# A Two Objective Linear Programming Model for VM Placement in Heterogenous Data Centers

Rym Regaieg(✉), Mohamed Koubàa, Evans Osei-Opoku, and Taoufik Aguili

Laboratoire des Systèmes de Communications,
Département des Technologies de l'Information et de la Communication,
École Nationale d'Ingénieurs de Tunis, Université de Tunis El Manar,
BP 37, Le Belvédère, 1002 Tunis, Tunisia
{rym.regaieg,mohamed.koubaa,evans.oseiopoku,taoufik.aguili}@enit.utm.tn

**Abstract.** Virtual Machine Placement (VMP) is one of the challenging problem arising in cloud computing data centers. VMP is the process of selecting the most suitable Physical Machine (PM) to host the Virtual Machines (VMs). The placement goal can be either maximizing the usage of existing available resources or it can be saving power by being able to shut down some servers (PMs). In this paper, we propose a new Two-Objective Integer Linear Programming (TOILP) model to solve the VMP problem aiming, for the first time as far as we know, at maximizing simultaneously the usage of PM resources while ensuring power efficiency. We also assume heterogeneous configuration for the data center which has been proven, through recent research work and industrial experience, to be more cost-effective for some applications especially those with intensive I/O operations. Two heterogeneous data center configurations are studied in order to ascertain the impact of each configuration on the performance of the proposed model. Simulation results point out the benefits brought by the TOILP model with an average number of used PMs gain of 32.45% and an average total potential cost of resource wastage gain of 60.62%. It was also reported that the cloud provider should not choose the PMs' configuration independently of the offered virtual machines.

**Keywords:** Cloud computing · Heterogeneous data centers
Virtual Machine Placement · Integer Linear Programming
Power consumption · Resource wastage

## 1 Introduction

Cloud Computing is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Infrastructure as a Service (IaaS), Platform as a Service (PaaS)

N. Boudriga et al. (Eds.): UNet 2018, LNCS 11277, pp. 167–178, 2018.
https://doi.org/10.1007/978-3-030-02849-7_15

and Software as a Service (SaaS) are the common categories of cloud computing service models. For IaaS model, cloud provider offers different kinds and amounts of virtualized computing resources (e.g., storage, processing, networks, etc.) gathered into a virtual machine (VM) over the Internet [1]. This provisioned VM allows customers to deploy and run the appropriate application in a personalized and isolated runtime environment.

The decision to place a VM into a particular host is known as the VM placement (VMP) problem [2]. The key challenge here is to maximize the number of cohosted VMs while optimizing a given placement goal. The VMP algorithms can be broadly classified into two categories with respect to their placement goal which fall under one of the following assumptions: maximizing the usage of existing resources or minimizing the power consumption in the data center by shutting down some of the physical machines (servers).

This paper proposes a Two-Objective Integer Linear Programming (TOILP) model that simultaneously optimizes the usage of PMs and power consumption. The TOILP model attempts, given a set of VMs to be set up, to place the VMs in the more suitable server without any VM migration. As the VMP problem has become a particularly challenging task in non homogeneous hardware infrastructures due to the resource variability of PMs, the performances of the proposed TOILP model are evaluated in two different heterogeneous data centers configurations. Two data center configurations are considered in order to study the impact of each PM combination over the different performance metrics (potential cost of resource wastage, number of used PMs, VM rejection ratio). The former configuration has an almost even distribution number of the PMs' configuration whereas the latter is characterised by a different distribution number.

The rest of the paper is organized as follows. Section 2 describes the problem tackled in this paper, Related works is given in Sect. 3. In Sect. 4, we define the notations used to present the proposed model described in Sect. 5. Section 6 shows the experiments evaluating our proposed model and their results. Section 7 concludes the paper.

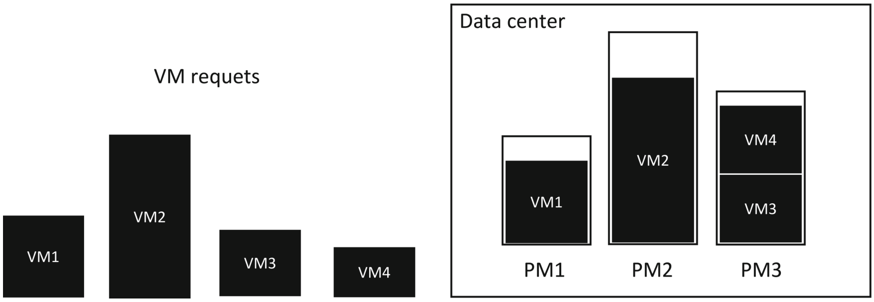


**Fig. 1.** The VM placement problem in a heterogeneous data center

## 2 Description of the Problem

The VMP problem can be stated as follows: for a set of PMs and the resource requirements of VMs, the VMs should be hosted on the PMs with respect to a given placement goal. Figure 1 shows an example of VMP with 4 VMs and 3 PMs in a heterogeneous data center with an end-goal of maximizing the PM usage. As it can be seen, after deploying the VMP process, $VM_1$ is hosted in $PM1$, $VM_2$ is hosted in $PM_2$ due to the insufficient resource capacity in $PM_1$ and $VM_3$ and $VM_4$ are hosted in $PM_3$ as a result of limited resources in both $PM1$ and $PM2$.

Actually, the VMP process usually produces a large amount of wasted resources due to the underutilization of the PMs. As a consequence, an increase in the number of active PMs is noticed leading to a high power consumption in the data center. In this paper, we look for the optimal VM-PM mapping so that the PMs can be used to their maximum efficiency while the energy consumption is minimized by hibernating or shutting down some of the PMs depending on the load conditions.

## 3 Related Work

The VM placement (VMP) problem has been well explored in cloud computing literature and mostly has been considered similar to the vector bin packing problem which is NP-hard [3,4]. The individual PMs can be considered as bins having different dimensions, corresponding to the resource capacities of the PMs. Similarly, the VMs can be considered as objects to be packed into these bins. For each VM, the amount of required resources (dimensional requirements of objects) is specified. The vector bin packing problem aims at allocating a given set of objects of known sizes into a minimum number of needed bins in order not to exceed each bin's capacity. Therefore, the VMP problem is strongly NP-hard.

Many existing algorithms have been proposed to solve the VMP problem. These algorithms include deterministic (eg. integer programming, constrained programming) [5–7], meta-heuristics (eg. randomized greedy, simulated annealing, genetics and evolution) [8–10] and heuristics.

In this paper, we review works which have focused only on the objectives of this paper (maximization of both PM usage and power consumption efficiency) and which have used deterministic algorithms to solve the offline VMP problem. In [11,12], Shi et al., have considered maximizing the cloud provider revenues, under the placement constraints such as full deployment, anti-colocation and security and also resource capacities constraints such as VM requirements and PM capacities. An Integer Linear Programming formulation is proposed to compute the exact solution. The authors have demonstrated that the proposed VMP approach was practical for the offline VM placement in both small and/or medium data centers. Both works [11,12], have evaluated the proposed VMP approach with the VMs of commercial pattern (i.e, predefined VMs resource capacities) in a homogeneous data center. In [4], Ribas et al., have considered minimizing the active physical machines number, under the PMs resource capacity constraints. A Pseudo-Boolean Constraint is proposed to obtain the exact solution. In [13], Sun et al., have considered minimizing power consumption, under the PM resource reservation constraints. A matrix transformation algorithm is proposed to obtain the exact solution. The proposed solution is evaluated with VMs of customized pattern, i.e., the Cloud user defines the VM resource requirements and in a heterogeneous data center. Most of the above-mentioned works use a VM placement approach with a single-objective to achieve resource utilization maximization or power consumption minimization. This paper addresses two challenges. Firstly, it proposes a new two-objective ILP model to address the offline VMP problem that simultaneously maximizes the usage of PM resources and power consumption efficiency. Finally, the solution is performed in two heterogeneous data centers with different configurations to ascertain the impact of each configuration.

## 4 Notations

We use the following notations and typographical conventions:

Index conventions

- i and j as subscript denote a virtual machine request and a physical machine index respectively.

The parameters

- N corresponds to the number of virtual machines arriving at the Data Center to be hosted. The VM request numbered i, denoted $v_i$, $\forall\ 1 \leq i \leq N$, is defined by the tri-tuple $(c_i, r_i, s_i)$ where $c_i$, $r_i$ and $s_i$ are the CPU, memory and storage requirements of VM $v_i$.
- M corresponds to the number of physical machines in the Data Center. The PM numbered j, denoted $P_j$, $\forall\ 1 \leq j \leq M$ is characterized by the tri-tuple $(C_j, R_j, S_j)$ where $C_j$, $R_j$ and $S_j$ are the CPU, memory and storage capacities of PM $P_j$.

The variables

- The binary variable $\lambda_{ij}$. $\lambda_{ij} = 1$ if the VM $v_i$ is hosted by the physical machine $P_j$. $\lambda_{ij} = 0$, otherwise.
- The binary variable $\phi_j$. $\phi_j = 1$, if there is at least one virtual machine hosted by physical machine $P_j$. $\phi_j = 0$, otherwise.

## 5 The Model

The Two-Objective ILP model relies on two separate steps to compute the optimal VM-PM mapping, as shown in Fig. 2. Using the previous notations, Step 1 and Step 2 are given in Table 1.

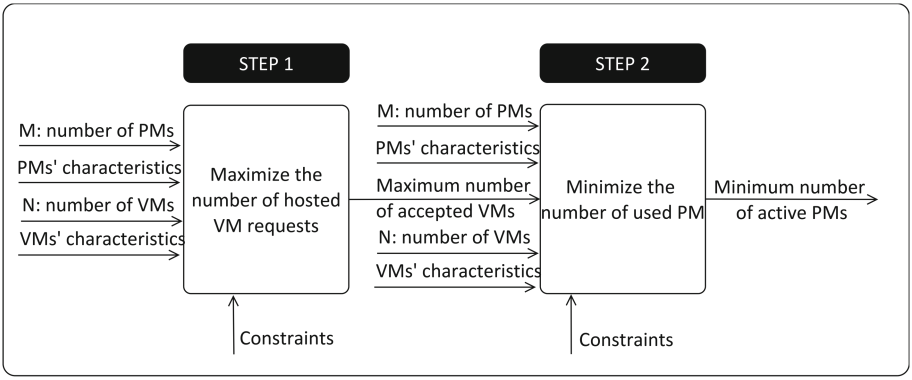


**Fig. 2.** The two-objective ILP model

**Table 1.** The two-objective ILP model

| **Step 1** | **Step 2** |
|---|---|
| **Given** N, M, $C_j$, $R_j$, $S_j$, $c_i$, $r_i$ and $s_i$ | **Given** N, M, $C_j$, $R_j$, $S_j$, $c_i$, $r_i$, $s_i$ and $\psi_{max}$ |
| **Maximize** | **Minimize** |
| $\psi_{max} = \sum_{i=1}^{N}\sum_{j=1}^{M} \lambda_{ij}$ (1) | $\theta = \sum_{j=1}^{M} \phi_j$ (7) |
| $\sum_{j=1}^{M} \lambda_{ij} \leq 1, \quad \forall 1 \leq i \leq N$ (2) | $\psi_{max} \leq \sum_{i=1}^{N}\sum_{j=1}^{M} \lambda_{ij}$ (8) |
| $\sum_{i=1}^{N} c_i\lambda_{ij} \leq C_j, \quad \forall 1 \leq j \leq M$ (3) | $\lambda_{ij} \leq \phi_j, \quad \forall 1 \leq i \leq N, \forall 1 \leq j \leq M$ (9) |
| $\sum_{i=1}^{N} r_i\lambda_{ij} \leq R_j, \quad \forall 1 \leq j \leq M$ (4) | $\phi_j \leq \sum_{i=1}^{N} \lambda_{ij}, \quad \forall 1 \leq j \leq M$ (10) |
| $\sum_{i=1}^{N} s_i\lambda_{ij} \leq S_j, \quad \forall 1 \leq j \leq M$ (5) | 2, 3, 4, 5 and 6 |
| $\lambda_{ij} \in \{0,1\}, \quad \forall 1 \leq i \leq N, \forall 1 \leq j \leq M$ (6) | $\phi_j \in \{0,1\}, \quad \forall 1 \leq j \leq M$ (11) |

Step 1 computes the VM-PM mapping with the objective of maximizing $\psi_{max}$, the number of hosted VM requests. Equations 2 ensures that each VM request $v_i$ is hosted by at most one physical machine $P_j$. Equations 3 ensures that the total amount of CPU consumed by the VMs hosted at a PM $P_j$ is at most equal to the total amount of CPU available at PM $P_j$, $C_j$. Equations 4 and 5 are roughly similar to 3 in that, the CPU resource is replaced by both the memory and storage resources respectively. Equations 6 ensures that $\lambda_{ij}$ variables are binary. It may happen that multiple VM-PM mapping solutions exist for the same number of rejected VM requests. Step 2 selects a solution that additionally minimizes the number of used PMs, $\theta$. Equations 8 ensures that the number of accepted VM requests must be at least $\psi_{max}$, computed by Step 1. Equations 9 and 10 define $\phi_j$ variables. Finally, Eqs. 11 ensures that $\phi_j$ variables are binary.

## 6 Simulation Results

In this section, we experimentally evaluate and compare the performance of the TOILP model in two heterogeneous data center configurations with 20 PMs each. The PMs' characteristics for both configurations, called C1 and C2, are given in Table 2(a) and (b) respectively. One may notice that both configurations have almost the same total amount over the CPU, RAM and disk resources. C2 exceeds C1 with 2.5% and 2% over the total usages of CPU and disk respectively.

**Table 2.** Hetrogeneous data center configurations

(a) First DC Configuration

| C1 | | | | |
|---|---|---|---|---|
| PM | CPU | RAM | DISK | COUNT |
| $PM_1$ | 32 | 64 | 500 | 7 |
| $PM_2$ | 64 | 128 | 1500 | 7 |
| $PM_3$ | 96 | 256 | 3000 | 6 |

(b) Second DC Configuration

| C2 | | | | |
|---|---|---|---|---|
| PM | CPU | RAM | DISK | COUNT |
| $PM_1$ | 32 | 64 | 500 | 5 |
| $PM_2$ | 64 | 128 | 1500 | 10 |
| $PM_3$ | 96 | 256 | 3000 | 5 |

We generated 50 test-scenarios, that is, 50 different VM requests instances each of which consists of N VM requests generated randomly from a predefined set of VM types (Small (S), Medium (M), Large (L) and XLarge (XL) according to the details given in Table 3. Figure 3 gives a detailed information on the average number of generated VMs of type S, M, L and XL for each value of N. The reason why the TOILP model cannot solve the VMP problem with over 290 VM requests is due to the NP-hardness of the problem. We used Optimization Programming language (OPL) [14] with CPLEX 12.6.3 [15], to solve both steps. The CPLEX solver is run on a windows 10 machine with an Intel Core i7, 2.6 GHz processor and 16 GB RAM. In the following, each couple of figure shows the same simulation results obtained by the TOILP model, considering both data center configurations respectively.

Figure 4 plots the average number of hosted VMs on each PM for both DC configurations for N = 130. Each bar in the plot shows the total number of VMs of type S, M, L, and XL hosted by each of the PMs. Subfigures (a) and (b) show the VM placement computed by Step 1 whereas subfigures (c) and (d) show the

**Table 3.** The VM configuration.

| VM | Vcore | Memory (GB) | Disk (GB) |
|---|---|---|---|
| S | 3 | 4 | 50 |
| M | 4 | 8 | 100 |
| L | 5 | 12 | 150 |
| XL | 6 | 24 | 250 |

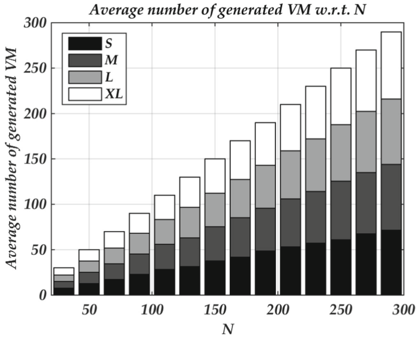


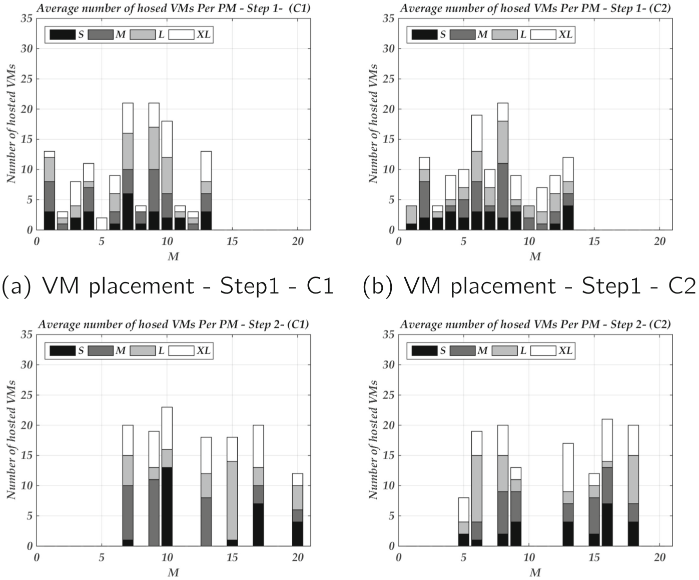
**Fig. 3.** The average number of generated S, M, L and XL w.r.t. N

VM placement computed by Step 2. One may notice that the number of used PMs computed by Step 2 is lower than the one computed by Step 1.

Figure 5 shows the CPU, RAM and disk usages on each PM for both DC configurations. Subfigures (a) and (d) show the CPU usage. Subfigures (b) and (e) plot the RAM usage. Subfigures (c) and (f) show the disk usage. The top-subfigures from (a) to (f) plot the resource usage computed by Step 1. The subfigures below plot the resource usage computed by Step 2. The height of the white bar shows the amount of available resource at the PM when no VM are hosted. The height of the black bar shows the amount of the consumed resource after hosting some VMs. One may observe that the PM resources are efficiently used in C1 compared to C2. The resource usage is almost equal in each of the PM's dimension. We also notice that the number of used PMs computed by C1 is lower than the one computed by C2. This last result will be investigated in the following.

Figure 6 plots the average number of used PMs w.r.t. N, the total number of VMs to be hosted. We report that Step 2 performs efficiently in both DC configurations with average gains (Step 2 achievement against Step 1 achievement) of 32.45% and 27.53% in C1 and C2 respectively. We notice that the average number of used PMs in C1 is lower than the number of used PMs considering configuration C2. Consequently, configuration C1 should hopefully lead to a better power consumption efficiency compared to C2.

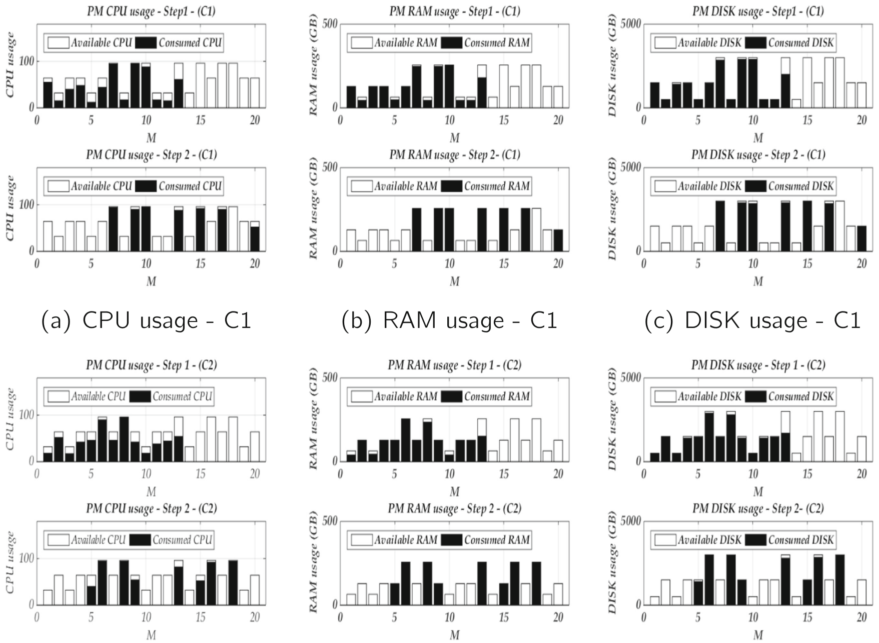
Figure 7 shows the average VM rejection ratio w.r.t. N. The average VM rejection ratio is computed as the ratio of the total number of rejected VMs to the total number of VMs arriving at the DC. We used errorbar plots to show the main VM rejection ratio information for each N. In this plot, the top and bottom bars are the highest and the lowest VM rejection ratio among the fifty generated test-scenarios, while the mid-points denote the average. We notice that in both DC configurations, the number of rejected VMs increases with N as the capacity of available resources per PM is decreasing. We also notice that the average VM rejection ratio computed for configuration C1 is lower than C2, as shown



(a) VM placement - Step1 - C1 (b) VM placement - Step1 - C2

(c) VM placement - Step2 - C1 (d) VM placement - Step2 - C2

**Fig. 4.** The average number of hosted VMs per PM, N = 130



(a) CPU usage - C1 (b) RAM usage - C1 (c) DISK usage - C1

(d) CPU usage - C2 (e) RAM usage - C2 (f) DISK usage - C2

**Fig. 5.** The averages CPU, RAM and DISK usage on each physical machine, N = 130

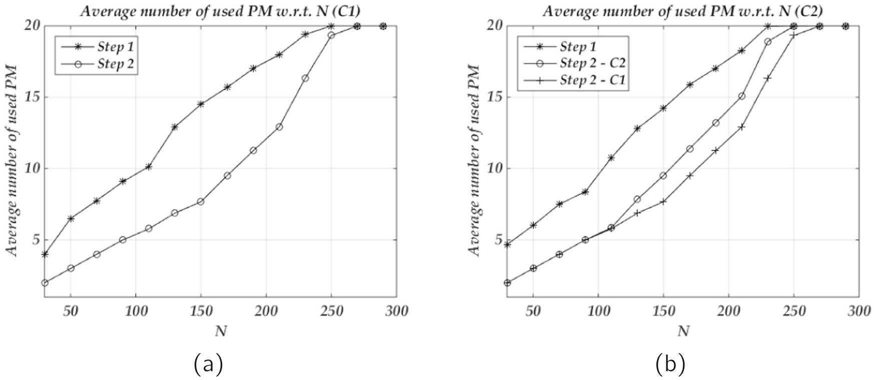


**Fig. 6.** The average number of used PMs w.r.t. N

in Fig. 7(b). The above results will be explained in details in the subsequent section. Table 4 shows the average rejection ratios for VMs of type S, L, M and XL w.r.t. N. The average rejection rate for each VM type $(T, T = S/M/L/XL)$ is computed as the number of VM rejected of type (T) to the total number of VM requests of type (T). We notice that in both configurations, the VM requests of type L and XL are the most rejected ones. This is mainly related to one of the TOILP's objectives which aims at maximizing the number of hosted VMs. Since, VM of type L and XL are more resource consuming.

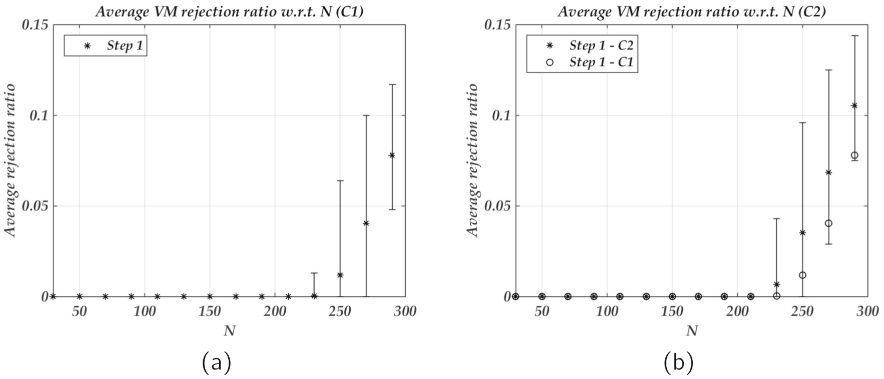


**Fig. 7.** The average VM rejection ratio w.r.t. N

Figure 8 plots the average total potential cost of resource wastage w.r.t. N. The total potential cost of resource wastage is computed as the sum of all the resource wastage amounts of PMs in the data center. The potential cost of

**Table 4.** The average rejection rates of S, M, L and XL VM requests

| N | C1 | | | | | | | | C2 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Step 1 | | | | Step 2 | | | | Step 1 | | | | Step 2 | | | |
| | S | M | L | XL | S | M | L | XL | S | M | L | XL | S | M | L | XL |
| 230 | 0 | 0 | 0.00008 | 0.00042 | 0 | 0 | 0 | 0.00052 | 0 | 0 | 0.00008 | 0.00668 | 0 | 0 | 0 | 0.00678 |
| 250 | 0 | 0.00008 | 0.00008 | 0.0116 | 0 | 0 | 0 | 0.01176 | 0 | 0.00008 | 0.00008 | 0.03528 | 0 | 0 | 0 | 0.03544 |
| 270 | 0 | 0 | 0 | 0.04036 | 0 | 0 | 0 | 0.04036 | 0 | 0.00006 | 0.00014 | 0.06844 | 0 | 0 | 0.00006 | 0.06858 |
| 290 | 0 | 0 | 0.00006 | 0.07792 | 0 | 0 | 0 | 0.078 | 0 | 0.00006 | 0.00012 | 0.10516 | 0 | 0 | 0 | 0.10536 |

resource wastage of the $j^{th}$ PM, $RW_j$ is given by [16]:

$$RW_j = \sum_{l!=k}(R^l - R^k), \qquad \forall 1 \leq j \leq M, \forall 1 \leq l, k \leq 3$$

where, $R^l$ denotes the normalized residual capacity of the $l^{th}$ resource dimension (CPU, RAM, disk), i.e., the ratio of residual resource to total resource. $R^k$ denotes the smallest residual resource rate of all dimensions. We notice that the average total potential cost of resource wastage gain in C1 (60.62%) is higher than in C2 (50.21%) (Step 2 achievement against Step 1 achievement). We also observe that in both DC configurations, the benefit gain of the total potential cost of resource wastage decreases sharply when a set of VMs starts to be rejected ($N \geq 230$). This is due to Step 2's difficulty of balancing the resource usage of each PM as one or more may be exhausted, and others remain unused. This preceding result explains the performance superiority of C1 over C2 in light of the PM resource utilization efficiency, the number of used PMs and the VM rejection ratio.

From Fig. 8(b), we notice that the TOILP model produces a lower average amount of resource wastage in C1 than C2. This can be explained by the fact that the VMs are more compatible with the PM combinations of C1 than C2 based on their resource configurations. This, thereby, explains the above-mentioned fact of the efficiency of PM resource utilization in C1. As a result, more VMs can be hosted using a less number of PMs in C1.

From the obtained simulation results, we point out the following conclusions:

- We report that the TOILP model performs efficiently in both DC configurations, in maximizing the PM usage while minimizing the number of used PMs. In average, the number of used PMs and the total potential cost of resource wastage are reduced by 32.35% and 60.62% respectively.
- The cloud provider should not choose the PM configurations independently of the offered virtual machines. This is due to the degree of compatibility between the VM-PM resource configurations as it has an impact on the amount of resource wastage in the DC.
- We realized that as the amount of the resource wastage is reduced in the DC, the VM rejection is lowered and the number of used PMs is also minimized. This should hopefully lead to lower power consumption in the DC.

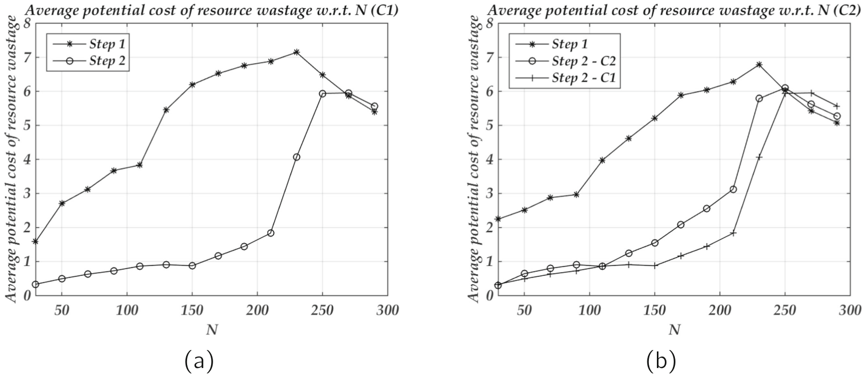


**Fig. 8.** The average potential cost of resource wastage w.r.t. N

## 7 Conclusion and Future Work

In this paper, we proposed a new two objective ILP model to address the VM placement problem in cloud service provider (CSP) data centers with heterogeneous PM configurations. The objectives are to improve the resources usage of physical machines while reducing energy consumption in a data center. We evaluated and compared the performances of the proposed solution in two heterogeneous data centers configurations to ascertain the impact of each configuration. Through extensive simulation scenarios, we reported the benefits brought by the TOILP model in both average gains of total potential cost of resource wastage and number of used PMs. We also reported that the cloud provider should not choose the PM configurations independently of the offered Virtual machines. In the future work, we will compare the performances of the TOILP model in both data center architectures (homogeneous and heterogeneous) in order to assess which of the architectures produces a better trade-off between the resource utilization and power consumption efficiency.

## References

1. The NIST Definition of Cloud Computing, pp. 2–3 (2018). http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
2. Lopez-Pires, F., Baran, B.: Virtual machine placement literature (2018). https://arxiv.org/pdf/1506.01509.pdf
3. Li, Y., Tang, X., Cai, W.: Dynamic bin packing for on-demand cloud resource allocation. IEEE Trans. Parallel Distrib. Syst. **27**(1), 157–170 (2016)
4. Ribas, B.C., Suguimoto, R.M., Montano, R.A.N.R., Silva, F., Castilho, M.: PBFVMC: a new pseudo-Boolean formulation to virtual-machine consolidation. In: Brazilian Conference on Intelligent Systems (BRACIS), pp. 201–206 (2013)
5. Chaisiri, S., Lee, B.S., Niyato, D.: Optimal virtual machine placement across multiple cloud providers. In: IEEE Asia-Pacific Services Computing Conference (APSCC), pp. 103–110 (2009)

6. Shi, L., Butler, B., Botvich, D., Jennings, B.: Provisioning of requests for virtual machine sets with placement constraints in IaaS clouds. In: IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp 499–505 (2013)
7. Van, H.N., Tran, F.D., Menaud, J.M.: Autonomic virtual resource management for service hosting platforms. In: Workshop on in Software Engineering Challenges of Cloud Computing, pp. 1–8 (2009)
8. Geyer-Schulz, A., Ovelgönne, M.: The randomized greedy modularity clustering algorithm and the core groups graph clustering scheme. In: Gaul, W., Geyer-Schulz, A., Baba, Y., Okada, A. (eds.) German-Japanese Interchange of Data Analysis Results. SCDAKO, pp. 17–36. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-01264-3_2
9. Dhingra, A., Paul, S.: Green cloud: heuristic based BFO technique to optimize resource allocation. Indian J. Sci. Technol. **7**(5), 685–691 (2014)
10. Tang, M., Pan, S.: A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data center. Neural Process. Lett. **41**(2), 611–621 (2015)
11. Shi, L., Butler, B., Wang, R., Botvich, D., Jennings, B.: Optimal placement of virtual machines with different placement constraints in IAAS clouds. In: Symposium on ICT and Energy Efficiency and Workshop on Information Theory and Security (CIICT), pp. 202–206 (2012)
12. Shi, L., Butler, B., Botvich, D., Jennings, B.: Provisioning of requests for virtual machine sets with placement constraints in IaaS clouds. In: IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 499–501 (2013)
13. Sun, M., Gu, W., Zhang, X., Shi, H., Zhang, W.: A matrix transformation algorithm for virtual machine placement in cloud. In: IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1778–1783 (2013)
14. Modeling with OPL (2018). http://www-01.ibm.com/software/commerce/optimization/modeling/
15. IBM Cplex Optimizer, January 2018. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
16. Xu, J., Fortes, J.A.B.: Multi-objective virtual machine placement in virtualized data center environments. In: IEEE/ACM Conference on Cyber, Physical and Social Computing (CPSCom) Green Computing and Communications (GreenCom), pp. 179–188 (2010)