Parameter Tuning for Scalable Multi-Resource Server Consolidation in Cloud Systems

Claudia Canali, Riccardo Lancellotti
Department of Engineering "Enzo Ferrari",
University of Modena and Reggio Emilia,
Via Vivarelli, 10, Modena, Italy
Email: {claudia.canali, riccardo.lancellotti}@unimore.it

***Abstract*—Infrastructure as a Service cloud providers are increasingly relying on scalable and efficient Virtual Machines (VMs) placement as the main solution for reducing unnecessary costs and wastes of physical resources. However, the continuous growth of the size of cloud data centers poses scalability challenges to find optimal placement solutions. The use of heuristics and simplified server consolidation models that partially discard information about the VMs behavior represents the typical approach to guarantee scalability, but at the expense of suboptimal placement solutions. A recently proposed alternative approach, namely Class-Based Placement (CBP), divides VMs in classes with similar behavior in terms of resource usage, and addresses scalability by considering a small-scale server consolidation problem that is replicated as a building block for the whole data center. However, the server consolidation model exploited by the CBP technique suffers from two main limitations. First, it considers only one VM resource (CPU) for the consolidation problem. Second, it does not analyze the impact of the number (and size) of building blocks to consider. Many small building blocks may reduce the overall VMs placement solution quality due to fragmentation of the physical server resources over blocks. On the other hand, few large building blocks may become computationally expensive to handle and may be unsolvable due to the problem complexity. This paper extends the CBP server consolidation model to take into account multiple resources. Furthermore, we analyze the impact of block size on the performance of the proposed consolidation model, and we present and compare multiple strategies to estimate the best number of blocks. Our proposal is validated through experimental results based on a real cloud computing data center.**

I. Introduction

Cloud computing is emerging as a successful paradigm for the provision of ICT services. The on-demand, pay-as-you-go philosophy is clearly suited to meet the demands of highly variable workloads that characterize modern services. The success of cloud computing is testified by the projected increase of two orders of magnitude in fifteen years for storage and processing power of the cloud Infrastructure as a Service (IaaS) data centers [1]. However, this growth of the cloud data centers determines new challenges at the level of infrastructure monitoring and management. The number of Virtual Machines (VMs) with variable demands in term of system resources is a challenge when we want to collect and analyze data to understand the dynamics of resource demands. The placement of these VMs over the physical servers is an even more critical issue because it involves the solution of a bin-packing problem encompassing the whole data center. Ensuring a scalable and effective solution for the VMs placement problem is currently a major challenge for the cloud computing industry.

The current solution to cope with such complexity is to simplify the VMs placement problem to reduce its complexity to a more manageable level. For example, we can discard the actual behavior of VMs in terms of resource demands (i.e., they consider all VMs of the same *nominal size* equal to each other [2], [3]). When the behavior of each VM is taken into account, the behavior model can be simplified, considering only few demand levels (e.g., day vs. night) or considering just a few resources (e.g., only CPU). Even the problem solution may be simplified exploiting highly simplified heuristics, such as the First Fit Decreasing (FFD) algorithm [4]. In every case, the result is a low quality solution for the VMs placement problem that leads to a waste of cloud data center resources.

In recent times, the authors proposed a novel approach, namely *Class-Based Placement* (CBP), that leverages similar behavior of classes of VMs (i.e., VMs hosting the same software component of the same application) to increase the scalability of the VMs placement [5]. Instead of considering a single bin-packing problem for VMs placement, the CBP approach splits the problem into small building blocks that are easy to solve and can be composed to reach a global solution. However, the initial proposal of CBP has two main limitations. First, it still considers only CPU as the main metric for the underlying VMs placement problem. While in most applications CPU is the main bottleneck resource [6], not considering other critical resources such as memory, network, and I/O, may limit the application scenarios for the technique. Second, the study in [5] does not provide a complete analysis of the impact of the building block size on the quality of the final VMs placement. The trade-off should be clear: on one hand, a large number of small building blocks can obtain a benefit in terms of scalability at the expenses of a less efficient placement due to unused spare capacity within each block (capacity fragmentation); on the other hand, larger blocks tend to avoid fragmentation effects, but the underlying placement problem is much more demanding from a computational point of view.

This paper contains three main contributions. First, we extend the existing server consolidation model considering only CPU utilization to take into account multiple resources for each VM. Second, we analyze the impact of the number and size of building blocks on the quality of the VMs placement solutions. Third, we propose and compare multiple strategies to automatically determine the best number of blocks taking into account the quality of the VMs placement. A preliminary version of the latter two contributions appeared in [7]. However, in this paper we provide a deeper evaluation of the impact of the number ad size of the building blocks that takes into account more complex scenarios thanks to the extended multi-resource model. Our results demonstrate that the new proposal for the block number estimation improves the previous approach, achieving a solution quality for the VMs placement that significantly outperforms the state of the art solution.

The remainder of this paper is organized as follows. Section II describes the Class-Based Placement and provides a model to determine an appropriate number of blocks. Section III describes the results of the technique evaluation. Finally, Section IV discusses the related work and Section V concludes the paper with some final remarks.

II. Virtual machines placement

In this section we present the proposed multi-resource consolidation model. We first outline the VMs placement problem, next we describe the Class-Based Placement (CBP) technique proposed in [5], that is the reference scenario for our proposal. Next, we provide a formal model for the multi-resource server consolidation, and we discuss the main parameter affecting the performance of the consolidation model, that is the number of the small-scale problems that are the building blocks for the global solution. In particular, we outline the pros and cons of having few large problems *vs.* having many small problems. Finally, we propose multiple strategies to determine the best number of building blocks for the consolidation model.

A. Problem overview

The generic VMs placement problem can be outlined as in Figure 1. Each VM has requirements in terms of resources that are necessary for the VM to run: such resources include CPU cycles, memory, network bandwidth, and I/O operations. We also have a physical infrastructure, composed of Physical Servers (PS), where each server can provide a given amount of resources. The VMs resource requirements and the physical server resource capacity represent the input of the VMs placement problem, whose key element is the server consolidation task: the final goal is to map VMs over the physical servers of the infrastructure, minimizing the number of used servers while satisfying the requirements of each VM in terms of resources usage.

The application of the class-based placement to a IaaS cloud data center is based on the following two assumptions. First, we consider that the VMs placement is a periodic task, based on the expected resource requirements for the next period. Second, we assume to be able to group VMs into classes with similar behavior, where VMs belonging to the same class exhibit similar resource requirements. The presence of classes of VMs with similar behavior represents a common condition that occurs every time an application is replicated over a distributed architecture for scalability and availability [8]. Even if the knowledge of replicated application deployment is not directly available to IaaS cloud providers, we can exploit proposals in literature that enable the clustering of VMs with similar behavior [9], [10].

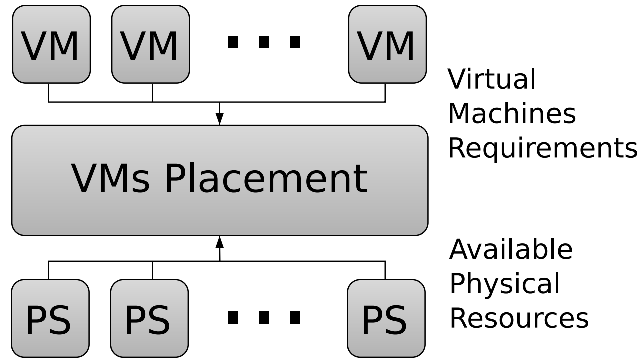

Fig. 1: VMs placement problem

B. Class-based placement

Class-based placement, introduced in [5], aims to improve the scalability of the VMs placement problem solution. The basic idea is to reduce the global server consolidation problem for VMs placement, that operates on the whole data center, to a smaller problem involving only few VMs for each class. The server consolidation process is usually modeled as an integer bin packing problem: the reduced size of the problem allows us to solve to optimality the bin packing considering a multi dimensional formulation with a number of time intervals that would not be possible to consider for the global problem; then, the obtained solution can be replicated as a building block to determine the global solution for the placement of the VMs in the cloud data center.

Figure 2 depicts the periodic VMs placement in a IaaS cloud data center that adopts the CBP approach. We consider as the input of the consolidation model the prediction of the future resource demands for the next planning period. Resource demands are expressed for each class of VMs (we present them as "F_1", ..., "F_C" in Figure 2). We also consider to have a description of the infrastructure (e.g, the servers on which the VMs are to be placed, marked with the letter "I") and we expect as the output a decision (letter "D") indicating the mapping of the VMs over the physical servers.

C. Multi-Resource Consolidation Model

Let us now present a formal model of the proposed multi-resource server consolidation. Let us consider a set $\mathbf{M}$ of VMs that have to be deployed on a set $\mathbf{S}$ of physical servers. We assume that the VMs are divided into a set $\mathbf{C}$ of classes, where all the VMs of a same class present similar resource requirements. Examples of VM classes can be provided considering the software components of a multi-tier application, where Web servers, DBMS servers or specialized component Web services can be mapped into classes. Let $\mathbf{M_c}$ be the set of VMs belonging to class $c \in \mathbf{C}$, and $\mathbf{T}$ the set of time intervals composing the next planning period considered for the server consolidation. The matrix Q represents the resource requirements of the VMs over the future time intervals. Specifically, we consider multiple resources of the VMs, such as CPU, memory, network, and I/O that may be critical for the server consolidation problem [11], [12], [13]. We define the set of possible resources as $\mathbf{R}$. Since VMs belonging to the same class are characterized by similar resource demand, we can

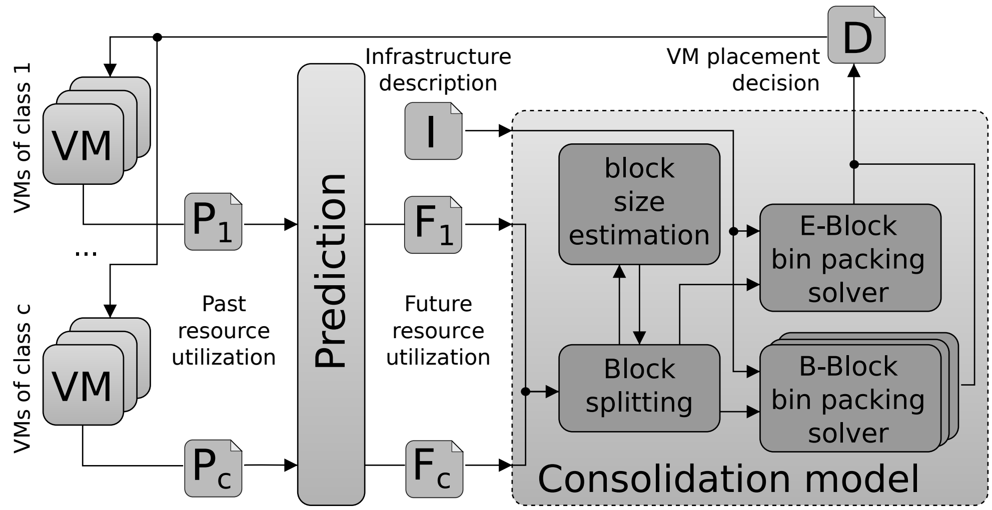

Fig. 2: Class-based VMs placement technique

define the demand for resource r ($r \in \mathbf{R}$) of a generic VM belonging to class c ($c \in \mathbf{C}$) for the time interval t ($t \in \mathbf{T}$) as $Q_{r,c,t}$. Furthermore, $V_{r,s}$ represents the available CPU capacity for resource r on server s ($s \in \mathbf{S}$).

The traditional approach to address the server consolidation problem is to solve the corresponding multi-dimensional bin packing problem (MBP). In such problem multi-dimensionality is due to both the multiple time intervals considered and to the different resources taken into account. The resulting number of constraints makes the global problem computationally intractable for medium-large data centers. To improve the scalability of the server consolidation, a possible solution is to simplify the bin packing problem by reducing the number of considered resources (e.g., considering only CPU) and/or increasing the length of the time intervals considered for resource demand estimation (thus reducing the cardinality of the set $\mathbf{T}$). Unfortunately these solutions tend to reduce the quality of the server consolidation, leading to the use of more physical servers with respect to the optimum. This motivates the proposal of an alternative solution that exploits the Class-based placement approach introduced in [5].

In Class-based placement, the global set of VMs is divided in $\overline{b}$ B-blocks composed by the same number of VMs for each class, while the remaining VMs form the E-block. A block number estimator (shown in Figure 2) determines the number of the B-blocks. For each class $c \in \mathbf{C}$, each B-block contains a set $\mathbf{B_c} \subset \mathbf{M_c}$ of VMs belonging to class c. The remaining set of VMs $\mathbf{E_c}$, that are not assigned to any B-block, is assigned to the E-block.

Since all the VMs of a same class present similar resource requirements, the placement solution computed for a single B-block can be replicated for all the B-blocks. We can thus formulate the optimization problem for the generic B-block as:

$$\min \sum_{s \in \mathbf{S}} O_s \tag{1}$$

subject to:

$$\sum_{s \in \mathbf{S}} I_{s,m} = 1 \qquad \forall m \in \bigcup_{c \in \mathbf{C}} \mathbf{B_c} \tag{2}$$

$$\sum_{c \in \mathbf{C}} \sum_{m \in \mathbf{B_c}} Q_{r,c,t} \cdot I_{s,m} \leq V_{r,s} \cdot O_s \qquad \forall s \in \mathbf{S}, \forall t \in \mathbf{T}, \forall r \in \mathbf{R} \tag{3}$$

$$I_{s,m} = \{0,1\} \qquad \forall s \in \mathbf{S}, \forall m \in \bigcup_{c \in \mathbf{C}} \mathbf{B_c} \tag{4}$$

$$O_s = \{0,1\} \qquad \forall s \in \mathbf{S} \tag{5}$$

Where O_s is a binary decision variable that discriminates if a physical server s in the data center is on or off, $I_{s,m}$ is a binary decision variable that decides if VM m is allocated on server s. Expression 1 is the objective function of the optimization problem that aims to minimize the number of used servers. Due to the set of constraints 2, every VM is allocated exactly on one physical server. The set of constraints 3 expresses the bound that on each server the allocated VMs must not exceed the overall capacity of the server for every considered time interval. Finally, the sets of constraints 4 and 5 model the boolean nature of the decision variables.

A similar optimization problem applies to the E-block problem.

D. Block size estimation

We now focus on how VMs are assigned to the B-blocks and to the E-block. The parameter $\overline{b}$ plays a major role in determining this assignment, hence the selection of the right value for $\overline{b}$ represents a critical factor for the performance of the proposed server consolidation model. The impact of $\overline{b}$ over the consolidation process is twofold. On one hand, as $\overline{b}$ is reduced, the size of the problem in the B- and E-blocks increases. This may have a detrimental effect on the resolvability of the corresponding bin packing problem due to the computational cost of the large-scale optimization required for the B-blocks. On the other hand, as $\overline{b}$ grows, we tend to have very small problems, where the amount of unused resources of the servers in each B-block becomes relevant. In this case we observe a fragmentation effect that may reduce the quality of the solution (the number of physical servers used is much higher than the optimum). The identification of the best value of $\overline{b}$ must solve a trade-off between computational cost and solution quality, ensuring that the splitting of the VMs placement problem is feasible. To this aim, we consider the following three strategies, that will be detailed in the following of this section: (1) Smallest B-block (SB), (2) Smallest B-block with E-block size constraint (SBE), (3) Largest solvable B-block (LB).

In order to formalize the three strategies, we consider the number of VMs belonging to class c ($c \in \mathbf{C}$) which are in the B- and E-blocks, that is $|\mathbf{B_c}|$ and $|\mathbf{E_c}|$, respectively:

$$|\mathbf{B_c}| = \left\lfloor \frac{|\mathbf{M_c}|}{\overline{b}} \right\rfloor \quad \forall c \in \mathbf{C} \tag{6}$$

$$|\mathbf{E_c}| = |\mathbf{M_c}| \% \overline{b} \quad \forall c \in \mathbf{C} \tag{7}$$

1) Smallest B-block (SB): In this case, we choose the maximum possible value for $\overline{b}$, with the only constraint that at least a representative of each class must be present in the B-block, that is:

$$|\mathbf{B_c}| \geq 1 \quad \forall c \in \mathbf{C} \tag{8}$$

This means that $\overline{b} = \min(\{|\mathbf{M_c}|, \forall c \in \mathbf{C}\})$. This solution has been adopted for the original proposal of the class-based placement technique in [5].

2) Smallest B-block with E-block size constraint (SBE): The SB strategy may result in an E-block that is larger than the B-block. An excessive size of the E-block may determine unwanted scalability problems. This concern motivates the proposal of an enhanced version of the SB strategy that places a constraint on the size of the E-block, that we require to be not larger than the B-block. The constraints for the computation of $\overline{b}$ are:

$$|\mathbf{B_c}| \geq 1 \qquad \forall c \in \mathbf{C} \tag{8}$$

$$\sum_{c \in \mathbf{C}} |\mathbf{B_c}| \geq \sum_{c \in \mathbf{C}} |\mathbf{E_c}| \tag{9}$$

In this case, we start with the smallest possible B-block as in the SB strategy and we check constraint 9. If the constraint is not satisfied, we decrease $\overline{b}$ and reiterate the process.

3) Largest solvable B-block (LB): This strategy follows an opposite vision with respect to the previous proposal. Basically, we aim to maximize the size of the B-block to limit the effect of fragmentation due to the unused physical resources in each B-block. The constraints of the strategy are:

$$|\mathbf{B_c}| \geq 1 \qquad \forall c \in \mathbf{C} \tag{8}$$

$$\sum_{c \in \mathbf{C}} |\mathbf{B_c}| \geq \sum_{c \in \mathbf{C}} |\mathbf{E_c}| \tag{9}$$

$$\sum_{c \in \mathbf{C}} |\mathbf{B_c}| \leq H \tag{10}$$

Where constraints 8 and 9 are the same of the previous strategy. Constraint 10 places a limit to the maximum number of VMs in a B-block. This bound is important because in previous studies [5] we found that, as the problem size grows, the bin packing problem becomes unsolvable and cannot be solved. This observation motivates the upper bound on the B-block size and provides an estimate for the threshold H: this is the size of the largest solvable problem, that must be obtained through preliminary experiments.

For the identification of the $\overline{b}$ value, we rely again on an iterative approach. We start with a value of $\overline{b} = \lceil \frac{|M|}{H} \rceil$, that descends from constraint 10: a lower value of $\overline{b}$ would automatically violate this condition. If we find a solution to the problem, then we have an acceptable block splitting. Otherwise, we increment $\overline{b}$ and we try again to solve the optimization problem. The maximum possible value for $\overline{b}$ is $\min(\{|\mathbf{M_c}|, \forall c \in \mathbf{C}\})$: any higher value of $\overline{b}$ would violate the inequality in constraint 8, as in the other strategies.

III. EXPERIMENTAL RESULTS

In this section we start describing the setup of our experiments, then we discuss the results regarding the proposed multi-resource server consolidation model, with a detailed analysis of the impact of different values for the $\overline{b}$ parameter.

A. Experimental setup

We obtained an extensive dataset from a private cloud data center. The set contains up to 1100 VMs traces for the resource usage of Web/application/database servers and ERP applications: the VMs belong to 44 different classes, where each class has a minimum cardinality between 8 and 10 VMs, and a maximum cardinality of 50 VMs. We use our traces as the future resource utilization for the server consolidation model (see Figure 2). For our experiments, we consider three main VMs resources, that are CPU utilization, memory occupation and number of I/O operation per second, each of which may be bottleneck resources for this type of applications [11], [12], [13]. The resource usage is measured in intervals of 5 minutes, that is a setup consistent with other experiments in literature [14].

We consider multiple scenarios characterized by different numbers of VMs to be placed on the physical servers of the virtualized data center. In particular, we consider a VMs set size ranging from 200 to 1100 VMs. For each VM the resource utilization is normalized in the range [0%-100%] and the resulting average values are 56%, 45%, 33% for CPU, memory, and I/O respectively. For each physical server the resource capacity is 400%, meaning that each server can host 4 VMs with resource utilization of 100%. For each scenario, we compare different consolidation models operating over a planning period of 24 hours. The proposed server consolidation model is solved with 288 five-minutes time intervals and is evaluated for different values of the $\overline{b}$ parameter. When evaluating the traditional MBP model (that is the model where a single multi-dimensional bin-packing problem is applied to the whole data center) we consider different setups where the length of the intervals for the three considered resource requirements ranges from 5 minutes to 24 hours. We also consider a First Fit Decreasing (FFD) heuristic [4] that is used to solve very large problems [5]. The experiments are run on 2.4 GHz, 4 cores Intel Xeon with 16 GB RAM, using IBM ILOG CPLEX 12.6 as the optimizer solver[1].

As a metric for the VMs placement quality, we consider the number of physical servers that are required for the allocation [11], [15]. The number of servers for each solution is expressed with respect to an estimation of the optimal solution for the considered scenario. The MBP model with five minute time interval (MBP-5min) represents a lower bound for all the feasible allocations, as this consolidation model exploits all the available information to find an optimal solution. However, the number of variables and constraints for this model increases rapidly with the VMs set size, producing an optimization problem instances whose computation takes extremely long times or does not produce any feasible solution due to the huge main memory requirements, that may finally cause the solver to abort the optimization processing. For this reason, we use the objective function value of the LP relaxation of the MBP-5min consolidation model (1) as a lower bound for the optimal number of physical servers to use. In other words, we relax the boolean nature of the decision variables (constraints 4 and 5), assuming that parts of a VM can be assigned to different physical servers. This allocation is obviously not feasible from a technical point of view but can be easily computed, hence we exploit it as a convenient lower bound for any feasible allocation [11].

It is worth to note that for many problems the resolution of the MPB consolidation models may take long times, such as hours or days, even for a limited number of time intervals. For that reason, we used a time limit of 30 minutes (1800 seconds) for each problem and considered the best integer solution found as the solution of the server consolidation model, as commonly done in similar research studies [11], [16].

B. Estimation of H threshold

The LB strategy for the Class-Based consolidation model is based on a threshold H to define the largest solvable problem. For this estimation we consider the size of the bin packing problem constraints matrix as a measure of the problem size, as suggested in [11]. We recall that the matrix size for the generic multi-dimensional bin packing problem has the following dimensions:

$$size = \Big(|\mathbf{M}|\cdot|\mathbf{S}|+|\mathbf{S}|\Big)\times\Big(|\mathbf{S}|+|\mathbf{R}|\cdot|\mathbf{S}|\cdot|\mathbf{T}|\Big) \qquad (11)$$

where $|\mathbf{M}|$ is the number of VMs, $|\mathbf{S}|$ is the number of physical servers, $|\mathbf{R}|$ is the number of resources taken into account, and $|\mathbf{T}|$ is the number of considered time intervals. For example, considering 400 VMs, 3 resources, and 288 5-minutes time intervals, we have a problem requiring in the order of 56 physical servers (we consider that each server has a resource capacity of 400% and the bottleneck resource average utilization is 56%). The resulting constraints matrix has size $22456 \times 48440 = 1.088 \cdot 10^9$.

Figure 3 provides an analysis of the solver performance in handling problems with different sizes. Specifically, we consider the consolidation model relying on the multi-dimensional bin packing problem (MBP) with different time granularities (from 5 minutes to 24 hours) and we show, for different sizes of the constraints matrix, whether the solver can provide an optimal solution, an integer (but sub-optimal) solution, or no feasible solution at all. We also consider the CBP consolidation model, and we provide the same evaluation. However, for the CBP model the constraints matrix size considered is the size of the global problem provided as the input of the consolidation problem and not the size of the B-block and E-block problems fed into the solver.

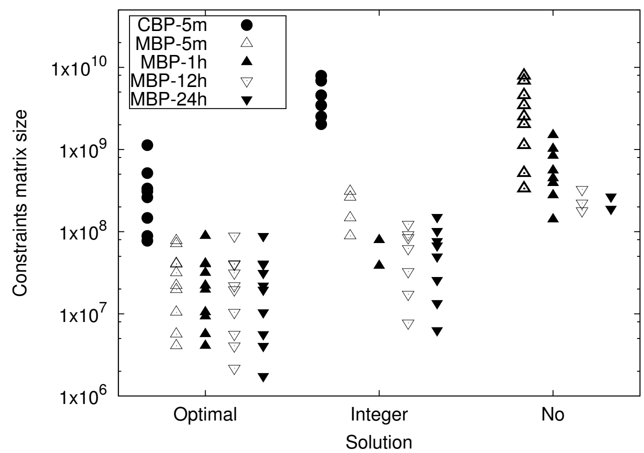

Fig. 3: Problem resolvability *vs.* Problem size

From Figure 3, we observe that for the MBP problem the performance of the solver are rather homogeneous for the different time granularities. Problems with a constraints matrix size up to $1 \cdot 10^8$ can be solved to optimality or, especially for a small number of time interval, can reach an integer solution (our numeric analyses suggest that the obtained solution, even if not guaranteed to be the optimal value, is usually very close to the optimum). Problems with a size up to $1.2 \cdot 10^8$ can be solved reaching an integer solution, but, when the constraints matrix size exceeds $1.2 \cdot 10^8$, the solver is no longer able to find an integer solution, either optimal or sub-optimal within the assigned time limit. From this observation, reversing the formula 11 we can derive the value for the threshold H that is used in constraint 10 of the LB strategy for the CBP consolidation model. In our case we obtain $H = 250$.

A final remark from the analysis of Figure 3 concerns the scalability of the CBP consolidation model compared to the traditional MBP model. The graph clearly demonstrates that the subdivision into small problems (in this case we use the LB strategy, but other strategies provides similar results) provides a major benefit from a scalability point of view. Indeed, the CBP consolidation can find solution for problems with a constraints matrix with a size close to $1 \cdot 10^{10}$, while the maximum size of a problem that can be handled by MBP is $5 \cdot 10^8$. Furthermore, we observe that the CBP consolidation model is always able to provide a problem solution, as testified by the lack of circle points in the "No Solution" part of graph.

C. Number of blocks selection

We now focus on the CBP consolidation model and we compare the different strategies for the selection of the number

[1] www.ibm.com/software/commerce/optimization/cplex-optimizer/

of B-blocks $\overline{b}$ outlined in Section II-D. To this aim, we compute the values of $\overline{b}$ for the three proposed strategies (that is SB, SBE, LB) for different sets of VMs.

Figure 4 shows the number of VMs in the B-blocks and E-block for an example case study with 1100 VMs. The parameter $\overline{b}$ starts from 2 (in this case we omit the case $\overline{b} = 1$ where the CBP consolidation model would revert to the MBP model) and reaches the value of 10 that is cardinality of the smallest VM class, that is $\min(\{|\mathbf{M_c}|, \forall c \in \mathbf{C}\})$.

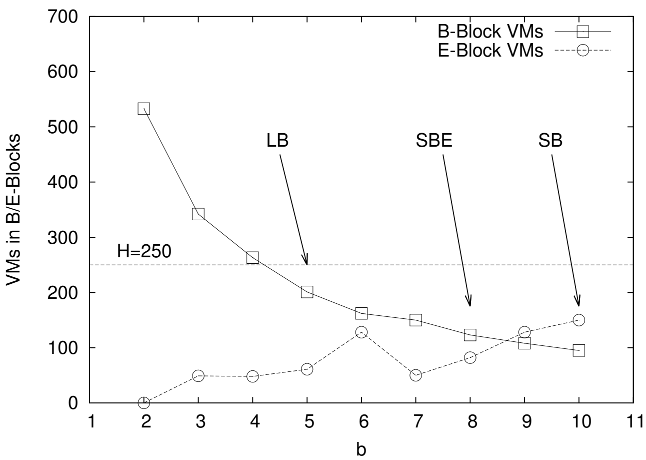

Fig. 4: Evaluation of 1100 VMs set size

The squares in Figure 4 represent the B-block size, while the circles are the E-block size. The threshold value $H = 250$ is represented as a dashed horizontal line. The three arrows highlight the values of $\overline{b}$ chosen by the three considered strategies.

For SB, the choice is simple: as the smallest class contains 10 VMs, the value of $\overline{b}$ selected by such strategy is 10. However, for this value of $\overline{b}$ we have an E-block that is more than 50% larger than the B-block. The SBE strategy rejects this value and decreases the value of $\overline{b}$ to 8, that is when the E-block becomes smaller than the B-block (thus satisfying constraint 9). The LB strategy uses a different method to estimate $\overline{b}$ starting with the smallest possible value and increasing until a problem of solvable size is found. In Figure 4 this occurs when the size of the B-block drops below the value of H and the E-block is still smaller than the B-block. In the considered example the value of $\overline{b}$ obtained by the LB strategy is 5.

It is worth to note that in this experiment we also validate the process for finding the value of H. Indeed, in our tests we found that all the problems with a B-block size selected using the LB strategy are solvable, while larger problem (that is problem with a value of $\overline{b}$ lower than the one found with the LB strategy) cannot be solved within the given time.

The last experiment compares the quality solutions obtained by the proposed server consolidation model exploiting the three different strategies (SB, SBE, LB) to determine the $\overline{b}$ parameter. We also consider as a term of comparison the solution achieved by state-of-the-art solutions based on Multiple Bin Packing (MBP) or FFD heuristic applied to the global placement problem. Table I shows the solution qualities for the considered consolidation models and $\overline{b}$ determination strategies for a VM set size that ranges from 200 to 1100 VMs. For the CBP consolidation model (first three columns), we report the $\overline{b}$ parameter value identified by each strategy. In the last column of the table, along with the solution quality, we report the state-of-the-art consolidation model that achieved the solution: we observe that for very large scenarios (1100 VMs) only the FFD heuristic is able to find a feasible integer solution, while feasible solutions can be achieved by MBP models with an increasing number of time intervals as the VM set size decreases.

TABLE I: Solution quality [%]

VMs Set Size	**Consolidation Models**			
	Class-Based			**State of the art solution**
	SB	**SBE**	**LB**	
200	134.02 ($\overline{b} = 8$)	110.56 ($\overline{b} = 6$)	101.34 ($\overline{b} = 1$)	101.34 (MBP-5m)
300	105.88 ($\overline{b} = 8$)	105.88 ($\overline{b} = 8$)	102.94 ($\overline{b} = 2$)	119.5 (MBP-1h)
400	110.25 ($\overline{b} = 8$)	110.25 ($\overline{b} = 8$)	107.69 ($\overline{b} = 2$)	126.76 (MBP-12h)
500	115.38 ($\overline{b} = 8$)	112.97 ($\overline{b} = 7$)	103.84 ($\overline{b} = 2$)	129.07 (MBP-12h)
600	111.29 ($\overline{b} = 8$)	110.89 ($\overline{b} = 7$)	106.76 ($\overline{b} = 3$)	131.43 (MBP-12h)
700	116.18 ($\overline{b} = 8$)	114.76 ($\overline{b} = 6$)	108.37 ($\overline{b} = 3$)	135.23 (MBP-12h)
800	114.91 ($\overline{b} = 10$)	111.81 ($\overline{b} = 8$)	107.97 ($\overline{b} = 3$)	134.16 (MBP-12h)
900	115.8 ($\overline{b} = 10$)	112.78 ($\overline{b} = 8$)	108.61 ($\overline{b} = 4$)	135.25 (MBP-12h)
1000	113.81 ($\overline{b} = 10$)	112.02 ($\overline{b} = 7$)	108.95 ($\overline{b} = 4$)	136.02 (MBP-1d)
1100	115.23 ($\overline{b} = 10$)	114.63 ($\overline{b} = 8$)	109.73 ($\overline{b} = 5$)	136.92 (FFD)

The message from Table I is manifold. First, the LB strategy for the determination of $\overline{b}$ allows the CBP model to obtain the best result in the solution quality for every VMs set size. Indeed, the quality of the solutions in the corresponding third column ranges from 101.34% to 109.73%, while for the SBE and SB strategies the quality ranges from 105.88% to 114.76% and from 105.88% to 134.02%, respectively. This result is motivated by the large size of the B-blocks that tends to limit the effect of fragmentation due to the unused physical resources in each building block problem. The second important result is that, for each VMs set that exceeds the maximum size solvable through a MBP-5m consolidation model, the CBP model significantly outperforms the state-of-the-art solutions for any value of the $\overline{b}$ parameter. Indeed, for a number of VMs above or equal to 300, even the worse choice of $\overline{b}$ leads to a gain in the solution quality at least equal to 13.69% (case of 500 VMs). This is due to the capability to take advantage of the characteristics of complementary workloads, that can not be exploited by MBP consolidation models with relaxed time granularity. Finally, we observe that the solution quality achieved by the CBP consolidation model with LB choice of $\overline{b}$ is equal to that of the MBP-5m when the VMs set size if lower that 250 VMs. In this case (first row of the table), the LB strategy automatically recognizes that the problem can be solved with time constraints of 5 minutes without the need of splitting the data in more blocks ($\overline{b} = 1$), bringing back the problem to a MBP-5m model.

IV. Related Work

The management of cloud data centers is posing new challenges due to the growing size and complexity of these infrastructures. In particular, the placement of VMs over the physical servers of the data center represents a critical task to limit the costs of the infrastructure management and avoid waste of computing resources. An efficient placement aims to minimize the number of physical servers required to allocate a given set of VMs in a cloud data center, while meeting the VMs requirements in terms of system resources. Large data centers can exploit techniques such as selectively powering down idle servers or using hardware support for idle sleep states [17] to improve their efficiently. However, exploiting these techniques requires the resolution of the optimization problem described in Section II-B, to determine how to map VMs over the physical servers of the cloud infrastructure. This problem is a multi-dimensional bin-packing with bounds related to the requirement of multiple VM resources at different time intervals over a future planning period. Solving this problem is a challenge from a computational point of view, where standard optimization algorithms struggle to reach an optimal solution within acceptable time frames.

Solutions to reduce the dimensionality of the problem have been proposed in literature or applied in real systems. For example, some data centers [2], [3], [18] discard any information about VM demands over time and consider only the nominal maximum requirements of each VM. This approach is very effective in simplifying the bin-packing problem and is therefore widely adopted, but introduces the unreal assumption that every VM uses the 100% of its resources. Any underutilized VM determines a waste of resources in the data center and increases the carbon footprint of the cloud infrastructure. Heuristics have been proposed to reduce the computational cost of the solution. However, as pointed out in [20], most research is focused on problems with few dimensions (e.g., from one to three [21], [22]), while if we consider the impact of multiple resources considered in multiple time intervals in a future planning period, the number of dimensions is in the order of several hundreds. Finally, the last solution is to introduce simplification in the bin packing problem model. For example, instead of considering multiple resources (CPU, memory, network I/O, disk I/O) and a fine-grained division of the planning period, the focus is limited to just the CPU requirement during a 24-hour long time interval [19], [11].

A recent approach to address the scalability issues of the VMs placement problem has been proposed in [5]; however, this preliminary proposal did not address the problem of determining the best number and size of B-blocks. A first investigation of this problem is provided in [7], where the proposed consolidation model considers only one resource, that is the CPU usage. The present paper further extends the previous work in two ways: first, we consider an improved model taking into account multiple VMs resources; second, we propose a more comprehensive set of experiments, providing a detailed comparison of multiple strategies to select the number and size of the B-blocks.

V. Conclusions

In this paper we tackle the problem of parameter tuning in the Class-based placement technique, explicitly addressing the case where the VMs placement takes into account multiple resources. Specifically, we consider the parameter $\overline{b}$ that is the number of B-blocks in which the global VMs placement problem is split. We pointed out a trade-off between using many small building blocks (with the risk of reducing the overall VMs placement solution quality due to the fragmentation of the physical server resources over blocks) and using few large building blocks (with the risk of being unable to solve the optimization problem for the B-block). We compare three different strategies to determine the $\overline{b}$ parameter applied to a real scenario based on a cloud infrastructure. Our experiments demonstrate that a simple threshold determined through preliminary experiments can guarantee the identification of a solvable B-block problem. The use of this threshold allows the CBP consolidation model to exploit large building blocks, thus avoiding the fragmentation of physical server resources that are likely to reduce the placement solution quality.

References

[1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the Future*, 2012, http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf.

[2] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas *et al.*, "Reservoir - When one cloud is not enough," *IEEE computer*, vol. 44, no. 3, pp. 44–51, 2011.

[3] K. Mills, J. Filliben, and C. Dabrowski, "Comparing VM-Placement Algorithms for On-Demand Clouds," in *Proc. of IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2011, pp. 91–98.

[4] M. Kao, *Encyclopedia of Algorithms*. Springer, 2008.

[5] C. Canali and R. Lancellotti, "Exploiting Classes of Virtual Machines for Scalable IaaS Cloud Management," in *Proc. of the 4th Symposium on Network Cloud Computing and Applications (NCCA)*, Munich, Germany, Jun. 2015.

[6] M. Andreolini, S. Casolari, and M. Colajanni, "Models and framework for supporting runtime decisions in Web-based systems," *ACM Transactions on the Web*, vol. 2, no. 3, pp. 1–43, 2008.

[7] C. Canali and R. Lancellotti, "Automatic parameter tuning for class-based virtual machine placement in cloud infrastructures," in *Proc. of 23rd International Conference on Software, Telecommunications and Computer Networks*, ser. SoftCOM 2015. IEEE, Sep. 2015.

[8] M. Rabinovich and O. Spatscheck, *Web caching and replication*. Addison-Wesley Boston, USA, 2002.

[9] C. Canali and R. Lancellotti, "Improving Scalability of Cloud Monitoring through PCA-Based Clustering of Virtual Machines," *Journal of Computer Science and Technology*, vol. 29, no. 1, pp. 38–52, 2014.

[10] ——, "Automatic virtual machine clustering based on Bhattacharyya distance for multi-cloud systems," in *Proc. of International Workshop on Multi-cloud Applications and Federated Clouds*, Prague, Czech Republic, Apr. 2013, pp. 45–52.

[11] T. Setzer and M. Bichler, "Using matrix approximation for high-dimensional discrete optimization problems: Server consolidation based on cyclic time-series data," *European Journal of Operational Research*, vol. 227, no. 1, pp. 62–75, 2013.

[12] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers," *Computer Networks*, vol. 57, no. 1, pp. 179 – 196, 2013.

[13] R. Zhang, R. Routray, D. M. Eyers, D. Chambliss *et al.*, "IO Tetris: Deep storage consolidation for the cloud via fine-grained workload analysis," in *Proc. of IEEE 4th International Conference on Cloud Computing (CLOUD)*, Washington, USA, Jul. 2011.

[14] B. Addis, D. Ardagna, B. Panicucci, M. S. Squillante, and L. Zhang, "A hierarchical approach for the resource management of very large cloud platforms," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 5, pp. 253–272, 2013.

[15] D. Breitgand and A. Epstein, "Improving consolidation of virtual machines with risk-aware bandwidth oversubscription in compute clouds," in *Proc. of IEEE INFOCOM*, Orlando, FL, March 2012.

[16] L. Zhang and D. Ardagna, "SLA Based Profit Optimization in Autonomic Computing Systems," in *Proc. of International Conference on Service Oriented Computing (ICSOC)*, New York, USA, Nov. 2004.

[17] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE computer*, vol. 40, no. 12, pp. 33–37, 2007.

[18] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A scalable application placement controller for enterprise data centers," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 331–340. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242618

[19] T. Setzer and A. Stage, "Decision support for virtual machine reassignments in enterprise data centers," in *Proc. of Network Operations and Management Symposium*, Osaka, Japan, Apr. 2010.

[20] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007.

[21] O. Faroe, D. Pisinger, and M. Zachariasen, "Guided local search for the three-dimensional bin-packing problem," *Informs journal on computing*, vol. 15, no. 3, pp. 267–283, 2003.

[22] T. G. Crainic, G. Perboli, and R. Tadei, "Ts 2 pack: A two-level tabu search for the three-dimensional bin packing problem," *European Journal of Operational Research*, vol. 195, no. 3, pp. 744–760, 2009.