Balancing Performances in Online VM Placement

Sonja Filiposka[1,2], Anastas Mishev[1], Carlos Juiz[2]

[1] Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University – Skopje, Republic of Macedonia
{sonja.filiposka,anastas.mishev}@finki.ukim.mk
[2] Architecture and Performance of Computer and Communication Systems Group, University of the Balearic Islands, Palma de Mallorca, Spain
cjuiz@uib.es

Abstract. Optimal usage of data center resources has become one of the most important topics in the cloud computing research community. Increased efficiency alongside decreased power consumption becomes a desired goal. Crucial point in achieving this goal is the process of virtual machine placement. In this paper, we analyze and compare several heuristics aiming to evaluate their capabilities with special attention to balanced resource usage versus total number of used physical machines. The presented results identify the preferred placement heuristic that achieve maximum balancing performances based on the data center characteristics, size of the cloud services and their diversity.

Keywords: Cloud data center, heuristics, performances, VM placement

1 Introduction

Cloud computing becomes today's prevalent computing trend. The centralized resources that reside inside the data centers are flexibly answering to the elastic online demand from cloud users [1]. The key technology that enables cloud computing is virtualization, facilitating the separation of the physical servers from the operating systems and user applications, thus making the usage of computing resources more economically consolidated. While seeking to maximize the utilization of the available hardware resources, datacenters are simultaneously striving for two potentially diverging targets: maximum performance and minimum energy usage.

The cloud users' demand in Infrastructure as a Service (IaaS) environment is represented as a set of tightly coupled virtual machines (VMs) that are governed by the same user. This set of user controlled VMs represents a cloud service [2] that can be comprised of one or multiple VMs with possibly different resource demands (CPU, memory, etc.).

Key component of the cloud datacenter physical machines (PMs) resource manager is the VM placement algorithm that maps the demanded virtual machines resources onto carefully selected target PMs. The mapping abilities of these algorithms are crucial for achieving the best physical resources consolidation and maximizing the profit. Opposed to traditional grid computing and the problem of job scheduling, in the cloud

S. Loshkovska and S. Koceski (eds.), *ICT Innovations 2015*,
Advances in Intelligent Systems and Computing 399,
DOI: 10.1007/978-3-319-25733-4_16

environment the arrival of cloud service demands can not be controlled by the broker. This situation makes the employment of a batch offline method for deciding the best placement of all cloud services at once an unacceptable solution. Thus, the VM placement algorithm in the case of cloud computing VM placement needs to work *online*: dynamically deciding on the placement of the VMs belonging to a given cloud service independently as the services arrive in the requests queue.

The VM placement problem represents an instance of the bin-packing problem, which is known to be strongly NP-hard [3]. In our case the PMs represent the bins that are to be packed with items, i.e. VMs. Finding optimal solutions to this problem has been a big challenge for the research community which is intensified in the recent period by considering the most general case of bin-packing where both bins and items are described as a vector in the$n$-dimensional space, thus allowing the VMs and PMs to be defined with their resources, each dimension representing one type of resource (e.g., CPU, RAM, etc.)[4]. Opposed to the one-sized bins problem, where once a placement is made on a given PM, it becomes not-usable even though there are still available resources on it; the variable bin size across resources [5] enables the representation of non-homogenous data centers that have potentially different resources available after each cloud service placement. The usual methodology for solving such problems is to build a mathematical representation or a metric of the resource utilizations by different VMs and PMs [6]. This metric is typically a function of the normalized resource utilization of individual resource types, sometimes called resource utilization vector. Some approaches use metric that is a weighted sum of the resources [7], while others use a more complex mathematical function of resources [8].

The final goal of all VM placement algorithms is to map the cloud service into a minimum number of bins available, which is commonly implemented as a heuristic approach that aims to minimize or maximize a given objective function based on the metrics used to describe the problem. Thus, the most popular approaches fall into the greedy types of First Fit or Best Fit heuristics, wherein the ordering is defined using a size function which returns a scalar for each bin and item. Note that all more complex approaches using multi-objective functions are still based on the combination of the heuristic approaches that are examined in this paper. While striving for most efficient packing, the difference in the implementations can also be in whether they take into account balanced resource utilization [9]. Although, load balancing seems indifferent on the small scale of one cloud service placement, it has major repercussions on the overall resource utilization and performances of the entire datacenter. The main objective in this case is to minimize the number of used PMs but in such a way that the used resources are optimally utilized, i.e. the PMs have a small amount of wasted unutilized resources along any dimension of the resource vector.

Thus, in this paper we aim to analyze the performances of the most popular online VM placement heuristic algorithms from the balancing efficiency point of view and how it is influenced by the different characteristics of the datacenter PMs and the demanded cloud services from the user side. The rest of the paper is organized as follows: In the next section we describe the variable size n-dimensional implementation of four different heuristics. In section 3 the results from the extensive performance analysis are presented. The final section concludes the paper.

2 VM Placement Heuristics

The most commonly implemented VM placement algorithms are based on the following heuristics: BinCentric(BC), DotProduct (DP), Vector Based Load Balancing (VBLB) and Vector Based Consolidation (VBC). Since we are mostly interested in the balancing performance of these heuristics, in the first part of this section we give just a brief overview of their packing strategy. For more information please refer to the corresponding references.

The Bin centric [10] heuristic belongs to the Best Fit Decreasing group. The packing starts from the smallest bin and iterates while the list of bins is not empty. It continuously places the biggest item that can fit into the selected bin until such items no longer exist, after which the selected bin is removed from the list. The scalar sizes of the items and bins used for ordering inside this heuristic are calculated as weighted sums of the respective vector components (requirements for items, and remaining capacities for bins). Among the different proposed scaling coefficients used for the weighted sums, we used the best performing BC with capacity normalized fitting implementation where the normalization is based on bins capacities.

The DotProduct [11] approach is an example of First Fit Decreasing heuristic. Its goal is to maximize the weighted similarity of the bin and the item, i.e. the scalar projection of the item requirements onto the bin remaining capacities. In our DP implementation, we normalize both requirements and capacities, thus minimizing the angle between the bin and item vectors. Note that, in order to determine the maximum similarity, dot products for all pairs of bins and items must be computed, which lowers the performance of this heuristic in terms of computational time.

Opposed to the previous approaches, the Vector Based Load Balancing [6] heuristic aims at balancing the load of the bins. Inside this heuristic, the current load of the bins falls into one of three categories: low, medium and high, with respect to the normalized resource utilization vector. When placing a new item, the heuristic tries to discover the least loaded bin that has complementary resource usage with respect to that item.

In the cases when the main goal is to minimize the number of used bins, instead of load balancing, Vector Based server Consolidation [6] heuristic can be used. In this situation, when placing a new item, the heuristic tries to find the bin with the highest load that has complementary resource usage with respect to the item.

2.1 Dynamic Online VM Placement Illustration

As a first step towards discovering the underlying VM placement mechanisms of the discussed heuristics we present an overview of the online placement efficiency in terms of balanced physical machines for 3 sample cloud services (see Fig.1).

Each quadrant represents a single PM described with two normalized physical resources (CPU – x axes, and RAM memory – y axes). Each cloud service is comprised of different VMs and their placement across the PMs is color coded (yellow, green, blue). The light blue rectangles represent the free capacity still available for further use on the PM.

If during VM placement, one of the PM's resources becomes depleted (the VM rectangle reaches the borders of the PM's quadrant, as marked on Fig.1-a), the rest of the PM's resources are being wasted. For achieving efficient use of the data center resources this type of placement is not desirable and eventually leads to using a larger number of PMs that increases the costs and power usage of the datacenter. Thus, one of the important characteristics of the chosen VM placement heuristic must be uniform, i.e. balanced, usage of the PMs that should (in ideal cases) follow the main resource utilization vector represented as the main diagonal of the PM quadrant.

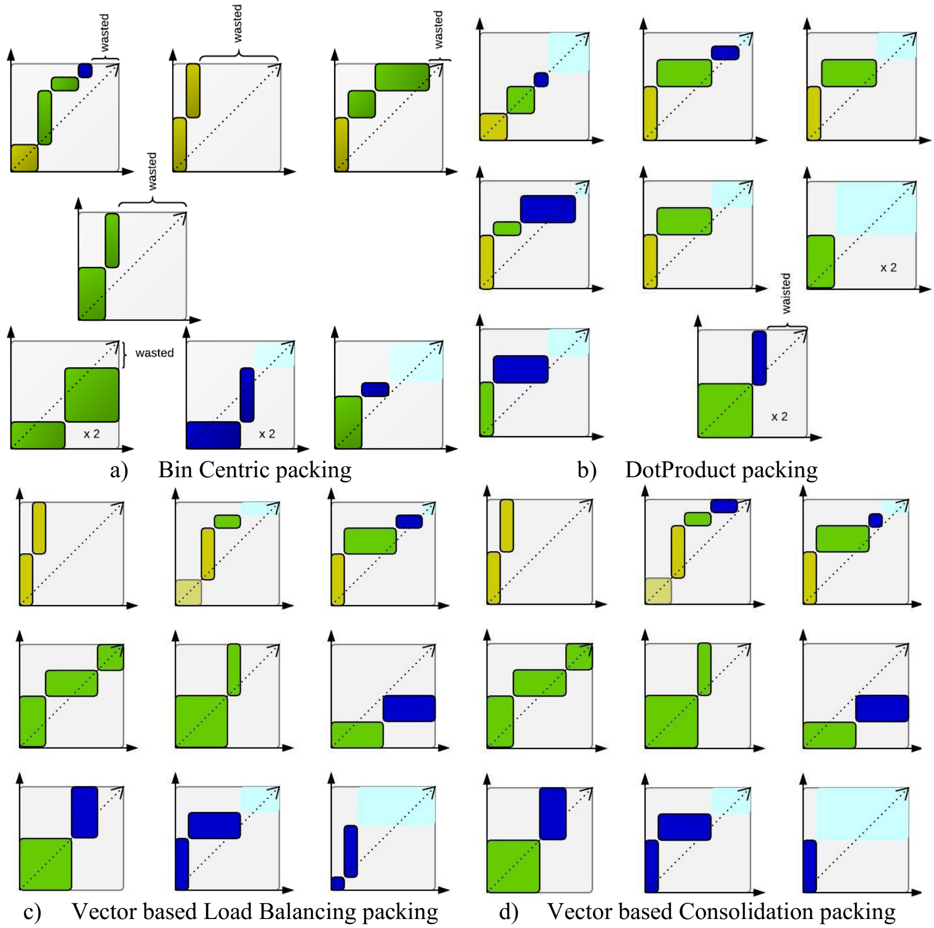


Fig. 1. Placement decision for three sample cloud services using different heuristics

When comparing the different heuristics, we can conclude that the Bin Centric (BC) packing heuristics exhibits the worst performances on balanced packing of the presented sets of VMs, while on the other hand the DotProduct (DP) heuristic achieves the maximum possible balance. However, in order to achieve the maximum balancing DotProduct uses 10 PMs compared to the other 3 heuristics that need only 9 PMs to accommodate the same VMs. The two variations of the Vector based packing differ in the placement of two very small VMs due to the consolidation effort of the second heuristic, which results with slightly better packing. Another remark that should be noted for the presented placement is the small number of variations in the placement decisions across all heuristics, which leads to the conclusion that, when compared on a larger scale, the heuristics should have similar performances, with DotProduct using a slightly larger number of PMs in order to achieve better balancing. However, as presented below, this is not the case.

3 Performance Analysis

In order to analyze the performances of the four heuristics in the case of online VM placement of a large number of cloud services thus recreating a typical cloud datacenter setting, we defined a number of different simulation scenarios by varying the main cloud service description parameters, as well as, the datacenter PMs resources. The results presented in the rest of the paper are obtained different cases of online placement of 1000 to 8000 cloud services, each defined with minimum 5 and maximum 20 VMs. Each VM is randomly generated with 1, 2 or 4 cores and 2, 4, or 8 GB RAM. The VMs are to be placed inside a 5000 or 10000 PMs homogenous cloud datacenter wherein each PM has 8 cores and 16 GB RAM, or 16 cores and 32 GB RAM. Note that the heuristics are deciding on the placement on each cloud service separately, one by one, i.e. online VM cloud service placement, as opposed to the batch mode where all cloud services are placed at once as a complete set of VMs.

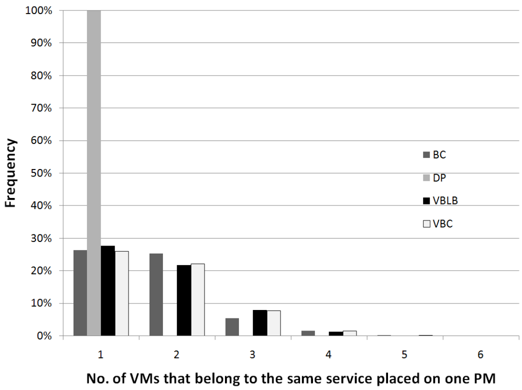
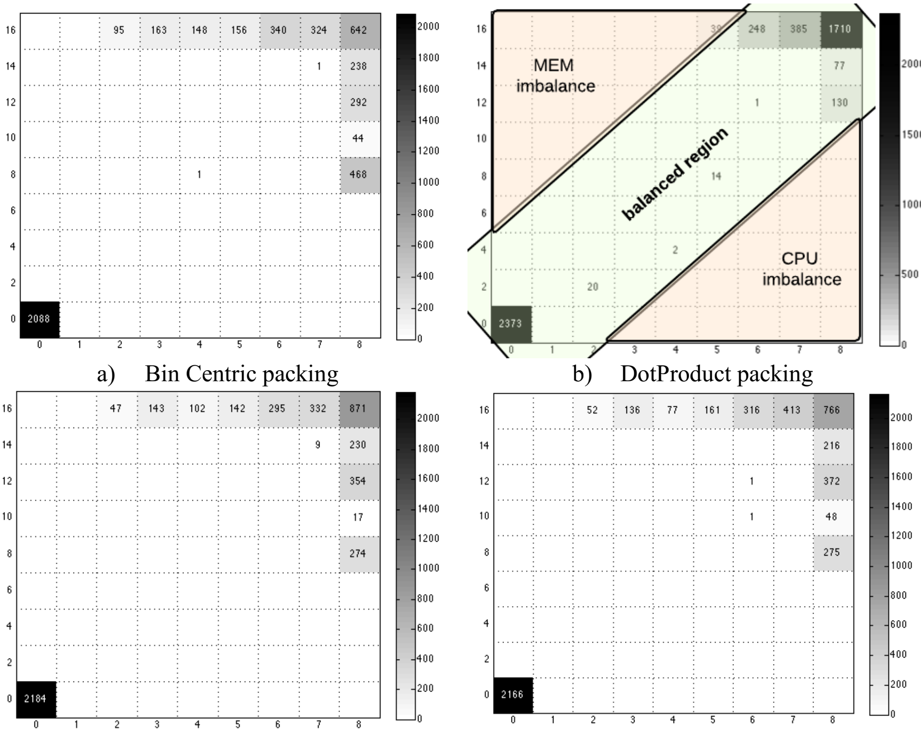


Fig. 2. VM placement diversity across PMs

We first tested the persistence of the DP's typical behavior that was already noticed in Fig. 1. Namely, while DP places one set of VMs that belong to the same cloud service, it aims at placing each VM on a different PM. This however is not a regular case for the other heuristics, where there are also large number of cases when 2, 3 or more VMs that belong to the same cloud service are placed on the same PM as it is represented in Fig. 2. This behavior exhibited by DP is one of the main reasons for achieving the best balancing compared to the other heuristics and is due to the DP's aim towards a global minimum when observing the total placement of all cloud services. However, this strategy's pitfall can be manifested in the case of having a fraction of cloud services consisting of an extremely high number of VMs. In this case DP, following the motto of 1 VM on 1 PM per cloud service, will have to allocate new PMs, while the other heuristics will consolidate the placement better and yield to better resource usage.



a) Bin Centric packing

b) DotProduct packing

c) Vector based Load Balancing packing

d) Vector based Consolidation packing

Fig. 3. Balancing VM placement "heat" maps

The resource usage of a cloud datacenter with 5000 PMs is given in the corresponding heat maps in Fig.3after the online placement of 1500 cloud services, each with max 10VMs. The cell annotation represents the number of PMs that have the corresponding used resources (CPU x-axis, RAM y-axis).Note that lowest leftmost cell (0,0) represents the empty, not-used PMs, and the top right cell (8,16) contains the number of fully occupied PMs that have no wasted resources.

All four heat maps depict the dense packing ability of the chosen heuristics, where there is a very small number of PMs that are not close to fully packed according to at least one resource dimension. When considering the performances of the different heuristics via the number of used PMs only, the absolute winner is DP, followed by VBLB and VBC that show slight differences, and BC as the worst performer. We use the represented heat maps in order to gain a deeper insight on the way these performances are achieved, especially from the point of view of balanced or wasted resources, and future usage of the not fully used PMs.

Following the examples from Fig. 1, we define the usage of the resources to be *balanced* if the majority of the PMs are within the region around the main diagonal (consider the annotation on Fig. 3-b). Outside this region the PMs can be unbalanced due to the higher number of cores used while a larger portion of the memory remains unused, the so-called *CPU imbalance* region, or due to the higher quantity of memory used while there is a large number of cores still available, the *MEM imbalance* region.

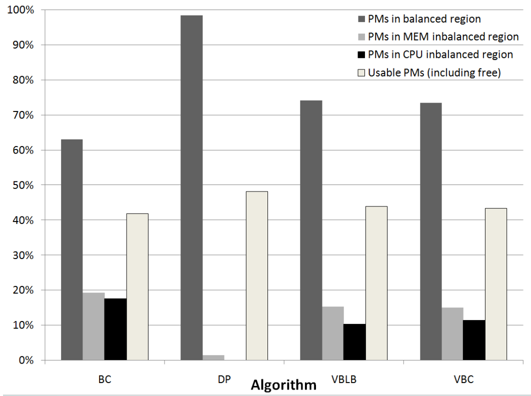


Fig. 4. Comparison of the balanced PM resource usage

The performances of the different heuristics in terms of number of PMs in the balanced versus imbalanced region, are presented in Fig. 4. As it is expected, DP has almost all of the PMs (98%) placed in the balanced region, with only a few in the memory unbalanced region. Also, DP has the highest number of still usable PMs (used CPU<8, and used Memory<16) upon the placement of the full set of cloud services. Next in performances are VBLB and VBC, while BC is last, having lowest number for both balanced and usable PMs.

In order to further understand the influence of the PM resource capacity on the heuristics behavior, we compared the balancing performances of the placement decisions for the cases when the cloud data center is built using PMs with 8 cores and 16 GB RAM vs. 16 cores and 32 GB RAM. As shown in Fig. 5, the PM capacity does not strongly affect DP's balancing performances, although it performs slightly better when the PMs have smaller capacity. Aside from DP, when increasing the PM resource capacity, the heuristic performances are decreasing because of the accentuated non-balanced packing when compared to DP.

An interesting observation is that VBC down-performs relative to the rest of the heuristics, with its performances falling in the case when larger PMs are used. Thus, when working with PMs with higher resource capacity, the difference between the two vector-based approaches is more pronounced.

The differences in performances per heuristic that can be observed in Fig. 5 are due to the different nature of the cloud services that are to be placed, or more precisely, the size in terms of maximum VMs per cloud service. While this parameter has no influence on DP's performances, the changes in balancing performances for the other three heuristics are especially pronounced for smaller number of total VMs. These variations are largest for the BC's performances, and this is the reason why this heuristic has been chosen as a representative for the results presented in Fig. 6. VBLB and VBC exhibit high to moderate performances with similar, less obvious, behavior.

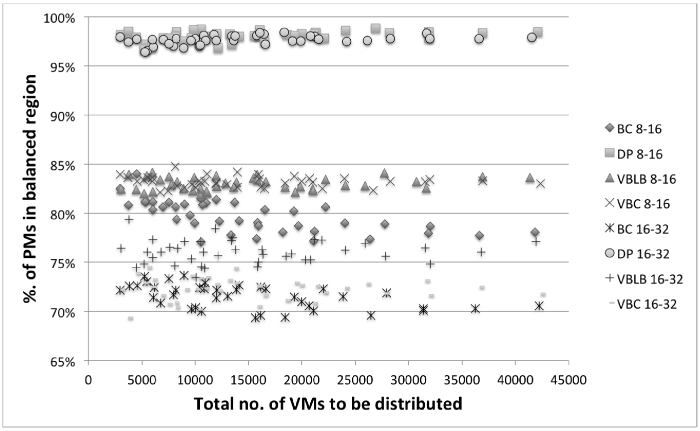


Fig. 5. Balanced PMs distribution

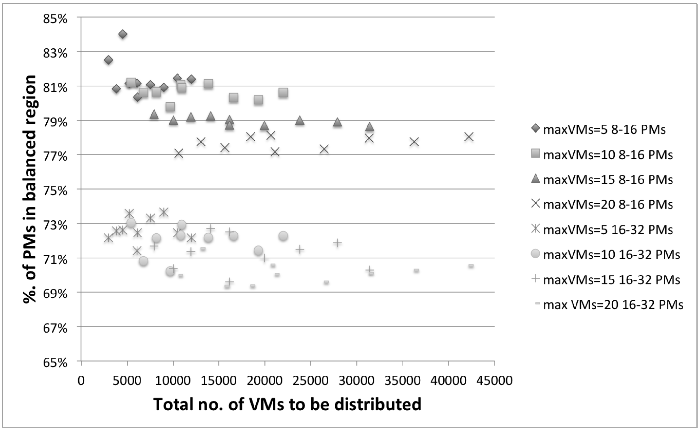


Fig. 6. Cloud service size influence on BC's balancing performances

Fig. 6 clearly demonstrates that in both cases for different initial resource capacities of the underlying PMs, BC's balancing performances are best in the cases when the online cloud service placement is done for small cloud service sizes (max 5 VMs). As the cloud service size grows the heuristic balancing performances are dropping due to the fact that in each service placement the heuristic tends to fill the PMs up their maximum capacity thus making more imbalanced packing decisions.

However, one must bear in mind that the total number of PMs used to accommodate the same set of cloud services changes drastically with the change of initial PM resource capacity. Fig.7 represents the overall packing performances of the four heuristics, when larger and smaller PMs are used. It is expected that when using larger PMs the number of used PMs is lower. Yet, the figure shows that the overall performances of the DP heuristic are deeply influenced by the PM's initial resource capacity. When using smaller PMs (8 cores and 16 GB RAM), the DP performances are the highest, using the fewest PMs of all compared heuristics. But, in the case of larger PMs, DP underperforms even the so far worst heuristic BC, using the largest number of PMs compared to the other heuristics. This is due to the same fact that makes DP the best balancer: choosing to place all VMs from one cloud service on different PMs makes DP use more PMs per cloud service. In the case of high capacity PMs, reaching the full used capacity requires a huge number of cloud services to be distributed.

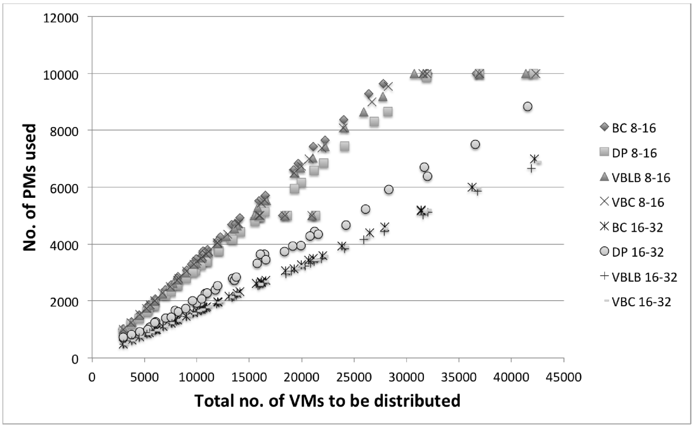


Fig. 7. Initial resource capacity influence on number of used PMs

For the rest of the heuristics, their performances are tending to equalize for higher capacity PMs, showing slight variations in the number of large capacity PMs used. All of the observed differences in performances are due to the fact of tasking the heuristics to pack proportionally much smaller items into large bins that becomes an exceptionally difficult problem for DP, while BC is most resilient to changes compared to the other three, although still exhibiting worst performances.

4 Conclusion

In this paper we analyzed the balancing performances of the most popular online VM placement heuristics used today by cross comparing them to the overall global performances. Our goal was to determine the influence different factors like: characteristics of the cloud services that are to be placed, and features of the physical resources capacity, have on the overall and balancing performances.

Balancing performances of online VM placement heuristics in cloud data centers is crucial for determining the long-term behavior and efficiency of the data center as a whole. While using heuristics that provide best balancing (DP) ensures the best possible usage of the PMs resources, there are cases when due to dimensioning mismatches the price that will be paid for an efficient resource usage is the engagement of a larger number of PMs leading to higher power consumption inside the datacenter.

Thus, a careful highly tailored choosing of the VM placement heuristic that is going to be employed needs to be made in order to align the datacenter physical characteristics with the users demand in the form of cloud services. The overall results show that the BC heuristics is the worst choice for all analyzed cases no matter the type of cloud services or underlying resource capacities. On the other side, DP holds best performances for well-matched cloud service – physical capacities. Hence, if there is no prior knowledge on the compatibility of cloud service demands with the available physical resources, the conservative approach would be to use vector load balancing, while vector consolidation exhibits slightly lower performances.

References

1. Mell, P., Grance, T.: The NIST definition of cloud computing. National Institute of Standards and Technology, 53(6), 50 (2009)
2. Tordsson, J., Montero, R. S., Moreno-Vozmediano, R., Llorente, I. M.: Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. Future Generation Computer Systems 28:2, 358-367 (2012)
3. Magazine, M. J., Chern, M.-S.: A note on approximation schemes for multidimensional knapsack problems. Mathematics of Operations Research, 9:2, 244-247 (1984)
4. Xi Li; Ventresque, A.; Murphy, J.; Thorburn, J.: A Fair Comparison of VM Placement Heuristics and a More Effective Solution.2014 IEEE 13th International Symposium on Parallel and Distributed Computing (ISPDC), 35:42, 24-27 (2014)
5. Gabay, M., Zaourar, S.: Variable size vector bin packing heuristics - Application to the machine reassignment problem. OSP (2013)
6. Mishra, M., Sahoo, A.: On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach. IEEE 4th International Conference on Cloud Computing, 275-282 (2011)
7. Arzuaga E. et al: Quantifying load imbalance on virtualized enterprise servers. In Proc. of the 1st joint WOSP/SIPEW int. conf. on Performance engineering, 235–242. ACM(2010)
8. Wood,T. et al: Black-box and gray-box strategies for virtual machine migration. In Proc. Networked Systems Design and Implementation, 17-17 (2007)
9. Chen, L., Zhang, J., Cai, L., Li, R., He, T., Meng, T.: MTAD: A Multitarget Heuristic Algorithm for Virtual Machine Placement. International Journal of Distributed Sensor Networks, Article ID 679170, (2015)
10. Panigrahy, R., Talwar, K., Uyeda, L., Wieder, U.: Heuristics for vector bin packing. research.microsoft.com (2011)
11. Singh, A., Korupolu, M., Mohapatra, D.: Server-storage virtualization: integration and load balancing in data centers. In Proc. of the ACM/IEEE conf. on Supercomputing (2008)