Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH Open Access

Temperature and energy-aware consolidation algorithms in cloud computing

Maede Yavari[1], Akbar Ghaffarpour Rahbar[1*] and Mohammad Hadi Fathi[2]

Abstract

Cloud computing provides access to shared resources through Internet. It provides facilities such as broad access, scalability and cost savings for users. However, cloud data centers consume a significant amount of energy because of inefficient resources allocation. In this paper, a novel virtual machine consolidation technique is presented based on energy and temperature in order to improve QoS (Quality of Service). In this paper, two heuristic and meta-heuristic algorithms are provided called HET-VC (Heuristic Energy and Temperature aware based VM consolidation) and FET-VC (FireFly Energy and Temperature aware based VM Consolidation). Six parameters are investigated for the proposed algorithms: energy efficiency, number of migrations, SLA (Service Level Agreement) violation, ESV, time and space complexities. Using the CloudSim simulator, it is found that energy consumption can be alleviated 42% and 54% in HET-VC and FET-VC, respectively using our proposed methods. The number of VM migrations is reduced by 44% and 52% under HET-VC and FET-VC, respectively. The HET-VC and FET-VC can improve SLA violation by 62% and 64%, respectively. The Energy and SLA Violations (ESV) are improved by 61% under HET-VC and by 76% under FET-VC.

Keywords: Cloud computing, Consolidate virtual machines, Energy consumption, Meta-heuristic method, FireFly algorithm

Introduction

Today cloud computing is an important extensible computing method in information technology. In cloud computing, virtualized resources are often provided as processing services through communication networks. Cloud computing delivers three basic services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [1]. The basis of giving services to a customer is to provide on-demand resources based on pay-as-you-go without requiring any especial equipment by the customer or being aware of the location of these processors [1, 2]. The main purpose of cloud computing is to attain a huge amount of virtualization computing resources.

Cloud computing has significant benefits, especially for IT industry, mainly reducing cost, providing scalability tailored to the needs of customers' service requirements, and so on [2]. The trend of customer increase using cloud services has forced cloud service providers to extend capacity and the number of data centers in world [3]. On the other hand, this growth has led to consumption of gallons of energy to supply power consumption [4]. A data center on average consumes energy approximately equal to 25,000 households [5]. This increase in energy consumption and cost is not suitable for a cloud service provider. In addition, energy consumption results in considerable carbon dioxide (CO2) dissemination that increases the greenhouse effect. It is predicted that data centers in USA will consume 140 billion Kilowatts per year by 2020 [6]. Therefore, governments increase pressure on cloud service providers to optimize energy consumption in order to reduce CO2 effecting the climate change [7]. As a result, researchers have decided to study cloud computing environment to address this challenge.

Data centers waste energy due to the inefficiency of hardware resources such as shortage and inadequacy of cooling system [8], network equipment [9], servers [10]. and software resources [11]. Servers are the main

* Correspondence: ghaffarpour@sut.ac.ir
[1]Faculty of Electrical Engineering, Sahand University of Technology, Tabriz, Iran
Full list of author information is available at the end of the article

resources of energy usage and their performances must be optimized through software management in order to reduce energy consumption.

Virtualization technology is the inseparable part of cloud computing, where hypervisor allows to create multiple Virtual Machine (VM) instances on a Physical Machine (PM), thus providing improved utilization of resources [4]. When a customer needs more resources, his/her request must be resolved swiftly. Since there are multiple VMs on a PM, increase of resources for one or more VMs may lead to exceed the amount of load on the PM beyond its capacity. This situation of PM, called overload, can increase response times, time-outs or failure. On the other hand, if a customer has some unused resources, they must be released. However, releasing resources from many VMs may intensively decrease the usage of resources so that the PM tends to the idle state. This situation of PM is called under-load.

The research conducted on more than 5000 servers over a period of 6 months indicate that most servers are usually active, but most of the time their utilization are 10%–50% of their total capacity [12]. Another research expresses that energy consumption of idle servers is about 70% rather than full capacity [13]. Therefore, avoiding servers from being over-loaded can save energy, decrease SLA (Service Level Agreement) violation, and reduce temperature. In addition, recognition of idle and under-loaded PMs and switching them into either sleep mode or hibernate mode could be another utility for the purpose.

Thus, there will be a small number of active servers that can help for better energy management. Timely migration of VMs can prevent from overloading and under-loading states. Live migration reallocates running VMs dynamically from one PM to another PM. Migration can consolidate VMs to minimal PMs. Migration tries to keep application performance requirements and delivers services to the customer with minimal side effects. However, migration provides downtime on the delivered services considering its low impact. Although a user may not understand this downtime, there is a violation for SLA [14, 15]. Another effect of migration is an increase in utilization of resources and all chips, thus increasing energy consumption. Therefore, the best solution is to find a trade-off between the number of migrations and energy consumption and SLA.

Some researches [16, 17] indicate that high number of migrations either increases costs of computing resource and decreases performance of systems, thus affecting quality of service (QoS). It is essential for cloud computing environments to protect QoS defined by SLAs between a provider and a customer; therefore, the cloud provider must minimize energy consumption while meeting SLAs.

Keeping a trade-off between the number of migrations, performance and energy is an optimization problem. Since regulating workloads on servers is an NP-Hard problem (Man, 2011) [18], heuristic and meta-heuristic methods are often proposed for solving them.

Zahedi et al. [19] have proposed the Dynamic Threshold Maximum Fit (DthMf) algorithm that considers temperature and classification of servers. It can reduce energy consumption and number of migrations. The DthMf can reduce energy consumption rather than the method in [7, 20]. In [19], servers are divided into three categories based on their speeds in Million Instruction Per Second (MIPS), and their energy consumption as:

(1) High performance server: with high MIPS and less heat produced compared to other categories.
(2) Low performance server: with low MIPS and more heat produced compared to high performance server.
(3) Medium performance server: less heat produced rather than low performance server and more heat rather than high performance server. It has lower MIPS than high performance server and higher MIPS than low performance server.

If VMs are to be consolidated into low performance servers in comparison with high performance servers, their whole energy consumption comes close together. However, the heat problem will reduce dramatically using high performance servers [19].

The DthMf tries to migrate VMs from low performance servers to high performance servers, and switches almost all low performance servers to the sleep mode. The migration problem is divided into four sub-problems as [20]:

1 Overload Server detection: select PMs with higher temperature rather than a given threshold temperature.
2 Under-load server detection: select servers with the lowest CPU utilization. This sub-problem selects low utilized servers from firstly low performance servers, then medium, and finally from high performance servers.
3 VM selection: Zahedi et al. [19] has introduced a new algorithm for this selection called Maximum Fit (MF) that selects the VM with the lowest deviation between its utilization and threshold.
4 VM placement: select destination server for the VM, where it is tried to choose from high performance servers primitively, then medium, and finally low performance servers.

Many researches have been carried out in the field of energy optimization, but most of them do not consider the effect of multiple resources. Moreover, they try to reduce energy consumption without focusing on the temperature and type of servers. This has motivated us to introduce two heuristic and meta-heuristic methods that reduces energy consumption by considering multiple resources.

The objective of this work is to propose two energy and temperature-aware algorithms based on heuristic and meta-heuristic (based on nature-inspired Firefly Optimization (FFO)) methods that can reduce energy consumption in cloud computing environments. In this paper, CPU, memory and temperature performances are considered and it is tried to optimize their usages. Some challenges about energy management are discussed and two policies and algorithms for making better virtualization and reducing energy consumption in cloud data centers, are proposed, thus getting closer to green cloud computing.

The main contributions of this work are as follows: (1) propose two heuristic and meta-heuristic algorithms for the problem of efficient dynamic VM consolidation, and (2) consider temperature and type of servers in the proposed algorithms for reducing temperature of data centers and improving resource utilization.

The rest of this paper is organized as follows. The related works are studied in Section 2. In Section 3, the system model is presented. Section 4 presents the firefly algorithm. The proposed methods are described in Section 5. Section 6 shows simulation results and evaluation of the proposed methods. Finally, conclusions are proposed in Section 7.

Related work

In past years, many researches have been carried out in the field of energy efficiency and resource utilization for cloud computing environments. Researches are either based on building less consuming equipment, [16, 21], or based on optimization of resource allocation [22, 23].

Wang et al. (Wang, Liu, Chen, & [24]) have proposed a distributed approach, where the VM placement phase is based on a non-centralized agent algorithm that makes a local negotiation for the migration phase. This algorithm considers an agent on every PM, where the agent tries to select the biggest VM to host. The agent determines a bid about its suggestion for the VM that tends to host, and broadcasts that bid to all other agents. If an agent has a lower bid rather than the arrived bid, it sends an acknowledgment message for the sender agent; otherwise, it sends a negative acknowledgment. In the mentioned algorithm, VMs are placed on the PM with the best bid. This method provides high traffic between PMs because of bids, acknowledgment and negative-acknowledgment messages, thus increasing energy consumption and temperature of network devices.

The work in [16] has presented a Modified Particle Swarm Optimization (MPSO)-algorithm for VM consolidation and live migration. It considers the multiple-resource threshold and Euclidean distance as an optimum factor. The lower the factor, the more optimal VM placement. It considers a fixed threshold for processor and disk. However, the cloud environment is a dynamic environment and fixed parameters cannot behave as well as dynamic thresholds.

Younge et al. [22] have proposed green cloud computing framework to reduce power consumption. Two power-based and temperature-based approaches for reducing total power of servers and temperature of data centers have been introduced. The authors have also proposed a new approach to reduce the size of VM in order to reduce migration time.

Beloglazov and Buyya [20] have defined several forms of optimal online and offline algorithms for VMs migration and the consolidation problem. They have proposed a novel adaptive heuristic solution for the problem of energy consumption and efficient dynamic VMs consolidation, where upper and lower thresholds are set periodically based on the recent resource utilization, thus providing a tradeoff between performance and energy consumption. They have also introduced a dynamic server consolidation framework [7] that tries to keep server utilization between upper and lower thresholds of CPU utilization. This algorithm is Modified Best Fit Decreasing (MBFD) for the VM placement problem in order to reduce energy and minimize the number of SLA violations. [25] have worked on dynamic upper and lower thresholds algorithms with an approach to reduce energy consumption without considering trade-off between performance and migration, and migration cost.

The work in [26] considers heat island for servers, where in a high-temperature environment, it can cause serious problems. This work proposes a virtual machine placement algorithm for energy saving considering server reliability. The authors try for rapid elimination of a server heat by placing virtual machines on few servers. It considers relationship between server utilization and power consumption, the relationship between server utilization and server heat, the relationship between server utilization and power consumption of cooling systems, the relationship between server heat and server reliability and availability of system at a specific time. Also, it considers redundancy of the servers in order to secure the reliability of servers operating in a high temperature environment. This work has introduced a heuristic approach to solve virtual machine placement with trying to find the best solution based on power consumption and temperature of the server rack to avoid of heat island.

Research in [27] is based on fast and accurate models with awareness of the relationships with power of non-traditional parameters (such as temperature and frequency). This work combines both energy and thermal-aware strategies. The authors work on a set of single-objective and multi-objective best-fit decreasing based policies and a meta-heuristic-based optimization policy that relies on the simulated annealing algorithm. All methods optimize the energy consumption of the data center considering both IT and cooling parameters. A cooling strategy based on VM placement is another contribution of the paper that tries to maintain overall data center at a safe temperature. The algorithm aims to find the highest cooling set point of the computer room air conditioning units to ensure a safe operation for the data center infrastructure. Finally, the cooling set point is set to the lowest value within the maximum cooling set point for all the servers, guaranteeing that the infrastructure operates below the maximum safe CPU temperature. Authors work on VM placement and only detect overload servers.

The problem of dynamic placement of applications in virtualized heterogeneous systems has been formulated in [28] as continuous optimization with considering power consumption and performance. The placement problem is based on the bin-packing problem with variable bin sizes and costs. This work uses a heuristic method to solve the problem. Live VM migration has been performed at each time frame for obtaining a new placement. However, the proposed algorithm does not consider SLA.

The authors in [29] have proposed new strategies for VM consolidation. They consider three types of consolidations: static (monthly, yearly), semi-static (days, weeks) and dynamic (minutes, hours). However, they only consider static and semi-static consolidation techniques, whereas the studied environment is completely dynamic, and therefore, the static and semi-static evaluation cannot result in realistic results.

Since nature inspired computing (such as Ant Colony Optimization (ACO) [30], Artificial Bee Colony (ABC) [31] and Firefly Optimization (FFO) [32]) are self-organizing, self-repairing, navigating and flourishing with their local knowledge and without any centralized control [33], many researchers choose them to solve the VM consolidation and energy management problems.

Farahnakian et al. [34] have proposed a method for improving consolidation of VMs for green cloud computing using the ant colony system named ACS-VMC. They use Linear Regression based on CPU Usage Prediction (LIRCUP) to detect overloaded hosts, and then find the best PM for VM placement using the ant colony. Finding near-optimal solution can improve energy consumption, while maintaining the agreed level of performance with users.

Kansal and Chana [35] have designed an artificial bee colony based on energy-aware resource utilization technique to allocate jobs to resources. Their model manages cloud resources and increases their utilization. This model can also reduce energy consumption by proper VM consolidation based on the past resource utilization and energy consumption data. It also tries to maintain the performance of user applications and to sleep idle nodes.

In [36], an energy-aware virtual machine migration has been introduced based on the Firefly algorithm. Their problem formulation and energy model are based on [35]. This technique considers two different types of workloads as CPU-intensive and memory-intensive. It migrates most-loaded VMs from a node with high-energy consumption to an active node with the least-energy consumption, where the selection criteria are based on the firefly algorithm.

In Fathi, Khanli (Fathi, Khanli 2018) [37] an energy-aware virtual machine consolidation method has been proposed using Harmony Search Algorithm (HSA). They claim that HSA has proven its efficiency in power problems. However, their simulations are weak and have only evaluated one model of random workload.

System model

The simulation system is in the IaaS environment with k data centers each containing N disparate physical Each PM is defined by three resources as: CPU, memory and network bandwidth. There are two characteristics for a CPU: number of cores and computation speed as million instructions per second (MIPS). Servers do not have local disks, and their storages are located on a storage area network (SAN). This easily enables live migration of VMs.

The types of application workloads are not considered. Properties of PMs are given in Table 1 based on [22] and properties of VMs are based on the Amazon EC2 instance type, but with this difference that all VMs are single core (see Table 2).

Table 1 Characteristics of servers (PMs)

Server	HP ProLiant ML110 G5	HP ProLiant DL360 G7	HP Proliant DL360 Gen9
Processor (MIPS)	2660	3067	2300
Number of core	2	12	36
Memory (GB)	4	16	64
Network BW (GB/s)	1	1	1

Table 2 Characteristics of Virtual Machines (VMs)

Virtual machines	Large	Medium	Small	Micro	Nano
Processor (MIPS)	2000	1000	1000	500	250
Memory (GB)	2048	2048	1024	1024	512
Network BW (GB/s)	1	1	1	1	1

In this paper, the system model is based on [19, 20]. The mentioned system model has two managers: global and local. The local manager is one part of VMM (Virtual Machine Manager) and gathers information of each PM such as resource utilization, resizing VMs based on user needs, decides for migrating VMs (but VMM migrates VMs) and so on. The global manager is on the master node and collects information of all local managers. The proposed optimization algorithms are executed on this manager.

Energy consumption metric

In a server, CPU is the basic power consumer resource compared to other resources such as memory, disk storage and network interfaces. The work in [28, 38] indicates that Application on Dynamic Voltage and Frequency Scaling (DVFS) has near linear power-to-frequency relationship for CPU. Hence, the power consumption of other resources can be ignored. Anton Beloglazov in [7] has proposed Eq.(1) for power consumption:

$$p(u) = K \times p_{max} + (1\text{–}K) \times p_{max} \times u \tag{1}$$

where p_{max} is the power consumption of server when its CPU utilization is 100%, K is the power consumption of an idle server expressed in percentage, and u is the CPU utilization. Workload varies during time so that CPU utilization may be variable. Therefore, total energy consumption is defined as a function of time as follows:

$$E = \int_{t_0}^{t_1} p(u(t))dt. \tag{2}$$

VM migration cost metric

In live migration, a VM is reallocated from its PM to another suitable PM when both of PMs and VM are at the running state. Although there is not considerable effect at user level, live migration provides bad impact on the performance of running application at the migration phase, and therefore, the less time for migration is better. In addition, migration affects both performance of source and destination PMs. The work in [20] shows that a single live migration approximately has 10% CPU overhead that leads to violation of SLA. Since migration causes degradation of performance and SLA violation, the number of live VM migrations must be reduced. The work in [20] defines migration time and performance degradation experienced by VM j as follows:

$$T_{m_j} = \frac{M_j}{B_j} \tag{3}$$

$$U_{d_j} = 0.1 \int_{t_0}^{t_0+T_{m_j}} u_j(t)dt \tag{4}$$

where U_{d_j} is total degradation of performance by VM j, parameter t_0 is the start time of migration, T_{m_j} denotes the finish time of migration, $u_j(t)$ is the CPU utilization of VM j, M_j is the amount of memory used by VM j, and B_j is the available network bandwidth.

Service level agreement (SLA) violation metric

SLA has a direct relation with provided QoS, where high QoS is one of the basic parameters for holding and attracting customers. In the cloud environment, SLA includes different parameters such as minimum throughput, minimum bandwidth, maximum response time and maximum downtime. Since these parameters are dependent on the application type, a definition for autonomous metrics for SLA assessment in the IaaS level is required. Two parameters have been defined in [20] for SLA:

1. SLA Violation Time per Active Host (SLATAH): percentage of time that the CPU utilization of a server is 100%. This is defined because at the full CPU capacity, a PM cannot provide the required performance by VMs (see Eq.(5)).
2. Performance Degradation due to Migration (PDM): total performance degradation due to live migration experienced by VM j (see Eq.(6)):

$$SLATAH = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{s_i}}{T_{a_i}} \tag{5}$$

$$PMD = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{d_j}}{C_{r_j}}, \tag{6}$$

where N is the number of servers, T_{s_i} is total time that CPU utilization of server j is 100%, T_{a_i} is total time that server j is active, M is the number of VMs, C_{d_j} is the estimated performance degradation of VM j during migration time, and C_{r_j} is total amount of CPU capacity requested by VM j. In [10], C_{d_j} is set to 10% of CPU utilization in MIPS during all migrations of VM j. Authors in [20] have defined a new combined metric that encompasses both performance degradation parameters as in Eq.(7):

$$SLAV = SLATAH \times PDM \tag{7}$$

The ESV metric

The fourth parameter is Energy and SLA Violations (ESV) introduced in [20] and used in [19] defined as Eq.(8), where the energy consumption parameter (*Energy*) was defined in Section 3.1. Note that energy consumption and SLAV are the main metrics because they must be in the minimum state to improve efficiency of resources.

$$ESV = Energy \times SLAV \tag{8}$$

Time and space complexity

Time complexity describes the amount of time that takes to run an algorithm and is based on the number of steps that an algorithm uses on a particular input to obtain result. Space complexity is the amount of space or memory that an algorithm needs for gaining results for a particular input set. Both of the time and space complexities for the meta-heuristic algorithm will be computed.

Firefly optimization algorithm

The Firefly Optimization (FFO) algorithm has been developed by Xin-She Yang in 2008. It is based on the social behavior of fireflies and inspired from their flashing property [32]. The author has simplified their bioluminescent communication as the subsequent rules:

(1) Attraction of a firefly to another firefly has nothing to do with its sex and all fireflies are considered unisex.

(2) Attraction is proportional to the brightness of fireflies in a way that a less-brighter firefly can be absorbed to a brighter firefly and moves toward it. If a firefly cannot find a firefly brighter than itself, it moves in an optional way.

(3) An objective function specifies brightness or light intensity of a firefly.

When two fireflies have distance r from each other, their attractiveness parameter (β) is defined as [39, 40]:

$$\beta = \beta_0 e^{-\gamma . r^2} \tag{9}$$

where both β_0 and γ parameters are predestined parameters of the algorithm. Parameters β_0 and γ denote attractiveness at $r = 0$ and absorption coefficient, respectively. When firefly i is attracted to brighter firefly j, its amount of movement is computed by [39, 40]:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma . r^2}\left(x_j^t - x_i^t\right) + \alpha_t \varepsilon_i^t \tag{10}$$

where $\alpha_t \in [0, 1]$ is the randomization parameter and ε_i^t is the vector of random variables calculated from either a Gaussian or uniform distribution at time t. If $\beta_0 = 0$, movement becomes a simple random move.

Energy and temperature-aware Algorithms

In the proposed work, servers are considered based on three categories introduced in [19] as high, medium and low performance servers (see Section 1). As mentioned in Section 1, an idle server consumes high power proportional to the server running at full utilization. The work in [7] expresses that power consumption of an idle server is close to 70% of power consumption of a full running server. Therefore, switching an idle server to sleep mode can save energy efficiently. This fact results in consolidating of VMs in fewer servers with full utilization. Moreover, selecting servers from high performance ones can significantly benefit for optimal consolidation problem, because they are the main consumers of the datacenter. Therefore, consolidating of VMs on high performance servers can reduce energy consumption and heat.

The VM consolidation problem is complex and is divided into four sub-problems, server overload detection, server underload detection, virtual machine selection, and virtual machine placement. In the following, four steps of the VM consolidation problem are detailed.

Over-load server detection

In this section, the first part of dynamic VM consolidation problem (i.e., whether a server is overloaded or not) will be proposed (see Algorithm 1). When the load on a server exceeds its capacity, it is considered to be overloaded, resulting in fault of services in the server. Therefore, this situation must be prevented before occurrence. Some works [16, 41, 42] have defined an upper threshold for finding an overload server, where this threshold could be either static or dynamic. Since cloud computing environment is completely dynamic, it is best to consider dynamic upper threshold. First, among all servers in a data center, the sever with the highest maximum

Operation Per Second (OPS) per watt is chosen, called as optimum server. As described before (see Section 3.2), 10% of CPU utilization is required for completing VM live migration. Thus, 90% of CPU utilization could be considered as an upper-bound threshold and its temperature is determined at 90% CPU utilization based on ("https://www.spec.org/power_ssj2008/results/"). This temperature is called $Threshold_{temperature}$. In [18], an overloaded server is selected only based on temperature and other resources are not considered.

In the proposed work, all three parameters of CPU utilization, memory utilization and temperature of server are considered as the main parameters that affect energy consumption. However, there is an exception in some cases about temperature and resources utilization. For example, the temperature of a server may be higher than $Threshold_{temperature}$ in some optimal utilizations ("https://www.spec.org/power_ssj2008/results/,"), where the CPU and memory utilization could be low. This situation leads to additional VM migrations causing an increase in CPU overhead and energy consumption. For this reason, overloading has three conditions: temperature of the server is higher than $Threshold_{temperature,}$ CPU utilization is higher than 50%, and memory utilization is higher than 50%. The complexity of Algorithm 1 is $O(m + m) = O(m)$, where m is the number of servers.

Underload server detection

The second step in the VM consolidation problem is to select the server that its utilization of resources is very low. By migrating all the VMs running on the server and switching it to the sleep mode, energy consumption can be reduced. For this sub-problem, Algorithm 2 is proposed in which for the low performance servers, the server with CPU utilization fewer than 50% and memory utilization fewer than 20% is selected. For the medium performance servers, 30% and 10% lower bound thresholds are considered, respectively, for CPU and memory utilization for choosing an underload server. Finally, for the high performance servers, the server with 10% and 5% lower bound thresholds, respectively, for CPU and memory utilization is selected.

At the end, the algorithm tries to first select the underloaded server from the low performance, then from the medium performance, and finally from the high performance servers if there is no choice in the other two categories. The objective of this algorithm is to cause to put to sleep low and medium performance servers as far as possible because they consume more energy in the comparison with high performance servers. This algorithm has time complexity equal to $O(m)$, where m is number of servers.

Virtual machine selection

When an underloaded server is selected, all VMs running on it should be migrated to another suitable server. However, when an overloaded server is selected, only one VM

Algorithm 1: Overload Server Detection

```
Input: $ServerList$
Output: $OverloadList$
//$OPS$ stands for Performance Over Energy
$OPS_{maximum}=0$
$Overload\ List = \{\}$
For each $Server$ in $ServerList$
    $OPS = P/E$          //compute performance over energy
    If $OPS > OPS_{maximum}$ Then
        $OPS_{maximum} = OPS$
        $OptimumServer = Server$
    End If
End For
$Threshold_{temperature}$ = temperature of optimum server at 90% utilization
For each $Server$ in $ServerList$
    $Threshold$ = Get Temperature of $Server$
    If ($Threshold > Threshold_{temperature}$ and Utilization of CPU $\geq$ 50% and Utilization of RAM $\geq$ 50%) then
        Add $Server$ to $Overload\ List$
    End If
End For
Return $OverloadList$
```

on it must be migrated. After this migration, if the server is still overloaded, then another VM is chosen for migration. This process is repeated until the server is no longer overloaded. The Maximum Fit (MF) algorithm [19] for VM selection (stated in Section 1) is chosen.

Virtual machine placement

The last step in VM consolidation is to select a suitable server for the selected VM in the migration list that consists of all VMs in the queue of migration. Since the VM consolidation problem is an NP-Hard problem, finding an exact solution requires a lifetime. Therefore, heuristic and meta-heuristic methods should be used to solve it. Two new methods are introduced in this paper as heuristic-based and meta-heuristic based.

Heuristic energy and temperature-aware virtual machine consolidation (HET-VC)

In this section, our first heuristic algorithm is introduced that can find near optimal solution in a tolerable time. It has an objective function as Eq.(11):

$$F = \partial \times U_{CPU} + \beta \times U_{RAM} + \kappa \times Temp \qquad (11)$$

where ∂, β and κ are coefficients and their summation is 1.0. Parameter U_{CPU} is the utilization of CPU, U_{RAM} is the utilization of RAM, and *Temp* is the normalized temperature of the server that is between 0 and 1 based Eq.(12):

$$Temp = \left|\frac{X-a}{b-a}\right| \qquad (12)$$

where *Temp* is positive normalized temperature, *X*

Algorithm 2: Underload server detection

```
Input: $ServerList$
Output: $UnderloadServer$
Catrgorized Server Based on Type, $Utilization_{CPU}=0$, $Utilization_{RAM}=0$
$N$= Get number of Server Type
For $i$=1 to $m$      // m is the number of servers
  $k$=Type of Server
  If $k$ =1 Then   //Low performance Servers
      $Utilization_{CPU}$ = Get Server Utilization of CPU
      $Utilization_{RAM}$=Get Server Utilization of RAM
      If ($Utilization_{CPU}$ <50% and $Utilization_{RAM}$ <20%)
          Underload Servers in Type $k$ = $Server$
      End If
  End If
  If $k$ =2 Then   //medium performance Servers
      $Utilization_{CPU}$ = Get Server Utilization of CPU
      $Utilization_{RAM}$ = Get Server Utilization of RAM
      If ($Utilization_{CPU}$ <30% and $Utilization_{RAM}$ <10%)
         Underload Servers in Type $k$ = $Server$
      End If
  End If
  If $k$=3 Then //high performance Servers
      $Utilization_{CPU}$ = Get Server Utilization of CPU
      $Utilization_{RAM}$ = Get Server Utilization of RAM
      If ($Utilization_{CPU}$ <10% and $Utilization_{RAM}$ <5%)
         Underload Servers in Type $k$ = $Server$
      End If
  End If
End For
For $i$=1 to $m$   //search all machines
      If there is underload Server in Type $N$ Then
         $UnderloadServer$ = Underload Server in Type $N$
         Exit loop
      End if
End For
Return $UnderloadServer$
```

is temperature of the given server, a is the minimum temperature of server among all temperatures experienced, b is the maximum temperature of server among all temperatures experienced both based on the information obtained from ("https://www.spec.org/power_ssj2008/results/").

The proposed Heuristic Energy and Temperature Aware Virtual Machine Consolidation (HET-VC) calculates the amount of objective function based on Eq. (11) for all servers that are not overloaded / underloaded in a data center (see Algorithm 3). Then, the server with the lowest F is chosen. Another point considered in this algorithm is server classification. The algorithm tries to select a high performance server first, then a medium performance, and finally a low performance server.

The complexity of HET-VC in Algorithm 3 is $O(mN + m) = O(mN)$, where N is the number of server classifications. In this paper, we have $N = 3$ (low, medium and high), and m is the number of checked servers. Therefore, the complexity is $O(3m + m) = O(m)$.

Firefly energy and temperature aware virtual machine consolidation (FET-VC)

A heuristic method has two main problems: trapping at local optimum points and early convergence to these points. On the other hand, a meta-heuristic method can resolve these problems. The Firefly algorithm is a meta-heuristic method used in this section. First, it calculates the intensity of servers using Eq.(11). If there is a server with lower intensity than the server that VM is running on it now, FET-VC updates new position parameters based on the steps of the Firefly algorithm, where position parameters are utilization of resources and temperature. Therefore, the candidate server and its utilizations and temperature must be updated.

Finally, the algorithm selects a server with minimum F and utilization and temperature. In the Firefly algorithm, Eq.(10) is used for calculating parameters for movement to new position. Here, Eq. (13) is used for updating utilization and temperature of a server:

$$U_{new} = \mathrm{U}_{past} + F_{past}\left(U_{after} - \mathrm{U}_{past}\right) + \alpha \tag{13}$$

Algorithm 3: HET-VC

```
Input: $Server\ List$
Output: $SuitableServer$
Categorized Server Based on Type
$N$= Get Number of Server Type
For $i$= 1 to $N$
    $Minimum_N = 1$
    For each $Server$ in Type $N$
        $HeuristicValue$ = Calculate $HeuristicValue$ of $Server$ using $F$ function
        If $HeuristicValue < Minimum_N$ Then
            $LowloadServerinTypeN$ = Server
            $Minimum_N = HeuristicValue$
        End if
    End For
End For
For $N$=3 Downto 1
    If there is Server in $LowloadServerinTypeN$ Then
        $SuitableServer$= $LowloadServerinTypeN$
        Exit for Loop
    End if
End For
Return $SuitableServer$
```

where U_{new} is the new utilization of each resource (CPU and RAM), and U_{past} is past utilization of resource mapped to the first term in Eq.(10) as the last position of firefly. Parameter F_{past} is the intensity before allocating VM to PM that is set instead of the attractiveness parameter (i.e., β) in the firefly algorithm, U_{past} is the utilization of the resource before allocating VM, and U_{after} is utilization of the resource after allocating VM. They are set instead of each term in Eq.(10). Parameter α is a random number between 0 and 1 based the Gaussian distribution, set for the third term in Eq.(10). Generally, it is an error parameter because fireflies do not move directly in the real world. Hence, this parameter sets their movement to be closer to the real world. Note that utilization should always be less than 1. Besides, a new temperature is calculated as:

$$T_{new} = T_{past} + F_{past}\left(T_{after} - T_{past}\right) + \alpha \tag{14}$$

where T_{new} is temperature of the server after migration, T_{before} is the temperature before allocating VM, and T_{after} is an estimation of temperature after allocating VM. Algorithm 4 describes the FET-VC method step by step.

The complexity of FET-VC in Algorithm 4 is O($mN + m$) = O(mN), where N is the number of server classifications. Here $N = 3$ (low, medium and high), and m is the number of checked servers. Therefore, the complexity is O(3 $m + m$) = O(m).

Experimental setup

As mentioned in Section 3, the environment for this paper is IaaS, and it is required to be evaluated on a large scale virtualized infrastructure data center. Surely, the proposed algorithm evaluation is so difficult on real infrastructure. Simulator software (e.g., Cloudsim, SimGrid, GangSim and so on) are one way to prove the efficiency of the proposed work. The Cloudsim toolkit is chosen as our simulation platform because it can simulate cloud environments very well [43]. Besides, it can adapt with dynamic workloads. In this paper, the implemented extensions have been included in the 3.0.3 version of Cloudsim toolkit.

In the simulation process, one data center that comprises 300 heterogeneous physical machines is adapted, where 100 of them are HP Proliant ML 110 G5, 100 are HP Proliant DL360 G7 and the remaining are HP Proliant ML110 G9. The characteristics of the servers are given in Table 1. The number of VMs is varied based on data workloads. Ten days of them are

```
Algorithm 4: FET-VC
Input : Server List
Output : DestinationServer
$N$= Get number of Server type
$Intensity = 1$
For $i$=1 to $N$
    Intensity$_N$ = 1
    For each Server in Type $N$
        $MinF$= calculate intensity with $F$ function
        If($MinF$ < Intensity$_N$) Then
            Intensity$_N$ = $MinF$
            $SuitableServerinTypeN = Server$
            Update Loads and Temperature
        End If
    End For
End For
For $i$=$N$ Downto 1
    If $SuitableServerinTypeN$ is not NULL
        $DestinationServer = SuitableServerinTypeN$
        Exit for loop
    End If
End For
Return $DestinationServer$
```

Table 3 Characteristics of workload data

Date	Number of Virtual Machines
2011/03/03	1052
2011/03/06	898
2011/03/09	1061
2011/03/22	1516
2011/03/25	1078
2011/04/03	1463
2011/04/09	1358
2011/04/11	1233
2011/04/12	1054
2011/04/20	1033

chosen randomly for evaluation as shown in Table 3. The workload data are based on the CoMon project [44]. The interval of measurements is 5 min.

Performance metrics

To prove the efficiency, the proposed algorithms are compared with the DthMf algorithm. These comparisons are based on several metrics. First metric is total energy consumption by PMs expressed in Section 3.1. Second metric is the number of migrations and the number of VM replacements accomplished by VM manager. The third metric is SLA violation based on the model stated in Section 3.3 and the fourth metric is the ESV metric detailed in Section 3.4.

Simulation results

Here, HET-VC, FET-VC and DthMf algorithms are compared. Based on Eq.(11), there are three coefficients that sum of them must be one. Different combination of these coefficients are considered as combination 1: ($\alpha = 1/3$, $\beta = 1/3$, $\kappa = 1/3$), combination 2:($\alpha = 0.5$, $\beta = 0.25$, $\kappa = 0.25$) and combination 3:($\alpha = 0.25$, $\beta = 0.25$, $\kappa = 0.5$). In the following subsections, confidence intervlas (CI) of 95% are also provided for performance evaluatiosn parameters during 10 days of simulations.

Energy consumption

As shown in Fig. 1, HET-VC can decrease energy consumption, displayed in kilowatt hours (kWh) very well. The HET-VC under combination ($\alpha = 0.25$, $\beta = 0.25$, $\kappa = 0.5$) works better than the other two combinations of HET-VC algorithms. Energy consumption is affected from resource utilization and temperature. On the other hand, CPU utilization and temperature have a direct connection with each other. Therefore, by increasing temperature coefficient in combination ($\alpha = 0.25$, $\beta = 0.25$, $\kappa = 0.5$) and selecting servers with the lowest temperature, energy consumption can be decreased. The FET-VC has the best performance totally because of trying to find the best solution. The HET-VC tries to find the PM with the best F (see Eq.(11)) based on the current condition. However, FET-VC decides based on the current and future conditions in order to find the best solution. On average, FET-VC has the lowest energy consumption (7.65 kWh) with CI: (7.45,7.86) rather than three combinations of HET-VC (8.59 kWh with CI:(8.27, 8.92), 8.13 kWh with CI:(7.72,8.51) and 7.97 kWh with CI:(7.81,8.14)) and DthMf (14.89 kWh) with CI:(14.02, 15.75).

Number of migrations

Live migration is one of the best tools of virtualization for improving the consolidation problem. However, if the number of migrations is very high, energy

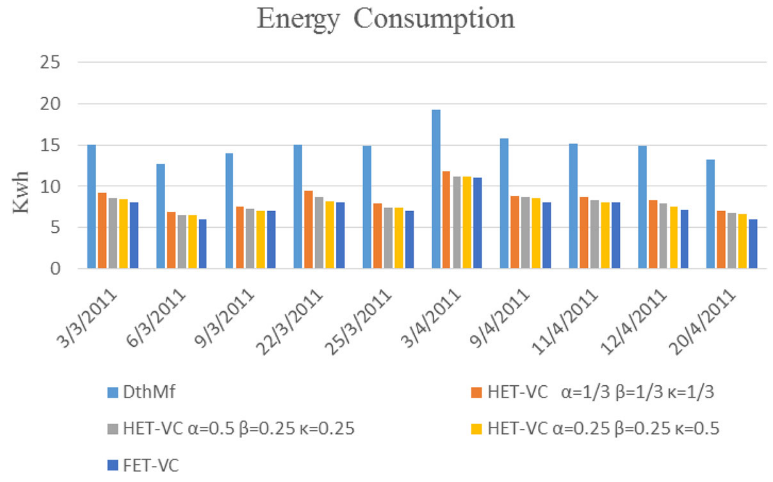

Fig. 1 Energy consumption

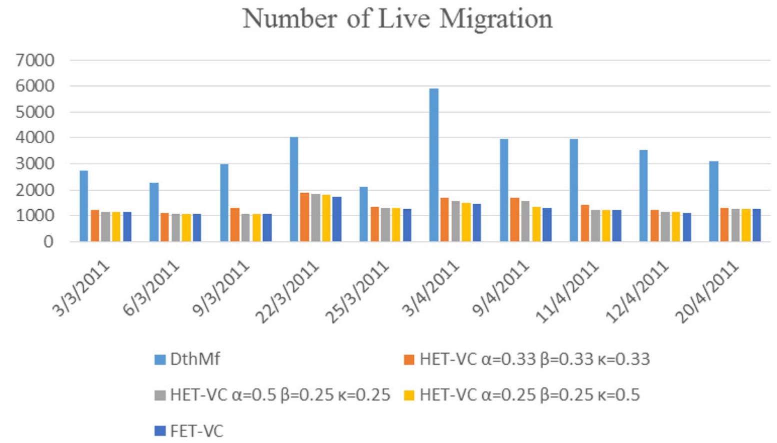

Fig. 2 Number of live migrations

consumption will increase. Therefore, an efficient solution must decrease energy consumption with the lowest number of live VM migrations.

Figure 2 depicts the number of VM migrations. The FET-VC and all of three coefficient combinations of HET-VC have lower migrations than DthMf. Besides, FET-VC provides the lowest number of migrations because of finding the best solution space that avoids from extra migrations. On average, FET-VC has 1259 migrations with CI:(1168,1350), combination 1 of HET-VC has 1412 migrations with CI:(1301, 1522)., combination 2 of HET-VC has 1315 migrations with CI:(1200, 1429), combination 3 of HET-VC has 1279 migrations with CI:(1181,1378), whereas DthMf has 3455 migrations with CI:(2973, 3936). Combination 3 of HET-VC has lower migration than two other combinations. This is because combination 3 selects the servers with the lowest temperature. Recall that a lower temperature server has a lower CPU utilization, and therefore, this server rarely enters in the overload status, thus reducing the number of migrations.

SLA violation

As described in Section 3.3, SLA violation is originated from SLATAH and PDM. Both HET-VC and FET-VC can reduce the number of VM migrations. Since migration has bad effect on performance and services, less migration decreases PDM and lower PDM leads to lower SLA violation. The FET-VC tries to find the best solution between all possible solutions and considers the current and the next positions, thus providing mostly lower number of migrations.

Therefore, its SLA violation is better than HET-VC. As shown in Fig. 3, FET-VC has near SLA violation value with combinations 2 and 3 of HET-VC, but FET-VC totally is better than the others. On average, the SLA

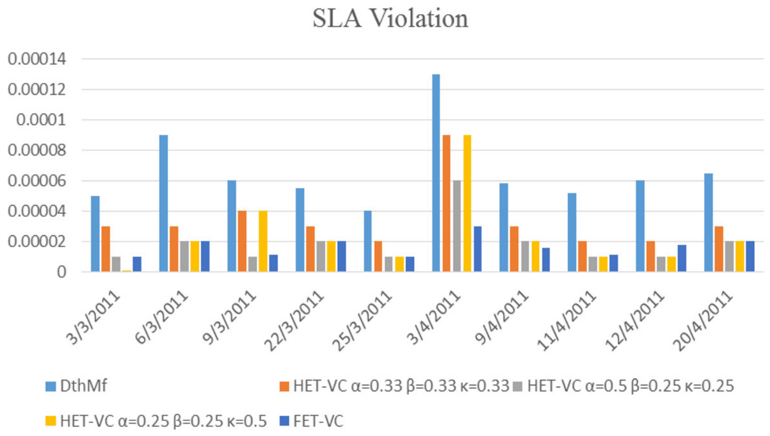

Fig. 3 SLA Violation

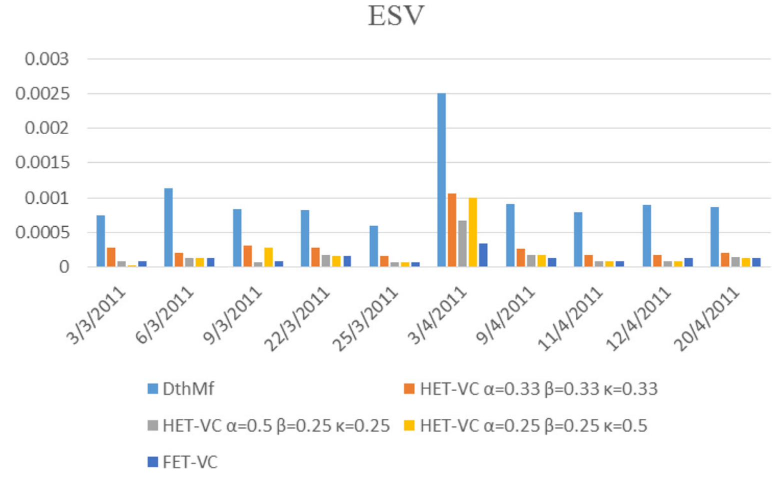

Fig. 4 ESV parameter

violation of FET-VC, combination 1 of HET-VC, combination 2 of HET-VC, combination 3 of HET-VC, and DthMf are 1.66×10^{-5} with CI:(1.38×10^{-5}, 1.93×10^{-5}), 3.4×10^{-5} with CI:(2.49×10^{-5}, 4.3×10^{-5}), 1.94×10^{-5} with CI:(1.43×10^{-5}, 2.46×10^{-5}), 2.43×10^{-5} with CI:(1.96×10^{-5}, 2.91×10^{-5}), and 6.6×10^{-5} with CI:(5.46×10^{-5}, 7.73×10^{-5}), respectively. In the previous parameters, combination 3 of HET-VC provided the best results, but here combination 2 provides the best result. This is because SLATAH and PDM that are the main parameters for calculating SLAV are based on CPU (see Eq.(7)) and in this combination CPU has more coefficient.

ESV

Recall ESV is based on energy consumption and SLAV. Hence, if one of them decreases, ESV goes down as a result. As described in Sections 6.3.1 and 6.3.3, both of the energy and SLAV parameters have been reduced by three combinations of HET-VC and FET-VC. Thus, ESV is reduced as a result as displayed in Fig. 4.

Figure 5 illustrates the threshold of three types of servers in different methods. As one can observe in the DthMf method, upper threshold is 90% for G7 and G9 and about 70% for G5 servers. The proposed methods have the same threshold because the overload and underload server detection is the same for both HET-VC and FET-VC but are different from DthMf. Upper threshold is close to 70% for G5 servers, 80% for G7 and 90% for G9. The proposed threshold is better than DthMf because both of HET-VC and FET-VC consider resources utilization and temperature to find servers with high temperature and utilization.

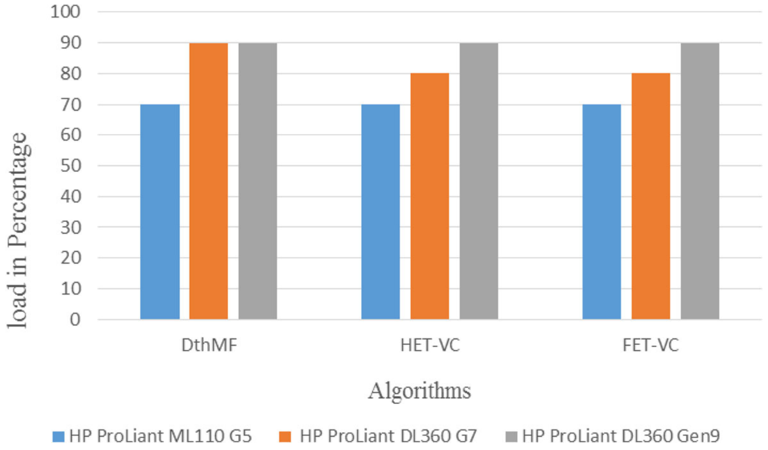

Fig. 5 Thresholds of servers in different load and different methods

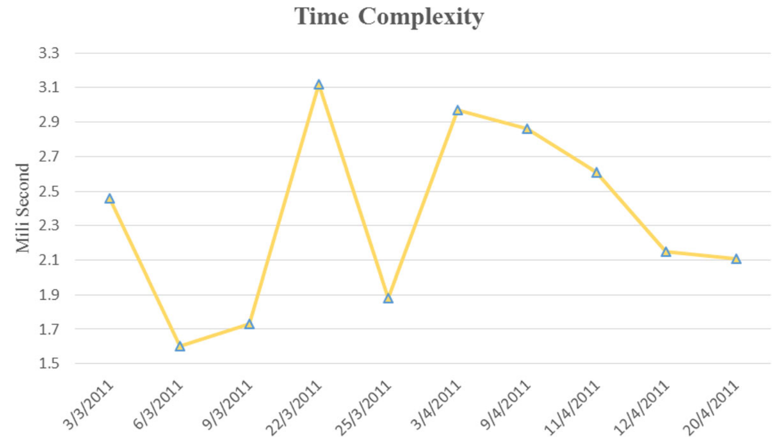

Fig. 6 Time Complexity

Space and time complexities

Two important parameters to be evaluated for any algorithm are time complexity and space complexity. Sometimes one algorithm has very good results but its time or space complexity is very high and unacceptable. Although, it is shown that FET-VC has very good performance results, it is still required to check time and space complexities. Therefore, these parameters for the FET-VC algorithm are measured. A personal computer with 8GB memory, Intel CPU core i5–3210 2.50 Ghz was used for the simulation of the proposed algorithms.

Figure 6 shows the diagram of time complexity parameter (in milliseconds) about simulation time of FET-VC. Time consumption of FET-VC was measured at the end of each time interval and at the end of simulation of each day, and calculate mean of all gained times. As one can see, the time complexity of the proposed method is very low. Hence, time complexity of FET-VC is acceptable.

Figure 7 depicts space complexity parameter in percentage (i.e., the ratio of used memory over total memory). The amount of memory usage of the PC for calculating this complexity was measured. Each point is gained based on the average memory usage within a day of simulation with respect to total memory. Measurements show that our space complexity is relatively low and it is a strong point for the FET-VC algorithm.

Conclusion and future works

In this paper, two novel algorithms as heuristic energy and temperature aware based virtual machine consolidation (HET-VC), and Firefly energy and temperature aware based virtual machine consolidation (FET-VC) were introduced to improve the consolidation problem of VMs. They can reduce energy consumption of servers while decreasing the number of migrations and SLA violation. On the other hand, a trade-off is obtained among the number of migrations and SLA violation.

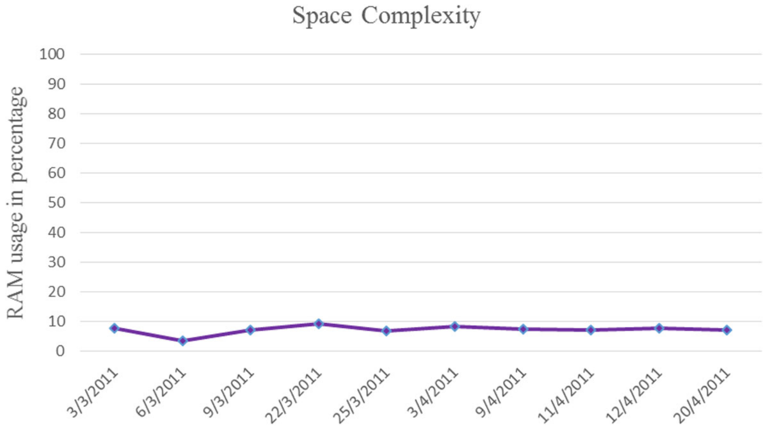

Fig. 7 Space Complexity

Based on performance evaluation results, both of the proposed algorithms outperform DthMf since DthMf considers only CPU as resource, but HET-VC and FET-VC consider CPU and RAM as resources. The DthMf tries to consolidate VMs on high performance servers with the lowest CPU utilization, but HET-VC and FET-VC consolidate VMs on high performance servers with the lowest CPU utilization, the lowest RAM utilization and the lowest temperature. The HET-VC and FET-VC find the best solution in the solution space. In addition, FET-VC considers current position and position after migration with updated distances.

Abbreviations

CI: Confidence Interval; DthMf: Dynamic Threshold Maximum Fit; DVFS: Dynamic Voltage and Frequency Scaling; ESV: Energy and SLA Violations; FET-VC: FireFly Energy and Temperature aware based VM Consolidation; FFO: Firefly Optimization; HET-VC: Heuristic Energy and Temperature aware based VM consolidation; IaaS: Infrastructure as a service; kWh: kilowatt hours; LIRCUP: Linear Regression based on CPU Usage Prediction; MBFD: Modified Best Fit Decreasing; MF: Maximum Fit; MIPS: Million Instruction Per Second; MPSO: Modified Particle Swarm Optimization; OPS: Operation Per Second; PaaS: Platform as a service; PDM: Performance Degradation due to Migration; PM: Physical Machine; QoS: Quality of service; SaaS: Software as a service; SAN: Storage Area Network; SLA: Service Level Agreement; SLATAH: SLA Violation Time per Active Host; VM: Virtual Machine

Acknowledgements

There is no acknowledgement.

Authors' contributions

Maede Yavari is the main researcher, Akbar Ghaffarpour Rahbar is thesis advisor, Mohammad Hadi Fathi is coauthor. All authors read and approved the final manuscript.

Authors' information

M. Yavari received her M.Sc. degree in Computer Network from Sahand University of Technology.
A.G.Rahbar is a full Professor at Sahand University of Technology. He received his Ph.D. degree from Ottawa University in Computer Science in 2006. He is a senior member of IEEE.
Mohammad H. Fathi received his M.Sc. degree in Computer Software from University of Tabriz.

Funding

There is no funding.

Availability of data and materials

The number of VMs is varied based on data workloads. Ten days of them are chosen randomly for evaluation as shown in Table 3. Our workload data are based on the CoMon project [42].

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Faculty of Electrical Engineering, Sahand University of Technology, Tabriz, Iran. [2]Electrical and Computer Eng. Department, University of Tabriz, Tabriz, Iran.

Received: 10 January 2019 Accepted: 30 July 2019
Published online: 27 August 2019

References

1. Mell P, Grance T (2011) The NIST definition of cloud computing. https://doi.org/10.6028/NIST.SP.800-145
2. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Futur Gener Comput Syst 25(6):599–616. https://doi.org/10.1016/j.future.2008.12.001
3. Khoshkholghi MA, Derahman MN, Abdullah A, Subramaniam S, Othman M (2017) Energy-efficient algorithms for dynamic virtual machine consolidation in cloud data centers. IEEE Access 5:10709–10722. https://doi.org/10.1109/ACCESS.2017.2711043
4. Liu Y, Sun X, Wei W, Jing W (2018) Enhancing energy-efficient and QoS dynamic virtual machine consolidation method in cloud environment. IEEE Access 6:31224–31235. https://doi.org/10.1109/ACCESS.2018.2835670
5. James, M., Forrest, W., & Kindler, N. (2008). Revolutionizing data center energy efficiency: McKinsey & Company, July
6. Asad Z, Chaudhry MAR (2017) A two-way street: green big data processing for a greener smart grid. IEEE Syst J 11(2):784–795. https://doi.org/10.1109/JSYST.2015.2498639
7. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Futur Gener Comput Syst 28(5):755–768
8. Greenberg S, Mills E, Tschudi B, Rumsey P, Myatt B (2006) Best practices for data centers: lessons learned from benchmarking 22 data centers. In: Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings in Asilomar, CA. ACEEE, pp 76–87
9. Heller B, Seetharaman S, Mahadevan P, Yiakoumis Y, Sharma P, Banerjee S, McKeown N (2010) Elastictree: saving energy in data center networks. Paper presented at the Nsdi
10. Greenberg A, Hamilton J, Maltz DA, Patel P (2008) The cost of a cloud: research problems in data center networks. ACM SIGCOMM Comput Commun Rev 39(1):68–73. https://doi.org/10.1145/1496091.1496103
11. Amoon M (2018) A multi criteria-based approach for virtual machines consolidation to save electrical power in cloud data centers. IEEE Access 6:24110–24117. https://doi.org/10.1109/ACCESS.2018.2830183
12. Barroso LA, Hölzle U (2007) The case for energy-proportional computing. Computer 40(12):33–37
13. Fan X, Weber W-D, Barroso LA (2007) Power provisioning for a warehouse-sized computer. Paper presented at the ACM SIGARCH computer architecture news
14. Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limpach C et al (2005) Live migration of virtual machines. In: Paper presented at the proceedings of the 2nd conference on symposium on networked systems Design & Implementation-Volume 2
15. Islam SS, Mollah MB, Huq MI, Ullah MA (2012) Cloud computing for future generation of computing technology. In: Paper presented at the Cyber Technology in Automation, control, and intelligent systems (CYBER), 2012 IEEE international conference on
16. Li H, Zhu G, Cui C, Tang H, Dou Y, He C (2016) Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing. Computing 98(3):303–317
17. Voorsluys W, Broberg J, Venugopal S, Buyya R (2009) Cost of virtual machine live migration in clouds: a performance evaluation. Paper presented at the IEEE International Conference on Cloud Computing
18. Cao Le Thanh M, Kayashima M (2011) Virtual machine placement algorithm for virtualized desktop infrastructure. 2011 IEEE International Conference on Cloud Computing and Intelligence Systems. IEEE.
19. Fard SYZ, Ahmadi MR, Adabi S (2017) A dynamic VM consolidation technique for QoS and energy consumption in cloud environment. J Supercomput 73(10):4347–4368
20. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr. Comput. Pract. Experience 24(13):1397–1420
21. Qiu M, Ming Z, Li J, Gai K, Zong Z (2015) Phase-change memory optimization for green cloud with genetic algorithm. IEEE Trans Comput 64(12):3528–3540. https://doi.org/10.1109/TC.2015.2409857
22. Younge AJ, Von Laszewski G, Wang L, Lopez-Alarcon S, Carithers W (2010) Efficient resource management for cloud computing environments. In: Paper presented at the green computing conference, 2010 international
23. Zhang J, Huang H, Wang X (2016) Resource provision algorithms in cloud computing: a survey. J Netw Comput Appl 64:23–42. https://doi.org/10.1016/j.jnca.2015.12.018
24. Wang F, Liu J, Chen M, Wang H (2016) Migration towards cloud-assisted live media streaming. IEEE/ACM Trans Netw 24(1):272–282

25. Yang C-T, Liu J-C, Huang K-L, Jiang F-C (2014) A method for managing green power of a virtual machine cluster in cloud. Futur Gener Comput Syst 37:26–36
26. Choi J (2019) Virtual machine placement algorithm for energy saving and reliability of servers in cloud data centers. J Netw Syst Manag 27(1):149–165. https://doi.org/10.1007/s10922-018-9462-3
27. Arroba P, Risco-Martín JL, Moya JM, Ayala JL (2018) Heuristics and metaheuristics for dynamic management of computing and cooling energy in cloud data centers. Softw Pract Exp 48(10):1775–1804. https://doi.org/10.1002/spe.2603
28. Verma A, Ahuja P, Neogi A (2008) pMapper: power and migration cost aware application placement in virtualized systems. In: Paper presented at the proceedings of the 9th ACM/IFIP/USENIX international conference on middleware
29. Verma A, Dasgupta G, Nayak TK, De P, Kothari R (2009) Server workload analysis for power minimization using consolidation. In: Paper presented at the proceedings of the 2009 conference on USENIX annual technical conference
30. Dorigo M, Gambardella LM (1997) Ant colonies for the travelling salesman problem. Biosystems 43(2):73–81. https://doi.org/10.1016/S0303-2647(97)01708-5
31. Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Retrieved from
32. Yang X-S (2010) Nature-inspired metaheuristic algorithms. Luniver press
33. Dressler F, Akan OB (2010) A survey on bio-inspired networking. Comput Netw 54(6):881–900. https://doi.org/10.1016/j.comnet.2009.10.024
34. Farahnakian F, Ashraf A, Pahikkala T, Liljeberg P, Plosila J, Porres I, Tenhunen H (2015) Using ant colony system to consolidate VMs for green cloud computing. IEEE Trans Serv Comput 8(2):187–198. https://doi.org/10.1109/TSC.2014.2382555
35. Kansal NJ, Chana I (2015) Artificial bee colony based energy-aware resource utilization technique for cloud computing. Concurr Comput Pract Exp 27(5): 1207–1225. https://doi.org/10.1002/cpe.3295
36. Kansal NJ, Chana I (2016) Energy-aware virtual machine migration for cloud computing-a firefly optimization approach. J Grid Comput 14(2):327–345. https://doi.org/10.1007/s10723-016-9364-0
37. Fathi MH, Khanli LM (2018) Consolidating VMs in Green Cloud Computing Using Harmony Search Algorithm. In: Proceedings of the 2018 International Conference on Internet and e-Business (ICIEB '18). ACM, New York, NY, USA, 146-151. https://doi.org/10.1145/3230348.3230369
38. Gandhi A, Harchol-Balter M, Das R, Lefurgy C (2009) Optimal power allocation in server farms. Paper presented at the ACM SIGMETRICS Performance Evaluation Review
39. Yang X-S (2008) Algorithms, N.-I. M. Luniver press, Beckington, pp 242–246
40. Yang X-S, He X (2013) Firefly algorithm: recent advances and applications. Int J Swarm Intell 1(1):36–50
41. Wolke A, Bichler M, Setzer T (2016) Planning vs. dynamic control: resource allocation in corporate clouds. IEEE Trans Cloud Comput 4(3):322–335
42. Xiao Z, Song W, Chen Q (2013) Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Trans Parallel Distrib Syst 24(6):1107–1117. https://doi.org/10.1109/TPDS.2012.283
43. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw Pract Exp 41(1):23–50
44. Park K, Pai VS (2006) CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Oper Sys Rev 40(1):65–74. https://doi.org/10.1145/1113361.1113374

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.