

Exploiting Classes of Virtual Machines for Scalable IaaS Cloud Management

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—A major challenge of IaaS cloud data centers is the placement of a huge number of Virtual Machines (VMs) over a physical infrastructure with a high number of nodes. The VMs placement process must strive to reduce as much as possible the number of physical nodes to improve management efficiency, reduce energy consumption and guarantee economical savings. However, since each VM is considered as a black box with independent characteristics, the VMs placement task presents scalability issues due to the amount of involved data and to the resulting number of constraints in the underlying optimization problem. For large data centers, this condition often leads to the impossibility to reach an optimal solution for VMs placement. Existing solutions typically exploit heuristics or simplified formulations to solve the placement problem, at the price of possibly sub-optimal solutions. We propose an innovative VMs placement technique, namely *Class-Based*, that takes advantage from existing solutions to automatically group VMs showing similar behavior. The *Class-Based* technique solves a placement problem that considers only some representatives for each class, and that can be replicated as a *building block* to solve the global VMs placement problem. Our experiments demonstrate that the proposed technique is viable and can significantly improve the scalability of the VMs placement in IaaS Cloud systems with respect to existing alternatives.

I. INTRODUCTION

Cloud computing is emerging as an innovative paradigm for the deployment of novel applications that must face highly variable workloads and to reduce the costs associated to the management of the ICT infrastructure. The typical platform for cloud computing provides computational, storage and networking resources with a on-demand, pay-as-you-go philosophy. The expected growth in the amount of data stored and processed in cloud systems by two order of magnitude in fifteen years [1] is a clear sign of the cloud success. The growth of cloud computing is leading to the deployment of larger and even more powerful data centers, hosting an always increasing number of Virtual Machines (VMs). The complexity of such infrastructures opens new challenges for automated monitoring and management processes. In particular, a critical task for the management of IaaS cloud computing data centers is the VMs placement on the physical nodes of the infrastructure, that presents scalability issues as the data centers grow in size. The typical problem formulation is that of a multi-dimensional bin packing where the optimization goal is to minimize the number of physical nodes required to host the VMs, and the capacity requirements of each VM correspond to the future demand of multiple resources (e.g., CPU, memory, network traffic) at different times in the future [2], [3]. However, as the

nature of problem is *NP-hard*, reaching an optimal solution for the problem may be not feasible as the number of VMs increases, because of the huge amount of time and memory taken by the optimization problem solver. The state of the art solutions typically rely on simplifications to reduce the dimensionality of the problem and/or on simple heuristics to reduce its computational cost. The most straightforward approach to reduce the problem dimensionality is to consider only the nominal capacity of each VM [4], [5], [6] instead of taking into account their actual future requirements. This solution increases the scalability of the VMs placement because it removes a major source of additional dimensions in the bin packing problem, that is the change over time of VMs resource demand. However, considering only the nominal capacities has the major drawback of overestimating the resources that must be provided to the VMs, because the actual utilization of resources for each VM is typically below 100% [7]. Hence, a model based on nominal capacity determines an inefficient use of the cloud data center, resulting in a higher-than-needed number of physical nodes used for the overall infrastructure. Another approach is to reduce the dimensionality of the problem by limiting the number of resources that are considered in the bin packing problem and/or the number of time intervals that are considered for the constraints of the optimization problem [8], [3]; but, again, this approach is likely to result in sub-optimal solutions. Furthermore, even with these approaches the computational cost for solving the VM placement problem remains rather high, especially for large data centers, to the point where the time to obtain a solution may be not acceptable for the management of the infrastructure. To address these issues, simple heuristics are typically preferred to more complex and computationally expensive approaches [9] because they can provide a VM placement solution in a short time. However, the most widespread heuristics, such as First Fit Decreasing (FFD) [10], can only manage few dimensions of the placement problem, thus hindering the use of multiple time intervals. Hence, adoption of heuristics reduces the placement solution quality. We can summarize that state of the art approaches for the VMs placement problem typically fails to consider the actual behavior of VMs and/or rely on simple heuristics, producing in both cases low quality solutions that lead to a waste of cloud data center resources.

The main contribution of this paper is the proposal of a novel technique for VMs placement over the physical nodes of a cloud data center. The proposed technique, namely *Class-Based Placement*, leverages the knowledge of *classes* of VMs

with a similar behavior in terms of resource usage: to this aim, we exploit recent methodologies to cluster together similar VMs [11], [12] in cloud systems. Through our proposal, we shift the point of view from a single bin-packing problem, that considers the whole data center, to a much smaller problem, limited to a few representatives of each class, that can be replicated as a building block to create the solution for the global VM placement problem. The small size of the building-block problem can be solved to optimality in short time even taking into account an amount of data and constraints that would not be possible to consider in the global bin-packing problem. We claim that our technique can reduce the computational demand while achieving higher quality in the VM placement solution compared to existing alternatives. To the best of our knowledge, no other study follows this approach to obtain in short time a high-quality solution for the VMs placement problem in cloud computing.

We apply our technique to traces obtained from a real data center to evaluate the feasibility of the proposed solution. We compare the quality of our solution with state of the art models for VMs placement [2]. Our results demonstrate that: (1) exploiting similarities among VMs provides a viable solution for the VM placement problem in IaaS clouds; (2) standard techniques based on the solution of optimization problem solvers cannot reach optimal solutions for the bin-packing problem unless the number of VM is rather small (in the order of 150-200 VMs); even worse, in the case of large problems (in the order of 1000 VMs), the solvers cannot obtain any integer feasible solution within a reasonable time frame; (3) with the exception of very small problems, our proposal outperforms existing techniques from both the points of view of resolution time and quality of the solution. This last result is very important because cloud data centers tend to host a very large number of VMs: hence, the typical cloud scenario corresponds to the case where our proposal provides major benefits.

The remainder of this paper is organized as follows. Section II describes the reference scenario for our proposal, while Section III describes our model for solving the VM placement problem. Section IV describes the results of the methodology evaluation. Finally, Section V discusses the related work and Section VI concludes the paper with some final remarks and outlines open research problems.

II. REFERENCE SCENARIO

In this section we describe the reference scenario for our proposal. Starting from this scenario, we illustrate the characteristics of the proposed technique for the management of a cloud data center, focusing on the operations that decide the placement of the VMs over the physical nodes of the infrastructure.

The application of our proposal to a cloud data center is based on the following two assumptions.

First, we consider that the VMs placement is a periodic task that aims at mapping VMs over the physical nodes of the infrastructure with the goal of minimizing the number of used nodes, while satisfying the requirements of each VM in terms of resource usage. The details of the VM placement process

are shown in Figure 1 and described in the following of this section.

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: in this case, a dispatcher distributes the requests among the VMs running the same software component of an application, with the goal of balancing the application load; this mechanism ensures that the VMs of same class exhibit a similar behavior in terms of resource requirements [13]. Automatic methodologies to cluster VMs with similar behavior have been recently proposed in literature. Some solutions require a long time of observation to define a VM behavior model [11], [14], [12] and are more suitable for IaaS cloud characterized by long term commitment of the VMs customer (as in the case of the Amazon cloud reserved instances), while other methodologies can provide rapidly a preliminary classification [15] and are suitable for a more dynamical scenario. Another scenario where we have knowledge of VM classes is the case of private clouds or infrastructure supporting a SaaS cloud, as the cloud provider has a complete knowledge of the software running on the VMs.

Figure 1 depicts the periodic VMs placement in a cloud data center that adopts the proposed approach. We start from multiple VMs that are grouped into classes (left margin of the Figure): we recall that VMs of the same class exhibit similar resource requirements over time. It is worth to note that the VM resource usage is constantly monitored, and the monitoring process may take advantage of the knowledge of VM classes to reduce the amount of data collected and improve scalability, as discussed in [11], [12]. The output of the monitoring process is represented as the data objects marked as “ P_1 ”, ..., “ P_c ”, grouped by class. Such samples on the past resource usage are fed into a *Prediction* step. This task can be implemented according to multiple techniques, ranging from the simplest solutions assuming that resource demands follow a periodical cycle with a length of 24 hours [16], to complex predictive techniques that can cope with trends, periodic behaviors and state changes [17]. The output of the prediction is an estimation of the resource usage in the future for each class of VMs (data marked with “ F_1 ”, ..., “ F_c ”). The future demands and the description of the infrastructure of the data center (marked with the letter “I”) are the input of the *Consolidation model*, that is the core of our proposal. The consolidation model is based on the bin-packing problem and its output is a solution of such problem that contains the decision (marked with letter “D” in Figure 1) on the mapping between VMs and physical nodes. The placement decision is then applied to the VMs by powering on and off the physical nodes of the cloud infrastructure.

III. PROBLEM FORMULATION

We now discuss the consolidation model that represents the core of the VMs placement technique. First, we describe the consolidation model that is typically used in literature [2], [3]; then, we discuss the possible simplifications that can be applied to improve the scalability of this task, and we present the Class-based consolidation model proposed in the paper.

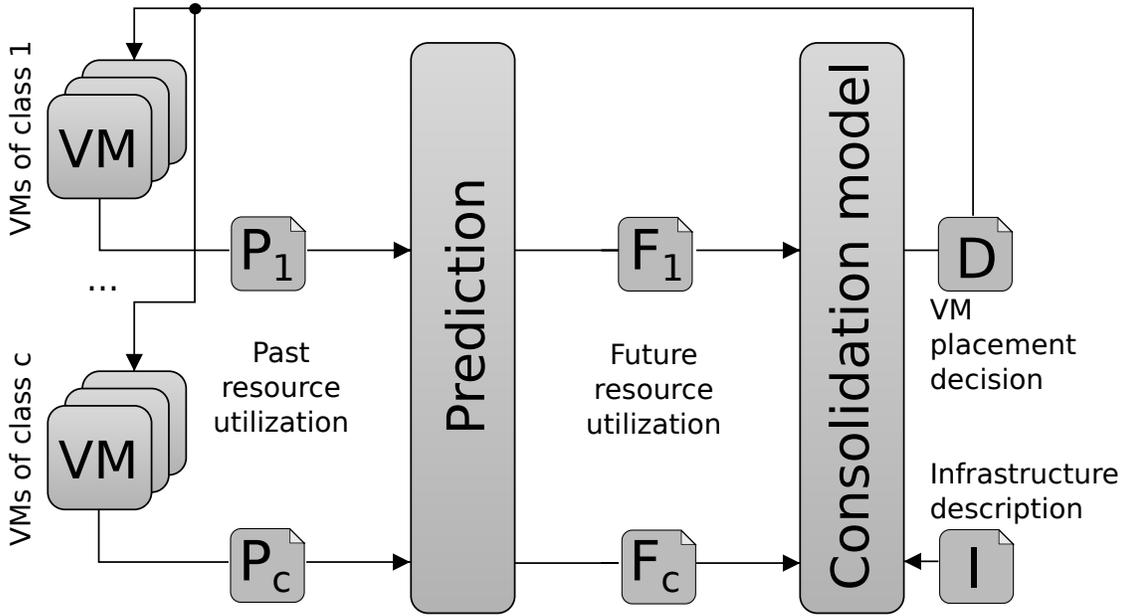


Fig. 1: VMs placement in a cloud data center

A. Multi-Dimensional Bin Packing model

The consolidation model used for VMs placement is typically based on a multi-dimensional bin packing problem, where one or more VM resources are considered for consolidation during the next planning period, and the planning period is divided into a set of time intervals \mathbf{T} . The multi-dimensional bin-packing model is shown in Figure 2. The model input is the prediction of future requirements for every VM in multiple time intervals (the data with the letter “F”). In this case we do not divide the future requirements by class as in Figure 1 because this consolidation model is not class-aware. An additional input of the consolidation model is a description of the data center infrastructure, with the available physical nodes and their capacity (the data of the letter “I”). A single bin-packing problem is solved for the whole data center providing the placement of VMs over the nodes of the data center (the output is represented as the data with the letter “D”). The problem can be formalized as follows.

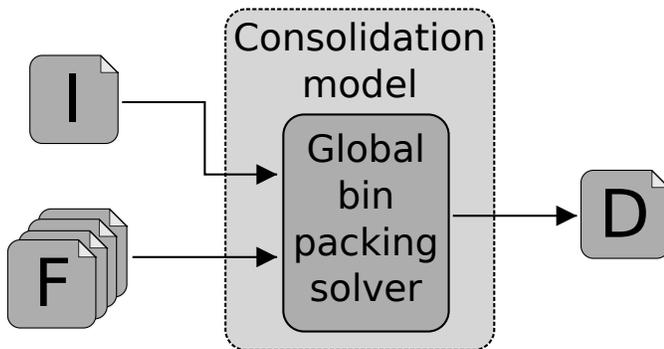


Fig. 2: Consolidation model with multi-dimensional bin-packing

Let us consider a set \mathbf{M} of VMs that have to be deployed on a set \mathbf{N} of physical nodes. The matrix R represents the

resource requirements of the VMs over multiple observation time intervals. Although the most general version of the problem involves multiple resources of the VMs, such as CPU, memory, network, and disk [2], [18], [19], we limit our model to a single resource, that is the CPU utilization that is typical bottleneck resource for cloud infrastructure [20]. However, it is worth to note that an extension of our model to include multiple resource is straightforward. In our model $R_{m,t}$ represents the CPU requirement of VM m ($m \in \mathbf{M}$) for the time interval t ($t \in \mathbf{T}$). Furthermore, for each node, V models the available capacity on the node. Specifically, V_n represents the available CPU capacity on node n ($n \in \mathbf{N}$). We can define the optimization problem as follows:

$$\min \sum_{n \in \mathbf{N}} O_n \quad (1)$$

subject to:

$$\sum_{n \in \mathbf{N}} I_{n,m} = 1 \quad \forall m \in \mathbf{M} \quad (2)$$

$$\sum_{m \in \mathbf{M}} R_{m,t} \cdot I_{n,m} \leq V_n \cdot O_n \quad \forall n \in \mathbf{N}, \forall t \in \mathbf{T} \quad (3)$$

$$I_{n,m} = \{0, 1\} \quad \forall n \in \mathbf{N}, \forall m \in \mathbf{M} \quad (4)$$

$$O_n = \{0, 1\} \quad \forall n \in \mathbf{N} \quad (5)$$

Where O_n is a binary decision variable that discriminates if a physical node n in the data center is on or off, $I_{n,m}$ is a binary decision variable that decides if VM m is allocated on node n . Expression 1 is the objective function of the optimization problem that aims to minimize the number of used nodes. Due to the set of constraints 2, every VM is allocated exactly on one physical node. The set of constraints 3 expresses the bound that on each node the allocated VMs must not exceed the overall capacity of the node for every considered time interval. Finally,

the sets of constraints 4 and 5 model the boolean nature of the decision variables.

When solving bin packing problems, the number of dimensions (in this case the number of time intervals $|\mathbf{T}|$ considered in our problem formulation) has major impact on the time to reach a solution. To improve the scalability of VMs placement problem, a common approach is to reduce the cardinality of constraints 3 in the problem formulation. To this aim, we introduce a different set of time intervals \mathbf{T}' such that $|\mathbf{T}'| < |\mathbf{T}|$ and we bind each new interval $t' \in \mathbf{T}'$ to a set of time intervals $\{t_1, \dots, t_k\} \in \mathbf{T}$. The new constraint formulation will consider for each VM m a requirement $R_{m,t'} = \max(R_{m,t_1}, \dots, R_{m,t_k})$. In the extreme case, when the number of time intervals is reduced to one, the multi-dimensional bin packing reverts to a one-dimensional bin packing problem. In this case, we can exploit heuristics such as the First Fit Decreasing (FFD) algorithm to reach an approximate solution of the problem in a very short time [10]. However, the reduction of dimensionality typically leads to suboptimal solutions for the VM placement problem.

B. Class-Based Consolidation Model

The Class-based consolidation model aims to leverage the knowledge of classes of VMs with similar behavior in terms of resource usage. It is worth to note that, by exploiting recently proposed techniques [11], [12] that automatically cluster similar VMs, such knowledge can be obtained even in IaaS cloud systems, where the cloud providers typically do not have any knowledge of the applications running on the VMs. The basic idea is to reduce the global bin packing problem, that operates on the whole data center, to a smaller problem involving only few VMs for each class. The reduced size of the problem allows us to solve to optimality the consolidation model 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 solution for the global VM placement problem.

The Class-Based placement is shown in Figure 3. Again the future VMs requirements are given as input, but in this case 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 (data on resource requirements are labeled from “F₁” to “F_c” for the different classes). The basic idea is to divide the global set of VMs in a number \bar{b} (given as input to the consolidation model) of B-blocks, all composed by the same number of VMs for each class, and one E-block containing the rest of the VMs. These blocks of reduced size are exploited to determine the global solution to the VMs placement problem, as formalized in the rest of this section.

For each class $c \in \mathbf{C}$, we define a set of VM \mathbf{M}_c belonging to that class such that:

$$\begin{aligned} \bigcup_{c \in \mathbf{C}} \mathbf{M}_c &= \mathbf{M} \\ \mathbf{M}_{c_1} \cap \mathbf{M}_{c_2} &= \emptyset \quad \forall c_1, c_2 \in \mathbf{C} \end{aligned}$$

We recall that the VMs belonging to the same class run the same software component of an application, so we can assume

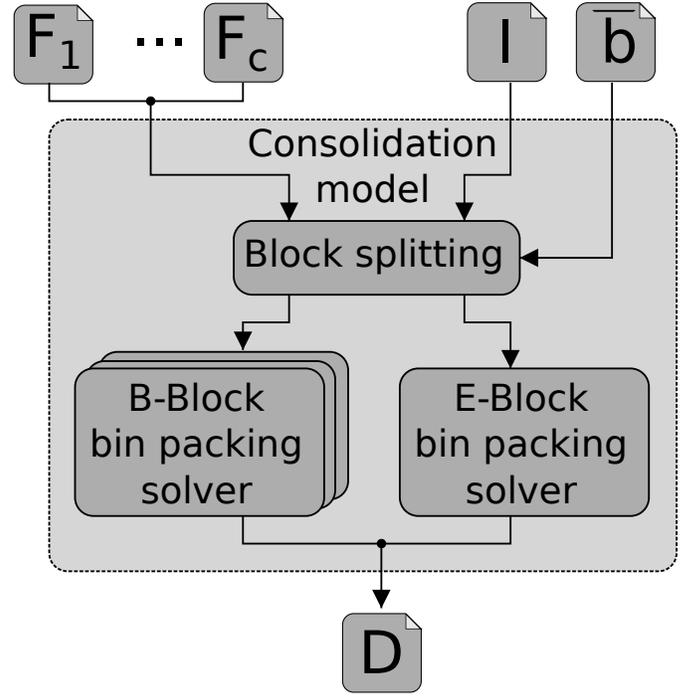


Fig. 3: Class-based consolidation model

that they are characterized by similar resource demand. Hence, we can define their resource demand as:

$$R_{m,t} = R_{c,t} \quad \forall c \in \mathbf{C}, \forall m \in \mathbf{M}_c, \forall t \in \mathbf{T} \quad (6)$$

The global set of VMs is divided in \bar{b} B-blocks composed by the same number of VMs for each class, while the rest of the VMs form the E-block. For each class $c \in \mathbf{C}$, each B-block contains a set $\mathbf{B}_c \subset \mathbf{M}_c$ of VMs belonging to that class c . The remaining set of VMs \mathbf{E}_c that are not assigned to any B-block, is assigned to the E-block. Given the number of B-blocks \bar{b} , the cardinality of each set is:

$$\begin{aligned} |\mathbf{B}_c| &= \lfloor |\mathbf{M}_c| / \bar{b} \rfloor \quad \forall c \in \mathbf{C} \\ |\mathbf{E}_c| &= |\mathbf{M}_c| \% \bar{b} \quad \forall c \in \mathbf{C} \end{aligned}$$

Since all the VMs of a same class present similar resource requirements, the placement solution computed for a single B-block can be replicated on all the remaining B-blocks. The B-block sub-problem is an optimization problem similar to the one in Section III-A, but applied to the subset of VMs $\bigcup_{c \in \mathbf{C}} \mathbf{B}_c$. Considering the presence of VMs classes (Equation 6), we can express the constraint 3 as:

$$\sum_{c \in \mathbf{C}} \sum_{m \in \mathbf{B}_c} R_{c,t} \cdot I_{n,m} \leq V_n \cdot O_n \quad \forall n \in \mathbf{N}, \forall t \in \mathbf{T}$$

A similar set of constraints applies to the E-block problem.

To summarize, the global placement solution can be obtained through the following steps: 1) solving the placement problem for one B-block and replicate the solution for all

the B-blocks; 2) solving the placement problem for the E-block. The reduced size of these blocks allows us to solve the corresponding placement problems considering a multi-dimensional formulation with several time intervals within an amount of time that is acceptable for a cloud data center management.

It is worth to note that the choice of the parameter \bar{b} would deserve a more accurate analysis, but in this preliminary work we assume that \bar{b} is equal to the cardinality of the smaller class in the system: $\bar{b} = \min_{c \in C} (|\mathbf{M}_c|)$. We leave further analysis about the choice of \bar{b} and its impact on the VMs placement solution as a future work.

IV. EXPERIMENTAL EVALUATION

In this section we present the results of the experimental evaluation regarding the proposed placement technique. For our evaluation, we consider the quality of the solution and the corresponding resolution time.

A. Experimental setup

We obtain an extensive dataset from a private cloud data center. The set contains up to 1200 VMs traces for the resource usage of Web/application/database servers and ERP applications, where the VMs belongs to 44 different classes, with each class containing from 8 to 50 VMs. We use our traces as the output from the prediction step in the VM placement problem. For our experiments we consider the CPU resource, that is well-know to be the bottleneck resource for this type of applications [20]. The resource usage is measured in intervals of 5 minutes, that is a setup consistent with other experiments in literature [21].

We consider multiple scenarios characterized by different numbers of VMs to be placed on the physical nodes of the virtualized data center. In particular, we consider a VMs set size ranging from 150 to 1200 VMs. For each VM the CPU utilization is in the range [0%-100%] with an average value of 54%. For each physical node the CPU capacity is 800%, meaning that each node can host 8 VMs with CPU utilization of 100%. For each scenario, we compare different consolidation models operating over a planning period of 24 hours. The proposed Class-Based Placement (CBP) is solved with 288 five-minutes time intervals and the \bar{b} parameter is set to the size of the smallest class that is 8. For the Multiple Bin Packing (MBP) model, we consider multiple setups with a different dimensionality of the problem (in terms of number of time constraints). The considered numbers of dimensions for the MBP model are 288 (five-minutes intervals), 24 (1 hour), 2 (12 hours) and a single time interval (24 hours). We have also implemented a First Fit Decreasing (FFD) heuristic [10], that we used as a term of comparison to evaluate the placement solution quality. The FFD heuristic does not allow to consider time constraints, hence it is evaluated with one time interval of 24 hours. All the experiments are run on 2.4 GHz, 16 cores Intel Xeon with 16 GB RAM, using IBM ILOG CPLEX 12.6 as the optimizer solver¹.

As a metric for the VMs placement quality, we consider the number of physical nodes that are required for the allocation. The number of nodes 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 will 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 nodes to use. In other words, we relax the boolean nature of the decision variables, assuming that parts of a VM can be assigned to different physical nodes. In the formulation this corresponds to simply removing the constraint 4. 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[2].

It is worth to note that for many problems, starting from a medium size (e.g 400 VMs), 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 placement problem, as commonly done in similar research studies [2], [22].

B. Experimental results

We start our experiments with a simple comparison of the different consolidation models to evaluate if they can reach an optimal or a viable solution within the expected time limit. Table I shows for which scenarios it was possible to solve the problem instances to optimality (S), reach an integer solution even if not optimal (L), or not even find any feasible integer solution (N) within the 30 minutes time limit. We evidence the cells related to unsolvable problem instances with a gray background. For the CBP technique (second column of the table) we report two values separated by a slash symbol: the first refers to the B-block and the second to the E-block. In this first analysis, we omit the FFD heuristics and focus only on consolidation models relying on the use of an optimization problem solver (due to the heuristic simple nature, FFD always reaches a feasible solution within 30 minutes). A More detail discussion of the FFD heuristics is provided in the following of the section, when we discuss the results about the quality of the provided solution.

We observe that only small sized problem instances (up to 200 VMs) can be solved to optimality by every consolidation model. On the other hand, starting from 250 VMs the resolution process lasts longer than the imposed time limit for every MBP model with more than one time interval. For MBP models considering short time intervals of 5 minutes and 1 hour, it is not possible to find a feasible integer solution within the time limit starting from medium sized problems of

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

TABLE I: Resolvable scenarios

VMs Set Size	Consolidation Models				
	CBP 5min	MBP 1d	MBP 12h	MBP 1h	MBP 5min
150	S/S	S	S	S	S
200	S/S	S	S	S	S
250	S/S	S	L	L	L
300	S/S	S	L	L	L
400	S/S	L	L	L	N
500	S/S	L	L	L	N
600	S/S	L	L	N	N
700	S/S	L	L	N	N
800	L/S	L	L	N	N
900	L/S	L	L	N	N
1000	L/S	L	L	N	N
1100	L/S	L	N	N	N
1200	L/S	N	N	N	N

400 and 600 VMs, respectively; for larger time of 12 hours and 1 day, the size of resolvable problems grows to 1000 and 1100, respectively. On the other hand, the breakdown in building blocks allows the CBP model to find a feasible integer solution for every VMs set size, with the possibility to solve to optimality even scenarios up to 700 VMs. From these results, it is evident that the CBP technique allows us to solve significantly larger problems with respect to a MBP approach, even when the MBP problem considers few or just one time intervals.

Let us now evaluate the quality of the solutions for the different consolidation models. We start from the analysis of problems characterized by small-medium VMs set sizes (from 150 to 300 VMs), that are the only scenarios for which a feasible solution of the MBP-5min model can be found. The solution qualities are compared in Fig. 4, where we also show the results achieved by the FFD heuristic. We recall that the solution quality is measured as the number of physical nodes required for the allocation with respect to the objective function value of the LP relaxation of MBP (5-min). In other words, a solution quality of 110% means that the allocation required 10% physical nodes more with respect to the solution of the LP relaxation problem.

The first observation is that the FFD heuristic provides low-quality results for every considered problem size, requiring up to one-third more physical nodes with respect to the lower bound solution. Every consolidation model based on CBP and MBP outperforms the FFD heuristic. A second observation is that the quality solution of the MBP models increases with the number of considered time constraints; in particular, the MBP-5min always achieves the best solution in terms of number of required physical nodes. If we analyze the performance of the proposed CBP-5min model, we note an interesting trend: while for small scenarios (150 and 200 VMs) the quality of the CBP-5min solution is worse with respect to all the MBP models, the relation changes as the VMs set size increases: for 250 and 300 VMs, the CBP-5min outperforms the MBP-1d and MBP-12h models; for the 300 VMs, it achieves the same quality with respect to the MBP-1h model.

To better evaluate the behavior of the different consolidation models for small-medium scenarios, we analyze the

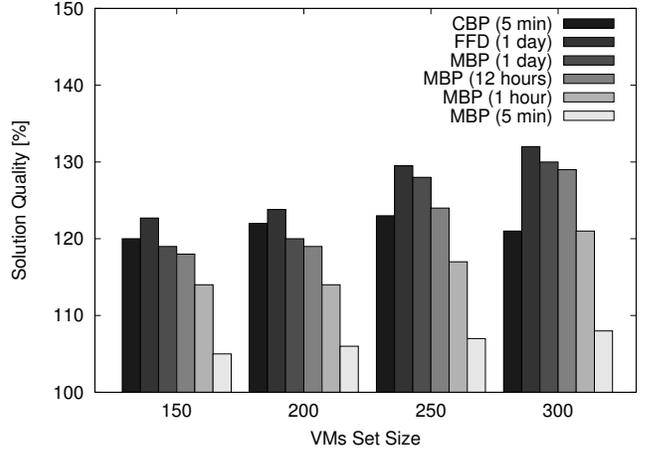


Fig. 4: Solutions quality for small VMs set size

resolution times for the considered problem instances, that are shown in Table II. From the table, we clearly see that the CBP-5min can always be resolved in a short time, while for the MBP models the resolution time rapidly increases with the VMs set size. For 250 and 300 VMs, the resolution of every MBP model with more than one time interval exceeds the time limit of 1800 seconds. These results show that for a small problem (under 200 VMs) the MBP model is the preferable choice, and can be solved to optimality even if it considers 5 minutes time intervals. On the other hand, as the number of VMs to allocate increases, the resolution time tends to rapidly increase: in these cases, there is a trade-off between resolution time and solution quality, and the preferable choice depends on the management requirements of the data center. A final consideration refers to the FFD heuristic: the poor quality of the achieved solution is due to the simple nature of this consolidation model that cannot take into account the time-related characteristics of the workload. However, due to its simplicity, the FFD heuristic can provide results in less than one-tenth of second for the considered problem sizes.

We now pass to analyze the case of medium-large scenarios (from 300 to 1200 VMs). Fig. 5 shows the solution qualities for the different consolidation models. This graph does not report the results for the MBP-5min model, because it cannot find any feasible solution for instances larger than 300 VMs.

The graph clearly shows how the increase in the number of VMs causes the impossibility to find a feasible integer solution within the time limit for the MBP models: we see the histogram bars referred to MBP models gradually disappearing as the number of VMs increases. For 1200 VMs, only the FFD heuristic and the proposed CBP model are able to find a feasible solution within 30 minutes. From the graph is also evident how MBP models with few time constraints allow the resolution of problem instances with larger numbers of VMs, but at the expense of solution quality.

If we now analyze the relation between the CBP-5min proposal and the MBP-1h model, that can be solved within the time limit only until scenarios of 500 VMs, we observe a trend reversal with respect to the previous results: starting from 400 VMs the CBP-5min outperforms the MBP-1h model, showing

TABLE II: Resolution times

Consolidation Models	VMs Set Size			
	150	200	250	300
CBP-5min (B/ε)	0.42/0.46	0.49/0.28	0.54/0.49	0.98/0.40
FFD-1d	0.05	0.05	0.06	0.07
MBP-1d	0.21	11.36	45.28	147.73
MBP-12h	4.13	79.39	1800 (L)	1800 (L)
MBP-1h	31.87	91.20	1800 (L)	1800 (L)
MBP-5min	233.09	270.59	1800 (L)	1800 (L)

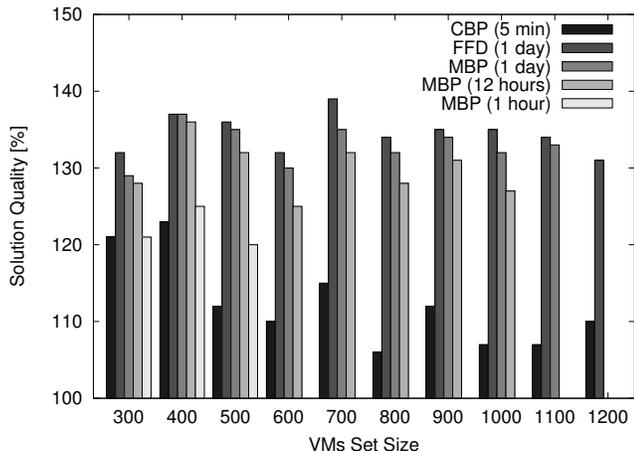


Fig. 5: Comparison of solutions quality

that for medium sized scenarios even an hourly aggregation of the resource usage data does not provide the consolidation problem with enough information to find a solution as efficient as our proposal. For scenarios with more than 500 VMs, the solutions achieved by MBP-12h and MBP-1d models are significantly worse with respect to the CBP-5min model, with a quality difference ranging from 15% to 26%.

We can conclude that the CBP-5min allows to find a feasible solution for every VMs set size, up to 1200 VMs. While for small scenarios the preferable choice remains a MBP-5min that could solve to optimality the global allocation problem, starting from medium-sized scenarios (400 VMs) the proposed solution outperforms other consolidation models both in terms of resolution time and number of required physical nodes.

V. 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 nodes 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 nodes required to allocate a given set of VMs in a cloud data center. To this purpose, large data centers can better leverage techniques such as selectively powering down idle servers or using hardware support for idle sleep states [7]. However, exploiting these techniques requires the resolution of the optimization problem described in Section III, to determine how to map VMs over

the physical nodes 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. To reduce the dimensionality of the problem, multiple solutions have been proposed in literature or applied in real systems.

The most straightforward solution is to simply 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 [4], [5], [6], but introduces the unreal assumption that every VM uses the 100% of its resources. Any under-utilized VM determines a waste of resources in the data center and increases the carbon footprint of the cloud infrastructure.

Another approach to tackle the computational issues of VM placement is to reduce the problem dimensionality by limiting the number of VM resources and time intervals that are considered in the bin-packing problem. 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 [8], [2]. In our experiments, we consider the CPU requirements over 5-minutes time intervals, and our approach can be easily extended to take into account additional resources.

Finally, the last approach to tackle the computational issues of the bin-packing problem is to exploit heuristics to reduce the computational cost of the solution. However, as pointed out in [9], most research is focused on problems with few dimensions, while if we consider the impact of multiple resources considered in multiple time intervals in a future planning period, the number of dimensions significantly grows. The most popular heuristics are applied to problems characterized by a number of dimensions ranging from one to three [23], [24]. As the dimensionality of the problem exceeds these values, the quality of the solution identified by the heuristics drifts away from the optimum. Our approach is completely different from these studies, because we reduce the number of VMs and nodes involved in the bin-packing problem to form a building block of limited size where we can easily apply complex optimization, without the need to reduce the number of dimensions and constraints.

VI. CONCLUSIONS

In this paper we tackled the critical problem of VMs placement in IaaS cloud computing data centers. Specifically,

we pointed out the scalability challenges of this task in large cloud infrastructures.

To cope with the scalability issues of current placement techniques, we propose an alternative approach where VMs are not considered as black boxes with independent resource requirements. Exploiting recent solutions that can cluster together VMs exhibiting similar behaviors in terms of resource usage, we propose a novel VMs placement technique, namely *Class-Based*, that solves a small-size placement problem and replicates it as a building block to obtain the global solution.

An extensive set of experiments demonstrates that our proposal outperforms existing solutions by reducing the number of physical nodes required to host the VMs in a cloud data center from 15% to 26% with respect to widely used alternatives. Furthermore, the performance gain of our proposal is more evident for large data centers, that represent the most challenging scenario for cloud computing.

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*, vol. 2007, pp. 1–16, 2012.
- [2] 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.
- [3] B. Speitkamp and M. Bichler, "A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers," *Services Computing, IEEE Transactions on*, vol. 3, no. 4, pp. 266–278, Oct 2010.
- [4] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman *et al.*, "Reservoir when one cloud is not enough," *IEEE computer*, vol. 44, no. 3, pp. 44–51, 2011.
- [5] K. Mills, J. Filliben, and C. Dabrowski, "Comparing vm-placement algorithms for on-demand clouds," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 91–98.
- [6] 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>
- [7] L. A. Barroso and U. Hözlze, "The case for energy-proportional computing," *IEEE computer*, vol. 40, no. 12, pp. 33–37, 2007.
- [8] 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.
- [9] 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.
- [10] M. Kao, *Encyclopedia of Algorithms*. Springer, 2008.
- [11] 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.
- [12] —, "Exploiting ensemble techniques for automatic virtual machine clustering in cloud systems," *Automated Software Engineering*, pp. 1–26, 2013. Available online.
- [13] M. Rabinovich and O. Spatscheck, *Web caching and replication*. Addison-Wesley Boston, USA, 2002.
- [14] C. Canali and R. Lancellotti, "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.
- [15] —, "An Adaptive Technique to Model Virtual Machine Behavior for Scalable Cloud Monitoring," in *Proc. of IEEE Symposium on Computers and Communications (ISCC)*, Madeira, Portugal, Jun. 2014.
- [16] A. K. Iyengar, M. S. Squillante, and L. Zhang, "Analysis and characterization of large-scale web server access patterns and performance," *World Wide Web*, vol. 2, no. 1-2, pp. 85–100, 1999.
- [17] S. Casolari and M. Colajanni, "On the selection of models for runtime prediction of system resources," in *Run-time Models for Self-managing Systems and Applications*. Springer, 2010, pp. 25–44.
- [18] 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. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128612003301>
- [19] R. Zhang, R. Routray, D. M. Ebers, D. Chambliss, P. Sarkar, D. Willcocks, and P. Pietzuch, "IO Tetris: Deep storage consolidation for the cloud via fine-grained workload analysis," in *Proceedings - 2011 IEEE 4th International Conference on Cloud Computing, CLOUD 2011*, 2011, pp. 700–707.
- [20] 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.
- [21] 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.
- [22] L. Zhang and D. Ardagna, "Sla based profit optimization in autonomic computing systems," in *Proceedings of the 2Nd International Conference on Service Oriented Computing*, ser. ICSOC '04. New York, NY, USA: ACM, 2004, pp. 173–182. [Online]. Available: <http://doi.acm.org/10.1145/1035167.1035193>
- [23] O. Faroe, D. Pisinger, and M. Zachariasen, "Guided local search for the three-dimensional bin-packing problem," *Informatics journal on computing*, vol. 15, no. 3, pp. 267–283, 2003.
- [24] 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.