Cloud Resources Reassignment based on CPU and Bandwidth Correlation

Sergi Vila, Nilson Moraes, Fernando Guirado, Josep L. Lérida, Fernando Cores
and Fábio L. Verdi

Abstract Using virtualization, cloud environments satisfy dynamically the computational resource necessities of the user. The dynamic use of the resources determines the demand of working hosts. Through virtual machine (VM) migrations, datacenters perform load balancing to optimise the resource usage and solve saturation. In this work, a policy is implemented to choose which virtual machines are more suitable to be migrated. The policy evaluates, for each VM, both the CPU load and the Network traffic influence on the assigned host. The corresponding Pearson correlation coefficients are calculated for each one of them, and then, they are evaluated together. The main goal is clearly to 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 72% in the number of migrations, and thus reducing the use of bandwidth (12%), network saturation (32%) and over-saturated hosts (13%).

Acknowledgements This work has been possible thanks to MEIyC-Spain TIN2017-84553-C2-2-R and São Paulo Research Foundation (FAPESP) under the grant #15/19766-9.

1 Introduction

Cloud Computing has become an effective alternative to having 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 resources through a network service and forms a virtual computing resource pool allocated in a classical data center. Thus, it is possible to combine the hosts capacity on an on-demand pay-per-cycle basis. The end users of a cloud computing network usually have no idea where their services are physically located as they are virtualized and allocated to a Virtual Machine (VM) assigned to one of the hosts. The users

---

Sergi Vila, Fernando Guirado, Josep L. Lérida, Fernando Cores
Universitat de Lleida, Spain e-mail: svila@diei.udl.cat, fguirado@diei.udl.cat, jlerida@diei.udl.cat, fcores@diei.udl.cat

Nilson Moraes, Fábio L. Verdi
Department of Computer Science, Federal University of São Carlos, Sorocaba, Brazil e-mail: nilsonmoraes1307@gmail.com, verdi@ufscar.br

trust on the cloud provider in order to maintain the defined Service Level Agreement (SLA).

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

In the literature, there are many methods to avoid overloading hosts in data centers. Some of those methods are based on load balancing strategies that facilitate to distribute the workload equally on available resources [6]. Load balancing also provides scalability and flexibility for growing services. Other proposals apply VM migration in order 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 become critical in order to achieve and maintain the application performance. By this reason, the migration process requires to correctly identify a candidate migratable VM that ensures not only the host load reduction but also to keep contained use of the inter-VMs communication links thus avoiding the SLA violation.

In the present paper, we propose the use of the Pearson correlation factor to correctly identify those VMs that are highly affecting the host overload taking into account both parameters: computational and communication resource usage. The proposed method not only tackles with the current host state, but also evaluate 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. It is also important to avoid unnecessary VM migrations as they introduce much more network traffic, reducing the available bandwidth and compromising the QoS and SLA. 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 hosts usage avoiding the overload and also reducing the global number of VMs migration.

The rest of the work is organised 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 presents 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 [1] present a survey on VM migration and server consolidation. They evaluated multiple migration schemes and they took into account different parameters to compare them. 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 timeline taking snapshots periodically. An important feature that distinguishes our proposal is that is not only based on the current state but also on the host behaviour and tendency.

Raja et al. also assessed the fact that computing and communication resources are both important in the VM migration and host consolidation, but in the presented survey there was not any method based on evaluating the correlation of both parameters and their influence on the host overload and SLA. Our proposal is based on the evaluation of both computing and communication parameters, and thus determining their influence on the VM load and application performance.

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 knowledge fields that demonstrate the correlation importance among multiple parameters in order to take correct decisions. Douglas et al. in [3] and Winter et al. in [10] 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 on 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 [5] was based on the Spearman's Rank Correlation Coefficient to select the optimal VM according to the present workload and datacenter resources availability. Their work was mainly focused on green computing and power preservation techniques applied to Cloud Computing. The obtained results, compared with the VM Random Selection, demonstrated less energy consumption while maintaining the required SLA. Moghaddam et al. in [7] proposed a VM selection algorithm focused on energy reduction and also considering the SLA parameter. The algorithm is based on the Pearson correlation coefficient and it was used to determine both VMs' CPU utilisation and the correlation with their co-hosted VM. Their proposal was evaluated through simulation on the CloudSim environment, and using two different real Cloud data sets by the CoMon project (PlanetLab) and Google. The results showed that the use of correlations improve the VM identification as migratable. The results also demonstrate improvements on energy consumption. Sun et al. in [9] addressed the problem of online migration of multiple correlated VMs among multiple datacenters. This work was focused on the migration performance optimisation. 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 both computing and communication load for each allocated VM. The evaluation is continuous during the host execution time-line, getting snapshots periodically. When an overloaded host has been identified our method determine the VM candidate to be migrated. Applying our proposal, the overall migrations were reduced, thus reducing the network saturation, increasing the host utilisation and without compromising the SLA.

3 Problem Statement

The policy presented in this paper, hereinafter referred to as Pearson Selection Policy (PSP), is based on two main ideas: (1) evaluate the hosts execution on the time-line to determine the resource usage behaviour of each one and detect the overload situations, and (2) evaluate 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 highly related to the hosts overload. Our first goal is to have knowledge of the host load during their execution. This information is obtained getting snapshots of the system periodically. These snapshots contain information about the required resources by VMs and the resources truly 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 resources usage. Each host has allocated multiple VMs and each one 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 overload reduction as the external host communication can be increased due to the VM new allocation. By this, we propose to consider both computation and communication resources usage to identify their influence on the host overload and to determine the relationship between the VMs inside the host.

The idea behind the use of Pearson correlation is to determine the similarities on the CPU and network resources usage between the host and each VM, with the aim to identify 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 less number of migrations. 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 in order to maintain the consistency of the coefficient obtained.

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

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 gets bigger, the other variable tends to get bigger. A negative correlation means that if one variable gets bigger, the other variable tends to get smaller. The weakest linear relationship is indicated by a correlation coefficient equal to 0.

In this paper, *rcpu* is measuring the relationship between the Host CPU usage (x) and a VM's CPU usage (y), in the same way *rnet* is measuring 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 PSP Policy is to identify the candidate VMs to be migrated that eliminate the host saturation with the minimum VMs migrations. In order to meet these objectives, we consider that the VM with higher positive *rcpu* is the best candidate to be migrated with the aim to reduce host saturation. However, in the case that this VM also has a high positive *rnet*, the migration of this VM will produce the increase of data transfers throw the external communication channels, thus fostering the external communication channels saturation and producing a negative impact on the global performance. To prevent this occurring, we should consider the migration of VMs with high positive CPU correlation $rcpu \simeq 1$, but with a weak network correlation *rnet*. While there is the possibility to migrate a group of VMs highly correlated with internal data transfers, this option substantially increases the number of migrations and the total migration cost.

With the aim to consider the VMs consumption of both resources CPU and communications on the migration decision process, we propose a heuristic function computed by Eq. 2, that provides to each VM a value based on the magnitude of both *rcpu* and *rnet* correlation coefficients.

$$hval = \frac{1-(w*rcpu)}{(1+(w*rcpu)-(w*rnet))} \tag{2}$$

The heuristic function allows to module the relationship between both correlation coefficients providing a mechanism to compare the VMs within a Host. Figure **??** shows the contour lines projected by the *hval* function on the plane formed by *rcpu* and *rnet* correlation coefficients. The w variable defines the slope of the *hval* function, and thus is controlling the *hval* values scale. With a value of $w = 1$ the function *hval* tends to 0 irrespective of the correlation values. On the contrary, with $w = 0$ the resulting values tend to 1. The tuning of the w variable can be useful on exceptional cases with the values located in a bunch. However, with $w = 0.5$, we obtain intermediate values that fit our objective.

By way of example, Table 1 shows the corresponding *hval* value for different VMs with different combinations of *rcpu* and *rnet* correlation classes. Figure **??** shows the *hval* value for each VM and their location on the plane. We established the premise that the VMs directly related to the CPU usage are a good candidate to be migrated provided that it is also weakly correlated to the internal network communication. We can observe that this is the case for VM1 obtaining the minimum *hval* value. The VM4 is in the same case than VM1 but with less *rcpu* value resulting in a bigger *hval*. On the other side, the VM0 is equally related to *rcpu* usage than VM1 but due to its strong *rnet* correlation the *hval* value was bigger. These results showed that the *hval* function can be used to prioritize the candidate VMs to be migrated.

Label	*rcpu*	*rnet*	*hval*
VM0	direct (0.9)	direct (0.9)	0.55
VM1	direct (0.9)	weak (0.5)	0.38
VM2	weak (0.5)	direct (0.9)	1.82
VM3	weak (0.5)	weak (0.5)	0.75
VM4	direct (0.8)	weak (0.5)	0.52

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

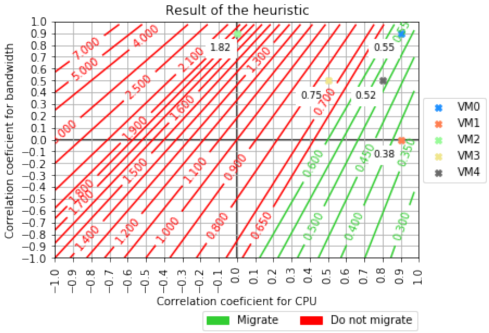

Fig. 1 *hval* function representation and example results

3.1 PSP decision threshold

The proposed *hval* function allows the prioritisation of the VMs to pursue the required objectives, i.e. elimination of Host saturation and reduction of migrations.

Algorithm 1 Pearson Selection Policy - VM selection algorithm

```
Require: $\mathcal{H}$: Overloaded host, $VM_{\mathcal{H}}$: set of VMs $\in \mathcal{H}$
Ensure: $\mathcal{SVM}$: Selected migratable VMs
declare $\mathcal{MVM}$: migratable VMs area, $rcpu$: current cpu correlation values, $rnet$: current network correlation values, $\mathcal{H}val$: heuristic values for each $vm \in \mathcal{MVM}$, $hval\_th$: PSP decision threshold
$\mathcal{MVM} \leftarrow getMigratableVMs(\mathrm{VM}_{\mathcal{H}})$
for each $vm \in \mathcal{MVM}$ calculate
    $rcpu \leftarrow calculate\_cpu\_correlation(\mathcal{H}, vm)$
    $rnet \leftarrow calculate\_net\_correlation(\mathcal{H}, vm)$
    $\mathcal{H}val^{vm} \leftarrow calculate\_hval(rcpu, rnet)$
end for
while $isSaturated(\mathcal{H})$ do
    $vm \leftarrow$ Select vm $\in \mathcal{MVM}$ with $\min(\mathcal{H}val)$ $and$ $\mathcal{H}val \leq hval\_th$
    if $vm$ $is$ $NULL$ then
        $break$
    end if
    $\mathcal{MVM} \leftarrow \mathcal{MVM} - vm$
    $\mathcal{SVM} \leftarrow \mathcal{SVM} \cup vm$
end while
return $\mathcal{SVM}$
```

However, in those cases where the remaining VMs are weakly related to the host CPU *rcpu* but directly related to the network *rnet*, could be interesting to avoid migrations to prevent further negative effects. To tackle with these situations, becomes necessary to define a threshold *hval_th*. The threshold defines the frontier that delimits the migration area where the VMs to be migrated can be selected, see green area in Figure **??**.

With the aim to help to establish a suitable $hval\_th$ value, we conducted an experimentation discussed here. There is a direct relationship between the number of migrations and the quantity of Unsatisfied Traffic. Besides this, the higher is the number of migrations, the lesser is the host saturation. Network saturation and host saturation are indirectly correlated, the improvement of one will produce the worsening of the other, all orchestrated by the number of migrations. Observing Figure 2 we can determine that at 0.6, the number of migrations is stable. When the number of migrations becomes stable, further decisions are equivalent to migrate the enough quantity of VMs to desaturate the host, losing the mean to select only the most representative. In Figure 3, also occurs a similar issue. The unsatisfied traffic becomes stable around 0.55, with a further little constant growing. From this point, any value will get worse network performance. Finally, Figure 4 shows clearly that before 0.4, the number of over-saturated hosts drops drastically. Further improvements after this point are very discrete and the lowest peak is found in 0.6.

Taking these results into consideration, we decided to use 0.6 as the PSP decision threshold ($hval\_th$), because it is representing the critical point where the number of saturated hosts is stabilised, after this value, there is no improvement, even the number of migrations and data saturation are increased.

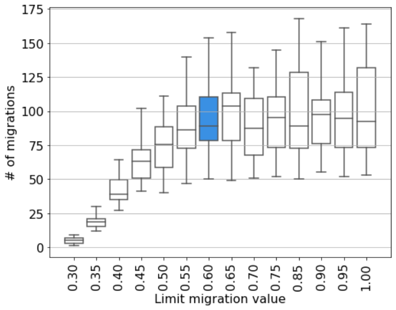

Fig. 2 Number of migrations

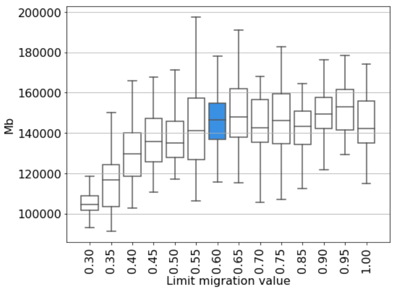

Fig. 3 Unsatisfied Traffic

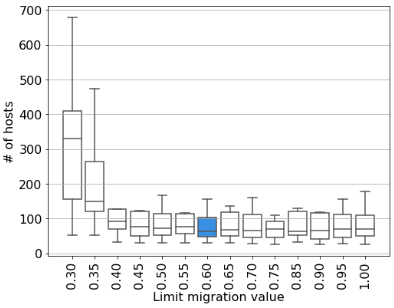

Fig. 4 Over-saturated hosts

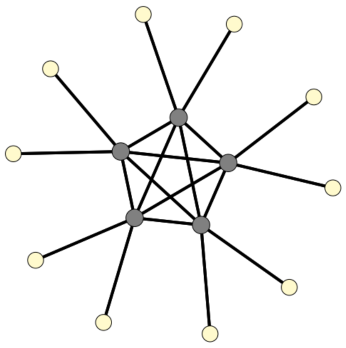

Fig. 5 Topology

3.2 Pearson Selection Policy

The PSP policy proposed in this paper is represented by Algorithm 1. The PSP 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 to migrate. First, there are obtained the VMs that are not being migrated (line 2). Then, the Pearson correlation coefficients are obtained for each VM, in relation to the CPU ($rcpu$) and network ($rnet$) usage on 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. Next, we obtain the heuristic value $hval$ for each VM (line 6). Finally, the VMs with the minimum $hval$ are selected for migration until the saturation problem is solved or no more VMs are eligible ($hval$ values exceeds the PSP decision threshold $hval\_th$). When no VMs meet this criterion, a null value is returned.

4 Experimentation and results

This section describes the configuration of the experimental environment, mainly based on the CloudSim simulator [4] and the obtained results.

4.1 Experimental setup

The CPU load traces used are part of the PlanetLab environment. They are obtained with the CoMon monitoring system [8]. There is a set of traces corresponding to 10 days of execution with around 1000 virtual machines. The first 150 files of the trace 20110303 are the traces used for the experimental study, each containing 288 values corresponding to a day of performing.

The frequency of PlanetLab traces update is 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 machines configuration. Each VM contains one cloudlet acting as endless tasks being the percentage of CPU load defined by the PlanetLab traces. The experimentation uses 150 VMs classified in function of their CPU requirement. Thus, there are four categories; tiny, small, medium and large, for 500, 1000, 2000 and 2500 MIPS respectively.

Type	# CPUs	MIPS	BW (Mbps)	Quantity
Tiny	1	500	100	38
Small	1	1000	100	38
Medium	1	2000	100	37
Large	1	2500	100	37

Table 2 Virtual machine characteristics

Table 3 shows the main characteristics of the hosts used based on those present by default in CloudSim. Thus, there are six small hosts with 1860 MIPS and 4 large hosts with 2660 MIPS, all of them with 2 CPUs.

An interaction is defined as the communication between two VMs throughout the simulation. The network traces, representing the VM interactions, were gener-

Table 3 Host characteristics

Type	Model	# CPUs	MIPS	BW (Mbps)	Quantity
Small	HP ML110 G4	2	1860	1000	6
Large	HP ML110 G5	2	2660	1000	4

ated synthetically[1]. They are characterised to be easily predictable, without abrupt changes on their values. We assume a bandwidth limit usage of 10 Mbps.

We have defined three different interactions, shown in Table 4. The Probability column indicates the occurrence of each class. The Range column defines the use of the communication channel. For each interaction, the type and range are randomly defined according to the values in the table. It was also established that 10% of all the possible VM interactions occur within the same host, while only 0.05% occurs outside the host. To test the VM preservation capabilities of our policy, and after balancing CPU and network resources, we determined that a 10% of internal communications offer enough traffic to maintain VMs in the hosts but with opportunities to leave it depending on the CPU load.

Table 4 Types of interaction

Type	Probability	Range (%)
Low	50%	[0, 20]
Medium	30%	[20, 60]
High	20%	[60, 95]

The interconnection topology is shown in Figure 3.1 where central nodes are switches, being the leaf nodes, hosts.

Table 5 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 will affect the number and typology of the interactions among the VMs.

Table 5 Configuration parameters of CloudSim

Parameter	Value	Parameter	Value
CPU trace	PL 20110303	BW between Host-Switch	1000 Mbps
% Internal interactions	10%	CPU Saturation limit	70%
% External interactions	0.05%	Simulation time	86400 s
# of switches	5	# of experiments per technique	30
# of hosts	10	Host saturation detection technique	CloudSim IQR
# of virtual machines	150	VM allocation technique	CloudSim Default
Window size	6	Underutilised host shutdowns	Disabled
Max. BW of the interactions	10 Mbps		

---

[1] https://git.io/fjC0u

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 being compared. The techniques, presented in [2], used in the comparison are the following:

- Random Search (RS): One migratable VM is chosen randomly.
- Minimum Migration Time (MMT): Chooses the VM that requires less RAM.
- Minimum Utilisation (MU): Chooses the VM which requested fewer MIPS.
- 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 correlation in relation to the rest of the VMs.

4.3 Results

The metrics analysed in the present work are the following:

- Total migrations: Sum of all the migrations done at each snapshot.
- Generated traffic: Sum of the quantity of data that has crossed the links of the topology in each snapshot.
- Unsatisfied traffic: Quantity of Mb that exceeds the capacity of a link.
- Saturated hosts: Hosts that overcome 70% of CPU usage but they still can meet the CPU demand.
- Over-saturated hosts: Hosts that exceed 100% of CPU usage and cannot satisfy the CPU demand.

The median for all of these metrics is shown in Table 6 that summarises the complete experimentation. Additionally, in the last column (labelled as NM) it has added the results when migrations are deactivated in CloudSim. This values allows to have a base reference for the rest of methods.

The *Host Saturation metric* (Table 6-1) measures the number of hosts that triggers the VMs migration process, the *Over-saturated Hosts metric* (Table 6-2) measures the inability of the VMs to perform their calculations on the expected speed ratio. The *Number of migrations metric* (Table 6-3) measures the total number of migrations triggered by all hosts. A high number of saturated hosts produce a huge amount of migrations, nevertheless, the effects of some of those migrations can be negligible on the over-saturated hosts metric. Thus, the correct migratable VMs selection is crucial to reduce all these metrics and obtaining better performance. To this end, it is really important to identify those VMs which are really responsible of the saturation.

The results for both metrics *saturated* and *over-saturated host* do not show great differences with regard the tested literature policies, while our proposal is obtaining up to 8% better performance with respect the second best policy (MMT). These results are even more impressive considering that our proposal achieves these results performing considerably less migrations than the other policies, a 62.7% less migrations with regard the literature policy with less migrations (RS) (85 vs 228). Observing the *Number of Migrations*, in Figure 6, all others methods show a huge

Table 6 Means of the results

Index	Metric	PSP	MMT	RS	MC	MU	NM
1	Saturated hosts (>70%)	**936**	1004	1055	1019.5	1093	956
2	Over-saturated hosts (>100%)	**67.5**	73.5	78.5	75.5	86	359
3	Number of migrations	**85**	347	228	229.5	756.5	0
4	Traffic (Mb)	**608039**	671905	688294	692099	714044	536871
5	Unsatisfied Traffic (Mb)	**140960**	189897	202938	210585	230556	96689

amount of VMs migrations. Our proposal is able to globally reduce the number of migrations up to 72% in average. The less number of migrations provides greater availability of the communication links.

Figure 7 and Figure 8 analyse the *Generated Data Traffic*, and the *Unsatisfied Traffic*, respectively, being up to -12% and -32% lower than the the other methods. The VMs that have been migrated were those that do not interfere in the network links, thus maintaining locally communications.

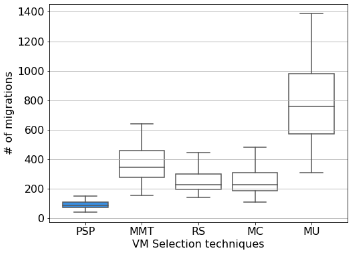

Fig. 6 Number of migrations

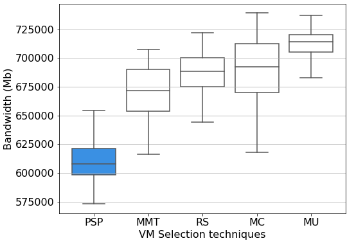

Fig. 7 Traffic

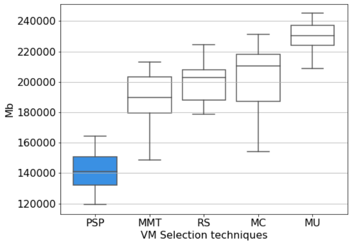

Fig. 8 Unsatisfied traffic

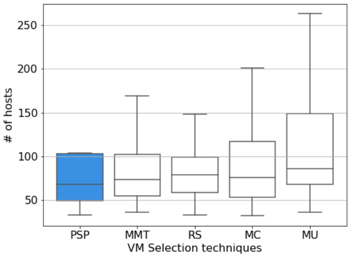

Fig. 9 Over-saturated hosts

Thus, it can be said that our proposal shows a tendency to limit the number of migrations, improving the use of the network since the VMs with greater internal communications are kept on the same host. As a consequence of preventing the dispersion of interconnected VMs, less bandwidth is used and less link saturation happens in central switches.

5 Conclusions

In this paper the authors defined a policy that applies the Pearson correlation coefficient to evaluate the influence of the VMs CPU and Network utilisation over the Host. This allows to determine the correct migratable VMs that is able to reduce the Hosts overload up to 18.7% compared with other methods from the literature. The use of our proposal also results in a reduction of up to 12% of the used bandwidth and reduced up to 32% the data traffic.

Furthermore, the number of migrations was reduced up to 72%, which provides better resources usage and load balance. Results showed the importance of taking in consideration network traffic in the migration decision process.

In the future, the authors are interested on take into account the VMs resources utilisation (CPU and Network) as a weighting factor for the corresponding $rcpu$ and $rnet$. Our hypothesis is that by this we will obtain a much more accurate $hval$ having then better migration decisions.

References

1. 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)
2. 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)
3. Bonett, D.G., Wright, T.A.: Sample size requirements for estimating pearson, kendall and spearman correlations. Psychometrika **65**(1), 23–28 (2000)
4. 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)
5. Choudhary, S., Kothari, A.: Green data center using spearmans ranking algorithm. International Journal of Computer Science and Information Technologies **6**(2), 1672–1676 (2015)
6. 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 (2019)
7. 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)
8. Park, K., Pai, V.S.: Comon: a mostly-scalable monitoring system for planetlab. ACM SIGOPS Operating Systems Review **40**(1), 65–74 (2006)
9. Sun, G., Liao, D., Zhao, D., Xu, Z., Yu, H.: Live migration for multiple correlated virtual machines in cloud-based data centers. IEEE Tr. on Services Computing **11**(2), 279–291 (2018)
10. 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)