Virtual Machine Profiling for Analyzing Resource Usage of Applications

Xuesong Peng, Barbara Pernici, and Monica Vitali

Politecnico di Milano, piazza Leonardo da Vinci 32, 20133 Milano, Italy

Abstract. From the cloud provider perspective, applications are usually black boxes hosted on Virtual Machines. Managing these black boxes without knowing anything about the features of the workload can generate inefficiencies in the performance. In fact, this information can be relevant to take deployment decisions which consist both in considering the interferences between applications with similar resources demands and predicting future peak demands avoiding performance degradation. Monitoring applications in cloud facilities and data centers is the only approach to manage and ensure the performance level of the hosted applications. This paper considers applications as black boxes and, using monitoring data analysis of the VMs on which applications are running, provides a methodology for building an application profile reflecting relevant behavioral features of a VM. This information is precious to lead deployment and adaptive decisions in data center management. The approach is validated on a real monitoring data set of an Italian data center.

Keywords: VM profile; Intensiveness; Periodicity; Applications resource usage; data center

1 Introduction

Cloud computing helps achieving high-performance levels in enterprise applications for a potentially lower cost than traditional ways. Application deployed using the Infrastructure as a Service (IaaS) paradigm can take advantage of the scalability and agility of the cloud to meet variable workload demands and to improve overall availability. Most of the applications running in a cloud infrastructure are black boxes for the cloud provider, which is not aware of what the application does and doesn't have any clue on the expected behavior and resource demand. The only knowledge on the application is obtained by the configuration and monitoring of the Virtual Machines (VMs) hosting it. Taking decisions on application deployment and migration is not easy in this blind situation. In fact, deployment should take into consideration the resources required by the application and possible interferences with other applications hosted on the same physical machine. The lack of knowledge about the application behavior usually results in a waste of computational power: servers are usually under-loaded to avoid performance issues in case of a peak load. Currently, the cloud service model has been flanked by the edge paradigm, in which applications are executed as near as possible to the final customer or to where the data that they have to analyze are collected.

This is possible as far as enough computational resources are available in such limited devices. In this context, knowing the expected behavior of an application is even more crucial than in the cloud scenario. In fact, a wrong decision can affect the experienced quality of service.

In this paper, we propose a methodology for building an application profile from the data collected by the monitoring system during the application execution on a VM. The aim of the profile is to capture the dynamic behavior and resources intensity of an application hosted on a monitored VM. Having this information can be helpful to attain several objectives: supporting deployment decisions, detecting anomalies, and classifying homogeneous VMs in terms of resource usage and patterns of usage in time. The profile proposed in this paper takes into consideration two main aspects: (i) *intensiveness* in resource usage of the VM and (ii) *periodicity* of the VM behavior. We base the analysis on monitoring data of typical indicators for data centers, extracting VMs profiles and developing techniques to analyze them.

The paper is structured as follows. In Sect. 2, we discuss related work on monitoring and profiling applications and virtual machines in data centers and on analyzing periodic behaviors in general. In Sect. 3, we outline the method for analyzing VM profiles, then detailing its steps in Sect. 4 and Sect. 5. Finally, in Sect. 6, we analyze in detail the characteristics of VMs in a real data center, exploiting the proposed VM profiles.

2 Related Work

The management of data centers is a complex task that is getting more and more challenging with the increase of the heterogeneity of both the infrastructure employed in most of the data centers (old and new generation servers) and of the applications hosted by the infrastructure. In order to provide a reliable management, the monitoring system plays a key role since it collects all the information needed to detect issues and to ensure the required performance levels. However, the big amount of data collected and its complexity can make this fundamental task difficult. In [1] the authors analyze the issue of cloud monitoring, focusing both on its challenges and its properties, stating the important role that monitoring systems have nowadays in such an environment. In [4], different cloud monitoring solutions are compared. Monitoring is often associated with data analytics since it enables the extraction of relevant information from the collected data. As described in [22], this task can be expensive and time-consuming and a trade-off between benefits and costs has to be analyzed. The issue of managing the big amount of monitoring data in a scalable way is faced in [7], where the authors propose an innovative data warehousing system for performance-related information. Monitoring information can be also addressed to improve the energy efficiency and the sustainability of clouds and data centers [17]. In projects like the European FP7 project ECO_2Clouds[1], the issue of sustainability has been managed with an adaptive approach basing decisions about application deployment on the data retrieved with the monitoring system [21, 3]. Finally, in [5] the authors propose a method to optimize application deployment based on the maximization of the quality and completeness of information gathered by the monitoring system in a multi-cloud environment.

[1] http://eco2clouds.eu

Many researches exploited the monitoring data to model the application, the VMs, and the physical machine behavior profile. In those profiles, multiple monitoring dimensions are involved, including mostly CPU, memory usage, IO, and network usage, and sometimes also power and temperature are considered for energy efficiency and thermal awareness[23, 9]. Various metrics are carried out to measure specific properties. For instance, [16] measures average disk writes per second, [10] measures the percentage of CPU time occupied by the process of user state. Some work [14, 8] use the profiles directly to estimate resources demands and support decision making, but other work try to go a step further. In [12], authors analyze sensitivity of VMs to cloud resources (CPU, Memory, and Storage) in order to profile the application behavior under intensive workloads. In [11], authors map the VM profile to an execution state space and use this to interpret state transitions of co-located VMs. A penalty-based profile matching algorithm (PPMA) is developed in [15] to obtain an assignment solution, which gives near-optimal allocations whilst satisfying energy-efficiency, resource utilization efficiency and application completion time constraints. The methodology is based on the strong assumption of a stable workload. Most research takes the profile as an instant screen shot of the system, rather than an aspect of self-repeatable and stable normality of the system. The authors of [24] exploit recurring patterns to estimate future resource consumption for VM consolidation. But their method assumes the behaviors of all VMs in all dimensions are 100% periodic, which is generally not true. Another focus of the VM behavior profile is to analyze the performance bottlenecks. For instance, the memory used in system buffers can help measuring memory intensiveness [2], the number of interrupts per second can help measuring CPU intensiveness [10], throughput can be used to evaluate the performance of applications [13]. These approaches analyze only simple benchmark applications under pre-defined workloads.

In this paper we claim that periodicity of VMs is a relevant information for data center management. The two most common used functions to measure periodicity of a signal are Periodogram and ACF (auto-correlation function). Periodogram function analyzes the signal in frequency domain using the Discrete Fourier Transforms (DFT). However, considering the increasing size of DFT, the resolution of Periodogram becomes very coarse for longer periods. Due to this reason, detecting large periods with Periodogram can be very inaccurate, sometimes false alarms are raised because of the absences of power in the DFT bin. ACF examines how similar a sequence x is to its shifted (lagged) copies for different t lags, calculating the auto-correlation with the sequence itself. In the auto-correlation graphs, multiples of the same basic period also appear as peaks. Therefore, the method introduces many false alarms that need to be eliminated in a post-processing phase. As both functions have some deficiencies in detecting periodic behaviors of VM indicators in isolation, in the following of the paper we base our work on a fusion method of the two above-mentioned functions, which has been proposed by [18].

3 Profiling Methodology

In this section, we illustrate the proposed method for deriving and analyzing VM profiles in data centers. The methodology is illustrated in Figure 1, which describes the

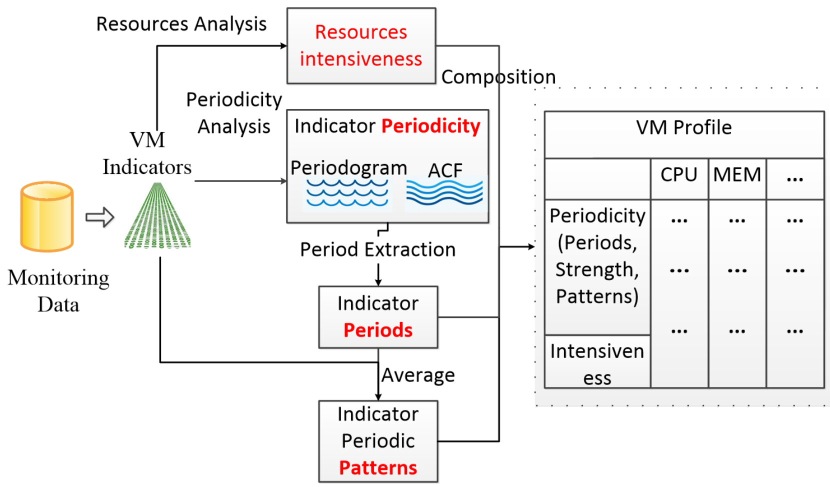


Fig. 1. Method for profiling VMs

process through which the monitoring data are transformed into VM profiles for single VMs and how the profiles are used to detect similarities between VMs. The figure shows the main data items and the connecting edges are labeled with the transformation steps. We assume to have a monitoring system able to collect information on the VMs behavior and store them in a monitoring database. To build the profile, we extract the raw data from this database for each of the VMs that we want to analyze.

Two parallel analyses are performed to build the profile: (i) *Resource intensiveness analysis* - the data collected through the monitoring system are analyzed in order to assess the intensiveness of the VM according to the resource consumption; (ii) *Indicator Periodicity* - collected data are used to assess if a periodic behavior in the resource usage of the VM can be detected. Combining together the characteristics analyzed above, we build a *VM profile* which describes resources characteristics *Intensiveness* and *Periodicity* of the VMs, for all relevant indicators. In the following, we illustrate the two steps for building the VM profile, analyzing the intensity of resources in Sect. 4, then defining periodicity in Sect. 5.

We study VMs through the performance indicators recorded by the monitoring system, and in particular, we focus on four indicators: (i) CPU, (ii) MEM, (iii) BW, and (iv) IO. These indicators are available in most monitoring systems and are generally used to analyze VMs in data centers [21]. We illustrate two examples of profile for two VMs, denoted as $A12$ and $C3$, in Fig. 2, which shows intensiveness and periodicity (with associated periods and strength) for the considered indicators. The periodicity contains also the patterns for the periodic indicators, as shown in Fig. 3 for the CPU weekly pattern of VM $C3$.

VM name		CPU	MEM	BW	IO
A12	PERIODS/STRENGTH(ACF)	1 day/0.66	1 day/0.31	-	0.3day/0.19
		-	-	-	7 days/0.28
	INTENSIVENESS	medium-intensive	non-intensive	non-intensive	medium-intensive
C3	PERIODS/STRENGTH(ACF)	1 day/0.64	1 day/0.15	1 day/0.73	1 day/0.73
		7 days/0.63	3.5 days/0.2	-	-
		-	7 days/0.7	-	-
	INTENSIVENESS	medium-intensive	non-intensive	medium-intensive	non-intensive

Fig. 2. Example profiles of two VMs

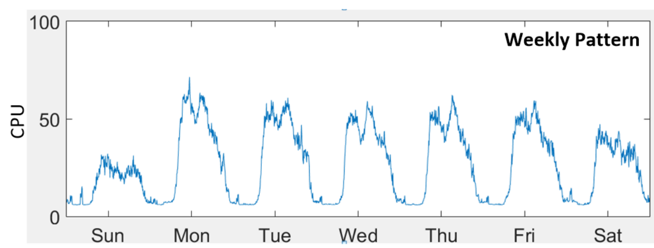


Fig. 3. CPU weekly pattern of VM C3

4 Profiling Intensity of Application Usage of Resources

In this section we discuss the methodology used to detect the intensiveness of applications according to their resource usage. The VM profile includes resources usage characteristics of VMs, in order to be able to identify the specific resources that can limit VM performances. The aim is to find relevant resources for the applications running on the VM, enabling us to prevent shortages of these resources and improve performance. Existing methods for detecting intensiveness are based on the knowledge of the applications running on the VMs. As an example, in [12] the authors provide a methodology to extract the profile systematically. The authors stress the different system components and, as a result, associate tags to the applications: CPU-intensive, MEM-intensive, disk-intensive, and so on.

If information about applications is not available, which is the case for our scenario, we can only mine the historical resource usage from monitoring data on VMs to derive the resource intensiveness. For each resource of a specific VM we propose three candidate metrics as follows:

– avg: the average resource usage of all the samples in the dataset;
– $p_{warning}$: the percentage of samples of a specific resource which exceed a given threshold $th_{warning}$;

– $p_{critical}$: the percentage of samples of a specific resource which exceed a given threshold $th_{critical}$.

The avg metric takes all samples into consideration while the $p_{warning}$ and $p_{critical}$ only consider stressful situations of the resource. Therefore, $p_{warning}$ and $p_{critical}$ focus on the crucial moments when a resource becomes a bottleneck for the VM performance. This is important for CPU and memory since the shortage of these resources might cause inefficiency and QoS failures in a very short time. To get appropriate thresholds for $th_{warning}$ and $th_{critical}$, we refer to the literature of data center management practice. The Data Center Maturity Model (DCMM) [6] is a reference to evaluate the maturity of individual data centers. It suggests best practice according to the level of maturity that the data center aims at achieving. The highest maturity level (called "Visionary"), which should be reached in the next five years, requires the average monthly CPU utilization above 60%. Another reference for setting thresholds for the VM CPU and memory usage is the VMware Knowledge Base [19, 20]. According to this source, the CPU is considered in a warning condition if its usage is above 75% and in an alarm condition if its usage is above 90% for 5 minutes. Similarly, the memory warning condition sets a threshold of 85%, while an alarm condition is detected if the resource usage is above 95% for 10 minutes.

According to intensiveness, we intend to classify the VMs into three groups (*intensive*, *medium-intensive*, and *non-intensive*), and also to define the conditions to place a VM in these groups. Since thresholds can change according to the specific scenario, we can choose thresholds for each resource starting from the thresholds values mentioned above. For the network bandwidth BW and IO, unlike for CPU and MEM, the immediacy of handling stress peaks is usually not emphasized in the literature, if only the VM can finish its work smoothly. Therefore, the intensiveness of BW and IO are mainly relying on the total resource demands, but not only on the stressful demands peaks. We will therefore base the intensiveness analysis on the average consumptions of VMs, grouping them in three groups with homogeneous characteristics, to define intensive, medium-intensive, and non-intensive characteristics. A detailed example of intensiveness analysis in a real data center can be found in the case study discussed in Sect. 6.1.

5 Profiling Application Periodicity

5.1 Identifying periods

The goal of this step is to identify periods for the indicators. In real data centers, noises makes detecting the periodic behaviors of VMs a hard job. We exploit the mechanism described in [18] to filter insignificant periods of the VMs indicators, ensuring that we are analyzing VMs excluding noises. As discussed in Sect. 2, both ACF and Periodogram functions can examine periodic signals, but the detected period accuracy and the false positive rate could be a problem if we adopt these techniques separately, so, following their proposal, this work exploits both functions sequentially to identify the true and precise periods. Fig. 4 illustrates the methodology which is proposed by [18]: first, the Periodogram is used to extract period candidates, and then ACF is applied

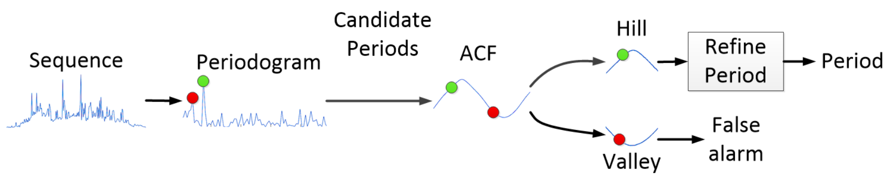


Fig. 4. Period Detection methodology [18]

to validate those candidates. More specifically, if the candidate period from the Periodogram lies on a hill of ACF, it can be inferred that a peak of ACF is nearby which confirms the validity of the period (*valid period*), otherwise, if the Periodogram falls in a valley of ACF it is discarded as a false alarm. Finally, the ACF peak nearer to the Periodogram position is taken as the true period, refining the candidate period.

In this step, we analyze the data gathered by the monitoring system for each VM in order to build the periodicity part of its profile. For each of the selected indicators, the periodicity is detected using the described approach. Fig. 5 gives an example. The original data shows CPU usage of a VM in almost 100 days. We can observe obvious daily peaks and a weak weekly pattern. Thus we expect the periodicity to be 1 day and 7 days. In periodicity analysis, we focus on peaks of Periodogram which indicate their periodic strength, and select k candidate frequencies by filtering with their power and distances between them. For simplicity of the graph, we take $k = 3$ in this example. Then we map candidate frequencies to periods (namely, 0.3 days, 1 day and 7.2 days), and by verification of ACF, we take 1 day and 7.2 days as valid periods, but discard the 0.3 days as a false alarm because it is a valley in ACF diagram, according to the methodology of [18]. Following the procedure described above, we refine the 7.2 day period to the corresponding hill peak of ACF, namely, 7 days.

It is useful to analyze and compare behaviors characteristics of different indicators. Referring to Fig. 2, we see that for VM A12 most of the indicators are periodic, except for bandwidth. However, the periods are different: CPU and memory have daily periods, while IO has a weekly period.

5.2 Establishing periodic patterns

Detecting periodicity of a VM is not enough to represent its behavior. For having a deeper knowledge it is also important to represent how the VM behaves in this periods according to resources demands. Each valid period, identified in the previous step, corresponds to a periodic behavior of a VM indicator, namely, a *pattern*, which repeats itself through the considered time interval for the analyzed VM. For instance, as a result of human activities characteristics, most applications on the cloud have daily, weekly, or yearly workloads patterns (although other periods can be significant in some cases), so that we can find these patterns on most VM indicators. The pattern is essential to understand VM behavior and it provides the possibility to predict and optimize VM resources demands in data centers.

Once the relevant periods for each indicator of a VM have been detected, we can move to the analysis of the periodic behavior by considering the raw data. The goal

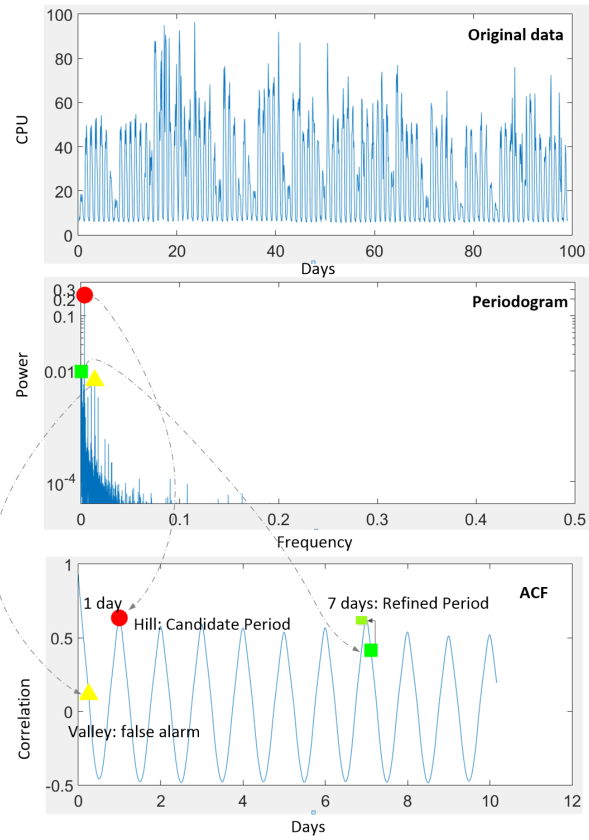


Fig. 5. Detecting periods of CPU for VM C3

is to build a typical shape for the considered signal. In order to do so, the continuous indicators are first cut into several segments, where each segment is as long as the refined period, representing an instance of the repeating behavior. Then for every time stamp in this period, we take the average of all instances to construct a pattern describing usual behavior in this period. Taking the VM in Fig. 5 as an example, we build the weekly pattern and observe clear differences between workdays and weekends, as Fig. 3 depicts.

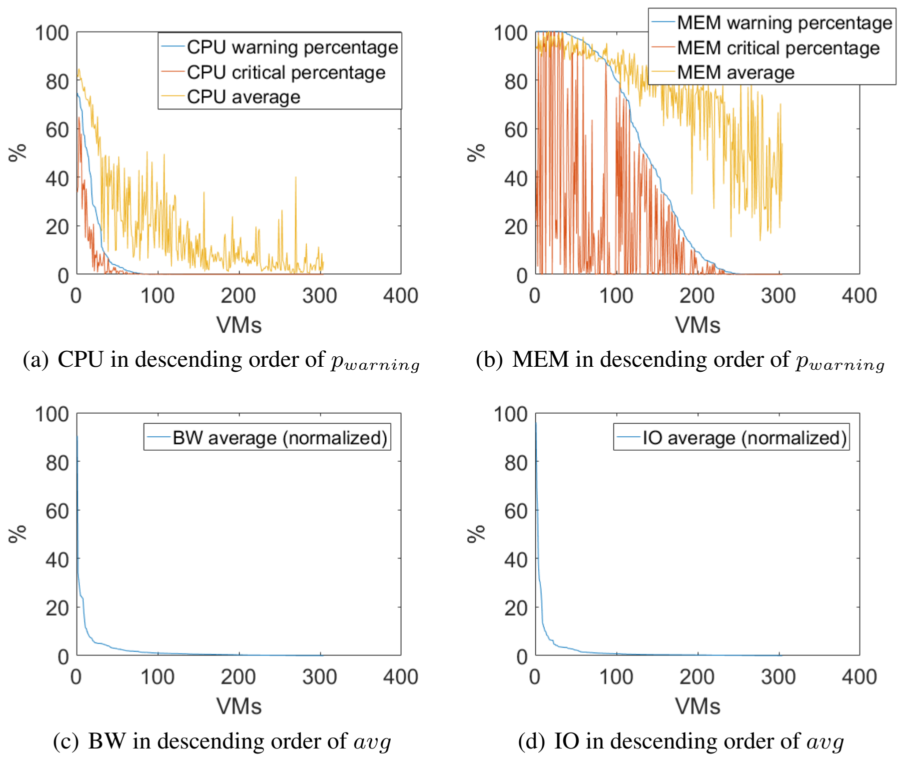
6 Profiling of Virtual Machines in a Real Data Center

In the previous sections, we illustrated the method to derive application profiles, and in this section, we derive and analyze the profiles to understand behaviors of VMs in a data center using a private cloud. The considered dataset, from a real data center of a telecommunication company in Italy, consists of the monitoring data of 304 VMs, and all indicators were sampled simultaneously with a sampling interval of 5 minutes. The indicators include the usage of multiple resources of VMs and hosts, covering CPU, memory, bandwidth and I/O, and so on. On the other hand, the dataset contains no information about the applications so that we do not know what kind of applications are running on the VMs and what is the performance of the applications. The data used in this paper cover the period of 3 months. All the steps of the profiling method discussed in Sect. 4 and Sect. 5 are validated with this dataset.

6.1 Intensiveness validation

Using the available historical monitoring data of the real data center, we have analyzed the intensiveness of resource usage for each VM considering the four main resources: CPU, memory, bandwidth, and IO. The results of the analysis are shown in Fig. 6 in which on the x-axis we represented each VM in the test bed (with the VMs arranged in a descending order) and on the y-axis we draw the normalized value of the metrics of the VM. As already discussed in Sect. 4, the average value is the only information used for evaluating the intensiveness of the VMs for the bandwidth and IO metric, while the $p_{critical}$ and $p_{warning}$ metrics are used for CPU and memory. To analyze our dataset, we applied the thresholds for CPU and memory intensiveness indicated by VMware [19, 20] as a starting point. Thresholds are set as follows: (i) $th_{warning}$(CPU) = 75%; (ii) $th_{critical}$(CPU) = 90%; (iii) $th_{warning}$ (MEM) = 85%; (iv) $th_{critical}$(MEM) = 95%.

Looking at the results shown in Fig. 6, we classified the VMs into three groups, namely, intensive VMs, medium-intensive VMs, and non-intensive VMs. According to the values of the four metrics, we identify the conditions of groups for each of them. Considering the CPU usage, only the VMs for which $p_{warning}(CPU) > 10\%$ are classified as CPU-intensive, because the average CPU usage of this group of VMs is significantly higher (e.g., the $avg(CPU)$ is mostly larger than 50% for this group). VMs are considered non-intensive in relation to CPU if they never exceed the warning threshold ($p_{warning}(CPU) = 0\%$). This group contains the most of the VMs analyzed and they are characterized by a low average CPU consumption (most of these VMs have $avg(CPU) < 30\%$). Finally, other VMs with $0\% < p_{warning}(CPU) \leq 10\%$ are classified as medium-intensive in CPU usage. Thus, we have defined the conditions of CPU intensiveness over the metric $p_{warning}$. As an alternative to exploiting only $p_{warning}$ to classify the groups, we also considered to use the values for $p_{critical}$ and avg to build alternative conditions (e.g., a VM is intensive if $p_{critical}(CPU) > 10\%$ and $avg(CPU) > 60\%$). We choose $p_{warning}$ because it is a mixture of the other two metrics, and its behavior is more suitable to classify VMs into the 3 intensiveness groups. Similarly, observing Fig. 6(b) we selected as threshold for memory intensiveness $p_{warning}(MEM) > 90\%$. As a result of this threshold, we classified 75 VMs as memory intensive, since they exceed this threshold for most of their lifetime. We also classified as non-intensive memory group the VMs with $p_{warning}(MEM) \leq 10\%$, since their average consumption of MEM is generally below 80% and they seldom exceed the warning threshold. As a result, medium-intensive VMs for memory are the VMs with $10\% < p_{warning}(MEM) \leq 90\%$. As discussed in Sect. 4, we use only avg to evaluate intensiveness for IO and bandwidth. As shown in Fig. 6(c) and Fig. 6(d), most VMs use the network and IO rarely, only a small number of VMs use the network



(a) CPU in descending order of $p_{warning}$ (b) MEM in descending order of $p_{warning}$

(c) BW in descending order of avg (d) IO in descending order of avg

Fig. 6. Intensive metrics for VM indicators

Table 1. Conditions of resource intensiveness groups

	CPU	MEM	BW	IO
Intensive	$p_{warning} > 10\%$	$p_{warning} > 90\%$	$avg > 2\%$	$avg > 2\%$
Medium-intensive	$0\% < p_{warning} \leq 10\%$	$10\% < p_{warning} \leq 90\%$	$0.6\% < avg \leq 2\%$	$0.4\% < avg \leq 2\%$
Non-intensive	$p_{warning} = 0\%$	$p_{warning} \leq 10\%$	$avg \leq 0.6\%$	$avg \leq 0.4\%$

or IO occasionally, and some of them use the network or IO extremely frequently. Based on the shape of avg we define the thresholds for CPU, MEM, IO, and BW as in Tab. 1.

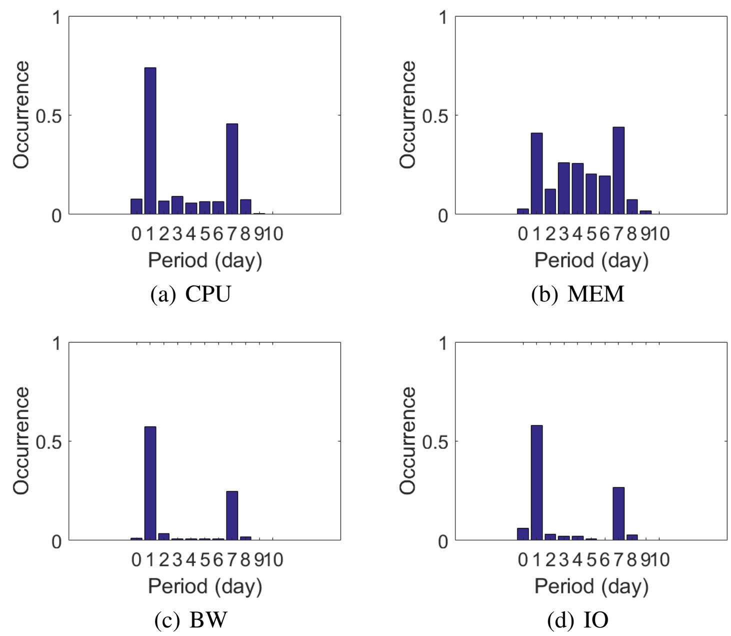
Using the discovered thresholds, it is possible to detect the intensiveness of each VM in the dataset. As an example, for the two VMs considered in Fig. 2, we evaluated the intensiveness metrics and derived intensiveness as shown in Tab. 2

6.2 Analysis of application periodicity

The second step for building the profile of a VM consists in evaluating the VM periodicity. We computed the periodicity profiles of VMs in the data center as illustrated

Table 2. Intensiveness of two VMs

VM id	**A12**				**C3**			
Metric	**Intens.**	**p_warning**	**p_critical**	**avg**	**Intens.**	**p_warning**	**p_critical**	**avg**
CPU	not-int.	0.04%	0.03%	28.77%	medium	4.07%	1.58%	26.38%
MEM	non-int.	4.40%	1.83%	74.05%	non-int.	9.28%	0.00%	78.71%
BW	non-int.	–	–	0.48%	medium	–	–	0.96%
IO	medium	–	–	0.83%	non-int.	–	–	0.38%



(a) CPU (b) MEM

(c) BW (d) IO

Fig. 7. Period occurrences (normalized) over four indicators

in Sect. 4. As a result, we detected that: (i) 275 VMs have some CPU-periodic behavior; (ii) 278 VMs have some memory-periodic behavior; (iii) 223 VMs have some BW-periodic behavior; (iv) 227 VMs have some IO-periodic behavior. Using the periodicity profiles, we can also extract a general overview of the VMs periodicity of the data center, summarizing the occurrence of periods for each indicator and showing the distribution of periods for each indicator as shown in Fig. 7. For an easier comparison of the different cases, the number of occurrences is given in terms of percentages in the figure.

The two most common periods for all indicators are the daily period and weekly period, this conforms to the reality as most application workloads are daily or weekly periodic. It can also be noticed that periodicity varies for different indicators. From the IO perspective, more than half of VMs have a daily period, and more than 25% of VMs are weekly periodic. This characteristic may relate to the daily/weekly dumps of some

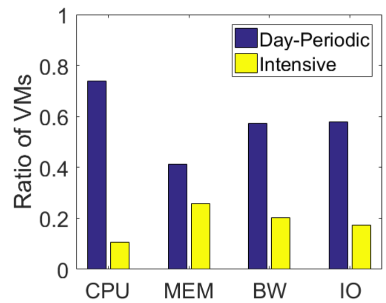


Fig. 8. Resources usage characteristics of the data center

applications (in this case we see peaks of high values for BW/IO weekly, for a short time, and mainly in the weekend and at night. The BW of the data center is very similar to the IO behavior: This similarity can be explained because most IO activities generate or are generated by network transmissions. The CPU periodicity is also strong, with almost 70% of the VMs with a daily and 50% with a weekly periods. However, the MEM behaves differently: the daily/weekly periods are no longer stand-out compared to others. The possible reason might be that the memory demands of most applications are generally very static and the periodicities are not significant.

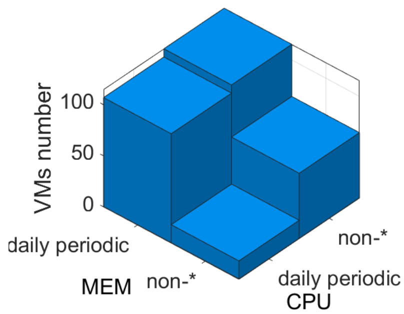
6.3 Relating VMs behavior to VM profiles

We analyze the behavior of the profiles of the 304 VMs of the analyzed data center, and we compare periodicity characteristics with the intensity of resources. We have also analyzed the correlation between the resource intensiveness and the VMs migrations observed in the data center[2].

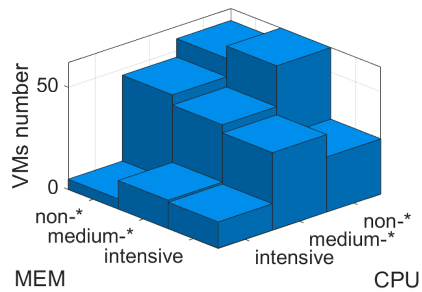
Periodicity and resource usage relation In this section, we analyze the characteristics of resources consumption on both periodicity and intensiveness, and try to find if there is a relation between them. We calculate the percentage of VMs that are daily-periodic and the percentage of VMs that are intensive for certain resources, as Fig. 8 depicts. The percentage of VMs having a daily periodicity is very high in this data center, especially for CPU, where more than 70% of VMs are daily periodic. The overall resource intensiveness in the data center is limited (below 30% for all the resources). The most intensive resource is memory. As can be observed, there isn't a strong relation between intensiveness and periodicity for a given resource. A possible reason could be that the periodic behaviors are easier to predict thus the administrator can prevent the intensive situations better compared to the non-periodic behaviors.

We have also analyzed the combined characteristics of multiple resources, for both periodicity and intensiveness. For the sake of brevity, we focused on CPU and memory. Fig. 9 depicts the relation between CPU and memory periodicity and intensiveness. As can be observed in Fig. 9(b), most VMs are either non-intensive to CPU or non-intensive to MEM, only 13 VMs are intensive for both CPU and MEM. This may help

[2] Information on migrations is available in the monitoring data of the data center.

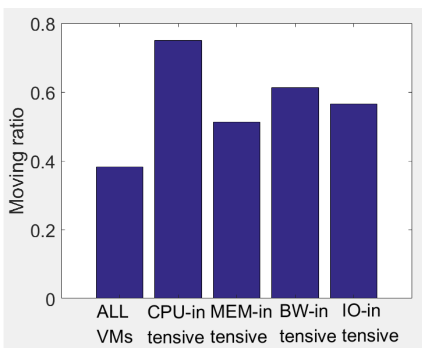


(a) periodicity



(b) intensiveness

Fig. 9. Periodicity and intensiveness situations over CPU and MEM



(a) VMs movements for intensive groups

	CPU	MEM	BW	IO
Intensive group	0.75	0.51	0.61	0.57
Medium-intensive group	0.36	0.50	0.33	0.48
Non-intensive group	0.32	0.16	0.32	0.27
ALL VMs	0.38			

(b) VMs movements for different intensiveness groups

Fig. 10. Impact of resource intensiveness on VM migration

us to understand the critical situations in the data center, and the administrator needs to be careful with those 13 VMs.

Impact of resource intensiveness on VM migration In data centers, moving VMs between different physical servers is common practice, which can improve the total efficiency of resources utilization and decrease power consumption. Usually, a VM is moved to a new host in two situations: (i) the server hosting the VM is overused; (ii) the server hosting the VM is underused.

In order to validate our intensiveness metric, we extracted the events of VM movements from the dataset, and analyze the relations between the VM migrations and its resource intensiveness. Fig. 10(a) depicts the percentage of VMs migrated grouped for intensive resource. As can be seen in the first column, most VMs (more than 60%) have never been migrated during the analyzed period of 3 months. However, the intensive VMs have a higher probability to be migrated, regardless of which is the intensive resource. The CPU intensiveness is the most important factor for VM movements: 75% of CPU-intensive VMs have been migrated at least once in this period. A more detailed analysis is summarized in Fig. 10(b).

7 Conclusions and Future Work

This paper introduced a systematic method to build profiles of VMs in data centers, focusing on their resource intensiveness characteristics and on their periodic behavior. In the paper, we have defined a VM profile as composed of two main part: (i) the resource intensiveness in which for each resource (CPU, memory, bandwidth, IO) the level of intensiveness has been evaluated, and (ii) the resource periodicity in which the periodic behavior of each resource is analyzed and described. Applying the methodology for building the profile of VMs to the monitoring data of a real data center, we have demonstrated that the intensity of resource usage and periodicity are important features in the identification of a VM profile. We have also discussed the relationships between intensity and periodicity and the relationship between intensity and VM migration decision in the studied date set, retrieving that intensive use of resources, especially CPU, can be a driver for migration.

In future work we envision to exploit the VM profile for further analysis tasks, such as resource planning, enabling better VM placements and migrations, anomaly detection, and to support the analysis of the periodicity of bursts.

Acknowledgments

The authors thank the company Eco4Cloud[3] for sharing monitoring data for the purpose of this analysis.

References

1. G. Aceto, A. Botta, W. De Donato, and A. Pescapè. Cloud monitoring: A survey. *Computer Networks*, 57(9):2093–2115, 2013.
2. M. Awasthi, T. Suri, Z. Guz, A. Shayesteh, M. Ghosh, and V. Balakrishnan. System-level characterization of datacenter applications. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering, Austin, TX, USA, January 31 - February 4, 2015*, pages 27–38, 2015.
3. C. Cappiello, T. T. N. Ho, B. Pernici, P. Plebani, and M. Vitali. CO_2-Aware Adaptation Strategies for Cloud Applications. *IEEE Trans. Cloud Computing*, 4(2):152–165, 2016.
4. G. Da Cunha Rodrigues, R. N. Calheiros, V. T. Guimaraes, G. L. d. Santos, M. B. de Carvalho, L. Z. Granville, L. M. R. Tarouco, and R. Buyya. Monitoring of cloud computing environments: concepts, solutions, trends, and future directions. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 378–383. ACM, 2016.
5. E. Fadda, P. Plebani, and M. Vitali. Optimizing Monitorability of Multi-cloud Applications. In *Advanced Information Systems Engineering - 28th International Conference, CAiSE 2016, Ljubljana, Slovenia, June 13-17, 2016. Proceedings*, pages 411–426, 2016.

[3] http://www.eco4cloud.com

6. J. Huusko, H. de Meer, S. Klingert, and A. Somov, editors. *Energy Efficient Data Centers - First International Workshop, E2DC 2012, Madrid, Spain, Mai 8, 2012, Revised Selected Papers*, volume 7396 of *Lecture Notes in Computer Science*. Springer, 2012.
7. C. Loboz, S. Smyl, and S. Nath. Datagarage: Warehousing massive performance data on commodity servers. *Proceedings of the VLDB Endowment*, 3(1-2):1447–1458, 2010.
8. I. S. Moreno, R. Yang, J. Xu, and T. Wo. Improved energy-efficiency in cloud datacenters with interference-aware virtual machine placement. In *11th International Symposium on Autonomous Decentralized Systems, ISADS 2013, Mexico City, Mexico, 6-8 March 2013*, pages 1–8, 2013.
9. R. Nasim, J. Taheri, and A. J. Kassler. Optimizing virtual machine consolidation in virtualized datacenters using resource sensitivity. In *2016 IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2016, Luxembourg, December 12-15, 2016*, pages 168–175, 2016.
10. J. Peng, J. Chen, X. Zhi, M. Qiu, and X. Xie. Research on application classification method in cloud computing environment. *The Journal of Supercomputing*, 73(8):3488–3507, 2017.
11. N. Rameshan, L. Navarro, E. Monte, and V. Vlassov. *Stay-Away*, protecting sensitive applications from performance interference. In *Proceedings of the 15th International Middleware Conference, Bordeaux, France, December 8-12, 2014*, pages 301–312, 2014.
12. J. Taheri, A. Y. Zomaya, and A. Kassler. vmBBThrPred: A black-box throughput predictor for virtual machines in cloud environments. In *Service-Oriented and Cloud Computing - ESOCC 2016, Vienna, Austria, September 5-7, 2016, Proceedings*, pages 18–33, 2016.
13. J. Taheri, A. Y. Zomaya, and A. Kassler. vmBBProfiler: a black-box profiling approach to quantify sensitivity of virtual machines to shared cloud resources. *Computing*, 99(12):1149–1177, 2017.
14. M. Vasudevan, Y. Tian, M. Tang, and E. Kozan. Profile-based application assignment for greener and more energy-efficient data centers. *Future Generation Comp. Syst.*, 67:94–108, 2017.
15. M. Vasudevan, Y.-C. Tian, M. Tang, and E. Kozan. Profile-based application assignment for greener and more energy-efficient data centers. *Future generation computer systems*, 67:94–108, 2017.
16. S. Verboven, K. Vanmechelen, and J. Broeckhove. Black box scheduling for resource intensive virtual machine workloads with interference models. *Future Generation Comp. Syst.*, 29(8):1871–1884, 2013.
17. M. Vitali and B. Pernici. A survey on energy efficiency in information systems. *Int. J. Cooperative Inf. Syst.*, 23(3), 2014.
18. M. Vlachos, S. Y. Philip, and V. Castelli. On periodicity detection and structural periodic similarity. In *SDM*, volume 5, pages 449–460. SIAM, 2005.
19. VMware Knowledge Base. Virtual machine CPU usage alarm, 2015.
20. VMware Knowledge Base. Virtual machine memory usage alarm, 2015.
21. U. Wajid, C. Cappiello, P. Plebani, B. Pernici, N. Mehandjiev, M. Vitali, M. Gienger, K. Kavoussanakis, D. Margery, D. García-Pérez, and P. Sampaio. On achieving energy efficiency and reducing CO_2 footprint in cloud computing. *IEEE Trans. Cloud Computing*, 4(2):138–151, 2016.
22. C. Wang, K. Schwan, V. Talwar, G. Eisenhauer, L. Hu, and M. Wolf. A flexible architecture integrating monitoring and analytics for managing large-scale data centers. In *Proceedings of the 8th ACM International Conference on Autonomic Computing*, ICAC '11, pages 141–150, 2011.
23. L. Wang, S. U. Khan, and J. Dayal. Thermal aware workload placement with task-temperature profiles in a data center. *The Journal of Supercomputing*, 61(3):780–803, 2012.
24. A. Wolke. *Energy efficient capacity management in virtualized data centers*. PhD thesis, Technical University Munich, 2015.