IETE Technical Review

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/titr20

Energy-Efficient Service Allocation Techniques in Cloud: A Survey

Sambit Kumar Mishra , Sampa Sahoo , Bibhudatta Sahoo & Sanjay Kumar Jena

To cite this article: Sambit Kumar Mishra , Sampa Sahoo , Bibhudatta Sahoo & Sanjay Kumar Jena (2020) Energy-Efficient Service Allocation Techniques in Cloud: A Survey, IETE Technical Review, 37:4, 339-352, DOI: 10.1080/02564602.2019.1620648

To link to this article: https://doi.org/10.1080/02564602.2019.1620648

Published online: 30 May 2019.

IETE TECHNICAL REVIEW
2020, VOL. 37, NO. 4, 339–352
https://doi.org/10.1080/02564602.2019.1620648

REVIEW ARTICLE

Energy-Efficient Service Allocation Techniques in Cloud: A Survey

Sambit Kumar Mishra, Sampa Sahoo, Bibhudatta Sahoo and Sanjay Kumar Jena

Computer Science & Engineering Department, National Institute of Technology, Rourkela, India

ABSTRACT
The demand for cloud computing infrastructure is increasing day by day to meet the requirement of small and medium enterprises. The data center-centric cloud technology has a high share of energy consumption from the IT-industry. The amount of energy consumption in a data center depends on the allocation of user service requests to virtual machines running on the different host. Minimization of energy consumption in the data center is a significant issue and addressed by optimal allocation of cloud resources. In this paper, we have discussed how service allocation strategies have been used to optimize the energy consumption in a cloud system. A generalized system architecture is presented based on which we define the service allocation problem and energy model. Further, we present the taxonomy of various energy-efficient resource allocation techniques found in the literature. In the end, various research challenges related to the energy-efficient service allocation in cloud are discussed.

KEYWORDS
Cloud computing; Energy-aware; QoS; Service allocation; Virtualization; VM

1. INTRODUCTION

Recently, in the field of distributed computing system innovation, Cloud Computing has emerged as a critical worldview that gives versatile and dynamic virtual resources to meet the client's requirement through the web. Cloud computing is a conveyance display or a model that conveyed the on-interest registering resources from applications to the server over the Internet on a pay-for-utilization premise. Widely accepted definition of cloud computing is defined by the National Institute of Standards and Technology (NIST) [1] where a popular model is explained called 5-4-3 model. This model constitutes five essential characteristics of cloud computing, four delivery models, and three service models. The computing paradigm is presented as a model where everything could be obtained as a service called Everything-as-a-Service (XaaS) model [1]. XaaS also known as Anything-as-a-Service and includes communication-as-a-service (CaaS), infrastructure-as-a-service (IaaS), software-as-a-service (SaaS), platform-as-a-service (PaaS), databases-as-a-service (DBaaS), security-as-a-service (SaaS), identity-management-as-a-service (IMaaS), desktop-as-a-service (DaaS), and so on [2]. The cloud service provider (CSP) provides these services to the clients and abides by certain Service Level Agreements (SLAs). The CSP assures the availability of various services as required by the client in cost or time frame. The SLA assures to serve the user to meet its requirement. In the cloud computing paradigm, the CSP deals with a pool of computing resources (*e.g.* servers, storage, networks, applications, services, *etc.*) to guarantee user needs. Cloud computing uses virtualization technology and multi-tenancy model to provide services to the user in terms of infrastructure, platform, and software [1]. Every CSP provides services through one or multiple data centers geographically distributed across the Internet. The user can procure and discharge these services as per his requirement through self-service interfaces. Service utilizations are consequently metered, permitting users to pay for the services that are utilized. In this environment, the CSP picks up a chance to increase their benefits through the economies of scale and users try to access more resources without any additional cost.

The demand for cloud computing infrastructure is increasing to meet the requirement of small and medium enterprises using a service delivery model over the Internet. More than 60% of the small and medium industries have adopted cloud computing technologies for performing various IT-related services. In recent time, close to 50% of total companies have gone for cloud computing. The adoption of cloud computing by the small and medium enterprises results in the reduction of infrastructure cost. Cloud computing uses scalable resources and lifts the overhead of software and hardware setup from the users which play a vital role in cost reduction [3]. The major motivations to adopt cloud computing technology by IT industries are stated in Figure 1.

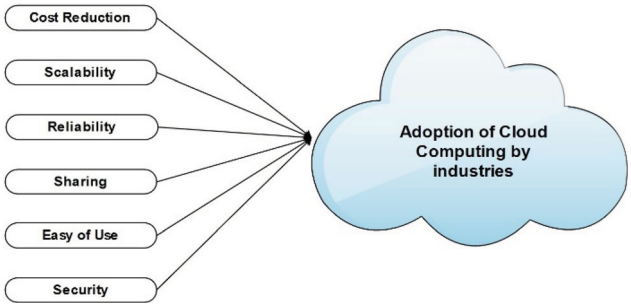

Figure 1: Reasons to adopt cloud computing

Major research topics in cloud computing include service delivery model, locality/energy/reliability-aware scheduling, workflow scheduling, *etc.* [4]. Load on a cloud computing environment are the result of users service requests that are executed by the host. Hence, service allocation is one of the key issues to optimize energy consumption. Therefore, scheduling or allocation of services in a cloud system plays an important role with different objectives. The services or the user requests are submitted to the CSP to perform the required operations. If the CSP can perform the execution with the specified SLAs, then the task is executed with the help of cloud resources. The mapping problem of cloud tasks or services to the cloud resources or VMs is the service allocation problem. In this paper, we have focused our study on energy-aware service allocation techniques in the cloud. The amount of energy consumption in a data center depends on the allocation of user service requests to virtual machines (VMs) running on different host. Various researchers have successfully applied heuristic techniques to address the service allocation problem in cloud environment [4–7]. Always the scheduling technique is applied to the consumers (or cloud users or service requests) and the cloud resources (or VMs) available in the data center. So, the strategy or the allocation algorithm influences either the consumers or resources or both [8]. Some researchers are also argued the distinction of terms allocation and scheduling for the discussed problem. The term allocation points to resource allocation (*i.e.* resource point of view) and scheduling seen as a consumer's point of view. Here, both the terms representing the same general mechanism.

Various user requests or tasks coming to the cloud environment can be real-time, *i.e.* time-bound tasks or non-real-time. Tasks can be further categorized as CPU or I/O–intensive. In the case of a real-time task, it is the tricky task to satisfy both deadline and energy efficiency constraint. Researchers presented various allocation techniques for the non-real-time tasks [9–13] as well as for the real-time tasks [14–22]. These techniques are also applicable to CPU-intensive tasks [9–12, 14–22], and I/O-intensive tasks [9,14,22]. After receiving the service request from the user, the resources (*e.g.* CPU, network bandwidth, main memory, secondary storage, *etc.*) of the cloud data center are virtualized. The cloud computing system is powerful mostly because of this virtualization technique. This virtualization technique is to create virtual instances of a device or resource, where the framework partitioned the resource into multiple execution environments virtually. The technology support virtualization is a virtual machine monitor (VMM) or hypervisor that separates the computer environments from the actual physical infrastructure. Energy management techniques in the smart grid can be estimated with the help of cloud computing applications [23]. Energy efficiency is one of the challenges to provide service with acceptable QoS for the tasks that use a large amount of interactive data or information [24]. The conservation of energy, as well as the proper management of physical resources, can be achieved by an efficient mapping between the VMs and physical hosts. We have discussed two taxonomies that present a detailed and complete look at the energy minimization issue in the cloud environment. Such detail is necessary to support meaningful comparisons of the different energy model for energy-aware service allocation problem to have an optimal energy model. This paper can assist the CSPs to have a robust task allocation framework to save energy.

1.1 Motivation

The rapid growth in the user demand in the cloud computing system increases the number of data centers, which in turn boosts energy utilization. *Greenpeace* estimation on 2015 states that the number of the data center increases to 21% in 2017 where *i.e.* 15% in 2012 [25]. From the literature, it has found that the data center infrastructure causes over 70% of the total heat generation [26]. The higher consumption of energy also leads to increase in carbon dioxide (CO_2) emissions that have negative impacts on the environment. The sources of energy release CO_2, the main ingredient of the "greenhouse gasses" into the atmosphere, which is responsible for global warming. The CSP uses single or multiple data centers to provide services to the user through the Internet. A majority of the cloud computing research communities are looking forwards to optimize the power consumption of the data center. In a cloud environment, resources of a host are shared in between all the VMs on that host. So, the energy consumption of a host cannot be determined by considering a single VM, and for this, we should go for the consolidated condition of all the VMs on that host. The allocation of service requests to

a VM running on the different physical host mapped in a manner that the over provisioning and under provisioning of resources are prohibited. Over provisioning may cause high energy consumption and underutilization of the computing systems. Under provisioning causes loss of customers and revenues. In both cases, there is wastage of energy for under-utilized resources. So, an energy-efficient resource allocation is the significant need for the cloud computing system. To design an energy-aware cloud data center, researchers proposed various strategies such as optimal VM placement, or service allocation, or data replication, or by using power-aware management scheme like dynamic voltage frequency scaling (DVFS) [27].

1.2 Contributions

In this paper, we provide a systematic overview of energy-efficient resource allocation techniques in the cloud computing system. An analytical cloud system model is discussed to compute the total energy consumed. Computation of total energy consumption in the cloud system illustrated representing service request with the help of Expected Time to Compute (ETC) matrix. Our principal contributions to the research are:

- We present a cloud system model that includes data center model, host model, VM model, and service model to explain energy-efficient resource allocation.
- An analytical study presented to explain service allocation problem and total energy consumed in a cloud environment.
- Energy-efficient service allocation techniques discussed under two different categories: one is based on the input cloud task and another based on the migration in the cloud or virtualized server.
- A comparison of various energy-efficient techniques is presented based on the migration of tasks or VMs in the cloud system.

1.3 Organization

We organize the remaining of this paper as follows. In Section 2, we briefly discuss the general cloud system architecture. In Section 3, the proposed cloud system model along with different models for the cloud and also the service allocation problem are presented including an illustration of example for a basic service allocation policy. A brief description of various energy-efficient techniques for service allocation in the cloud and their discussion along with comparison are presented in Section 4, followed by a conclusion remark in Section 5.

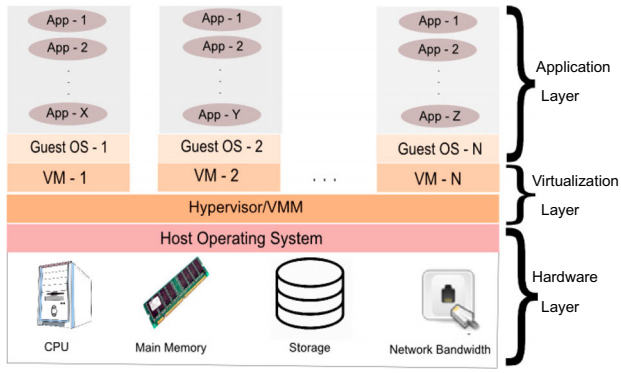

Figure 2: Single host architecture in cloud

2. SYSTEM ARCHITECTURE

Several features are considered in the design of the system model, namely cost, complexity, speed, system portability, security, *etc.* Cloud computing architecture varies from the traditional distributed system architecture in the following aspects: (1) it is highly scalable, (2) it is an abstract entity and addresses distinct levels of services to the cloud consumer, (3) economies of scale, and (4) it delivered on demand dynamic services through virtualization. One of the system architecture of a single host in cloud environment followed by many researchers is shown in Figure 2.

The hardware layer consists of the raw hardware resources (processor, main memory, secondary storage, and network bandwidth) which are virtualized. VMM or hypervisor (such as Xen, VMware, UML, and Denali) will act as an interface between the guest operating system and VMs. This VMM allows multiple operating systems to run applications on a single hardware platform concurrently. Different numbers of heterogeneous applications are running on each guest operating system or VM which is the basic unit to execute an application or a service request.

3. CLOUD SYSTEM MODEL

In general, a cloud system model represents IaaS that manages the physical resources and the virtual resources to support the requirements of cloud users. To enhance the energy efficiency of the system, the researcher contributes more attention towards the designing of the system model. A general system model is shown in Figure 3. We have explained the cloud system model with the help of data center model, host model, VM model, and service model.

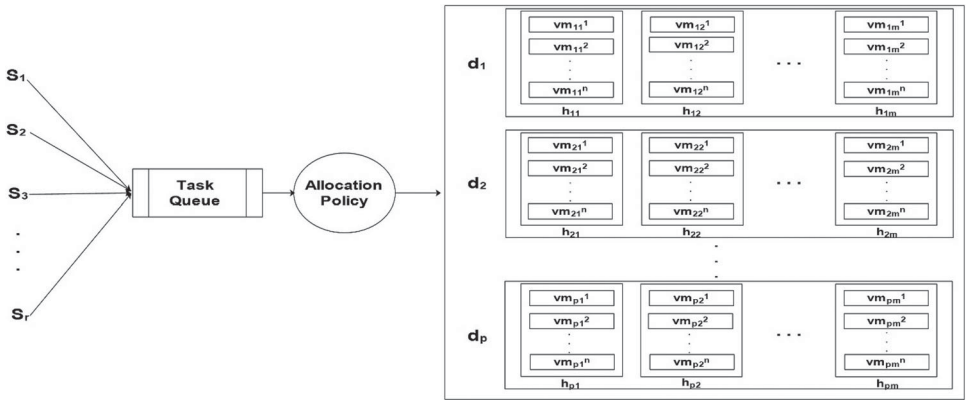

Figure 3: A cloud system model

3.1 Data Center Model

Several physical servers constitute a data center and there is an interconnection of high speed LAN-network and high bandwidth link to the internet from each physical server. The cloud can be modeled with the resources across the fine set of heterogeneous data centers $D = \{d_1, d_2, \ldots, d_p\}$. Each data center has 9 tuples, *i.e.* $d_i = \{id_i, arch_i, VMM_i, HL_i, TZ_i, CP_i, CM_i, CS_i, CB_i\}$, where id_i is the data center identification of data center d_i, $arch_i$ is the system architecture, VMM_i is the virtual machine manager, HL_i is the host list running on data center d_i, TZ_i is the time zone till the resources located, CP_i is the cost of using processor as resource, CM_i is the cost of using main memory as resource, CS_i is the cost of using storage as resource, and CB_i is the cost of using bandwidth as resource.

3.2 Host Model

The cloud computing environment is with m heterogeneous physical hosts in data center d_i, $H_i = \{h_{i1}, h_{i2}, \ldots, h_{im}\}$. Each host has 5 tuples, *i.e.* $h_{ij} = \{id_{ij}, PE_{ij}, M_{ij}, \lambda_{ij}, S_{ij}\}$, where id_{ij} is the host identification of host h_{ij} running on the ith data center, PE_{ij} is the processing element list, M_{ij} is the host memory size, λ_{ij} is the total bandwidth of host, and S_{ij} is the secondary storage of the host.

3.3 VM Model

A VM is an emulation of a particular host. There are n number of VMs running on the jth host of the ith data center, $V_{ij} = \{vm_{ij}^1, vm_{ij}^2, \ldots, vm_{ij}^n\}$. A VM can be modeled as $vm_{ij}^k = \{id_{ij}^k, Bid_{ij}^k, PS_{ij}^k, PE_{ij}^k, M_{ij}^k, {}_{ij}^k, S_{ij}^k, VMM_{ij}^k\}$, where id_{ij}^k is the virtual machine identification of vm_{ij}^k, Bid_{ij}^k is the broker identification, PS_{ij}^k is the processing speed of vm_{ij}^k in terms of million instructions per second (MIPS), PE_{ij}^k is the number of processing elements for vm_{ij}^k, M_{ij}^k is the main memory size of vm_{ij}^k, ${}_{ij}^k$ is the bandwidth of vm_{ij}^k, S_{ij}^k is the secondary storage of vm_{ij}^k and VMM_{ij}^k is the VMM on which vm_{ij}^k is running.

3.4 Service Model

A service request from the user is assigned to one or more VMs. Let 'S' be the set of 'r' services $\{S_1, S_2, \ldots, S_r\}$. Each service S_i can be modeled as $S_i = \{Sid_i, W_i, PE_i, Flsz_i, Flos_i, CPU_i, M_i, \lambda_i\}$, where Sid_i is the service identification of service S_i, W_i is the workload of service S_i in terms of million instruction (MI), PE_i is the number of required processing elements for the service S_i, $Flsz_i$ is the file size of service S_i, $Flos_i$ is the size of the output file, CPU_i is the CPU time requirement for the service S_i, M_i is the main memory requirement for the service S_i and λ_i is the bandwidth requirement of service S_i.

The cloud user submits their requests to the CSP, where a task queue is maintained. These tasks require cloud resources for the execution purpose. Therefore, an efficient allocation policy is needed for the distribution of tasks to the cloud resources (VMs). Allocation policy plays the key role in energy-efficient resource allocation to various service requests in the cloud.

3.5 Service Allocation Problem and Energy Model

A cloud is designed to server service requests originate from the multiple users over the Internet. The challenge of allocating service requests to a set of VMs running on different hosts while achieving the terms and conditions stated in the SLAs and without degrading the QoS is referred to as the service allocation problem [10,13,16,28,29]. The service allocation problem is a

well-known NP-complete problem [30,31]. To design an energy efficient solution for the service allocation problem, we need to define certain assumptions in our defined model. For the allocation of tasks (services), we have used the ETC model proposed by Ali *et al.* [32]. This task model considered the heterogeneity of computing resources and input tasks. The ETC matrix contains the time required to execute each task in different VMs. Here, all the service requests are independent, heterogeneous, and non-preemptive in nature. VMs are heterogeneous with reference to their resources. It shows system heterogeneity of the actual system that varies significantly regarding their processor speed, main memory size, and other resources. We consider that all the services execute successfully and the waiting time of a service request is negligible.

For simplicity, we have considered a single datacenter d_1 that consists of a set of $H_1 = \{h_1, h_2, \ldots, h_m\}$, m independent heterogeneous, uniquely addressable computing entity (hosts) in a cloud system. We have a set of VM sets: $\{V_1, V_2, \ldots, V_m\}$ of $(n \times m)$ heterogeneous VMs, where $Vj = \{vm_{j1}, vm_{j2}, \ldots, vm_{jn}\}$, and each host has n VMs, and the n value varies from host to host. Let there be $S = \{S_1, S_2, \ldots, S_r\}$, r number of heterogeneous services, where each service S_i has a service length L_i in terms of a MI. ES^i_{jk} is the expected time to compute the service S_i on virtual machine vm_{jk}, $1 \leq j \leq m, 1 \leq k \leq n$. Each VM is capable of executing all types of services. This can be represented by an ETC matrix of size: $r \times (n \times m)$, where the number of services and the number of VMs are r and $(n \times m)$, respectively. In the ETC matrix, the elements along a row indicate the execution time (ET) of a given service on different VMs in terms of the second. Each virtual machine vm_{jk} has a processing speed P_{jk} in terms of MIPS. Then, the ES^i_{jk} is $L_i \div P_{jk}$, where L_i is the service length of S_i and $1 \leq i \leq r, 1 \leq j \leq m, 1 \leq k \leq n$.

The ETC matrix elements are filled based on different cloud resources. Most of the researchers considered the CPU resource as a parameter to design ETC matrix. Similarly, we have also considered the CPU as the primary resource. Variation in the ETC matrix entries represents the VM and service heterogeneity; ultimately satisfying the heterogeneity property of the cloud model. The allocation of services can be categorized in two ways: one static and another dynamic allocation. In the static allocation, the complete system information, as well as task information, known prior. However, in dynamic allocation, according to the objectives (reducing energy consumption), the allocation is changed periodically. This study discusses the dynamic allocation in the proposition of the energy model. When a VM processing a task, it is in the active state or in the idle state. Energy consumed by the virtual machine vm_{jk} in active state is represented as β_{jk} Joules/MI, where $\beta_{jk} = 10^{-8} \times (P_{jk})^2$ [29,33]. The assignment of a service to a VM is given by

$$X^i_{jk} = \begin{cases} 1, & \text{if } S_i \text{ is allocated to } vm_{jk} \\ 0, & \text{if } S_i \text{ is not allocated to } vm_{jk} \end{cases} \tag{1}$$

where $i = 1, 2, \ldots, r$, $j = 1, 2, \ldots, m$, and $k = 1, 2, \ldots, n$.

Total ET of all the services assigned to kth VM of the jth host is given in the following equation:

$$ET_{jk} = \sum_{i=1}^{r} X^i_{jk} \times ES^i_{jk} \tag{2}$$

The total time taken by the jth host (ETH_j) to execute the assigned tasks to their VMs is given in the following equation:

$$ETH_j = \sum_{k=1}^{n} ET_{jk} \tag{3}$$

One of the significant performance matrices is the makespan of the system. Makespan (M) is the maximum ET among all the physical hosts, *i.e.*

$$M = \text{Maximum } (ETH_j), \quad 1 \leq j \leq m. \tag{4}$$

The energy consumption of a virtual machine vm_{jk} in the active state in terms of Joules per MI is given in the following equation:

$$EV_{jk} = (ET_{jk} \times \beta_{jk}) \tag{5}$$

The energy consumption of the jth host in the active state is given in the following equation:

$$EH_j = \sum_{k=1}^{n} EV_{jk} \tag{6}$$

The average energy consumption of the jth host in the active state is given in the following equation:

$$AEH_j = EH_j / ETH_j \tag{7}$$

The energy consumption of the idle state machine is 60% of the energy consumption of the active state machine [5]. The average energy consumption of the jth host in the

Procedure 1: Calculation of Energy Consumption and Makespan.
Input: ETC matrix
Output: Makespan (*M*), Energy Consumption (*E*)

Step 1: Find the allocation matrix (*X*) using the ETC matrix and scheduling policy (*e.g.* FCFS, RANDOM, *etc.*) using Equation (1).
Step 2: Calculate Makespan (*M*) of the host using Equation (4).
Step 3: Find energy consumption of individual VM (*EV*) and host machine (*EH*) in the active state using Equation (5) and Equation (6), respectively.
Step 4: Calculate average energy consumption (*AEH*) of the host in the active state from Equation (7).
Step 5: Find energy consumption of the host machine (*IEH*) in the idle state using Equation (8).
Step 6: Calculate the total energy consumption (*E*) of the host machine using Equation (9).
Step 7: Overall energy consumption of the cloud system (ε) is funded using Equation (10).

idle state is given in the following equation:

$$IEH_j = 0.6 \times AEH_j \tag{8}$$

The total energy consumption of the *j*th host is given in the following equation:

$$E_j = EH_j + (M - ETH_j) \times IEH_j \tag{9}$$

So, the total energy consumption of the cloud system is given in the following equation:

$$\epsilon = \sum_{k=1}^{m} E_j \tag{10}$$

The objective is to minimize the energy as given in Equation (8) by using appropriate resource allocation techniques.

In general, the service request (task) from the client has different computing and input/output (I/O) requirements. Considering the input tasks to be CPU-intensive and I/O-intensive, the total energy consumed by the data center can be computed as follows.

3.6 Scheduling Procedure

Procedure 1 describes the steps used to find the values of different performance metrics. In our case, metrics are energy consumption and makespan.

To illustrate the procedure, an example is presented in this section. Let there are 6 service requests (tasks) {S_1, S_2, S_3, S_4, S_5, S_6}, 3 hosts {H_1, H_2, H_3}, and 3 VMs {V_1, V_2, V_3}. For simplicity, we have considered each host has a single VM. Table 1 presents the task length in MI and Table 2 presents the processing speed of VMs in MIPS. Table 3 presents the ETC matrix and a random allocation is considered which is shown in Table 4. The value is 1 in Table 4 when the task is allocated to the corresponding VM, otherwise, that value is 0.

Table 1: Task set with task length

Service-id	S_1	S_2	S_3	S_4	S_5	S_6
Task length	3000	4000	8000	10000	6000	5000

Table 2: VM set with processing speed of VM

VM-id	V_1	V_2	V_3
VM Speed in MIPS (P_{jk})	1000	1200	1500

Table 3: ETC matrix

	V_1	V_2	V_3
S_1	3	2.5	2
S_2	4	3.33	2.67
S_3	8	6.67	5.33
S_4	10	8.33	6.67
S_5	6	5	4
S_6	5	4.17	3.33

Table 4: Allocation matrix

	V_1	V_2	V_3
S_1	1	0	0
S_2	0	1	0
S_3	0	0	1
S_4	1	0	0
S_5	0	0	1
S_6	0	1	0

Table 5: Execution time of tasks in VMs

	ET_1	ET_2	ET_3
Execution Time (ET)	13	7.5	9.33

After execution of all tasks, we get the makespan (*M*) of the system as 13 s using Equation (4) as shown in Table 5.

The energy consumption of each VM is calculated using Equation (5) and shown in Table 6. Since each host has a single VM, the active state energy consumption of host (EH_i) and EV_i values are the same as in Equation (6), $i = 1, 2, 3$.

The calculation of average energy consumption of a host (AEH_i, $i = 1, 2, 3$) using Equation (7) and the idle state energy consumption of the host (IEH_i, $i = 1, 2, 3$) using Equation (8) are listed in Table 7.

The total energy consumption of each host is calculated using Equation (9) and we get $E_1 = 0.13$, $E_2 = 0.1553$,

Table 6: Energy consumption of VMs

	EV_1	EV_2	EV_3
Energy Consumption of VM (EV)	0.13	0.108	0.21

Table 7: Average and idle state energy consumption of host

AEH_1	AEH_2	AEH_3	IEH_1	IEH_2	IEH_3
0.01	0.0144	0.0225	0.006	0.0086	0.0135

and $E_3 = 0.2595$. So, the total energy consumption of the system is the sum of energy consumption of all hosts, *i.e.* 0.5448 Joules.

4. ENERGY-EFFICIENT TECHNIQUES

Currently, most of the techniques of resource allocation in the cloud or virtualized server attract numerous researchers working on this hotly debated issue. Among those researches, the energy-aware algorithm for scheduling of task (service) is an emerging research issue which has been addressed by different researchers [5,6,15,22,34–36]. To make the cloud system energy-efficient, researchers presented various energy models in their work. In the literature, various energy-aware resource allocation for homogeneous as well as the heterogeneous environment are presented, and some of those are multi-objective also. Giacobbe *et al.* [35] have focused on current research trends on energy saving and also explained some energy-related terminologies such as energy efficiency, energy sustainability, energy cost-saving, and renewable energy. Energy efficiency is nothing but the minimization of the requirement of energy to provide services. Energy cost-saving is a mechanism to reduce the monetary costs for energy usage. The energy sustainability concept is that the product not only meets the present requirement, but also meets the future need. They distinguished the energy sources in two ways: renewable energy and non-renewable energy. The taxonomy for the research work based on the input cloud task and based on the migration of tasks or VMs are presented below.

4.1 Taxonomy Based on Input Task

The input tasks to the cloud environment may be associated with a deadline or without a deadline. The real-time tasks are deadline-bound. It can be further classified as CPU-intensive (task with more CPU time) and I/O-intensive task (task with more I/O requirement). Some of the research works based on real time and non-real time tasks are presented in Figure 4.

Energy-efficient techniques for the real-time task have to deal with both deadlines (time) and energy issues. If the deadline is the primary consideration, then the energy savings may not be up to the mark, or if we prefer for energy conservation, then the deadline may be missed. So, the energy-efficient techniques for time-constraint task need to be handled carefully to meet the SLA. Otherwise, the CSP has to pay the penalty to the user. A brief review of researchers finding to deal with service allocation is presented for the real-time task to reduce the energy requirements in cloud computing environment.

4.1.1 SOCCER

The SOCCER (Self-Optimization of Cloud Computing Energy-efficient Resources) technique was proposed by Singh *et al.* [14]. This is an autonomic mechanism for the energy-aware cloud system considering the heterogeneous cloud workloads. The considered input tasks are CPU-intensive and memory (I/O)-intensive. The total energy consumed "*E*" for executing finite number of task can be calculated in [14] as

$$E = E_D + E_T + E_M + E_N \tag{11}$$

where E_D represents the energy consumption due to datacenter's; E_T represents the energy consumption due to switching equipment; E_M represents the energy consumption due to the storage device; and E_N represents the energy consumption due to other parts.

Singh *et al.* [14] have provided algorithms for resource scheduling with the aim of energy optimization. They have estimated the actual energy consumption of the workload before execution. If the actual required energy value is less than the threshold value of energy consumption, then the Resource Executor (*RE*) will execute workloads otherwise *RE* will make alert for rescheduling of resources. One of the drawbacks of this technique is that for some cases, the same workload is rescheduled several times which violate the system stability.

4.1.2 EESUB

Calheiros and Buyya have proposed an Energy-Efficient Scheduling of Urgent Bag-of-Tasks (EESUB) technique for reducing the power expected to execute CPU-intensive urgent bag-of-tasks on the cloud system [15]. They have used DVFS with modern processors to run the CPU at the least voltage level that facilitates the application to perform before the deadline. They gave more priority to the task having smaller deadline value. In the proposed scheme, the frequency level is set for each VM to speed up or slow down the CPU core allotted to each VM. Therefore, according to the frequency level and the number of CPU cores, the energy consumption is monitored. However, no energy model is explained properly in this contribution [15].

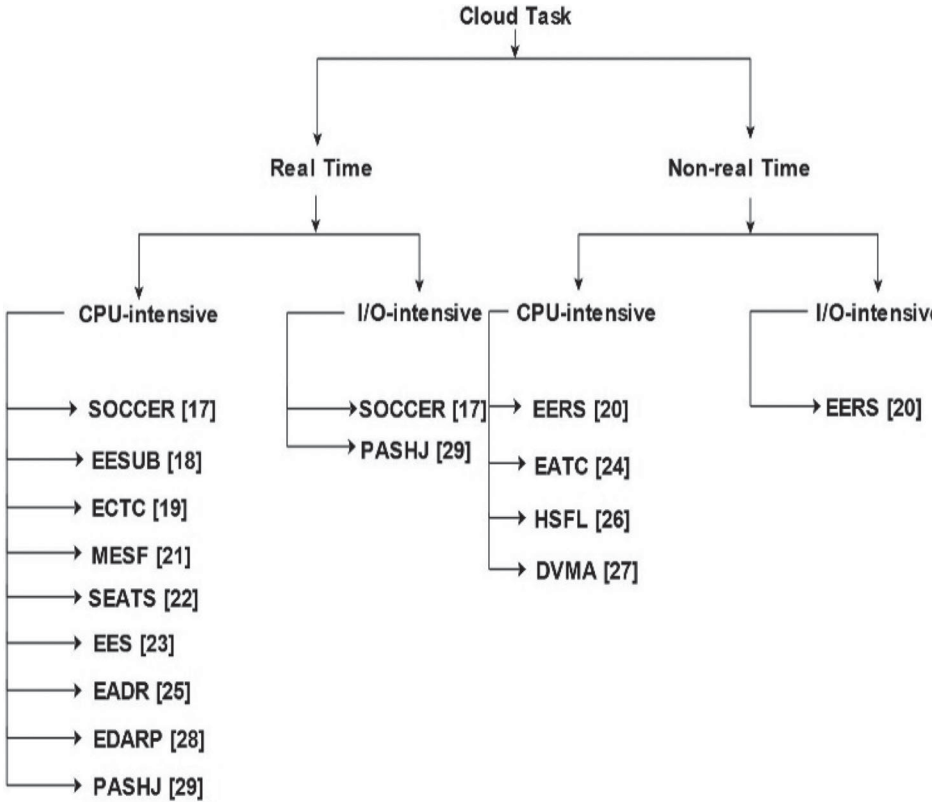

Figure 4: Taxonomy based on input task

4.1.3 ECTC and MaxUtil

Lee and Zomaya have presented two energy-aware task consolidation heuristic algorithms [16], namely Energy-Conscious Task Consolidation (ECTC) and Maximize Resource Utilization (MaxUtil) in the cloud environment. This work focuses towards resource utilization and energy consumption of homogeneous resources during task execution in the cloud. They have considered processing times as hard deadlines. The energy model used in [16] is based on the processing time and resource utilization for the execution of task, and is formulated as

$$E_i = (\rho_{\max} - \rho_{\min}) \times U_i + \rho_{\min} \tag{12}$$

where U_i is the utilization of the ith resource, $U_i = \sum_{j=1}^{n} u_{i,j}$. Here, the total number of tasks is n and $u_{i,j}$ is the usage of the ith resource by the jth task, ρ_{max} is the power consumption at the peak level (*i.e.* 100%) CPU utilization, and ρ_{min} is the minimum power consumption. Both the heuristics follow similar steps with a difference in their cost functions. They have stated that the migration of task causes more energy consumption of the system as compared to the system considering without migration of tasks.

4.1.4 MESF

A greedy task scheduling model, Most Efficient Server First (MESF), proposed by Dong *et al.* [17] is used to schedule the real-time task to the most energy efficient servers of a data center. They have proposed the *most efficient server first* scheduling scheme to find the smallest number of active servers, whereas maintaining the data center response time within a deadline. The energy consumption (E) is formulated as a function of some active servers and represented as an integer programming problem with the intention of minimizing energy consumption as defined in the following equation:

$$E = \sum_{i=1}^{V} \sum_{j=1}^{M} M_{i,j} \rho_{i,j} x_{i,j} \tag{13}$$

In Equation (16), $\rho_{i,j}$ is the power consumed by server S_j to complete task type-i, $\mu_{i,j}$ is the average processing time of type-i task by S_j, $x_{i,j}$ is the number of type-i tasks assigned to S_j, V is the total number of task types, and M is the number of active servers in the data center. In this case, energy conservation is achieved at the cost of longer response time.

4.1.5 SEATS

A VM scheduling algorithm for the real-time task: smart energy-aware task scheduling (SEATS) was proposed by

Hosseinimotlagh *et al.* [18]. The technique helps to compute the best utilization level of a server to perform a certain number of instructions with the aim of minimization of energy consumption of the host. The host reaches its optimal utilization level through faster execution of its VMs, and idle hosts are turned off to save power. The total power of a server is expressed as

$$P = P_{dynamic} + P_{static} \tag{14}$$

where static power $P_{static} = \alpha P_{max}$ and dynamic power is $P_{dynamic}(u) = (P_{max} - P_{static})u^2$. u is the node utilization at a given time. P_{max} is the power consumption of a host when it works at maximum utilization and α is the rate of the static power of a server to its maximum power $(0 \leq \alpha \leq 1)$. The SEATS algorithm aims to attain the optimal level of utilization by allowing more computing power to VMs. It gives more priority to a lengthy task with short deadlines.

4.1.6 EES

The Energy Efficient Scheduling (EES) technique was proposed by Garg *et al.* [19], where a sub-optimal scheduling scheme that employs heterogeneity over multiple data centers for a CSP has been discussed. They have considered various energy saving factors such as energy cost and CO_2 emission rate which varies across different data center based on their location, and architectural design. As data centers are located in different geographical areas with different CO_2 emission rates. Here, CPUs and cooling systems are considered to be optimized so that the energy can be saved. They have considered homogeneous processing units within a data center.

4.1.7 EADR

Beloglazov *et al.* have presented Energy-aware Allocation of Data center Resources (EADR) scheme for a cloud computing framework [20]. This allocation strategy is mainly concentrated on the placement and selection of VMs. Placement of VMs on different hosts using bin-packing problem and then optimized the allocation of VMs to hosts using the best fit decreasing algorithm. VMs selection is done for the migration of VMs from one host to another host. If the CPU utilization of a host is weaker than the least threshold utilization, then migrate all the VMs running on that host and switched the host to sleep mode. In other schenario, if the CPU utilization of a host is over the higher threshold of utilization, then select VMs using some effective policy and migrate those VMs. Since, change in workload varies the CPU usage at the different time, the total energy consumption of a host is defined as an integral of the power consumption function over a given period. The simulation results show in favor of dynamic reallocation of VMs according to the current CPU utilization in comparison to static resource allocation schemes.

4.1.8 EDARP

Gao *et al.* have provided Energy and Deadline-Aware Resource Provisioning (EDARP) approach in the cloud environment [21]. They have modeled the user requests or workload using directed acyclic graphs. The power consumption of a host D_x includes both the static power consumption $(P^x_{static}(t))$ and the dynamic power consumption $(P^x_{dynamic}(t))$ at time t. The static power consumption is constant and if CPU utilization value is 0, then there is no power consumption. They have presented two states of dynamic power consumption and considered a threshold value as 70%. The two states are (1) CPU utilization value is greater than the threshold value and (2) CPU utilization value is less than the threshold value. The total energy consumption (E) of M servers is given in the following equation:

$$E = \sum_{x=1}^{M} \left\{ \sum_{t=1}^{L_{\max}} \{P^x_{statics}(t) + P^x_{dynamic}(t)\} \right\} \tag{15}$$

where $L_{\max}$ is the upper bound of length of all the applications.

4.1.9 PASHJ

Cano *et al.* have explained some energy-based job model and present a model: Power-Aware Scheduling for Heterogeneous Job (PASHJ) [22]. In the job model, they have considered the release time and processing time of independent jobs and also characterizes job type as CPU intensive, memory intensive, I/O intensive, network intensive, *etc.* They introduced a hybrid model which includes the power consumption of individual applications and their combinations. The power consumption of a processor at a particular time consists of idle state power consumption and active state power consumption. They have provide a sequence of power consumed due to a different range of processor utilization. They have addressed the two-level scheduling approach. In the first one, they have a greedy acceptance policy to accept a task and then provide services to that task. In the second approach, they have used a preemptive earliest due date algorithm, which assigns priority to tasks based on their deadlines and then executes accordingly. They have not considered heterogeneity for user requests and also the performance evaluation of their proposed technique is missing.

As the non-real-time tasks are not time-bound, we mainly concentrate on energy savings as compared to the real-time task. Following are some of the techniques used

by the researchers to reduce the energy consumption of non-real-time task execution in the cloud.

4.1.10 EERS

The Energy Efficient Resource Scheduling (EERS) technique was proposed by Fayyaz *et al.* [9] to achieve energy efficiency through VM/task consolidation. First, the tasks are assigned to suitable servers satisfying the required constraints. VMM is used for finding under-utilized, partially filled, over-utilized, and empty servers. These idle and under-utilized servers are turned off using the dynamic threshold voltage scaling technique to save power. They have also used the same power consumption model as EADR [20]. In their proposed VM consolidation algorithm [9], servers are created earlier, and after that according to the task requirements, VMs are created. In their algorithm, the over-utilized servers are removed, but they have not explained about all allocated tasks to that removed server.

4.1.11 EATC

Hsu *et al.* have presented a technique for minimizing energy consumption and the technique is termed as Energy-Aware Task Consolidation (EATC) [10]. According to this model, the energy utilization of any VM is determined based on CPU and the major concern is to keep the CPU utilization of VMs under the specified CPU utilization threshold. This proposed technique consolidates tasks amongst virtual clusters. They have considered the network latency during the task migration from one virtual cluster to another. If the allocated virtual cluster does not provide services for a task, then that virtual cluster needs resource support from other virtual clusters. If multiple virtual clusters can provide services, then the selection will be according to the minimum energy consumed cluster. The energy constraint, *i.e.* the maximum level of energy consumption is decided earlier. It affects the throughput level because of a limited number of tasks completed. In their proposed work, the data center for virtual cluster and VMs works on relatively constant bandwidth.

4.1.12 HSFL

Luo *et al.* have presented a Hybrid Shuffled Frog Leaping (HSFL) algorithm as a complete resource management scheme in the cloud [11]. They have designed a cloud system framework that implements live migration policy from current resource utilization and saved energy by switching some low utilized servers into sleeping mode while ensuring some degree of SLAs. They have calculated the updated upper threshold (*upper_th*) of processor utilization through the targeted SLAs of the host, and the current SLAs of the host. They have used *upper_th* $\in [0.5, 0.95]$. The consumption of energy of an idle host needs 70% of a fully utilized host [11]. The live migration of VMs process approximately consumes 10% of CPU utilization. Their energy model used the average rate of processor utilization of the host and the time window for migrating a set of VMs. They have used a static VM placement approach which leads to some more energy consumption of the system along with less throughput value.

4.1.13 DVMA

Wang *et al.* [12] have suggested a Decentralized VM Migration Approach (DVMA) for cloud infrastructure. They have considered that the utilization of CPU is related to the workload intensity, and the consumption of power of a physical host is predominantly affected by its current CPU utilization. The power consumption P_i of the ith physical host is defined as

$$P_i = \alpha \times P_i^{MAX} + (1 - \alpha) \times \theta \times P_i^{MAX} \tag{16}$$

where P_i^{MAX} is the power consumption of the ith host at peak level (100%) CPU utilization, α is the fraction of the power used by an idle host and θ be the current CPU utilization. To overcome the one point failure problem, they proposed a decentralized mechanism for the VM management within a data center.

4.2 Taxonomy Based on Migration

The VM or task migration can be used for load balancing, fault tolerance, energy saving, *etc.* In migration, VM or task transferred from its current server to a new one for further execution. The taxonomy of energy efficient service model for the cloud environment is shown in Figure 5. Energy saving through migration can be achieved by moving task or VM to a new server whose power consumption is less as compared to currently running server. Migration can be static or dynamic. In static migration, currently running server is stopped for some time to transfer all the running entity, data to the newly selected server. Once the transfer is over, the new server is ready to start execution. This process may introduce an extra delay to the running applications.

In dynamic migration, moving takes place in real-time. For example: let two servers are running various applications. After some time, the number of applications running can be accommodated in a single server. Then, migration is performed dynamically. But, this process will also introduce transfer overhead and extra time. Dynamic migration is preferred for a large system because optimal allocation is possible among available resource. Task migration took place when the issues

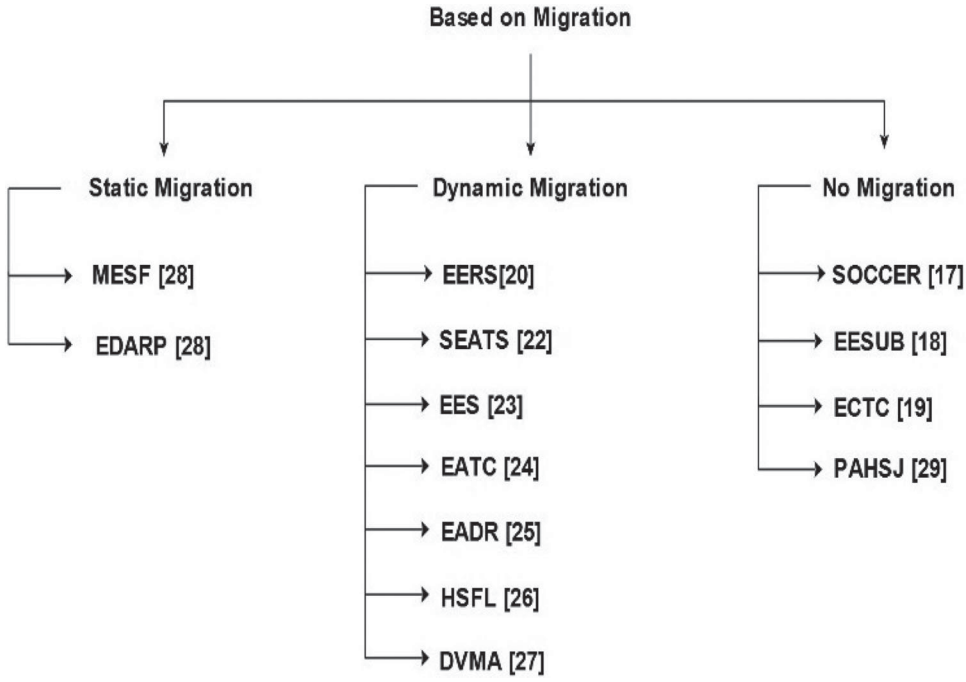

Figure 5: Taxonomy based on migration of task or VMs

pointed towards tasks. It means if there is a need for multiple resources (VMs) for the execution of a task, then the task is migrated from one VM to another in the specified sequence. The VMs involve for a task migration may deploy in the same host or different hosts.

VM migration tasks placed when the issues pointed towards VMs. It means if VM is failing during the running state, then the current state of the VM is migrated to another VM of the same host or another host. The migration of VM depends on the resource availability of the hosts of the system. In another case, if the system is an efficient one and few tasks are running on a VM. If that VM is the single VM of a host, then if there is a possibility of migration of that VM to another active host, then that host mode is converted to the sleep mode.

4.3 Discussion

Xindong *et al.* [37] have provided a survey on the energy efficient cloud system. According to them, there are a larger number of surveys conducted on energy consumption of the overall cloud system in recent time as compared to other perspectives. The cloud-based energy efficient problem has moved extensive interest in research during the last 10 years. Many of the energy-based systems have outlined for the different layers in the cloud system. Energy-based surveys and research works related to different layers of the cloud computing system are shown in Table 8.

The energy-aware approach summary that includes task type, whether migration is applied or not and various performance metrics (*e.g.* cost, deadline) that are used to measure the energy efficiency of the system is shown in Table 1. Observations made by the algorithms are: (i) Mostly, the energy efficiency measured based on CPU utilization. (ii) Few researchers have used the migration (VM/Task) technique [11,12]. (iii) Some of the researchers classified the input tasks as CPU, I/O, memory or network intensive [22], classes based on job type [9], and deadline-aware task [17,18,21]. (iv) Few of techniques to save power is to turn off idle hosts [18] or migrate VMs of under-utilized hosts to another host based on some threshold value [11,20,38]. In some cases, tasks are also migrated from one VM to another [10,39,40]. This migration can be done statically or dynamically. In static migration, information regarding the complete process and the system characteristics are known earlier. For example, if a task required a set of resources and that is available in multiple VMs, then the task, allocate to the first VM initially, then after completion of a portion of job in first VM, the task is migrated to second VM, and so on. In dynamic migration, the allocation is done initially, but when the system behavior changes suddenly which was not expected, then also some migration will be done to overcome that situation. Here, we explain the case of task migration and similar to the task migration, the VM migration also carried out. Researchers also consider the average response time required to perform certain tasks [16,17].

Table 8: Comparison of various energy-efficient techniques

Ref	Approach	Energy	CPU intensive	I/O intensive	Static Migration	Dynamic Migration	Deadline	QoS	Energy Cost	Avg. Res. Time
[14]	SOCCER	✓	✓	✓	×	×	✓	✓	✓	×
[15]	EESUB	✓	✓	×	×	×	✓	✓	×	×
[16]	ECTC	✓	✓	×	×	×	✓	×	×	✓
[16]	MaxUtil	✓	✓	×	×	×	✓	×	×	✓
[9]	EERS	✓	✓	✓	×	✓	×	✓	×	×
[17]	MESF	✓	✓	×	✓	×	✓	✓	✓	✓
[18]	SEATS	✓	✓	×	×	✓	✓	✓	✓	×
[19]	EES	✓	✓	×	×	✓	✓	×	✓	×
[10]	EATC	✓	✓	×	×	✓	×	×	✓	×
[20]	EADR	✓	✓	×	×	✓	✓	✓	×	×
[11]	HSFL	✓	✓	×	×	✓	×	✓	×	×
[12]	DVMA	✓	✓	×	×	✓	×	✓	×	×
[21]	EDARP	✓	✓	×	✓	×	✓	✓	✓	×
[22]	PAHSJ	✓	✓	✓	×	×	✓	✓	×	×

As discussed in the literature, the key research challenges associated with energy-efficient service allocation techniques are as follows: (1) the optimal trade-off between energy consumption and system performance. (2) Optimal VM placement in the cloud system, *i.e.* finding a suitable host for the deployment of a specific VM. (3) Similarly, VM selection plays an important role in the cloud system. This process includes selecting an appropriate VM for a task out of several VMs to maximize the system performance. (4) In some cases, a task required multiple VMs for the successful execution. So it required an efficient sequence of VM execution so that task execution can be completed within specified system constraints. (5) Sometimes a task or VM need to be migrated to achieve improvement in system performance such as minimization of energy consumption and ensuring SLA. During migration, various factors need to be considered such as finding an appropriate VM or a physical machine for hosting VM and avoid under or over utilization of VM/physical machine. Also, the migration time should not be high so that ET of a task should be within specified constraint. (6) How to develop efficient decentralized and scalable algorithms for the allocation of cloud resources? (7) In the current time, solutions for the multi-objective problem play a vital role. The solution must answer the query: How to generate a comprehensive solution by combining different objectives in several allocation techniques? Some objectives are contradictory to each other like minimize energy consumption while limiting SLA violation. Service allocation techniques with this type of objectives need special effort to find the tradeoff between conflicting entities.

5. CONCLUSION

In this study, we have manifested various energy efficient algorithms for service allocation in a cloud environment. These algorithms are categorized based on the input task model and migration on of task or VMs. An analytical cloud system model is discussed to compute the total energy consumed in a cloud environment considering data center model, host model, VM model, and service model. We further describe various energy model proposed by researchers to calculate energy consumed in the cloud system for service allocation problem to highlight the ongoing research trend. We hope this article can provide a roadmap for future research in the emerging domain of energy-efficient service allocation in the cloud environment.

Due to the extensive research interest in the recent time on energy efficient cloud system, there is a little range to reduce the energy consumption at every level of the systems. Therefore, how to connect the energy efficient policies at every level to form an energy efficient skeleton is a subject for future study. There are also numerous studies on the energy-aware data management policies in the cloud storage systems. So, this may also identify some methods of reducing energy consumption in cloud storage systems. However, further research can be taken up to design or develop the energy model that is appropriate to specific services provided by the cloud.

REFERENCES

1. N. Phaphoom, X. Wang, and P. Abrahamsson, "Foundations and technological landscape of cloud computing," *ISRN Software Engineering*, Vol. 2013, pp. 1–31, 2013.

2. H. E. Schaffer, "X as a service, cloud computing, and the need for good judgment," *IT Prof.*, Vol. 11, no. 5, pp. 4–5, 2009.

3. P. Gupta, A. Seetharaman, and J. R. Raj, "The usage and adoption of cloud computing by small and medium businesses," *Int. J. Inf. Manage.* 33, no. 5, pp. 861–74, 2013.

4. T. Ma, Y. Chu, L. Zhao, and O. Ankhbayar, "Resource allocation and scheduling in cloud computing: Policy and algorithm," *IETE Tech. Rev.*, Vol. 31, no. 1, pp. 4–16, 2014. DOI:10.1080/02564602.2014.890837.

5. M. Sampaio, J. G. Barbosa, and R. Prodan, "Piasa: A power and interference aware resource management strategy for heterogeneous workloads in cloud data centers," *Simul. Model. Pract. Theory.*, Vol. 57, pp. 142–60, 2015.

6. H. M. Lee, Y.-S. Jeong, and H. J. Jang, "Performance analysis based resource allocation for green cloud computing," *J. Supercomput.*, Vol. 69, no. 3, pp. 1013–26, 2014.

7. D. Ergu, G. Kou, Y. Peng, Y. Shi, and Y. Shi, "The analytic hierarchy process: Task scheduling and resource allocation in cloud computing environment," *J. Supercomput.*, Vol. 64, no. 3, pp. 835–48, 2013.

8. T. L. Casavant and J. G. Kuhl, "A taxonomy of scheduling in general-purpose distributed computing systems," *IEEE Trans. Software Eng.*, Vol. 14, no. 2, pp. 141–54, 1988.

9. A. Fayyaz, M. U. Khan, and S. U. Khan, "Energy efficient resource scheduling through VM consolidation in cloud computing," in IEEE 13th International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2015, pp. 65–70.

10. C.-H. Hsu, K. D. Slagter, S.-C. Chen, and Y.-C. Chung, "Optimizing energy consumption with task consolidation in clouds," *Inf. Sci.*, Vol. 258, pp. 452–62, 2014.

11. J. P. Luo, X. Li, and M. R. Chen, "Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers," *Expert. Syst. Appl.*, Vol. 41, no. 13, pp. 5804–16, 2014.

12. X. Wang, X. Liu, L. Fan, and X. Jia, "A decentralized virtual machine migration approach of data centers for cloud computing," *Math. Probl. Eng.*, Vol. 2013, pp. 1–11, 2013.

13. S. K. Mishra, D. Puthal, B. Sahoo, S. K. Jena, and M. S. Obaidat, "An adaptive task allocation technique for green cloud computing," *J. Supercomput.*, 74, no. 1, pp. 1–16, 2018.

14. S. Singh, I. Chana, M. Singh, and R. Buyya, "SOCCER: Self-optimization of energy-efficient cloud resources," *Cluster. Comput.*, Vol. 19, no. 4, pp. 1787–800, 2016.

15. R. N. Calheiros and R. Buyya, "Energy-efficient scheduling of urgent bag-of-tasks applications in clouds through DVFS," in Cloud Computing Technology and Science (CloudCom), Singapore, 2014, pp. 342–9.

16. Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *J. Supercomput.*, Vol. 60, no. 2, pp. 268–80, 2012.

17. Z. Dong, N. Liu, and R. Rojas-Cessa, "Greedy scheduling of tasks with time constraints for energy-efficient cloudcomputing data centers," *J. Cloud Comput.*, Vol. 4, no. 1, pp. 1–14, 2015.

18. S. Hosseinimotlagh, F. Khunjush, and R. Samadzadeh, "Seats: smart energy-aware task scheduling in real-time cloud computing," *J. Super-Comput.*, Vol. 71, no. 1, pp. 45–66, 2015.

19. S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Energy-efficient scheduling of HPC applications in cloud computing environments," arXiv preprint arXiv: 0909.1146.

20. A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, Vol. 28, no. 5, pp. 755–68, 2012.

21. Y. Gao, Y. Wang, S. K. Gupta, and M. Pedram, "An energy and deadline aware resource provisioning, scheduling and optimization framework for cloud systems," in Proceedings of the Ninth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, IEEE Press, Montreal, Quebec, Canada, 2013, p. 31.

22. F. A. Cano, A. Tchernykh, J.-M. Corts-Mendoza, R. Yahyapour, A. Drozdov, P. Bouvry, D. Kliazovich, and A. Avetisyan, "Heterogeneous job consolidation for power aware scheduling with quality of service", Workshop on Network Computing and Supercomputing, 1482, 2015, pp. 687–97.

23. S. Bera, S. Misra, and J. J. Rodrigues, "Cloud computing applications for smart grid: A survey," *IEEE Trans. Parallel Distrib. Syst.*, Vol. 26, no. 5, pp. 1477–94, 2015.

24. A. Abdalla and A. K. Pathan, "On protecting data storage in mobile cloud computing paradigm," *IETE Tech. Rev.*, Vol. 31, no. 1, pp. 82–91, 2014. DOI: 10.1080/02564602.2014.891382.

25. GREENPEACE. http://www.greenpeace.org/usa/wp-content/uploads/legacy/Global/usa/planet3/PDFs/2015ClickingClean.pdf, May 2015.

26. K. Chen, C. Hu, X. Zhang, K. Zheng, Y. Chen, and A. V. Vasilakos, "Survey on routing in data centers: Insights and future directions," *IEEE Netw.*, Vol. 25, no. 4, pp. 6–10, 2011.

27. T. Kaur and I. Chana, "Energy efficiency techniques in cloud computing: A survey and taxonomy," *ACM Comput. Surv. (CSUR)*, Vol. 48, no. 2, p. 22, 2015.

28. A. Hameed, *et al.*, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," *Computing*, Vol. 98, no. 7, pp. 751–74, 2016.

29. T. Shi, M. Yang, X. Li, Q. Lei, and Y. Jiang, "An energy-efficient scheduling scheme for time-constrained tasks in local mobile clouds," *Pervasive. Mob. Comput.*, Vol. 27, pp. 90–105, 2016.

30. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

31. S. K. Mishra, D. Puthal, J. J. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable service allocation using metaheuristic technique in fog server for industrial applications," *IEEE Trans. Ind. Inf.*, Vol. 14, pp. 4497–506, 2018.

32. S. Ali, H. J. Siegel, M. Maheswaran, and D. Hensgen, "Task execution time modeling for heterogeneous computing systems," in 9th Proceedings on Heterogeneous Computing Workshop (HCW 2000), IEEE, Cancun, Mexico, 2000, pp. 185–99.

33. E. Grochowski and M. Annavaram, "Energy per instruction trends in intel microprocessors," *Technology@ Intel Magazine*, Vol. 4, no. 3, pp. 1–8, 2006.

34. S. Usman, K. Bilal, N. Ghani, S. U. Khan, and L. T. Yang, "Thermal-aware, power efficient, and makespan realized pareto front for cloud scheduler," in IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops), USA, 2015, pp. 769–75.

35. M. Giacobbe, A. Celesti, M. Fazio, M. Villari, and A. Puliafito, "Towards energy management in cloud federation: A survey in the perspective of future sustainable and cost-saving strategies," *Comput. Netw.* Vol. 91, pp. 438–52, 2015.

36. M. Sajid and Z. Raza, "Turnaround time minimization-based static scheduling model using task duplication for fine-grained parallel applications onto hybrid cloud environment," *IETE J. Res.*, 2015. DOI:10.1080/03772063.2015.1075911.

37. X. You, Y. Li, M. Zheng, C. Zhu, and L. Yu, "A survey and taxonomy of energy efficiency relevant surveys in cloud-related environments," *IEEE Access*, Vol. 5, pp. 14066–78, 2017.

38. S. Oh. Murtazaev, "Sercon: server consolidation algorithm using live migration of virtual machines for green computing," *IETE Tech. Rev.*, Vol. 28, no. 3, pp. 212–31, 2011.

39. W. D. Mulia, N. Sehgal, S. Sohoni, J. M. Acken, C. L. Stanberry, and D. J. Fritz, "Cloud workload characterization," *IETE Tech. Rev.*, Vol. 30, no. 5, pp. 382–97, 2013.

40. A. Horri, M. S. Mozafari, and G. Dastghaibyfard, "Novel resource allocation algorithms to performance and energy efficiency in cloud computing," *J. Supercomput.*, Vol. 69, no. 3, pp. 1445–61, 2014.

Authors

Sambit Kumar Mishra obtained his PhD degree in computer science & engg from National Institute of Technology, Rourkela, India. His areas of research include cloud computing, parallel and distributed computing system, wireless sensor networks. He obtained his MTech and MSc in computer science from Utkal University, India. He is a member of IEEE Computer Society.

Corresponding author. Email: skmishra.nitrkl@gmail.com

Sampa Sahoo is pursuing a PhD in Department of Computer Science & Engg at National Institute of Technology, Rourkela, India. Her areas of research include cloud computing, parallel and distributed computing system. She obtained her MTech and BTech in computer science & engg. from Berhampur University, India and Biju Patnaik University of Technology (BPUT), India respectively. She is a member of IEEE Computer Society.

Email: sampaa2004@gmail.com

Bibhudatta Sahoo obtained his MTech and PhD degree in computer science & engineering from NIT, Rourkela. He has 24 years of teaching experience in undergraduate and graduate level in the field of computer science & engineering. He is presently Assistant Professor in the Department of Computer Science & Engineering, NIT Rourkela, INDIA. His technical interests include data structures & algorithm design, parallel distributed systems, networks, computational machines, algorithms for VLSI design, performance evaluation methods and modeling techniques distributed computing system, networking algorithms, and web engineering. He is a member of IEEE & ACM.

Email: bibhudatta.sahoo@gmail.com

Sanjay Kumar Jena received his MTech in computer science and engineering from the Indian Institute of Technology Kharagpur and PhD from the Indian Institute of Technology Bombay, in 1982 and 1990, respectively. He is a fulltime professor in the Department of Computer Science and Engineering, National institute of Technology Rourkela. He is a senior member of IEEE and ACM and a life member of IE(I), ISTE, and CSI. His research interests include data engineering, information security, parallel computing and privacy preserving techniques.

Email: skjena@nitrkl.ac.in