Indian Journal of Science and Technology, *Vol 10(32), DOI: 10.17485/ijst/2017/v10i32/113054, August 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Energy-Efficient Load-Aware VM Placement using Multi-Metrics Analysis

Narander Kumar* and Swati Saxena

Department of Computer Science, Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow – 2260252, Uttar Pradesh, India; swatesaxena@gmail.com, nk_iet@yahoo.co.in

Abstract

Objective: Cloud data centres comprise of a huge number of physical machines which host a number of heterogeneous virtual machines. These VMs exhibit fluctuating behaviour in terms of resource usage and specifications leading to disproportionate resource utilization in physical machines. This load imbalance on a host degrades its performance and violates many SLA features. To maintain even load in a data centre, an effective virtual machine placement technique is required. This paper proposes an efficient VM placement and migration technique based on three-tier architecture and considers load as a prime objective. **Method:** To maintain an even-load in a data centre, many load balancing techniques have been suggested which consider either migration cost or heterogeneity as the overall objective. This paper uses a multi-metrics analysis to distribute VMs evenly and maintains a stable equilibrium inside a data centre. We have applied Analytical Hierarchy Process (AHP) for efficient virtual machine placement using four post placement metrics which defines some of the key Service Level Agreement (SLA) parameters. The metrics considered for placement and migration are all load-centric and promise fewer migrations and SLA violations. **Findings:** Simulation results show a remarkable reduction in migrations which improves energy conservation inside the data centre. Application of AHP in balancing a data centre's load is still unexplored. The presented placement technique selects the best candidate machine for placement, hence upgrades the performance. **Improvement:** Further enhancement includes adding more metrics for placement to fine tune performance and ensure better resource utilization along with reduction in overall energy consumption of a data centre.

Keywords: Analytical Hierarchy Process, Energy Efficiency, Load Balance, Multi-Metrics Placement, Service Level Agreement

1. Introduction

Physical Machines (PMs) in a data centre symbolize the real resources used for computation such as CPU, memory, I/O, etc. Virtual Machines (VMs) represent applications or tasks of cloud users. Figure shows a typical cloud computing layout where multiple cloud users submit their tasks to a data centre. Virtualization technology enables a single PM to host multiple VMs with varied resource requirements, thereby, providing high resource utilization and better user satisfaction. It also helps a cloud data centre to tackle dynamic and fluctuating service demands of users in a flexible and efficient manner. However, sometimes multiple heterogeneous VMs with variable and unstable workloads may trigger an imbalance of resource usage in a host PM. This load disparity negatively affects the performance of a data centre and violates SLA metrics. A common load scenario in a data centre involves unsteady arrival of multiple user-service requests with non-uniform resource demands. These diverse workloads consume a host PM's resources in a disproportionate manner resulting in its performance deterioration. Whether it is a single VM overwhelming its host PM or a group of VMs, an unbalanced host will adversely affect its guest VMs and the overall performance of its data centre will degrade. To avoid such

**Author for correspondence*

situations, load balancing mechanisms are deployed in cloud data centres which ensure optimum resource utilization without over-burdening PMs. Such mechanisms place VMs by carefully analyzing the existing load of PMs to select the best possible hosts and ensuring that the resources are consumed in the most favourable manner (VM placement). In situations of excess load experienced by any PM, load balancing algorithms re-distribute the VMs (excess workload) to ease out an over-burdened host (VM migration). Figure 1 shows the relation between cloud users, VMs and PMs.

The execution of load balancing techniques takes place at two levels- at application level it becomes the responsibility of application scheduler whereas at VM level, VM manager takes care of implementing balancing techniques. It is an NP-hard problem with approximate solutions. A general outline of existing load balancing techniques is shown in Table 1.

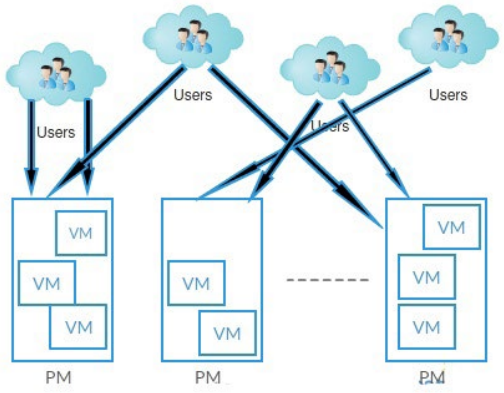

Figure 1. A typical data centre scenario.

In this paper, we propose a 2-phase load balancing technique implemented at VM level. In the first phase, we apply Analytical Hierarchy Process (AHP) to place an incoming VM at the best possible host PM. This phase is termed as Load-Aware VM Placement. In the second phase, we use migration to ease the VM load of an over-burdened PM with the aim of energy-conservation and better user experience. Proposed technique considers heterogeneous VMs with multiple resource types and is implemented online in a dynamic fashion.

Analytical Hierarchy process is an analysis tool for multi criteria decision problems. It is a branch of operation research and has been applied extensively in many areas of planning and management. In cloud computing, however, it has not been used in an extensive manner. This paper utilizes its core idea to select the best host for a VM placement. Figure 2 outlines a general structure of AHP.

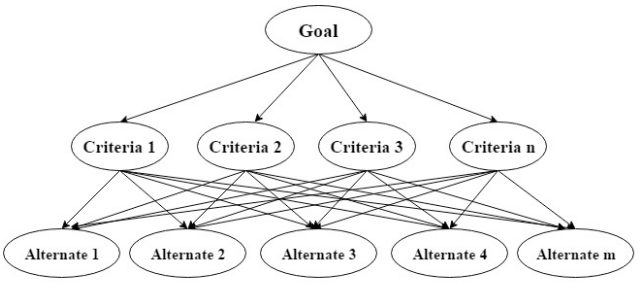

Figure 2. Analytical hierarchy process.

VM placement with the objective of load balancing in a data centre comes with its set of challenges. Virtualization has been in use since 1960s and it enhances the ability of a data centre to service multiple users in a shared environment[1]. However, this also leads to resources being used in a disproportionate manner in a data centre, usually due to a mismatch between a VM and its host[2]. This imbalance reduces the overall performance of a data centre and dishonours service level agreement (SLA) parameters[3]. To prevent such scenarios, load balancing techniques have been introduced which not only maintains an even resource utilization among the PMs, it also ensures scalability, better response time and sturdiness[4]. In this regard, a genetic solution is proposed[5] to map the VMs and PMs by balancing load and avoiding unnecessary migrations. It is implemented on a hierarchical structure and uses historical data to choose the best host for

Table 1. Categories of load balancing techniques

Category	Heuristic	Meta-heuristic	Hybrid
Techniques	Greedy Algorithms	Genetic Algorithms	Mix of heuristic and meta-heuristic
Features	Easy implementation, low time cost	Better load-balance results, high time cost	Increased implementation complexity

a VM. The downfall of this method is its complexity. An ant colony based optimized load balancing technique is presented[6] which achieves lesser migrations, better load balance and less SLA violations. However, it operates on a small number of hosts, thus, making it unsuitable for large data centres. A dynamic migration management technique[7] considers communication cost as a major parameter to reduce the number of over-loaded hosts, but considers homogeneous VMs which lower its practicality. Moreover, only CPU resource is considered as a host's load which further reduces its advantages. An initial VM placement method[8] maps VMs to PMs using probability approach and relieves hosts from over-loading. This technique provides useful results to balance the load but does not consider migration scenarios in a data centre.

A Dynamic and Integrated Resource Scheduling Algorithm[9] (DAIRS) is proposed which considers different resource types as integrated like CPU, memory and I/O. It is based on queues system which selects the front VM in the queue to place on a host. DAIRS, however, does not consider communication cost in a data centre to place a VM. On similar lines, load balancing algorithm considering various resources is presented[10] in real time environment. A hybrid placement algorithm is proposed[11] which balances the VMs while reducing the number of migrations. Based on 2 phases, this hybrid technique uses a heuristic approach to place VMs in the first phase and later optimizes their placements using a meta-heuristic algorithm. With the aim of servicing as many user requests as possible, an ant colony based VM load balancing method is presented[12] which considers multiple heterogeneous VMs and allocates them dynamically in a meta-heuristic strategy. It considers rejection of VMs which is not a possible scenario in data centres today. An offline VM allocation algorithm is suggested[13] based on resource reservation policy. It achieves promising results in balancing the load; however, its downfall is it being a static procedure and not well suited for practical purposes.

A distributed architecture based resource management technique is presented[14] that tightly couples node agents with PMs and uses a multi-criteria based placement strategy. It however, considers only usual criteria for comparing possible hosts and ignores SLA violations. A detailed technical report of their work is given[15]. Resource allocation techniques[15,16] has been suggested for non-virtualized data centres consisting of two layers of agents, namely, local and global. Based on separate analytical models, these works use resource demands prediction methods. Both[15,16] apply queuing theory with 16 going further and combining it with decomposition learning approach. Allocation work presented[17] introduces a commercial computing system called Unity for non-virtualized data centres based on layered architecture and utility model. Virtualization brought the resource utilization problem into the main stream with the 2-level agents getting encased into virtual machines. Research works outlined[18,19,20] use centralized methods of resource distribution and are applicable in virtual data centres. Migration overhead is given due weightage[21,22] and is used as an important parameter to elect the most suitable host for VM placement.

Rest of the paper is as follows: Section 2 presents the system architecture, followed by the VM placement technique and its simulation results in Section 3. Section 4 gives the conclusion and future perspectives.

2. System Architecture

The data centre topology considered in our proposed load balancing mechanism is 3-tier hierarchical switch-centric topology. This is the most common topology deployed in modern data centres and consists of network switches at three layers, namely, core, aggregate and access. We consider all the PMs under a single access layer switch to form one cluster. Thus the size of a cluster remains static with permanent PM members. Total number of clusters is determined by the number of access switches used. Every cluster has an intra-cluster module which monitors the load status of each member PM and places an incoming VM to the most suitable PM after applying the proposed load balancing mechanism. All the clusters are centrally monitored by an inter-cluster module which maintains a dynamic list of all clusters in non-decreasing order of their mean load. The VM demands of cloud users are forwarded by the inter-cluster module to a cluster with the least mean load. The intra-cluster module of this selected cluster forms the solution space by choosing member PMs whose resources availability matches the VM's demands. Further, AHP is applied to this solution space to select the best possible member PM to host the incoming VM. The detailed system architecture used for the proposed load-balancing mechanism is shown in Figure 3 and the notations and symbols used throughout in the paper are given in Table 2.

Table 2. Notations and symbols used

Symbol	Meaning	Symbol	Meaning
a_1	CPU availability of a PM	**m**	No. of PMs in a cluster
a_2	Memory availability of a PM	**t1**	CPU intensive VM
a_3	I/O availability of a PM	**t2**	Memory intensive VM
α	CPU demand of a VM	ML_C	Mean Load of a cluster C
β	Memory demand of a VM	$L(PM_r)$	Load on a PM r
γ	I/O demand of a VM	$A(PM_r)$	Resource availability on a PM r
l_1	CPU load on a PM r	$D(VM_g)$	Demand of a VM g
l_2	Memory load on a PM r	**SS**	Solution Space
l_3	I/O load on a PM r	$CP(PM_r)$	Total Capacity of PM r
$CM(PM_r)$	Capacity Makespan of a PM r	$t(VM_j)$	Execution time of VM j
$ART(PM_r)$	Average Response Time of a PM r	**RM**	Reciprocal Matrix

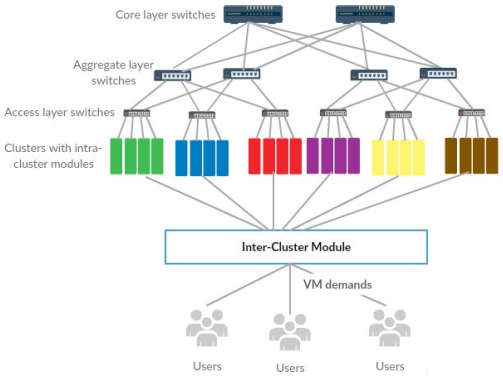

Figure 3. Proposed system architecture.

2.1 Load-Aware VM Placement

VM placement or the initial load distribution is the first phase of our proposed load balancing mechanism. In this phase, an incoming VM is forwarded to the lightest loaded cluster. Member PMs of this chosen cluster, whose resource availability is higher than the incoming VM's demand, are identified to form the solution space. This solution space is further reduced to contain only those PMs which are compatible with the type of incoming VM. After this reduction, multi-criteria based Analytical Hierarchy Process (AHP) is applied to the solution space and the best PM to host an incoming VM is selected. A detailed step-by-step procedure of load-aware VM placement is given below-

- As stated in the system architecture, all PMs are grouped into clusters based on their physical connectivity with the access level switches. An inter-cluster module present in the data centre maintains a dynamic list of these clusters in non-decreasing order of their mean load as shown in Figure 3. Mean Load of a cluster is calculated as

$$ML_{cl} = \frac{1}{m}\sum_{r=1}^{m} L\left(PM_r\right)$$

An incoming VM_g is forwarded to the first cluster in the list i.e. cluster with the minimum ML_c.

- An intra-cluster module, present in every cluster, compares the VM_g's demands with the availability of each member PM_r and builds up the solution space as

$$SS = \forall PM_r : A\left(PM_r\right) \rangle D\left(VM_g\right)$$

Member PMs with availability less than the VM's demand are removed from the solution space. Now we consider PMs whose availability is greater than the demand.

- Solution space is further reduced by considering PMs whose load characteristic is compatible with the VM type. For simplicity, we are considering only two types of VMs, namely CPU intensive and memory intensive. If the incoming VM is memory intensive (t2) then PMs with least memory load are considered for placement. Likewise, for CPU intensive VM (t1), PMs with least CPU

load are considered ideal host candidates. The directional cosines of VM_g will determine the type of VM as given below-

If cos α > cos β then VM_g is type t1 else type t2. Similarly, if $\cos l_1 > \cos l_2$, then PM_r has least memory load else PM_r has least CPU load.

- Now, we apply AHP to the remaining PMs in the solution space as shown in Figure 4.

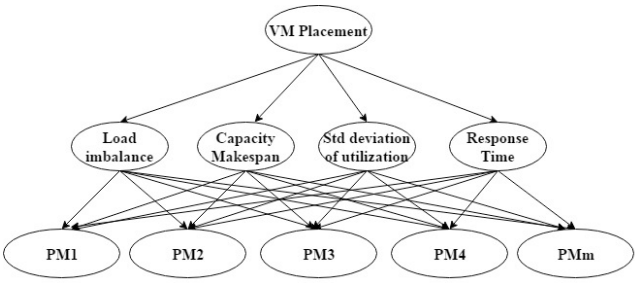

Figure 4. Analytical hierarchy VM placement process.

The core concept of AHP involves pair-wise comparison of alternatives for each criteria and then deducing their overall rankings. A similar concept is used in our proposed technique where VM placement is the goal. We have considered four post placement metrics to select the best host PM from the solution space for incoming VM_g. Figure 5 shows an example of VM placement using AHP while considering multi metrics. These metrics are load-centric SLA parameters and the aim of the proposed work is to honour their values in order to prevent any SLA violations. Candidate PMs in the solution space left after execution of step 3 are considered as alternatives for placement. We assume there are m such hosts. The value of each post placement metric is calculated by assuming that VM_g has been placed at PM_r. The description of these four metrics is given below-

Load Imbalance: Load imbalance of a PM_r after placing VM_g is the difference between its new load (after placing VM_g) and the mean load of its cluster. If the load of a candidate PM_r before placement was

$$L(PM_r) = l_1 i + l_2 j + l_3 k$$

And the incoming VM_g resource demand is

$$D(VM_g) = \alpha i + \beta j + \gamma k$$

Then the new load of PM_r after placing VM_g is

$$L_{new}(PM_r) = (l_1 + \alpha)i + (l_2 + \beta)j + (l_3 + \gamma)k$$

And the load imbalance (LI) of PM_r will be

$$LI(PM_r) = L_{new}(PM_r) - ML_{ci}$$

Like-wise, Load imbalance of each PM present in the solution space will be calculated assuming that the incoming VM is placed on it. The aim of proposed placement technique is to minimise the load imbalance.

Capacity Makespan: This metric combines the total demands of all VMs hosted on a PM with their execution times, i.e.

$$CM(PM_r) = \sum_{j=1}^{v} D(VM_j) * t(VM_j)$$

Here, v is the total number of VMs hosted on a PM before placement of VM_g. Now assume that VM_g is placed on PM_r then the new capacity makespan will be

$$CM_{new}(PM_r) = CM(PM_r) + [D(VM_g) * t(VM_g)]$$

Capacity makespan for each PM is calculated considering VM_g is placed on it. The aim of proposed placement technique is to minimise the capacity makespan.

Standard Deviation of Utilization: For each PM can be calculated as the square root of its load imbalance

$$SDU = \sqrt{LI(PM_r)}$$

Standard deviation should be the minimum for optimal placement.

Average Response Time: Time taken by VM_g to give its first response will vary from PM to PM depending on the already present load of that PM. Hence, this value will be calculated for each PM considering VM_g is placed on it.

$$ART(PM_r) = \frac{1}{v} \sum_{j=1}^{v} RT(VM_j)$$

The present load-aware VM placement can now be seen as-

Let's understand the applied technique in the following steps using an example.

i. The first step is to prioritize the post placement criteria using a square matrix, called Reciprocal Matrix (RM), as shown in Figure 6.

Values in the matrix are provided by the cloud service provider and are in the range of 1 to 9 with the following meanings[23] –

Values 2, 4, 6 and 8 are intermediate values. If rm_{ij} value is p in the RM matrix then the value of rm_{ji} will be

1/p Figure 6. Next, we compute eigen vector of RM[4] to obtain the ranking of post placement metrics as shown in Figure 7.

ii. Now, we construct four square matrices each for one post placement metric and all the alternatives. In the example, we have taken 5 alternatives or candidate PMs, with ids, PM_5, PM_7, PM_{12}, PM_{15}, PM_{19} and the matrices are shown in Figure 8.

Values in the load imbalance matrix are the values of post placement load imbalance LI for each candidate PM. Then we compute eigen vector of LB[5]. Similarly, we construct CM[5], SDU[5], RT[5] and compute eigen vectors of each as shown in Figure 8. This step gives the ranking of each alternative PM for each post placement criteria.

iii. In the last step, eigen vector calculated in step i is multiplied by a square matrix of eigen vectors calculated in step ii. Their product will give the final ranking of all the candidate/alternate PMs as shown in Figure 9.

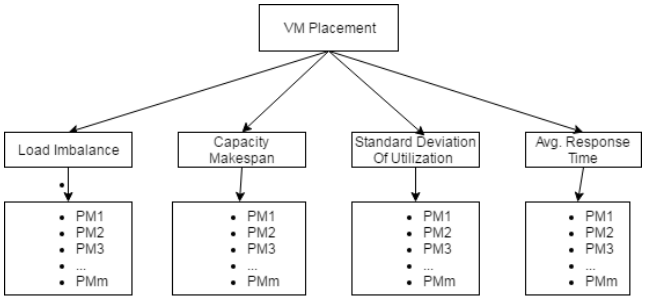

Figure 5. Example of AHP based VM placement.

$$\begin{array}{c} \\ LI \\ CM \\ SDU \\ RT \end{array} \begin{array}{c} \begin{array}{cccc} LI & CM & SDU & RT \end{array} \\ \begin{bmatrix} 1 & 4 & 3 & 5 \\ \frac{1}{4} & 1 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 3 & 1 & \frac{1}{2} \\ \frac{1}{5} & 3 & 2 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} 0.8983 \\ 0.1215 \\ 0.2593 \\ 0.3306 \end{bmatrix}$$

Figure 6. Reciprocal matrix.

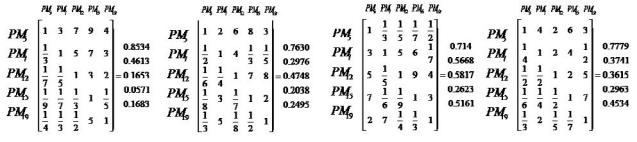

Figure 7. The reciprocal matrix.

$$\begin{bmatrix} 0.8534 & 0.7630 & 0.7140 & 0.7779 \\ 0.4613 & 0.2976 & 0.5668 & 0.3741 \\ 0.1653 & 0.4748 & 0.5817 & 0.3615 \\ 0.0571 & 0.2038 & 0.2623 & 0.2963 \\ 0.1683 & 0.2495 & 0.5161 & 0.4534 \end{bmatrix} * \begin{bmatrix} 0.8983 \\ 0.1285 \\ 0.2593 \\ 0.3306 \end{bmatrix} = \begin{bmatrix} 1.3070 \\ 0.7233 \\ 0.4798 \\ 0.2435 \\ 0.4670 \end{bmatrix}$$

Figure 8. Candidate PMs rankings for each post placement metrics.

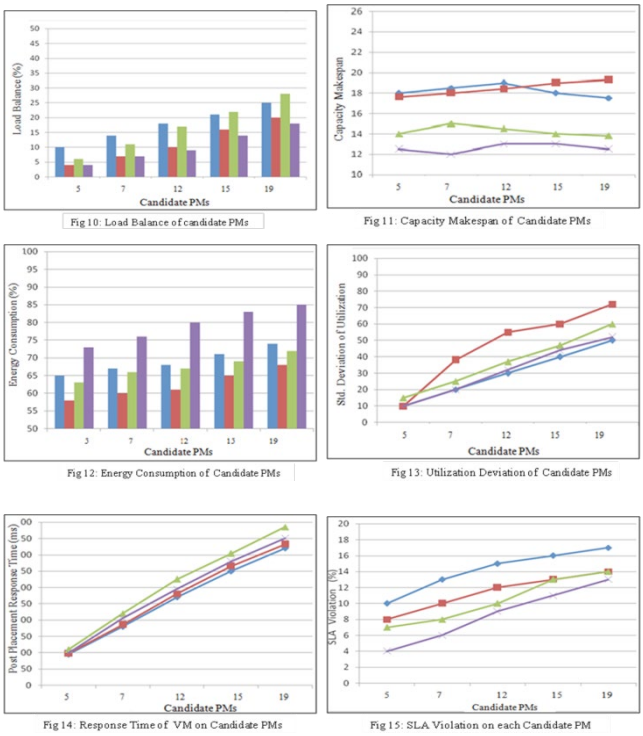

Figure 9. Final ranking of candidate PMs.

3. Results and Discussion

The proposed energy efficient load-aware VM placement technique is simulated on CloudSim simulator. The simulation setup consist of a cloud data centre with 100 PMs and 150 VMs and the performance was evaluated w.r.t. load balance, resource utilization, makespan, response time and energy consumption. In Figure 10 to 15, these post placement metrics are shown graphically for 5 candidate PMs namely, PM_5, PM_7, PM_{12}, PM_{15} and PM_{19}. These PMs surpassed the availability and type testing discussed in Steps i to iii of the proposed placement technique.

It is quite clear from the results that candidate PM_5 scores the highest in terms of post placement evaluation; hence it is chosen the host for an incoming VM. As the

load in the data centre reaches a stable state, instances of migration reduces and therefore energy consumption also stabilizes (Figure 12). An optimum utilized host gives its best performance which leads to a reduction in overall SLA violations; same is shown in Figure 15.

Simulation results indicate that our proposed placement technique is promising in terms of stable energy consumption, less performance violations and optimum resource utilization.

4. Conclusion and Future Scope

This paper proposes a novel VM placement technique based on multiple SLA metrics and uses a hierarchical approach to rank the PMs in the order of their preference to host VM. Decision making based on multiple attributes is a popular and successful technique and is applied in many crucial areas. However, in cloud computing, this technique is still unexplored. Our attempt is to apply its core idea in a data centre with the aim to maximize total resource utilization and prevent overwhelming of a host due to uneven loads by placing a VM in the best suitable PM. Simulation results of the presented technique shows promising results and effectively reduces number of SLA violations besides preventing peak energy consumptions. In the future, the same technique can be fine tuned to include more metrics for VM placement and shall include running optimization of VMs covering migration scenarios.

5. References

1. Singh A, Korupolu M, Mohapatra D. Server-storage virtualization: Integration and load balancing in data centers. Proceedings of the 2008 ACM/IEEE Conference on Supercomputing IEEE Press; 2008. Crossref
2. Randles M, Lamb D, Taleb-Bendiab A. A comparative study into distributed load balancing algorithms for cloud computing. Advanced Information Networking and Applications Workshops (WAINA) 2010 IEEE 24th International Conference on IEEE; 2010. p. 551–6. Crossref
3. Garcia JOG, Nafarrate AR. Agent-based load balancing in cloud data centers. Cluster Computing. 2015; 18(3):1041–62. Crossref
4. Kansal NJ, Chana I. Cloud load balancing techniques: A step towards green computing. IJCSI International Journal of Computer Science Issues. 2012; 9(1):238–46.
5. Hu J, Gu J, Sun G, Zhao T. A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. 2010 3rd International symposium on parallel architectures, algorithms and programming IEEE; 2010. p. 89–96.
6. Wen WT, Wang CD, Wu DS, Xie YY. An aco-based scheduling strategy on load balancing in cloud computing environment. 2015 9th International Conference on Frontier of Computer Science and Technology IEEE; 2015. p. 364–9.
7. Song X, Ma Y, Teng D. A load balancing scheme using federate migration based on virtual machines for cloud simulations. Mathematical Problems in Engineering. 2015; 2015.
8. Ni J, Huang Y, Luan Z, Zhang J, Qian D. Virtual machine mapping policy based on load balancing in private cloud environment. 2011 International Conference on IEEE Cloud and Service Computing (CSC); 2011. p. 292–5.
9. Tian W, Zhao Y, Zhong Y, Xu M, Jing C. A dynamic and integrated load-balancing scheduling algorithm for cloud datacenters. 2011 IEEE International Conference on Cloud Computing and Intelligence Systems IEEE; 2011. p. 311–5. Crossref PMid:21798070 PMCid:PMC3176265
10. Xu M, Tian W. An online load balancing scheduling algorithm for cloud data centers considering real-time multidimensional resource. 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems IEEE; 2012. p. 264–8.
11. Thiruvenkadam T, Kamalakkannan P. Energy efficient multi dimensional host load aware algorithm for virtual machine placement and optimization in cloud environment. Indian Journal of Science and Technology. 2015; 8(17):1–11. Crossref
12. Cho KM, Tsai PW, Tsai CW, Yang CS. A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing. Neural Computing and Applications. 2015; 26(6):1297–309. Crossref
13. Tian W, Xu M, Chen Y, Zhao Y. Prepartition: A new paradigm for the load balance of virtual machine reservations in data centers. 2014 IEEE International Conference on Communications (ICC); 2014. p. 4017–22. Crossref
14. Yazir YO. Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis. 2010 IEEE 3rd International Conference on Cloud Computing; Miami FL. 2010. p. 91–8.
15. Yazır YO, Matthews C, Farahbod R, Guitouni A, Neville S, Ganti S, Coady Y. Dynamic and autonomous resource management in computing clouds through distributed multi criteria decision making. University of Victoria Department of Computer Science. 2010; 91–8.

16. Bennani MN, Menasce DA. Resource allocation for autonomic data centers using analytic performance models. ICAC '05, Proceedings of the 2nd International Conference on Automatic Computing; 2005. p. 229–40. Crossref
17. Tesauro G. Online resource allocation using decompositional reinforcement learning. AAAI'05, Proceedings of the 20th National Conference on Artificial Intelligence AAAI Press; 2005. p. 886–91.
18. Das R, Kephart J, Whalley I, Vytas P. Towards commercialization of utility-based resource allocation. ICAC '06, IEEE International Conference on Autonomic Computing; 2006. p. 287–90. Crossref
19. Bobroff N, Kochut A, Beaty K. Dynamic placement of virtual machines for managing SLA Violations. IM '07, 10th IFIP/IEEE International Symposium on Integrated Network Management; 2007. p. 119–28.
20. Wang X, Lan D, Wang G, Fang X, Ye M, Chen Y, Wang Q. Appliance-based autonomic provisioning framework for virtualized outsourcing data center. Fourth International Conference on Autonomic Computing 2007, ICAC '07; 2007. p. 29–29. Crossref
21. Kochut A. On impact of dynamic virtual machine reallocation on data center efficiency. IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008; 2008. p. 1–8. Crossref
22. Van HN, Tran FD. Autonomic virtual resource management for service hosting platforms. ICES '09, Proceedings of the International Conference on Software Engineering Workshop on Software Engineering Challenges of Cloud Computing. IEEE Computer Society. 2009. p. 1–8. PMid:19133138 PMCid:PMC2628941
23. Saaty RW, The analytic hierarchy process- what it is and how it is used. Mathematical Modelling. 1987; 9(3):161–76. Crossref