Computing and Informatics, Vol. 37, 2018, 781–814, doi: 10.4149/cai_2018_4_781

EQUAL: ENERGY AND QOS AWARE RESOURCE ALLOCATION APPROACH FOR CLOUDS

Ashok Kumar, Rajesh Kumar, Anju Sharma

Department of Computer Science and Engineering
Thapar University
Patiala-147004, India
e-mail: ashok.khunger@gmail.com, {rakumar, anju.sharma}@thapar.edu

Abstract. The popularity of cloud computing is increasing by leaps and bounds. To cope with resource demands of increasing number of cloud users, the cloud market players establish large sized data centers. The huge energy consumption by the data centers and liability of fulfilling Quality of Service (QoS) requirements of the end users have made resource allocation a challenging task. In this paper, energy and QoS aware resource allocation approach which employs Antlion optimization for allocation of resources to virtual machines (VMs) is proposed. It can operate in three modes, namely power aware, performance aware, and balanced mode. The proposed approach enhances energy efficiency of the cloud infrastructure by improving the utilization of resources while fulfilling QoS requirements of the end users. The proposed approach is implemented in CloudSim. The simulation results have shown improvement in QoS and energy efficiency of the cloud.

Keywords: Energy efficiency, resource utilization, resource allocation, antlion optimization, quality of service

1 INTRODUCTION

Cloud computing delivers Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [1] on pay per usage basis. These services are provided through shared pool of configurable computing resources such as networks, servers, storage, and applications, which are rapidly provisioned and released on demand. The liability of resource management lies with the service provider. This effortless computing paradigm (from the point of view of cloud users) has resulted

in dramatic increase in number of cloud users. To cope with resource demands of increasing number of users, cloud market players such as Amazon, Microsoft, Google, Gogrid, Flexiant, etc., establish large sized data centers. Due to this cloud computing infrastructure grows approximately 36 % every year, and is estimated to touch $ 19.5 billion market by 2016 [2]. Further, data centers consume huge amount of energy. It has increased threefold between 2007 and 2012 [3]. In 2013, data centers in U.S consumed an estimated 91 billion kWh of electricity, which was enough to power entire New York for two years [5]. Moreover, average resource utilization of data centers is approximately 15–20 % [4, 6]. The most part of the energy consumption of a data center is wasted due to under utilization of resources because even an idle resource consumes 50 % of its maximum power utilization [7]. This means low utilization decreases the energy efficiency of the resources.

On the other hand, performance may degrade if the VMs executing the tasks are not allocated resources as per requirements [8], and hence may cause Service Level Agreement (SLA) violations. Consequences of performance degradation can be critical. One direct consequence may be losing users. For example, 100 ms delay results 1 % decrease in sales of Amazon, and Google observed 20 % decrease in traffic with 0.5 seconds delay in search page generation [9].

The huge amount of energy consumption and liability to fulfill QoS requirements demand for efficient allocation of resources. Therefore, a novel Energy and QoS aware resource allocation approach Using AntLion optimization (EQUAL) is proposed. EQUAL allocates proportion of computing capability of a resource to a VM. It can be managed to operate in power, performance, or balanced mode. Further, the VMs encapsulate time constrained users' tasks which are distributed among them in a round robin fashion. The major contributions of the proposed resource allocation approach are:

1. A novel energy and QoS aware resource allocation approach is proposed.
2. Antlion optimization is employed to group VMs on lesser number of physical resources in order to optimize energy consumption.
3. The proposed approach can be tuned to power aware, performance aware, or balanced mode.
4. EQUAL is implemented in CloudSim and tested with VMs/tasks having different processing requirements.
5. Up to 15.04 % reduction in energy consumption is achieved.

The rest of the paper is organized as follows: Related work is presented in Section 2. In Section 3, the proposed resource allocation approach, power model, problem definition, and antlion optimization are presented. Resource provisioning using antlion optimization is elaborated in Section 4. Performance evaluation and comparative analysis is given in Section 5. Section 6 discusses conclusion and future scope for expansion.

2 RELATED WORK

Number of researchers have done significant work on energy efficiency and QoS aware resource allocation. They considered various application domains such as high-performance scientific computing, multi-tier web applications, or workflow applications, and used variety of approaches to obtain better results. In this section, an extensive survey on various resource allocation related approaches and state of the art techniques is conducted.

2.1 Resource Allocation with Traditional Algorithms

This subsection discusses various resource allocation approaches existing in literature that are not based on nature inspired meta heuristics. Quan et al. [10] presented resource allocation framework that improves utilization and hence the energy efficiency of the cloud infrastructure. Lee et al. [11] proposed performance based resource allocation strategy for green cloud. Each physical machine in the data center is assigned a performance value based on its CPU processing speed, number of cores and memory capacity. The physical machines resources are provisioned in the order of their performance value. Quarati et al. [12] proposed resource allocation for hybrid cloud with the objective to maximize broker's revenue and user satisfaction. The requested service is allocated resources on either private or public cloud depending on reserved quota of private cloud resources. Further, the service is run on a physical machine (PM) having maximum availability of free resources. Resource allocation is modeled as bin packing problem in order to improve utilization of resources [13, 14]. The PMs are treated as bins and VMs are assumed the items to be packed in. Bobroff et al. [13] presented an approach which periodically runs an offline bin packing algorithm to calculate VMs to PMs mapping. The approach eliminates hot-spots and minimizes the number of PMs in use. Takeda and Takemura [15] proposed ranking of physical servers for consolidation and VM placement. Servers with higher priorities are considered more reliable than the servers with lower priority value. Higher priorities are assigned to newly installed servers. The VMs are consolidated on more reliable servers to conserve energy. Son et al. [16] introduced workload and location-aware resource allocation scheme (WLARA) with automated SLA negotiation mechanism but they have not considered energy efficiency. Chieu et al. [17] proposed an architecture for dynamic allocation of resources to workloads, based on threshold number of active sessions. The proposed work is capable of maintaining higher resource utilization, thus reducing infrastructure and management costs. Wu et al. [18] advocated SLA based provisioning technique which reduces resource cost and SLA violations. The management of customer requests, mapping them with resources is defined along with the supervision of different types of workloads by considering QoS such as execution time. Raycroft et al. [19] analyzed the effect of global virtual machine allocation policy on energy consumption. Kim et al. [20] proposed energy credit scheduler which allocates resources to a VM based on its energy credit. The resources allocated to VM are preempted when its

energy credit vanishes. Xu and Fortes [56] proposed multi-objective VM allocation algorithm. The authors have taken CPU, and memory parameters for VMs and have claimed reduction in power consumption, thermal dissipation costs, and resource wastage. Wu et al. [22] presented a technique to increase the utilization and efficiency of hardware equipment. Dynamic voltage frequency scaling is employed to decrease energy consumption for executing jobs without sacrificing its performance. The authors in [15, 23, 24, 25] offered approaches to conserve energy and maximize resource utilization without affecting the performance of the system. They used energy-conscious consolidation heuristics to improve utilization and conserve energy. Beloglazov et al. [26] proposed power efficient and QoS aware resource allocation heuristics. An algorithm for minimization of number of VM migrations is also proposed. Upper and lower threshold utilization levels are used to detect overloaded and underloaded machines. When the resource utilization of a particular server falls below the lower threshold value, all the VMs running on the machine are shifted to some other machine. If utilization of a machine is above upper threshold, one or more VMs are shifted to other machines to keep the utilization between the threshold values. They proposed algorithms for single core machines. Kusic et al. [27] proposed performance and power efficient resource-management approach based on look-ahead control method for virtualized heterogeneous environments. Prediction is employed for dynamic reallocation of resources. Gao et al. [28] presented a dynamic resource management approach for energy saving and service level agreements fulfillment. CPU speed is considered as the bottleneck of performance. Dynamic voltage/frequency scaling and server consolidation are used for energy saving.

2.2 Resource Allocation Based on Nature-Inspired Metaheuristics

Feller et al. [29] presented a multi-dimensional ant colony optimization based job consolidation algorithm. The algorithm uses resource utilization history to predict future resource demands and dynamically overbooks the resources. The authors tested the proposed algorithm on homogeneous PMs. Gao et al. [30] proposed multi-objective ant colony system algorithm for virtual machine placement that minimizes total resource wastage and power consumption. Ant colony optimization technique for assigning real-time tasks to heterogeneous processors is proposed by Chen et al. [31]. Local search technique is applied to improve energy efficiency of the feasible assignment solution generated by the proposed assignment algorithm. Huang et al. [32] presented genetic algorithm based adaptive sub-optimal resource management scheme to estimate number of VMs required to provide desired level of service. Kansal and Chana [33] suggested a model based on artificial bee colony to improve utilization of resources. The model supports energy efficient allocations of tasks to resources and minimizing execution time of applications. Chimakurthi and Madhu Kumar [34] offered ant colony based adaptive resource allocation framework for hosting applications with throughput and response time as QoS requirements. Further, it supports reduction in power consumption of data center resources. Hu et al. [35] expressed problematic issue of VM placement as a multi-objective opti-

mization problem. An improved ant colony system algorithm is offered for the data centers to reduce total resource wastage and energy consumption. Liu et al. [36] gives ant colony optimization based solution for VM placement on physical servers in order to decrease the number of active physical servers. Portaluri et al. [37] presented genetic algorithm based trade-off solutions between tasks completion time and system power consumption. The system allocates resources to independent tasks on homogeneous single-core resources. Xiong and Xu [38] presented a multiresource energy efficiency VM allocation model based on particle swarm optimization for energy efficiency of cloud data center. Total Euclidean distance is used as a fitness function to keep balance between resource utilization and energy consumption. This algorithm avoids falling into local optimal solution, which is common in traditional heuristic algorithms. Kumar and Raza [39] presented particle swarm optimization based strategy for VM allocation to physical machines in order to reduce total energy consumption and resource wastage. Dashti and Rahmani [40] proposed modified particle swarm optimization solution to guarantee quality of service of users' tasks, and reduce energy efficiency. Response time and deadline are considered as QoS parameters. This approach reallocates virtual machines from the overloaded host, and dynamically consolidates under-loaded hosts for power saving. Kansal and Chana [41] used firefly optimization to enhance energy efficiency of the cloud without sacrificing the performance. The authors used VM migration to enhance energy efficiency. Kumar et al. [42] presented two level ant colony based resource allocation approach to minimize total cost of execution, total execution time and total energy consumption. They used server consolidation and dynamic performance scaling to conserve energy. Kumar et al. [43] proposed power and performance aware resource allocation. They improved performance and energy efficiency of the cloud employing cuckoo optimization.

2.3 Motivation

Resource allocation in cloud computing can be accomplished using either traditional deterministic algorithms [11, 12, 13, 14, 15, 18] or metaheuristic algorithms [29, 30, 31, 32, 33, 41, 42, 43]. Deterministic algorithms suffer from local optima entrapment, i.e., they got struck in local solutions and consequently fail to find the true global optimal solution. Moreover, resource allocation using deterministic algorithms is NP-hard. So in the recent years metaheuristic algorithms have been employed for efficient allocation of resources [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43]. The fundamental characteristic of metaheuristic algorithms is stochastic operators which are used for finding optimal solution in the search space. Stochastic operators help them to escape local solutions. Due to their random behavior, they are able to obtain different solutions in each run. They start with some random solutions, called candidate solutions, of the problem at hand, and then improve the candidate solutions iteratively. Their solution finding process is completely independent from the problem. We get motivation from the way metaheuristic algorithm operates to find optimal solution of the problem. When a metaheuristic algorithm gets trapped in

local solution, stochastic operators make random changes in the solution and eventually help in escaping from local optimal solution. In a nutshell, all metaheuristic algorithms follow a general and common framework, in which they improve a set of randomly created solutions iteratively. The algorithms differ in the method of improving the initial random solutions. We preferred antlion optimization over other existing metaheuristic algorithms used [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43] for resource allocation because it provides very competitive results in terms of improved exploration, local optima avoidance, exploitation, and convergence [49]. Further, our work significantly differs from the other metaheuristic based resource allocation approaches in the area of cloud computing as it can be tuned to operate in power aware, performance aware, or balanced mode.

The detailed working of the proposed resource allocation approach is presented in the next section.

3 ENERGY AND QOS AWARE RESOURCE ALLOCATION

Resource allocation is a process of provisioning resources to VMs. Resources are allocated to VMs with the aim to minimize energy consumption while satisfying QoS requirements. In this work, we used antlion optimization for energy and QoS aware allocation of resources to VMs. The proposed approach can be operated in power, performance, and balanced mode. In power aware mode, a VM is allocated to a resource that causes minimum increase in energy consumption. Whereas in performance mode, a VM is allocated to a resource that has maximum available computational capacity. In balanced mode, power and performance are given equal weightage while allocating resources. The VMs encapsulate users' tasks which are scheduled in Earliest Deadline First (EDF) order. Each task has a deadline, a point of time, by which execution of the task should finish. If deadline of a task is missed then SLA violation is said to have occurred.

In the proposed work, the following assumptions are taken into consideration:

1. Each task is independent of other tasks.
2. A VM can be executed on a server with lesser free available resources than required but at the cost of reduced performance.
3. Resources can be switched to sleep mode to conserve energy.
4. Energy consumption of a resource in sleep mode is negligible.

The major components of energy and QoS aware resource allocation approach are shown in Figure 1. The *bag of tasks* is the collection of time constrained tasks submitted by the end users. The detailed information about each resource like a type of resource and its computational capability is provided by *resource description* component. The *resource allocation* component refers *resource description* component while allocating resources to VMs. The resources are allocated to VMs employing antlion optimization. Once the resources are provisioned to VMs, *resource scheduler* manages the scheduling of VMs on the provisioned resources. Utilization of each

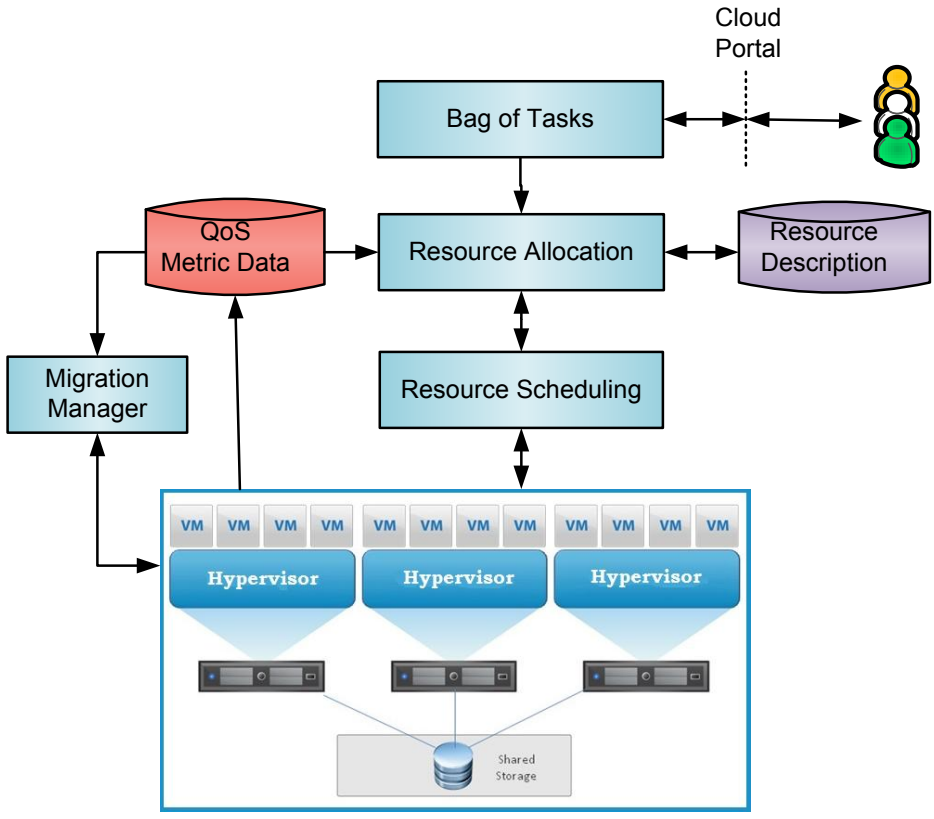

Figure 1. Energy and QoS aware resource allocation

resource (server) is monitored at regular interval of time, and saved in *QoS metric database*. Resource utilization information is used by *migration manager* to perform server consolidation. It is invoked after a fixed interval of time. A VM is selected for migration based on interquartile range (IQR) [44] of the utilization history data. VM migration is performed in two cases. First, when a resource is under-loaded, i.e., the utilization of the resource is below the lower green threshold (LGT) limit. In this case, all the VMs running on the under-loaded resource are shifted to other resources and the resulted idle resource is switched to low power (sleep) mode to conserve energy. Second, when the resource is over-loaded, i.e., the utilization of the resource is above the upper green threshold (UGT). In this case, one of the VMs running over the resource is migrated to other resource to bring the resource utilization below UGT.

3.1 Power Model

Generally, the resources have different run-time power consumption because of their heterogeneous processor architectures, processing speeds, hardware features, etc. Power consumption of a resource is given by Equation (1):

$$P_{total} = P_{dynamic} + P_{static}, \tag{1}$$

static power consumption (SPC), P_{static}, is due to leakage current and is independent of clock frequency and usage scenario. SPC can be reduced by switching idle resources to sleep mode [45]. However, dynamic power consumption (DPC), $P_{dynamic}$, is due to circuit activity, and it depends on resource utilization. DPC of a resource increases linearly with its utilization [26], and is given by Equation (2).

$$P = P_{idle} + (P_{max} - P_{idle})U \tag{2}$$

where P_{idle} is the power consumption when the resource is idle, P_{max} is the power consumption at 100 % utilization, and P is the power consumption of the resource at utilization $U \in [0, 1]$.

3.2 Problem Definition

Cloud computing leverages virtualization such as XEN [46], KVM [47], or VMWare [48] to support execution of multiple VMs on a single physical resource. Each VM has some resource demands such as CPU, number of processing cores, memory, network bandwidth, etc. If a VM is not allocated the required resource capacity then it processes encapsulated tasks at slower speed thereby elongating the tasks' completion time. Consequently, some of the tasks may miss the deadline. When deadline of a task is not observed, it is considered as SLA violation.

Suppose a set $J = \{J_i | 1 \leq i \leq n\}$ of n tasks, and each task J_i is associated with deadline time d_i and processing volume w_i. The processing volume is the amount of processing in millions of instructions (MI) that must be carried out to finish the task. Tasks are distributed among V number of VMs. Further, $S = \{S_j | 1 \leq j \leq m\}$ is a set of m resources. The problem is to allocate resources to VMs in such a way to minimize the number of active resources and their energy consumption while observing QoS requirements (deadline) of the end users' task. We considered only processing requirements while allocating resources because CPU is accounted for major part of energy consumption by a physical machine [45]. The allocation of resources to VMs is NP-hard. Therefore, antlion metaheuristic optimization is employed for allocation of resources to VMs. The proposed approach groups VMs over a small number of resources and thus allows turning off those resources that are not in use. Energy efficiency and QoS are considered while allocating resources to the VMs. The suitability of resource r for VM j is determined from fitness function $f_{j,r}$, given by Equation (3), which helps in fulfilling the following goals:

1. Allocation of a VM to a resource that results in minimum increase in energy consumption of the cloud.
2. Provisioning VMs on reduced number of resources.
3. Performance requirements are taken into consideration while allocating resources.

$$f_{j,r} = \frac{\left[\frac{\Re_r^a}{\Re_j^d}\right]^{\theta}}{\left[\gamma \triangle E_{j,r} + (1-\gamma)\kappa_r \underbrace{(1 - \sum_{i \in S, i \neq j} \Re_{i,r})}_{y}\right]^{1-\theta}}, \quad \forall r \tag{3}$$

where $\Re_r^a$ is available processing power of resource r, $\Re_j^d$ is processing demand of VM j. $\triangle E_{j,r}$ is energy contribution of VM j on resource r, κ_r is energy affinity, $\Re_{i,r}$ is fraction of processing power allocated to VM i on resource r, and $0 \leq \gamma \leq 1$ is a constant. $0 \leq \theta \leq 1$ is trade off between performance and energy. By changing the value of θ, EQUAL can be operated in one of the three modes, namely:

1. power aware,
2. performance aware, or
3. balanced mode.

EQUAL operates in power aware mode when θ is set to 0. In this mode, the increase in energy consumption of the resource is considered while allocating VMs. Thus, a VM is allocated to a resource which results in minimum increase in energy consumption. When θ is set to 1, EQUAL operates in performance aware mode, and thus allocates a VM to a resource which has maximum available computing power at disposal. In balanced mode, when θ is 0.5, EQUAL maintains the balance between power and performance while allocating resources to VMs. EQUAL inclines towards power aware allocation if $\theta < 0.5$, and towards performance aware allocation if $\theta > 0.5$.

In this work, $\frac{\Re_r^a}{\Re_j^d}$ is called performance affinity which is desired to be greater than or equal to 1. When performance affinity value is less than one, VM would not get sufficient resource and would therefore slow down the execution of encapsulated tasks. Energy affinity, κ_r, is the minimum energy consumption of the resource, i.e. energy consumption in idle state. Therefore, EQUAL gives preference to resources having lesser energy consumption in idle state. A resource having low power consumption in the idle state has higher value of fitness function and is therefore given preference over others while resource provisioning. The term y allows to group VMs on lesser number of resources. The value of term y for a resource r decreases as more and more VMs are deployed on it. Consequently, fitness function of the resource r increases and thereby enables grouping of VMs on lesser number of resources.

When θ is set to 1, power consumption of a resource does not contribute in finding suitable resource for the candidate VM. The machine having maximum available resource capacity is given preference over the others and the allocation approach reduces to worst-fit decreasing. In order to consider power affinity while searching for the best resource and to pack VMs on lesser number of machines, denominator of the Equation (3) should not evaluate to 1. This is possible only if θ is assigned value smaller than one. Thus, we used $\theta = 0.95$ to bias EQUAL towards performance aware allocation while taking advantage of its VM packing capability in order to save power consumption. However, when θ is set to 0, EQUAL reduces to best-fit approach which strives to pack VMs on lesser number of resources. The machine which is hosting more VMs and has low energy affinity is given preference over the others. The processing power of the resources is not taken into consideration. In order to consider computing capability of the resource in power aware mode θ should be assigned small positive value other than 0. In this work, we used $\theta = 0.05$ for power aware allocation in order to consider computing power of a resource in addition to its power consumption.

3.3 Application and Infrastructure Model

Cloud computing is suitable platform for deadline constrained scientific applications in areas such as astronomy, bioinformatics, and physics [57]. In this work, we proposed resource allocation approach, EQUAL, that can be used for deadline constrained applications such as Montage, which is used for generation of sky mosaics; Cyber-Shake, used for earthquake risk characterization; LIGO, used for detection of gravitational waves and SIPHT, used in bioinformatics. All these four applications are characterized by Juve et al. [58]. Scientific application (task) consists of thousands of sub-tasks, and can take benefit of large-scale infrastructure of cloud computing. Scientific application has soft deadline which is required to be accomplished. A soft-deadline does not make the computation useless if the task is not completed in time [59]. A computation begets maximum benefit if deadline is achieved. A scientific application may consist of sub-tasks and may have dependencies between them. Each task i has a deadline d_i and processing volume w_i associated with it. Deadline of a task determines the time to accomplish the execution of the task from the moment it is submitted to EQUAL, which manages the execution of tasks, allocates VMs to them, and schedule their execution in the cloud. EQUAL offers a set of four VM types denoted by set $V = \{A0, A1, A2, A3\}$. Each VM type offers different amount of resources. There is no limit on the number of VMs of each type that can be running at any moment for the execution of tasks. The problem addressed in this work is the execution of tasks latest by deadline time at the smaller possible energy cost. The problem is solved by the efficient allocation of resources using antlion optimization.

3.4 Antlion Optimization

Antlion optimization [49] is proposed by Seyedali Mirjalili in 2015. Antlions belong to the myrmeleontidae family. An antlion larvae makes cone-shaped pit and hides itself underneath the pit waiting for prey to be trapped in. The size of the pit is proportional to the level of hunger. When an insect is trapped in the pit, the antlion tries to catch it by intelligently throwing sand towards to edge of the pit to slide the prey into the bottom of the pit. Once the insect is caught, it is pulled under the sand and then consumed.

The following are the reasons for selecting antlion optimization for resource allocation:

1. Random selection of antlions guarantees the exploration of search space.
2. Adaptive shrinking boundaries of antlions' traps guarantee the exploitation of search space.
3. The promising regions of search space are guided by antlions.
4. It is a gradient-free algorithm and considers the problem as a black box.

4 RESOURCE ALLOCATION USING ANTLION OPTIMIZATION

The antlion optimization algorithm, which mimics behavior of antlions and ants, is used for discovering resource for a VM. The objective is to find a resource that fulfills not only the resource requirements of the VM but also causes a minimum increase in energy consumption. Each antlion provides initial guess of the resource, and then a resource better than the initial guess is searched through random walk of an ant around the antlion. When a better resource is found, the location of the antlion is replaced with the location of the corresponding ant.

The location of ant and antlion, each representing a resource, are saved in matrix M^a and M^{al}, respectively. Location of i^{th} ant, W_i^t, and j^{th} antlion, V_j^t, at t^{th} iteration are represented by i^{th} and j^{th} rows of matrices M^a and M^{al}, respectively. In each iteration, location of an ant is updated to reflect its latest position. The fitness value of an ant, which determines goodness of a solution, is also updated in each iteration. The fitness value of an ant/antlion is evaluated from Equation (3). When the fitness value of an ant becomes greater than the fitness value of the antlion, the location and fitness value of the antlion are replaced with the location and fitness value of the corresponding ant. The fitness values of ants and antlions are saved in matrix M^{fa} and M^{fal}, respectively.

Random walk of an ant i in the search space is modeled by Levy Flight (LF) [50], which can be expressed by Equation (4).

$$W_i^t = W_i^{t-1} + \alpha \; L(s, \lambda) \tag{4}$$

where α is the scaling factor for step size s. Levy exponent, λ, is a constant. $L(s, \lambda)$ is Levy distribution with parameters s and λ. W_i^t is the location of an ant i at t^{th} iteration.

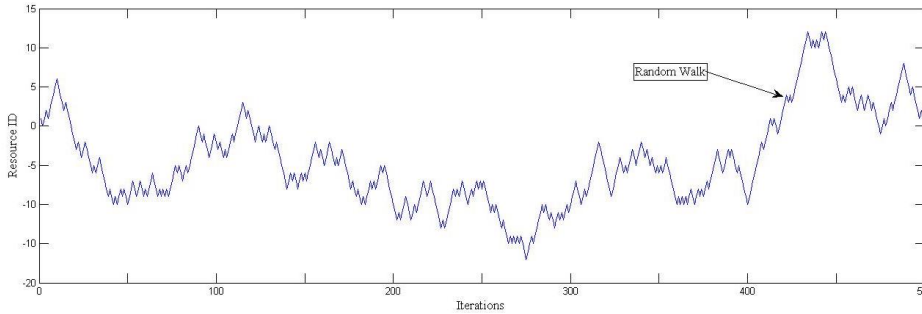

Figure 2. Random walk of an ant

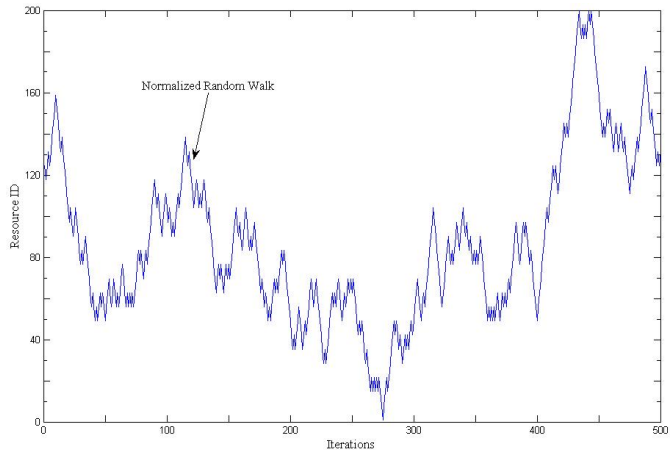

Figure 3. Normalized random walk of an ant

Figure 2 shows random walk for an ant generated from Equation (4). Number of iterations is represented along x-axis, whereas, resource ID (identification) is denoted by y-axis. Since search space, consisting of identification of each resource, has a range of permitted values, so max–min normalization, as shown in Equation (5), is applied to keep the random walk within the desired range.

$$W_i^t = \left\lceil \frac{(W_i^t - a_i)(d_i^t - c_i^t)}{b_i - a_i} + c_i^t \right\rceil \tag{5}$$

where a_i, b_i are the minimum and maximum of random walk of i^{th} ant, and c_i^t, d_i^t are the minimum and maximum of search space at t^{th} iteration. Figure 3 shows normalized random walk of the ant generated using Equation (5). Random walk of an ant (shown in Figure 2) is normalized to range of resource identifications used in EQUAL, i.e. from 1 to 200.

Random walks of ants are affected by positions of antlions. An ant is allowed to move around an antlion which is selected using roulette wheel. The range of search space at t^{th} iteration for random walk of an ant i around the antlion j is mathematically modeled by Equations (6) and (7).

$$c_i^t = V_j^t + c^t, \tag{6}$$

$$d_i^t = V_j^t + d^t \tag{7}$$

where c^t and d^t are the minimum and maximum of the search space at t^{th} iteration, c_i^t and d_i^t are the minimum and maximum for i^{th} ant, and V_j^t is the position of the selected j^{th} antlion at t^{th} iteration. In order to find the best resource, values of c^t and d^t are updated in each iteration using Equations (8) and (9), respectively.

$$c^t = \left\lceil \frac{c^t}{I} \right\rceil, \tag{8}$$

$$d^t = \left\lceil \frac{d^t}{I} \right\rceil, \tag{9}$$

here, $I = 10^w \frac{t}{T}$, where t is the current iteration, T is the maximum number of iterations, and w is a constant that can adjust the level of exploitation and is defined on the basis of the current iteration.

As discussed before, the location of an antlion is replaced with the location of corresponding ant when the fitness of a resource referred by an ant becomes greater than the fitness of a resource referred by the antlion. This situation is represented by Equation (10).

$$V_j^t = W_i^t, \text{if } f(W_i^t) > f(V_j^t) \tag{10}$$

where V_j^t is location of j^{th} antlion at t^{th} iteration, and W_i^t is location of i^{th} ant at t^{th} iteration.

EQUAL maintains record of the best resource (solution). The best solution, called elite, is saved in each iteration. The elite solution has the highest fitness value and effects the random walk of each ant (as shown in Equation (11)).

$$W_i^t = \frac{W_i^t + W_e^t}{2} \tag{11}$$

where W_i^t is random walk of ant i around an anlion and W_e^t is random walk of elite e at t^{th} iteration.

The detailed resource provisioning process is given in Algorithm 1. It employs antlion optimization to find the best resource for a VM. The resource search process is repeated until maximum iterations T has elapsed or the elite solution is same for three consecutive iterations.

Algorithm 1 Pseudo code for energy and QoS aware resource allocation using Antlion Optimization

Input: Set V of VMs; Set A of ants; Set L of antlions; Set S of resources
Output: VMs-Resources map (M_{VR})
for each VM $v \in V$ **do**
  Initialize ants' position matrix M^a randomly.
  Initialize antlions' position matrix M^{al} randomly.
  Evaluate suitability of resource referred by each ant $i \in A$ ($f_{v,i}$) and antlion $j \in L$ ($f_{v,j}$) for VM v from fitness function (Equation (3)) and store the values at i^{th} and j^{th} row of matrix M^{fa} and M^{fal}, respectively.
  Find an antlion (say e) for which value of fitness function (Equation (3)) is maximum (say $f_{v,e}$) and call it elite solution.
  set iteration counter $t \leftarrow 1$
  while ($t \leq$ T) **and** (until e is same for three consecutive iterations) **do**
    for each ant $i \in A$ **do**
      Select an antlion j using Roulette wheel.
      Calculate c^t and d^t using Equations (8) and (9).
      Evaluate c_i^t and d_i^t using Equations (6) and (7) to select range of random walk for ant i
      Generate random walk for ant i using Equation (4)
      Normalize random walk for ant i using Equation (5)
      Update random walk on ant i using Equation (11).
      Update position vector (M_i^a) and fitness value (M_i^{fa}) of ant i.
      if ($f_{v,i} > f_{v,j}$) **then**
        set $M_j^{fal} = M_i^{fa}$
        set $M_j^{al} = M_i^a$
      end if
    end for
    Find new elite solution among antlions and assign it to e.
    set $t \leftarrow t + 1$
  end while
  Allocate VM v to resource referred by elite solution e, and add VM-resource pair to map M_{VR}
end for
return M_{VR}

5 PERFORMANCE EVALUATION AND COMPARATIVE ANALYSIS

Type	Processing	PEs	RAM	Storage	BW
1	2 933	4	8	500	10
2	3 067	4	8	500	10
3	2 933	12	12	500	10
3	3 067	12	16	500	10

Processing, processing speed in millions of instructions per second; PEs, number of processing elements; RAM, random access memory in GB; Storage, permanent storage capacity in GB; BW, network bandwidth in gigabits per second.

Table 1. Specification of resources

VM Type	CPU	PEs	RAM	BW
A0	500	1	768	1 000
A1	1 000	1	1 792	1 000
A2	1 500	2	3 584	1 000
A3	2 000	4	7 168	1 000

CPU, processing speed in millions of instructions per second; PEs, number of cores; RAM, random access memory in megabytes; BW, network bandwidth in megabits per second.

Table 2. Specification of virtual machines

A number of cloud simulation tools such as CloudSim [51], CloudAnalyst [52], GreenCloud [53], NetworkCloudSim [54], etc., are available to implement and evaluate a resource allocation approach on large scale, repeatable, and controlled cloud environment. But the proposed approach is implemented in CloudSim because it supports modeling of various cloud entities such as datacenters, servers, virtual machines, and tasks with ease. The proposed resource allocation approach can be implemented easily by extending VM allocation policy of CloudSim.

For performance analysis, EQUAL is compared with Artificial Bee Colony (ABC) [33], Genetic Algorithm (GA) [56], and non-QoS aware resource allocation (NQRA) which is designed by combining round robin and earliest deadline first scheduling approach that allocates resources using best effort approach.

5.1 Experimental Setup

The simulation testbed consists of a datacenter containing 200 resources. The specification of four types of resources used in simulation is as per Table 1. We created equal number of resources of each type in a simulation run. The datacenter models

instances of general purpose compute-basic tier of Microsoft Azure [55], and the parameters relevant for the experiments are shown in Table 2. The tasks having diverse CPU and memory requirements are used, and the number of tasks are varied from 200 to 1 000. Further, the tasks are modeled as Cloudlets and their processing requirements are represented in MI. Simulation is repeated forty to fifty times with different number of resources, VMs, and tasks. The different parameters used during simulation are shown in Table 3.

Number of Resources	50–200	
Number of Tasks (Cloudlets)	200–1 000	Varied in every simulation run
Size of tasks	10 000 + (5–30 %) MI	in millions of instructions (MI)
Simulation Span	86 400 s	Simulation time period
Idle Time	10 min.	Time to switch PM to sleep mode
UGT	0.85	Upper Green Threshold limit
LGT	0.20	Lower Green Threshold limit
HT	0.95	Hot-spot Threshold
CT	0.15	Cold-spot Threshold

Table 3. Simulation parameters

5.2 Simulation Results

Case 1: EQUAL in Balanced Mode

EQUAL switches to balanced mode when 0.5 is assigned to θ. In this mode, energy and performance are given equal weightage while allocating resources to VMs. In order to group VMs over minimum number of resources 0.05 is assigned to γ.

Figure 4 shows the comparison of number of resources used by EQUAL in balanced mode (EQUAL-B), NQRA, ABC, and GA for different number of VMs. The number of used resources increases with increase in number of VMs to be deployed. But EQUAL-B uses lesser number of resources than NQRA, ABC, and GA for given number of VMs. It has been observed that EQUAL-B uses approximately 8.68 %, 4.47 %, and 6.84 % lesser number of resources than NQRA, ABC, and GA, respectively.

Figure 5 depicts the comparison of total energy consumption of EQUAL-B, NQRA, ABC, and GA. It is observed that EQUAL-B consumes lesser energy than NQRA, ABC, and GA for given number of VMs. In EQUAL-B, energy consumption of 107.67 kWh is measured for 200 VMs, and it increases to 735.65 kWh for 1 000 VMs. It is observed from simulation results that EQUAL-B consumes 10.8 %, 5.44 %, and 7.69 % lesser amount of energy than NQRA, ABC, and GA, respectively.

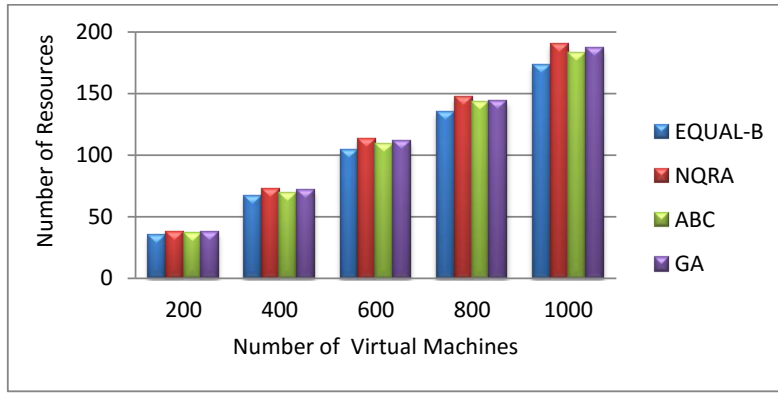

Figure 4. Comparison of the number of resources required

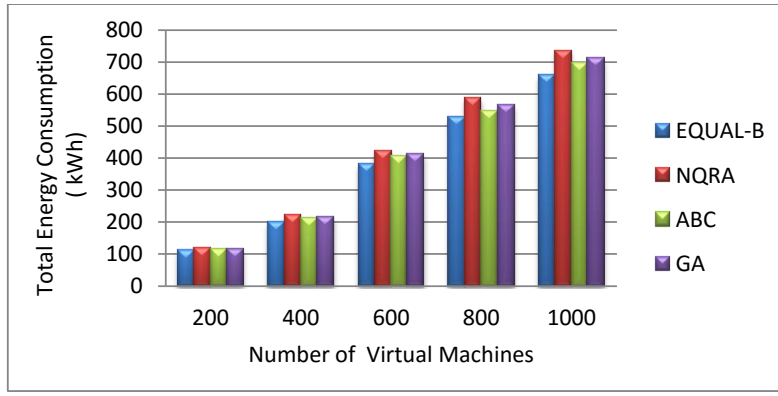

Figure 5. Comparison of total energy consumption

Figure 6 narrates comparison of average number of VM migrations performed in EQUAL-B, NQRA, ABC, and GA for different number of VMs. A VM is selected for migration using IQR. Resources becoming idle because of consolidation are switched to low power (sleep) mode to conserve energy. The number of VM migrations increases with the increase in number of VMs. In EQUAL-B 16.75 %, 10.97 %, and 16.65 % lesser number of VM migrations is observed than in NQRA, ABC, and GA, respectively.

Figure 7 describes comparison of number of hot-spots created in EQUAL-B, NQRA, ABC, and GA. The number of VMs is varied from 200 VMs to 1 000 VMs with increment of 200 VMs. A resource is considered as a hot-spot if its utilization is above HT. A hot-spot adversely affects the reliability and performance

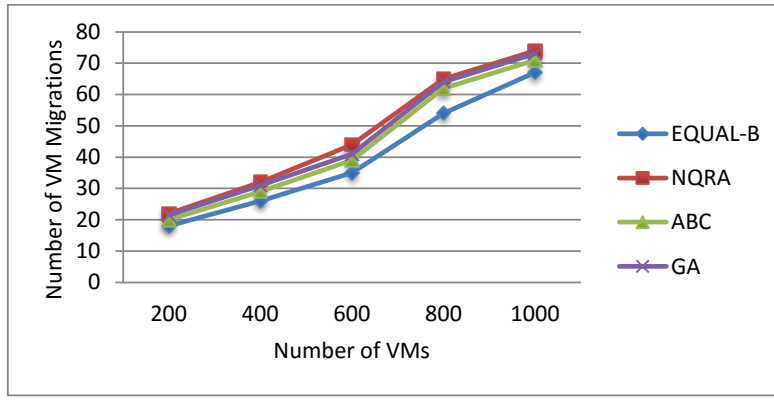

Figure 6. Comparison of the number of VM migrations

of the resource. Moreover, a hot-spot also demands better cooling arrangements. EQUAL-B improves reliability and energy efficiency of the resource as it creates 8.33 %, 18.20 %, and 14.22 % lesser number of hot-spots than NQRA, ABC, and GA, respectively.

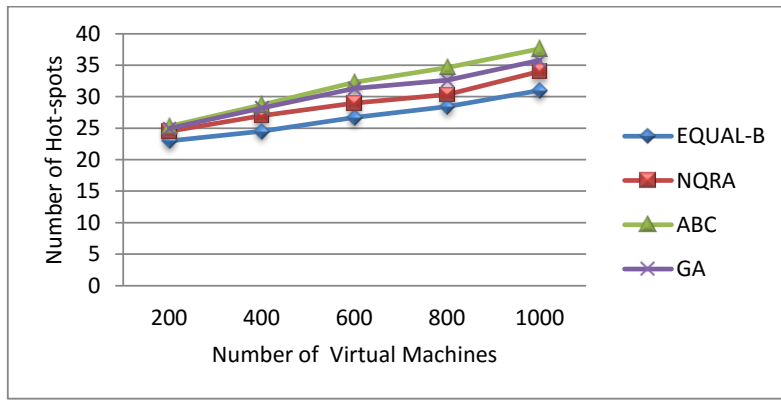

Figure 7. Comparison of the number of hot-spots

Figure 8 outlines the comparison of number of cold-spots observed in EQUAL-B, NQRA, ABC, and GA as the number of VMs is varied from 200 to 1 000 VMs. A resource is considered as a cold-spot if its utilization is below CT. The number of cold-spots portrays the extent of resource wastage. EQUAL-B creates 18, whereas NQRA, ABC, and GA create 25, 19.41, and 21.6 average number of cold-spots when tested with 1 000 VMs. The percentage of cold-spots generated in EQUAL-B, NQRA, ABC, and GA is 9.77 %, 11.52 %, 10.52 %, and

11.25 %, respectively. This shows that EQUAL-B manages the resources most efficiently.

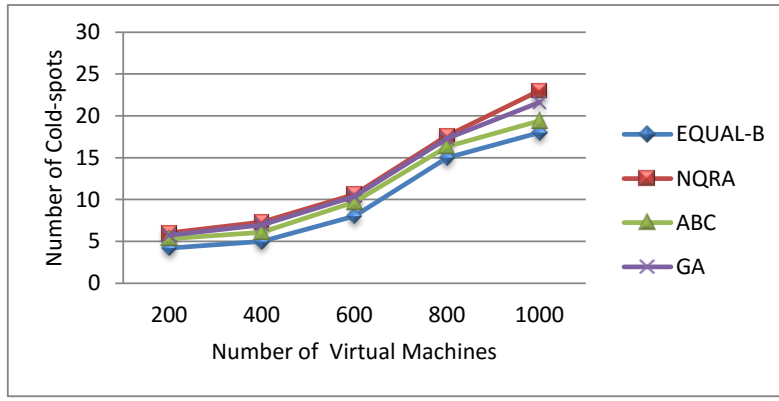

Figure 8. Comparison of the number of cold-spots

Figure 9 outlines number of deadlines missed in EQUAL-B, NQRA, ABC, and GA. A time constrained task executing in a VM is said to miss the deadline if it is not accomplished in stipulated time. The number of tasks is varied from 200 to 1 000. In each run of the simulation 200 VMs are used. The tasks are distributed equally among the VMs. It is observed that the number of deadlines missed increases with the increase in the number of tasks, but the rate of increase of missed deadlines is least in EQUAL. In EQUAL-B, 28.57 %, 11.76 %, and 25.00 % less deadline misses are observed than NQRA, ABC and GA, when number of tasks is 200. However, for 1 000 tasks, 21.58 %, 9.57 %, and 17.33 % less tasks miss their deadline in EQUAL-B as compared to NQRA, ABC, and GA.

Figure 10 shows comparison of allocation overhead that is total time taken by an algorithm to find the most suitable resource for each VMs. Allocation overhead of EQUAL is more than that of NQRA. However it is lesser than ABC and GA. In case of EQUAL-B, allocation overhead is about 26 s for 200 VMs and it increases to 112 s for 1 000 VMs. However, allocation overhead of NQRA, ABC and GA for 200 VMs is 20 s, 28.5 s and 30 s, and for 1 000 VMs it is 92 s, 122 s and 128 s, respectively. The results indicate that EQUAL-B has a higher convergence rate than ABC and GA.

Case 2: EQUAL in Energy-Aware Mode

The variable θ is assigned value 0.05 to operate EQUAL in energy aware mode (EQUAL-E). In this mode of operation, energy consumption of resources is considered while allocating resources to VMs. A VM is allocated to a resource

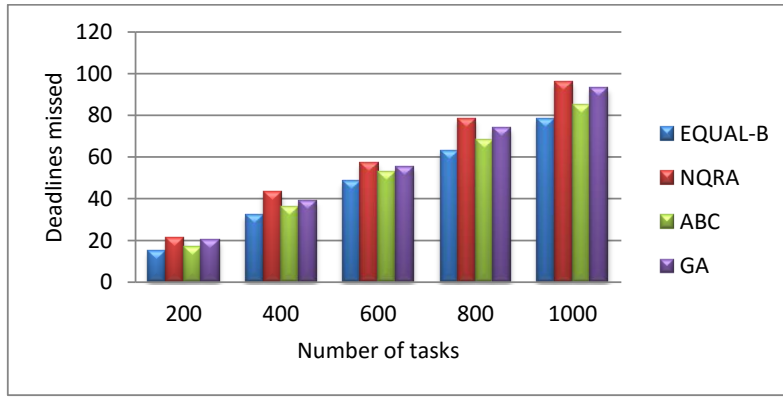

Figure 9. Comparison of number of deadlines missed

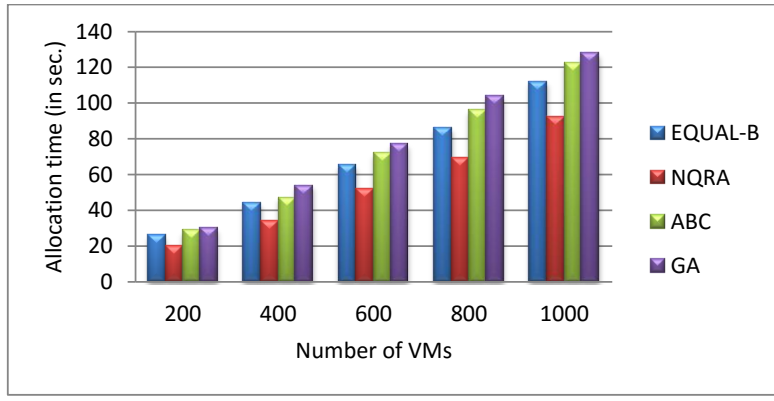

Figure 10. Comparison of total allocation time

that results in minimum increase in energy consumption. Further, the control variable γ is given value 0.05 to pack VMs on minimum number of resources.

Figure 11 depicts the comparison of number of resources used by EQUAL-E, NQRA, ABC, and GA for different number of VMs. The results show that EQUAL-E uses lesser number of resources than NQRA, ABC, and GA for a given number of VMs. It is observed that EQUAL-E uses 14.47 %, 10.05 %, and 12.54 % lesser number of resources than EQRA, ABC, and GA, respectively.

Figure 12 outlines the comparison of energy consumption of EQUAL-E, NQRA, ABC, and GA. EQUAL-E saves energy by packing VMs on lesser number of

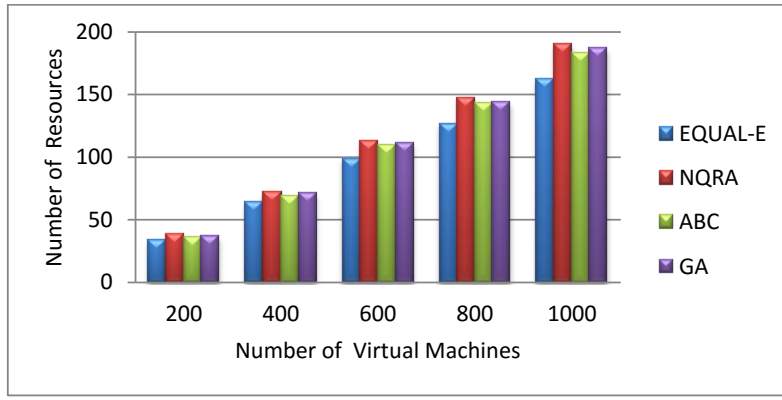

Figure 11. Comparison of number of resources required

resources. As compared to NQRA, ABC, and GA, average energy savings of 15.04 %, 11.91 %, and 14.30 % are observed in EQUAL-E.

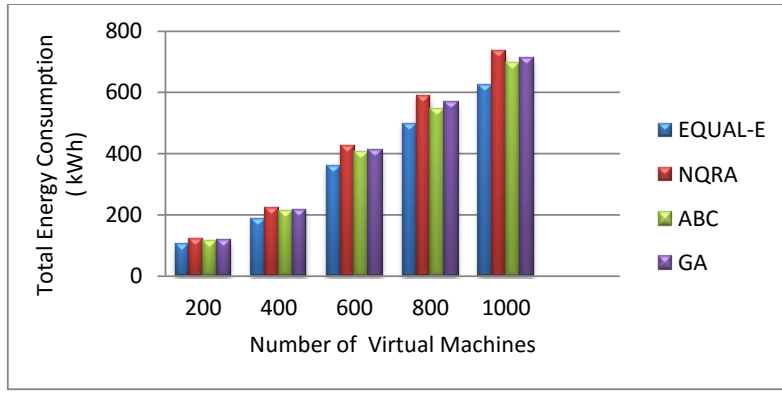

Figure 12. Comparison of total energy consumption

Figure 13 sketches comparison of average number of VM migrations performed in EQUAL-E, NQRA, ABC, and GA. VMs are migrated from either under-loaded or over-loaded resources. A resource is considered under-loaded if its utilization is below LGT, and over-loaded if its utilization is above UGT. It was observed that the number of VM migrations increases when the number of VMs increased from 200 to 1 000. In EQUAL-E, 9.37 % 3.45 %, and 6.05 % lesser number of VM migrations were observed than in NQRA, ABC, and GA, respectively.

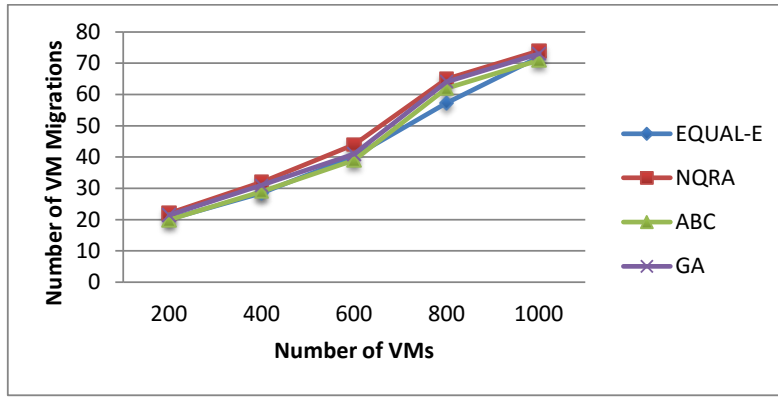

Figure 13. Comparison of the number of VM migrations

Figure 14 depicts the comparison of the number of hot-spots created in EQUAL-E, NQRA, ABC, and GA as the number of VMs are changed from 200 to 1 000 VMs. As compared to EQUAL-B, EQUAL-E packs VMs on lesser number of resources. As a result, the number of hot-spots increases and the gap of percentage number of hot-spots between EQUAL-E and NQRA, ABC, and GA was reduced to 3.86 %, 13.33 %, and 9.52 %, respectively.

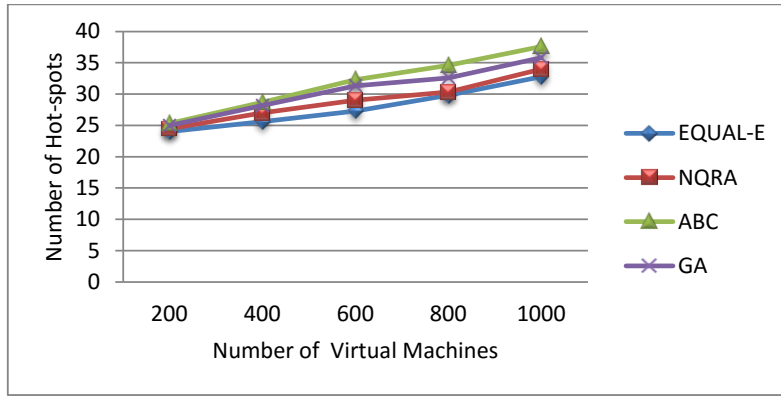

Figure 14. Comparison of the number of hot-spots

Figure 15 shows the comparison of the number of cold-spots observed in EQUAL-E, NQRA, ABC, and GA. In EQUAL-E, fewer number of cold-spots are observed than in EQUAL-B. On the average, approximately 16 cold-spots are observed in EQUAL-E against 1 000 VMs compared to 23, 19.41, and 21.6 average number

of cold-spots in NQRA, ABC, and GA, respectively, for the same number of VMs.

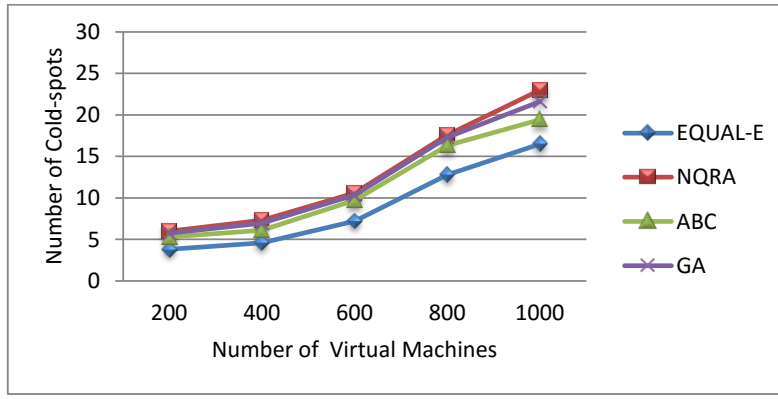

Figure 15. Comparison of the number of cold-spots

Figure 16 narrates comparison of the number of tasks that missed their deadline in EQUAL-E, NQRA, ABC, and GA. The number of tasks is changed from 200 to 1 000. In each simulation run 200 VMs are used. Further, the tasks are distributed equally among the VMs. In EQUAL-E, VMs are mapped on lesser number of resources than in EQUAL-B. As a result, tasks encapsulated in the VMs do not get sufficient resources causing increase in number of deadline miss. Due to this, percentage deadlines missed gap between EQUAL-E and the other three approaches, i.e., NQRA, ABC, and GA reduce to 13.25 %, 6.28 %, and 7.31 %, respectively.

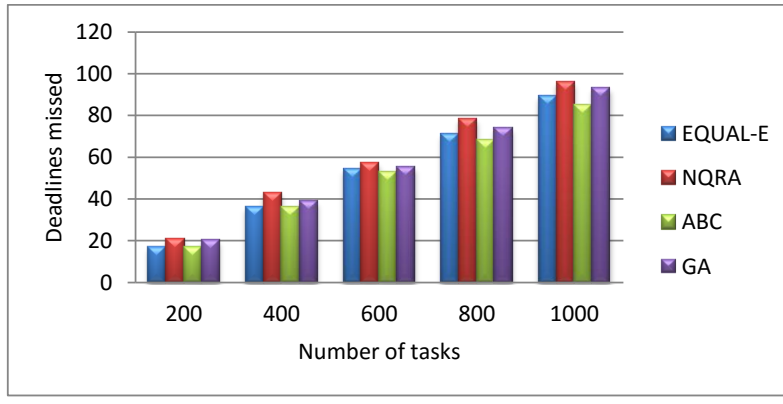

Figure 16. Comparison of the number of missed deadlines

Case 3: EQUAL in Performance-Aware Mode

The variable θ is assigned value 0.95 to tune EQUAL to performance-aware mode (EQUAL-P). In this mode of operation, available computational capacity of each resource is considered while discovering suitable resource for a VM. Since the VMs are required to be allocated on a minimum number of resources, so value 0.05 is assigned to control parameter γ.

Figure 17 shows the comparison of the number of resources used by EQUAL-P, NQRA, ABC, and GA. In EQUAL-P, a resource with higher performance affinity value is given preference over the others. In EQUAL-P more number of resources are used than in EQUAL-B and EQUAL-E for the given number of VMs. It is observed that EQUAL-E uses 8.20 %, 4.98 %, and 7.23 % lesser number of resources than NQRA, ABC, and GA, respectively.

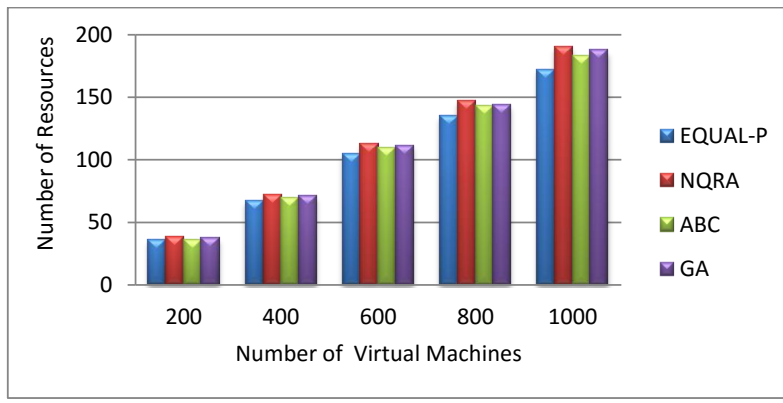

Figure 17. Comparison of the number of resources required

Figure 18 depicts the comparison of energy consumption of EQUAL-P, NQRA, ABC, and GA. Energy consumption in EQUAL-P for given number of VMs is higher than energy consumption in EQUAL-B and EQUAL-E because it uses larger number of resources. However, energy consumption in EQUAL-P is lower than that of ABC and GA. Energy consumption of EQUAL-P is measured 8.77 %, 4.73 % and 6.94 % lower for 200 VMs, and 9.75 %, 5.04 % and 6.98 % lower for 1 000 VMs than that of NQRA, ABC and GA, respectively.

Figure 19 represents a comparison of the average number of VM migrations performed in EQUAL-P, NQRA, ABC, and GA. In EQUAL-P, in average 16.8 and 64 migrations are observed for 200 VMs and 1 000 VMs, respectively. In EQUAL-P, the number of migration is lesser than the number of migrations in the balanced and energy aware mode.

Figure 20 depicts the comparison of the hot-spots created in EQUAL-P, NQRA, ABC, and GA. In EQUAL-P, fewer hot-spots than in EQUAL-P and EQUAL-B

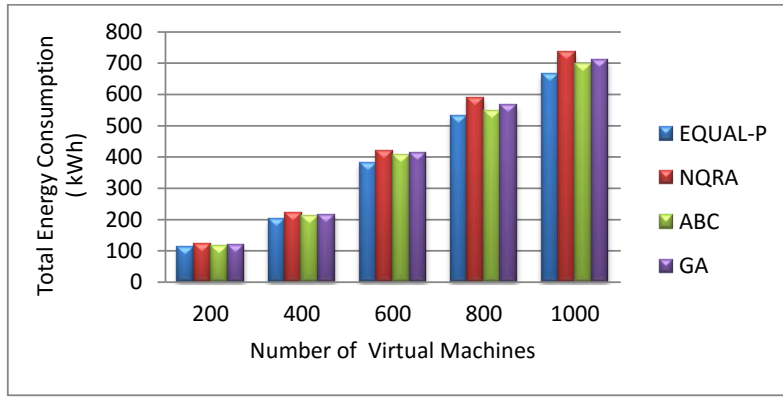

Figure 18. Comparison of the total energy consumption

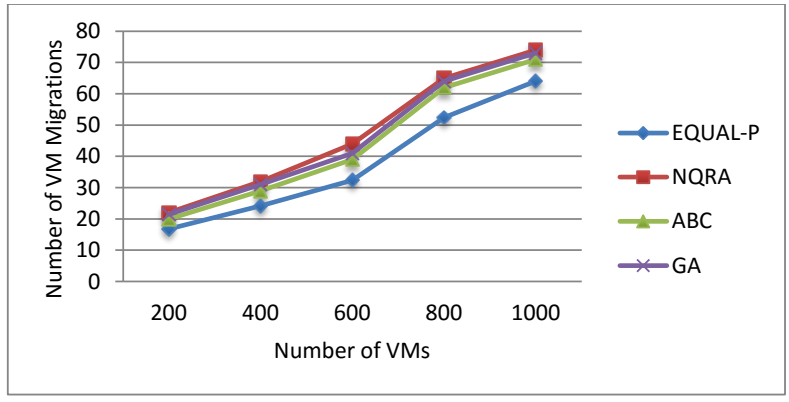

Figure 19. Comparison of the number of VM migrations

are observed. It is observed that EQUAL-P creates 11.6 %, 21.31 %, and 17.21 % lesser number of hot-spots than NQRA, ABC, and GA, respectively.

Figure 21 shows the comparison of number of cold-spots observed in EQUAL-P, NQRA, ABC, and GA. A resource is considered as a cold-spot if its utilization is below CT. A large proportion of the resource capacity goes wasted if it is a cold-spot. Therefore, the larger is the number of cold-spots the greater is the resource wastage. It is observed that EQUAL-P generates 13.68 %, 6.12 %, and 11.66 % lesser number of cold-spots than NQRA, ABC and GA. Therefore, it utilizes the resource more efficiently.

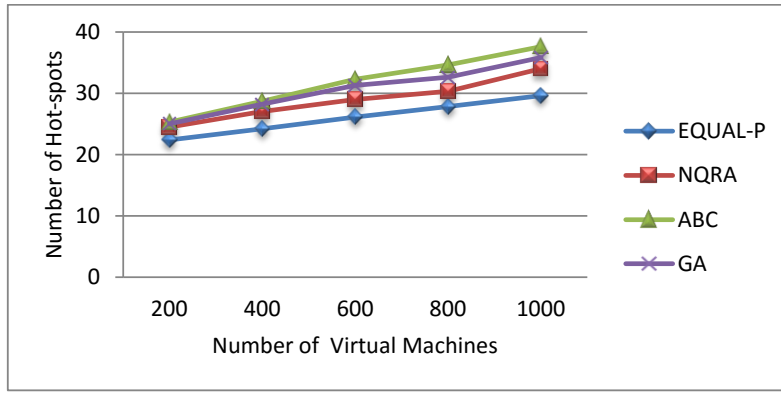

Figure 20. Comparison of the number of hot-spots

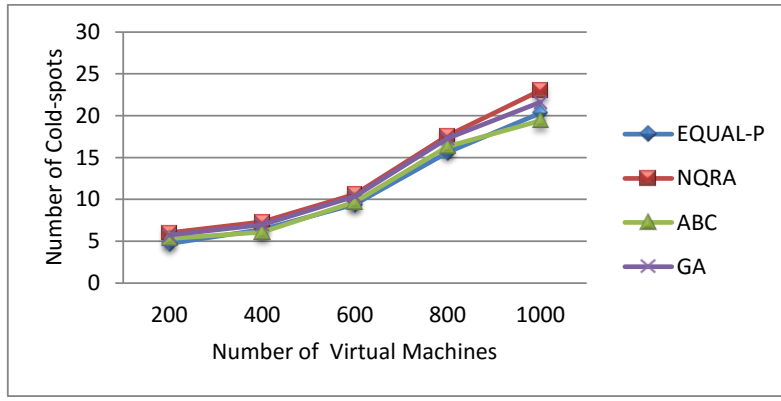

Figure 21. Comparison of the number of cold-spots

Figure 22 outlines the comparison of the average number of deadlines missed by EQUAL-P, NQRA, ABC, and GA. In each simulation run 200 VMs are used. Number of task is varied from 200 to 1 000 and tasks are distributed equally among the VMs. In EQUAL-P, fewer tasks miss their deadlines than in EQUAL-B and EQUAL-E. This is due to the fact that EQUAL-P uses more resources to map a given number of VMs. In EQUAL-P, on the average, 12 tasks miss their deadlines when the total number of tasks is 200, whereas 63 tasks miss the deadlines when the total number of tasks increased to 1 000. However, for NQRA, ABC and GA the number of tasks that missed their deadline is 21, 17 and 20 for total 200 tasks, and 96, 85 and 93 for 1 000 tasks, respectively.

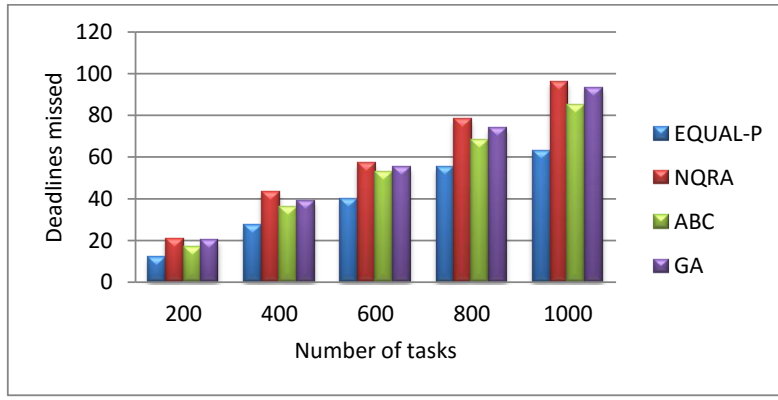

Figure 22. Comparison of the number of missed deadlines

6 CONCLUSION AND FUTURE WORK

In this work, the energy and QoS aware resource allocation approach (EQUAL) is proposed. Antlion optimization is used for allocation of resources to VMs which encapsulate heterogeneous time constrained tasks. EQUAL can be governed to operate in one of the three modes namely power aware, performance aware and balanced mode. The proposed approach was implemented in CloudSim, and tested with VMs/tasks having diverse resource requirements. The experimental results have proved that the proposed approach reduces the energy consumption up to 15 %, and also improves the quality of service in terms of reduction in the percentage of tasks that missed their deadlines. In future, the proposed approach can be further extended for tasks having mixed characteristics such as CPU intensive, memory intensive, input/output intensive, etc.

Appendix A SYMBOLIC NOTATIONS USED IN EQUAL

Table A1: List of Symbols

Symbol	Definition
P_{total}	Total power consumption of a physical machine
$P_{dynamic}$	Dynamic power consumption of a physical machine
P_{static}	Static power consumption of a physical machine
P_{idle}	Power consumption of a physical machine when idle
P_{max}	Power consumption of PM at 100 % utilization
U	Utilization of a PM
P	Power consumption of PM at U % utilization

Continued on next page

Table A1 – continued from previous page

Symbol	Definition
e	Elite solution representing the best resource
n	Number of tasks
J_i	i^{th} task
d_i	Deadline of task J_i
w_i	Processing volume of task J_i
V	Number of virtual machines
m	Number of resources
S	Set of resources
A	Set of ants
L	Set of antlions
S_j	j^{th} resource
$\Re_r^a$	Available processing power of resource r
$\Re_j^d$	Processing demand of VM j
$\triangle E_{j,r}$	Energy contribution of VM j on resource r
κ_r	Energy affinity
$\Re_{i,r}$	Fraction of processing power allocated to VM i on resource r
γ	Constant that controls energy contribution and VMs consolidation
θ	Trade off between performance and energy
$f_{j,r}$	Fitness of VM j on resource r
M^a	Matrix to store location of ants
M^{al}	Matrix to store location of antlions
M^{fa}	Matrix to store fitness values of ants
M^{fal}	Matrix to store fitness values of antlions
W_i^t	Location of i^{th} ant at t^{th} iteration
V_j^t	Location of j^{th} antlion at t^{th} iteration
α	Scaling factor for step size s
λ	Levy exponent
$L(s, \lambda)$	Levy Distribution with parameters s and λ
a_i	Minimum of random walk of i^{th} ant
b_i	Maximum of random walk of i^{th} ant
c_i^t	Minimum of search space at t^{th} iteration
d_i^t	Maximum of search space at t^{th} iteration
M_{VR}	VM-Resource map
UGT	Upper Green Threshold limit
LGT	Lower Green Threshold limit
HT	Hot-spot Threshold
CT	Cold-spot Threshold

REFERENCES

[1] Armbrust, M.—Fox, A.—Griffith, R.—Joseph, A. D.—Katz, R. H.—Konwinski, A.—Lee, G.—Patterson, D. A.—Rabkin, A.—Stoica, I.—Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing. Technical Report No. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.

[2] Columbus, L.: Predicting Enterprise Cloud Computing Growth. http://www.forbes.com/sites/louiscolumbus/2013/09/04/predicting-enterprise-cloud-computing-growth/, online; accessed 24-Jan-2016.

[3] Fargo, F.—Tunc, C.—Al-Nashif, Y.—Akoglu, A.—Hariri, S.: Autonomic Workload and Resources Management of Cloud Computing Services. International Conference on Cloud and Autonomic Computing (ICCAC), 2014, pp. 101–110, doi: 10.1109/ICCAC.2014.36.

[4] Vogels, W.: Beyond Server Consolidation. Queue, Vol. 6, 2008, No. 1, pp. 20–26, doi: 10.1145/1348583.1348590, doi: 10.1145/1348583.1348590.

[5] America's Data Centers Consuming and Wasting Growing Amounts of Energy. http://www.nrdc.org/energy/data-center-efficiency-assessment.asp, online; accessed 24-Jan-2016.

[6] Kaplan, J. M.—Forrest, W.—Kindler, N.: Revolutionizing Data Center Energy Efficiency. Technical report, McKinsy and Company, 2008.

[7] Vrbsky, S. V.—Galloway, M.—Carr, R.—Nori, R.—Grubic, D.: Decreasing Power Consumption with Energy Efficient Data Aware Strategies. Future Generation Computer Systems, Vol. 29, 2013, No. 5, pp. 1152–1163, doi: 10.1016/j.future.2012.12.016.

[8] Dahiphale, D.—Karve, R.—Vasilakos, A. V.—Liu, H.—Yu, Z.—Chhajer, A.—Wang, J.—Wang, C.: An Advanced MapReduce: Cloud MapReduce, Enhancements and Applications. IEEE Transactions on Network and Service Management, Vol. 11, 2014, No. 1, pp. 101–115, doi: 10.1109/TNSM.2014.031714.130407.

[9] Hamilton, J.: The Cost of Latency. http://perspectives.mvdirona.com/2009/10/the-cost-of-latency/.

[10] Liang, Q.—Liang, J.—Zou, F.: The Resource Configuration Method with Lower Energy Consumption Based on Prediction in Cloud Data Center. Journal of Networks, Vol. 9, 2014, No. 7, pp. 1692–1700.

[11] Lee, H. M.—Jeong, Y.-S.—Jang, H. J.: Performance Analysis Based Resource Allocation for Green Cloud Computing. The Journal of Supercomputing, Vol. 69, 2014, No. 3, pp. 1013–1026.

[12] Quarati, A.—Clematis, A.—Galizia, A.—D'Agostino, D.: Hybrid Clouds Brokering: Business Opportunities, QoS and Energy-Saving Issues. Simulation Modelling Practice and Theory, Vol. 39, 2013, pp. 121–134, doi: 10.1016/j.simpat.2013.01.004.

[13] Bobroff, N.—Kochut, A.—Beaty, K.: Dynamic Placement of Virtual Machines for Managing SLA Violations. 10th IFIP/IEEE International Symposium on Integrated Network Management, 2007, pp. 119–128, doi: 10.1109/INM.2007.374776.

[14] Wang, M.—Meng, X.—Zhang, L.: Consolidating Virtual Machines with Dynamic Bandwidth Demand in Data Centers. Proceedings of INFOCOM, 2011, IEEE, 2011, pp. 71–75, doi: 10.1109/INFCOM.2011.5935254.

[15] Takeda, S.—Takemura, T.: A Rank-Based VM Consolidation Method for Power Saving in Datacenters. IPSJ Transactions on Advanced Computing Systems, Vol. 3, 2010, No. 2, pp. 138–146, doi: 10.2197/ipsjtrans.3.88.

[16] Son, S.—Jung, G.—Jun, S. C.: An SLA-Based Cloud Computing That Facilitates Resource Allocation in the Distributed Data Centers of a Cloud Provider. The Journal of Supercomputing, Vol. 64, 2013, No. 2, pp. 606–637.

[17] Chieu, T. C.—Mohindra, A.—Karve, A. A.—Segal, A.: Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment. IEEE International Conference on e-Business Engineering (ICEBE '09), 2009, pp. 281–286.

[18] Wu, L.—Garg, S. K.—Buyya, R.: SLA-Based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments. 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2011, pp. 195–204.

[19] Raycroft, P.—Jansen, R.—Jarus, M.—Brenner, P. R.: Performance Bounded Energy Efficient Virtual Machine Allocation in the Global Cloud. Sustainable Computing: Informatics and Systems, Vol. 4, 2014, No. 1, pp. 1–9.

[20] Kim, N.—Cho, J.—Seo, E.: Energy-Credit Scheduler: An Energy-Aware Virtual Machine Scheduler for Cloud Systems. Future Generation Computer Systems, Vol. 32, 2014, pp. 128–137, doi: 10.1016/j.future.2012.05.019.

[21] Xu, J.—Fortes, J. A. B.: Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. IEEE/ACM International Conference on Green Computing and Communications (GreenCom), 2010, pp. 179–188, doi: 10.1109/GreenCom-CPSCom.2010.137.

[22] Wu, C.-M.—Chang, R.-S.—Chan, H.-Y.: A Green Energy-Efficient Scheduling Algorithm Using the DVFS Technique for Cloud Datacenters. Future Generation Computer Systems, Vol. 37, 2014, pp. 141–147, doi: 10.1016/j.future.2013.06.009.

[23] Beloglazov, A.—Buyya, R.: Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints. IEEE Transactions on Parallel and Distributed Systems, Vol. 24, 2013, No. 7, pp. 1366–1379, doi: 10.1109/TPDS.2012.240.

[24] Nathuji, R.—Schwan, K.: VirtualPower: Coordinated Power Management in Virtualized Enterprise Systems. ACM SIGOPS Operating Systems Review, Vol. 41, 2007, No. 6, pp. 265–278, doi: 10.1145/1294261.1294287.

[25] Hsu, C.-H.—Chen, S.-C.—Lee, C.-C.—Chang, H.-Y.—Lai, K.-C.—Li, K.-C.—Rong, C.: Energy-Aware Task Consolidation Technique for Cloud Computing. IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), 2011, pp. 115–121, doi: 10.1109/CloudCom.2011.25.

[26] Beloglazov, A.—Abawajy, J.—Buyya, R.: Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. Future Generation Computer Systems, Vol. 28, 2012, No. 5, pp. 755–768, doi: 10.1016/j.future.2011.04.017.

[27] Kusic, D.—Kephart, J. O.—Hanson, J. E.—Kandasamy, N.—Jiang, G.: Power and Performance Management of Virtualized Computing Environments via Lookahead Control. International Conference on Autonomic Computing (ICAC '08), 2008, pp. 3–12, doi: 10.1109/ICAC.2008.31.

[28] Gao, Y.—Guan, H.—Qi, Z.—Song, T.—Huan, F.—Liu, L.: Service Level Agreement Based Energy-Efficient Resource Management in Cloud Data Centers. Computers and Electrical Engineering, Vol. 40, 2014, No. 5, pp. 1621–1633.

[29] Feller, E.—Rilling, L.—Morin, C.: Energy-Aware Ant Colony Based Workload Placement in Clouds. IEEE/ACM 12th International Conference on Grid Computing (GRID), 2011, pp. 26–33, doi: 10.1109/Grid.2011.13.

[30] Gao, Y.—Guan, H.—Qi, Z.—Hou, Y.—Liu, L.: A Multi-Objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing. Journal of Computer and System Sciences, Vol. 79, 2013, No. 8, pp. 1230–1242.

[31] Chen, H.—Cheng, A. M. K.—Kuo, Y.-W.: Assigning Real-Time Tasks to Heterogeneous Processors by Applying Ant Colony Optimization. Journal of Parallel and Distributed Computing, Vol. 71, 2011, No. 1, pp. 132–142.

[32] Huang, C.-J.—Guan, C.-T.—Chen, H.-M.—Wang, Y.-W.—Chang, S.-C.—Li, C.-Y.—Weng, C.-H.: An Adaptive Resource Management Scheme in Cloud Computing. Engineering Applications of Artificial Intelligence, Vol. 26, 2013, No. 1, pp. 382–389.

[33] Kansal, N. J.—Chana, I.: Artificial Bee Colony Based Energy-Aware Resource Utilization Technique for Cloud Computing. Concurrency and Computation: Practice and Experience, Vol. 27, 2015, No. 5, pp. 1207–1225.

[34] Chimakurthi, L.—Kumar, S. D. M.: Power Efficient Resource Allocation for Clouds Using Ant Colony Framework. CoRR abs/1102.2608, `http://arxiv.org/abs/1102.2608`.

[35] Hu, W.—Zheng, J.—Hua, X.—Yang, Y.: A Computing Capability Allocation Algorithm for Cloud Computing Environment. Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), 2013, pp. 2258–2262, doi: 10.2991/iccsee.2013.566.

[36] Liu, X.-F.—Zhan, Z.-H.—Du, K.-J.—Chen, W.-N.: Energy Aware Virtual Machine Placement Scheduling in Cloud Computing Based on Ant Colony Optimization Approach. Proceedings of the 2014 Conference on Genetic and Evolutionary Computation (GECCO '14), 2014, pp. 41–48, doi: 10.1145/2576768.2598265.

[37] Portaluri, G.—Giordano, S.—Kliazovich, D.—Dorronsoro, B.: A Power Efficient Genetic Algorithm for Resource Allocation in Cloud Computing Data Centers. IEEE 3rd International Conference on Cloud Networking (CloudNet), 2014, pp. 58–63, doi: 10.1109/CloudNet.2014.6968969.

[38] Xiong, A.-P.—Xu, C.-X.: Energy Efficient Multiresource Allocation of Virtual Machine Based on PSO in Cloud Data Center. Mathematical Problems in Engineering, 2014, pp. 1–8, http://www.hindawi.com/journals/mpe/2014/816518/.

[39] Kumar, D.—Raza, Z.: A PSO Based VM Resource Scheduling Model for Cloud Computing. 2015 IEEE International Conference on Computational Intelligence and Communication Technology, 2015, pp. 213–219, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7078697.

[40] Dashti, S. E.—Rahmani, A. M.: Dynamic VMs Placement for Energy Efficiency by PSO in Cloud Computing. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 28, 2016, No. 1-2, pp. 97–112, doi: 10.1080/0952813X.2015.1020519.

[41] Kansal, N. J.—Chana, I.: Energy-Aware Virtual Machine Migration for Cloud Computing – A Firefly Optimization Approach. Journal of Grid Computing, Vol. 14, 2016, No. 2, pp. 327–345, doi: 10.1007/s10723-016-9364-0.

[42] Kumar, A.—Kumar, R.—Sharma, A.: Energy Aware Resource Allocation for Clouds Using Two Level Ant Colony Optimization. Computing and Informatics, Vol. 37, 2018, No. 1, pp. 76–108.

[43] Kumar, R.—Kumar, A.—Sharma, A.: A Bio-Inspired Approach for Power and Performance Aware Resource Allocation in Clouds. MATEC Web of Conferences, Vol. 57, 2016, Art. No. 02008, 6 pp.

[44] 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, Vol. 24, 2012, No. 13, pp. 1397–1420, doi: 10.1002/cpe.1867, doi: 10.1002/cpe.1867.

[45] Beloglazov, A.: Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing. Ph.D. thesis, University of Melbourne, 2013.

[46] Barham, P.—Dragovic, B.—Fraser, K.—Hand, S.—Harris, T.—Ho, A.—Neugebauer, R.—Pratt, I.—Warfield, A.: Xen and the Art of Virtualization. ACM SIGOPS Operating Systems Review – SOSP '03, Vol. 37, 2003, No. 5, pp. 164–177, doi: 10.1145/945445.945462.

[47] Kivity, A.—Kamay, Y.—Laor, D.—Lublin, U.—Liguori, A.: KVM: The Linux Virtual Machine Monitor. Proceedings of the Linux Symposium, Ottawa, Ontario, Canada, Vol. 1, 2007, pp. 225–230, http://linux-security.cn/ebooks/ols2007/OLS2007-Proceedings-V1.pdf.

[48] Walters, B.: VMware Virtual Platform. Linux Journal, 1999, No. 63 es., Art. No. 6, http://dl.acm.org/citation.cfm?id=327906.327912.

[49] Mirjalili, S.: The Ant Lion Optimizer. Advances in Engineering Software, Vol. 83, 2015, pp. 80–98, doi: 10.1016/j.advengsoft.2015.01.010.

[50] Yang, X.-S.—Deb, S.: Cuckoo Search via Lévy Flights. World Congress on Nature and Biologically Inspired Computing (NaBIC), 2009, pp. 210–214, doi: 10.1109/NABIC.2009.5393690.

[51] Calheiros, R. N.—Ranjan, R.—Beloglazov, A.—De Rose, C. A. F.—Buyya, R.: CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing

Environments and Evaluation of Resource Provisioning Algorithms. Software: Practice and Experience (SPE), Vol. 41, 2011, No. 1, pp. 23–50.

[52] Wickremasinghe, B.—Calheiros, R. N.—Buyya, R.: CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications. 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010, pp. 446–452, doi: 10.1109/AINA.2010.32.

[53] Kliazovich, D.—Bouvry, P.—Audzevich, Y.—Khan, S. U.: GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers. Global Telecommunications Conference (GLOBECOM 2010), IEEE, 2010, pp. 1–5, doi: 10.1109/GLOCOM.2010.5683561.

[54] Garg, S. K.—Buyya, R.: NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations. 2011 Fourth IEEE International Conference on Utility and Cloud Computing (UCC), 2011, pp. 105–113, doi: 10.1109/UCC.2011.24.

[55] General Purpose Compute: Basic Tier. `https://azure.microsoft.com/en-in/pricing/details/virtual-machines/`, online; accessed 29-March-2016.

[56] Xu, J.—Fortes, J. A. B.: Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. IEEE/ACM International Conference on Green Computing and Communications (GreenCom), 2010, pp. 179–188, doi: 10.1109/GreenCom-CPSCom.2010.137.

[57] Calheiros, R. N.—Buyya, R.: Meeting Deadlines of Scientific Workflows in Public Clouds with Tasks Replication. IEEE Transactions on Parallel and Distributed Systems, Vol. 25, 2014, No. 7, pp. 1787–1796, doi: 10.1109/TPDS.2013.238.

[58] Juve, G.—Chervenak, A.—Deelman, E.—Bharathi, S.—Mehta, G.—Vahi, K.: Characterizing and Profiling Scientific Workflows. Future Generation Computer Systems, Vol. 29, 2013, No. 3, pp. 682–692.

[59] Abbott, R.—Garcia-Molina, H.: Scheduling Real-Time Transactions. ACM SIGMOD Record – Special Issue on Real-Time Database Systems, Vol. 17, 1988, No. 1, pp. 71–81.

Ashok Kumar received his M.Sc. degree in information technology from Punjab Technical University, Jalandhar. Currently he is pursuing his doctoral degree in cloud computing from Thapar University, Patiala. His research interests include cloud computing, internet of things and fog computing. He has five research publications in reputed journals and conferences.

Rajesh Kumar is currently working as Professor in the Computer Science and Engineering Department, Thapar University, Patiala. He received his M.Sc., M.Phil. and Ph.D. degrees from IIT Roorkee. He has more than 21 years of UG & PG teaching and research experience. He wrote over 101 research papers for various international and national journals and conferences. He has guided 10 Ph.D. and 23 M.E./M.Sc. theses so far. His current areas of research interests include FANETs, resource scheduling and fault tolerance in clouds.

Anju Sharma is working as Assistant Professor in the Computer Science and Engineering Department, MRSPTU, Bathinda. Her research interests include smart grid computing, cloud computing, IoT and fog computing. She has varied numbers of publications in international journals and conferences of repute. She is Senior Member of International Association of Computer Science and Information Technology (IACSIT) and a professional member of ACM India, IEEE. She is an active member (TCM and reviewer) of varied conferences.