J Grid Computing
DOI 10.1007/s10723-016-9364-0

Energy-aware Virtual Machine Migration for Cloud Computing - A Firefly Optimization Approach

Nidhi Jain Kansal · Inderveer Chana

Received: 18 March 2015 / Accepted: 10 February 2016

Abstract Energy efficiency has grown into a latest exploration area of virtualized cloud computing paradigm. The increase in the number and the size of the cloud data centers has propagated the need for energy efficiency. An extensively practiced technology in cloud computing is live virtual machine migration and is thus focused in this work to save energy. This paper proposes an energy-aware virtual machine migration technique for cloud computing, which is based on the Firefly algorithm. The proposed technique migrates the maximally loaded virtual machine to the least loaded active node while maintaining the performance and energy efficiency of the data centers. The efficacy of the proposed technique is exhibited by comparing it with other techniques using the CloudSim simulator. An enhancement in the average energy consumption of about 44.39 % has been attained by reducing an average of 72.34 % of migrations and saving 34.36 % of hosts, thereby, making the data center more energy-aware.

Keywords Cloud computing · Energy awareness · Firefly optimization · Virtualization · Virtual Machine (VM) migration

N. J. Kansal (✉) · I. Chana
Computer Science, Engineering Department,
Thapar University, Patiala, Punjab, India
e-mail: nidhi.kavya@thapar.edu

I. Chana
e-mail: inderveer@thapar.edu

1 Introduction

Cloud computing [1] characterizes a vital step in computing by offering shared computational power of the resources on demand [10]. Being grounded on the fundamental concept of virtualization [2], it has significantly transformed the manner of delivering the IT services with minimized infrastructural requirements. The virtual environment involves the creation of multiple VMs (or virtual servers) on a single physical node. In actual context, the multiple operating systems (OSs) can run on a single OS underlying the same hardware platform. The running of virtual servers minimizes the resource idle time, thus preventing the resource under-utilization [16, 17]. Additionally, the reduction in the amount of required hardware lowers the power needed for operation which consequently cuts down the energy demand. The diminution in the energy demand by the ICT (Information and Communication Technology) sector is highly appreciated in the current scenario of rising energy crisis. Energy efficiency [3] has thus gained prominence in the ICT data centers that host massive servers resulting in the induced upsurge of energy consumption levels [13].

The emergence of cloud computing and the virtualization support offered by it, has further corroborated the efforts for realizing energy efficient computing. It has been observed that the virtualized cloud data centers require lesser energy as compared to the non-virtualized ICT data centers. The extended facility of migrating the running VMs without any

Published online: 19 February 2016

perceptible downtime, from the heavily-loaded nodes to the lowly-loaded nodes, helps to manage the workload to minimize the energy consumption. The decrease in the consumed energy is due to the improved node utilization that results from a well-adjusted distribution and execution of workload on the nodes. The composed distribution of the workload among the nodes prevents node over-utilization that would have otherwise occurred. The optimally utilized nodes consume less energy as compared to the nodes that are over-utilized or under-utilized [18]. The under-utilization of a node indicates that the node is sitting idle while the over-utilization of a node means it is running tasks beyond its capability. The concept of dynamically and transparently migrating the VMs from one host to the other, to find the best target host is known as Live migration [11]. Apart from this, the key benefit of VM migration is the identification of hot-spots in the data centers [12]. The over-utilized nodes are the hot-spots and their identification helps to lower the energy consumption by migrating their load to the less utilized nodes, leading to green cloud data centers.

Obtaining the energy optimization through VM migration by regulating the workload on individual nodes is an NP (Nondeterministic Polynomial) - hard problem [14] and the heuristic methods are often used to resolve such kind of problems. To find an optimal solution, local heuristics may not be adequate, therefore, meta-heuristic approaches are suitable to efficiently crack these types of problems. Meta-heuristic is a repetitive primary procedure to guide and amend the jobs of secondary heuristics to yield high-grade results [15]. The nature-inspired multi-agent Firefly optimization (FFO) [4] meta-heuristic algorithm is chosen for this work as it is an efficient and powerful tool to find a nearly optimum solution by first performing local search and then global search on the problem's search space. The local search is called as Diversification while the task of global searching is called as Intensification [4, 5, 7].

The motivation for this work is to put forward an energy-aware VM migration technique applicable in the cloud environment which will help to lower the consumed energy in the cloud data centers. This work proposes a FireFly Optimization based Energy-aware Virtual Machine Migration (FFO-EVMM) technique to find the best VM-Host pair. It intends to maximize the energy-efficiency through the optimum migration of VMs, thereby improving the resource utilization levels. Several Bio-inspired techniques that exploit the behavioural and social instincts of the biological creatures exist today. The reasons for choosing FFO technique for our work over other social behavior inspired techniques are: (1) It offers systematic partitioning and capability to handle multiple modes, (2) its computation time is less in possibility of finding the global optimized answer, (3) it has high speed of convergence which is due to the quality parameters that can be regulated, (4) a balanced and optimal solution is obtained by properly exploiting and exploring the problems search space, (5) it involves lesser number of function evaluations, (6) its status can be changed from one optimization point to the other one, (7) random variables are used and the answers have the probable nature [5, 7–9].

The contribution of our work is as follows:

- An energy-aware meta-heuristic technique that performs live migration of the VMs from one active node to the other active node.
- Our approach makes use of a bio-inspired Firefly optimization technique to achieve energy efficiency in cloud data centers.
- The energy-efficiency has been maximized through the optimum migration of VMs, thereby improving the resource utilization levels.
- This approach also sustains scalability to the large number of heterogeneous cloud nodes.
- The achievability of the proposed technique has been shown by executing it in the CloudSim simulator [59].
- The efficacy of the proposed technique is exhibited by comparing it with other techniques. An enhancement in the average energy consumption of about 44.39 % has been attained by reducing an average of 72.34 % of migrations and saving 34.36 % of hosts.

The rest of the paper is structured as follows: Section 2 briefly discusses the related literature. The Firefly Optimization (FFO) Algorithm is discussed in Section 3. In Section 4, the proposed FFO-based VMM technique is described. Section 5 presents the existing reference algorithms and Section 6 demonstrates the experimental setup used for the simulations and result analysis. Section 7 lays out the derived conclusion and future work.

2 Related Work

A lot of research is being conducted in the area of cloud computing to reduce the power consumption in the data centers as surveyed in [56, 57]. Many different techniques to overcome the power wastage have been proposed and devised with and without VM migration. Live VM migration is being vigorously investigated since long and numerous techniques have been developed to migrate a running VM from one active node to the other active node. It has been observed to be an influential technique for efficiently managing the data center energy. Most of the prevailing VM migration methodologies for energy management in cloud data centers are not straight forward, as primarily they involve VM consolidation or VM placement approaches at the higher level of implementation. Thissection briefly discusses such techniques.

Feller et al. [30] have put forward a scalable and autonomic VMs management framework that uses a centralized ACO-based VM consolidation algorithm to locally consolidate the VMs. Tarighi et al. [24] have offered a fuzzy decision making based VM migration scheduling algorithm, that discovers the maximally loaded servers and takes a migration decision by using TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) approach. Wood et al. [25, 26] have suggested two gray-box and black-box approaches for virtualized cluster to diminish the hotspots by monitoring and detecting hotspots and then allowing the live migration of VMs. Marzolla et al. [32] have projected a VM consolidation protocol based on a coarse-grained gossip that apply local VM consolidation by migrating the VMs from the smallest laden node to the greatest laden node. All the above mentioned techniques have considered VM migration in one or the other way, but there is no reflection of energy savings. Thereby, these techniques differ from our proposed approach, which principally aims to achieve energy savings with the help of VM migration in the cloud data centers.

Nathuji et al. [45] have designed an architecture for the management of the energy in the virtualized data centers by using VM live migration to consolidate multiple VMs on a single server. Tolia et al. [29] have practiced a short-term VM migration for consolidating workloads and to put the under-utilized servers in the sleep mode. Lim et al. [27] have presented a way of consolidating VMs onto a lesser number of hosts by dynamically migrating virtual machines to save energy in a virtualized environment. A multi-objective profit-oriented algorithm to place VMs has been proposed by Goiri et al. [38]. Performance in terms of SLA violations, energy efficiency and overheads of virtualization have been considered in this algorithm. Ghribi et al. [13] have offered a combination of an exact VM allocation algorithm and an exact VM migration algorithm for reducing the number of nodes and hence to save energy in cloud data centers. Verma et al. [28, 34] have proposed a framework that examines the VM placement algorithms by considering the energy and the migration costs as well as the performance benefit in a virtualized sever cluster, to maximize performance and to minimize energy consumption. Mehta and Neogi [40] have preseneted a ReCon tool to dynamically consolidate servers in data centers. The VMs are mapped to the servers by considering the static and the dynamic costs of physical servers, the cost of VM migration, and the resource consumption data from the history.

The work cited in [13, 27–29, 34, 38, 40, 45], does not consider energy consumption done by memory unlike our FFO-EVMM technique which in addition to the CPU energy consumption offers energy optimization at the memory level as well. In other words, it tries to handle the impact of time-space parameters in terms of the consumed energy.

An approach for VM consolidation has been offered by Cardosa et al. [37] that agrees to the highest and lowest resource requirements of the VMs to achieve energy-efficiency. Resource utilization is improved and energy consumption is reduced by consolidating several VMs onto a single server. Effectual energy-aware heuristics to allocate VMs dynamically have been advocated by Beloglazov et al. [17, 35, 36]. These heuristics practice live migration of VMs to minimize energy consumption by reducing the number of used nodes and without having required the knowledge of VMs applications [44, 61]. A solution for VM placement and consolidation that is grounded on Bernoulli trials has been proposed by Mastroianni et al. [31] by considering energy and migration cost. Dong et al. [41] have recommended a scheme to place VMs that meets several resource restrictions to reduce energy consumption by enhancing resource utilization and by saving number of used servers and network elements. All these papers [17, 31, 35–37, 41, 44, 61] have emphasized on the VM placement techniques and their focus is mainly on reducing the used servers in order to lower the energy levels whereas our work employs a VM migration technique to reduce the energy consumption by saving hosts and minimizing VM migrations.

Vu et al. [43] have projected a VM placement algorithm that enhances the performance of communication by decreasing the overall cost of the virtual machine traffic and saves energy by increasing the utilization of the CPUs. Sekhar et al. [46] have designed an energy efficient VM live migration technique based on greedy heuristics to curtail the consumed energy in cloud data centers. Jung et al. [50] have established a framework to optimize the energy consumption by using the live VM migration for consolidating virtual servers and by switching-off the idle servers in cloud data centers. Bila et al. [49] have offered a technique that partially migrates the VMs that are idle and are running on the desktops of the users to a consolidation server to reduce overall consumed energy. Graubner et al. [39] have extended the Eucalyptus cloud management framework to incorporate the support for live migration and consolidation. Xiaoli et al. [42] have presented an energy-aware VM placement algorithm for making cloud data centers more energy-efficient by increasing resource utilization. The resource utilization as well as the energy cost in migrations have been considered in their approach. Unlike our technique, none of the techniques listed in this paragraph, deals with CPU and memory utilization for lowering the energy consumption. Also, they attempt to diminish the consumed energy without considering the hosts and the VM migrations whereas FFO-EVMM cuts down the used energy accomplished by saving the number of nodes and by lowering the number of migrations.

The work in [63] focuses on enhancing cloud service reliability by using storage and network resources optimally, where as our work focuses on enhancing the energy efficiency of the cloud data centers by improving the usage of CPU and memory resources. Although in both the papers, CloudSim has been used for the implementation, but, their algorithm exploits data centers network architecture characteristics and node failure predictor to minimize the network resource usage, whereas, our algorithm migrates the maximally loaded virtual machine to the least loaded active node while maintaining the performance and energy efficiency of the data centers.

The technique mentioned in [33] mainly targets the reduction in the number of active nodes and the number of VM migrations to cut down the consumed energy in the overall data center. The FFO-EVMM also minimizes the number of nodes and VM migrations but it individually computes the energy consumption of VM and node thereby keeping a track of the energy consumption of each and every VM and node in the cloud data center. The purpose behind computing the individual node and VM energy is to analyze the workload handling capacity of the node. The workload handling capacity of the node can be considered as the capability of a node to process the types of workloads while keeping the energy consumption under a set threshold. Our pre-existing work for the FFO-EVMM technique is the resource utilization technique described in [6] deals with two different types of workloads- CPU-intensive and memory-intensive. It is important to carefully consolidate variable workloads in order to avoid contention of resources. The contention among resources can cause performance degradation and hence energy wastage. The FFO-EVMM also provides workload scalability such that a large number of workloads can be processed without violating the energy constraints.

The authors in [35] advocate different energy-aware heuristics for dynamically allocating VMs in accordance with the current resource utilization. The live migration of VMs is practiced to set aside the free resources that are then switched to the sleep mode, hence cutting down the energy consumption done by them when in idle mode. The main focus for preserving the free resources is to lower the SLA violations and to improve the energy-efficiency of the data center. These heuristics run on varied underlying infrastructure and assorted VMs while maintaining the SLA constraints imposed by the users. They have primarily attempted to optimize the energy consumption done by the processor while missing out the energy consumption done by the memory. The memory is one of the most vital elements of emphasis in the power and energy usage optimization in the current scenario [58]. In order to achieve an optimal VM migration and placement, it is important to consider the current utilization of processor and memory which have been observed to be the major power consuming units in a system [35, 58]. Our proposed FFO-EVMM technique effectively offers the energy optimization at both the processor and memory level. Apart from this, it attempts to minimize the number of VM migrations and the number of hosts, thus avoiding further energy wastage. Furthermore, it is based on an energy-aware resource utilization technique that helps to improve the utility levels of the resources while preventing the performance degradation. It also takes into account the energy consumed by NAS unlike the above two techniques.

Based on the investigation of the existing works, it can be inferred that most of the techniques have focused on energy management largely through VM consolidation and VM placement. Likewise, the proposed technique focuses majorly on the improvement in the performance and energy consumption levels through VM migration. Additionally, our technique is based on the bio-inspired FFO technique. With the rising diversity and complexity of large-scale distributed computing services, there is a necessity to design more scalable, heterogeneous and sustainable computing techniques that can conjointly deal with the other issues such as heterogeneity and growing energy crisis as well. Thus, apart from the underlying infrastructure support (available through cloud computing in this case), it is important to explore and adopt new paradigms.

Currently, many researchers are focusing and implementing the biologically inspired computing as a preferable paradigm to handle these issues with proficiency and without the augmented complication. In spite of the several inherent challenges encountered while surviving in an enormous, dynamic, incredibly diverse, and highly complex environment, the biological organisms evolve, self-organize, self-repair, navigate, and flourish. This is possible with their local knowledge and without any centralized control [53, 54]. This prompted the research community to discover and learn lessons from the biological systems such as Ant Colony Optimization (ACO) [20, 22], Artificial Bee Colony (ABC) [21], Bacterial Foraging Optimization (BFO) [62], Particle swarm optimization (PSO) [4, 22] techniques etc.

For our work, we have chosen biological behaviour of firefly insects and have devised FFO-based migration technique. The criteria for choosing it is its faster convergence speed and global optimization attainment. Furthermore, it exploits the concept of curtailing the overall upsurge in the incremental power due to the new VM migrations and has never been used previously for VM migration approach. Like most of the VM migration techniques, the designed FFO based technique saves the storage space by using the capability of Network Attached Storage (NAS) [47, 48] which is not the case in all the above mentioned techniques except the work done in [33]. The use of NAS helps to store the VM images and data thereby saving space and offering faster data access capabilities. In the scenario of energy consumption by VM migration, a linear model based on FFO is formulated that runs an FFO algorithm which is able to solve the energy consumption issue with the attraction property of fireflies. The capability of fireflies to get attracted towards the brighter fireflies is the basis for considering the FFO.

3 Firefly Optimization (FFO) Algorithm

The Firefly Optimization (FFO) algorithm has been designed by Xin-She Yang in the late 2007 and 2008 at Cambridge University [4, 5, 7]. It is centred on the flashing features of fireflies and uses the subsequent three idealized procedures: (1) One firefly is attracted to the other fireflies irrespective of their sex as all fireflies are unisex, (2) The attractiveness is proportionate to the brightness, thus they both decrease as their distance increases and for any two flashing fireflies, the less brighter one will travel near the brighter one. If no firefly is brighter than a specific firefly, it moves arbitrarily and (3) The brightness of a firefly is regulated by the landscape of the objective function to be optimized.

Thus, the variation of the attractiveness β with the distance r can be defined as [4, 5, 7]:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{1}$$

where β_0 is the attractiveness at $r = 0$. The movement of a firefly i is attracted to another more attractive (brighter) firefly j is determined by [4, 5, 7]:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} \left(x_j^t - x_i^t \right) + \alpha_t \epsilon_i^t \tag{2}$$

where α_t is the randomization parameter that controls the randomness and ϵ_i^t is a vector of random numbers drawn from a Gaussian distribution or uniform distribution at time t. The second term is due to the attraction and the third term is due to randomization. If $\beta_0 = 0$, it becomes a simple random walk. On the other hand, if $\gamma = 0$, it reduces to a variant of particle swarm optimization (PSO) [4, 22].

4 Proposed FireFly Optimization—Energy-Aware Virtual Machine Migration (FFO-EVMM) Technique

The prior work related to the scheduling aspect of the proposed technique has already been done and is available in our previously published work [6]. The previous work proposed an energy-aware resource utilization model which is shown in Fig. 1.

The model facilitates the energy-aware scheduling decisions by properly and efficiently managing the cloud resources. It further uses an Artificial Bee Colony (ABC) based energy-aware resource utilization technique to provide the required resources to the users' applications in a way to improve the resource utilization levels and to diminish the energy consumption in the cloud data centers without degrading the performance. The energy saving is also done by keeping the idle nodes in a sleep mode. Also, the energy-aware decisions are based on the past resource utilization and energy consumption data. Therefore, it can be said that the model enhances the utility levels of the server resources, reduces the energy consumption and hence the heat dissipation in the cloud data centers, thus contributing directly to the green computing.

Its problem formulation, energy model, mathematical explication and the detailed working are given in our previously published work [6].

The present work proposes an energy-aware virtual machine migration technique for cloud computing that is based on the flashing behavior of fireflies. This technique tries to migrate the most loaded VM from an active node which satisfies a minimum criteria for energy consumption, to another active node that consumes the least energy. It consists of four main parts, A) Selection of source node, B) Selection of VMs, C) Selection of destination node and D) Distance updated values. The description of all the above parts is as follows:

A. Selection of source node: The source node is the active node from where the VMs have to be migrated. The active node which is at the least distance(defined in step 4.) from the destination node, is selected as the source node. For this, the following steps are done.

Step 1. The CE of each active node is calculated using the (3) and then, all the values are stored in a list.

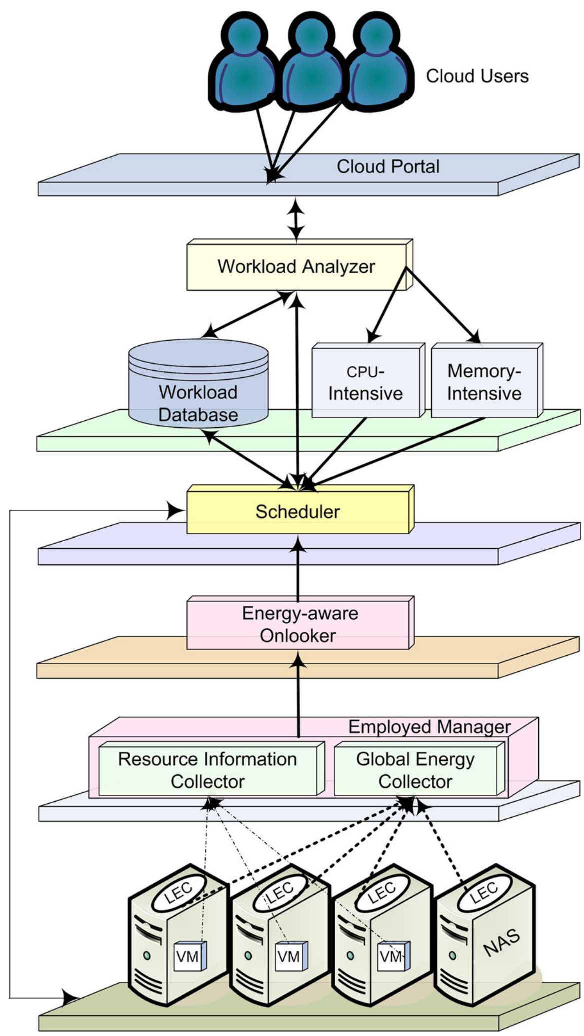


Fig. 1 Energy-aware resource utilization model[6]

1. Compute CE for each active node using the following equation

$$CE_i = \frac{\left(\sum_{j=1}^{v}\sum_{k=1}^{u} cpu_{ijk}\right)\left(\sum_{j=1}^{v}\sum_{k=1}^{u} mu_{ijk}\right)}{M} \times t \tag{3}$$

where v is the number of VMs running on the ith node and u is the number of jobs assigned to v VMs. cpu_{ijk} and mu_{ijk} are the processor and the memory utilizations of k jobs running in j VMs on the ith node respectively and M is the number of memory units [6].
2. Store these values in a list, *CE*.

Step 2. Time-based optimization: After computing the CE of each active node, the next step is to optimize the proposed technique for performance in terms of minimizing the execution time (Node Computation Time). The Node Computation Time (NCT) is calculated for each active node using the (4) and the values are again stored in a list.

1. Compute Node Computation Time (NCT) for each active node using the following equation

$$NCT_i = \sum_{j=1}^{v}\sum_{k=1}^{u} NCT_{ijk} \tag{4}$$

where NCT_{ijk} is the execution time of k jobs running in j VMs on the ith node [6].
2. Store these values in a list, *NCT*.

Step 3. Attraction Index (AI): The attraction property of the fireflies has been modelled by computing an AI value. The AI value is calculated using Indexed based searching and a sorted AI list is prepared according to CE values. The active node with the least CE is obtained as the first element of the list.

1. Compute $AI_i(CE_i, NCT_i)$ for each active node,
   where AI_i is the Attraction Index, CE_i is the energy consumption and NCT_i is the Node Computation Time for the ith node respectively.
2. Store these values in a list, *AI* and sort this list in an ascending order according to CE values.

Step 4. The node to be selected as a potential source for VM migration, must satisfy a minimum criteria for energy consumption and this is controlled by a distance value which is computed by using the (5). When finding a solution, this distance has to be the least in order to keep the energy consumption to the minimum. Thus, the node is selected, which has the CE value nearest to the computed distance value, to be the source node from where the VMs will be migrated.

1. Compute Distance, using the following equation:

$$Distance = Avg(AI_{mid}, AI_{max}) \tag{5}$$

where AI_{mid} and AI_{max} are the middle and the maximum values from the AI list.
2. Select the node with CE value closest to the above computed $Distance$ value.

B. Selection of VMs: The VMs to be migrated are determined.

Step 5. To select the VMs to be migrated from the source node, calculate the load of each VM on the source node according to the (6). Then, these values are stored in a list and the list is arranged from higher load value to the lower load value. The VM with the highest load value is selected to be moved to the destination node.

1. Compute Load for each VM of the above selected node at time instance Δt, using the following equation:

$$Load_{ij} = \frac{\sum_{j=1}^{v} job_j}{\left(\frac{\sum_{j=1}^{v} cpu_{ij} \sum_{j=1}^{v} mu_{ij}}{M}\right) \times \Delta t} \tag{6}$$

where job_j represents the total number of jobs running in the j^{th} VM on i^{th} node and $\frac{\left(\sum_{j=1}^{v} cpu_{ij}\right)\left(\sum_{j=1}^{v} mu_{ij}\right)}{M}$ is the total consumed power of the i^{th} node in Δt units of time.

2. Store these values in a list, *Load* and sort this list in the descending order.
3. Move the first element of the obtained list, *Load* to the first element of the *AI* list, i.e. move the most loaded VM towards the most brightest node.

C. Selection of destination node: The destination node for VMs is discovered.

Step 6. Discovering the most brightest node: The property of fireflies to move towards the brighter nodes requires to identify the brighter node. The node is said to be the brightest if its energy consumption (CE) is minimum. The active node with the least CE is obtained as the first element of the AI list and is thus the primary contender for migrating the overloaded/culprit VMs to it.

D. Distance Updated Values: The distance values are updated.

Step 7. The updation involved after each iteration is the updated value for the distance, which is given by the (7) as follows:

$$(Distance)^{t+1} = (Distance)^{t} + \frac{\sum_{j=1}^{v} job_j}{\left(\frac{\sum_{j=1}^{v} cpu_{ij} \sum_{j=1}^{v} mu_{ij}}{M}\right) \times \Delta t} + \epsilon \quad (7)$$

where $(Distance)^t$ & $(Distance)^{t+1}$ are the distance values at time t & $t+1$, second term is due to the *Load* and ϵ is the gaussian distribution error. The corresponding pseudocode is given below and the pseudoflow & interaction chart for the FFO-EVMM technique has been presented in Fig. 2.

Algorithm 1 FFO-EVMM Technique Algorithm

```
Data: Scheduling Information consisting of set of jobs, set of nodes, resource utilization & energy information of nodes [6]
Result: Finding the best VM-node pair
begin
    Source_Node()
    for (each Node) do
        Compute Energy Consumption using equation (3)
        CE[ ] ← Energy Consumption value
        Calculate Execution Time using equation (4)
        NCT[ ] ← Execution Time value
    for (each Node) do
        Compute Attraction Index
        AI[ ] ← Attraction Index value
    Sort AI[ ] in an ascending order according to Energy Consumption values
    Compute Distance using equation (5)
    Find the node with Energy Consumption value nearest to the calculated Distance value from the sorted AI[ ]

    Culprit_VM()
    for (each VM on the Source Node) do
        Compute Load using equation (6)
        Load[ ] ← Load value
    Sort Load[ ] in a descending order

    Destination_Node()
    Get the first element of sorted AI[ ] as the destination node

    Move the first element of Load[ ] to the first element of AI[ ]
    Update the Distance value using equation (7)
```

An example has been provided to have a better understanding of the algorithm. Suppose that there are 5 active nodes ($N_1 .... N_5$) and 12 VMs ($VM_1 .... VM_{12}$) are running on them. Let the VMs VM_1 and VM_3 are running on the node N_1 which is represented as $N_1 = (VM_1, VM_3)$, and same is for all the other four nodes, i.e. $N_2 = (VM_2, VM_4, VM_6)$, $N_3 = (VM_5, VM_7)$, $N_4 = (VM_8, VM_{10}, VM_{11})$ and $N_5 = (VM_9, VM_{12})$. The CE values of these active nodes are calculated using the (3) as 1308.89

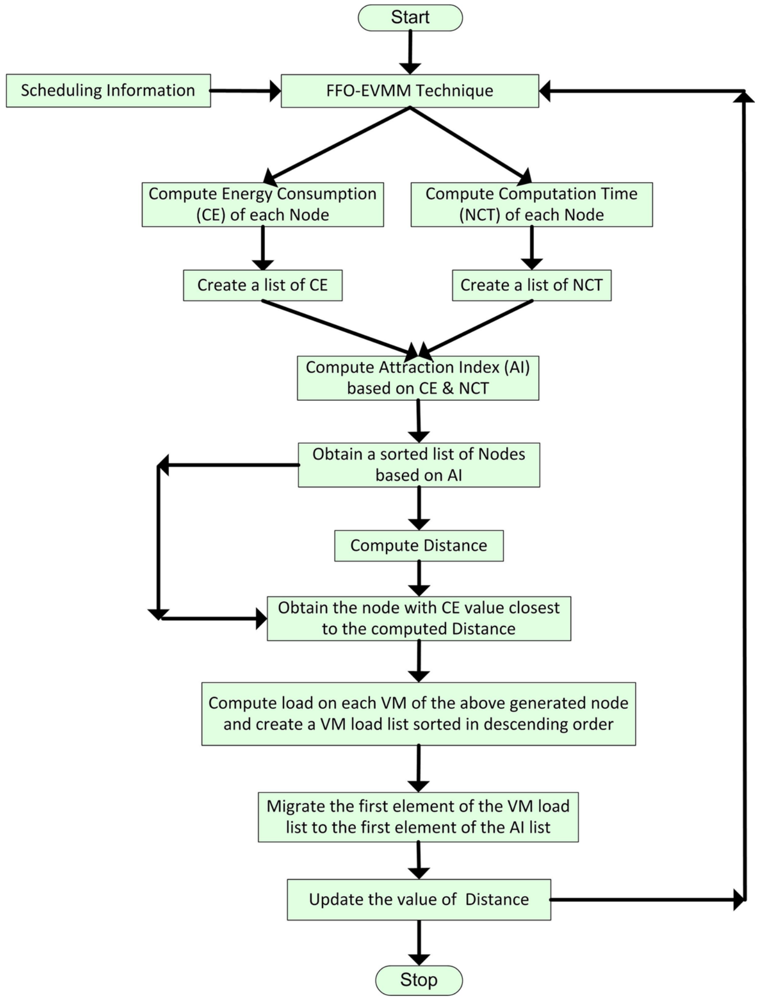


Fig. 2 Pseudoflow & interaction chart for FFO-EVMM technique

Wsec, 2507.24 Wsec, 1764.06 Wsec, 2872.72 Wsec and 1922.52 Wsec respectively, and the NCT values of these active nodes are calculated using the (4) as 0.36 sec, 0.38 sec, 0.42 sec, 0.49 sec and 0.53 sec respectively. These values are used to find the AI values as $AI_1 = (1308.89, 0.36)$, $AI_2 = (2507.24, 0.38)$, $AI_3 = (1764.06, 0.42)$, $AI_4 = (2872.72, 0.49)$ and $AI_5 = (1922.52, 0.53)$. The AI values are stored in a list which is sorted in an ascending order of CE, to obtain another list as ($AI_1 = (1308.89, 0.36)$, $AI_3 = (1764.06, 0.42)$, $AI_5 = (1922.52, 0.53)$, $AI_2 = (2507.24, 0.38)$ and $AI_4 = (2872.72, 0.49)$). Now, the distance value is computed as the average of $AI_5 = (1922.52, 0.53)$ and $AI_4 = (2872.72, 0.49)$ using the (5), which is equal to $(1922.52 + 2872.72)/2 = 2397.62$ Wsec. Next step is to select that node from the sorted AI list whose CE value is nearest to 2397.62 Wsec, which is the node having value 2507.24, that is the node N_2 which becomes the source node from where the VMs will be migrated. Now the load of each VM on the node N_2 is calculated using the (6) as $VM_2 = 835.75$ Wsec, $VM_4 = 1253.62$ Wsec and $VM_6 = 417.87$ Wsec. These values are stored in a list and that list is sorted in descending order and a list is obtained as (VM_4, VM_2, VM_6). Therefore, the VM_4 becomes the VM to be migrated. Next, the destination node is chosen where this VM will be moved. As the destination node is the node with the least CE value, it is obtained from AI list where The nodes are arranged in ascending order of CE. So, the first element of the sorted AI list, that is the node N_1 is the destination node where VM_4 will be migrated and the distance value will be updated using the (7) for the next iteration.

Whenever a running VM needs to be migrated, the entire state of the VM (embracing the virtual CPUs, the drivers' configuration, the memory of VM and the storage) is relocated [47]. For the efficient VM migration and the centralized availability, the images and data of all the VMs have been stored on a common storage called the NAS. NAS is reachable to all the nodes and serves as a storage for the VMs where non-redundant data is stored [48]. Whenever a VM is migrated, only in-memory states and CPU registers of that VM need to be migrated from one node to the other as its image and storage contents are accessed from NAS [47, 48]. This saves the storage space, improves the search rate and the look up time, gears up the flow rate of the data during a VM migration. This further helps in diminishing the data transfers and hence the time taken to migrate a VM, thereby dropping the consumed energy and the associated energy expenditures incurred while migrating VMs from one node to another. The faster execution of the tasks results in the minimization of the overall system execution time and hence improves the performance. Wholely, FFO-EVMM with the use of NAS, tends to deal with the overheads involved in VM migrations.

5 Existing Reference Algorithms

Ant Colony Optimization (ACO) [20, 22] and First Fit Decreasing (FFD) [52] algorithms have been used as reference algorithms.

5.1 FFD

One of the eminent greedy algorithms for classic bin packing problems is FFD algorithm. In FFD algorithm, the objects are sorted by their decreasing order followed by packing each object in the most suitable bin that can accommodate it. Although, this algorithm becomes quite effective by sorting the list of objects decreasingly, yet it does not give an assurance of an optimal solution. The running time of this algorithm may escalate for the extensive lists. However, it is identified that for at least one existing order of the objects, FFD yields an optimal solution [51, 52].

5.2 ACO

Ant Colony Optimization (ACO) [20, 22] algorithm has been used as reference algorithm in this work. ACO is a meta-heuristic to find near optimal solutions by means of a probabilistic technique, which can be used for problems belonging to the NP class. M. Dorigo discovered ACO algorithm [20] by perceiving the usual food-discovering manners of actual ants that converse indirectly via their surroundings by depositing a chemical element called pheromone. This mode of conversation is called stigmergy. A probabilistic verdict is practiced by the ants for their travel to search the food where the routes with higher quantity of pheromone are likely to be chosen. On discovering the food, the ants dreg the pheromone on their return to persuade the other ants to trail the food source. A natural pheromone disappearance process is used to lessen the volume of pheromone over time to retain the regularly used paths that lead to the better solutions. Synthetic ants act as a multi-agent system and create a complex solution when applied on combinatorial problems like Bin Packing Problem (BPP) [19, 20, 22, 23].

6 Experiments & Results

This section evaluates the proposed FFO-EVMM algorithm and compares it with the ACO-based and FFD-based algorithms, using the CloudSim toolkit [59]. The CloudSim toolkit is an existing cloud computing simulation framework for accomplishing simulation-based trials to percept the actual behavior of the algorithm. It is an entirely adaptable tool practiced for continuous exhibiting, simulation, and investigation of evolving cloud computing frames and application amenities. It permits research community and industry-based developers to focus on explicit system design concerns that they need to explore, without bothering about the low-level aspects linked to Cloud-based frames and facilities. It has been used for the assessment owing to the subsequent reasons [59, 60]:

– Enables the cloud users to demonstrate and simulate huge & extendable virtualized data centers, offering adaptable strategies for allocating the resources to the VMs.

- Helps to model and simulate energy-conscious computational resources, data center network topologies and message-passing applications.
- Provisions the run-time insertion of simulation components, break and continuation of simulation.
- Provides the facility to allocate hosts to VMs as per the customized procedures.
- Supports policies to allocate the host resources to VMs.

6.1 Performance Comparison

Table 1 gives the specification details of the hosts and the VMs in the cluster. Up to 200 VMs and 200 hosts(fireflies) have been simulated and the simulation is repeated for 40 runs.

The Consumed Energy (CE), the number of saved hosts and the number of reduced migrations have been calculated through the proposed FFO-EVMM technique. The energy consumption has also been computed using different node utilization thresholds. The obtained results have been compared to the ACO-based and the FFD-based techniques and are shown in Figs. 3–11.

Figure 3 presents the comparative view of the three techniques, that is, FFD, ACO and FFO-EVMM on the basis of the required number of active hosts over the number of virtual machines as independent axis. It is important to keep a track of the number of hosts operating in the system in order to prevent situations where the probability of most of the hosts sitting idle and consuming unnecessary power is high, which will violate the minimum energy requirement criteria. Upon identification of idle hosts, they are set to sleep mode. Based on the analysis of the obtained results, it is evident that FFO-EVMM technique runs lesser number of active hosts in comparison to the other two techniques. This is because FFO-EVMM runs firefly optimization algorithm that chooses accurate nodes for VM allocation with reduced discovery time resulting in an optimal utilization of the host nodes. It attains global optimization with faster convergence speed. The improvement in the resource utility levels minimizes the number of VM migrations thereby averting energy wastage.

The same is shown through Fig. 4. The graph in Fig. 4 depicts the number of VM migrations done by the three techniques. As observed in the Figs. 3 and 4, the FFO-EVMM technique uses lesser number of hosts and performs lesser number of VM migrations in contrast to FFD and ACO. The capability of the FFO-EVMM to pro-actively discover the best node for VM allocation without compromising with the energy consumption affects the future migration decisions and number of VM migrations required. The overall energy consumption in the system is optimal upon the arrival of the new workload, as all the previous workload allocations to the VMs running on the system nodes have been done considering the energy thresholds. Thereby, the need to incur more and more VM migrations is low while making further workload allocations as per the energy constraints.

Table 1 Simulation parameters

Parameter	Value	Comment
No. of VMs	20–200	Backing Cloud Environment
No. of Hosts	10–200	Hosts running VMs
Bandwidth	2.5 Gbps	Maximum allowed data rate
Host_Types	2	Types of Hosts used
Host_PES	2	Dual-core Hosts
Host_MIPS	1860–2660	MIPS allocated to each Host
Host_RAM	4GB	Primary Memory allocated to each Host
Host_Storage	1 TB	Secondary storage allocated to each Host
VM_Types	4	Types of VMs used
VM_PES	1	Single-core VMs
VM_MIPS	500–2500	MIPS allocated to each VM
VM_RAM	.5GB-4GB	Primary Memory allocated to each VM
VM_SIZE	2.5GB	Secondary Memory allocated to each VM

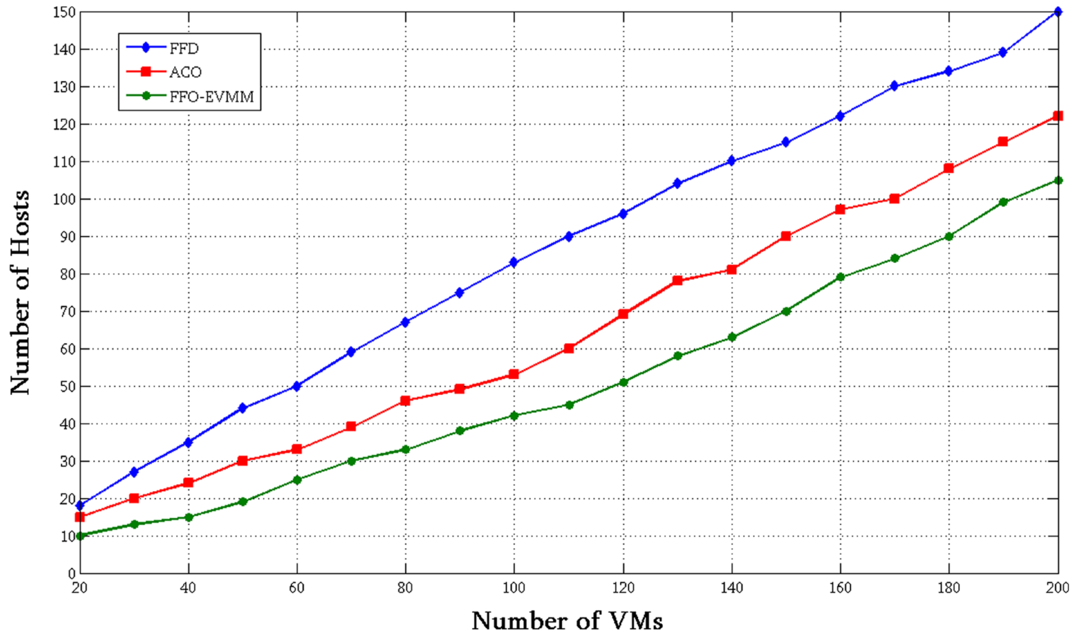


Fig. 3 VMs vs hosts

With the lesser number of required hosts and VM migrations, FFO-EVMM poses lesser energy demand. The energy consumption done by FFO-EVMM is low as compared to FFD and ACO as observed from the graph given in Fig. 5. The tendency to discover and reduce the number of active but idle hosts curtails the energy demand. It attains efficient resource utility levels affecting the number of VM migartions.

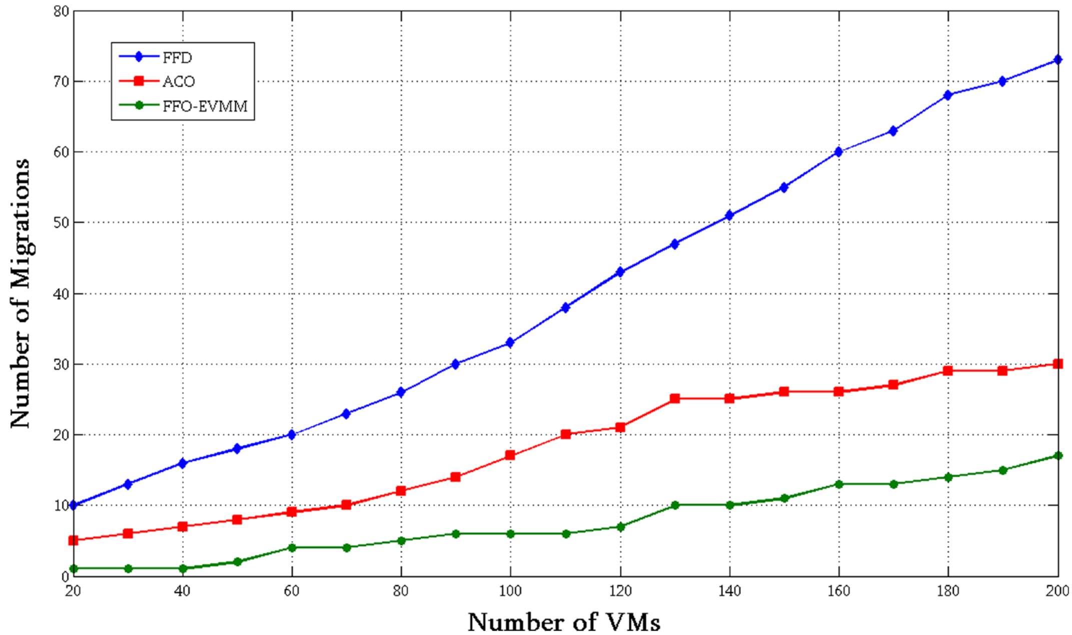


Fig. 4 VMs vs number of migrations

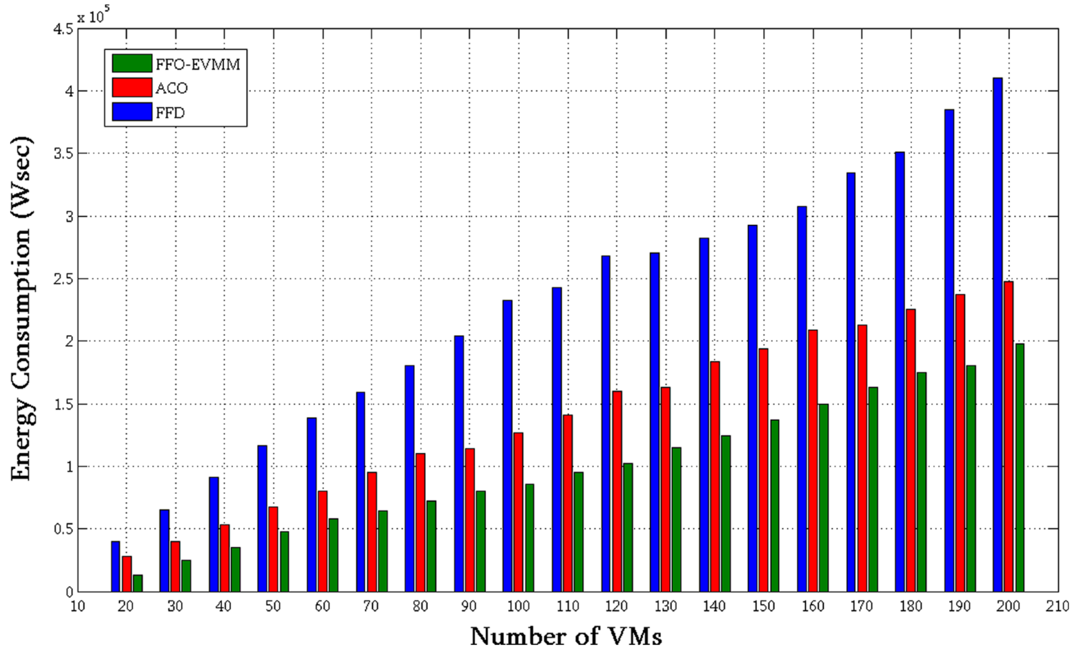


Fig. 5 VMs vs energy consumption

The reduction in the number of VM migrations cuts down the amount of the energy consumed, that would have otherwise been wasted when the VMs are being migrated. Consequently, the required operational energy and the energy consumption level drops down.

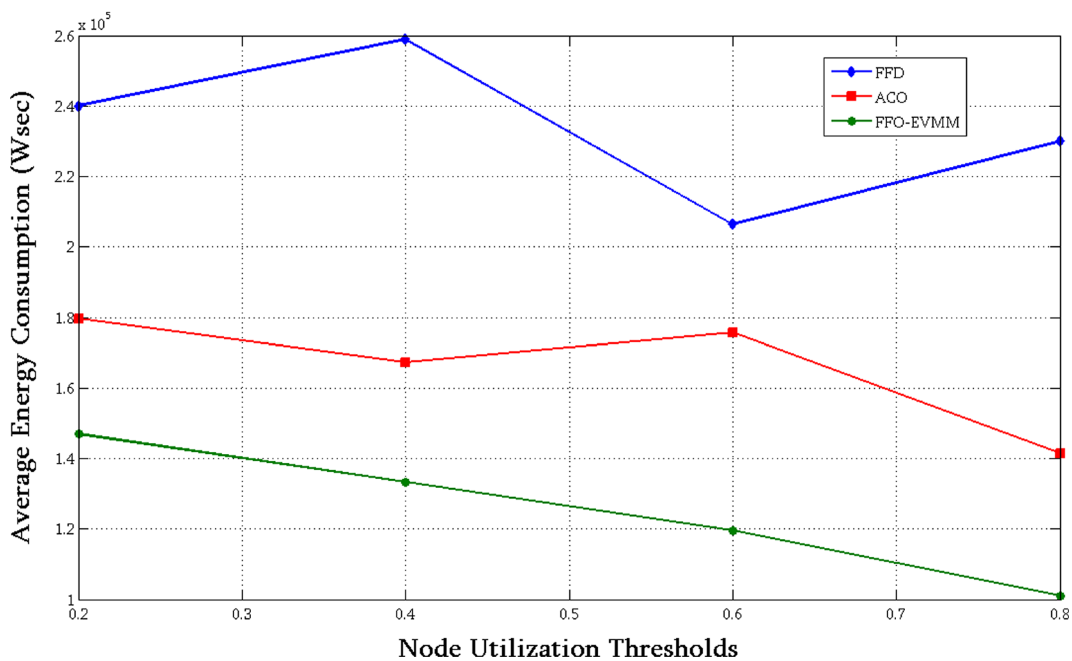


Fig. 6 Thresholds vs avg energy consumption

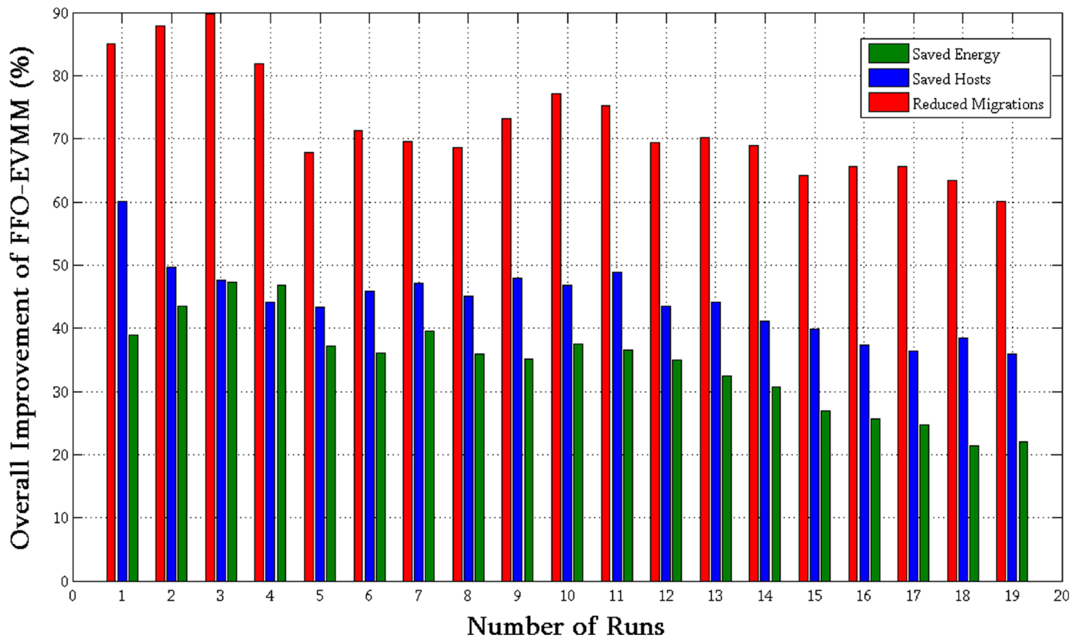


Fig. 7 Overall improvement of FFO-EVMM

Figure 6 depicts the average energy consumption at different threshold values according to the host utilization levels for all the three techniques. It is visible from the graph that as the utilization of the hosts increases from 20 % to 80 %, the threshold values also vary accordingly. Upon simulation, it has been concluded that at different values of thresholds, in FFD and ACO based techniques, there is a fluctuation in the energy consumption done by FFD and ACO. The energy consumed by FFD and ACO

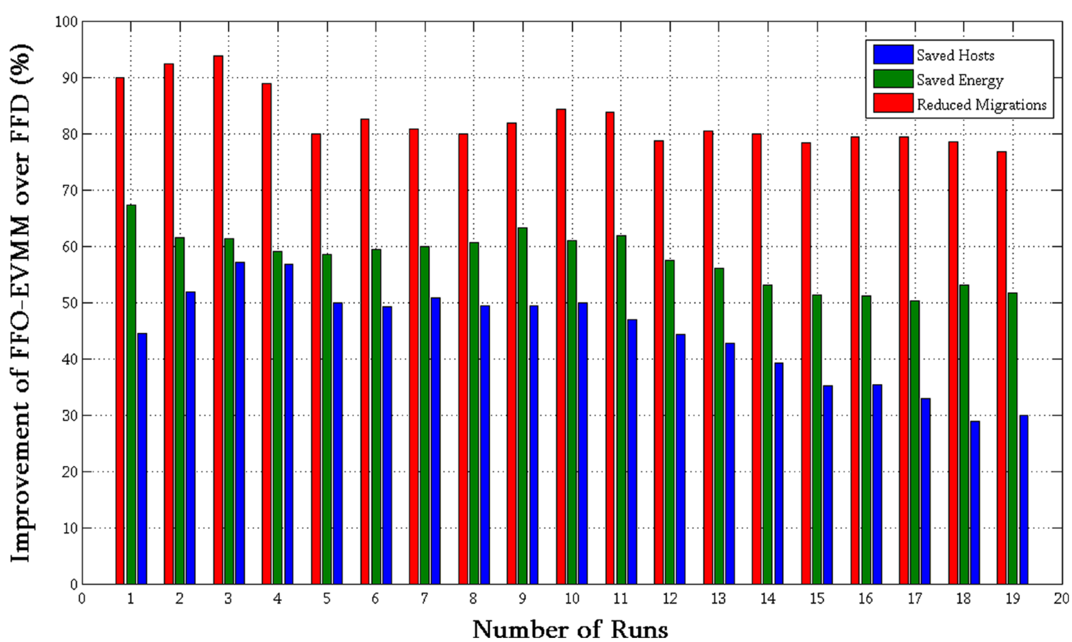


Fig. 8 Improvement graph of FFO-EVMM over FFD

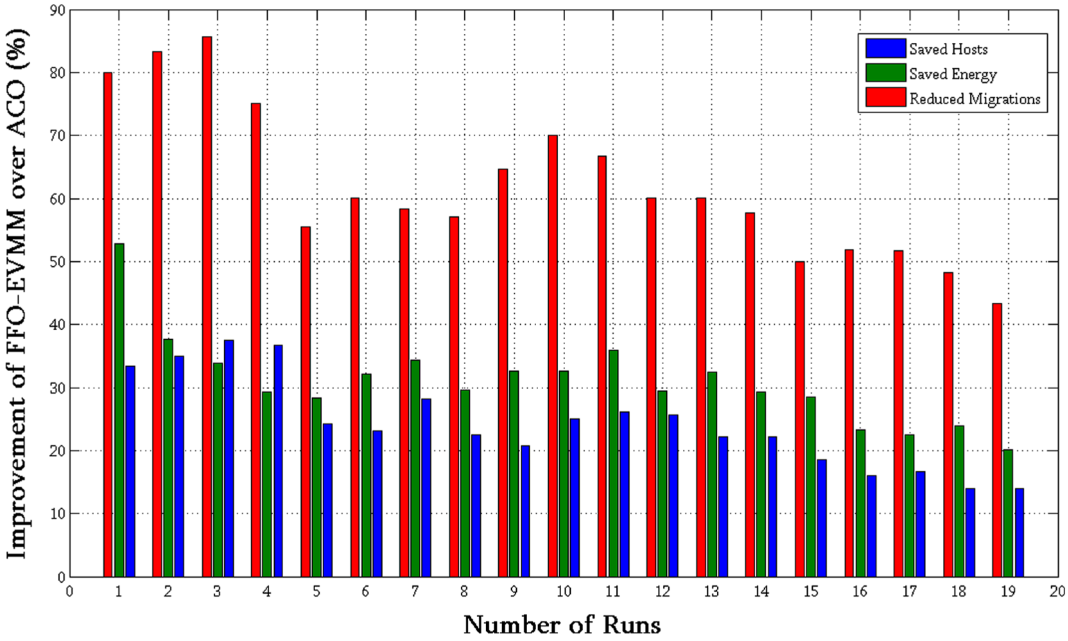


Fig. 9 Improvement graph of FFO-EVMM over ACO

in comparison to our proposed technique follows an irregular behaviour, that is, the energy consumption keeps on increasing or decreasing for different threshold values. Relatively, the energy consumption in the FFO-EVMM technique decreases consistently with varying thresholds. The declining trend in the energy consumption of FFO-EVMM is due to the improvement attained in the host utility levels because of energy-aware VM allocation decisions.

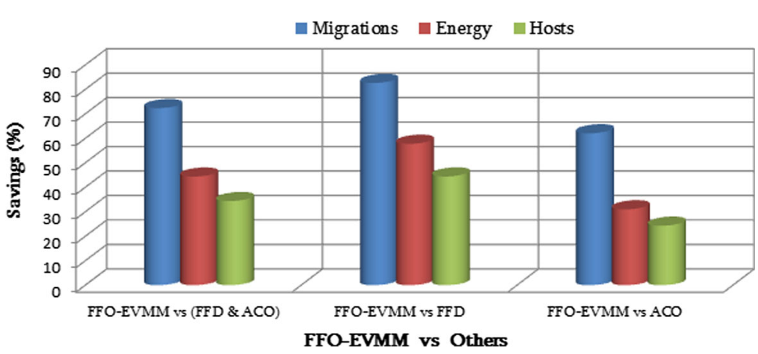
6.2 Discussion

Figures 7–10 vindicate the reliability and competence of our proposed technique. Figure 7 indicates the overall enhancement of the proposed technique over the other two. The outcomes determine that an average of 72.34 % of migrations have been reduced and 34.36 % of hosts have been saved with FFO-EVMM.

Due to the less number of migrations and hosts, an average of 44.39 % of energy has been saved using FFO-EVMM over ACO-based and FFD-based techniques. Further, the comparison of the proposed FFO-EVMM technique has been done to FFD-based and ACO-based techniques separately, to get an insight into its efficacy over each technique.

When, the FFO-EVMM technique is compared to FFD-based technique as shown in Fig. 8, on an average, a reduction of 82.61 % migrations and a saving of 44.43 % hosts & 57.77 % energy have been observed.

Fig. 10 Savings



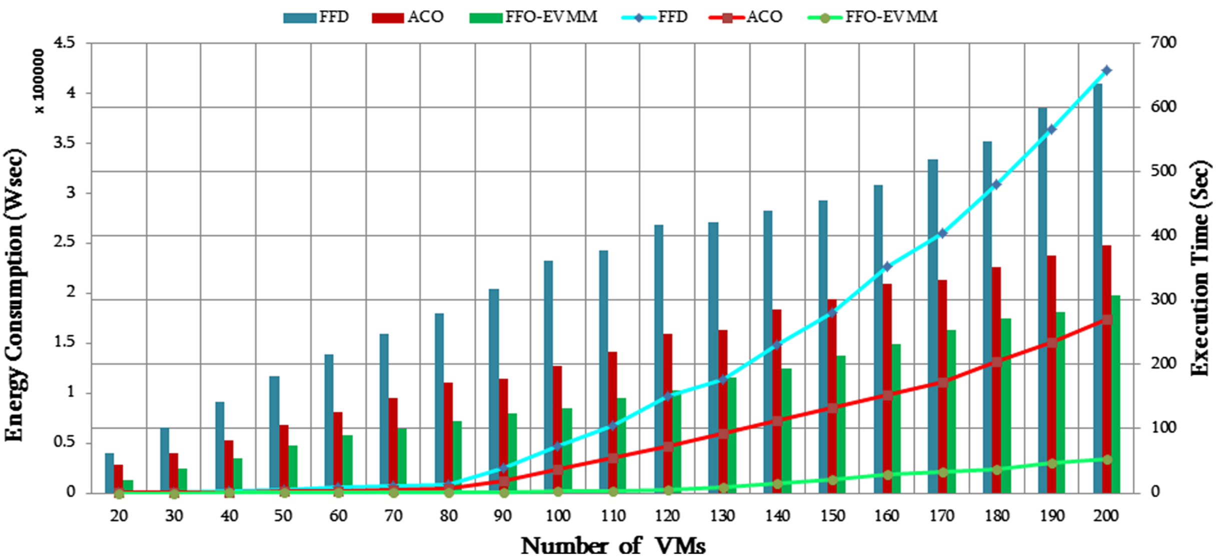


Fig. 11 Performance analysis

Figure 9 denotes the comparison of the FFO-EVMM over ACO-based technique indicating the saved percentage of the nodes and the energy as 24.29 % & 30.99 % respectively, whereas the percentage of the reduced migrations is 62.07 %.

Figure 10 gives the overall and individual saving percentage of the FFO-EVMM over the rest. The improvement in the energy consumption and the decrease in the number of hosts state the effectiveness of the proposed solution showing the optimization with reduced number of migrations. The performance evaluation of VM migration illustrates that the proposed schema can effectively be used for the larger cloud environment, thus making it highly scalable.

Figure 11 shows the run time analysis for all the three approaches. It can be interpreted from the graph that the FFO-EVMM has outperformed the other two techniques. Being based on FFO, the execution time of the proposed technique is better than ACO and FFD as the convergence rate of FFO is very high. The trending curve shows the improvement in execution time of the overall applicability of the proposed algorithm. The lower execution time also decreases the overall complexity of the system.

7 Conclusion and Future Work

Energy efficiency has appeared as the utmost essential design requirements for the current computing systems in recent years. It extends from single servers to data centers and Clouds, as they consume massive volumes of electrical power. For this reason, an effectual energy management is particularly essential for cloud data centers.

Currently, many researchers are focusing and implementing the biologically inspired computing as a preferable paradigm to handle heterogeneity and growing energy crisis with proficiency and without the augmented complication. Likewise, for our work, we have chosen biological behaviour of firefly insects and have devised FFO-based migration technique. The criteria for choosing it is its faster convergence speed and global optimization attainment. Furthermore, it exploits the concept of curtailing the overall upsurge in the incremental power due to the new VM migrations and has never been used previously for VM migration approach. In the scenario of energy consumption by VM migration, a linear model based on FFO is formulated that runs an FFO algorithm which is able to solve the energy consumption issue with the attraction property of fireflies. The capability of fireflies to get attracted towards the brighter fireflies is the basis for considering the FFO.

In other words, this paper has proposed an energy-aware virtual machine migration technique that performs live migration of the VMs from one active node to the other active node. The proposed technique makes use of a bio-inspired Firefly optimization technique to find the best node for the overloaded VMs to be migrated, to achieve energy efficiency in cloud data centers. It maximizes the energy-efficiency through the optimum migration of VMs, thereby improving the resource utilization levels.

The proposed approach can be used as an effective solution for VM Migrations in cloud environment where a large number of nodes are available with the energy restrictions. Improvement in the results with respect to the existing approaches—ACO & FFD, proves the efficacy of the proposed algorithms with higher scalability and lower number of host usage. The proposed technique is better in achieving the energy efficiency as compared to the other techniques as it saves an average of 44.39 % of energy by saving an average of 34.36 % of hosts and by reducing an average of 72.34 % of migrations. Thus, this technique reduces the energy consumption of cloud data centers by saving the nodes and the number of migrations, thereby contributing towards the green computing.

Future work is targeted to study the robustness of FFO-EVMM technique and further expand its performance by verifying it in an existent cloud computing environment like Aneka [55] or some private, public or hybrid cloud. Besides, this technique will additionally be used to design an energy-aware load balancing technique for cloud computing. That load balancing technique will experimentally be investigated for performance and efficiency using a real environment.

Acknowledgments This research was supported by University Grants Commission (UGC) sponsored major research project Energy Aware Resource Scheduling for Cloud Computing under $F.No.41-629/2012(SR)$. Authors' addresses: Nidhi Jain Kansal, Computer Science and Engineering Department, Thapar University, India, Email:nidhi.kavya@thapar.edu; Dr. Inderveer Chana, Computer Science and Engineering Department, Thapar University, India, Zip Code-147004, Email: inderveer@thapar.edu

References

1. Rings, T., Caryer, G., Gallop, J., Grabowski, J., Kovacikova, T., Schulz, S., Stokes-Rees, I.: Grid and cloud computing: opportunities for integration with the next generation network. J. Grid Comput. **7**(3), 375–393 (2009)
2. Min, C., Kim, I., Kim, T., Eom, Y.I.: VMMB: Virtual Machine Memory Balancing for Unmodified Operating Systems. J. Grid Comput. **10**(1), 69–84 (2012)
3. Rodero, I., Viswanathan, H., Lee, E.K., Gamell, M., Pompili, D., Parashar, M.: Energy-efficient thermal-aware autonomic management of virtualized HPC cloud infrastructure. J. Grid Comput. **10**(3), 447–473 (2012)
4. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press, UK (2008)
5. Yang, X.S.: Firefly algorithms for multimodal optimization. In: Watanabe, O., Eugmann, T. (eds.) The proceedings of 5th Symposium on Stochastic Algorithms, Foundations and Applications. Lecture Notes in Computer Science, vol. 5792, pp. 169–178. Springer, Berlin (2009)
6. Kansal, N.J., Chana, I.: Artificial bee colony based energy-aware resource utilization technique for cloud computing. Concurreny and Computation: Practice and Experience (CCPE), vol. 27, Issue 5. Wiley Online Library, pp. 1207–1225. doi:10.1002/cpe.3295 (2014)
7. Yang, X.S., He, X.: Firefly algorithm: recent advances and applications. Int. J. Swarm Intell. **1**(1), 36–50 (2013). doi:10.1504/IJSI.2013.055801
8. Khaze, S.R., Maleki, I., Hojjatkhah, S., Bagherinia, A.: Evaluation the efiiciency of artificial bee colony and the firefly algorithm in solving the continuous optimization problem. Int. J. Comput. Sci. Appl. (IJCSA) **3**(4) (2013)
9. Basu, B., Mahanti, G.K.: Firefly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna. Progr. Electromagn. Res. B **32**, 169–190 (2011)
10. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, vol. 25, pp. 599–616 (2009). 6
11. Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., Warfield, A.: Live migration of virtual machines. In: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation, pp. 273–286 (2005)
12. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. **1**(1), 7–18 (2010)
13. Ghribi, C., Hadji, M., Zeghlache, D.: Energy efficient VM scheduling for cloud data centers: exact allocation and migration algorithms. In: Proceedings of 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (2013). doi:10.1109/CCGrid.2013.89
14. Man, C.L.T., Kayashima, M.: Virtual machine placement algorithm for virtualized desktop infrastructure. In: Proceedings of IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Beijing, pp. 333–337 (2011)
15. Voss, S.: Meta-heuristics: the state of the art. In: Nareyek, A. (ed.) Local Search for Planning and Scheduling. Lecture Notes in Artificial Intelligence, vol. 2148, pp. 1–23. Springer, Berlin (2001)
16. Weiss, A.: Computing in the clouds. Networker Mag. **11**(4), 16–25 (2007)
17. Beloglazov, A., Buyya, R.: Energy efficient allocation of virtual machines in cloud data centers. In: 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), Melbourne, Australia, pp. 577–578 (2010)
18. Lee, Y.C., Zomaya, A.Y.: Energy efficient utilization of resources in cloud computing systems. J. Supercomput. **60**(2), 268–280 (2012)

19. Dorigo, M., Colorni, A.: The ant system optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B **26**(1), 1–13 (1996)
20. Dorigo, M., Gambardella, L.M.: Ant colonies for the traveling salesman problem. BioSystems **43**, 73–81 (1997)
21. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report - TR06, October (2005)
22. Ludwig, S.A., Moallem, A.: Swarm intelligence approaches for grid load balancing. J. Grid Comput. **9**(3), 279–301 (2011)
23. Blum, C.: Ant colony optimization: introduction and recent trends. Phys. Life Rev. **2**, 353–373 (2005)
24. Tarighi, M., Motamedi, S.A., Sharifian S.: A new model for virtual machine migration in virtualized cluster server based on fuzzy decision making. J. Telecommun. **1**(1), 40–51 (2010)
25. Wood, T., Shenoy, P.J., Venkataramani, A., Yousif, M.S.: Black-box and gray-box strategies for virtual machine migration. In: Proceedings of 4th USENIX Symposium on Networked Systems Design and Implementation (NSDI'07), Cambridge, pp. 229–242 (2007)
26. Wood, T., Shenoy, P.J., Venkataramani, A., Yousif, M.S.: Sandpiper: black-box and gray-box resource management for virtual machines. Comput. Netw. **53**(17), 2923–2938 (2009)
27. Lim, M.Y., Rawson, F., Bletsch, T., Freeh, V.W.: PADD: power aware domain distribution. In: Proceedings of the 29th IEEE International Conference on Distributed Computing Systems (ICDCS'09), Montreal, pp. 239–247 (2009)
28. Verma, A., Ahuja, P., Neogi, A.: pMapper: power and migration cost aware application placement in virtualized systems. In: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (Middleware'08), Leuven, Belgium, pp. 243–264. Springer, Berlin (2008)
29. Tolia, N., Wang, Z., Marwah, M., Bash, C., Ranganathan, P., Zhu, X.: Delivering energy proportionality with non energy-proportional systems - optimizing the ensemble. In: Workshop on Power Aware Computing and Systems (HotPower '08), San Diego (2008)
30. Feller, E., Rilling, L., Morin, C.: Snooze: a scalable and autonomic virtual machine management framework for private clouds. Rapport de recherche RR-7833, INRIA (2011)
31. Mastroianni, C., Meo, M., Papuzzo, G.: Self-economy in cloud data centers: statistical assignment and migration of virtual machines. Euro-Par 2011 Parallel Processing, pp. 407–418 (2011)
32. Marzolla, M., Babaoglu, O., Panzieri, F.: Server consolidation in clouds through gossiping. In: IEEE International Symposium on the World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–6. IEEE (2011)
33. Murtazaev, A., Oh, S.: Sercon: server consolidation algorithm using live migration of virtual machines for green computing. IETE Tech. Rev. **28**(3), 212–231 (2011)
34. Verma, A., Dasgupta, G., Nayak, T., De, P., Kothari, R.: Server Workload Analysis for Power Minimization Using Consolidation, p. 28. USENIX Association, Berkeley (2009)
35. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. J. Futur. Gener. Comput. Syst. **28**(5), 755–768 (2012)
36. Beloglazov, A., Buyya, R.: Energy efficient resource management in virtualized cloud data centers. In: Proceedings of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 826–831 (2010). doi:10.1109/CCGRID.2010.46
37. Cardosa, M., Korupolu, M., Singh, A.: Shares and utilities based power consolidation in virtualized server environments. In: Proceedings of IEEE, pp. 327–334 (2009). doi:10.1109/INM.2009.5188832
38. Goiri, I., Berral, J.L., Fitó, O., Julià, F., Nou, R., Guitart, J., Gavalda, R., Torres, J.: Energy-efficient and multi-faceted resource management for profit-driven virtualized data centers. Futur. Gener. Comput. Syst. **28**(5), 718–731 (2012)
39. Graubner, P., Schmidt, M., Freisleben, B.: Energy-efficient management of virtual machines in eucalyptus. In: Proceedings of the 2011 IEEE 4th International Conference on Cloud Computing, ser. CLOUD '11, pp. 243–250. [Online]. Available: doi:10.1109/CLOUD.2011.26 (2011)
40. Mehta, S., Neogi, A.: ReCon: a tool to recommend dynamic server consolidation in multi-cluster data centers. In: Proceedings of the IEEE Network Operations and Management Symposium, NOMS'08, Salvador (2008)
41. Dong, J., Jin, X., Wang, H., Li, Y., Zhang, P., Cheng, S.: Energy-saving virtual machine placement in cloud data centers. In: Proceedings of 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (2013)
42. Xiaoli, W., Zhanghui, L.: An energy-aware VMs placement algorithm in cloud computing environment. In: Proceedings of the Second International Conference on Intelligent System Design and Engineering Application. IEEE (2012)
43. Vu, H.T., Hwang, S.: A traffic and power-aware algorithm for virtual machine placement in cloud data center. Int. J. Grid Distrib. Comput. **7**(1), 21 (2014)
44. Beloglazov, A., Buyya R.: Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In: Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science, MGC '2010, Bangalore (2010)
45. Nathuji, R., Schwan, K.: Virtualpower: coordinated power management in virtualized enterprise systems. ACM SIGOPS Oper. Syst. Rev. **41**(6), 265–278 (2007)
46. Sekhar, J., Jeba, G.: Energy efficient VM live migration in cloud data centers. Int. J. Comput. Sci. Netw. (IJCSN) **2**(2), 71–75 (2013)
47. Jo, C., Gustafsson, E., Son, J., Egger, B.: Efficient live migration of virtual machines using shared storage. In: Proceedings of the 9th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, VEE'13, Houston, pp. 41–50 (2013)
48. Strunk, A., Dargie, W.: Does live migration of virtual machines cost energy? In: Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications, pp. 514–521 (2013)
49. Bila, N., Lara, E.D., Joshi, K., Lagar-Cavilla, H.A., Hiltunen, M., Satyanarayanan, M.: Jettison: efficient idle desktop consolidation with partial vm migration. In: Proceedings of the 7th ACM European Conference on Computer Systems. EuroSys '12, New York, pp. 211–224 (2012)

50. Jung, G., Hiltunen, M., Joshi, K., Schlichting, R., Pu, C.: Mistral: dynamically managing power, performance, and adaptation cost in cloud infrastructures. In: Proceedings of 30th IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 62 –73 (2010)
51. Setzer, T., Stage, A.: Decision support for virtual machine reassignments in enterprise data centers (2010)
52. Yue, M.: A simple proof of the inequality FFD(L) ≤ (11/9)OPT(L) + 1, for all L, for the FFD bin-packing algorithm. Acta Math. Appl. Sin. **7**(4), 321–331 (1991)
53. Dressler, F., Akan, O.B.: A survey on bio-inspired networking. Comput. Netw. J. **54**(6), 881–900 (2010)
54. Meisel, M., Pappas, V., Zhang, L.: A taxonomy of biologically inspired research in computer networking. Comput. Netw. J. **54**(6), 901–916 (2010)
55. Vecchiola, C., Chu, X., Buyya, R.: Aneka: a software platform for.NET-based cloud computing. High Performance & Large Scale Comp. Advances in Parallel Computing 267–295 (2009)
56. Kaur, T., Chana, I.: Energy efficiency techniques in cloud computing - a survey and taxonomy. ACM Comput. Surv. **48**(2, Article 22, 46 pp) (2015)
57. Beloglazov, A., Buyya, R., Lee, Y.C., Zomaya, A.: A taxonomy and survey of energy efficient data centers and cloud computing systems. Adv. Comput. **82**(2), 47–111 (2011)
58. Minas, L., Ellison, B.: Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers (2009)
59. Buyya, R., Ranjan, R., Calheiros R.N.: Modeling and simulation of scalable cloud computing environments and the cloudSim Toolkit: challenges and opportunities. In: Proceedings of the 7th High Performance Computing & Simulation Conference (HPCS 2009), pp. 1–11, Leipzig, Germany, pp. 21–24. IEEE Press, New York (2009)
60. Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F.D., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. In: Software: Practice and Experience (SPE), vol. 41, Issue 1, pp. 23–50, ISSN 0038–0644. Wiley Press, New York (2011)
61. Beloglazov, A., Buyya, R.: 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 (CCPE), vol. 24, Issue 13, pp. 1397–1420. Wiley, New York (2012)
62. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control. Syst. Mag. **22**, 52–67 (2002)
63. Zhou, A., Wang, S., Zheng, Z., Hsu, C., Lyu, M., Yang, F.: On cloud service reliability enhancement with optimal resource usage. In: IEEE Transactions on Cloud Computing, vol. PP, no. 99, pp. 1–1 (2014). doi:10.1109/TCC.2014.2369421