

# WPSP: a multi-correlated weighted policy for VM selection and migration for Cloud computing<sup>\*</sup>

Sergi Vila<sup>1</sup>[0000-0002-4536-5025], Josep L. L rida<sup>1</sup>[0000-0002-9352-9338], Fernando Cores<sup>1</sup>[0000-0003-2910-6709], Fernando Guirado<sup>1</sup>[0000-0003-1352-9555], and Fabio L. Verdi<sup>2</sup>[0000-0002-5455-8910]

<sup>1</sup> Universitat de Lleida, INSPIRES, Lleida, Spain

{svila, jlerida, fcores, f.guirado}@diei.udl.cat

<sup>2</sup> Federal University of S o Carlos, Department of Computer Science, Sorocaba, Brazil

verdi@ufscar.br

**Abstract.** Using virtualization, cloud environments satisfy dynamically the computational resource necessities of the user. The dynamic use of the resources determines the demand for working hosts. Through virtual machine (VM) migrations, datacenters perform load balancing to optimize the resource usage and solve saturation. In this work, a policy, named WPSP (Weighted Pearson Selection Policy), is implemented to choose which virtual machines are more suitable to be migrated. For each VM, the policy evaluates both the CPU load and the Network traffic influence on the assigned host. The corresponding Pearson correlation coefficients are calculated for each of the VMs and then weighted in order to provide a relationship between the values and the host behavior. The main goal is to clearly identify and then migrate the VMs that are responsible of the Host saturation but also considering their communications. Using the CloudSim simulator, the policy is compared with the rest of heuristic techniques in the literature, resulting in a reduction of 89% in the number of migrations, and thus reducing the use of bandwidth (5%), network saturation (20%) and over-saturated hosts (51%). Additionally, an improved VM allocation technique to reduce the distance the VMs must travel in order to be migrated is presented, obtaining an average reduction of 87% in the quantity of migrated data.

**Keywords:** Cloud computing · planning · virtual machines · migrations · Pearson correlation coefficient · load balancing · CloudSim.

## 1 Introduction

Cloud Computing has become an effective alternative to local servers for many users, whether to allocate the resources of companies or to compute scientific programs in research centers. It provides dynamic and scalable virtualization

---

<sup>\*</sup> This work has been possible thanks to MEIyC-Spain TIN2017-84553-C2-2-R and S o Paulo Research Foundation (FAPESP) under the grant #15/19766-9.

resources through a network service and forms a virtual computing resource pool allocated to a classic data center. Thus, it is possible to combine the hosts' capacity on an on-demand pay-per-cycle basis, guaranteeing the defined Service Level Agreement (SLA) to the users.

However, variability in the request-rate from the cloud service consumers at any given time can seriously affect not only the Quality of Service (QoS) but also the SLA. The applications, and particularly the network-intensive ones, often need to communicate frequently, and the network I/O performance would affect the overall VM performance notably. In this situation, hosts become overloaded and unable to resolve all the requests, negatively impacting the SLA.

In the literature, there are many methods to avoid overloading hosts in data centers. Some of these are based on load balancing strategies that facilitate the distribution of the workload equally over the available resources [8]. Other proposals apply VM migration to provide the required resources to the VMs responsible for the host overload. However, the migration process can produce unexpected network latency or congestion that becomes critical for achieving and maintaining the performance of the application. That is why the migration process requires correctly identifying a candidate migratable VM that ensures not only the host load reduction but also keeps use of the inter-VMs communication links contained, thus avoiding an SLA violation.

In the present paper, we propose the use of the Pearson correlation factor to correctly identify those VMs that are seriously affecting the host overload, taking into account both the computational and communication resource usage. The proposed method not only tackles the current host state, but also evaluates the previous states captured during the host execution timeline. When the VM is identified, the migration process moves it to another host, releasing the corresponding resources and effectively reducing the host overload. Moreover, our proposal incorporates a weighting factor that provides a much closer relationship between VM correlation and Host behaviour. To avoid unnecessary VM migrations, our proposal attempts to find the balance between the quantity of CPU released by a VM and the communication affinity with the rest of the VMs within the host.

The experimental results have been compared with the most well-known heuristic methods from the literature, and demonstrate that our proposal improves the host usage avoiding the overload and also reducing the global number of VM migrations.

The rest of the work is organized as follows: in Section 2, the state of the art used for the present work is described. Section 3 presents the VM selection policy. Section 4 contains the experimental study, and finally, the conclusions and future work are discussed in Section 5.

## 2 State of the art

In the literature, there are many works related to the virtual machine migration process. Raja et al. in [2] present a survey of VM migration and server consoli-

dation. They evaluated multiple migration schemes and they took into account different parameters to compare them. Their conclusions pointed to the fact that unnecessary and uncontrolled migrations are the main reason for SLA violation. Most of the proposed solutions to initiate the migration process were based on processing discrete data-captures to evaluate the QoS while others were based on applying machine learning-based adaptive thresholds. In the present paper, we propose an effective correlation-based method with data obtained from tracking the host execution time-line taking snapshots periodically.

The correlation between two sets of data is a statistical measure that calculates the strength of the relationship between the relative values of two variables. There are many studies in the literature applied to different fields of knowledge that demonstrate the importance of correlation between multiple parameters for taking correct decisions. Douglas et al. in [4] and Winter et al. in [15] compared some correlation factors and their quality. In the current paper, given the continuous nature of the variables (CPU and network load values) and the sensitivity to variations in the differences between the sample values, we decided to focus on the well-known Pearson correlation coefficient.

There are different works using correlation coefficients applied to Cloud Computing. Choudhary et al. in [6] was based on Spearman's Rank Correlation Coefficient to select the optimal VM according to the present workload and datacenter resources availability to reduce the energy consumption. The results obtained, compared with the VM Random Selection, demonstrated lower energy consumption while maintained the required SLA. Moghaddam et al. in [9] proposed a VM selection algorithm focused on energy reduction and also considering the SLA parameter. The algorithm was based on the Pearson correlation coefficient and was used to determine both VMs' CPU utilization and the correlation with their co-hosted VM. Their proposal was evaluated through simulation in the CloudSim environment, using two different real Cloud data sets by the CoMon project (PlanetLab) and Google. The results show that the correlation improves the VM identification as migratable and reduces the energy consumption. Sun et al. in [13] addressed the problem of online migration of multiple correlated VMs among multiple datacenters. This work was focused on the optimization of migration performance. The authors treated both bandwidth and routing required for the VM migration process and use the correlation to determine those VMs that must be migrated all together. The results reduced the remapping cost and the average migration time and downtime of the VMs.

Our proposal differs mainly from previous works in the fact that we use the correlation coefficients to determine the influence of the VM on the resource usage of the allocated host. We evaluate periodically both computing and communication load for each allocated VM. When an overloaded host has been identified, our method determines the VM candidate to be migrated. Applying our proposal, the overall migrations were reduced, thus reducing the network saturation, increasing the host utilization and without compromising the SLA.

In [1], Abdelsamea et al. presented a host saturation algorithm based on multiple regressions (CPU, RAM and Bandwidth), decreasing the energy con-

sumption and SLA. Additionally, they combined Local Regression (LR) with Loess' method to develop a hybrid version of their algorithm. The results show that the implemented algorithms have better results for energy but obtains worse SLA violation results due to being inversely correlated to energy. Ali [7] et al. presented a Weighted Linear Regression algorithm for resource prediction, CPU, RAM and network bandwidth. The algorithm was compared with other detection techniques in the literature shows reductions in the energy consumption while providing a high level of commitment to the SLA, maintaining a similar level of migrations.

In the present paper, we evaluated not only the similarity in behavior between VMs and Hosts, represented by the correlation factor, but also the influence that these VMs had on generating this behavior. Thus, we propose the use of a weighting factor applied to the correlation factors that allows to identify the most suitable VM to be migrated. Our proposal is also combined with a new assignment method with the main aim of decreasing the distance to be crossed in the migration process and then reducing the network utilization.

### 3 Problem Statement

The policy presented in this paper, hereinafter referred to as Weighted Pearson Selection Policy (WPSP), is based on three main ideas: (1) evaluating the host execution in the time-line to determine the resource usage behavior of each one and detect the overload situations, (2) evaluating the use of both computing (in terms of CPU usage) and communication (in terms of data transfer volume within the host) VM resource usage to correctly identify the VMs closely related to the host overload, and (3), applying a weighting process to the volume of CPU and network used by the VMs in relation of their host to adjust the obtained correlations, finally defining which VMs are provoking host saturations. Our first goal is to obtain knowledge of the host load during their execution. This information is acquired from snapshots taken of the system periodically. These snapshots contain information about the resources required by VMs and the resources really assigned by the hosts.

The second core element of our proposal is to determine the VM that has the greatest influence on the overloaded hosts' resource usage. Each host allocates multiple VMs and each with different resource requirements. It must be taken into account that some of these VMs can be related to the same service so that migrating any VM does not ensure the reduction of overload as the external host communication can increase due to the new VM allocation. For this, we propose to consider both computation and communication resource usage to identify their influence on the host overload and determine the relationship between the VMs inside the host.

The idea behind the use of the Pearson correlation is to determine the similarities between the CPU and network resources usage between the host and each VM, with the aim of identifying the VMs with a wider impact on the host resource usage. Knowing which VMs are the most influential, we can migrate those

causing the biggest impact on the release of resources but triggering a smaller number of migration. The correlation coefficients assume  $n$  samples of two variables,  $x$  (host) and  $y$  (VM). The Pearson correlation coefficient is calculated by Eq. 1, where  $\bar{x}$  and  $\bar{y}$  represent the arithmetic mean of  $x$  and  $y$ , respectively. In addition, each pair of values corresponding to the same point in time cannot be altered so as to maintain the consistency of the coefficient obtained.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

The correlation coefficient between two ordered sets of values measures the strength of the relationship between the relative movements of the two variables. The values range is  $[-1, 1]$ . A positive correlation means that if one variable increases, the other variable also tends to increase. A negative correlation means that if one variable increases, the other variable tends to diminish. The weakest linear relationship is indicated by a correlation coefficient equal to 0.

In this paper,  $rcpu$  measures the relationship between the Host CPU usage ( $x$ ) and a VM's CPU usage ( $y$ ), in the same way  $rnet$  measures the relationship between the Host internal communications ( $x$ ) and a VM's data transfers to other VMs inside the same Host ( $y$ ), both calculated by Eq. 1. The main aim of the WSPSP Policy is to identify the candidate VMs to be migrated to eliminate the host saturation with the minimum VM migrations. In order to meet these objectives, we consider that the VM with the highest positive  $rcpu$  is the best candidate to be migrated with the aim of reducing host saturation. However, in the case that this VM also has a high positive  $rnet$ , the migration of the VM will produce an increase in data transfers through the external communication channels, thus fostering the saturation of these channels and producing a negative impact on global performance. To prevent this occurring, we should consider the migration of VMs with a high positive CPU correlation  $rcpu \simeq 1$ , but with a weak network correlation  $rnet$ . While there is the possibility of migrating a group of VMs highly correlated with internal communications, this option substantially increases the number of migrations and their cost.

Normalization is the process through which a set of values  $V$ , ranging from  $[\min(V), \max(V)]$  are scaled to  $[0, 1]$ . In our policy, the CPU and BW usage of each VM is normalized in this sense. These are essential values ( $ncpu$  and  $nnet$ ) for evaluating the role of each VM inside the host, and in collaboration with the correlation value, we can figure which VMs are the most influential in terms of volume and oscillations over time. Eq. 2 shows how the  $ncpu$  and  $nnet$  values are calculated, where  $x$  is the value (CPU or network) of the VM,  $x_{max}$  the highest value among the VMs of the same host and  $x_{min}$ , the minimum.

$$n = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

With the aim of considering the VMs' consumption of both CPU and communication resources, we propose a heuristic function computed by Eq. 3. This provides each VM with a value based on the magnitude of both the  $rcpu$  and  $rnet$  correlation coefficients and the  $ncpu$  and  $nnet$  ponderation values.

$$hval = \frac{1 - (w * rcpu * ncpu)}{(1 + (w * rcpu * ncpu) - (w * rnet * nnet))} \quad (3)$$

The heuristic function allows the relationship between both correlation coefficients and weighted values to be modelled providing a mechanism to compare the VMs within a Host. The resulting value *hval* defines the migration priority for each VM, the lowest value being the best option. Additionally, after a specific threshold, *hval.th*, the VMs are not allowed to be migrated, as can be seen in Figure 1. The *w* variable defines the slope of the *hval* function, and thus controls the *hval* value scale. With a value of  $w = 1$ , the *hval* function tends to 0 irrespective of the values used. On the contrary, with  $w = 0$  the resulting values tend to 1. The tuning of the *w* variable can be useful in exceptional cases with the values located in a bunch. However, with  $w = 0.5$ , the resulting values are bounded in a smooth curve that allows diversity.

By way of example, Table 1 shows the corresponding *hval* value for a set of VMs with different combinations of *rcpu*, *rnet*, *ncpu* and *nnet* values. Figure 1 shows the *hval* value for each VM and their location on the plane. It shows the contour lines projected by the *hval* function on the plane formed by *rcpu*, *rnet*, *ncpu* and *nnet* values. We established the premise that the VMs directly related to the CPU usage and have a high weight are good candidates to be migrated, provided that they are also weakly correlated to the internal network communication.

We can observe this is the case for VM4, obtaining the minimum *hval* value. It has notable CPU values, having a correlation of 0.7 and a big usage of CPU with a weight of 1, on the contrary, the network usage is small. VM0 is in the same case than VM4 but, even has more CPU impact and network load, so it obtains a bigger *hval* value. By counterpart, VMs 1, 2 and 3 have no chances to be migrated. VM1 due to it shows low weights, VM2 due to its network usage is too much valuable than the CPU load, that is too low, and VM3 is near the threshold, however CPU usage should be higher. These results show that the *hval* function can be used to prioritise the candidate VMs to be migrated.

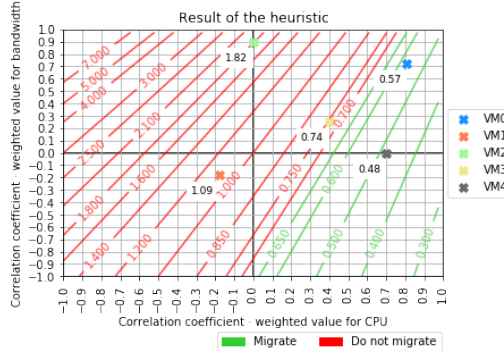
### 3.1 Weighted Pearson Selection Policy

The WPSP policy proposed in this paper is represented by Algorithm 1. The WPSP policy is executed when the saturation of one of the hosts is detected. The algorithm is executed until the saturation is solved or until no more VMs are selected for migration. First, the VMs able to be migrated are obtained (line 2). Then, the Pearson correlation coefficients are obtained for each VM, in relation to the CPU (*rcpu*) and network (*rnet*) usage in the host  $\mathcal{H}$  (lines 4-5). In the case of the *rnet* correlation coefficient, only communications between virtual machines within the same host are taken into account. In lines 6-7, the weighting values *ncpu* and *nnet* are obtained. Next, we obtain the heuristic value *hval* for each VM (line 8). Finally, the VMs with the minimum *hval* are selected for migration until the saturation problem is solved or no more VMs are eligible

**Table 1.** Example. *hval* results for *rcpu* and *rnet* correlation classes

Label	<i>rcpu</i>	<i>rnet</i>	<i>ncpu</i>	<i>nnet</i>	<i>hval</i>
VM0	d (+0.9)	d (+0.9)	s (0.9)	s (0.8)	0.57
VM1	i (-0.9)	i (-0.6)	w (0.2)	w (0.3)	1.09
VM2	w (-0.3)	d (+0.9)	w (0.0)	s (1.0)	1.82
VM3	w (+0.5)	w (+0.5)	s (0.8)	w (0.5)	0.74
VM4	d (+0.7)	w (+0.2)	s (1.0)	w (0.0)	0.48

s=strong, w=weak, d=direct, i=indirect



**Fig. 1** *hval* function representation and example results

(*hval* values exceed the WPSP decision threshold *hval.th*). When no VMs meet this criterion, a null value is returned.

### 3.2 Minimum Distance Group VM Allocation policy

To complement our selection policy, WPSP, being focused on the reduction of migrations, we are interested in an allocation policy that helps it to succeed on this task. We propose an improvement based on the distance the VMs must travel across the network. The "CloudSim most efficient host" allocation technique is improved by applying the Minimum Distance Group, MDG, that is, instead of selecting the most efficient suitable host from the whole datacenter, the hosts are grouped and ordered by number of jumps inside the network, trying to migrate the VM to the nearest group. If this is not possible, the next nearest group is tested each time until it can be allocated, as can be observed in Algorithm 2. If the VM does not fit any host, that is, trying to allocate it at full CPU usage, the migration is not performed. In general, any criteria for the selection function can be implemented. For this experimentation, the *getPowerAfterAllocation* function provided by CloudSim was selected. This returns the host with the lowest increment in its energy consumption after the VM is located.

## 4 Experimentation and results

This section describes the configuration of the experimental environment, mainly based on the CloudSim simulator [5] and the results obtained.

### 4.1 Experimental setup

The CPU load traces used are part of the PlanetLab environment. They are obtained with the CoMon monitoring system [11]. There is a set of traces corresponding to 10 days of execution with around 1000 virtual machines. The first

---

**Algorithm 1** Weighted Pearson Selection Policy - VM selection algorithm

---

**Require:**  $\mathcal{H}$ : Overloaded host,  $VM_{\mathcal{H}}$ : set of VMs  $\in \mathcal{H}$ **Ensure:**  $SVM$ : Selected migratable VMs

```

1: declare  $MVM$ : migratable VMs area,  $rcpu$ : current cpu correlation values,  $rnet$ :
   current network correlation values,  $ncpu$ : current cpu weighted value,  $nnet$ : current
   network weighted value,  $Hval$ : heuristic values for each  $vm \in MVM$ ,  $hval.th$ :
   WPSP decision threshold
2:  $MVM \leftarrow \text{getMigratableVMs}(VM_{\mathcal{H}})$ 
3: for each  $vm \in MVM$  calculate
4:    $rcpu \leftarrow \text{calculate\_cpu\_correlation}(\mathcal{H}, vm)$ 
5:    $rnet \leftarrow \text{calculate\_net\_correlation}(\mathcal{H}, vm)$ 
6:    $ncpu \leftarrow \text{calculate\_cpu\_weight}(MVM, vm)$ 
7:    $nnet \leftarrow \text{calculate\_net\_weight}(MVM, vm)$ 
8:    $Hval^{vm} \leftarrow \text{calculate\_hval}(rcpu, rnet, ncpu, nnet)$ 
9: end for
10: while  $isSaturated(\mathcal{H})$  do
11:    $vm \leftarrow \text{Select } vm \in MVM \text{ with } \min(Hval) \mid Hval \leq hval.th$ 
12:   if  $vm$  is  $NULL$  then
13:      $break$ 
14:   end if
15:    $MVM \leftarrow MVM - vm$ 
16:    $SVM \leftarrow SVM \cup vm$ 
17: end while
18: return  $SVM$ 

```

---



---

**Algorithm 2** MDG - VM allocation algorithm

---

**Require:**  $\mathcal{H}$ : Overloaded host,  $\mathcal{VM}$ : VM to be migrated,  $\mathcal{SH}$ : Set of Hosts -  $\{\mathcal{H}\}$ 

```

1:  $bestHost = h_i \in \mathcal{SH} \min(\text{distance}(\mathcal{H}, h_i)) \cap \min(\text{getPowerAfterAlloc}(\mathcal{VM}, h_i))$ 
2: return  $bestHost$ 

```

---

150 files of the trace 20110303 are the ones used for the experimental study. Each contains 288 values corresponding to a day of performing. The PlanetLab traces are updated every 5 minutes. This determines the snapshots ratio in which VMs and Hosts data is obtained in order to evaluate the correlation coefficients.

Table 2 shows the virtual machine configuration. Each VM contains one cloudlet acting as endless tasks, whose percentage of CPU load being defined by the PlanetLab traces. Table 3 shows the main characteristics of the hosts used based on those present by default in CloudSim.

An interaction is defined as the communication between two VMs throughout the simulation. The network traces, which represent the VM interactions, were generated using the FNSS tool [12], obtaining sin cyclo-stationary traffic ( $\sigma = 0.8$ ,  $\log\psi = -0.33$ ) according to [10]. This traffic has an equivalent behaviour to the Sprint Europe network [14]. We assume a limit of 3 Mbps for the bandwidth use for each one. Throughout the simulation, the values of the interactions are updated using the network traces. We defined three different types of interaction: low, with an occurrence of 50% and ranged between 0 and



**Table 2.** Virtual machine characteristics

Type	# CPUs	MIPS	BW (Mbps)	RAM (MB)	Quantity
Tiny	1	750	100	870	100
Small	1	1500	100	1740	100
Medium	1	3000	100	1740	100
Large	1	3750	100	613	100

**Table 3.** Host characteristics

Type	Model	# CPUs	MIPS	BW (Mbps)	Quantity
Small	HP ML110 G4	2	1860	1000	21
Large	HP ML110 G5	2	2660	1000	15

0.6 Mbps; medium, 30% of occurrences, ranged between 0.6 and 1.8 Mbps; and high, with an occurrence of 20%, ranged between 1.8 and 2.85 Mbps.

It was also established that 15% of interactions occur within the same host, while only 0.05% occur outside the host. In order to test the VM preservation capabilities of our policy, and after balancing CPU and network resources, we determined that an initial 15% of internal communications offers enough traffic to maintain the VMs in the same hosts but with opportunities to leave them depending on the CPU load. During the simulation, and due to migrations, these percentages varied, increasing the number of external communications and reducing the internal ones.

The interconnection topology, where central nodes are switches, the leaf nodes being hosts, is shown in Figure 2.

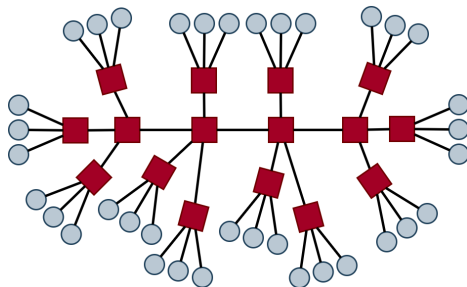
**Fig. 2.** Topology

Table 4 shows a summary of the most important CloudSim configuration parameters used during the experimentation. For each technique, 30 experiments were carried out varying the initial placement of VMs in the hosts, which affected the number and typology of the interactions among the VMs. The metrics analysed in the present work are presented in Table 5.

The WPSP threshold,  $hval.th$ , was set at 0.65 after analysing which value obtains the lowest ratio of unsatisfied/satisfied MIPS. Due to space reasons, this analysis is not presented.

**Table 4.** Configuration parameters of CloudSim

Parameter	Value	Parameter	Value
CPU trace	PL 20110303	BW between Host-VM	100 Mbps
% Internal interactions	15%	BW between Host-Switch	1000 Mbps
% External interactions	0.05%	CPU Saturation limit	70%
# of switches	16	Simulation time	86400 s
# of hosts	36	# of experiments per technique	30
# of virtual machines	400	Host saturation detection technique	CloudSim IQR
Window size	6	VM allocation technique	MDG
Max. BW of the interactions	3 Mbps	Underutilised host shutdowns	Disabled

## 4.2 Virtual Machine Selection Policies

The selection policy selects the VM candidate to be migrated. CloudSim’s default techniques were used with the aim of compare them with the policy proposed in the present work. The techniques [3] used in the comparison are the following:

- Random Search (RS): Among the candidate VMs to be migrated, one is chosen randomly.
- Minimum Migration Time (MMT): Chooses the VM that requires the least RAM memory.
- Minimum Utilisation (MU): Chooses the VM which requested fewest MIPS during the simulation.
- Maximum Correlation (MC): A linear regression is generated transposing a matrix with the percentage of use of the last 12 instants for each VM, choosing the VM with the highest CPU usage correlation in relation to the rest of the VMs.

## 4.3 Default allocation vs. MDG

In order to investigate the effects of applying the improved version of the Default VM Allocation Algorithm implemented by CloudSim, a comparison with the different tested VM selection policies is presented, all of them using the VM allocation policy IQR. The metrics analyzed in the present work are shown in Table 5. The median for all of these metrics are shown in Table 6, which summarizes the complete experimentation. Bold values show the best values in DA vs. MDG comparison.

The main objective of the policy is to improve the BW used and the distance the migrated VMs must travel across the network, reducing the distance between the origin and destination hosts. Observing *RAM in BW* values (Table 6-10) is it clear the objective is achieved, reducing the quantity of MB moved across the

**Table 5.** Metrics for experimentation

Metric	Description
Traffic	Sum of the topology links traffic
Unsatisfied Traffic	Traffic that surpassed the available BW
Migration distances	Average number of jumps that a VM must do to be migrated
Number Migrations	Sum of all the migrations done at each snapshot
Unsatisfied VMs	VMs not providing the required MIPS due host over-saturation
Saturated hosts	Hosts that exceed 70% of CPU but can meet the CPU demand
Unsatisfied hosts	Hosts exceeding 100% of CPU, unsatisfying the CPU demand
% Unsatisfied MIPS	Percentage of MIPS not executed due to host over-saturation
Migrated RAM	Sum of all the sizes of the migrated VMs
RAM in BW	RAM moved across the network
Time Migrating	Sum of the time the migrated VMs spent moving

network by an average of 40%. In a similar way, the number of jumps (Table 6-3) done by migrated VMs is reduced by 25% for WSPSP and an average of 45.24% for the rest of the techniques. Moreover, the fact the VMs are migrated to nearby hosts does not affect the rest of the metrics negatively. On the contrary, except for the average ratio of unsatisfied MIPS (Table 6-4), which shows dispersed values from -7% (WSPSP) to 3.3% (MC), the rest of the metrics are improved. There is a reason for the improvement in migration jumps being higher in the other metrics. WSPSP starts migrating influential VMs, and, hence, VMs with higher CPU demand, which means that not all the VMs fit other hosts, thus limiting the migrations to a few hosts that could be distant. On the contrary, the other techniques migrate all kinds of VM, facilitating nearby hosts for VMs with low CPU demands. The number of *Unsatisfied Hosts* (Table 6-8) is especially reduced, with an average improvement of 12.76% being obtained. Other metrics like *Number of Migrations* (Table 6-5) or *Unsatisfied Traffic* (Table 6-2) obtain little upturns, around 5% on average, but the policy is almost guaranteed not to harm these.

**Table 6.** DA-MDG VM allocation policies comparison

Index Metric	WSPSP		MC		MMT		MU		RS	
	DA	MDG	DA	MDG	DA	MDG	DA	MDG	DA	MDG
1 Traffic	<b>1560913</b>	1561781	1646162	<b>1610919</b>	1637468	<b>1602595</b>	1646674	<b>1616205</b>	1652648	<b>1611330</b>
2 Unsat. Traffic	229150	<b>228607</b>	277088	<b>263239</b>	273139	<b>261452</b>	317057	<b>300819</b>	281150	<b>264989</b>
3 Migration Distances	5.18	<b>4.10</b>	5.17	<b>2.90</b>	5.17	<b>2.76</b>	5.18	<b>2.77</b>	5.18	<b>2.90</b>
4 % Unsat. MIPS	2.41	<b>2.24</b>	<b>2.48</b>	2.57	<b>2.12</b>	2.15	3.29	<b>3.14</b>	<b>2.49</b>	2.51
5 # of Migrations	200	<b>188.5</b>	1373	<b>1300.5</b>	1637.5	<b>1571</b>	4455	<b>4144.5</b>	<b>1407</b>	1409
6 Unsat. VMs	7444.5	<b>7039.5</b>	8281.5	<b>7933</b>	7624.5	<b>7202</b>	15712.5	<b>14022</b>	8525.5	<b>8183.5</b>
7 Overloaded Hosts	3739	<b>3618.5</b>	3979	<b>3882</b>	4001	<b>3872.5</b>	4105.5	<b>3992</b>	3979.5	<b>3819.5</b>
8 Unsat. Hosts	1441	<b>1382</b>	2799	<b>2382.5</b>	2293	<b>1956</b>	3565	<b>3014</b>	2904.5	<b>2478.5</b>
9 Migrated RAM	254040	<b>241346</b>	1679898	<b>1630029</b>	1163570	<b>1129352</b>	5547126	<b>5185030</b>	1763130	<b>1757446</b>
10 RAM in BW	1304229	<b>987856</b>	8735037	<b>4705944</b>	6027358	<b>3210863</b>	28758469	<b>14255744</b>	9133906	<b>5092168</b>
11 Time migrating	4065	<b>3862</b>	26878	<b>26080</b>	18617	<b>18070</b>	88754	<b>82960</b>	28210	<b>28119</b>

#### 4.4 VM selection policies comparison

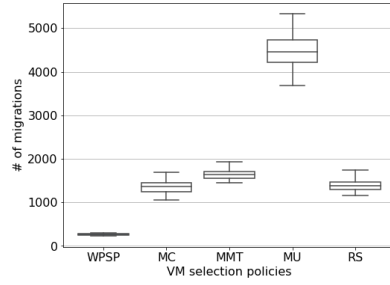
The next experiment was conducted to know the degree of improvement in both WPSP and MDG policies against the VM selection policies and the default VM allocation policy provided by CloudSim. Table 6 shows the obtained results. Underline values reveal which is the best value in the entire comparison for each metric.

A high number of saturated hosts produces a huge number of migrations. Nevertheless, the effects of some of those migrations can be negligible on the *Unsatisfied Hosts* metric. Thus, the correct migratable VM selection is crucial for reducing all these metrics and obtaining better performance. To this end, it is vitally important to identify those VMs which are really responsible for the saturation.

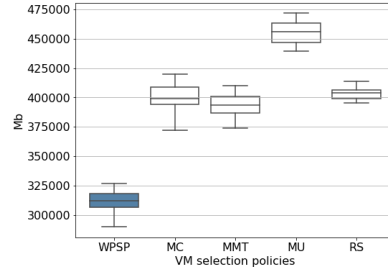
The results for both the *Saturated* and *Unsatisfied Hosts* metrics show great differences with regard to the policies tested in the literature. Observing host saturation, there is an average improvement of 10%, with a 9.56% improvement compared to the second best policy, MMT. Regarding Figure 5, the *Unsatisfied Hosts*, there is a big average improvement of 51% on average, with a 52% improvement over the next technique, RS. Not only is it important to observe how many times the hosts were working over their capacities, but all the MIPS that did not perform during these periods. It can be perceived in Table 6-4, % of *Unsatisfied MIPS*, that WPSP achieves an improvement of 10% for RS and MC, and a big upturn of 32% for MU. These results are even more impressive considering that our proposal achieves them while performing considerably fewer migrations than the other policies, 86.3% reduction in migrations compared with the policy with fewer migrations (MC) (188.5 vs 1300.5).

Observing the *Number of Migrations*, in Figure 3, all other methods show a huge number of VM migrations. Our proposal is able to reduce the overall number of migrations by an average of up to 89%. The lower number of migrations provides greater availability of the communication links and this is obtained without any prejudice on the host loads.

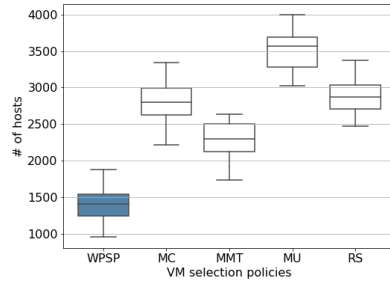
Furthermore, analyzing the interconnection links, from the point of view of the used *Traffic*, Table 6-1, and the *Unsatisfied Traffic*, Figure 4, shows values of up to -5% and -20% respectively. The VMs that were migrated were those that do not interfere in the network links, thus maintaining locally the VMs with inner communications. During the migration process, the VMs must cross the network to reach their host destination. The traffic generated by these VMs could significantly exceed the data interchanged by default. Even if a method (MDG) is implemented to reduce this issue, the *Number of Migrations* is a determining factor. In Figure 6, *RAM in BW*, and Table 6-11, *Time Migrating*, our technique has an average improvement of 89.5% and 86.7% over the other techniques. The results are consistent with the average improvement in the *Number of Migrations* (89%).



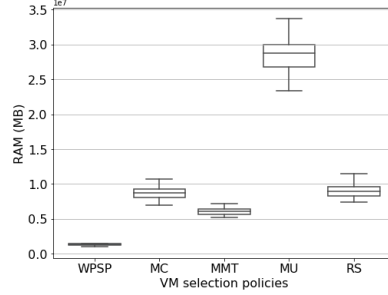
**Fig. 3.** Number of migrations



**Fig. 4.** Unsatisfied Traffic



**Fig. 5.** Over-saturated hosts



**Fig. 6.** RAM across network

## 5 Conclusions

In this paper, the authors defined a VM selection policy that applies the Pearson correlation coefficient with weighting values to evaluate the influence of the VM CPU and Network utilization on the Host. This allows the correct migratable VMs to be determined that are able to reduce the Hosts overload by up to 10% compared with other methods from the literature, with an improvement of 51% in the number of unsatisfied hosts. The use of our proposal also results in a reduction of up to 5% in the bandwidth used and reduced the data traffic by up to 20% .

Additionally, an improvement, MDG, in the default VM allocation policy provided by CloudSim is implemented. This reduces the distance the migrated VMs must travel across the network. The technique is able to reduce the quantity of data moved by the migrated VMs by 89.5%.

Furthermore, the number of migrations was reduced by up to 89%, which provides better resource usage and load balance. The results show the importance of taking network traffic into consideration in the migration decision process.

In the future, the authors are interested in taking into account the way in which migratable VMs can affect the possible assigned Hosts prior to the migration process. Thus, the assignment process would be much more consistent in the future Host behavior, reducing the final number of migrations and the

network utilization. Finally, it could be interesting to discover the limits of WPSP policy in terms of the CPU and network VM stability and ranges.

## References

1. Abdelsamea, A., El-Moursy, A.A., Hemayed, E.E., Eldeeb, H.: Virtual machine consolidation enhancement using hybrid regression algorithms. *Egyptian Informatics Journal* **18**(3), 161–170 (2017)
2. Ahmad, R.W., Gani, A., Hamid, S.H.A., Shiraz, M., Yousafzai, A., Xia, F.: A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *Journal of Network and Computer Applications* **52**, 11 – 25 (2015). <https://doi.org/https://doi.org/10.1016/j.jnca.2015.02.002>, <http://www.sciencedirect.com/science/article/pii/S1084804515000284>
3. Beloglazov, A., Buyya, R.: 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* **24**(13), 1397–1420 (2012)
4. Bonett, D.G., Wright, T.A.: Sample size requirements for estimating pearson, kendall and spearman correlations. *Psychometrika* **65**(1), 23–28 (2000)
5. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A., Buyya, R.: Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience* **41**(1), 23–50 (2011)
6. Choudhary, S., Kothari, A.: Green data center using spearman's ranking algorithm. *International Journal of Computer Science and Information Technologies* **6**(2) (2015)
7. Khoshkholghi, M.A.: Resource usage prediction algorithm using weighted linear regression for virtual machine live migration in cloud data centers (2017)
8. Kumar, P., Kumar, R.: Issues and challenges of load balancing techniques in cloud computing: A survey. *ACM Comput. Surv.* **51**(6), 120:1–120:35 (Feb 2019). <https://doi.org/10.1145/3281010>, <http://doi.acm.org/10.1145/3281010>
9. Moghaddam, S.M., Piraghaj, S.F., O'Sullivan, M., Walker, C., Unsworth, C.: Energy-efficient and sla-aware virtual machine selection algorithm for dynamic resource allocation in cloud data centers. In: 2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC). pp. 103–113. IEEE (2018)
10. Nucci, A., Sridharan, A., Taft, N.: The problem of synthetically generating ip traffic matrices: Initial recommendations. *ACM SIGCOMM Computer Communication Review* **35**(3), 19–32 (2005)
11. Park, K., Pai, V.S.: Comon: a mostly-scalable monitoring system for planetlab. *ACM SIGOPS Operating Systems Review* **40**(1), 65–74 (2006)
12. Saino, L., Cocora, C., Pavlou, G.: A toolchain for simplifying network simulation setup. *SimuTools* **13**, 82–91 (2013)
13. Sun, G., Liao, D., Zhao, D., Xu, Z., Yu, H.: Live migration for multiple correlated virtual machines in cloud-based data centers. *IEEE Transactions on Services Computing* **11**(2), 279–291 (March 2018). <https://doi.org/10.1109/TSC.2015.2477825>
14. Vu, H.T., Hwang, S.: A traffic and power-aware algorithm for virtual machine placement in cloud data center. *International Journal of Grid & Distributed Computing* **7**(1), 350–355 (2014)
15. de Winter, J.C., Gosling, S.D., Potter, J.: Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological methods* **21**(3), 273 (2016)