International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 10 (2018) pp. 098–106
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Modeling and Analysis of Quality of Service and Energy Consumption in Cloud Environment

Abdellah Ouammou, Abdelghani Ben Tahar, Mohamed Hanini and Said El Kafhali

Computer, Networks, Mobility and Modeling Laboratory
Faculty of Sciences and Technologies, Hassan 1st University, Settat, Morocco
a.ouammou@uhp.ac.ma, bentahara@yahoo.fr,
mohamed.hanini@uhp.ac.ma,said.elkafhali@uhp.ac.ma

Abstract: **Cloud computing is an innovative technology that poses several challenges to all organizations around the world. The primary role of cloud providers is to provide a high quality of service (QoS) to their customers as long as they do not consume a lot of energy. A lot of researchers have been interested in this topic and many algorithms have been proposed to manage cloud resources to balance QoS and energy cost. The goal is to improve the QoS of the system, maximize resources utilization and reduce the overall energy consumption. In this paper, we study techniques to manage resources utilization by exploiting the concept of virtual machine migration in a cloud data center (CDC). Two algorithms are presented and studied (Combinations of Migrations (CM) and High Priority (HP)). The obtained numerical results demonstrate the effectiveness of the proposition in terms of makespan and energy efficiency while ensuring the quality of service.**

Keywords: Energy efficiency, Cloud computing, Execution time, Virtualization, VM Live migration, VM placement.

I. Introduction

Cloud computing is a model for enabling convenient on-demand network access to a shared pool of congurable computing resources (e.g network, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. A Cloud data center (or CDC) is a focal and critical concept in cloud computing; it is a facility composed of networked servers or physical machines (PMs) used to organize, process, store and disseminate large amounts of data. Each PM can house a group of multiple virtual machines (VMs) [2]. The cloud computing market is invested by many operators; Amazon, Microsoft, Google, and Apple are among the best known; it is projected to reach 411 billion dollars by 2020, and it is predicted that, by 2021, 28% of all IT spending will be for cloud-based infrastructure, middleware, application and business process services [3]. However, this evolution cannot be without negative ratings. The most important is the energy consumption concern. Indeed, the energy consumption of CDC worldwide is estimated at 26GW corresponding to about 1.4% of the global electrical energy consumption with a growth rate of 12% per year [4]. Depending on a recent study, data centers are the most energy consumers in the ICT ecosystem. Moreover, the initial cost of purchasing equipment for data center already exceeds the cost of its ongoing electricity consumption [5].

On other hand, the virtualization technique plays a central role in cloud environment, and it can be used to reduce the energy and improve the time efficiency [6]. This technology enables a single physical machine to run multiple VMs simultaneously. Moreover, through VMs migration and consolidation, virtualization reduces the total CDC energy consumption. It is known that a significant amount of power is consumed even when the PM is idle (approximately 70% of the power consumed by the PM running at full CPU utilization) [7, 8]. So, by migrating VMs and turns off the inactive PMs we can reduce energy cost. In addition, it can help to reduce the whole number of idle hosts and optimize task waiting time, execution time and operating costs [9]. Nevertheless, an aggressive consolidation of VMs can cause a delay in the execution time of a set of tasks. Therefore, could providers have to deal with energy consumption and Quality of Service trade-off. The achievement of this balance is the main objective of the work presented in this paper. For that, we propose capacity balancing algorithms to improve system operation, reduce the waiting time, and minimize the makespan of the system. We also make a comparative analysis between our algorithms (Combination Migration Policy (CM) and the High Priority Policy (HP)) and random selecting algorithm.

This paper is a major extension of our conference paper that appeared in [10] which focused on proposing a technique capable of managing the migration of the VMs between PMs in a CDC. This proposal technique is based on a double thresholds that manages the selection and placement of VMs in PMs, this management deal with power/performance trade-off. In this extended version, we provide further details for our proposed model by considering the time evolution and proposing a second algorithm for selecting VMs to migrate. Also we give more results and analysis for the performance of the model. Specifically, we concentrate on the customer service in the system, especially the waiting time as well as the execution time and we try to minimize them with the same concept to designate two thresholds and keep the total

CPU utilization between them. Furthermore, we provide numerical results with new figures and substantial discussion.

The rest of the paper is organized as follows: Section II summarizes the related work. The proposed model is presented in section III. Section IV presents the analysis of the proposed model. Section V presents the performance analysis and the numerical results. Finally, section VI is devoted to the conclusion.

II. Related Work

Some studies about cloud energy consumption and QoS analysis were proposed recently. For instance, Srikantaiah *et al.* [11] presented an energy-aware consolidation technique to decrease the total energy consumption of a cloud computing system. The authors modeled the energy consumption of servers in CDC as a function of CPU and disk utilization. For that, they described a simple heuristic algorithm to consolidate the processing works in the cloud computing system. Performance of the proposed solution is evaluated only for very small input size. In [12], authors proposed a mathematical model for server consolidation in which each service was implemented in a VM to reduce number of used PM. Verma *et al.* [13] proposed and formulated a problem of energy consumption for heterogeneous CDC with workload control and dynamic placement of applications. They applied a heuristic for bin packing problem with variable bin sizes. In addition, they introduced the notion of cost of VM live migration, but the information about the cost calculation is not provided. The proposed algorithms do not handle SLA requirements.

A heuristic algorithm for dynamic adaption of VM allocation at run-time is proposed by Beloglazov *et al.* [14]. This algorithm based on the current resources utilization by applying live migration and switching idle nodes to sleep mode. The simulation results show that the proposed algorithm reduces significantly the global energy consumption. Authors in [15], proposed energy consumption formulas calculating the total energy consumption in cloud environments. They provided empirical analysis and generic energy consumption models for idle server and active server states. In [16], the authors proposed a dynamic and adaptive energy-efficient VM consolidation mechanism considering SLA constraints for CDC. Using live migration and switching idle servers to the sleep mode allow cloud providers to optimize resource utilization and reduce energy consumption, considering CPU as the main power consumers in servers. Arroba *et al.* [17] proposed an algorithm based on bin packing problem to minimize the number of bins used. The algorithm speeds up consolidation and the elastic scale out of the IT infrastructure, presenting a global utilization increase of up to 23.46% by reducing the number of active hosts by 44.91%. In [18], authors introduced a unique replication solution which considers both energy efficiency and bandwidth consumption of the geographically CDC systems. The proposed solution improves communication delay and network bandwidth between geographically dispersed CDC as well as inside of each CDC.

Other recent works in the literature focused on Markov chain and queueing theory to evaluate the performance of cloud systems. For example, Hanini *et al.* [19, 20] presented a scheme based on Continuous Markov Chain (CTMC) to manage VMs utilization in a PM with a workload control of the system. They analyzed the proposed scheme using mathematical evaluations of the QoS parameters of the system. Also, they modeled and evaluated the power consumption of the system under the proposed mechanisms. The obtained results demonstrate the usefulness of the proposed model to prevent overload in the system and to enhance its performances such as loss probability, number of jobs, and sojourn time in the system and throughput. In terms of power consumption, the proposed mechanism saves power significantly, and gives a command tool for this which is the arrival rate control parameter. Salah *et al.* [21] proposed a queuing model to predict the needed number of VMs to satisfy a predefined SLA requirement. They conducted experimental measurement on the AWS platform to validate the proposed model. The obtained results show that the proposed model can be extremely useful in achieving proper elasticity for cloud clients. In [22], the authors proposed a queuing mathematical model to study and analyze the performance of multi-core VMs hosting cloud SaaS applications. The proposed model estimates under any incoming workload the number of proper multi-core VM instances required to satisfy the QoS parameters. The queuing model is validated by simulation. The obtained results from analysis and simulation show that the proposed model is powerful and able to predict the system performance and cost and to determine the number of VMs cores needed for SaaS services in order to achieve QoS targets under different workload conditions.

In contrast to the discussed studies, we propose efficient heuristics for dynamic adaption of allocation of VMs in runtime applying live migration according to current utilization of resources and thus minimizing energy consumption. The proposed approach can effectively handle strict QoS requirements, heterogeneous infrastructure and heterogeneous VMs. In addition, in our proposed model we have considered the time evolution of the system. The algorithms do not depend on a particular type of workload and do not require any knowledge about applications executed on VMs.

III. Proposed Model

A. Problem statement

In this work, we consider a data center with m homogeneous PMs, each PM contains a number of heterogeneous VMs, In order to manage the CDC, there are many objectives that have been defined in the literature, among theme, minimizing the number of PMs required or minimizing the number of VMs executed per unit of time. All these objectives contribute to minimize energy consumption or to increase the level of performance. The proposed model in this paper aims to optimally consolidate VMs in a minimum number of PMs while minimizing the energy consumption. In addition, we aim to optimize the execution time of each PM as well as to minimize the withing time for each VM. However, consideration should be given to avoid excessive consolidation that gives us a minimal use of energy but also has a negative impact on the quality of service.

When the provider allocates several tasks to the overloaded PMs, the performance of the CDC system degrades. The idea

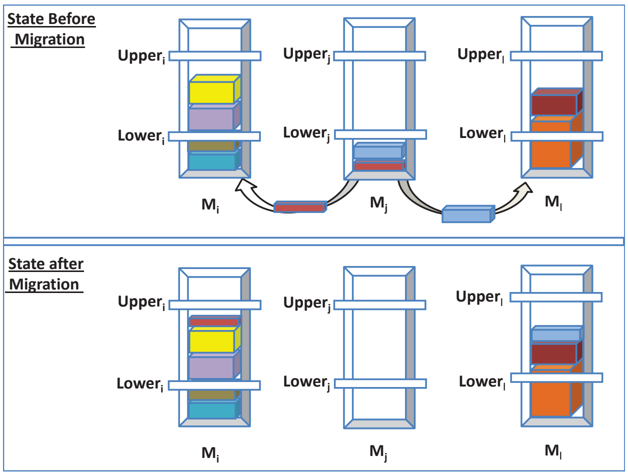

Figure. 1: A use case of the Lower threshold to control VM migration

is to define double thresholds in each PM; namely, the upper threshold and the lower threshold. These thresholds are used to keep the total CPU usage of all VMs in the PM between these two values. If the total use of a PM is below the lower threshold, then all of the VMs running in this PM must be migrated from that host, and that host must be disabled to eliminate active power consumption (Fig.1), in the other case if total CPU usage exceeds the upper threshold, some VMs must be migrated from that host to reduce usage and to avoid potential contract violation at the service level (Fig.2).

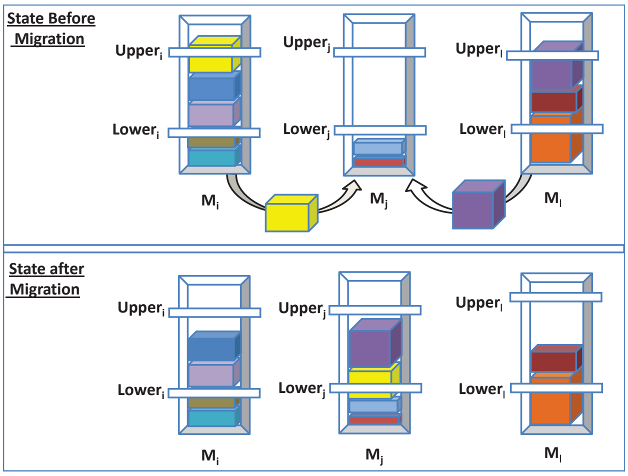

Figure. 2: A use case of the Upper threshold to control VM migration

B. Mathematical formulation

In this section, we formulate our proposed solution using a mixed integer linear programming model. The notation used in this paper are given in the Table 1 below.

The objective of this proposed problem is to balance the total execution time on each physical machine and minimize the number of PMs as much as possible. The objective function can be expressed as follows

$$\min \sum_{j=1}^{m} \int_0^T y_{jt} \mathrm{d}t \tag{1}$$

where y_{jt} are intermediate variables defined by the following equation

$$y_{jt} = \begin{cases} 1, & \text{Machine } M_j \text{ is used at time } t \\ 0, & \text{otherwise} \end{cases} \tag{2}$$
$$\forall j \in [1, m] \ \ \forall t \in [0, T]$$

This optimization is subject to a number of linear constraints depending on the capacity of the host; the VM can exist only on one server at time t; and a server can host VMs according not only the amount of remaining capacity but also according to the effect of this hosting on the system [23].

- The total capacity does not exceed the capacity of system
$$\sum_{j=1}^{m} Cap_j y_{jt} \leq Upper, \quad \forall t \in [0, T] \tag{3}$$
- Total capacity of virtual machines used for each physical machine M_j must be between the thresholds $Upper_j$ and $Lower_j$
$$Lower_j y_{jt} \ \leq \sum_{i=1}^{n} C_{ij} x_{ijt} \leq Upper_j y_{jt}, \tag{4}$$
$$\forall j \in [1, m], \forall t \in [0, T]$$
- No virtual machine can exist in two physical machines at the same time t
$$\sum_{j=1}^{m} x_{ijt} \leq 1, \quad \forall i \in [1, n], \forall t \in [0, T] \tag{5}$$

where x_{ijt} are decision variables defined by

$$x_{ijt} = \begin{cases} 1, & i^{th} \text{ VM mapped to } j^{th} \text{ at time } t \\ 0, & \text{otherwise} \end{cases} \tag{6}$$
$$\forall j \in [1, m], \forall i \in [1, n], \forall t \in [0, T]$$

These can be summarized by combining the objective function with all constraints in the following set of equations

$$\begin{array}{ll} \min \sum_{j=1}^{m} \int_0^T y_{jt} \mathrm{d}t & \\ \text{subject to} & \sum_{j=1}^{m} Cap_j y_{jt} \leq Upper, \\ & Lower_j y_{jt} \ \leq \sum_{i=1}^{n} C_{ij} x_{ijt} \leq Upper_j y_{jt} \\ & \sum_{j=1}^{m} x_{ijt} \leq 1, \\ & \sum_{i=1}^{n} x_{ijt} \leq v_j. \end{array} \tag{7}$$
$$\forall i \in [1, n], \forall j \in [1, m], \forall t \in [0, T]$$

IV. Model Analysis

The proposed mechanism attempts to optimize the current state of the VMs in two phases. In the first phase, we select the VMs to be migrated. In the second step, using Bin Packing algorithm, the selected VMs are moved to another host that verify all constraints defined above.

1) VM selection:

When the total CPU utilization is no longer limited within the two thresholds, by either exceeding the upper threshold or being beneath the lower. Within the last case in redistributing, all the VMs must be moved to another host, while in the other case we choose just some VMs in order to migrate them to bring back the balance between the thresholds. This selection is acquainted with two policies: the Combinations of Migrations (CM) and High Priority (HP). The CM policy is an improvement of that proposed in our previous paper [10] and HP is a new proposed policy based on VM utilization. The two policies are described below:

1. **The Combinations of Migrations (CM)** : This policy selects the necessary number of VMs to migrate from a host based on combination between the machines and choosing the minimum VMs they can return the CPU utilization below the upper threshold if the upper threshold is exceeded. The CM policy finds a set R$_j$ of VMs which must be migrated from the host j. Let V$_j$ the set of VMs currently allocated to the host j. For each VM$_i$ in the host j, C$_{ij}$ is its capacity. The R$_j$ as a subset of V$_j$ is defined by the following equations:

   $if \;\; Upper < Ucurr$:

$$R_j = \{V_{ij} \in V_j | \min_{1\le k\le |V_j|} \sum_{i=1}^{\binom{n}{k}} C_{ij}x_{ij} \ge Ucurr_j.y_{jt} - Upper_j.y_{jt}\} \tag{8}$$

   $if \;\; Ucurr < Lower$:

$$R_j = V_j \tag{9}$$

2. **High Priority (HP)** : This policy specifies a set R$_j$ of VMs where the most used VMs have priority to stay in the same PM. All VMs having the lowest CPU usage are ordered from smallest to largest until we reach the value that exceeds the threshold are migrated. We can assume V$_{ij}$ so that $C_{1j} < C_{2j}, \ldots$

$$R_j = \begin{cases} \{V_{1j}, \ldots, V_{kj}\} & \text{if } Upper < Ucurr \\ V_j & \text{if } Ucurr < Lower \\ \emptyset & \text{Otherwise} \end{cases} \tag{10}$$

   where

$$k = \text{argmin}_{1\le k\le |V_j|} \sum_{i=1}^{k} C_{ij}x_{ijt} \ge Ucurr.y_{jt} - Upper.y_{jt}$$

2) VM Migration :

Finding the best location of the selected VMs can require a list of information about the latter, their number, capacity, and the former replacement of the VMs. To provide all that we need to define a bivalent variable D$_{ijk}$ expressing replacement of VM$_k$ from PM$_i$ to PM$_j$. This bivalent variable D$_{ijk}$ guide us to propose inequalities that have to be respected during the migration of VMs.

1. Once a VM is migrated to a server, we can not migrate it again from that server (Fig.3). This is reflected by the following inequality

$$D_{ijk} + D_{jlk} \le 1, \forall l \tag{11}$$

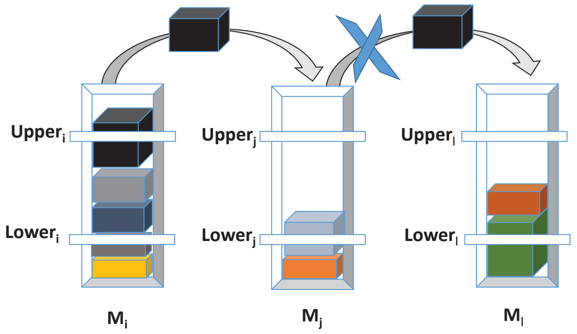

Figure. 3: The first constraint on VM migration

2. To reinforce the previous condition, an inequality is added to ensure that when a VM$_k$ is migrated from PM$_i$ to PM$_j$ at time t , the migration to the other nodes $l(l \neq j)$ is forbidden, in other words each machine has one and only one destination (Fig.4).

$$\sum_{j=1, j\neq i}^{m'} D_{ijk} \le 1. \tag{12}$$

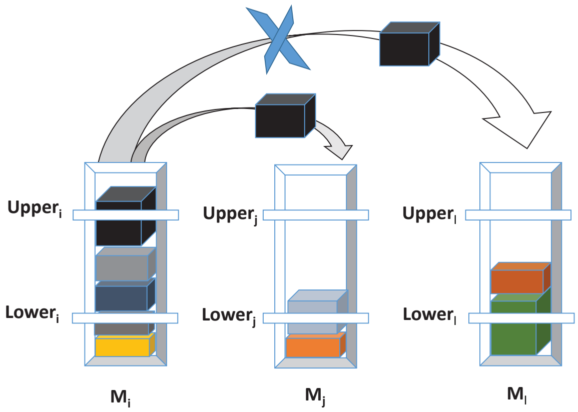

Figure. 4: The second constraint on VM migration

3. If a PM$_i$ indicates that its usage is below the lower threshold, then it must migrate all of its hosted virtual machines to turn it off permanently:

$$\text{if} \quad Ucurr < Lower, \sum_{j=1, j\neq i}^{m'} \sum_{k=1}^{v_i} D_{ijk} = v_i y_{it}. \tag{13}$$

To solve this part we apply a First Fit Decreasing Scheduling (FFDS) bin packing algorithm. In favor of the heterogeneity of the nodes it is possible to choose the most effective power to suit the rhythm of the system. The pseudo code for the algorithm is presented in algorithm 1.

Algorithm 1 First Fit Decreasing Scheduling (FFDS)

```
Input: List of PMs , List of VMs
Output: Allocation of VMs
sortDecreasing all virtual machines
for each VM in List of VMs do
    for each PM in List of PMs do
        if Utilization of PM has not exceed upper threshold and has enough capacity for VM then
            Utilization of PM = Utilization of PM + Utilization of VM
            Remain=Upper-Utilization of PM
            if Remain is the minimum between the values provided by all VMs then
                Allocated VM in PM
                Remove VM in List of VM
            endif
        endif
    endfor
endfor
return VMs allocation
```

V. Performance analysis

In migration algorithm, the upper threshold is set to avoid the SLA violations and ensure smooth task continuity. Each PM periodically executes an overload detection strategy to trigger migration. A PM is overload, when the resource utilization reaches the upper threshold. The upper threshold should be adjusted, depending on the specific system requirements, to avoid performance degradation and SLA violations. A PM is considered to be under loaded, when the resource utilization is under the lower threshold. The lower threshold significantly affects the energy efficiency and the amount of migrations.

We perform modelization in Matlab to evaluate our model. Modelization has been chosen to evaluate the performance of the proposed algorithms, we have fixed the number of heterogeneous VMs to 100 and the number of PMs ranges from 10 to 100 by 10. Runs were performed to compare our methods with the one where no policy is used.

A. Analysis of energy consumption

To evaluate the efficiency of the proposed model, we evaluate the amount energy consumed in CDC. To achieve this end, the following formula is used to calculate the energy

$$E_{Tot} = E + \sum_{activePM_j} (P_j + \sum_{VM_{ij} \in V_j} \alpha \times U(VM_{ij})) \quad (14)$$

where E is the power needed for monitoring the CDC in idle state, P_j is the power consumption in idle state for a PM_j, $U(VM_{ij})$ is the utilization of the VM_{ij} and α is a power weight coefficient.

The influence of high and low thresholds is evaluated on energy efficiency in data center which consists of 100 VMs.

In Figure 5, we plot the amount of energy consumed under our proposed algorithms compared to the case where a random selecting algorithm is used. As the Figure 5 shows, the system energy consumption vary with the value of number of PMs. The tendencies of the system energy consumption obtained from the two experiences are almost similar. We remark that, when the number of PM is more than 10, CM policy and HP policy are both minimizing the total consumed energy. However, it increases faster from 70 to 90 and the total energy consumption is always lower when using our algorithm, in this case of energy consumption, CM policy outperform HP policy by minimized more energy. The total consumed energy is minimizing by an average of 20% (respectively 16%) by applying CM policy (respectively HP policy), the saved energy is due to the way of migration proposed in our algorithm that turn off unneeded PMs.

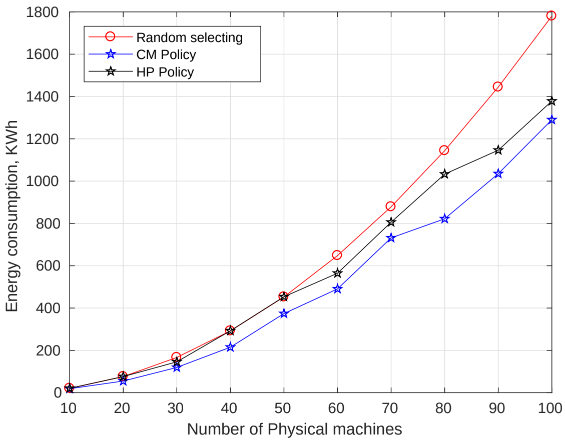

Figure. 5: Comparison of the system energy consumption

B. Analysis of execution time and CPU utilization

In this part, our prime aim is to minimize the makespan which is the time required to complete the execution of all input tasks by the system. It can be represented as follow

$$Makespan = \max_{1 \le j \le m} \sum_{i=1}^{v_j} AccoTime_{ij} \quad (15)$$

where, $AccoTime_{ij}$ is the accomplishment time of i^{th} VM on j^{th} PM [24]. During the execution of task, our proposed algorithm is about minimizing the total execution time on each physical machine. We assume that all the VMs are running respectively from he highest to smallest and each VM/task could be executed in the period $[s_{ij}, s_{ij} + p_{ij}]$.
where s_{ij} is the starting time and p_{ij} is the processing time.
The total running capacity of PM $_j$ is the sum of capacity of all VMs running in this PM.

$$Cap_j = \sum_{i=1}^{v_j} C_{ij} \quad (16)$$

Now, the duration time of i^{th} VM is estimated using

$$AccoTime_{ij} = C_{ij} \times S \quad (17)$$

Where S is the time needed to implement a unit capacity
To analyze the performance of our proposed algorithm FFDS in function of the waiting time earned by applying the HP policy, we perform an evaluation with 10 PMs and 60 VMs

with $S = 0.02$ ms. Figure 6 illustrates the execution time in two cases: (1) in case of random selecting (2) in case of HP policy. We remark that the execution time has been decreased significantly by an average of 69%. These results show one other great advantage of our algorithm regrading the execution time of VMs in CDC.

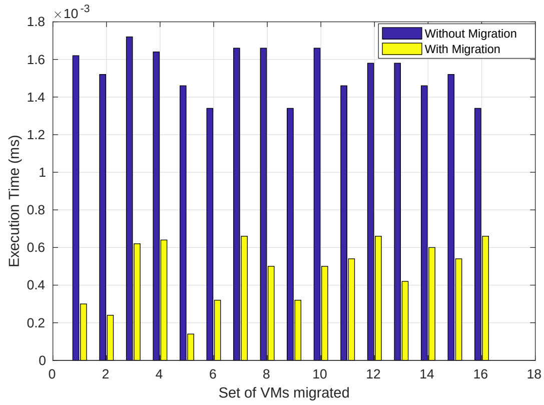

Figure. 6: The execution time vs migrated VMs

The next experiment focuses on the impact of our algorithm applying both policies CM and HP to manage the CPU utilization between 10 PMs mapping 60 VMs. As shown in Figure 7, the selecting random deal with allocation the VMs in a wrong way, where some PMs exceed the upper threshold while other PMs are active just to serve some VMs that have a small capacities, and that poses two problems, the first depends on the energy consumed, which will be higher because all the PMs are active, the second concerns the overload on some PMs which can have a negative impact on the quality of service (Figure 6). On the other hand, both policies find a way to improve significantly the CPU utilization of the system, and this is done by keeping the CPU utilization of all PMs between the upper and lower thresholds that are in our experiment 70% and 30% respectively.

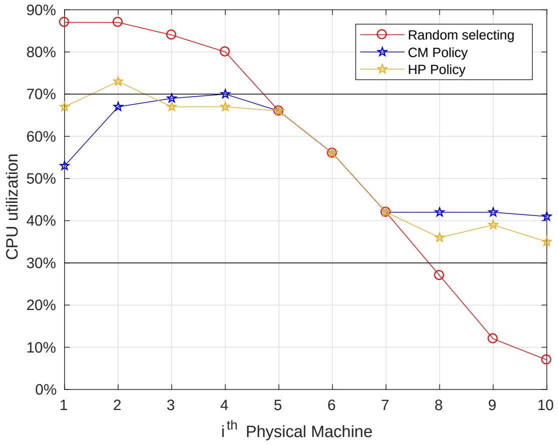

Figure. 7: Variation of CPU utilization on each physical machine

Table 2: Cost of migration

Number of VM	Number of migration due to exceeding upper threshold	Cost
10	1	30
20	8	-32
30	12	-26
40	17	-50
50	14	-10
60	25	-80
70	29	-100
80	25	-58
90	38	-130
100	40	-122

C. *Cost of VM migration*

VMs migrate at the beginning of each monitor period. It has negative impacts on performance of the running tasks. We assume that each VM migration costs the same amount of resources. So it is crucial to minimize the number of VM migrations ensuring the QoS and energy conservation. In this work we are interested in evaluating the cost of migration with a view to study and minimize it in future work. Each migration has a cost in QoS. However, when the migration is monitored due to exceeding of the upper threshold, the QoS of the migrated machine and of remained VMs in the PM where the migration is performed from is improved. Then, the total cost of this operation can be computed using the following formula

$$Cost_{Mig} = \beta \times NM_{Tot} - \gamma \times NM_{up} \qquad (18)$$

where NM_{Tot} is the total number of migrated VM, NM_{up} is the number of VM migrated due to exceeding of the upper threshold, β is the cost of a migration and γ is the gain in QoS when the migration is monitored due to exceeding of the upper threshold.

Moreover, we can suppose that β is very small compared to γ based to the fact that migration techniques used in CDC are developed and permits minimizing the cost due to this operation. In our experimentation, we use $\beta = 2$ and $\gamma = 4$ Table 2 shows the cost of migration when the number of VMs is varying. We remark that when the number of VMs is increasing, the cost becomes not positive, which means that we gain at the QoS. This is due to the fact that there is more chance to have an exceeding of the upper threshold.

Figure 8 shows that when the number of PMs increases the provider has more gain than loss (negative cost). When we compare the two proposed policies (CM and HP), we remark that HP policy performs better with lower number of PMs, and CM policy is better with large number of PMs.

VI. Conclusion

In this paper we have proposed a capacity balancing algorithm to manage the VM migration between the PM in a CDC in order to deal with power-performance trade-off. Our proposition is based on double thresholds that manage the selection and placement of VMs in physical machines using both policies CM and HP. This model is presented as a mixed integer linear mathematical model. The numerical evaluation of the proposed technique showed that our model enables to save energy consumed and minimizes the execution time in CD-

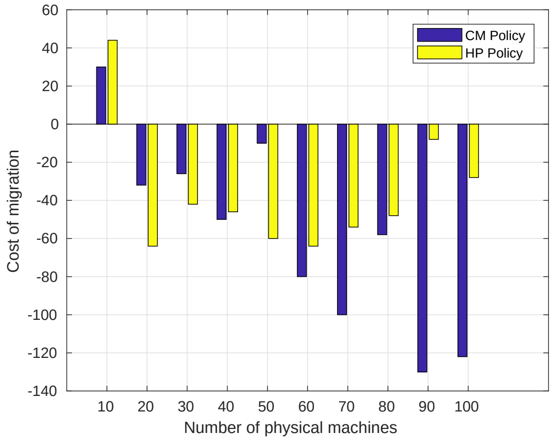

Figure. 8: The comparison of the double threshold policies in terms of cost

C. However, this work has to be detailed in terms of QoS performances.

References

[1] P. Mell, T. Grance. The NIST definition of cloud computing. *NIST report 2011*, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, 2011.

[2] S. El Kafhali, K. Salah. Stochastic modelling and analysis of cloud computing data center. *Proceedings of 20th Conference Innovations in Clouds, Internet and Networks (ICIN'17)*, pp. 122-126, 2017.

[3] `https://www.forbes.com/`, accessed on Feb 14, 2018

[4] J. Koomey. Estimating Total Power Consumption by Servers in the U.S. and the World. February 2007.

[5] Digital Power Group. The cloud begins with coalBig data, big networks, big infrastructure, and big power, 2013.

[6] M. Dayarathna, Y. Wen, R. Fan. Data center energy consumption modeling: A survey. *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732-794, 2016.

[7] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, G. Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster computing*, Vol. 12, No. 1, pp. 1-15, 2009.

[8] S. El Kafhali, K. Salah. Modeling and analysis of performance and energy consumption in cloud data centers. *Arabian Journal for Science and Engineering*, pp. 1-14, 2018.

[9] T. Chatterjee, V. K. Ojha, M. Adhikari, S. Banerjee, U. Biswas, V. Snášel. Design and Implementation of an Improved Datacenter Broker Policy to Improve the QoS of a Cloud. *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA'14)*, pp. 281-290, 2014.

[10] A. Ouammou, M. Hanini, S. El Kafhali, A. Ben Tahar. Energy consumption and cost analysis for data centers with workload control. *Proceedings of the 8th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA'17)*, pp. 92-101, 2017.

[11] S. Srikantaiah, A. Kansal, F. Zhao. Energy aware consolidation for cloud computing. *Proceedings of the 2008 conference on Power aware computing and systems (HotPower'08)*, vol. 10, pp. 1-5, 2008.

[12] B. Speitkamp, M. Bicher. A Mathematical Programming Approach for server consolidation Problems in Virtualized Data Centers. *IEEE Transitions on Service Computing*, vol. 3, no. 4, pp. 266-278, 2010.

[13] A. Verma, P. Ahuja, A. Neogi. pMapper: power and migration cost aware application placement in virtualized systems. *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, pp. 243-264, 2008.

[14] A. Beloglazov, J. Abawajy, R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755-768, 2012.

[15] U. Awada, K. Li, Y. Shen. Energy consumption in cloud computing data centers. *International Journal of Cloud Computing and services science*, vol.3, no.3, pp. 145-162, 2014.

[16] S. Y. Z. Fard, M. R. Ahmadi, S. Adabi. A dynamic VM consolidation technique for QoS and energy consumption in cloud environment. *The Journal of Supercomputing*, vol. 73, no. 10, pp. 4347-4368, 2017.

[17] P. Arroba, J. M. Moya, J. L. Ayala, R. Buyya. Dynamic Voltage and Frequency Scalingaware dynamic consolidation of virtual machines for energy efficient cloud data centers. *Concurrency and Computation: Practice and Experience*, vol. 29, no. 10, pp. 1-13, 2017.

[18] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, A. Y. Zomaya. Energy efficient data replication in cloud computing datacenters. *Cluster Computing*, vol. 18, no. 1, pp. 385-402, 2015.

[19] M. Hanini, S. El Kafhali. Cloud computing performance evaluation under dynamic resource utilization and traffic control. *Proceedings of the 2nd ACM International Conference on Big Data, Cloud and Applications (BDCA17)*, pp. 1-6, 2017.

[20] M. Hanini, S. El Kafhali K. Salah. Dynamic VM Allocation and Traffic Control to Manage QoS and Energy Consumption in Cloud Computing Environment. *International Journal of Computer Applications in Technology*, 2018.

[21] K. Salah, K. Elbadawi, R. Boutaba. An analytical model for estimating cloud resources of elastic services. *Journal of Network and Systems Management*, Vol. 24, No. 2, pp. 285-308, 2016.

[22] S. El Kafhali, K. Salah. Performance Analysis of Multi-Core VMs hosting Cloud SaaS Applications. *Computer Standards & Interfaces*, vol. 55, pp. 126-135, 2018.

[23] Q.T. Nguyen, N. Quang-Hung, N. H. Tuong, V. H. Tran, N. Thoai. Virtual machine allocation in cloud computing for minimizing total execution time on each machine. *Proceedings of the International Conference on Computing, Management and Telecommunications (ComManTel)*, pp. 241-245, 2013.

[24] S. K. Mishra, M. A. Khan, B. Sahoo, D. Puthal, M. S. Obaidat, K. F. Hsiao. Time efficient dynamic threshold-based load balancing technique for Cloud Computing. *Proceedings of the IEEE CITS International Conference on Computer, Information and Telecommunication Systems*, pp. 161-165, 2017.

Author Biographies

Abdellah Ouammou is a Ph.D. student in Applied Mathematics at Computer, Networks, Mobility and Modeling laboratory, Faculty of Sciences and Technologies, Hassan 1st University, Settat, Morocco. He has completed his Bachelors degree in Mathematics and Applications at the Faculty of Sciences, Ibn Zohr University, Agadir, Morocco, in 2013, and he received the Masters degree in Mathematics and Applications from Faculty of Sciences and technologies, Hassan 1st University, Settat, Morocco, in 2016. His current research interests Probability, Stochastic optimization, Discrete stochastic processes, Discrete optimization, and Cloud computing environments.

Abdelghani Ben Tahar received his PhD degree in applied mathematics from Hassan II University, Casablanca, Morocco in 2001. He was in an INRIA post-doctoral at Rocquencourt and a CNRS post-doctoral fellows at LIRMM, Montpellier, France, and a lecturer at LMRS (UMR 6085 CNRS-Univ. Rouen). Since 2009 he is a Professor at Hassan 1st University, Settat, Morocco. His current fields of research interests are focusing on networks performance evaluation and queueing network models.

Mohamed Hanini is currently a professor at the department of Mathematics and computer science in the Faculty of Sciences and techniques, Hassan 1st University Settat, Morocco. He obtained his PhD degree in mathematics and computer in 2013. He is the author and co-author of several papers related to the fields of modeling and performance evaluation of communication networks, cloud computing and security. He participated as TPC member and as an organizing committee member in international conferences and workshops, and he worked as reviewer for several international journals.

Said El Kafhali is a professor at the Department of Mathematics and Computer Science, Faculty of Sciences and Technologies, Hassan 1st University, Settat, Morocco. He received the B.S. degree in Computer Sciences from Faculty of Sciences, Sidi Mohamed Ben Abdellah University, Fez, Morocco, in 2005, the M.S. degree in Mathematical and Computer Engineering from Faculty of Sciences and Technologies, Hassan 1st University, Settat, Morocco, in 2009, and the Ph.D. degree in Computer Science from the same institution in 2013. He joined Faculty of Sciences and Technologies in January 2018. Prior to joining Faculty of Sciences and Technologies, Dr. Said El Kafhali worked for four years at the Department of computer and telecom engineering, National School of Applied Sciences, Hassan 1st University, Khouribga, Morocco. His current research interests Queuing Theory, Performance Modeling and Analysis, Cloud Computing, Internet of Things, Fog/Edge computing and Networks Security. Dr. Said El Kafhali is closely associated with several international journals as a reviewer. He serves as international programme committee member in many international and peer-reviewed conferences. He has several publications in the areas of cloud computing, computer security, computer networks, Internet of Things, Fog/Edge computing and performance modeling and analysis.

Table 1: Notation and Terminology

Notation	Description
m	Number of physical machines $(M_1, M_2, ..., M_m)$
n	Number of virtual machines in all physical machines
m'	Number of physical machines $(M_1, M_2, ..., M_{m'})$ possible to host the VM
v_j	Maximum number of virtual machines allocated to a physical machine M_j
x_{ijt}	bivalent variable indicating that VM_i is assigned to a M_j
C_{ij}	Capacity of each virtual machine i mapped in physical machine j
$Upper_j$	Up Threshold for each physical machine j
$Lower_j$	Down Threshold for each physical machine j
Cap_j	Total capacity of the physical machines j
$Upper$	Up Threshold for the total physical machines
P_i	Processing time of virtual machine i
s_i	Starting time of virtual machine i
T	The total execution time