



**HAL**  
open science

## Energy-aware VM Placement based on intra-balanced resource allocation in data centers

Imene El-Taani, Mohand-Cherif Boukala, Samia Bouzefrane

► **To cite this version:**

Imene El-Taani, Mohand-Cherif Boukala, Samia Bouzefrane. Energy-aware VM Placement based on intra-balanced resource allocation in data centers. 8th International Conference on Future Internet of Things and Cloud (FiCloud), Aug 2021, Rome, Italy. pp.400-405, 10.1109/FiCloud49777.2021.00065 . hal-03381631

**HAL Id: hal-03381631**

**<https://hal.science/hal-03381631v1>**

Submitted on 17 Oct 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-aware VM placement based on intra-balanced resource allocation in data centers

Imene El-Taani

*MOVEP Lab*

*USTHB*

Algiers, Algeria

imene.eltaani@edu.usthb.dz

Mohand-Cherif Boukala

*MOVEP Lab*

*USTHB*

Algiers, Algeria

mboukala@usthb.dz

Samia Bouzefrane

*CEDRIC Lab*

*Cnam*

Paris, France

samia.bouzefrane@cnam.fr

**Abstract**—The virtualization of operating systems and network functions results in a large amount of virtual machines (VM) that are hosted in data-center servers called physical machines (PM). This requires an efficient management and consolidation in order to use as few physical resources as possible to reduce the energy consumption. This paper addresses the VM placement problem with the aim of minimizing energy consumption as well as optimizing resource wastage in a data center. To achieve this goal, we propose an approach based on statistical methods to measure the similarity between resource vectors of PMs and VMs. The proposed strategy allows minimizing the wastage of resources by placing VMs on PMs while keeping the intra-workload balancing of the different resources. The energy consumption minimization is taken into account by reinforcing the selection of the target PM criterion as well as the growth rate of energy consumption. The energy efficiency and the optimization of resource wastage are demonstrated through simulation experiments. The obtained results show that our proposed IRB-VMP algorithm significantly outperforms the state-of-the-art FFD and BFD algorithms regarding the considered metrics.

**Index Terms**—Data Center, Virtual Machine, Physical Machine, Placement, Allocation, Energy efficiency, Resource Wastage.

## I. INTRODUCTION

Cloud computing is an emerging technology that offers services to the costumers using a pay-as-you-go model to charge users for the amount of services consumed. To guarantee these services, a large number of data centers that consist of thousands of physical servers has been localized all over the world. The cloud services can be categorized according to three service models as following: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) [1].

Having a data center with an enormous number of physical machines (PMs) led to height operational costs such as: cooling, electrical power, maintenance costs, etc. [2]. Hence, optimizing these costs can allow the cloud providers to offer competitive services to their customers. In fact, the energy consumption is the largest fraction of the total cost in the data center and it has been shown in [3] that idle servers still consume about 70% of their peak power. Therefore, keeping servers underutilized is highly inefficient from the energy consumption perspective. To address this problem, modern data centers use the virtualization technology [4]. This latter

provides a potential solution to enhance resource utilization, by allowing a PM to be shared by many instances of separate environments known as virtual machines (VMs). Although the benefits provided by virtualization, an efficient strategy for VMs placement (VMP) is required to maximize the advantages offered by this technology. A VMP strategy assigns a set of VMs to the most suitable PMs in a data center. This is necessary when new requests for VMs are submitted to the data center, and when some currently running VMs need to be migrated to other PMs in order to avoid underload or overload utilization of resources. However, VMP is a complex computation task and is known as an NP-Hard optimization problem [5]. As there is no exact algorithm to solve the VMP problem in a reasonable time, many research works have been done in this field, and a lot of heuristic approaches are proposed in the literature. These approaches can be defined as a single or a multi objective optimization problem [1]. The main goal of a single objective optimization [6] is to find the best solution that corresponds to the minimum or the maximum value of a single objective function under certain criteria. On the other hand, the multi-objective approach [1] [6] tries to optimize two or more objectives simultaneously. For example, minimizing the number of active PMs while maximizing the resource utilization is a two objective optimization problem.

In this paper, we propose a bi-objective approach, with the aim of minimizing the energy power consumption and the wastage of resource utilization. In order to solve this optimization problem, we propose to maintain an intra-balance workload of different resources on the target PMs. Consequently, this will help to make the placement less prone to resource wastage, which will require less number of active PMs. To achieve this goal, we design an algorithm based on the similarity between the residual resources of PMs and the requested resources for VMs. As in [7] [8], our algorithm considers the CPU, RAM and bandwidth (BW) as elements of resource vectors.

The rest of this paper is organized as follows. In the next section, the related work is presented. Section III highlights the formulation of the problem and the resources representation. Our proposed IRB-VMP algorithm is described in Section IV. In section V, the experimental evaluations are discussed. Finally, Section VI is devoted to conclusions and future works.

## II. RELATED WORK

Several methods have been proposed for VM placement in the literature. The survey papers in [1], [6] and [9] present the state-of-the-art and the background of VMP strategies. Most of them consider the VMP as a bin packing problem [10] and applied the greedy heuristic models to solve it. However, most of these approaches have depended only on one-dimensional resource and did not pay a careful attention to the situation where provisioning VMs will saturate one type of resource on the target PM and waste the other resources.

One of the relevant works in the field is the work presented in [3] by Beloglazov and Buyya. In their work, the authors have proposed a power-aware best fit decreasing (PABFD) algorithm for dynamic VM consolidation. First, they sort all the unassigned VMs in a decreasing order of their CPU requirements, then allocate each VM to a host that provides the least increase of the power consumption caused by the allocation. However this solution does not consider other resource utilization, and their possible wastage.

Zhihua et al. [7] proposed an evidence-efficient affinity propagation scheme for VM placement (EEAP-VMP), which is capable of balancing the workload across various types of resources on the running PMs. A set of experiments is conducted to compare the EEAP-VMP to the well-known state-of-the-art methods as first fit decreasing (FFD) [10] and best fit decreasing (BFD) [11]. The obtained results have been demonstrated that this method is highly effective on reducing VM migrations, energy consumption and resource wastage.

The authors, in [8], proposed a Greedy Randomized VM Placement (GRVMP) algorithm. GRVMP inspires the ‘power of two choices’ model [12] and mixes Integer Linear Programming (MILP) with the objectives of optimizing consumption of power and resource wastage. The results showed that GRVMP significantly outperforms the baseline algorithms.

Similar to this later work, a Multi-Objective Integer Linear Programming (MOILP) model is proposed in [13], with the aim of optimizing simultaneously the energy consumption, the resource wastage and the number of active PMs. The MOILP model is solved through two different methods. The first method computes a unique solution for a given preference order over the objectives whereas the second computes a set of non-dominated solutions. Through the conducted simulations, they show the effectiveness of the second method over the first method.

Unlike the existing works, our contribution considers a vector of resources to handle distinct resources simultaneously during the VMP process while relying on statistical similarity methods to optimize energy consumption. Our contribution is presented in the next sections through the description of the VMP formulation and the corresponding algorithm.

## III. INTRA-BALANCE VMP FORMULATION

In this section, we present the mathematical formulation for our proposed intra-balance VM placement.

### A. VMP problem and resource modeling

Let  $P = \{\vec{p}_1, \vec{p}_2, \dots, \vec{p}_n\}$  and  $V = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_m\}$  be respectively a set of  $n$  PMs and  $m$  VMs. Our VMP can be represented as a mapping function from  $V$  to  $P$ ,  $f: V \rightarrow P$  such that  $f(\vec{v}_j) = \vec{p}_i$ .

As in [7], [8] and [14], each PM and VM can be defined as a vector in a 3-dimensional space, where 3 denotes the number of resources: CPU, RAM and BW. The resources capacity of the  $i^{th}$  PM is represented as  $C_i = (C_i^{CPU}, C_i^{RAM}, C_i^{BW})$  and the current resource demands for  $j^{th}$  VM is defined as  $D_j = (D_j^{CPU}, D_j^{RAM}, D_j^{BW})$ . Therefore, the residual resources of the  $i^{th}$  PM can be calculated by formula (1).

$$R_i = C_i - \sum_{f(v_j)=p_i} D_j \quad (1)$$

The total resource demands of all VMs placed on the  $i^{th}$  PM cannot exceed the capacity offered by this latter. This constraint can be expressed as shown in formula (2).

$$\sum_{f(v_j)=p_i} D_j^r \leq C_i^r \quad (2)$$

Where  $r \in RS = \{CPU, RAM, BW\}$

To deal with the numerical differences between the different resource types, we normalize the resource values with respect to the total capacity of each resource type. Hence, the total available resource can be represented as a unit vector (1, 1, 1). The remaining capacity vector for the  $i^{th}$  PM is expressed as in formula (3).

$$RN_i^r = \frac{R_i^r}{C_i^r} \quad (3)$$

Where  $r \in RS$

Note that the resource demands are normalized as in formula (3) with respect to the total capacity of the target PM. In the rest of this paper, we will refer to them as  $DN_j^r$ .

### B. Intra-resource balancing in VM placement

The motivation behind keeping an intra-resource balancing on the target PM during the VM placement is to avoid the situation where only one type of resource is fully occupied. In this case, no more VM can be placed on this PM and all the other residual resources are wasted. In order to fulfill the above objective, we propose the use of some similarity distances between the vector of VM resource demand and the vector of the PM residual capacity. More exactly, we use the cosine [15] and the Jaccard distance [16] to measure how much the VM requirements are complementary to the occupied resources on a physical host. The cosine similarity between two vectors  $\vec{p}_i$  and  $\vec{v}_j$  is defined as in formula (4).

$$\begin{aligned} \text{cosine}(\vec{p}_i, \vec{v}_j) &= \frac{\vec{p}_i \cdot \vec{v}_j}{|\vec{p}_i| |\vec{v}_j|} \\ &= \frac{\sum_{r \in RS} RN_i^r \times DN_j^r}{\sqrt{\sum_{r \in RS} (RN_i^r)^2} \sqrt{\sum_{r \in RS} (DN_j^r)^2}} \quad (4) \end{aligned}$$

The cosine distance measures the angle between two vectors and determines whether two vectors are having the same direction. A cosine value of 0 means that the two vectors are at  $90^\circ$  to each other (orthogonal) and have no match (see Fig. 1). The closer is the cosine value to 1, the smaller is the angle between vectors.

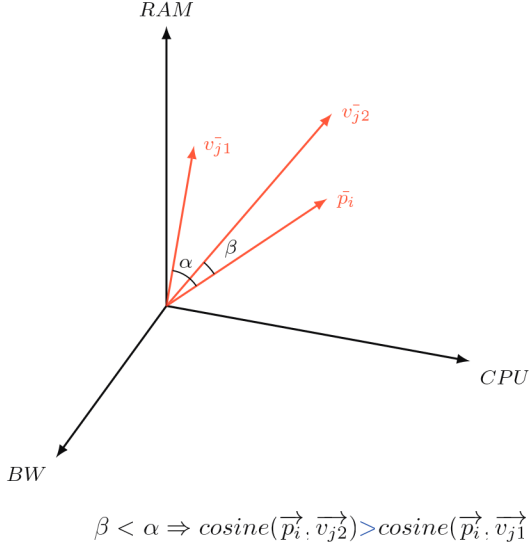


Fig. 1. Cosine Similarity

For concrete values, let's consider the  $p_i = (50, 30, 10)$ , and the  $v_{j1} = (9, 6, 2)$  vectors. The cosine between two vectors gives 0.99. This is due to the fact that the demands are proportional to the residual resources. However, if we consider another demands vector  $v_{j2} = (5, 20, 7)$ , which is disproportionate with the residual resource, the cosine similarity in this case, is measured to 0.59.

The second measure of similarity is the weighted Jaccard distance, which is illustrated in formula (5).

$$jacc(\vec{p}_i, \vec{v}_j) = \frac{\sum_{r \in RS} \min(\mathbf{RN}_i^r, \mathbf{DN}_j^r)}{\sum_{r \in RS} \max(\mathbf{RN}_i^r, \mathbf{DN}_j^r)} \quad (5)$$

The formula measures a ratio of commonality between the element's values of the two vectors as depicted in TABLE I. The values of the similarities lie between 0 and 1, where 1 indicates a perfect similarity and 0 indicates a lack of similarity.

	CPU	RAM	BW
$PM_i$	0.8	0.6	0.9
$VM_{j1}$	0.3	0.1	0.2
$VM_{j2}$	0.4	0.1	0.3

TABLE I  
JACCARD SIMILARITY

An illustrative calculation of Jaccard similarity for the vectors of TABLE I is done as follows:

$$Jacc(PM_i, VM_{j1}) = \frac{0.3+0.1+0.2}{0.8+0.6+0.9} = \frac{0.6}{2.3}$$

$$Jacc(PM_i, VM_{j2}) = \frac{0.4+0.1+0.3}{0.8+0.6+0.9} = \frac{0.8}{2.3}$$

### C. Energy-aware selection criterion

With the aim of reducing the power consumption and resource wastage simultaneously, the destination host selection criterion for a VM  $v_j$  is given in formula (6) as in [7].

$$P_{best} = \underset{p_i \in P}{\text{Argmax}} \left[ \text{similarity}(\vec{p}_i, \vec{v}_j) \cdot e^{-GE_{p_i}^{+v_j}} \right] \quad (6)$$

Where  $\text{similarity}(\vec{p}_i, \vec{v}_j)$ , in our case, is given either by equation (4) or (5). The idea of this criterion is that the best PM for a given  $v_j$ , will have the largest similarity vector  $\text{similarity}(\vec{p}_i, \vec{v}_j)$ , and the lowest growth rate of energy consumption:  $e^{-GE_{p_i}^{+v_j}}$ .

$$GE_{p_i}^{+v_j} = \frac{EC_{p_i}^{+v_j} - EC_{p_i}}{EC_{p_i}} \quad (7)$$

Where  $EC_{p_i}$  represents the energy consumption of  $p_i$ , and  $EC_{p_i}^{+v_j}$  is the energy consumption of  $p_i$  after that  $v_j$  is placed on it. In many studies [1], [17], [18], the energy consumption of a PM is defined as a linear function of CPU utilization, as shown in formula (8).

$$EC_{p_i} = EC_{p_i}^{idle} + (EC_{p_i}^{max} - EC_{p_i}^{idle}) \times U_{p_i} \quad (8)$$

Where  $EC_{p_i}^{idle}$  is the energy consumption of  $p_i$  in the idle state,  $EC_{p_i}^{max}$  is its maximum power consumption when the CPU is fully utilized, and  $U_{p_i}$  is its current normalized CPU utilization.

## IV. IRB-VMP ALGORITHM

In this section, we describe in detail our proposed intra-resource balancing algorithm for VMP (IRB-VMP). Then, we report the different metrics used to evaluate the performance of this algorithm.

### A. Algorithm description

Our proposed algorithm is based on a greedy heuristic approach, and aims to minimize at the same time energy consumption and resource wastage in a data center.

To achieve this goal, we first sort the list of PMs in descending order of their available resource capacity (see line 1 of Algorithm 2), which is represented as a magnitude vector like in formula (9).

$$|\vec{p}_i| = \sqrt{\sum_{r \in RS} (\mathbf{RN}_i^r)^2} \quad (9)$$

The algorithm runs the outer loop in order to find for each PM the most suitable VM candidate with respect to resource similarity and energy consumption. In other words, among  $VMsList$  (the *for* loop in line 3 of Algorithm 2), we select the one which has the trade-off between the maximum value of similarity and the drop of growth energy (see line 5 to 10 of Algorithm 1).

Then, we map the VM candidate to the PM while removing it from  $VMsList$  and updating the resource of the PM. When there is enough resource on the current PM, the

---

**Algorithm .1:** Selection of  $vm$  candidate

---

```
1 Function SelectOfVmCandidate  $pm,$   
    $VMsList$  ;  
2    $maxOPT = -\infty$   
3   foreach  $vm \in VMsList$  do  
4     if  $enoughResource(pm, vm)$  then  
       /* calculate cosine or jaccard  
         similarity */  
5      $resSim = Similarity(pm, vm)$   
       /* Calculate drop of growth  
         energy as in formula 7 */  
6      $dropEgy = \exp(-GE(pm, vm))$   
       /* calculate trad-off between  
          $resSim$  and  $dropEgy$  as in  
         formula 6 */  
7      $SimDE = resSim \times dropEgy$   
8     if  $maxOPT < SimDE$  then  
9        $maxOPT = SimDE$   
10       $vmCandidate = vm$   
11     end  
12   end  
13 end  
14 return  $vmCandidate$ 
```

---

---

**Algorithm .2:** IRB-VMP ALGORITHM

---

**Input:**  $PMsList, VMsList$

**Output:**  $MappingMatrix$

```
1 Sort  $PMsList$  by descending order of residual  
   resources  
2 foreach  $pm \in PMsList$  do  
3    $vmCandidate = SelectOfVmCandidate(pm, VMsList)$   
4   while  $enoughResource(pm, vmCandidate)$  and  
   ( $VMsList$  is not empty) do  
5      $Mapping.add(pm, vmCandidate)$   
6      $VMsList.remove(vmCandidate)$   
7     - Update resource capacity of  $pm$  as described  
       in formula 1  
8      $vmCandidate = SelectOfVmCandidate(pm, VMsList)$   
9   end  
10 end  
11 return  $MappingMatrix$ 
```

---

above-mentioned procedure is repeated to select the next VM candidate. When the resource of the current PM is exhausted, the algorithm goes to the next PM and so on, until all the VMs are mapped.

### B. Performance metrics

In order to determine the efficiency of our proposed algorithm, we consider three metrics, which are:

1) *The number of active PMs:* this metric shows how many PMs are required to host all the VMs and has a direct impact on the energy consumption of a data center.

2) *Total resource wastage:* we use this metric to measure how our VMP solution utilizes the different resources of a PM. For a given PM,  $i$ , the local resource wastage is calculated as in formula (10) like in [8].

$$RW_i = \frac{\sum_{r \in RS} [RN_i^r - \min(RN_i)] + \varepsilon}{\sum_{r \in RS} (1 - RN_i^r)} \quad (10)$$

Where  $\min(RN_i)$  indicates the minimum normalized remaining resource among all resource types of the PM  $i$ . Whereas  $\varepsilon$  is a small positive real number, that we set to a value as in [8] to 0.0001. The total resource wastage in the whole data center is then obtained by summing the resource wastage overall the PMs, as it is mentioned is the equation (11).

$$RW^{tot} = \sum_{i=1}^n RW_i \quad (11)$$

3) *Total energy consumption:* this metric depends on the aforementioned metrics. It is defined as given in the equation (12).

$$EC^{tot} = \sum_{i=1}^n EC_{p_i} \quad (12)$$

## V. PERFORMANCE EVALUATION

This section describes the evaluation experiments that have been carried out to validate our IRB-VMP algorithm.

### A. Simulation setup

We consider a data center that consists of PMs with three types of resources: CPU, RAM and BW. The capacity resource of each type is randomly generated as in [8] from the ranges:

- CPU: [4000-10000] MIPS
- RAM [4096-16384] MB
- BW [2000-10000] MB/S

To simulate the energy consumption by each PM, we use the formula (8). The maximum power consumption is generated randomly in the range [100-300] Watts. To get the power consumption in the idle state, we multiply the random maximum power by 0.7. This implies that the energy consumption of an idle PM is equal to 70% of its full utilization state, as it is mentioned in [4].

On the other hand, we consider a set of VMs. The characteristics of their resource type are given in TABLE V-A.

CPU [MISP]	RAM [MB]	BW [MB/S]
{500, 750, 1000, 1500}	{256, 512, 1024, 2048}	generated radomly from {100-1000}

TABLE II  
CONFIGURATION OF VM INSTANCES

Our simulations are implemented in Python language and were carried out on a personal computer with Intel Core i5 CPU, 1.6 GHz processor, and 16 GB RAM.

### B. Comparison of algorithms

To evaluate the performance of our proposed algorithm, we compared its metrics as indicated in section IV-B with the following state-of-the-art algorithms:

1) *First Fit Decreasing(FFD)* [10]: is one of the classical commonly used algorithm to study the effectiveness of the new proposed VMP algorithms. It is based on the idea of sorting the VMs into their decreasing order of CPU demands, and then each VM is placed into the first PM with enough resource capacity.

2) *Best Fit Decreasing(BFD)* [11]: like FFD, this algorithm sorts VMs decreasingly with respect to their CPU requirements. Then for each VM, it searches the PM where it fits the tightest in term of CPU resource.

### C. Results and discussions

To investigate the potential scalability of our algorithm, we conducted our experiments with different sizes of VMs [200, 300, 400, 500] whereas the number of PMs is fixed to 100. Several random tests have been carried out, the obtained results are depicted in Fig. 2.a, 2.b and 2.c.

As we can see from the results of Fig.2, our IRB-VMP algorithm outperforms the FFD and BFD algorithms in terms of all the considered metrics.

From Fig. 2.a, it is clear that our algorithm reduces the number of active PMs as compared to the other algorithms. We notice a reduction of 7 to 15 active PMs compared to FFD. The BFD algorithm shows the worst performance among the others. It uses an extra of 15 to 23 active PMs compared to our IRB-VMP algorithm. This reflects its mapping strategy, where it selects PMs with the lowest CPU resource capacity which leads to the need for a large number of PMs.

Focusing on Fig. 2.b, we illustrate the total energy consumption. From the results, we can conclude that the proposed IRB-VMP algorithm significantly consumes less energy than the other algorithms. An improvement of more than 2.2 Kwatts is noticed compared to BFD in the environmental setting with 500 VMs. This is due mainly to our Energy-aware selection criterion (see section III-C) which preferentially places the VMs on the PMs with the least energy consumption growth.

As presented in Fig. 2.c, we can obviously see that our proposed algorithm dramatically reduces the resource wastage, compared to FFD and BFD algorithms. As the number of VMs increases, the degree of reduction becomes more important, especially when cosine similarity is used (a reduction of about 11 to 20 over BFD). This amount of improvement is expected because our algorithm places VMs on PMs by relying on the similarity between resource vectors while the other algorithms do not consider it.

Also, we notice that using cosine similarity is more advantageous than Jaccard one. This advantage is more emphasized, when the number of VMs increases. For example, our IRB-VMP algorithm based on cosine similarity uses in average 24.8 PMs to host 200 VMs, against 25.7 when the Jaccard similarity is considered. However, to host 500 VMs, using cosine

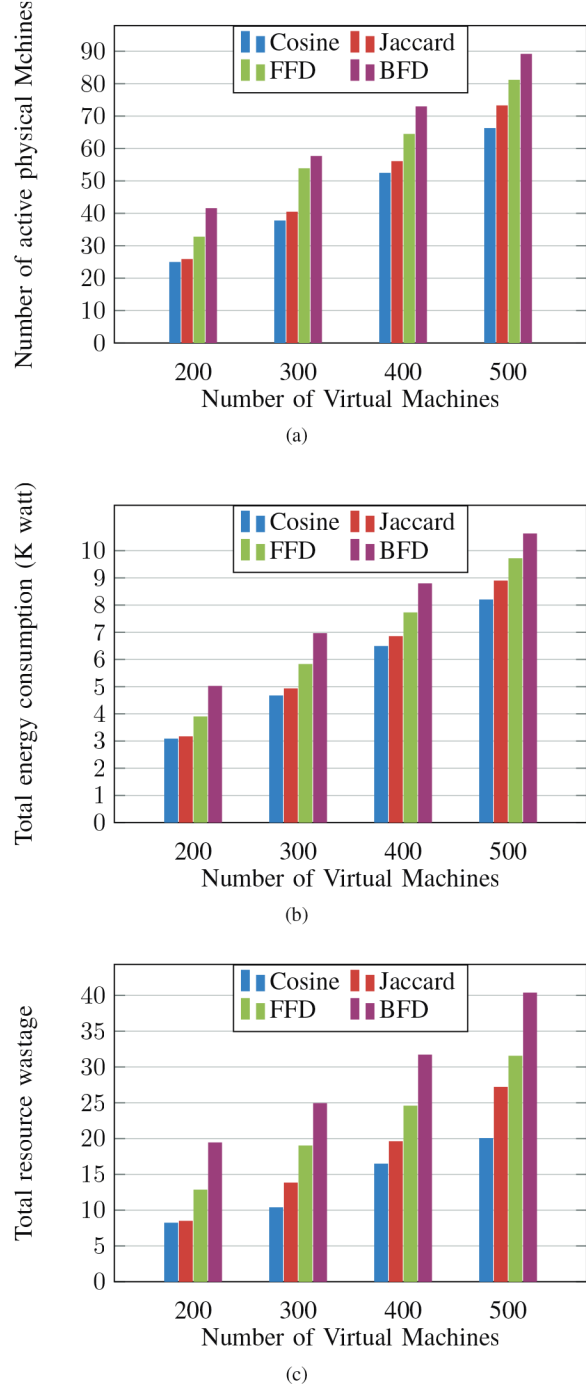


Fig. 2. Simulation results

similarity allowed us to minimize 7 active PMs compared to Jaccard similarity (an average of 66.1 vs 73.1 PMs).

## VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed a new VM placement algorithm based on statistical similarity measures. To efficiently exploit the available resources, the cosine and Jaccard similarities are calculated between the residual resource vectors of PMs and the requested resources for VMs. The destination host selection strategy is reinforced by an energy-aware criterion to ensure a reduction of the power consumption and the resource wastage simultaneously.

We evaluated the performance of our proposed algorithm against the state-of-the-art FFD and BFD algorithms through simulation experiments. The obtained results show that our IRB-VMP algorithm improves significantly the number of active PMs, the total energy consumption as well as the resource wastage. As future work, we want to investigate the use of machine learning methods so that to predict the load of PMs, that can allow a more efficient VM placement.

## REFERENCES

- [1] H. Talebian, A. Gani, M. Sookhak, et al, "Optimizing virtual machine placement in IaaS data centers: taxonomy, review and open issues." *Cluster Comput* 23, 837–878, 2020.
- [2] S. Shigeta, H. Yamashima, et al, "Design and implementation of a multi-objective optimization mechanism for virtual machine placement in cloud computing data center" In: *Proceedings of the International Conference on Cloud Computing*, pp. 21–31. Springer, Cham, 2013.
- [3] 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" *Concurr. Comput. Pract. Exp.* 24(13): 1397-1420, 2012.
- [4] R. Buyya, CS. Yeo et al. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility" *Future Generation Computer Systems*, 25(6):599–616, 2009.
- [5] R. M. Nauss, "Solving the generalized assignment problem: An optimizing and heuristic approach" *INFORMS Journal on Computing*, vol. 15, no. 3, pp. 249–266, 2003.
- [6] L. P Fabio, and B. Baran, "Virtual machine placement literature review." *arXiv preprint arXiv:1506.01509*, 2015.
- [7] L. Zhihua, G. Shujie and Y. Lei, "Evidence-Efficient Affinity Propagation Scheme for Virtual Machine Placement in Data Center" *IEEE Access*, vol. 8, 2020.
- [8] S. Azizi, M. Shojafar, et al. "GRVMP: A Greedy Randomized Algorithm for Virtual Machine Placement in Cloud Data Centers". in *IEEE Systems Journal*, 2020.
- [9] M. Masdari, S. Nabavi and V. Ahmadi "An overview of virtual machine placement schemes in cloud computing." *J. Netw. Comput. App.*, vol.66, pp. 106-127, 2016.
- [10] M. Yue, "A simple proof of the inequality  $FFD(L) \leq 11/9 OPT(L) + 1$ ,  $L$  for the FFD bin-packing algorithm." *Acta Mathematicae Applicatae Sinica* vol.7,no.4, pp. 321–331, 1991.
- [11] A. Beloglazov, J. Abawajy and R. Buyya "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing". *Future Generation Computer Systems*, 2011.
- [12] A. Richa, M. Mitzenmacher and R. Sitaraman "The power of two random choices: A survey of techniques and results". *Combinatorial Optimization*, vol. 9, pp. 255-304, 2001.
- [13] R. Regaieg, M.Koubaa et al. "Multi-objective optimization for VM placement in homogeneous and heterogeneous cloud service provider data centers". *Computing*, pp. 1-25, 2021.
- [14] W. Lin, C. Zhu, et al. "Novel algorithms and equivalence optimisation for resource allocation in cloud computing". *International Journal of Web and Grid Services*, vol. 11 no. 2, pp. 193-210, 2015.
- [15] J. YE. "Cosine similarity measures for intuitionistic fuzzy sets and their applications". *Mathematical and computer modelling*, vol. 53, no. 1-2, pp. 91-97, 2011.
- [16] S. Niwattanakul, J. Singthongchai, and al. "Using of Jaccard coefficient for keywords similarity". In *Proceedings of the international multicongference of engineers and computer scientists*, Vol. 1, No. 6, pp. 380-384, 2013.
- [17] M. K Gupta., T. Amgoth, "Resource-aware virtual machine placement algorithm for IaaS cloud". *The Journal of Supercomputing*, vol. 74, no. 1, pp. 122-140., 2018.
- [18] S. Farzai, M.H Shirvani, and M. Rabbani, "Multi-objective communication-aware optimization for virtual machine placement in cloud datacenters". *Informatics and Systems*, vol. 28, no.", pp. 100374., 2020.