investigated few areas like resource utilization, high performance, VM overheads, energy consumption and CO₂ emissions. However, all approaches work efficiently but compromise on SLA regulations. Devendra Singh Thakur [5] has calculated resource usage for a single application using single host mechanism. In this, the authors have introduced PaaS layer for task scheduling instead of IaaS layer. Moreover, heterogeneous workload consolidation technique is used to find energy consumption of each data center. It also includes allocation policies of PaaS layer with SLA and restricts the users from overloading resource usage. However, it works only on a single application mechanism but not in multi-platform or cross-platform computing software.

Ning Liu et al. [6] have proposed an optimization model for task scheduling. It is used to minimize energy consumption using integer programming. Furthermore, Greedy Task Scheduling Algorithm (GTSA) is used to reduce average response time and total number of active servers needed to execute a task. In order to make efficient utilization of data centers, Most Efficient Server First Task Scheduling (MESF) algorithm is used to find energy usage cost. Finally, the result shows that 70% energy is saved. Meanwhile, authors mainly focused

on response time reduction but not on server load balance and system performance.

Hongyang Sun et al. [7] have considered the problem of effective resource management on homogeneous High Performance Computing (HPC). In this, authors have introduced Spatio and Temporal Thermal Aware Scheduling used for energy saving in HPC system. Furthermore, while minimizing the makespan of the HPC system; authors have validated On-line Heuristic Scheduling Algorithm (OHSA). This algorithm performs both job scheduling and thermal management system. However, the model focuses only on homogeneous HPC system but not on heterogeneous cloud environment. Y. Peng et al. [8] have discussed about Energy and QoS to minimize performance degradation and bandwidth cost. Based on these two constraints, authors have proposed a new algorithm called Evolutionary Energy Efficient Virtual Machine Allocation (EEE-VMA). This genetic algorithm-based meta-heuristic model is considered for heterogeneity power-aware VM allocation. This technique named as PowerMark, is used to monitor power efficiency of cloud data centers.

Saurabh Kumar Garg et al. [9] have addressed the problem and concluded that HPC is not only profitable for commercial applications but also plays a significant role in consumer's IT applications. In this paper, the authors have introduced a Near Optimal Scheduling Policy for heterogeneous data centers in geographical locations. The proposed model is formulated based on the data center's location, architectural design and management systems. However, energy consumption and power-aware problems are discussed without concentrating on service level objectives. Anton Beloglazov et al. [10] have proposed optimizing resource usage and energy consumption in cloud data centers. It is achieved by Dynamic Virtual Machine (DVM) consolidation techniques using live migration and turning idle machines to sleep mode. Subsequently, authors have addressed the online deterministic algorithm, used for single host migration and dynamic VM consolidation. Furthermore, based on historical data, dynamic consolidation and resource usage were calculated. However, single-host migration is inefficient compared to multi-host consolidation techniques.

Jing Liu et al. [11] have investigated task-scheduling model based on Multi-Objective Genetic Algorithm (MO-GA). In this algorithm, parameters such as encoding rules, crossover operators, selection operators and method for sorting pareto solution are considered. However, MO-GA is suitable for efficient resource usage in cloud environment but not considered on strict QoS and SLA requirements. XiaochengLiu et al. [12] have suggested a two-tier VM architecture named as Aggressive Consolidation based on FCFS (ACFCFS) for parallel workload consolidation. Experimental results were compared with traditional FCFS, EASY (Extensible Argonne Scheduling sYstem), CCFCFS (Conservative Consolidation based FCFS). The proposed algorithm is allowed inaccurate CPU usage while parallel job execution. Furthermore, it reduces available CPU idle time and starvation problem.

Anton Beloglazov et al. [13] have discussed green cloud architectural framework for energy efficient computing. The objective of this paper is to address the problem of energy-aware heuristic resource scheduling and VM allocation policies. It is mainly used for cost saving and reducing energy consumption of allocated VMs. Moreover, authors have proposed two new algorithms such as MBFD and Minimization of Migration (MM) policy for efficient utilization of cloud resources. Therefore, exploration of these two algorithms works efficiently with negotiated QoS. However, the authors have defined an efficient mechanism without focusing on dynamic requirements of cloud resources.

Chia-Ming Wu et al. [14] have pointed out that green cloud is an emerging technology and increases resource utilization and reduces power consumption. In this paper, authors have proposed a green-cloud scheduling algorithm using DVFS technique. Further, priority is allocated based on VM weights to assign respective resources in cloud data centers. However, while decreasing energy consumption and execution time, the authors have not achieved load balancing during overload resource usage. Fabio D. Rossi et al. [15] have proposed an orchestration of different energy-saving techniques to improve energy consumption and application performance. Energy efficient cloud orchestrator is defined as E-eco, which helps to reduce energy consumption in dynamic cloud environment. The proposed Orchestration E-eco is compared to a power-agnostic approach. However, an E-eco approach has reduced 25% of energy consumption and cost of 6% SLA violation.

Yuyang Peng et al. [16] have introduced an EEE-VMA for geographical data center locations. It works based on a Genetic Algorithm (GA) metaheuristic optimization technique. The proposed model supports power heterogeneity aware VM allocation using powerMark technique, which diagnoses the power efficiency of different cloud data centers. Furthermore, they have also investigated performance degradation due to VM co-location and bandwidth cost between cloud users and providers. Ehsan Arianyan et al. [17] have investigated energy consumption using consolidation of VMs in cloud data centers. In this paper, authors have proposed holistic resource management and heuristic-based multi-criteria decision-making method. In this paper, authors have introduced a method called Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and SLA-aware Allocation (TPSA) policy for both resource allocation and to avoid SLA violation. It is used to discover underloaded host and then move to the appropriate server. This paper has considered many parameters, but it failed to achieve performance of the system.

Jiyuan Shi et al. [18] have defined the problem of efficient resource allocation for large-scale data centers. Optimization problem arises when multi-dimensional resources are requested from different cloud users. To solve this problem authors have proposed pattern-based resource allocation mechanism for efficient utilization of server resources. Mohammad-Hossein Malekloo et al. [19] have introduced multi-objective optimization approach to balance energy consumption and avoid SLA violations. In this paper, authors have proposed energy-aware and QoS-aware Multi-objective Ant Colony Optimization (MACO) approach for VM placement and consolidation. The objective of VM placement algorithm is to minimize energy consumption, resource wastage and communication cost of energy usage. The VM consolidation technique aims to reduce SLA violation, VM migration and active physical machines. Since the authors have focused on many performance metrics, it has increased computational complexity of cloud data centers.

Ehsan Arianyan et al. [20] have investigated dynamic voltage frequency scaling and consolidation technique for energy consumption and SLA efficient resource management. In this paper, authors have discussed novel fuzzy multi-objective techniques namely DVFS-aware Consolidated Optimization (DCO) policy, Utility based DVFS Governor (UDG), Power and SLA Fuzzy Weighted TOPSIS (PSFWT) allocation policy and LR-AR-DVFS and TACND-AR-DVFS policies for determining the overloaded and underloaded PMs. Moreover, multiple criteria like CPU, RAM, and network bandwidth are considered in these proposed algorithms. The drawback of this paper is that authors have emphasized multi-objective

heuristics mechanism without concentrating on computational cost.

Zhou Zhou et al. [21] have proposed two novel adaptive energy-aware algorithms for maximizing energy efficiency and minimizing SLA violation rate in cloud data centers. In this paper, before allocating any resources into a VM, authors have considered workload of application type, CPU and memory. Minimizing energy consumption and SLA violation, adaptive three-threshold mechanism is used for cloud data centers. This three-threshold mechanism further classified into four classes, namely less loaded hosts, little loaded hosts, normally loaded hosts, and overloaded hosts. Based on these loads, VM selection and migration policies were applied. Abbas Horri et al. [22] have discussed novel QoS-aware VM consolidation approach, which works based on resource utilization history of VMs present on the host. They have suggested SLA-aware resource allocation method which helps to achieve minimize energy consumption and maximize system performance.

K.R. Remesh Babu et al. [23] have proposed an interference aware prediction mechanism for auto-scaling in cloud environment. In this paper, authors have discussed dynamic VM migration and avoid interferences due to VM migrations. For that purpose, authors have introduced live VM migration architecture for auto-scaling and it helps to handle sudden load changes in homogeneous cloud data centers. The live VM migration architecture includes three components such as load balancer, virtual cluster monitor system and auto-provisioning system. The auto-scaling mechanism works based on prediction threshold values without degrade overall performance of the system and SLA violations. The prediction effectiveness is verified by measuring the standard error with different threshold intervals. Finally, experimental results are proved using optimal threshold range for minimum VM migrations.

The main objectives of this paper are as follows:

- The mathematical models are designed for SLA violation detection, power consumption, efficient VM placement and migration techniques in cloud data centers.
- Further, energy-efficient resource allocation heuristic algorithm is used for efficient utilization of cloud resources. It includes resource provisioning and allocation of VMs without violating SLA procedures and satisfies expected QoS.
- To minimize VM migration and maximize resource utilization, we are introducing a EMPMU VM migration policy. The proposed algorithm is compared with Highest Potential Growth (HPG) and Random Choice (RC) policies.
- To investigate SLA violation in cloud data centers, we are proposing SLA-aware MBFD algorithm. This algorithm is to reduce energy consumption and SLA violation without performance degradation.
- Finally, EMPMU algorithm with different threshold intervals is compared to Non Power-Aware (NPA) policy, Dynamic Voltage Frequency Scaling (DVFS) and Single Threshold (ST) VM migration policies.

The rest of the paper is organized as follows: Mathematical models of SLA violation, energy consumption and resource allocation are discussed in section 3. Experimental results are discussed in section 4. Conclusion and future scope of our research work are explained in section 5.

3. Mathematical Models

The mathematical model is designed for SLA violation detection, power consumption, efficient VM placement and migration techniques in cloud data centers. To reduce power consumption, DVFS technique plays a significant role in all electronic devices. It has in-built frequency controller to supply appropriate voltage based on the requirement of VMs. Unpredicted load of cloud data centers consume more energy, thus increasing operational cost. However, energy-efficient resource management technique helps to minimize resource allocation effectively. Over usage or under usage of cloud data centers accidentally increases cost of energy. Many users access same host for different needs; therefore, unbalanced load of data centers consumes more energy and violate SLA procedures. Moreover, dynamic placement of VMs and on-demand resource allocation on those VMs sometimes failed to satisfy user requirements. While meeting SLA procedure, cloud users always expect quality of services from the cloud providers. Reducing energy consumption and SLA violation, we move on to efficient resource allocation techniques for minimizing such problems. To ensure the expected quality of services, SLA-aware MBFD and EMPMU algorithms are used for power-aware resource provisioning without violating SLA procedures.

3.1 Model for SLA Violation Detection

The objective of this model is to find whether the host runs within SLA constraint or not. When the host executes in overloaded condition it may lead to violation of SLA. Sometimes, the host goes to underloaded state due to minimum load. Efficient utilization of resources in cloud data centers will focus on two important scenarios such as QoS and SLA violation. To maintain proper SLA conditions, we can eliminate unexpected QoS problems. For this reason, we have formulated SLA violation detection model before allocating any VMs into the server. Some of the required notations are used for SLA violation detection model described in **Table 1**.

Table 1. List of notations used for SLA violation detection

Notations	Description
H_i	The i^{th} Host in Data Center(DC)
V_{ij}	The j^{th} Virtual Machine on i^{th} host
N	Total number of host in DC
M_i	Total number of VMs in i^{th} host
MaxUtil_{H_i}	Maximum Utilization of host H_i
AllotUtil_{H_i}	Total Allocated Utilization for VM_j by host H_i
$\text{MaxUtil}_{V_{ij}}$	Maximum Utilization of j^{th} VM on host H_i
$\text{ReqUtil}_{V_{ij}}$	Request Utilization of j^{th} VM on host H_i
ReqUtil_{V_i}	Request Utilization of j^{th} VM in VMsToMigrate

3.1.1 SLA Formulation Statement

According to Zhibo Cao et al. [24], we have formulated the total number of VMs exceeding an allocated utilization host H_i . If total number of requested VMs exceeds more than the allocated VMs, SLA violated on host H_i . When allocated VMs are equal to requested VMs, no violation has occurred. To follow the above conditions, we can derive the following equations:

$$\left\{ \begin{array}{l} \frac{\sum_{j=1}^{M_i} \text{AllotUtil_V}_{ij}}{\text{MaxUtil_H}_i} < \frac{\sum_{j=1}^{M_i} \text{ReqUtil_V}_{ij}}{\text{MaxUtil_H}_i}, \text{SLA Violation} \\ \frac{\sum_{j=1}^{M_i} \text{AllotUtil_V}_{ij}}{\text{MaxUtil_H}_i} = \frac{\sum_{j=1}^{M_i} \text{ReqUtil_V}_{ij}}{\text{MaxUtil_H}_i}, \text{No SLA Violation} \\ \frac{\sum_{j=1}^{M_i} \text{AllotUtil_V}_{ij}}{\text{MaxUtil_H}_i} > \frac{\sum_{j=1}^{M_i} \text{ReqUtil_V}_{ij}}{\text{MaxUtil_H}_i}, \text{Impossible} \end{array} \right. \quad (1)$$

$$\frac{\sum_{j=1}^{M_i} \text{ReqUtil_V}_{ij}}{\text{MaxUtil_H}_i} = x_i \text{ for each Host } H_i \quad (2)$$

$$\left\{ \begin{array}{ll} 1.0 < x_i, & \text{SLA Violation} \\ 0 \leq x_i \leq 1.0, & \text{No SLA Violation} \\ x_i < 0, & \text{Impossible} \end{array} \right. \quad (3)$$

The objective of Eq.(3) is used to find whether the host is violated by SLA conditions or not. As stated by Eq.(3), we derived the following SLA conditions: Total allocated hosts are less than the requested hosts SLA violated on host H_i . There are three types of conditions considered for SLA violation detection model.

- i. Guaranteed SLA violation is defined as $x_i > 1.0$. In this situation host H_i is overloaded and unable to allocate required resources to those hosts.
- ii. No SLA violation or possible violation is defined as $0 \leq x_i \leq 1.0$. If host H_i executed within the limit, no violation happens. Sometimes, hosts H_i goes to underloaded condition when possible violation occurs.
- iii. Impossible state is also called as sleep state. There is no violation in any situation when $x_i = 0$.

Next, we find which host transforms into SLA violation mode. Sometimes, a few hosts may satisfy $0 \leq x_i \leq 1.0$, but go to violation mode. In this situation, we examine the following terms:

- 1) Requested utilization may change when users increase in a particular host. Therefore, x_i is defined as random variable and the mean is denoted by μ_i .
- 2) Based on historical records of x_i , we can find standard deviation σ_i , where $P(x_i)$ is taken as probability density function of x_i . The probability condition in subsequent intervals using Eq.(4) is as follows:

$$\begin{aligned} P\{x_i\} \geq 1 &= P\{x_i - \mu_i \geq 1 - \mu_i\}, \text{ where } 1 - \mu_i = \varepsilon_i \\ &= P\{x_i - \mu_i \geq \varepsilon_i\} \\ &\leq \int_{x_i - \mu_i \geq \varepsilon_i} \left(\frac{x_i - \mu_i}{\varepsilon_i} \right)^2 p(x_i) dx_i \\ &= \frac{1}{\varepsilon_i^2} \int_{\varepsilon_i + \mu_i}^{\infty} (x_i - \mu_i)^2 p(x_i) dx_i \\ &\leq \frac{1}{\varepsilon_i^2} \int_0^{\infty} (x_i - \mu_i)^2 p(x_i) dx_i \\ &= \frac{\sigma_i^2}{\varepsilon_i^2} \end{aligned}$$

(4)

Whenever mean value is greater than 1, host H_i may be assumed in overloaded condition and it is on violation mode. Meanwhile, if the mean value is less than 1, host H_i is considered as Eq.(4). Highest probability density function cannot exceed 1, but sometimes when unpredicted load occurs on VM, it goes to violation mode and probability ratio is greater than 1. Therefore, the following Eq.(5) is satisfied:

$$\begin{cases} \forall \sigma_i, \mu_i \geq 1.0 \\ \sigma_i \geq 1 - \mu_i, \mu_i < 1.0 \end{cases} \quad (5)$$

3.1.2 SLA Violation Detection Factors

Quality of services can be achieved through SLA formulation. Meeting such services is defined in terms of minimum throughput and maximum response time. Moreover, each service is varying from type of resources delivered to the particular host. Measuring SLA violation in IaaS, we used two metrics defined by R Buyya [10], namely SLA violation Time per Active Host (SLATAH) and Performance Degradation due to Migration (PDM). SLATAH is used to measure percentage of time, during which active hosts have experienced the CPU utilization of 100 %. When a host experiences 100 % utilization, it will not be able to allocate enough CPU to the VMs and will generate SLA violation. The SLATAH can be calculated using Eq. (6):

$$SLATAH = \frac{1}{N} \sum_{i=1}^N \frac{T_{si}}{T_{ai}}, \quad (6)$$

where N is the number of hosts, T_{si} is the total time when the host i goes to 100% utilization and leads to SLA violation mode. T_{ai} is the total number of active host which resides and serves VMs. PDM can be calculated using Eq. (7):

$$PDM = \frac{1}{M} \sum_{j=1}^M \frac{C_{dj}}{C_{rj}}, \quad (7)$$

where M is the number of VMs, C_{dj} is the estimate of the migration performance degradation of VM j , C_{rj} is the total CPU utilization requested by j during its lifetime and C_{dj} is the 10% CPU utilization of j during VM migration. Hence, reduction in SLA violation due to performance degradation and VM migrations can be calculated using Eq. (8):

$$SLAV = SLATAH \cdot PDM \quad (8)$$

3.2 Model for Power Consumption

Today, power consumption is the most threatening and cost-effective area in cloud research. Cloud providers spend huge amounts for power consumption and maintain 24/7 running servers. Rather than active servers, service providers strictly concentrate on idle servers, network interfaces, disk storage, SLA violation and cooling systems. According to Ed Weber [25] IBM announced 55% of energy is consumed by power and cooling system. However, 80% of power is wasted when the server is idle. Besides, varying resources execute with different workload and at different times rendering dynamic power consumption.

3.2.1 Power Consumption Approaches

For evaluating total power consumed by each VM, we can define a set of virtual machines $V = \{V_1, V_2, V_3, \dots, V_m\}$. Each VM has $V_i = \{F_{\min}, F_{\max}\}$ set of working frequencies and $V_i = \{V_{\min}, V_{\max}\}$ voltage range. Let $T = \{T_1, T_2, T_3, \dots, T_j\}$, where T is considered as set of tasks. Each task T depends on V workload. Let $ST = \{ST_1, ST_2, ST_3, \dots, ST_s\}$, where each task may be in the form of sub task(ST). This is to be considered as Eq(9):

$$\text{each } ST \in T_j \quad (9)$$

$$\text{If we denote } f \text{ as working frequency of each sub task, then } f = \int_{ST_{\min}}^{ST_{\max}} ST \quad (10)$$

$$\text{Power consumed by each VM is calculated as } P = \lambda v^2 f, \quad (11)$$

where P is power, parameter λ is defined as switching activity factor. It is calculated by multiplying flip frequency with load capacity. Flip frequency is also known as number of switching activity per clock cycle from 0 to 1. Voltage v may be increased or decreased depending on the workload. Resource i executes subtask j with varying working frequency of f . DVFS technique is used to dynamically adjust voltage and frequency to reduce power consumption in cloud data centers. In our proposed EMPMU power-scheduling algorithm, we take the value of λ as 1. Therefore, power consumption of a task can be calculated using Eq.(12) proposed by Sathya Sofia and Ganesh Kumar [26]:

$$PT_{(i,f)} = \lambda f(i)_j [v(i)_j]^2 PT_{(i,j)} \quad (12)$$

The total power consumption to complete a task in each VM can be calculated using Eq.(13). The list of notations used in power consumption model is given in **Table 2**.

$$\text{Power Consumption } PC_{\text{total}} = \sum_{i=1}^m \sum_{j=1}^n \lambda f(i)_j [v(i)_j]^2 PT_{(i,j)} \quad (13)$$

Power consumption of each CPU is calculated by Eq.(14):

$$P(u) = k \cdot P_{\max} + (1 - k) \cdot P_{\max} \cdot u, \quad (14)$$

where P is power, u is the current CPU utilization, k is the fraction of power consumed by idle server, and P_{\max} is maximum power consumed when the server is in full utilization.

Case 1: As per E.Elnozahy [27], idle server power consumption k is 70%, maximum power consumption P_{\max} is 250W, and current CPU utilization u is 0%:

$$P(0) = 70/100 \times 250 + (1 - 70/100) \times 250 \times 0$$

$$P(0) = 175W$$

Case 2: Idle server power consumption k is 70%, maximum power consumption P_{\max} is 250W, and current CPU utilization u is 100%:

$$P(100) = 70/100 \times 250 + (1 - 70/100) \times 250 \times 100/100$$

$$P(100) = 250W$$

Table 2. List of notations used for power consumption

Notation	Description
$V_1 \dots V_m$	Set of virtual machines(V) in a host
$V_{\min} \& V_{\max}$	Minimum and maximum frequency of V
$T_1 \dots T_j$	Set of tasks placed in V
$ST_1 \dots ST_s$	Set of sub task placed in task T
$ST_{\min} \& ST_{\max}$	Minimum and maximum frequency of sub task ST
P, λ, v, f	Power, switching activity factor, voltage, frequency
$v_k(i)$	Required voltage of i
$f_k(i)$	Frequency
$PT_{(i,j)}$	Processing Time to complete a task
PC_{total}	Total power consumption of V
u	Current CPU utilization
k	Fraction of power consumed by idle sever
P_{max}	Maximum power by server

3.3 Model for VM Allocation

Recently, cloud usage has increased rapidly. Resource provisioning and allocation policies are updated periodically with increasing number of cloud users. Efficient use of cloud resources is achieved by dynamic placement of VMs and live migration scheme. Hence, dynamic VM placement is a powerful technique used to allocate required VMs on physical machines. The VM allocation is divided into two major parts such as VM placement and VM optimization. VM placement is initiated when new request is received from each host. The second part is optimizing each VM based on VM migration policies. Therefore, SLA-aware MBFD algorithm is used for VM allocation and EMPMU is used for VM optimization.

3.3.1 VM Placement

Consolidation of VMs is closer to a bin packing problem, where each VM has varying size (bin size) and resource (items) requirement. In the proposed scheme, we focus on efficient utilization of resources and consolidating VMs using SLA-aware MBFD algorithm. Martello et al. [28] introduced the basic terminology of bin packing problem and it is described as follows:

Given n items (VM) and m bins (host), let us consider set of hosts are denoted by $H_1, H_2, H_3, \dots, H_k$ with same size C . List of n VMs and weight should be considered as $VM_1, VM_2, VM_3, \dots, VM_n$. Assign each VM to one host; then total weight of the VM in each host does not exceed C and the number of host used is minimized. Find an integer number of host B and B -partition considered as H_1UH_2, \dots, UH_k such that

$$= \sum_{j \in H_k} VM_j \leq C, \quad (15)$$

where $k=1,2,\dots,B$. Solution of B is optimal when it is used reduce minimum host usage. The B value of an optimal solution is taken as OPT. A possible mathematical formulation of the problem is considered in [29]:

$$\text{minimize } B = \sum_{i=1}^n v_i \quad (16)$$

Subject to $B \geq 1$,

$$\sum_{j=1}^n VM_j x_{ij} \leq C y_i, \forall i \in \{1, 2, \dots, n\} \quad (17)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad \forall j \in \{1, 2, \dots, n\} \quad (18)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \{1, 2, \dots, n\}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in \{1, 2, \dots, n\} \forall j \in \{1, 2, \dots, n\}$$

where $y_i = 1$ and if host i is used and $x_{ij} = 1$ if VM j is placed in host i . The weights of VM_j are positive integers. Therefore, without loss of generality, we can define C is a positive integer.

3.3.2 VM Optimization

VM optimization can be divided into VM selection and VM migration. There are many VM selection and placement policies that exist on current energy management research. First, we have to check over- and under utilization of VM from each host. Secondly, each VM has to be migrated using appropriate migration policy. In this section, we have discussed three VM selection and migration policies, namely Highest Potential Growth (HPG), Random Choice (RC) Policy and Enhanced minPower and maxUtilization Policy. HPG policy is more effective, when the upper threshold is violated and subsequently migrate those VMs to lowest CPU usage within the capacity. RC policy reveals random selection of VMs needs to decrease the CPU utilization by a host below the upper utilization threshold. EMPMU policy relies on minimum power and maximum utilization of resources without violating SLA constraints. To minimize the power consumption, EMPMU policy sort all VMs based on current CPU utilization in decreasing order and allocate each VM to a host with least increasing power consumption.

3.3.3 Enhanced minPower and maxUtilization(EMPMU) VM Migration Policy

To find total utilization of CPU and power consumption, we set lower and upper threshold value to get accurate results. When a threshold value exceeds the allocated limit, it will conclude that VM migration is required for that host. If the threshold value goes to lower limit, we consider those VMs to have migrated to idle state or sleep mode to minimize power consumption. Sometimes, over-utilized VMs are migrated to underutilized hosts to balance the server load. The complexity of the EMPMU algorithm is $u \times n$, where u is the over-utilized hosts and n is the number of virtual machines allocated on that host.

Algorithm 1. Enhanced minPower and maxUtilization(EMPMU) Algorithm

```

1  Input: VMsToMigrate, minPower
2  Output: MigrationMap
3  AscendingVMs=sort_Ascending(VMsToMigrate)
4  DecendingVMs=sort_Decending(VMsToMigrate)
5  foreach  $V_j$  in DecendingVMs do
6      AllotedHost=minPower( $V_j$ )
7      AscendingVMs.remove( $V_j$ )
8      DecendingVMs.remove( $V_j$ )
9      foreach  $V_k$  in AscendingVMs do
10         if AllotedHost.isHostUnder_Utilization( $V_k$ ) then
11             waitState.put(allotedHost,  $V_k$ ), goto step14
12         else if AllotedHost.isHostOver_Utilization( $V_k$ ) then
13             MigrationMap.put(allotedHost,  $V_k$ )
14             allotedHost.create( $V_k$ )
15             AscendingVMs.remove( $V_k$ )
16             DecendingVMs.remove( $V_k$ )
17         else
18             end
19  return MigrationMap

```

The optimization of VM migration is carried out using upper and lower bound of CPU utilization. Migration of each VM is calculated by total resource usage of current CPU utilization. Migrating VMs from one host to another, we have to consider the following questions: when, where and which VM has to migrate? **Algorithm 1** demonstrates minimum power consumption and maximum utilization of resources in cloud data centers. Here, we have used double-threshold VM migration policy to reduce power consumption. Migrations of VMs are addressed by two conditions. First, when CPU utilization reaches a lower threshold, all VMs are moved from current host to sleep state. After getting appropriate threshold value, all lower bound VMs are migrated to suitable host. Secondly, when CPU utilization exceeds upper bound, those VMs are moved to minimum utilization host by applying consolidation technique. Hence, the objective of VM migration is to reduce energy consumption, which utilizes heterogeneity resource effectively.

3.4 Model for Resource Allocation Algorithm

An increasing number of cloud users' are expecting uninterrupted services over the internet. These services are provided by the cloud providers and mainly focused on increasing resource utilization and ROI. In this scenario, we have proposed efficient VM scheduling algorithm to manage all those resources in cloud data centers. Whenever new tasks arrive into the VM, VM manager monitors the current workload of VM and allocate the jobs appropriately.

3.4.1 SLA-Aware Resource Allocation

Energy-efficient VM provisioning and allocation are recent research topics in cloud computing. While, cloud providers allocate resources for maximum utilization, it should be emphasized by SLA limitations. Providers cannot change their convenient VM consolidation or migration process as their needs. For privileged customers, cloud providers should satisfy QoS without violating SLA. Our proposed scheme defines VM provision using SLA-aware

MBFD algorithm. This requires placing the VM in an appropriate host using resource usage-based EMPMU migration technique.

3.4.2 SLA-Aware Modified Best Fit Decreasing(SLA-Aware MBFD) Algorithm

Reducing computational complexity and increasing optimal solution, many heuristic algorithms have been developed, such as first-fit, best-fit and worst-fit. Each algorithm produces different non-guaranteed optimal solution depending on the number of objects placed in each bin. Hence, for effective use of heuristic algorithm for VM placement, we adopt the SLA-aware MBFD algorithm. The proposed model we have applied $11/9.OPT+1$ where OPT is the number of bins (Host) given by the optimal solution [29]. The complexity of the SLA-aware MBFD algorithm is $n \times m$, where n is the number of virtual machines that need to be allocated on the host and m is the number of hosts that reside in a data center.

Algorithm 2. SLA-Aware Modified Best Fit Decreasing(SLA-Aware MBFD) Algorithm

```

1  Input: hostList, vmList, SLA_List,n
2  Output: allocation of VMs
3  vmList.sortDecreasingUtilization()
4  foreach vm in vmList do
5      minPower←MAX
6      allocatedHost←NULL
7      foreach host in hostList do
8          if host has enough resource for vm then
9              foreach SLA_Level in SLA_List do
10                 if SLA_Level≤n-1 then
11                     SLA_Lower←lowerCPU_Threshold
12                     SLA_Higher←higherCPU_Threshold
13                     power←estimatePower(host, vm)
14                     if power < minPower then
15                         allocatedHost←host
16                         minPower←power
17                 if allocatedHost ≠ NULL then
18                     allocate vm to allocatedHost
19  return allocation

```

Traditional MFBD algorithm sorts all the VMs in decreasing order, based on current CPU utilization. Further, it allocates each VM in the mode of power efficient nodes first. Apart from traditional MFBD, we have enhanced our algorithm with SLA constraints represented in [Table 1](#). Here, lower level SLA is assigned to minimum resource usage and higher-level is assigned to maximum resource requirement. Let n be the total number of SLA levels and L be the individual level. SLA allocation constraints are addressed from 0 to $n-1$, $0 \leq \text{SLA_Level} \leq n-1$. For example, [Table 3](#) shows SLA levels and number of VMs for each level. Lower level SLA initiates from 1–to–200, and higher-level SLA is assigned based on the dynamic load of current job. An increasing load of current CPU gradually increases SLA level in an increasing order. However, to avoid SLA violations and eliminate free resources, our proposed algorithm is used to minimize number of VM creations in cloud data centers.

Table 3. SLA Levels and Number of VMs

SLA Level	Number of VMs
0	1–200
1	201–400
2	401–600
3	601–800
4	801–1000
5	1000 and above

4. Performance Analysis and Results Discussion

An efficient resource allocation heuristics using SLA-aware MBFD and Enhanced minPower, maxUtilization migration policy is presented in section 3. Our experimental results demonstrate increasing resource utilization with less SLA violation and reduced power consumption in cloud data centers. To validate our model, we have chosen CloudSim 3.0.3 simulation toolkit. Moreover, the reason for choosing CloudSim toolkit is that it supports multi-objective functions in single platform such as resource allocation, VM placement, power management, network and storage management.

4.1 Experimental Setup

Simulation is carried out over 100 heterogeneous nodes for improving quality of results in large-scale environment. Each node is designed as 1000, 2000 or 3000MIPS (Million Instructions Per Second) with one-core CPU. In addition, 16GB RAM, 1 TB of storage space allocated for performance evaluation. Power consumption of each host is in the range 175–250W. Based on Eq.(14), 175W is consumed when a host in 0% utilization meanwhile, 250W is taken with 100% CPU utilization. Besides, each VM needs 1 CPU core and 250, 500, 750 or 1000MIPS, 128MB RAM and 1 GB storage space used.

4.2 Simulation Results

The experimental results of EMPMU are compared with HPG and RC policies. For comparing these two migration policies, we have tested our data sets using two-way ANOVA test powered by SPSS IBM. Each test concludes either no difference between each groups called null hypothesis or significant difference between each groups called reject hypothesis. From this observation, we have tested our samples within groups and between groups to find a critical value. We have tested all simulation results when threshold P-value=0.05 with 95% Confidence Interval (CI). **Fig. 2** represents energy consumption using EMPMU policy incorporated with different utilization threshold values. Lower Utilization Threshold (LUT) starts from 10% and gets increased upto 100% utilization. Here, 0% utilization of CPU consumes total energy consumption is 3.94224KWh and 100% utilization decreased upto 0.152155KWh. Lower bound and upper bound energy consumption of 95% CI:(0.6075, 2.4798). The mean square of energy consumption is calculated between groups 1.474 and within groups 0.942. We obtained two-way ANOVA test result with significant difference between groups 0.0155. Therefore, the result is concluded that there is no significant difference between groups when P-value>0.0155 and it belongs to null hypothesis group. Moreover, while utilization of CPU loads increases, energy consumption is reduced. **Fig. 3** presents SLA violations at different intervals like energy consumption. It shows that when utilization threshold increases, SLA violation also increases. The mean square of SLA

violation is obtained between groups 5.404 and within groups 6.914. Significant difference between groups is 0.993 with 95% CI:(0.2231, 6.2164). SLA violation result clearly demonstrates that there is a significant difference between groups. Therefore, it determines the reject hypothesis group.

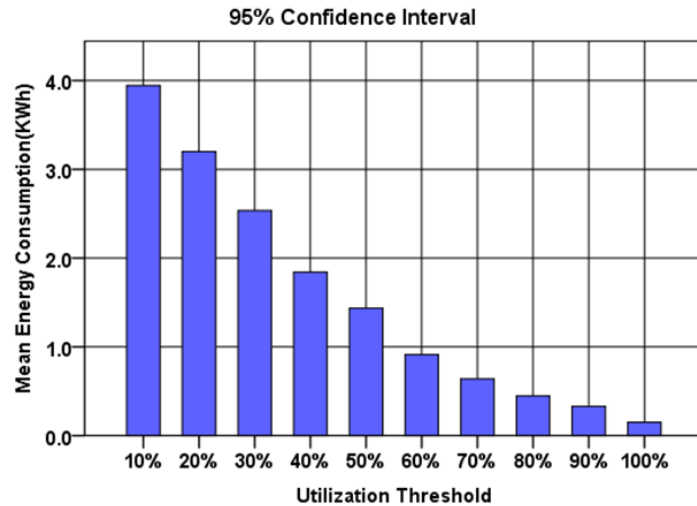


Fig. 2. Energy Consumption using EMPMU policy

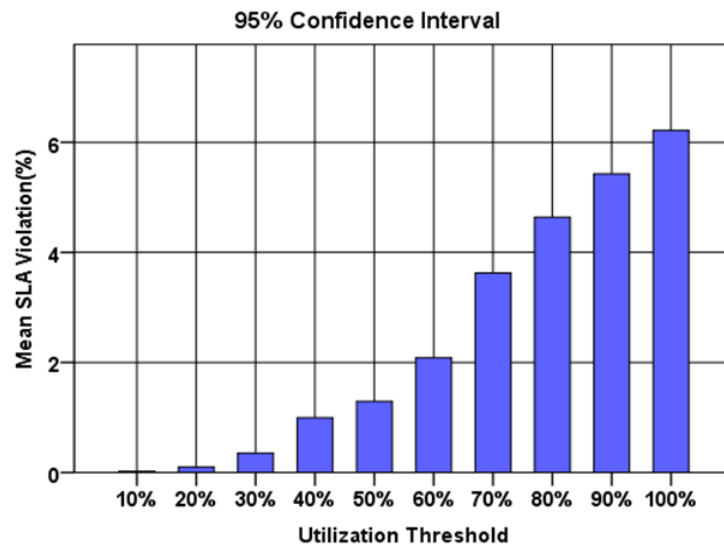


Fig. 3. SLA violations using EMPMU policy

The term Single Threshold (ST) is defined as set all hosts CPU utilization threshold with fixed value. If the threshold value exceeds the allocated limit, then it will be concluded that SLA violation has occurred on that host. However, single threshold controls number of VM migrations and saves more energy than DVFS technique. Moreover, Double Threshold (DT) is to set upper and lower utilization thresholds for active hosts. It maintains total utilization of CPU by all VMs run within these thresholds limit. Allocated CPU utilization of the host goes to lower limit whereas all lower threshold VMs migrated from this host switched to sleep state in order to minimize idle energy consumption. In addition, threshold values exceed the upper

limit and those VMs migrate from that host to minimize energy consumption and resource utilization. **Table 4** shows performance analysis of EMPMU VM placement policy. Upper and lower utilization of double threshold values is tested between 10% to 100% intervals. Each interval is compared with four scaling factors namely energy consumption, SLA violation, number of VM migrations and average SLA violation.

Table 4. Performance analysis of VM placement

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Energy (KWh)	3.606	1.611	1.127	0.774	0.477	0.351	0.281	0.252	0.190	0.114
	3.304	1.857	1.032	0.642	0.478	0.427	0.244	0.235	0.121	0.103
	2.650	1.423	0.884	0.535	0.358	0.201	0.192	0.094	0.034	0.028
SLA Violations	0.435	0.196	0.509	1.043	2.793	6.752	9.242	9.324	10.542	11.859
	0.435	0.638	0.840	2.037	4.118	7.526	8.734	9.264	10.453	10.934
	0.202	0.263	0.344	1.043	2.590	4.031	5.269	5.590	6.895	7.430
VM Migrations	110	145	210	261	305	421	599	648	762	895
	160	172	223	287	342	580	680	744	812	926
	65	89	120	176	230	310	445	523	690	712
Average	0.336	0.559	0.509	1.443	2.893	6.752	7.242	7.554	8.200	8.073
SLA	0.236	0.438	0.740	1.637	3.118	5.526	6.934	7.300	8.523	9.411
Violations	0.102	0.213	0.304	1.003	2.290	3.031	5.269	5.590	6.210	6.433

Fig. 4(a) illustrates mean energy consumption. Two-way ANOVA test EMPMU 95% CI: (0.6876, 1.7152) is less than HPG 95% CI: (1.2658, 2.1538) and RC 95% CI: (0.9231, 2.2582). The mean square of energy consumption is calculated between groups 0.167 and within groups 1.122. Calculated significant difference is 0.0132. Meanwhile, P-value > 0.0132 and HPG and RC policy consume more energy than EMPMU. **Fig. 4(b)** shows SLA violation using double threshold policy. The mean square SLA violation is noted between groups 2.356 and within groups 6.546 where P-value < 0.093. EMPMU 95% CI: (-0.2820, 3.5173), HPG 95% CI: (0.2074, 5.1149) and RC 95% CI: (0.5627, 5.5816). There is significant difference between groups when utilization of threshold value is increased. Therefore, the results of SLA violations are inconsistent in each interval and violated null hypothesis. **Fig. 4(c)** illustrates number of VM migrations in cloud data centers. In comparison with HPG and RC, EMPMU policy radically decreases number of VM migrations. The mean square of VM migrations are calculated between groups 36952.048 and within groups 29643.492, respectively. EMPMU 95% CI: (79.56, 330.44), HPG 95% CI: (160.46, 537.83) and RC 95% CI: (135.74, 450.26). Therefore, comparing EMPMU policy with HPG and RC policies, it achieves null hypothesis. **Fig. 4(d)** shows average SLA violations. EMPMU 95% CI: (0.7304, 3.8532), HPG 95% CI: (0.2991, 6.6518) and RC 95% CI: (-0.3295, 6.3206). HPG and RC policies are relatively equal with less variation in confidence interval. Mean square of average SLA violation is calculated between groups 4.180 and within groups 9.633. Moreover, result of null hypothesis is achieved when calculated hypothesis value statistically has no significant difference whereas P-value > 0.0021.

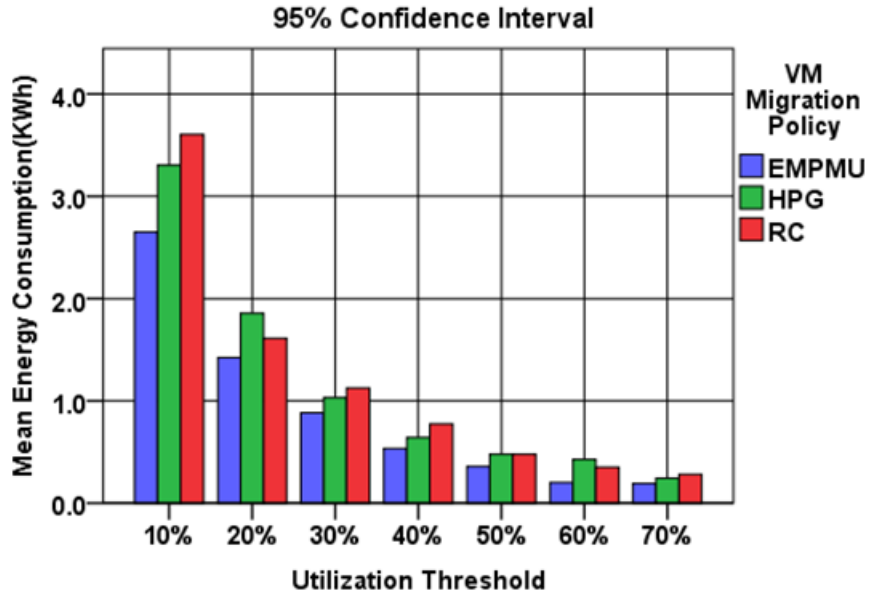


Fig. 4(a). Energy consumption

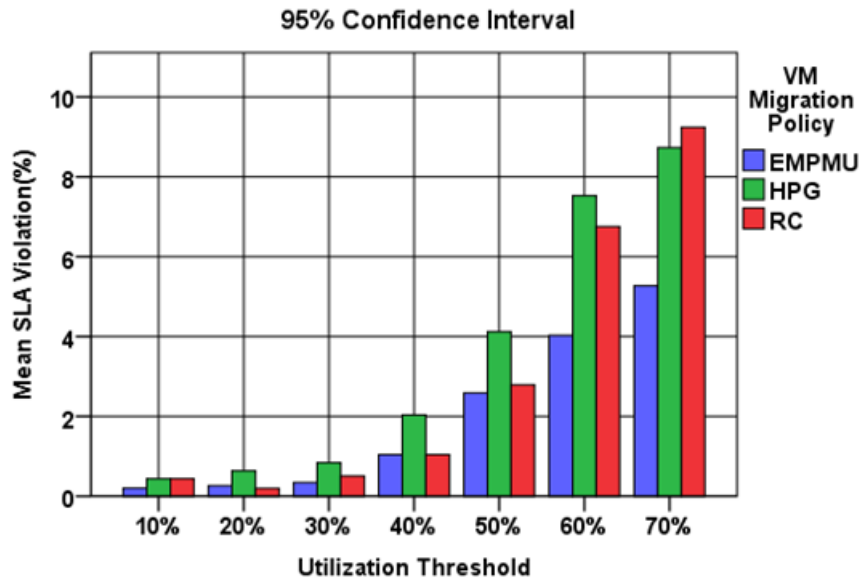


Fig. 4(b). SLA violations

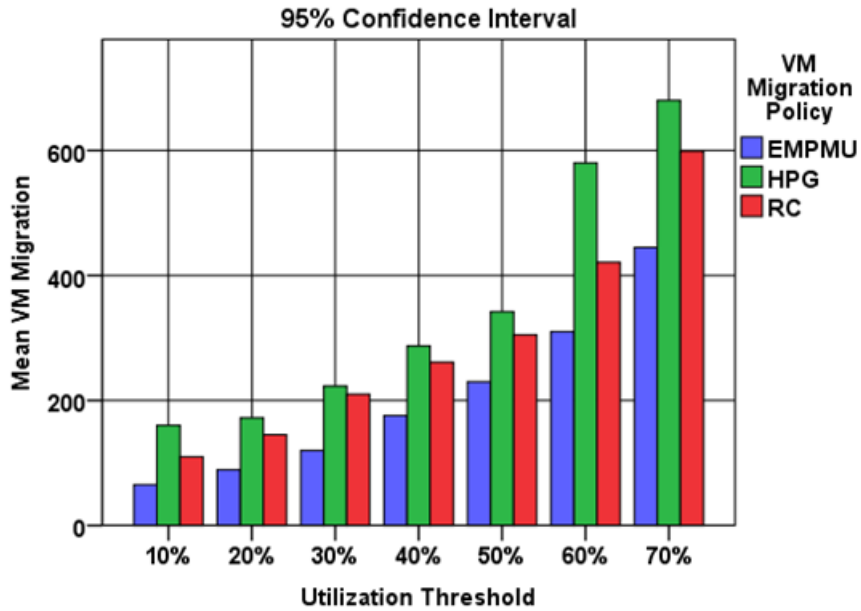


Fig. 4(c). Number of VM migrations

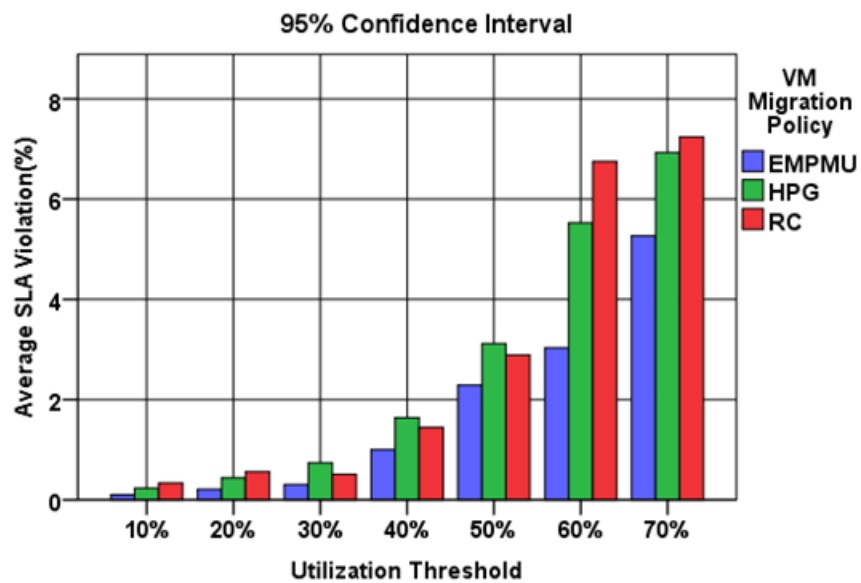


Fig. 4(d). Average SLA violation

Performance analysis of various VM migration policies and its different threshold intervals are represented in Table 5. We have compared our results with Non Power-Aware (NPA), DVFS and Single Threshold (ST) policies. Non power-aware policy does not concentrate on power-aware optimization strategies. It implies that all hosts are executing at 100% CPU utilization and it will consume more power than all other policies. DVFS technique performs based on dynamic voltage and frequency adjustment of each host. It will reduce maximum power consumption compared to NPA policy. However, both NPA and

DVFS do not adapt any VM allocation at run time. Both policies are mainly used for only reducing power consumption. Therefore, mean values of NPA, DVFS, ST and EMPMU policies are demonstrated with four scaling factors such as energy consumption, SLA violation, VM migration and average SLA violation.

Table 5. Final experimental results

Interval	Energy, KWh	SLA Violations	VM Migrations	Average SLA Violations(%)
NPA	9.15(9.03,9.27)	-	-	-
DVFS	4.40(4.39,4.41)	-	-	-
ST 40%	2.03(2.01,2.05)	12.6(12.4%,12.9%)	320(317,323)	12.84(12.82%,12.86%)
ST 70%	1.62(1.62,1.63)	15.4(15.2%,15.6%)	398(360,437)	16.55(16.52%,16.58%)
EMPMU 30%–70%	1.12(1.11,1.13)	3.32(3.30%,3.35%)	103(094,113)	4.43(4.42%,4.44%)
EMPMU 50%–90%	0.52(0.51,0.53)	4.23(4.22%,4.25%)	147(132,162)	6.91(6.90%,6.92%)

Fig. 5(a) shows energy consumption between four policies and different threshold intervals. NPA total power consumption is 9.15KWh with 95% CI: (9.03, 9.27), DVFS is 4.40KWh with 95% CI: (4.39, 4.41). ST 40% utilization is 2.03KWh with 95% CI: (2.01, 2.05), ST 70% utilization is 2.03 with 95% CI: (2.01, 2.05). EMPMU 30%–70% double threshold policy consumes 1.12KWh with 95% CI: (1.11, 1.13), EMPMU 50%–90% is 0.52 with 95% CI: (0.51, 0.53). Therefore, EMPMU policy consumes very less energy than other policies. **Fig. 5(b)** shows SLA violations with different VM migration policies. SLA violations and VM migrations are not applicable in NPA and DVFS techniques. SLA violation in ST 40% is 12.6% with 95% CI: (12.4%, 12.9%). ST 70% is 15.4% with 95% CI: (15.2%, 15.6%). EMPMU 30%–70% is 3.32% with 95% CI: (3.30%, 3.35%). EMPMU 50%–90% is 4.23% with 95% CI: (4.22%, 4.25%). Hence, SLA violation using single threshold is 12% where double threshold EMPMU policy is reduced upto 4%. **Fig. 5(c)** illustrates number of VM migrations. ST 40% VM migration is 320 with 95% CI: (317, 323). ST 70% is 398 with 95% CI: (360, 437). EMPMU 30%–70% is 103 with 95% CI: (094, 113). EMPMU 50%–90% is 147 with 95% CI: (132, 162). VM migration is also reduced when compared with ST policy. **Fig. 5(d)** presents average SLA violation. Average SLA violation in ST 40% is 12.84% with 95% CI: (12.82%, 12.86%). ST 70% is 16.55 with 95% CI: (16.52%, 16.58%). EMPMU 30%–70% is 4.43% with 95% CI: (4.42%, 4.44%). EMPMU 50%–90% is 6.91% with 95% CI: (6.90%, 6.92%). Therefore, observation of proposed EMPMU policy average SLA violation is reduced from 16% to 6%.

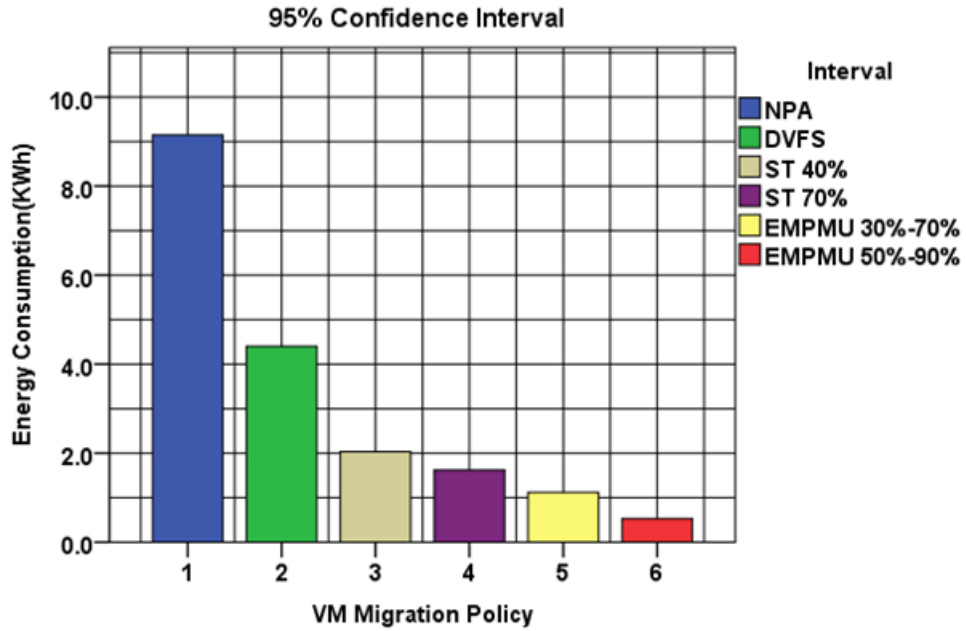


Fig. 5(a). Energy consumption

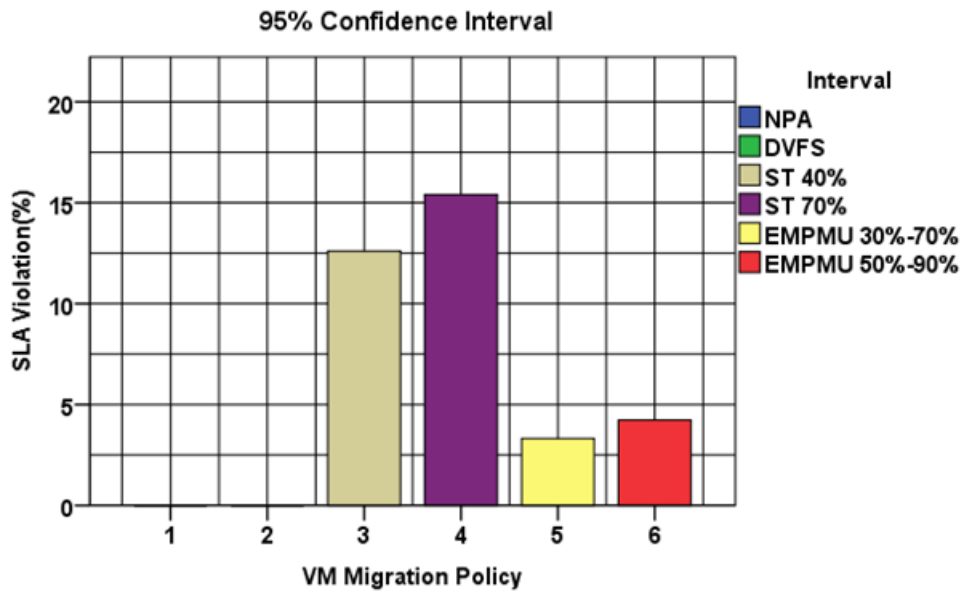


Fig. 5(b). SLA violations

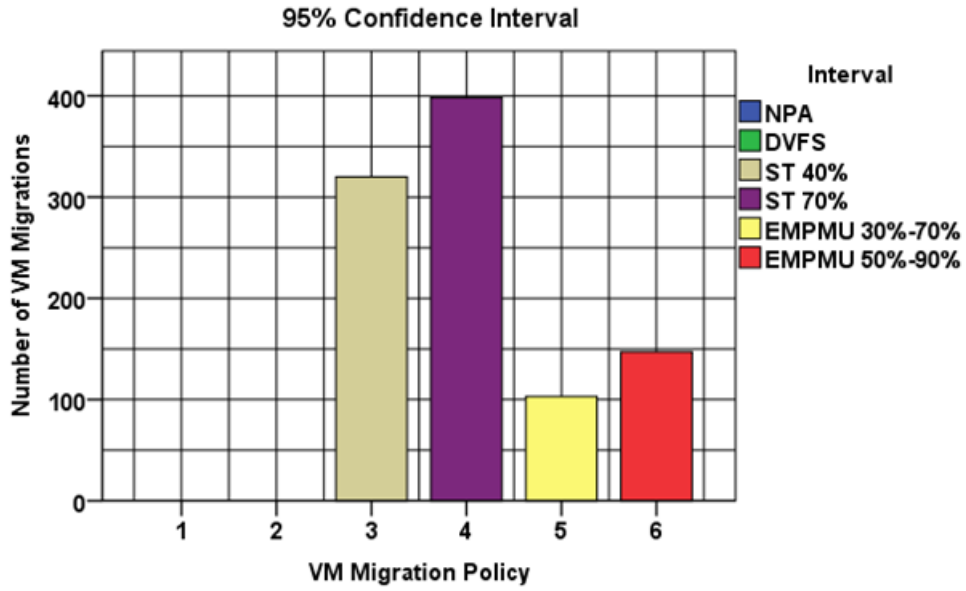


Fig. 5(c). Number of VM migrations

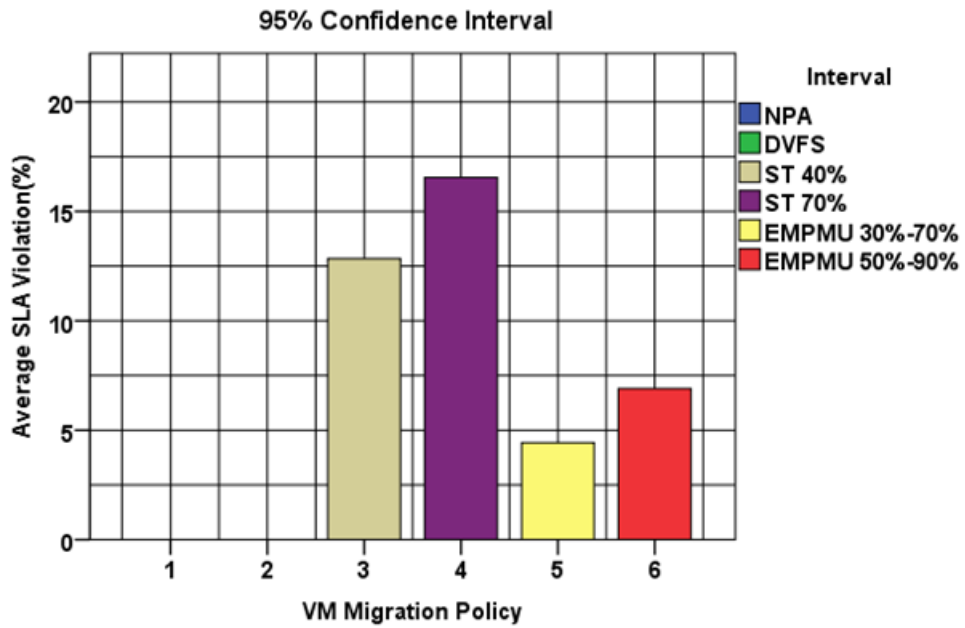


Fig. 5(d). Average SLA violations

5. Conclusion

Cloud computing offers unlimited resources over the internet based on pay-as-you go model. When number of users' increases, cloud data centers consume more energy thus decreasing the system performance. Heavy usages of cloud data centers should strictly maintain service level agreement for providing QoS. In this paper, we have addressed the problem of energy and SLA-aware resource allocation heuristics for reducing energy consumption and SLA violation. Proposed SLA-aware MBFD algorithm proved less SLA violation and minimum power consumption. This algorithm works based on current CPU workload and allocates resources without diminishing SLA violation. Moreover, EMPMU VM migration policy proved efficient resource allocation and reduced limited number of VM migrations than HPG and RC policies. Finally, EMPMU experimental results are compared with three VM migrations policies such as NPA, DVFS and ST. The proposed algorithm supports double threshold policy to allocate resources within upper and lower limit CPU utilization. Significant difference between simulation results are validated using two-way ANOVA test. The observation of each result is concluded by accept or reject hypothesis. However, efficient utilization of heterogeneous cloud resources performs 30% less energy consumption and 6% less SLA violations. Simulation results are evaluated using Cloudsim toolkit for repeated experiments. In the future scope of our research work, we have planned to implement our dynamic algorithms in Amazon EC2 to test different workload traces for better energy consumption and effective cost minimization.

Acknowledgements

My sincere thanks to Dr.G.M.Kadhar Nawaz MCA.,Ph.D, Director & Professor, Department of Master of Computer Applications, Sona College of Technology, Salem, India for his valuable guidance and constant support to do this research work.

References

- [1] <https://www.eia.gov/totalenergy/data/monthly/pdf/mer.pdf> [Article\(CrossRef Link\)](#)
- [2] <https://eta.lbl.gov/publications/united-states-data-center-energy> [Article\(CrossRef Link\)](#)
- [3] Mike Ebbers, Alvin Galea, Michael Schaefer, Marc Tu Duy Khiem., "The Green Data Center: Steps for the Journey," *IBM corporation*, 2008. [Article\(CrossRef Link\)](#)
- [4] Fei Cao, Michelle M. Zhu., "Energy Efficient Workflow Job Scheduling for Green Cloud," in *Proc. of IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum*, pp. 2218-2221, October 2013. [Article\(CrossRef Link\)](#)
- [5] Devendra Singh Thakur., "Energy Efficient Task Scheduling Algorithms in Cloud Data Center," *E-thesis*, pp. 1-26, September 2014. [Article\(CrossRef Link\)](#)
- [6] Ning Liu, Ziqian Dong, Roberto Rojas-Cessa., "Task Scheduling and Server Provisioning for Energy-Efficient Cloud-Computing Data Centers," in *Proc. of IEEE 33rd International Conference on Distributed Computing Systems Workshops*, pp. 226-231, December 2013. [Article\(CrossRef Link\)](#)
- [7] Hongyang Sun, Patricia Stolf, Jean-Marc Pierson., "Spatio-temporal thermal-aware scheduling for homogeneous high-performance computing datacenters," *Future Generation Computer Systems*, vol. 71, pp. 157-170, June 2017. [Article\(CrossRef Link\)](#)

- [8] Yuyang Peng, Dong-Ki Kang, FawazAl-Hazemi, Chan-Hyun Youn., “Energy and QoS aware resource allocation for heterogeneous sustainable cloud datacenters,” *Optical Switching and Networking*, vol. 23, no. 3, pp. 225-240, January 2017. [Article\(CrossRef Link\)](#)
- [9] Saurabh Kumar Garg, Chee Shin Yeo, Arun Anandasivam, Rajkumar Buyya., “Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers,” *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp.732–749, June 2011. [Article\(CrossRef Link\)](#)
- [10] Anton Beloglazov, Rajkumar Buyya., “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers,” *Concurrency and Computation: Practice and Experience, John Wiley & Sons, Ltd.* Vol. 24, no.13, pp. 1397-1420, September 2011. [Article\(CrossRef Link\)](#)
- [11] Jing Liu, Xing-Guo Luo, Xing-Ming Zhang, Fan Zhang and Bai-Nan Li., “Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm,” *IJCSI International Journal of Computer Science Issues*, vol. 10, no. 3, pp. 134-139, January 2013. [Article\(CrossRef Link\)](#)
- [12] Xiaocheng Liu, Yabing Zha, Quanjun Yin, Yong Peng, Long Qin., “Scheduling parallel jobs with tentative runs and consolidation in the cloud,” *The Journal of Systems and Software*, vol. 104, pp. 141–151, June 2015. [Article\(CrossRef Link\)](#)
- [13] Anton Beloglazov, Jemal Abawajy, Rajkumar Buyya., “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing,” *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, May 2012. [Article\(CrossRef Link\)](#)
- [14] Chia-Ming Wu, Ruay-Shiung Chang, Hsin-Yu Chan., “A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters,” *Future Generation Computer Systems*, vol. 37, pp. 141-147, July 2014.. [Article\(CrossRef Link\)](#)
- [15] Fabio D. Rossi, Miguel G. Xavier, Cesar A. F. De Rose., “E-eco: Performance-Aware Energy-Efficient Cloud Data Center Orchestration,” *Journal of Network and Computer Applications*, vol. 78, pp. 83-96, January 2017. [Article\(CrossRef Link\)](#)
- [16] Yuyang Peng, Dong-Ki Kang, Fawaz Al-Hazemi, Chan-Hyun Youn., “Energy and QoS aware resource allocation for heterogeneous sustainable cloud datacenters,” *Optical Switching and Networking*, vol. 23, no.3, pp. 225-240, January 2017. [Article\(CrossRef Link\)](#)
- [17] Ehsan Arianyan, Hassan Taheri, Saeed Sharifian., “Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers,” *Computers and Electrical Engineering*, vol. 47, pp. 222-240, October 2015. [Article\(CrossRef Link\)](#)
- [18] Jiyuan Shi, Junzhou Luo, Fang Dong, Jiahui Jin, Jun Shen., “Fast Multi-resource Allocation with Patterns in Large Scale Cloud Data Center,” *Journal of Computational Science*, May 2017. [Article\(CrossRef Link\)](#)
- [19] Mohammad-Hossein Malekloo, Nadjia Kara, May El Barachi, “An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments,” *Sustainable Computing: Informatics and Systems* vol. 17, pp. 9-24, March 2018. [Article\(CrossRef Link\)](#)
- [20] Ehsan Arianyan, Hassan Taheri and Vahid Khoshdel, “Novel Fuzzy Multi Objective DVFS-aware Consolidation Heuristics for Energy and SLA Efficient Resource Management in Cloud Data Centers,” *Journal of Network and Computer Applications*, vol. 78, pp. 43-61, January 2017. [Article\(CrossRef Link\)](#)
- [21] Zhou Zhou, Jemal Abawajy, Morshed Chowdhury, Zhigang Hu, Keqin Li, Hongbing Cheng, Abdulhameed A. Alelaiwi, Fangmin Li, “Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms,” *Future Generation Computer Systems*, August 2017, [Article\(CrossRef Link\)](#)
- [22] Abbas Horri, Mohammad Sadegh Mozafari, Gholamhossein Dastghaibyfar, “Novel resource allocation algorithms to performance and energy efficiency in cloud computing,” *The Journal of Super Computing*, vol. 69, no. 3, pp. 1445-1461, September 2014. [Article\(CrossRef Link\)](#)
- [23] K.R. Remesh Babu, Philip Samuel, “Interference aware prediction mechanism for auto scaling in cloud,” *Computers & Electrical Engineering*, December 2017. [Article\(CrossRef Link\)](#)

- [24] Zhibo Cao, Shoubin Dong., “An energy-aware heuristic framework for virtual machine consolidation in Cloud computing,” *Journal of Super Computing, Springer Science+Business Media New York*, vol. 69, no.1, pp. 429-451, July 2014. [Article\(CrossRef Link\)](#)
- [25] Ed Weber., IBM Systems Sales Manager, “Energy Efficient IT,” *IBM Corporation*, 2010. [Article\(CrossRef Link\)](#)
- [26] A. Sathya Sofia, P. GaneshKumar., “Multi-objective Task Scheduling to Minimize Energy Consumption and Makespan of Cloud Computing using NSGA-II,” *Journal of Network and Systems Management, Springer Science+Business Media, LLC*, vol. 26, no.2, pp. 463-485, April 2018. [Article\(CrossRef Link\)](#)
- [27] E. Elnozahy, M. Kistler, R. Rajamony, “Energy-efficient server clusters, Power-Aware Computer Systems,” in *Proc. of International Workshop on Power-Aware Computer Systems PACS 2002: Power-Aware Computer Systems* pp. 179–197, 2003. [Article\(CrossRef Link\)](#)
- [28] Martello, Silvano, Toth, Paolo., "Bin-packing problem, In: Knapsack Problems: Algorithms and Computer Implementations," *Chichester, UK: John Wiley and Sons*, ISBN 0471924202, 1990. [Article\(CrossRef Link\)](#)
- [29] M. Yue., “A simple proof of the inequality $FFD(L) < 11/9 OPT(L) + 1$, for all L for the FFD bin-packing algorithm,” *Acta Mathematicae Applicatae Sinica*, vol. 7, no. 4, pp. 321–331, October 1991. [Article\(CrossRef Link\)](#)



Mrs.K.Sutha is currently pursuing a doctoral degree in Computer Science from Bharathiar University, Coimbatore, India and working as an Assistant Professor in Dr.MGR Janaki College of Arts and Science for Women, Chennai, India. She received her Master degree in Computer Applications from Anna University, Chennai, India in 2009. Her research interest includes energy-efficient resource scheduling and load balancing in cloud computing. She is a Lifetime Member of Indian Society for Technical Education (ISTE), New Delhi.



Dr.G.M.Kadhar Nawaz is currently working as a Director in Master of Computer Applications, Sona College of Technology, Salem, India. He has more than 20 years of teaching and research experience in reputed engineering colleges. He has published around 50 research papers in highly indexed SCI, SCIE and SCOPUS journals. He received his Ph.D in Computer Science from Periyar University, Salem, India in 2007 and his area of research includes digital image processing and steganography. He is a Professional Member of Indian Society for Technical Education (ISTE), New Delhi.