Journal of King Saud University – Computer and Information Sciences xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment

Bela Shrimali [a,b,*], Hiren Patel [b]

[a] C.U. Shah University, Wadhwan City, Surendranagar, Gujarat 363030, India
[b] LDRP Institute of Technology and Research, Gandhinagar, Gujarat, India

ARTICLE INFO

Article history:
Received 8 July 2017
Revised 9 November 2017
Accepted 1 December 2017
Available online xxxx

Keywords:
Power efficiency
Optimization
Resource allocation
Resource utilization
Cloud

ABSTRACT

Cloud computing is a hybrid paradigm which makes use of utility computing, high performance cluster computing and grid computing and it offers various benefits such as flexibility, expandability, little or almost no capital investment, disaster recovery, moveable work space and much more. However, due to constantly increasing number of data centers worldwide, the issue of energy consumption by these data centers has attracted attention of researchers. Resource allocation and resource utilization are the major criterion in which the problem of energy efficiency can be addressed. In this research, we aim to provide energy-efficient resource allocation using Multi-Objective Optimization (MOO) method. Further, We propose MOO-based virtual machine (VM) allocation policy and implement it in CloudSim environment. The results are compared with existing policies. The results depict that MOO-based policy leads to saving in energy due to efficient resource allocation, without compromising performance of data center operations.

1. Introduction

Hasty revolution in virtualization technologies make Cloud computing one of the most popular domains in the era of high performance and distributed computing. It has come up as a large pool of resources that provides different services like platform, software and infrastructure (Mell and Grance, 2009). Platform as a service (PaaS) provides platform to the end users to develop,execute and manage the applications without worrying about internal infrastructure. Software as a Service (SaaS) provides software deployed over the internet on end user's request and Infrastructure as a Service (IaaS) provides physical resources on user's demand. Out of these three services offered, the Infrastructure as a Service (IaaS) is further categorized into three layers, viz. physical resource layer, virtual resource layer, and management tool layer (Sosinsky, 2010). The physical resource layer consists of traditional data center and contains thousands of servers, network devices, disks and non-IT components like cooling equipment, lightening, air conditioner, etc. These different services are available to end users through virtualization. Virtualization is a process of generating virtual computing resources from available physical resources. User's requests are assigned on one or more of these virtual computing resources based on availability and configuration. Creation and management of virtual resources in form of virtual machines (VMs) are carried out using a component known as Virtual Machine Monitor (VMM) or hypervisor. VMM is a part of virtual resource layer which contains a pool of computing or storage resources which are virtualized from the physical resource using virtualization techniques. Some examples of VMM are KVM (Kivity et al., 2007), Xen (Barham et al., 2003), VMWare (http://www.vmware.com) and Hyper-V (http://www.microsoft.com/hyper-v-server/en/us/default.aspx). The management tool layer is responsible for virtual resource management, accounting and monitoring. OpenNebula (Borja et al., 2009) and Eucalyptus (Nurmi et al., 2009) are the examples of this layer. The quality of service(QoS) in Cloud environment depends upon the performance of physical host/server in data center. Cloud infrastructure that contains a large number of processing elements, auxiliary storage, memory, network bandwidth etc. which make it possible to handle millions of requests from users around the globe. However, the consumers need to pay for usage to the service providers. On the

* Corresponding author at: LDRP Institute of Technology and Research, Gandhinagar, Gujarat, India.
E-mail addresses: bela_ce@ldrp.ac.in (B. Shrimali), hod_ce@ldrp.ac.in (H. Patel).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2017.12.001
1319-1578/

service provider's side, maintenance and management of this large-scale data centers need to be emphasized for revenue generation. Due to enormous computing requirement worldwide, during last decades, number of data centers has increased considerably. This has led to an issue of energy consumption by these data centers and subsequently affecting environment and financial impacts. For instance, in Amazon's data centers, it is identified and reported (Hamilton et al., 2009) that i) Expenses related to the cost and operation of the servers is 53% of total budget. ii) Energy-related costs is about to 42% of total budget that includes both direct energy consumption of 19% by servers and Power used in cooling the infrastructure about 23%. Gartner Report 2007 states that 2% of total CO_2 emissions in environment is contributed by IT industry. U.S.EPA report in 2007 also mentioned that 1.5% of total U.S power consumption is used by data centers and costs $ 4.5 billion. Thus, it is utmost significant for service providers to think upon these two major affecting factors viz. performance and power consumption in Cloud. An efficient resource allocation policies could be used to address these factors. The policy should take into account the process of appropriately mapping the job on virtual resource and mapping virtual resource on physical server. Energy-efficient resource allocation while maintaining performance is a challenging issue. Performance and energy based VM allocation can be viewed as the NP-Complete problem as our aim is to reduce energy consumption and to increase performance (Mann et al., 2015; Ghribi et al., 2013; Dabbagh et al., 2015; Masdari et al., 2016; Jing et al., 2013; Raycroft et al., 2014; Panda and Jana, 2015; Xiong and Perros, 2009). One of the challenges of energy-efficient and performance oriented scheduling algorithm is the trade-off between energy consumption and performance. In this paper, we address this issue by a method of the multi-objective optimization and propose a novel algorithm called Pareto optimal Multi-Objective Optimization based Allocation (MOOA) for Cloud environment. Multi-objective optimization is a mathematical method of systematic and simultaneous optimization of objectives' set. Unlike the single objective optimization, multi-objective optimization has a set of non-dominated solutions considering all objectives known as a pareto optimal solutions (Zitzler and Thiele, 1998). This pareto optimal outcome is called pareto front (Zitzler and Thiele, 1998). A simple single-objective optimization problem can be formulated as min f(x), where $x \in S$, where S is a set of constraints. Whereas, multi-objective optimization problem can be formulated as: min f1(x),f2(x),f3(x),...,fn(x) where $x \in S$. There are four categories of methods in multi-objective optimization (Marler and Arora, 2004).

- **Methods with apriori articulation of preferences:** Preferences are articulated by Decision Maker (DM) or user for different objective functions. This technique is used when the priority of objectives is articulated. The user needs to give the preference priory to obtain the desirable solution.
- **Methods with a posteriori articulation of preferences:** User selects the efficient solution from set of solutions and accordingly the preference of the objectives are finalized. This method is used when the user is aware of set of non-dominated solutions.
- **Methods with no articulation of preferences:** The DM cannot concretely define the preferences all the time. No preference is required in this method. It is a simplification of the apriori articulation of preference method. It performs operations directly on objectives.
- **Progressive articulation of preferences or Interactive method:** The DM needs to give the preferences at every fixed iteration. DM needs to be careful with the preference of objective to obtain an optimal solution.

Hierarchy shown in Fig. 1 summarizes the different method of Multi-objective optimization. Selection of the method depends on the type of problem. Their comparisons are shown in Table 1. Based on this comparison, a method of priory articulation of preference is identified as suitable for our work. This is because of the fact that the user is not aware about the possible solution/result, but can predict/give preferences to non dominated objectives to generate pareto front. To do so, we prefer to use a priory articulation of preferences. Further, comparison on different methods of priory articulation of preference are shown in Table 2. Based on this comparison, we found weighted sum method of apriori articulation of preferences suitable for our problem, as accuracy in weight selection generates pareto optimal set. In this method, prior weightages are given to different objectives. This weightages are

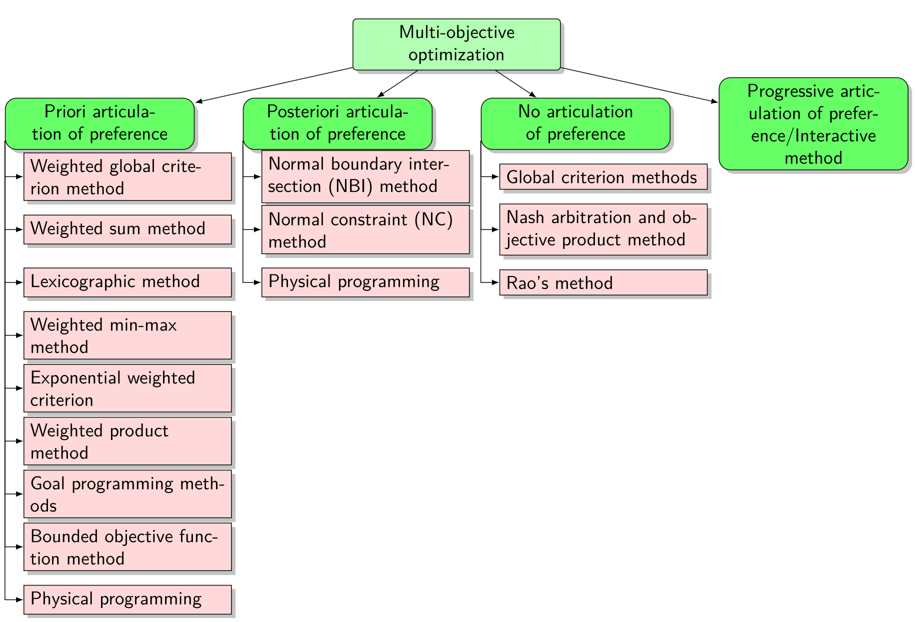

Fig. 1. Different methods of Multi-objective optimization (Marler and Arora, 2004).

Please cite this article in press as: Shrimali, B., Patel, H. Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment. Journal of King Saud University – Computer and Information Sciences (2017), https://doi.org/10.1016/j.jksuci.2017.12.001

Table 1
Comparisons of method of multi-objective optimization (Marler and Arora, 2004).

Method Name	Key characteristic	Preference of objective	Solution Identified/ Goal specified
A priori articulation of preferences	User indicates the relative importance of the objective functions or desired goals before running the optimization algorithm.	prior required	No
A posteriori articulation of preferences	Which entails electing single solution from a set of mathematically equivalent solutions	Not required	Yes
Progressive articulation of preferences	Decision-maker needs to continuously provide input during the running of the algorithm	Required continuously	Yes
No Articulation of Preferences	No preferences required	Not required	No

Table 2
Comparison of different methods of apriori articulation of preferences.

Method Name	Key characteristic	Preference articulation	Advantages	Disadvantages	Code complexity
Weighted Global Criterion	All objective functions are combined to form a single function	Method parameters are used to define preferences	It can give optimal pareto set in accurately defined preferences.	The fixed value of power will limit prediction of the calculation on weight/ preference.	Simple
Weighted Sum	All objective are combined to form a single function with weight attached with each objective.	A very less preference information required. User does not provide extensive input.	Easy and simple to implement. Accuracy in weight selection will give pareto optimal set.	Weights of objective function needs to be identified accurately otherwise pareto optimal points and set cannot be obtained.	Simple
Lexicographic	Objective functions are arranged in order of importance and after wards the objectives are minimized.	Clear.	Straight forward method	computational expense increase as multiple problems solve individually.	Average
Weighted Min–Max	The method is about to minimization and maximization of the objectives by introducing constraint.	Weightage is assigned to overall function. It is predefined.	It provides the complete Pareto optimal set by variation in the weights.	Number of constrain can increase the number of complexity.	Average
Exponential Weighted	Exponential is introduced in weighted sum method.	Predefined	Give pareto optimal set.	Large values of parameters used in method may lead to numerical overflow.	Complex
Weighted Product	Weight is apply as a power of objective function.	Unclear	A functions with different significance are handled.	As the characteristics of the weights are unclear cannot obtain efficient pareto optial set.	Average
Goal Programming	Goals are specified for each objectives.	Goals are clear.	It has wide range of application as it achieves/optimize goals one by one.	There is no guarantee of a Pareto optimal solution.Cannot handle larger objectives.	Average
Bounded Objective Function	Minimizes the single most important objective function, all other objectives are used to create additional constraints for objective function.	Not required, rather it sets limit on the objectives.	Consistent variation in $\overset{ndash}{D}$ D̆ parameters, may obtain the complete Pareto optimal set.	Selection of parameter to get feasible region is complex.	Average
Physical Programming	It creates utility function based on objective functions, constraints, and goals	Clear.	Customize a more complex and accurate individual utility function for each objective.	Significant knowledge of objectives, constraint and goals required.	Complicated

actually a service providers' priority to objectives as mention in Table 2.

Multi-objective optimization approach is also used for task level scheduling for heterogeneous Cloud in Panda and Jana (2015) as discussed in related work. For resource level scheduling, researches (Beloglazov and Buyya, 2010a,b, 2012; Dong et al., 2013; Buyya et al., 2010; Piao and Yan, 2010; Zamanifar et al., 2012) have focused on VM allocation policies. These allocation policies are further categorized based on different parameters like performance, energy, network,data awareness and SLA (Shrimali and Patel, 2015). Also, other researchers (Van et al., 2010; Xiao et al., 2013) have focused on VM allocation policies that consider both i.e performance and energy as important parameters. However, it also shows a noticeable tradeoff between both of the parameters. The rest of the paper is organized as follows. Section 2 describes related work associated with this domain. Section 3 illustrates analytical model and problem statement of multi-objective optimization allocation policy, followed by proposed algorithm shown in Section 4. Conclusion and future work are depicted in Section 5. The references used in the paper are listed in Section 6.

2. Related work

In this section, we have reviewed recent research on resource provisioning in virtualized computing environments. Quang-Hung et al. (2014) have presented energy-efficient scheduling of VMs by using Energy-aware Performance-per-watt Oriented Best-Fit heuristics algorithms in which energy consumption is handled efficiently but execution time of the application is considerable. Calheiros et al. (2011) have introduced VM based energy efficient data center architecture, which performs the energy efficient allocation and live migration of virtual machines along with server consolidation.Their research claims to achieve energy-saving goals with little performance overheads. Van et al. (2010) have proposed a resource management framework combining a utility-based dynamic virtual machine provisioning manager and a dynamic VM placement manager. Problems of power consumption and performance are modeled as constraint satisfaction problems in their work. Lee and Zomaya (2012) proposed two energy-conscious task consolidation heuristics on which the energy consumption for executing the task is explicitly or implicitly minimized without performance degradation of that task. They claimed to have promising energy-saving capabilities with little performance overhead. Xiao et al. (2013) have introduced heuristics algorithm to prevent overload in the system effectively with energy saving. They have also focused on both the parameters and their results of experiments show its high effectiveness in preventing server overload proactively. Lin et al. (2011) proposed a threshold-based dynamic resource allocation scheme that allocate the virtual resources dynamically among the Cloud computing applications based on their load. Their scheme monitors and predicts the resource requirement of the Cloud applications and adjust the virtual resources accordingly. However, they have not considered energy efficiency. In our previous work (Shrimali and Patel, 2015), we have carried out a comparative study of different VM allocation techniques based on a performance and energy that also depicts a noticeable trade-off between the two factors. A survey and experiments carried out in Jansen and Brenner (2011) compared different techniques of energy efficient VM allocation techniques. Their comparisons show that while managing energy, CPU load is increased. Kusic et al. (2009) handle power and performance using multi-objective optimization approach under Limited Lookahead Control(LLC). We consider their method with LLC as a base method for comparison with our work. In Hameed et al. (2016), Abdul et al. have identified few open challenges related with energy efficient resource allocation and presented research taxonomy on existing techniques (classified as hardware and software based) along with various dimensions such as resource adoption policy, objective function, allocation method, allocation operation and interoperability. Further, authors have highlighted several open issues and proposed various prospective research directions. Ghribi et al. (2013) have presented two scheduling algorithms (allocation and migration) for energy efficient VM scheduling based on VM migration on service departures with an objective to minimize power consumption. The problem has been considered as NP-hard bin packing problem. With experimentation, authors have claimed significant energy saving depending on system loads. Masdari et al. (2016) have defined the problem of resource allocation as finding an optimal pair(s) of a virtual machine and a physical machine. The optimal pair is believed to be effecting the performance, resource utilization and power consumption. Authors have surveyed various existing VM placement mechanisms and assessed their capabilities and objectives. Panigrahia et al. (2015) have recognized the tradeoff between maximizing resource utilization and minimizing energy consumption and proposed two algorithms (SLA-based task consolidation STC and Threshold-based task consolidation DTTC) to reduce energy consumption. With experimentation, authors have claimed the algorithms to outperform in comparisons FCFS in terms of energy consumption and number of task completion. With the help of fog computing, Shojafar et al. (2016a) have attempted to address the issue of delay and delay-jitter in real time Cloud services to vehicular clients. Authors have proposed an energy-efficient adaptive resource scheduler (NetFCs) with an aim to maximize the overall communication and computing energy efficiency while meeting QoS requirements (reducing transmission rates, improving delays and delay-jitters). The performance of the proposed scheduler is numerically tested against some state-of-art schedulers with real-world datasets. Authors have proposed to consider closed networked multi-tier computing infrastructures and live VM migration for forecasting as part of future work. Shojafar et al. (2016b) have addressed the issue of energy efficiency in Cloud data centers by considering the traffic engineering to dynamically adapt the number of active servers to the current workload. Authors have proposed an optimization framework known as MMGreen for compute intensive tasks such as multimedia data processing with huge amount of data. Through experimentation, authors have claimed to achieve saving in energy along with maintaining SLA. Zhao et al. (2015) have presented an on-going funded project titled Software Workbench for Interactive, Time Critical and Highly self-adaptive Cloud applications (SWITCH) for developing software methods and tools for the entire life cycle that addressed the issue for time critical applications such as customized development of dedicated infrastructure and maintenance of system performance in dynamic infrastructure. Authors have developed a concept known as application-infrastructure co-programming and control model with the functions of programmability and controllability. Kimovski et al. (2016) have recognized the issue of portability and vendor lock-in in federated Cloud and presented a concept known as ENTICE for VM repository and operational environment. To improve the loading time, delivery time and the execution time along with enhancement in QoS, ENTICE optimizes the VM images for specific Cloud infrastructure based on various factors through efficient development and management of VM images. Authors have proposed to work on developing new knowledge based model incorporating multi-objective optimization framework. In addition, a weighted sum method of multi-objective optimization is proposed in Panda and Jana (2015) for task level scheduling in heterogeneous multi-Cloud environment. It's aim is to minimize makespan and total cost of services for end users. However, in this research we have used multi-objective optimization approach for resource level scheduling in homogeneous single Cloud environment. Moreover, our aim is to minimize power consumption and SLA violations as compared to Panda and Jana (2015).

So, in this research, by considering energy and performance, we introduce a MOO-based technique for efficient resource allocation.

3. System model and problem formulation

In this section, we describe Cloud's Performance and Energy (PE) model. We also describe the virtual machine allocation problem targeted in our work. The details presented in the model are from the perspective of different service providers.

Consider a Cloud comprises of a large scale data center of homogeneous physical nodes. Fig. 2 depicts the overall organization of the Cloud which we consider for our experimental setup. The Cloud service provider has multiple datacenters situated at various geographical locations. Every data center is independent on other as far as functionality is concerned. Each datacenter is comprised of multiple clusters which are essentially nothing but group of nodes. Every cluster maintains a cluster controller (CC) which is a centralized component to monitor the status of each node's utilization under it. Every physical node is comprised of multiple virtual machines (VMs) which are provided as part of virtual resources to users upon their requests. The overall utilization of a physical node is directly proportional to the sum of utilization of virtual machine under the same.

Hereafter, the names, host, node and server, are used interchangeably. We have assumed the node to be in two modes viz (i) active and (ii) idle. In active mode, a node processes the requests whereas it does not processes any request in idle mode. Requests from different users are mapped to the different nodes based on the resource availability. Many criterion such as utilization,

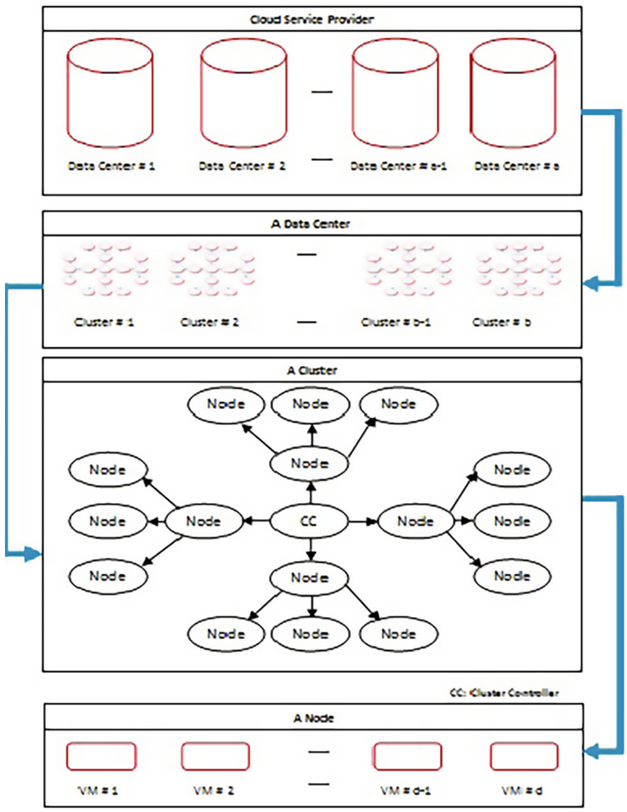

Fig. 2. Organizational Architecture of Cloud.

resource availability, priority from user, predicted workload of host etc are used to map request on a host. For our model, we have taken utilization factor for selection of the host. Further, we consider the issue of VM allocation in the terminology of request allocation to a specific host.

Before we study the performance and energy model, we would like to represent the architectural features of physical servers. Every server is comprised of n cores each having m Millions Instructions Per Second (MIPS) capacity, resulting into total $n * m$ MIPS capacity per server. We further assume that the required MIPS by any VM is less than available host capacity at the time of VM to host allocation.

3.1. Performance and energy (PE) model

A linear relation (Lin et al., 2015) between processor utilization and energy consumption is considered for selecting a server in current Cloud model. Beloglazov and Buyya (2012) have analyzed the power consumption of the selected server at different load level which claims that there is a small variance in a power consumption by ideal and active nodes. The power consumption by the selected servers are shown in Table 3. As can seen from table, an idle node consumes about 73.50% power consumption of its peak capacity. Hence, it is recommended to turn off these idle nodes for saving energy consumption.

Further, based on utilization of node, the new incoming job requests may be assigned to appropriate node. For this, we bifurcate the nodes among three categories viz. (i) under-utilized (ii) over-utilized (iii) moderately utilized.

It is apparent that assigning new request to already over utilized host could result into performance degradation and SLA violation, whereas, assigning these requests to underutilized hosts would block the hosts from turning off in future. So, we see that moderately utilized host are most suitable for energy efficient and performance aware allocation. Hence, utilization of node is considered for proper node selection.

For our model, we assume that resource demand by application is static and hence, our second parameter viz. performance of the system is determined by the performance of application deployed on a system. It can be determined by the characteristics like minimum throughput, maximum response time and maximum execution time. As these characteristics can vary for different workload, to evaluate performance independent metric, SLA is considered to evaluate performance (Beloglazov and Buyya, 2012). Performance of application is affected by fluctuation in CPU demands of host. If VM request of CPU demand exceeds to CPU capacity of host, it results into performance degradation and hence, violations of SLA. When CPU demand is higher than CPU capacity, VM will be migrated to new host to improve performance. When performance of application degrades, it affects to SLA. To evaluate SLA Violations (SLAV), we have considered SLA violation Time per Active Node (SLATAN) and the overall performance degradation due to VM migration(PDM). We are interested in selecting a host with less SLAV (Beloglazov and Buyya, 2012). Two measurement metrics are used to calculate SLAV:

1. SLA violation time per active node(SLATAN) (Beloglazov and Buyya, 2012) (where node experience maximum utilization) (Eq. (1))
2. Performance degradation due to migration(PDM) (Beloglazov and Buyya, 2012) of virtual machine (Eq. (2)).

$$SLATAN = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{si}}{T_{ai}} \tag{1}$$

$$PDM = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{dj}}{C_{rj}} \tag{2}$$

where,

- N: Total No. of nodes.
- T_{si}: Total time, node i experiencing full utilization.
- T_{ai}: Total time, node i in active state.
- M: Total No. of VMs.
- C_{dj}: Estimation of performance degradation of VM j due to migration.
- C_{rj}: CPU capacity requested by VM j during its lifetime.

From both of the above, SLAV (Beloglazov and Buyya, 2012) is defined by Eq. (3):

$$SLAV = SLATAN \times PDM \tag{3}$$

Table 3
Power consumption (KwH) of selected server under different load in % Beloglazov and Buyya (2012).

Server (%)	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant G4(KwH)	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5(KwH)	93.7	97	101	105	110	116	121	125	129	133	135

Please cite this article in press as: Shrimali, B., Patel, H. Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment. Journal of King Saud University – Computer and Information Sciences (2017), https://doi.org/10.1016/j.jksuci.2017.12.001

Host Utilization U_H (Beloglazov and Buyya, 2012) is defined as

$$U_H = \frac{\sum_{i=1}^{n}(U_{vmi} \times C_{vmi})}{C_H} \tag{4}$$

where,

- n: Total No. of VM on a host.
- C_H: Host capacity in MIPS, which is defined as Eq. (5)
- U_{vmi}: VM utilization
- C_{vmi}: Total VM capacity in MIPS.

$$C_H = Number\ of\ Core \times Individual\ Core\ Capacity \tag{5}$$

Energy Consumption by host E_H (Lee and Zomaya, 2012) is defined as

$$E_H = (P_{max} - P_{min}) \times U_H + P_{min} \tag{6}$$

where, P_{max} = Power consumption at peak load,

P_{min} = Minimum Power consumption of active mode (having utilization equal to lower threshold e.g. utilization = 30%),

U_H is taken from (4).

3.2. VM allocation problem

Problem Statement: Given a set of host H and set of request R, it is required to allocate a request R_i to host H_j such that energy consumption is reduced and SLA violations are minimized.

4. Proposed algorithm

The functional block diagram of overall system is as shown in Fig. 3. Requests submitted by end users are sent to Cloud data centers. Numerous nodes make data center. Every node has either Node Controller (NC) or Cluster Controller (CC) fit into it. Working of NC and CC is mentioned beneath:

4.1. Node controller

The role of node controller is to monitor and control utilization of hosts. It is responsible for setting up dynamic threshold values which are used to identify the over-utilized or under-utilized hosts based on utilization values. It has two components: viz. (i) Monitoring component (MC) and (ii) Decision Maker (DM).

4.1.1. Monitoring component (MC)

The role of MC is to continuously calculate and monitor the utilization of host (U_H) using Eq. (4). Utilization of node at every hour is calculated for 24 h. As we recommended the host with moderate utilization for VM allocation, it is required to bifurcate all the hosts in data center into those three categories. For the same, two dynamic threshold values (i) Upper threshold (U_{th}) (ii) Lower threshold (L_{th}) are generated by NC. Unlike static threshold values, dynamic threshold values are suitable for heterogeneous Cloud environment. An upper utilization threshold value is calculated using the technique proposed in Beloglazov and Buyya (2012) as shown in Eq. (7) and (8).

$$MAD = median_i(|X_i - median_j(X_j)|), \tag{7}$$

$$U_{th} = 1 - S \times MAD \tag{8}$$

where $S \in R^+$ (R is a set of positive real numbers), S is a safety parameter which allows the adjustment for efficiency of method.

Based on this upper threshold value, lower utilization threshold value is calculated as shown in Eq. (9).

$$Ł_{th} = \begin{cases} U_{th} - S \times MAD & \text{if } MAD \leqslant 20 \\ \frac{U_{th}}{MAD} & otherwise \end{cases} \tag{9}$$

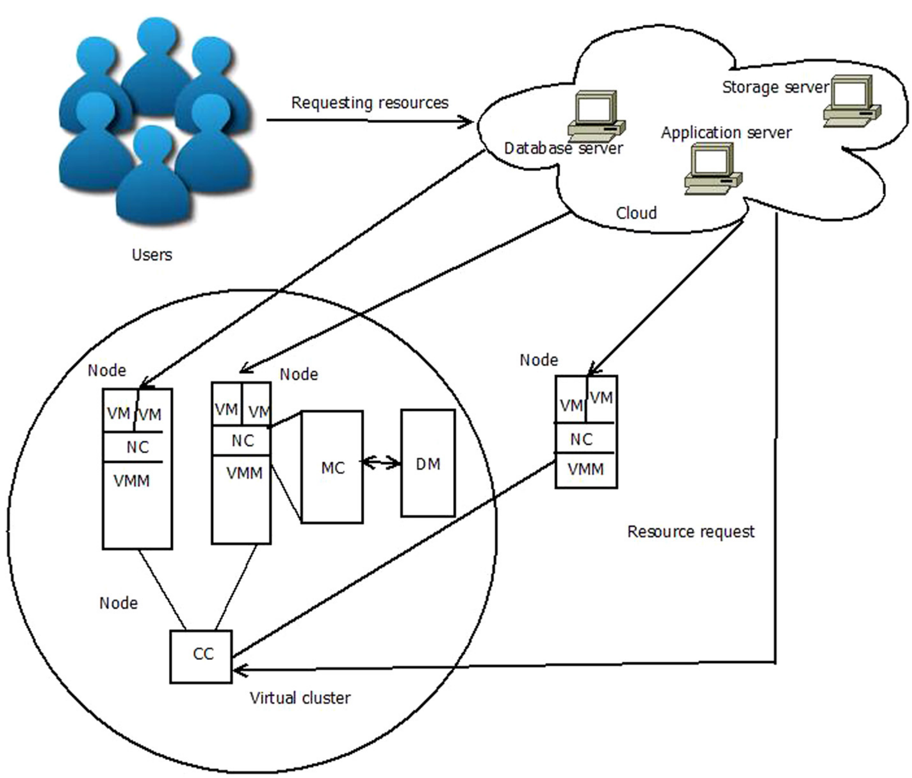

Fig. 3. System architecture.

Please cite this article in press as: Shrimali, B., Patel, H. Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment. Journal of King Saud University – Computer and Information Sciences (2017), https://doi.org/10.1016/j.jksuci.2017.12.001

4.1.2. Decision maker (DM)

The role of DM is to analyze the node regularly and pass the status to CC. DM sends the status of stable nodes to CC. Every time when resources are allocated or released and utilization of node is changed, it sends message/performance of node to CC.

4.2. Cluster controller

Cluster Controller (CC) is a part of host which creates cluster of a moderate hosts. The role of cluster controller is as under:

1. Creation of virtual cluster: The aim is to create a virtual cluster where the hosts are arranged in decreasing order of their utilization, to satisfy the requirement of energy efficiency.
2. Selection of a node: It selects the host using multi-objective optimization technique so that allocation becomes energy and performance efficient.

CC repeats the process continuously.

4.3. Multi-objective optimization based allocation

Considering the performance and energy efficiency as main criterion for allocation, many algorithms have been introduced, analyzed and compared in Shrimali and Patel (2015). Its experimental result shows that there is major trade-off between performance and energy. To reduce the trade-off, the multi-objective optimization based allocation policy is introduced in our work. It optimally selects the pair of Request- Host based on the proposed algorithm. The problem can be formulated as follows:

$$Minimize : \sum_{i=1}^{n} \mathbf{Pf_i}(\mathbf{R}, \mathbf{H}) \tag{10}$$

$$\text{subject to} : p1, p2, ..pn \in P \mid \sum_{j=1}^{n} P_j = 1 \tag{11}$$

where, P is the preference of objective.

Proposed Method: Here, we have used the weighted sum method, a type of apriori articulation of preferences method of multi-objective optimization, in which the weights of the objective functions are decided in advance by the user. The basic idea of the proposed method is as follows.

Let $f_1(x)$ be the function of performance and $f_2(x)$ be the function of energy efficiency. Since our objective is to minimize performance overhead by controlling SLA violations and power consumption, we need to minimize their linear combination F(x) is defined as follows.

$$F(x) = \lambda \times f_1(x) + (1 - \lambda) \times f_2(x) \tag{12}$$

where, λ is a weightage value set by Cloud service provider, $0 \leqslant \lambda \leqslant 1$ and x is a decision variable. Here, request (R_i) and host (H_j) are the decision variables on which the functions are evaluated. Hence, the function can be rewritten as:

$$Cal(r, h) = \lambda \times f_{energy}(Ri, Hj) + (1 - \lambda) \times f_{perfo}(Ri, Hj) \tag{13}$$

where, $f_{energy}(R_i,\ H_j)$ is defined by Lee and Zomaya (2012);

$$f_{energy}(Ri, Hj) = (P_{max} - P_{min}) \times U_H + P_{min} \tag{14}$$

and referring to Eqs. (2) and (1), $f_{perfo}\ (R_i,\ H_j)$ is defined by Beloglazov and Buyya (2012);

$$f_{perfo}(Ri, Hj) = SLAV = SLATAN \times PDM \tag{15}$$

The algorithm has two phases. In the first phase, the algorithm goes through the process of checking request queue and set of stable hosts(H). It further applies the energy and performance function on the selected request and server pair. New matrix *cal* is generated by multiplying preference value λ to both function. Note that the value of λ depends on the Cloud service provider which lies between 0 and 1. Identify the minimum value from metric Cal, which will be efficient request-host pair. Algorithm Selection of request-server pair using MOOA finds out the value based on request and host pair $(R_i,\ H_j)$ is as follows: Firstly, it checks the availability of request in request queue and accordingly it checks each (request,host) pair.

Algorithm 1 Selection of request, host pair using MOOA (request *R*, Host *H*)

```
if moderateHostList is NOT NULL then
  min ← MIN
  for each R in requestQueue do
      for each H in moderateHostList do
          Cal(req, s) = λ × f_energy(R_i, H_j) + (1 − λ)
          × f_perfo(R_i, H_j)(13)
      if (cal(R,H)<min) then
              Min ← cal(i,j)
              req ← i
              host ← j
          end if
      end for
  end for
      return(req, host)
else
  for each H in hostList do do
    if H.getUtil lies within UpperThreshold and
    LowerThreshold then
      moderateHostList.add(H)
    end if
  end for
  arrange the moderateHostList in decreasing
  order of their utilization
end if
```

4.4. Multi-objective allocation: Analysis

Initially, when the request is available on request queue, it checks the availability of average host. In the first case when average host is available, the total number of iterations will be $lambda \times n^2 \times nvm + (1 - \lambda) \times n^2 \times nvm \times nh$.

Hence, here $\lambda$is constant, it takes $O(n^2 \times nvm) + O(n^2 \times nvm \times nh)$ time. In the second case, when hostList is empty, it will check the utilization of each node and hence, the total number of iterations will be $nh.IttakesO(nh)$ time.

5. Experimental evaluation

In this section, initially we start with sample evaluation, which describes the evaluation scenario with the request and host pair. Further, we demonstrate the simulation scenario with the help of experimental results.

5.1. Sample evaluation

To perform sample evaluation on VM request allocation, we consider 3 requests and 3 hosts with configurations as mentioned in Table 4 and Table 5 respectively. Table 6 depicts evaluation of different functions used in our proposed algorithm for all possible combinations of given request and host pairs. As shown in Table 6, tuple 3 for R1 and tuple 5 and 9 for R2 and R3 respectively, depict that it chooses the request-host pair having the minimum energy

Please cite this article in press as: Shrimali, B., Patel, H. Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment. Journal of King Saud University – Computer and Information Sciences (2017), https://doi.org/10.1016/j.jksuci.2017.12.001

Table 4
Request configuration.

Request Id	No of cores	RAM (MB)	Cd (estimated perfo. Degradation)	CPU (MIPS)	Uti
R1	1	1740	0.26	2500	20
R2	2	1740	0.5	2000	30
R3	4	512	0.67	2000	50

Table 5
Host configuration.

HostId	No of cores	RAM (MB)	Pmax (watts)	Pmin (watts)	Tsi	Tai	Uti	CPU capacity
H1	2	4096	170	105	30	24	60	1000
H2	4	4996	169	130	20	40	40	1500
H3	4	4996	120	100	40	8	80	2500

consumption and less SLA violations. Hence, their linear combination presented by function *Cal*, gives minimum value. So, for the considered host-request pair, R1 and R2 are allocated to H3 and R3 is allocated to H2. We can analyze from the Table 6 that it optimizes the resource allocation by selecting the host with the minimum value of both the parameters.

5.2. Simulation scenario

We simulated our algorithm for performance and energy-efficient resource allocation. Simulation approach is used to repeat the experiments under an analogous environment. Thus, the comparison of different scheduling strategies is possible. The CloudSim (Calheiros et al., 2011) has been chosen as a simulation platform since it allows the demonstration of virtualized environments with on-demand resource provisioning and management. In our simulation, we have used 800 power host of two types namely HpProLiant G4 and HpProLiant G5. Also, to use workload traces collected from a real system, PlanetLab (Chun et al., 2003) workload is used, that consist different readings of CPU utilization collected at interval of 5 min of VMs of different host scattered around the world. We have considered random total 288 readings of different VMs for 24 h of different host. PlanetLab (Chun et al., 2003) workload covered readings of around 5000 virtual machines. In our experiments, we consider physical machine with multi-core having same core capacity for each core. VMs considered to be a resource request type, and the request characteristics are consider as the attributes of VM, which include the.

1. Processing capacity of computing node in Millions of Instructions Per Second (MIPS).
2. Main memory(RAM) in Mega Bytes (MBs).
3. Secondary storage in Mega Bytes (MBs)/ Giga bytes(GB)/Tera bytes(TB).

The configuration of physical machine/node and VMs are as shown in Table 7 and Table 8 respectively. Workload data characteristics is shown in Table 9.

5.3. Experimental comparisons

We have compared our multi-objective optimization technique against five other existing techniques namely Round Robin (Sempolinski and Thain, 2010),Watts Per Core(WPC) (Raycroft et al., 2014), Limited look ahead control (LLC) (Kusic et al., 2009), Non-Power Aware policy (NPA) (Beloglazov and Buyya, 2010b) and Dynamic Voltage and Frequency Scaling (DVFS) (Guérout et al., 2013). Description of these techniques are as follows:

1. Round Robin (Sempolinski and Thain, 2010): Round Robin is selected for the experiments as it is the default scheduling policy in the Eucalyptus (Sempolinski and Thain, 2010). Resource requests are assigned to host in a sequence based on the free resource availability of host. For every resource request, the policy iterates through the hosts sequentially, starting from where it is left, and again chooses the first host that can serve the virtual machine.
2. WPC: Watts per core is identified as a most energy efficient technique in the previous available literature (Raycroft et al., 2014). Hence, it is included in our experiments. In this strategy, the host using the least additional wattage per CPU core to perform task based on each host's power supply is selected for VM request.
3. LLC (Kusic et al., 2009): The Limited Look Ahead Control (LLC) based framework is considered as base method and it is included here as it allows for multi-objective optimization under explicit operating constraints and is applicable to computing systems with non-linear dynamics where control inputs must be chosen from a finite set.
4. NPA (Beloglazov and Buyya, 2010b) and DVFS (Guérout et al., 2013): Two standard policies, NPA and DVFS is considered as these techniques adjusts the voltage and frequency of CPU according to current utilization.

Table 6
Evaluation Metric.

Tuple#	Request R	Host H	λ	fenergy (R,H)	fperfo (R,H)	λ^* fenergy (R,H)	$(1-\lambda)^*$ fperfor (R,H)	Cal (R,H)	Uti	SLATAN	PDM
1	R1	H1	0.5	3355	0.000014	16.5	0.000007	1677.50	50	0.41	0.000035
2	R1	H2	0.5	1170	0.000006	585	0.000003	585.00	33.33	0.16	0.000035
3	R1	H3	0.5	300	0.000058	150	0.000029	150	20	1.6	0.000035
4	R2	H1	0.5	3795	0.000035	1897.5	0.000017	1897.5	60	0.41	0.000083
5	R2	H2	0.5	1430	0.000014	715	0.000007	715.0	40	0.16	0.000083
6	R2	H3	0.5	380	0.00013	190	0.000069	190.0	24	1.6	0.000083
7	R3	H1	0.5	6395	0.000047	3197.5	0.000023	3197.5	100	0.41	0.00011
8	R3	H2	0.5	700	0.00018	350	0.000009	350.0	40	1.66	0.000111
9	R3	H3	0.5	2470	0.000186	1235	9.0.000093	1235.0	66.6	0.16	0.0001

Table 7
Physical node configuration.

Host types	HOST MIPS	HOST PES (No)	HOST RAM (MBs)	HOST BW (bps)	HOST STORAGE (MB/GB/TB)
PowerModelSpecPowerHpProLiantMl110G4Xeon3040	1860	2	4096	1000000	1000000
PowerModelSpecPowerHpProLiantMl110G5Xeon3075	2660	2	4096	1000000	1000000

Please cite this article in press as: Shrimali, B., Patel, H. Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment. Journal of King Saud University – Computer and Information Sciences (2017), https://doi.org/10.1016/j.jksuci.2017.12.001

Table 8
VM configuration.

VM Types	VM MIPS	VM PES (No)	VM RAM (MBs)	VM BW (bps)	VM STORAGE (MB/GB/TB)
1	2500	1	870	1000000	2500
2	2000	1	1740	1000000	2500
3	1500	1	1740	1000000	2500
4	500	1	613	1000000	2500

Table 9
Workload data characteristics (CPU utilization).

Workload	No of host	Average host shutdown
Random	50	20
PlanetLab	800	212

5.4. Analysis of result

Although several experiments are performed by varying the values of user's preference λ in Eq. (12), in this section, we discuss the only key results having the λ values showing the equal priority ($\lambda = 0.5$) to each objective. Experiments are run for multiple times and average values of result are considered as final results as shown in Tables 10. The comparisons of results are summarized in graph (4) and (5). Further, we generate random values of λ, to assign random priority to objectives.

5.4.1. Effect on data center energy consumption

Since one of our objectives is to minimize the power consumption of data center, we compare the utilization of 5 hosts included in the experiment and select the host with the average utilization. To identify the average utilized host, utilization is calculated for the period of 24 h. The simulation results presented in Table 10 show that selection of average utilized host brings higher energy saving compared with other allocation policies. It can be noticed that while applying MOOA policy for selection of (request, host) pair, it predicts the possible power consumption of host and accordingly selects the most appropriate host. Energy consumption is measured in average value of KW/hr over the simulation period of 24 h. From the Table 10, we can notice that using MOOA policy, power consumption is about 30.2 Kw/hr, the values are nearer to best energy efficient technique WPC. This clearly shows the importance of selection of appropriate host, which can result in efficient power consumption in data center. Fig. 4 depicts the comparisons of energy consumption between different policies. Fig. 5 depicts the energy consumption (in Kilo Watt per Hour) for various policies considered for our experimentation. It is concluded for the figure that though WPC performs 3.27% better than MOOA, it outperforms with other techniques by 0.43%, 15.76%, 398.94% and 1939.07% in RR, LLC, NPA and DVFS respectively. It is worth to note here that NPA and DVFS are non-power aware policies, but we have considered here to have exhaustive list in our benchmark policies. Further, LLC is a multi-objective optimization technique and hence it is worth comparing MOOA with LLC.

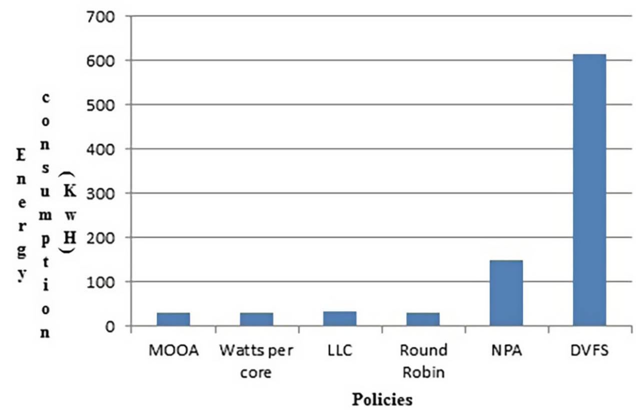

Fig. 4. Comparisons of different policies based on power consumption.

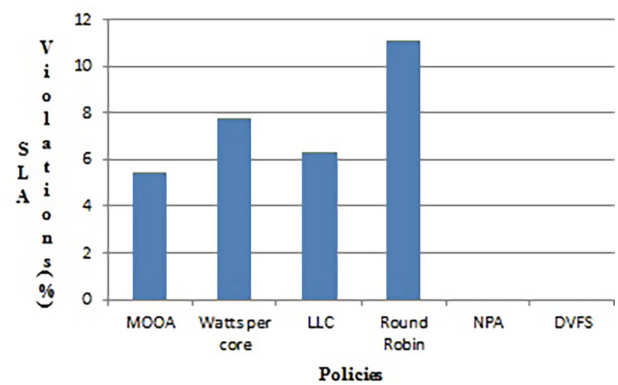

Fig. 5. Comparisons of different policies based on SLA violations (%).

5.4.2. Effect on performance

Another objective is to improve performance of the system by minimizing the SLA violations, which calculated by the number of SLA violation on active node and performance degradation. The performance of the MOOA policy quantifies by the number of SLA violations which results into the 5.41 percent. The comparisons are shown in Table 10. It depicts that MOOA based technique outperforms compare to other existing ones. The comparisons of policies are shown in Fig. 5.

6. Conclusion

We have implemented and validated a multi-objective optimization based allocation policy (MOOA) for performance and energy-efficient resource allocation in a virtualized computing environment. The problem formulation includes power consumption and SLA violations in the optimization problem. Experiment evaluation using Cloudsim environment shows that an allocation

Table 10
Simulation results of the different techniques.

Policy	SLA violations (Percent)	Energy Consumption (Kw/hr)	SLATAH	PDM (Percent)	VM migration
MOOA	5.41	30.2	9.81	0.26	4778
Watts per core	10.78	29.8	–	–	–
LLC	6.32	34.96	7.30	0.47	3345
Round robin	11.10	30.33	–	–	–
NPA	–	150.68	–	–	-
DVFS	–	615.8	–	–	-

Please cite this article in press as: Shrimali, B., Patel, H. Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment. Journal of King Saud University – Computer and Information Sciences (2017), https://doi.org/10.1016/j.jksuci.2017.12.001

using MOOA consumes, on average, 32% power and 5.41% SLA violations over a 24-h period with maintaining QoS requirements. Also, MOOA depicts 51% improvement in SLAV as compared to RR and shows 13% reduction in power consumption as compared to LLC. The policy is compared against three other techniques and proved better compared to them.

Future work may involve developing a fuzzy logic to generate a random decision variable for the same technique to study more effect on performance metrics, as well as a multi-objective optimization based more general VM consolidation technique.

References

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A., 2003. Xen and the art of virtualization. ACM SIGOPS Operat. Syst. Rev. 37 (5), 164–177.

Beloglazov, A., Buyya, R., 2010a. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers, Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science, Vol. 4. ACM.

Beloglazov, A., Buyya, R., 2010b. Energy efficient allocation of virtual machines in cloud data centers, 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid). IEEE, pp. 577–578.

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. Concurrency Comput. Practice Exp. 24 (13), 1397–1420.

Borja, S., Ruben, M., Ignacio, M., 2009. An open source solution for virtual infrastructure management in private and hybrid clouds. IEEE Internet Comput. 1, 14–22.

Buyya, R., Beloglazov, A., Abawajy, J., 2010. Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges, arXiv preprint arXiv:1006.0308.

Calheiros, R.N., Ranjan, R., Buyya, R., 2011. Virtual machine provisioning based on analytical performance and qos in cloud computing environments, 2011 International Conference on Parallel Processing (ICPP). IEEE, pp. 295–304.

Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A., 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.

Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., Bowman, M., 2003. Planetlab: an overlay testbed for broad-coverage services. ACM SIGCOMM Comput. Commun. Rev. 33 (3), 3–12.

Dabbagh, M., Hamdaoui, B., Guizani, M., Rayes, A., 2015. Energy-efficient resource allocation and provisioning framework for cloud data centers. IEEE Trans. Netw. Serv. Manage. 12 (3), 377–391.

Dong, J., Wang, H., Jin, X., Li, Y., Zhang, P., Cheng, S., 2013. Virtual machine placement for improving energy efficiency and network performance in iaas cloud, 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE, pp. 238–243.

Ghribi, C., Hadji, M., Zeghlache, D., 2013. Energy efficient vm scheduling for cloud data centers: Exact allocation and migration algorithms, 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). IEEE, pp. 671–678.

Guérout, T., Monteil, T., Da Costa, G., Calheiros, R.N., Buyya, R., Alexandru, M., 2013. Energy-aware simulation with dvfs. Simul. Model. Pract. Theory 39, 76–91.

Hameed, A., Khoshkbarforoushha, A., Ranjan, R., Jayaraman, P.P., Kolodziej, J., Balaji, P., Zeadally, S., Malluhi, Q.M., Tziritas, N., Vishnu, A., et al., 2016. A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. Computing 98 (7), 751–774.

Hamilton, J., Cooperative expendable micro-slice servers (cems): low cost, low power servers for internet-scale services. In: Conference on Innovative Data Systems Research (CIDR09) (January 2009), Citeseer, 2009, p. Nil.

URL,http://www.microsoft.com/hyper-v-server/en/us/default.aspx (2016).

URL,http://www.vmware.com/ (2016).

Jansen, R., Brenner, P.R., 2011. Energy efficient virtual machine allocation in the cloud, 2011 International Green Computing Conference and Workshops (IGCC). IEEE, pp. 1–8.

Jing, S.-Y., Ali, S., She, K., Zhong, Y., 2013. State-of-the-art research study for green cloud computing. J. Supercomput. 65 (1), 445–468.

Kimovski, D., Saurabh, N., Gec, S., Štefanic, P., Kecskemeti, G., Stankovski, V., Prodan, R., Fahringer, T., 2016. Towards an environment for efficient and transparent virtual machine operations: The entice approach, 2016 5th IEEE International Conference on Cloud Networking (Cloudnet). IEEE, pp. 242–247.

Kivity, A., Kamay, Y., Laor, D., Lublin, U., Liguori, A., kvm: the linux virtual machine monitor. In: Proceedings of the Linux Symposium, vol. 1, 2007, pp. 225–230.

Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N., Jiang, G., 2009. Power and performance management of virtualized computing environments via lookahead control. Cluster Comput. 12 (1), 1–15.

Lee, Y.C., Zomaya, A.Y., 2012. Energy efficient utilization of resources in cloud computing systems. J. Supercomput. 60 (2), 268–280.

Lin, W., Wang, J.Z., Liang, C., Qi, D., 2011. A threshold-based dynamic resource allocation scheme for cloud computing. Procedia Eng. 23, 695–703.

Lin, X., Liu, Z., Guo, W., 2015. Energy-efficient vm placement algorithms for cloud data center. In: Cloud Computing and Big Data. Springer, pp. 42–54.

Mann, Z.Á., 2015. Allocation of virtual machines in cloud data centersa survey of problem models and optimization algorithms. ACM Comput. Surv. (CSUR) 48 (1), 11.

Marler, R.T., Arora, J.S., 2004. Survey of multi-objective optimization methods for engineering. Struct. multidiscip. Optim. 26 (6), 369–395.

Masdari, M., Nabavi, S.S., Ahmadi, V., 2016. An overview of virtual machine placement schemes in cloud computing. J. Netw. Comput. Appl. 66, 106–127.

Mell, P., Grance, T., 2009. The nist definition of cloud computing. Natl. Inst. Stand. Technol. 53 (6), 50.

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D., 2009. The eucalyptus open-source cloud-computing system, CCGRID'09. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009. IEEE, pp. 124–131.

Panda, S.K., Jana, P.K., 2015. A multi-objective task scheduling algorithm for heterogeneous multi-Cloud environment, 2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV). IEEE, pp. 82–87.

Panigrahia, P., Pandab, S.K., Tripathyc, C.R., 2015. Energy efficient task consolidation algorithms for cloud computing systems. Int. J. Inf. Process. 9 (4), 34–45.

Piao, J.T., Yan, J., 2010. A network-aware virtual machine placement and migration approach in cloud computing, 2010 9th International Conference on Grid and Cooperative Computing (GCC). IEEE, pp. 87–92.

Quang-Hung, N., Thoai, N., Son, N.T., 2014. Epobf: Energy efficient allocation of virtual machines in high performance computing cloud. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems XVI. Springer, pp. 71–86.

Raycroft, P., Jansen, R., Jarus, M., Brenner, P.R., 2014. Performance bounded energy efficient virtual machine allocation in the global cloud. Sustainable Comput. Inf. Syst. 4 (1), 1–9.

Sempolinski, P., Thain, D., 2010. A comparison and critique of eucalyptus, opennebula and nimbus, 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom). IEEE, pp. 417–426.

Shojafar, M., Cordeschi, N., Baccarelli, E., 2016a. Energy-efficient adaptive resource management for real-time vehicular cloud services. IEEE Trans. Cloud Comput.

Shojafar, M., Canali, C., Lancellotti, R., Abawajy, J., 2016b. Adaptive computing-plus-communication optimization framework for multimedia processing in cloud systems. IEEE Trans. Cloud Comput.

Shrimali, B., Patel, H., 2015. Performance based energy efficient techniques for vm allocation in Cloud environment, Proceedings of the Third International Symposium on Women in Computing and Informatics. ACM, pp. 477–486.

Sosinsky, B., 2010. Cloud Computing Bible, 762. John Wiley & Sons.

Van, H.N., Tran, F.D., Menaud, J.-M., 2010. Performance and power management for cloud infrastructures, 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD). IEEE, pp. 329–336.

Xiao, Z., Song, W., Chen, Q., 2013. Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Trans. Parall. Distribut. Syst. 24 (6), 1107–1117.

Xiong, K., Perros, H., 2009. Service performance and analysis in cloud computing, 2009 World Conference on Services-I. IEEE, pp. 693–700.

Zamanifar, K., Nasri, N., Nadimi-Shahraki, M., 2012. Data-aware virtual machine placement and rate allocation in Cloud environment, 2012 Second International Conference on Advanced Computing & Communication Technologies (ACCT). IEEE, pp. 357–360.

Zhao, Z., Taal, A., Jones, A., Taylor, I., Stankovski, V., Vega, I.G., Hidalgo, F.J., Suciu, G., Ulisses, A., Ferreira, P., et al., 2015. A software workbench for interactive, time critical and highly self-adaptive cloud applications (switch), 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). IEEE, pp. 1181–1184.

Zitzler, E., Thiele, L., 1998. Multiobjective optimization using evolutionary algorithmsa comparative case study, International Conference on Parallel Problem Solving from Nature. Springer, pp. 292–301.

Please cite this article in press as: Shrimali, B., Patel, H. Multi-objective optimization oriented policy for performance and energy efficient resource allocation in Cloud environment. Journal of King Saud University – Computer and Information Sciences (2017), https://doi.org/10.1016/j.jksuci.2017.12.001