HAL
open science

Energy-Efficient Algorithm for Load Balancing and VMs Reassignment in Data Centers

Nabila Djennane, Rachida Aoudjit, Samia Bouzefrane

To cite this version:

Nabila Djennane, Rachida Aoudjit, Samia Bouzefrane. Energy-Efficient Algorithm for Load Balancing and VMs Reassignment in Data Centers. FiCloud Workshops, IEEE, Aug 2018, Barcelone, Spain. pp.225-230, 10.1109/W-FiCloud.2018.00043 . hal-03181645

HAL Id: hal-03181645

https://hal.science/hal-03181645v1

Submitted on 25 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Energy-Efficient Algorithm for Load Balancing and VMs Reassignment in Data Centers

Nabila Djennane
LARI Lab,
University Mouloud Mammeri
Tizi-Ouzou, Algeria
djennanenabila@gmail.com

Rachida Aoudjit
LARI Lab,
University Mouloud Mammeri
Tizi-Ouzou, Algeria
aoudjit_rachida@yahoo.com

Samia Bouzefrane
CEDRIC Lab,
Conservatoire National des Arts et Métiers
Paris, France
samia.bouzefrane@lecnam.net

***Abstract*—Nowadays, cloud computing is an effective solution for providing computing services to consumers. However, data centers that host computing resources are still faced with a misuse of resources and a workload imbalance of servers, where some servers become overloaded while others are underloaded or even idle. This results in performance degradation and resource wastage. The load balancing is a key aspect and has an important role in the management of cloud data centers. It allows an optimal use of the resources and improves the desired Quality of Service (QoS) using optimal methods for allocating resources and distributing workload.**

In this paper, we propose a load-balancing algorithm that is based on a new parameter called the *balance factor* of the data center, introduced here, to determine if a data center is imbalanced or not, in order to redistribute the workload equally over all the hosts. To minimize the energy consumption of the data center, our strategy relies on the live migration of virtual machines (VMs) while using a mathematical model. The simulation results, using the CloudSim toolkit, have shown that the energy efficiency can be managed by reassigning VMs to the data-center hosts.

***Keywords*-Cloud Data Centers, Consolidation, Energy, Load Balancing, Migration, Optimization.**

I. Introduction

The increasing proliferation of large data centers within which a great number of physical servers are deployed, has allowed cloud computing to gain more attention these last decades. Cloud computing consists of delivering available services over the Internet using hardware and software resources of the data centers according to a service model (SaaS, PaaS, and IaaS).

Behind cloud computing, virtualization techniques [1] are used allowing to host different tenants virtual resources within the same physical resource, hence facilitating the orchestration of these virtual resources and the transformation of the delivery of these resources into a service billable for consumption.

As the demand for resources by customers increases exponentially, the possible reduction of IT infrastructure management costs is one of the top business priorities of cloud providers.

In this context, load balancing is considered as one of the main and biggest challenge in cloud computing because the cloud providers must ensure an equilibrium in the available resources to satisfy the client requests and to meet the clients service level agreements. This equilibrium is done through the distribution of workloads across multiple computing resources, such as computers, computer clusters, disk drives [2], etc. Hence, cloud load balancing means to maintain data centers balanced among the servers in such a way that no servers will be overloaded or underloaded to reduce the risk of saturation, to minimize resource consumption, to facilitate the implementation of fault tolerance, and to guarantee scalability.

Several researches have been undertaken to find optimal and efficient algorithms with the objective of satisfying the workload-equilibrium constraint, and of optimizing the infrastructure management cost of the cloud data centers to gain credibility of customers.

In this paper, we propose to combine two algorithms to solve the problem of load balancing in the data centers, considered as an NP complete problem. The first algorithm is inspired from [3] used to manage the load balancing in MANETs. It is based on the clustering process in order to organize the nodes into clusterheads and to coordinate other nodes in the clusters. This solution has some advantages such as reducing energy consumption and improving bandwidth utilization by reducing communications overhead. Our algorithm purpose is to rebalance the workload of the data-center servers through the migration of Virtual Machines (VMs) from overloaded servers to underloaded ones. To minimize migration costs and energy consumption, the bMatching method [4] is used to complement our load-balancing algorithm.

The remainder of this paper is organized as follows: Section II presents the significant existing research works related to the load-balancing issue, and our motivations, followed by the description of our proposed method as detailed in section III. Section IV presents the power model of our method using the Cloudsim simulator. The conducted

experiments are described in section V before concluding with some perspectives in section VI.

II. RELATED WORK AND MOTIVATION

Given the key role of load-balancing algorithms in cloud computing, several researches have been done in this area and various classifications of load-balancing techniques have been proposed based on different criteria.

In [5], various load-balancing algorithms for cloud computing environments are discussed comprehensively and systematically according to a novel taxonomy. In this study, load-balancing algorithms have been divided into two broad categories, namely, nature-inspired and statistics-based, depending on their source of inspiration and operational model.

Authors in [6], surveyed research literature in the load-balancing area and have presented a new classification of the load-balancing techniques: (1) Hadoop MapReduce load-balancing category, (2) natural phenomenon based load-balancing category, (3) agent-based load-balancing category, and (4) general load-balancing category. In each category, some techniques were studied and analyzed in terms of metrics such as: throughput, makespan, scalability, fault tolerance, migration time, energy consumption, etc.

The authors of [7] surveyed the load-balancing problem by distinguishing between static and dynamic load balancing. The static algorithms are appropriate to the environment with low-loads variations, which requires a complete knowledge on the features and the performances of the system resources. The drawback of the static algorithms is that, once the VMs are assigned to dedicated machines, they cannot move to other ones, while migration of VMs after initial assignment is possible in dynamic load balancing algorithms.

The authors of [8] analyzed and compared three principal algorithms, which are, the Round Robin algorithm, the Active Load Balancing Monitoring algorithm and the Throttled Load Balancing Algorithm. These algorithms are used for load balancing at the scheduling level.

To monitor and balance the load of a cloud data center, authors of [9] propose a graph-based solution that is composed of two phases. In the first phase, the cloud data center is modeled as a network graph that is augmented with a minimum dominating set, inspired from the concept of graph theory, for monitoring its load. The second phase focuses on the system and network aware live migration.

Authors, in [10], combined load balancing and live migration techniques through a mathematical model that computes the average load of each server in terms of CPU, memory and bandwidth in order to determine the overloaded servers. In case of server-overload detection, a VM migration occurs once a destination server is found. The drawback of this solution is that the average resource load is computed within each server, which generates an additional overhead in terms of memory and energy consumption.

A multiple objective optimization framework named MOVMrB is presented in [11]. It aims to rebalance the placements of VMs in order to achieve load balancing of multiple resources in cloud computing. The proposed solution considers both the load balancing of each resource across Host Machines (inter-HM load balancing) and the load balancing of different resources within the same Host Machine (intra-HM load balancing) at the same time.

To reduce the number of SLA violations by migrating VMs before the host is overloaded, a proactive VM migration policy is presented in [12]. This method for the dynamic and anticipatory consolidation of VMs in IaaS cloud data centers uses short-term forecasts of the future utilization of hosts, to proactively detect overloaded hosts. If a host is classified as overloaded, some VMs have to be migrated to other hosts to avoid SLA violations. The predictive overload detection is based on time series technique.

To dynamically forecast the resource utilization and energy consumption in cloud data centers, authors of [13] proposed a multi-objective genetic algorithm which considers the CPU and the memory utilization of VMs and physical machines, and the energy consumption in the data center.

Our contribution, in this article, is to rebalance the workload of the data-center servers through a dynamic load-balancing approach that will detect overloaded and underloaded servers through well-defined thresholds. In order to reduce the consumed energy in our solution, an additional method is used to take into account the migration cost due to the VM migration that is done to remedy to the server-load imbalance.

III. MODEL DESCRIPTION

The cloud computing is based on several data centers distributed geographically. Each data center is composed of a set of physical servers that host VMs. The clients request resources in the form of a service (IaaS, PaaS or SaaS) and the service is offered as a set of VMs.

Workloads of a data center might span across multiple physical servers that are consolidated through live migration to maintain the load of the data center and to avoid overloaded and underloaded servers.

According to the different levels in computer systems defined for energy consumption in Beloglazov et al. [14], our work targets the energy-aware resource management in the data centers. In this context, the load of a data center is linked to its relatively high-energy capacity caused by the physical servers consolidation. The server consolidation is used to maximize the resource utilization and to minimize the energy consumption in the meanwhile. This consolidation is ensured thanks to the VM migration so that a minimum number of servers are utilized to host the VM load, which consequently reduces the energy consumption induced by the physical servers activity. Thus, this paper focuses on the load re-balancing of data centers based on

the live migration leading to the reduction of the energy consumption. As shown in Figure 1, our approach is built upon several modules. Each module is described as follows:

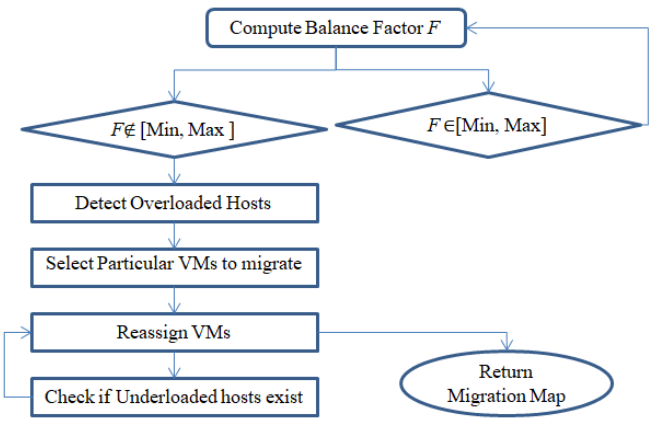

Figure 1: The general steps of our solution

- **Computing Balance Factor**: In order to maintain a balanced load in a data center, we introduce a balance factor that is computed regularly to detect imbalanced load. If this factor has a value ranged between Min and Max values then the load is balanced, otherwise the load-balancing algorithm must be launched. The Min and Max parameters vary with the workload of the data center. In our experiments, we first compute the range $[Min, Max]$ before launching our algorithm.
- **Detecting overload**: An overload-detection algorithm is invoked each time the balance factor is too low or too high.
- **Selecting particular VMs to migrate**: Once overloaded hosts are detected, the next step is to select the VMs that are candidate for migration. The VM selection is not done randomly but follows a specific strategy as detailed in the next section.
- **Effective migration**: In order to migrate the selected VMs, all the servers can be candidate to host the migrated VMs. The destination server that is chosen to host the VM is the one that generates a minimum energy cost among these VMs. The energy cost corresponds to the energy consumed during the execution of the VM within the source server, plus the energy consumed during the migration of the VM from source to destination. The objective of this module is to transfer the selected VMs without interrupting their execution while minimizing their migration cost. For this purpose, we propose a smart reassignment solution that can consolidate the physical substrate in few seconds as in [4]. This reassignment allows to migrate the VMs from the overloaded source to the destination server with a minimum migration cost in term of energy.
- **Check if there are underloaded hosts:** After migrating the VMs from the overloaded nodes, these nodes as well as others may become underloaded due either to the VM migration or to their workload variation. Hence, the system checks if there exists underloaded hosts in order to get rid of the VMs they host by migrating them to non-overloaded nodes with the aim of obtaining some idle servers that can be powered off.

IV. POWER MODEL BASED APPROACH

The power consumption of the data centers is mostly determined by the load of the physical-servers resources such as CPU, memory, disk storage, network interfaces. The CPU capacity contributes significantly in consuming energy. Moreover, the studies in [15] and [16] have shown that, on average, an idle server consumes approximately 70% of the power consumed by the server running at the full CPU speed, while a server in a maximum use can lead to a performance degradation hence paralyzing data-centers activities. To improve the server load balancing and in order to reduce the power consumption at the same time, we propose a power-model based approach that relies on the load-balancing algorithm that we have already used in MANETs [3], combined with the bMatching theory for live migration to ensure the dynamic consolidation of the VMs within the servers of a data center.

A. Load-Balancing Algorithm:

Unlike the existing research works that deal with load balancing in data centers, in our algorithm we introduce a new parameter called *balance factor* and denoted by F in order to maintain perfectly a balanced data-center load even after VM migration. F is defined as follows:

$$F = \frac{n}{\sum_{i=1}^{n} (load_i - \mu)^2} \tag{1}$$

n: is the number of nodes in the data center.
$load_i$: is the load of a node i.

$$\mu = \frac{\sum_{i=1}^{n} load_i}{n} \tag{2}$$

μ: is the average load of the data center.
As explained before, in order to maintain the data center stable and for a good load-balancing algorithm, the balance factor F must be a value in a range $[Min, Max]$, because if F is less than Min value then a load imbalancing will occur causing a degradation in the performances of the servers due to the presence of a great number of underloaded servers. However, if F is greater than Max value, the load-balancing process will be inefficient due to the presence of a great number of overloaded servers.

B. Dynamic Consolidation

To achieve the load balancing, we rely on the VM consolidation used to improve the utilization of data-center resources. VM consolidation is a process that supports VM live migration to reduce the number of physical servers that are powered on while maintaining an equilibrium between the servers workloads. Beloglazov and Buyya in [17] proposed different heuristics for VMs consolidation. Since we focus, in our work, on minimizing energy consumption due to the live migration, we follow some heuristics of [17] either to detect overloaded and underloaded hosts, or to select some VMs candidate for migration, as presented in the following paragraph.

Detection of overloaded hosts: To decide if a host is overloaded, we use the Local Robust Regression (LRR) heuristic proposed by Beloglazov and Buyya in [17]. As stated in [17], "this method of LRR is an iterative method of Local Regression (LR) that fits a simple model to localized subsets of data to build up a curve that approximates the original data".

VMs Selection: Once the overloaded hosts are found, the next step is to select specific VMs to migrate from these hosts. We have opted to the Minimum Migration Time (MMT) policy to select VMs based on the minimum migration time as a selection criterion. The migration time is evaluated as a ratio of the amount of RAM currently utilized by the VM and the spare network bandwidth available for the VM host.

Live migration: During this phase, we consider the optimization technique based on the bMatching theory that we have already proposed in [4]. It consists in migrating judiciously the selected VMs from a set of servers to the optimal ones leading to reduce data-center energy consumption. We assume to have in the data center: N VMs deployed on K available servers, and a set of VMs to be migrated as selected by the MMT policy described above, noted by: T_{init}= $\{VM_1,VM_2,...,VM_\theta\}$ with θ <N and each VM in T_{init} is currently hosted by one of the servers set noted by S= $\{S_1,S_2,...,S_k\}$ with $k < K$. The reassignment problem consists in placing or hosting optimally θ VMs on k available servers. We define $C_{v_{i,j}}$ as the migration energy cost that is considered by our optimization algorithm to recommend the migration of a VM v from a server i to a server j. The k servers of the data center, including the source server i of VM v, are investigated to host v. The destination server j that is selected to host v is the one that will generate the minimum energy cost estimated when v is supposed to migrate from server i to server j. Hence, as explained in the following:

- $C_{v_{i,j}}$ is equal to 0 if $i = j$, which means that, there is no transfer of v since the source host is confused with the destination host.
- $C_{v_{i,j}}$ is equal to a positive value that corresponds to the energy cost of v obtained when simulating the migration of v from server i to server j.

Based on this configuration, we build up a new bi-partite graph noted G=($V \cup S$, E) where: V represents a set of selected virtual machines to migrate with MMT policy and S is the set of all available servers (powered on). E represents a set of weighted edges $w_{(v_{i,j})}$. A weighted edge between V and S depends on the power l consumed by the VM during its execution on the host, and on the energy cost C related to the VM migration, as defined in the following.

1) $w_{(v_{i,j})}$=$l_{(v_{i,j})}$ + $C_{(v_{i,j})}$ if VM v is hosted in server $j \neq i$.
2) $w_{(v_{i,j})}$=$l_{(v_{i,j})}$ if VM v is hosted in server $j = i$.

At this stage according to different costs listed previously, we introduce the minimum weight bMatching objective to be assimilated to the reassignment problem that assigns each VM to the best server with a minimum migration cost. The objective function according to the described costs (migration cost and power consumption) is given as follows:

$$minObjbMatching = \sum_{e \in E, e=(v_i,j)} = (l_e + C_e \mathbb{1}_{ij}) x_e \quad (3)$$

Where:

$$\mathbb{1}_{ij} = \begin{cases} 0, & \text{If } i = j \\ 1, & \text{Otherwise} \end{cases}$$

e=$v_{i,j}$ is the weighted edge if migrating v from server i to server j.

Some additional constraints are added to completely describe the reassignment requirements that are summarized below:

1) Migration with a unique destination: each VM will be assigned to one and only one server.

$$\sum_{e \in \delta(v)} x_e = 1, \forall v \in V \quad (4)$$

2) Servers with power limits constraints: each server s has a resource capacity that we can't exceed.

$$\sum_{e \in \delta(s)} x_e \leq b(s), \forall s \in S \quad (5)$$

Where b(s) is the available resource capacity of the server s in term of CPU.

More details for bMatching method can be found in [4].

Detection of underloaded hosts: Upon all the overloaded hosts are found, and the selected VMs for migration are allocated on the destination hosts, the system will try to find the hosts with the minimum utilization rate in order to reassign the VMs of these underloaded hosts to other ones while keeping these latter not overloaded. This is done by launching again the live-migration phase with the aim of switching off the source hosts that will become idle, and hence reducing the number of active hosts which will minimize the global energy consumption in the meanwhile.

V. EXPERIMENTAL RESULTS ANALYSIS

1) **Experimental environment and platform:** Our target system is an IaaS Cloud platform. Therefore, we choose CloudSim Toolkit [18] as an experimental platform to model and simulate the energy efficiency of our proposed method using the power package. CloudSim simulator is an open source project highly extensible and supported by the research community to simulate, and experiment Cloud computing infrastructures and application services. We have simulated our environment with 800 heterogeneous physical nodes of two types: HP ProLiant ML110 G4 servers, and HP ProLiant ML110 G5. The CPU frequency of the servers is respectively 1860 MIPS and 2660 MIPS. Each server is modeled to have 1GB/s network bandwidth. The characteristics of their power consumption is given in Table I:

Server	HPProLiantML110G4	HPProLiantML110G5
0%	86	93.7
10%	89.4	97
20%	92.6	101
30%	96	105
40%	99.5	110
50%	102	116
60%	106	121
70%	108	125
80%	112	129
90%	114	133
100%	117	135

Table I: Power Consumption at different levels in Watts

We conduct the experiments using 1052 VMs collected from the workload traces of PlanetLab [19]. The VMs are of small, medium, large or extra large types. Each type of VM is assigned an energy consumption value taken from 10, 15, 20, 25 Watts respectively. Since all the servers have the same network bandwidth capacity of 1Gb/s, we assign for each configuration a random migration energy cost ranged between: 0 and 1.

2) **Performance metrics:** In order to evaluate the efficiency of our proposed load balancing bMatching migration cost (LBBMC) method, we made a comparison with an existing method integrated within CloudSim called Local Robust Regression Minimum Migration Time (LRRMMT) that uses LRR policy to detect overloaded and underloaded hosts, as well as MMT to select particular VMs to migrate, similarly to LBBMC. However, LRRMMT uses the best-fit decreasing strategy for the placement of the selected VMs while ours uses bMaching method. For our experiments, first we evaluate the balance factor defined in our LBBMC method. Then, the number of migrations, the energy consumption of the physical servers of the data center, and the execution time metrics are used to compare the LBBMC and LRRMMT methods.

3) **Results and Analysis:**

Balance Factor: We recall that the balance factor allows us to measure the load-balancing degree of the data center. According to Figure 2, the graph shows that the Factor F belongs at the beginning to high range values [50, 80] then it decreases before it stabilizes. This is explained by the consolidation of VMs in order to rebalance the workload among the servers. We can also see that the factor F is stabilized in the interval [35, 55]. Therefore, in order to maintain our system balanced we have to maintain the Balance Factor F in this interval.

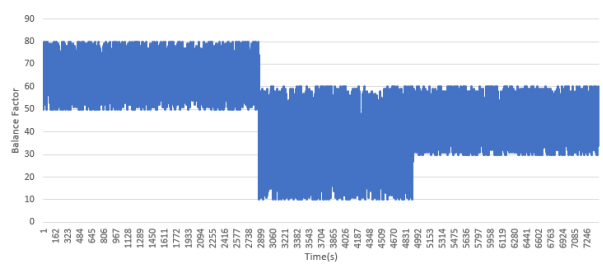

Figure 2: Balance Factor

Migration number: As can be observed by Table II, the number of migrations in LBBMC is lower than that in LRRMMT. As the number of migrations contributes in the energy consumption, our method reassignment avoids unnecessary migrations.

Number Of Migration	
LRRMMT	LBBMC
278	196

Table II: Migration Number

Energy consumption: Energy consumption is the principal parameter used to evaluate our method. As we can see in Table III, the consolidation through LBBMC consumes less energy than through LRRMMT. However, LBBMC has less number of shutdown servers compared to LRRMMT. This is due to the fact that there are more servers in LBBMC that are moderately loaded, which is good, because these active servers consume less energy. Also, this will allow keeping the data center balanced for longer.

Execution time: The execution time (see Table IV) in our LBBMC method is greater than that in LRRMMT method. This is due to the computing time of

Method	Energy Consumption(KWh)	Number of Shutdown Servers
LRRMMT	87.67	646
LBBMC	69.95	532

Table III: Energy Consumption

the balance factor F that is not taken into account in LRRMMT, as well as to the parameters of our bMatching method, used for live migration, that takes into account the energy consumption of the VM in the source server and the migration energy cost.

Execution Time(s)	
LRRMMT	LBBMC
86100	98290

Table IV: Execution Time(s)

VI. CONCLUSION AND FUTURE WORK

Our work shows that load balancing is essential to improve resources utilization and energy consumption by optimally distributing a workload among multiple servers in a data center. In this paper, we have combined a load balancing algorithm with a consolidation VMs through a smart method bMatching for reassignment. The simulation results show that our method improves the management of the workload and reduces the energy consumption for a great number of servers. From a perspective point of view, we envisage to extend our work to adapt our load-balancing algorithm to multiple data centers. The proposed reassignment method takes into account the migration energy cost and the VMs consumption energy in the source server. Our next objective is to investigate new parameters to better manage the energy in cloud computing.

REFERENCES

[1] L. Bouali, E. Abd-Elrahman, A. Hossam, S. Bouzefrane, M. Daoui. "Virtualization Techniques: Challenges and Opportunities", International Conference on Mobile, Secure and Programmable Networking (MSPN'2016), June 2016, pp. 49-62, France.

[2] S. H. Bokhari, M. A. Iqbal, J. H. Saltz, Performance tradeoffs in static and dynamic load-balancing strategies, Book, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, 1986 - 25 pages.

[3] R. Aoudjit, M. Lalam, A. Mzoughi, M. Belkadi, M. Daoui "Load Balancing: An Approach Based on Clustering in Ad Hoc Networks" Journal of Computing and Information Technology, Vol.17 No.2 pp. 177-184 2009.

[4] M. Hadji, N. Djenane, R. Aoudjit, S. Bouzefrane "A New Scalable and Energy Efficient Algorithm for VMs Reassignment in Cloud Data Centers"The IEEE 4th International Conference on Future Internet of Things and Cloud, August 2016, pp. 310-314, Austria.

[5] Avnish Thakur, Major Singh Goraya: A taxonomic survey on load balancing in cloud, Journal of Network and Computer Applications, Vol. 98, Issue C, pp. 43-57, Nov. 2017.

[6] Einollah Jafarnejad Ghomi, Amir Masoud Rahmani , Nooruldeen Nasih Qader, "Load-balancing algorithms in cloud computing: A survey", Journal of Network and Computer Applications Vol. 88, pp. 5071, 2017.

[7] J.-M. Shah et al. "Load Balancing in cloud computing: Methodological Survey on different types of algorithm" International Conference on trends in Electronics and Informatics, pp. 100-107, May 2017.

[8] V. N.Volkova et al. "Load Balancing in Cloud Computing" IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, pp. 387-390, 2018.

[9] Kanniga Devi, R., Murugaboopathi, G. & Muthukannan, M. Cluster Comput (2018). https://doi.org/10.1007/s10586-018-2303-z

[10] S. Hong, W. Weifeng, C. Shiping "Research and Implementation of load Balancing Technology for Cloud Computing" International Conference on Computer Network, Electronic and Automation (ICCNEA), pp. 12-19, 2017.

[11] Rui Li, Qinghua Zheng , Xiuqi Li, Jie Wu, "A Novel Multi-Objective Optimization Scheme for Rebalancing Virtual Machine Placement", IEEE 9th International Conference on Cloud Computing, 2016.

[12] Matthias Sommer, Michael Klink, Sven Tomforde, Jorg Hahner, "Predictive Load Balancing in Cloud Computing Environments based on Ensemble Forecasting", IEEE International Conference on Autonomic Computing, July 2016.

[13] Fan-Hsun Tseng, Xiaofei Wang, Han-Chieh Chao, "Dynamic Resource Prediction and Allocation for Cloud Data Center Using the Multiobjective Genetic Algorithm", IEEE Systems Journal, Vol. 12 Issue 2, 2017.

[14] A. Beloglazov, R. Buyya, Y.C. Lee, A. Zomaya, A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems, Journal of Advances in Computers, Vol. 82, pp. 47-111, 2011.

[15] Anton beloglazov, Rajkumar Buyya, "Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers" in MGC'10, 2010, 4:1-4:6

[16] Rajkumar Buyya, Anton Beloglazof, Jemal Abawajy, "Energy-Efficient Management of Data Center Ressources for Cloud Computing: A vision, Architectural, and Open Challenges" International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010), pp. 12-15, July 2010.

[17] A. Beloglazov, R. Buyya "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 archive, Vol. 24 Issue 13, pp. 1397-1420, Sept. 2012.

[18] Calheiros RN, Ranjan R, Beloglazov A, Rose CAFD, Buyya R. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms". Journal of Software: Practise and Experience, Vol. 41, Issue 1, pp. 23-50, Jan. 2011.

[19] Park, K.; Pai, V.S., "CoMon: a mostly scalable monitoring system for PlanetLab". ACM SIGOPS Oper. Syst. Rev., Vol. 40, Issue 1, pp.6574, 2006.