An Adaptive Technique to Model Virtual Machine Behavior for Scalable Cloud Monitoring

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

***Abstract*—Supporting the emerging digital society is creating new challenges for cloud computing infrastructures, exacerbating scalability issues regarding the processes of resource monitoring and management in large cloud data centers. Recent research studies show that automatically clustering similar virtual machines running the same software component may improve the scalability of the monitoring process in IaaS cloud systems. However, to avoid misclassifications, the clustering process must take into account long time series (up to weeks) of resource measurements, thus resulting in a mechanism that is slow and not suitable for a cloud computing model where virtual machines may be frequently added or removed in the data center. In this paper, we propose a novel methodology that dynamically adapts the length of the time series necessary to correctly cluster each VM depending on its behavior. This approach supports a clustering process that does not have to wait a long time before making decisions about the VM behavior. The proposed methodology exploits elements of fuzzy logic for the dynamic determination of time series length. To evaluate the viability of our solution, we apply the methodology to a case study considering different algorithms for VMs clustering. Our results confirm that after just 1 day of monitoring we can cluster without misclassifications up to 80% of the VMs, while for the remaining 20% of the VMs longer observations are needed.**

I. Introduction

Cloud computing is a fundamental enabling technology to allow users to access complex services, vast processing power and amount of data from heterogeneous and possibly mobile devices. The capability of cloud computing infrastructures to cope with the increasing resource demand in the next few years will be critical for the future development of the emerging digital society.

As cloud systems grow in size and complexity to accommodate an increasing number of virtual machines (VMs), the scalability issues related to the process of monitoring VM resource usage for management strategies become a major challenge. Resource monitoring is particularly challenging in Infrastructure as a Service (IaaS) cloud systems, where several customer applications are hosted in virtualized environments. A customer application typically consists of multiple software components (e.g., the tiers of a multi-tier Web application), and each component runs on a separate VM. In these cloud systems, VMs are usually considered as black boxes with independent behaviors, hence information needs to be collected about each single VM of the data center, thus exacerbating the scalability issues of the monitoring task.

Recent research studies [1], [2] show that automatically clustering VMs with similar behaviors in terms of resource usage may improve the scalability of the monitoring process in IaaS cloud systems. The identification of classes of VMs behaving in the same way allows to reduce the amount of globally collected data, limiting a fine-grained monitoring to few representatives for each class, and performing a coarse-grained monitoring on the other VMs. However, the solutions presented in [1], [2] show a clear trade-off between the VM clustering accuracy (that is the fraction VMs assigned to the correct class) and the length of the resource usage time series used for clustering. Specifically, long time series (up to weeks of collected measurements) should achieve a clustering that correctly groups every VMs. The resulting mechanism is poorly reactive to changes in VMs configuration, and may be suitable for quite static scenarios characterized by long-term commitments [3], where cloud customers purchase VMs for extended periods of time (for example, using the Amazon so-called *reserved instances*). However, the emerging cloud scenario requires solutions that support a dynamic behavior where VMs are frequently added and removed from the system.

The main contribution of this paper is an adaptive methodology that dynamically selects the length of the time series used to model a VM behavior depending on the degree of uncertainty resulting from the clustering process. The proposed methodology exploits elements of fuzzy logic for the dynamic determination of the resource time series length. This solution allows the system to take decisions on the VM behavior without the need to wait for long time series of resource measurements, leading to a reactive mechanism able to cope with changes, new deployments and removals of customer VMs in the cloud data center. It is worth to note that different VM clustering algorithms may be integrated in the proposed methodology, leading to a flexible solution where the clustering algorithm may be selected depending on the VMs characteristics.

We apply the proposed methodology to a dataset coming from a private cloud data center hosting a multi-tier e-health application, which is deployed on VMs running Web servers and DBMS. We show that the application of our adaptive methodology can identify up to 80% of the VMs that are correctly classified after just one day of monitoring. For these VMs, clustering-based solutions to improve monitoring scalability can be applied after just one day, while longer observation periods are required only for the classification of the remaining 20% of the VMs. Furthermore, we show that with respect to existing solutions the amount of data collected for VM clustering can be reduced of at least 30% in our

specific case study.

The remainder of this paper is organized as follows. Section II describes the reference scenario for the application of VM clustering. Section III presents the proposed adaptive methodology, while Section IV describes the experimental results. Section V discusses the related work and Section VI concludes the paper with some final remarks.

II. Cluster-based Monitoring and Management

We now describe the reference scenario for our proposal, where a IaaS cloud data center integrates a clustering technique [1], [2] to improve the scalability of monitoring and management. Throughout this section, we outline the main issues of this approach, namely *cluster-based* monitoring and management, and motivate the proposal of an adaptive methodology to model VMs behavior that aims to solve these problems.

We assume that the IaaS cloud system adopts a two-level management strategy, as in [4]. The first level consists in a *local management*, that is performed on each physical server: it detects overload conditions in real-time making use of the resource measurements of the VMs hosted on the server, and exploits live VM migration whenever overloaded servers are detected [5]. The second level is a *global management*, which is controlled by a central node: it is responsible for periodically executing a consolidation technique to place VMs on as few physical servers as possible to reduce the infrastructure costs and avoid expensive resource over-provisioning [6], [7]. Since consolidation strategies in IaaS cloud infrastructures usually consider each VM as a stand-alone object with independent resource usage patterns, detailed information has to be collected with high sampling frequency (typically 1 sample every 5 minutes [6], [7]) about each VM, thus creating scalability issues for the monitoring system.

The cluster-based monitoring may improve scalability by automatically grouping together VMs showing similar behaviors in terms of resource usage [1], [2]. The process of VM clustering is carried out periodically to identify classes of VMs that are running the same software component of the same customer application. Once the clustering is done, few representatives are selected for each identified class. We choose to select at least three representatives due to the possibility that a selected representative unexpectedly changes its behavior with respect to its class can be identified using quorum-based techniques. At this point, only the representative VMs of each class are monitored with high sampling frequency to collect information for the periodic consolidation task, while the resource usage of the other VMs of the same class is assumed to follow the representatives behavior. On the other hand, the non representative VMs of each class are monitored with coarse-grained granularity to identify behavioral drifts that could determine a change of class. Moreover, sudden changes leading to server overload are handled by the local management through live VM migration. This result can be achieved either assuming that the local management system has some knowledge of the load level of neighbor physical servers or that a query to the global management system is issued. The common point of both strategies is that that global consolidation strategy is bypassed to cope with this overload situation.

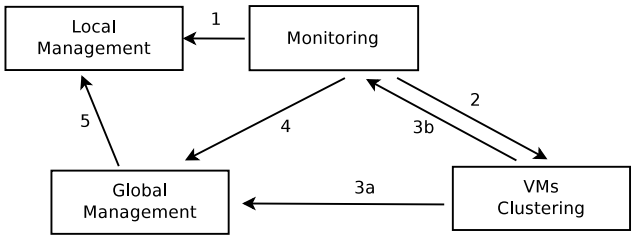

Fig. 1: Cloud system using VM clustering

Figure 1 depicts the interactions between the main components of the reference scenario. The *monitoring* process on each physical server collects data about resource usage of the hosted VMs and sends them to the *local management* system (arrow 1), which is responsible for triggering live VM migration in case of host overload [5]. Moreover, the *monitoring* processes periodically send data to the *VM clustering* system (2), which automatically groups similar VMs applying one of the techniques proposed in [1], [2]. The clustering results (identified VM classes and representatives) are sent to the *global management* (3a) and to the *monitoring* system (3b). The *monitoring* system exploits this information to differentiate the sampling frequency between representative and non representative VMs. The data collected with different granularity are sent to the *global management* system (4) which is responsible for two tasks. First, it periodically executes the cluster-based consolidation strategy, exploiting the resource usage of the representative VMs to characterize the behavior of every VM of the same class: the consolidation decisions are finally communicated to the *local management* systems (5) on each server to be executed. Second, the *global management* system checks for behavioral drifts of non representative VMs; it is worth to note that VMs that change their behavior, as well as VMs causing overload of physical servers detected by *local management*, are marked as unclassified VMs and are monitored again with high sampling frequency to be re-clustered.

Thanks to differentiated sampling frequencies based on the knowledge of VM clusters, this approach may significantly reduce the amount of data collected for the global management of the cloud data center, as discussed in [1], [2]. However, these studies show a clear trade-off between the accuracy of the clustering process (that is, the percentage of VMs assigned to the correct cluster) and the length of the resource usage time series used for VM clustering. The accuracy is quite high (above 80%) even for time series of one day, but for monitoring and management purposes a misclassification of one fifth of the VMs may represent a major problem. On the other hand, to have a clustering accuracy of 100%, long time series (up to 60 days) are required [2], [9] and the need to collect such long time series causes this mechanism to be slow and scarcely reactive to changes in VM configuration. While this delay may be acceptable in a scenario characterized by long-term commitments between cloud provider and customer, it is not suitable in the emerging dynamic cloud scenario. In the next section, we present a novel methodology that exploits a flexible and adaptive mechanism to take advantage of VM clustering without having to wait for long time series collection.

III. Adaptive Methodology

We now describe the proposed methodology to provide highly accurate clustering within a time frame that is compatible with IaaS cloud demands. The proposed mechanism can easily be integrated within the data center cluster-based monitoring and management strategy described in Section II. The proposed methodology exploits two innovative ideas not previously considered in the clustering process.

First, we describe the belonging of a VM to a cluster using concepts derived from fuzzy logic. Such solution is better suited to an adaptive technique than the standard boolean logic used in existing solutions. Specifically, we take into account a *degree of membership* of a VM to each possible cluster. The additional insight provided by the fuzzy logic allows the clustering process to discern between VMs that clearly belongs to a cluster and VMs where the cluster attribution is still undecided. To this aim, we introduce the concept of *gray* and *white areas* at the level of clustering data space. A VM in the gray area does not clearly belong to a single cluster and additional information is required to take a decision. On the other hand, a VM within the white area clearly belongs to one and only one cluster. From a data center point of view, cluster-based monitoring and management of a VM can start as soon as the VM enters the white area, while every VMs in the gray area must be finely-grained monitored for an additional period.

The second qualifying point of our proposal is the use for clustering purposes of time series with different lengths. Clustering occurs on the basis of a VM behavior model that is built starting from the time series collected by the monitor [1], [2]. We observe that longer time series determine better clustering results (that is higher clustering accuracy) as a result of a more accurate VM behavior model. We combine this observation with the previously introduced concept of white/gray areas as follows. After a clustering attempt, VMs within the white area are described by a behavior model that is considered sufficiently accurate to identify their membership. On the other hand, for VMs in the gray area, we need to improve the quality of the VM behavior model exploiting longer time series, hence additional monitoring is required. This leads to an adaptive selection of the time series length used to create the VM behavior models depending on the clustering results.

Figure 2 shows how the basic principles of our proposal are combined into the main steps of a methodology. We start with a group of VMs, whose behavior is unknown. Through fine-grained monitoring, we obtain a set of time series that define the VM behavior model and that are used for the clustering operations. The clustering output is then analyzed by taking into account not just the clustering solution, but also the degree of membership of each VM to every cluster. Using a threshold based on a parameter ϵ, the fuzzy gray area selection step separates the VMs as belonging to the white and gray areas. VMs in the white area are assigned to a cluster. From this point on, the "white" VMs are no longer monitored with a fine grained approach to build a VM behavior model: the existing behavior model is stored and re-used whenever a subsequent clustering operation is invoked. On the other hand, for VMs in the gray area, we collect additional data, we define a new behavior model based on a longer time series, and we re-iterate

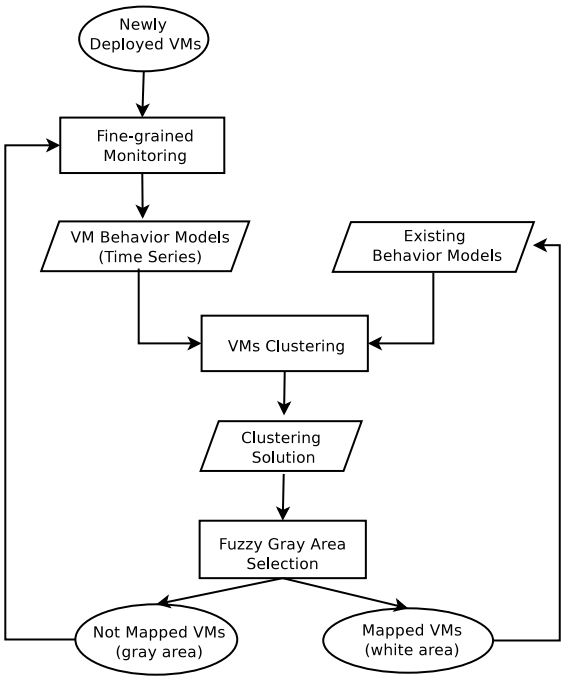

Fig. 2: Methodology steps

the process. The same process is also used to manage the creation of new VMs in the data-center.

We now detail the task of the fuzzy gray area selection. We recall that the clustering process maps each VM within a multi-dimensional space, where clusters and cluster centroids are determined. This step is common to both the k-means clustering described in [2] and to the spectral clustering used in [1]. Assuming that the space supports a euclidean distance operator, we can define a vector D^n for each VM n containing the distances of the VM n from the C centroids of the VM clusters as $D^n = \{d_1^n, \ldots, d_C^n\}$. We consider the distance of a VM from the centroid as a measure of the degree of membership of that VM to the cluster. This assumption is used to define the criteria for the fuzzy gray area selection. For each VM n and for each couple of cluster centroids c_i, c_j with $i, j \in [1, C], i \neq j$, we consider the distance of the VM from the centroids of the clusters d_i^n, d_j^n. If the VM is *nearly equidistant* from two or more centroids, we can consider that VM to be put in the gray area. This bound can be expressed as follows: given the parameter $\epsilon \in [0, 1]$, we consider a VM n to be on the gray area if and only if $\exists i, j$ with $i, j \in [1, C], i \neq j$ such that $1 - \epsilon < \frac{d_i^n}{d_j^n} < \frac{1}{1-\epsilon}$.

If we consider an example with only two clusters, we can plot the distance of each VM from each centroid in a plane such as the one on Figure 3. Then we can draw, for different values of ϵ, the different extensions of the gray area. From the figure we see that, as ϵ grows from 0 towards 1, the gray area increases in width.

The effectiveness of the proposed methodology depends on the size of the gray area and, consequently, on the value of ϵ: if the threshold is too low, we may reduce the accuracy to an unacceptable level. On the other hand, a too high value tends to overestimate the number of VMs that are undecided

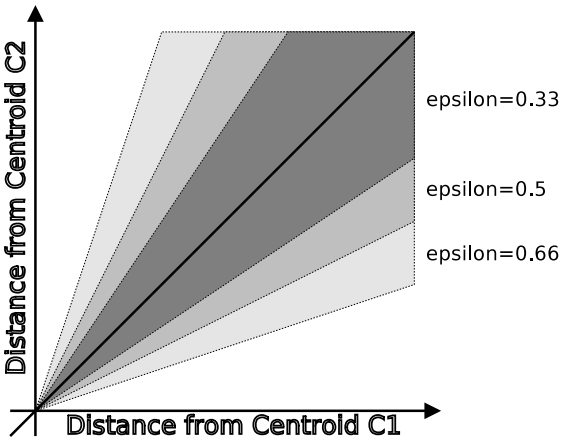

Fig. 3: Gray area definition

and reduces the efficiency of the adaptive clustering. In our experiments we found as a rule-of-thumb that having a gray area equal to one-third of the space (that is $\epsilon = 0.33$) satisfies both the requirements of accurate and efficient clustering. Furthermore, we found this value to be adequate for different clustering techniques, thus suggesting that it can applied to heterogeneous scenarios. However, we believe that identifying a way to automatically determine ϵ deserves a wide analysis comparing the performance of the adaptive clustering for several workloads. In this preliminary paper, we prefer to adopt an empirical value for ϵ and to leave such complex analysis as an open issue to be addressed by a future work.

IV. Experimental results

In this section we evaluate the applicability and the effectiveness of the proposed methodology when applied to a case study based on a real dataset coming from a private cloud data center. After describing the case study used for the experimental evaluation, we carry out two different experiments. In the first experiment, we perform just one iteration of the methodology to analyze the impact of the fuzzy gray area on the VM clustering results. In the second experiment, we reiterate the steps of the methodology on the dataset, showing how each iteration allows to dynamically adapt the length of the time series used for modeling the VM behavior depending on the clustering results of the previous iteration. We also discuss the achieved reduction in the amount of data collected by the monitoring system for global management purposes.

A. Case study

We consider a case study based on a dataset coming from a private cloud data center. Specifically, we consider an e-health Web application for the automated management of lab exams, which is hosted on the data center and deployed on 110 VMs according to a multi-tier architecture. The 110 VMs are divided between the two software components of the Web application: Web servers and back-end servers (that host a DBMS). The goal of our clustering is to correctly separate Web servers from DBMSs. No additional cluster is to be defined as we know that a load sharing system distributes evenly the request across the multiple instances of VMs. The accuracy is defined as the percentage of VM correctly identified.

For the clustering, we collect data about the resource usage of every VM for different periods of time, ranging from 1 to 40 days with a sampling frequency of 5 minutes. The resources monitored include CPU, memory and network, as described in [1]. For VM clustering, we consider two different techniques: a PCA-based technique exploiting the correlation among resource usages [2]; an approach based on Bhattacharyya distance combined with spectral clustering techniques [1].

B. Fuzzy gray area analysis

The first experiment aims to evaluate the impact on the VM clustering results of the fuzzy selection based on the gray area. To this aim, we apply just one iteration of the methodology to the entire set of VMs considering behavior models described by time series of 1-day. Then, we evaluate the fraction of VMs in the gray area and the accuracy of the VMs in the white area.

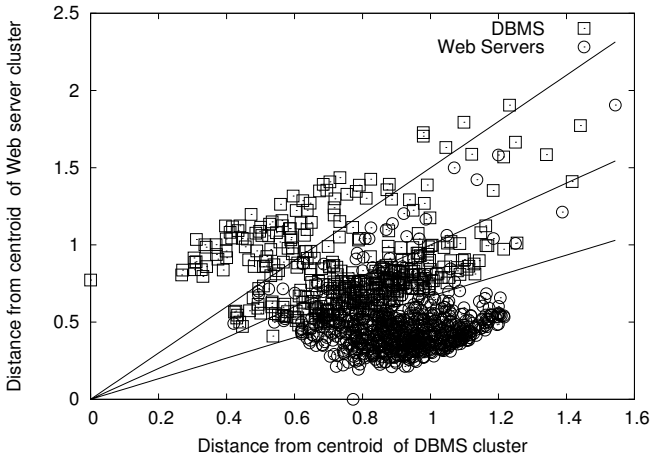

Fig. 4: Scatter plot of distances from cluster centroids

Figure 4 shows the scatter plot of the distances separating each VM from the centroids of the two identified clusters (Web servers and DBMSs) in the case of VM clustering carried out using the PCA-based technique. In the graph, Web servers are represented by circles, while DBMSs correspond to squares. The x-axis measures the distance from the centroid of the DBMSs cluster (represented by the square on the y-axis), while on the y-axis we have the distances from the centroid of Web servers cluster (represented by the circle on the x-axis). Figure 4 also shows three lines starting from the origin. The central line bisects the quadrant, and reveals the actual clustering solution: VMs below the line are classified as Web servers, while VMs above the line are DBMSs. On the other hand, the two external lines delimit the gray area, that is computed with $\epsilon = 0.33$: for VMs between the external lines the clustering is considered uncertain. The most important result in figure 4 is that any incorrectly classified VM is contained in the gray area. This confirms the effectiveness of the fuzzy gray area selection that allows us to achieve a clustering accuracy of 100% for the VMs included into the white area.

To better understand the impact of the ϵ parameter, we now evaluate the percentage of VMs included in the gray area and

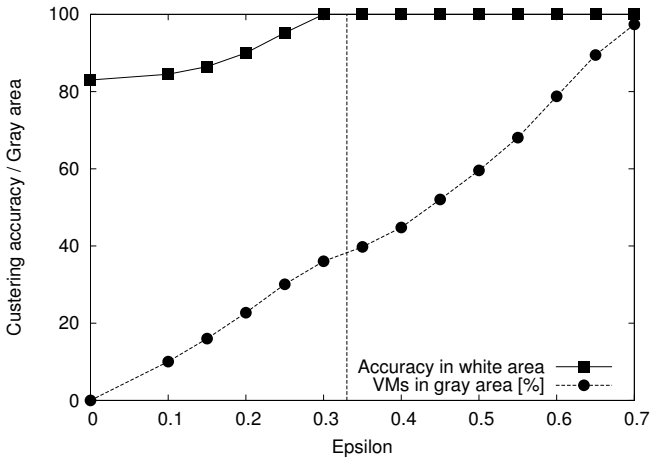

Fig. 5: Clustering accuracy and gray area as function of ϵ with PCA-based clustering

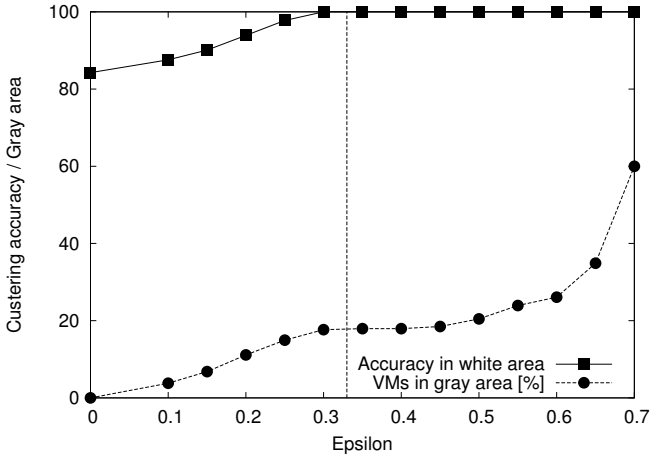

Fig. 6: Clustering accuracy and gray area as function of ϵ with Bhattacharyya-based clustering

the clustering accuracy about the VMs in the white area as a function of this parameter. The results are shown in Figures 5 and 6. In particular, Figure 5 refers to the PCA-based clustering technique, while Figure 6 refers to the clustering based on the Bhattacharyya distance. From these graphs, we see that for both methods a value of ϵ in the order of 0.33 is sufficient to achieve an accuracy of 100% in the white area. On the other hand, we note a difference between the two methods: the line related to the percentage of VMs in the gray area has a significantly different trend in the two graphs. When PCA-based clustering is applied, the size of the gray area grows almost linearly with ϵ; on the other hand, when considering a Bhattacharyya-based clustering, the gray area grows slowly for low values of ϵ, then presents a much more rapid increase for $\epsilon > 0.6$. This suggests that for Bhattacharyya-based clustering only a limited fraction of VMs are in the proximity of the bisecting line, and each cluster presents a large core of VMs apart from the other cluster. This difference can be explained by a better capability of the Bhattacharyya-based clustering to correctly classify a higher number of VMs even with short (1-day long) resource time series.

C. Methodology evaluation

Having demonstrated that the introduction of the fuzzy gray area selection may provides high accuracy in the clustering of the white area VMs, we now evaluate the complete methodology and its capability to dynamically adapt the time series length used for VM behavior modeling depending on the clustering results.

To this purpose, we iterate on the dataset the steps of the methodology described in Section III with a periodicity of 1 day. Initially, we start with time series of 1 day length and we carry out the clustering. For VMs in the white area we consider the clustering correct and we keep the VM behavior representation, while for VMs in the gray area we collect additional samples to create a new VM behavior model based on 2-days time series, then we perform the clustering again. We re-iterate these steps up to 40 days of data collection. In this experiment we consider clustering carried out using both PCA-based and Bhattacharyya distance-based techniques. In both cases we consider $\epsilon = 0.33$ and we confirm that this value is appropriate to achieve a 100% accuracy for clustering VMs in the white area for every iteration of the methodology.

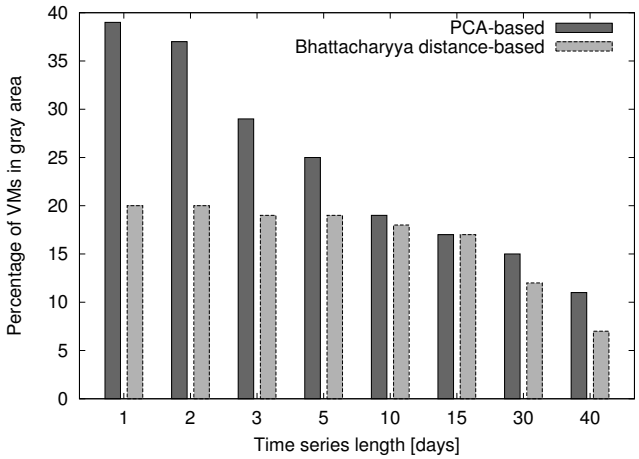

Fig. 7: Gray area for PCA-based and Bhattacharyya-based clustering

Figure 7 shows the results of the experiment for PCA- and Bhattacharyya-based techniques, respectively. For both cases, the x-axis reports the time series length used for VMs that remain in the gray area during the iterations of the methodology, while on the y-axis we show the percentage of VMs in the gray area. We observe that, for both clustering techniques, the size of the gray area decreases as we re-iterate the methodology and increase the time series length used to model the VMs remaining in the gray area. These results confirm that extending the sampling duration only for the VMs in the gray area allows the methodology to reduce the uncertainty of the clustering process, and progressively increases the number of VMs that may be observed through cluster-based monitoring. A comparison between PCA-based and Bhattacharyya distance-based clustering confirms the finding of the initial comparison between the two techniques when

applied to the proposed methodology. The spectral clustering used with the Bhattacharyya distance-based approach tends to create more separate clusters, and the gray area tends to be smaller with respect to the PCA-based alternative, especially for short time series. However, even if the Bhattacharrya-based clustering reduces faster than the alternative the gray area, both approaches need a long time to reduce the amount of VMs in the gray area. After 40 days the gray area for the Bhattacharyya distance-based technique remains higher than 5% of the global number of VMs, while for the PCA-based alternative after 40 days the gray area contains still more than 10% of the VMs.

D. Comparison with existing solutions

We now compare the proposed methodology with existing solutions [1], [2] in terms of clustering results and amount of data collected by the monitoring system. For sake of comparison, we consider a period of time up to 30 days, during which a cluster-based monitoring and management is applied to the 110 VMs of the considered case study. For this experiment, we indicate with $\overline{K}$ the amount of data collected to monitor with a fine granularity (1 sample every 5 minutes) a single VM in 24 hours (1 day).

Table I reports the results regarding the application of existing solutions: specifically, the third and fourth columns show the clustering accuracy achieved by PCA-based [1] and Bhattacharyya-based [2] clustering techniques, respectively. The achieved accuracy is obtained by monitoring all the 110 VMs with fine granularity for the period reported in the first column, thus leading to the amount of collected data in the second column of the table. We observe that the clustering accuracy does not reach 100% even with 30 days of measurements; moreover, with these approaches there is no way to know which VMs are correctly classified, hence the management of the data center has to cope with a not negligible percentage of misclassified VMs. It is worth to note that the fine-grained monitoring may be stopped before than 30 days to start with the cluster-based monitoring and management, but at the price of having an even higher percentage of misclassified VMs.

TABLE I: Clustering accuracy and collected data for existing solutions

Time Series Length [days]	Collected Data	Clustering Accuracy [%]	
		PCA based	Bhattacharyya based
1	$110 \times \overline{K}$	78	83
2	$220 \times \overline{K}$	79	84
3	$330 \times \overline{K}$	80	84
5	$550 \times \overline{K}$	83	85
10	$1100 \times \overline{K}$	84	85
15	$1650 \times \overline{K}$	85	87
30	$3300 \times \overline{K}$	87	92

On the other hand, Table II shows the results for 30 days of application of the proposed adaptive methodology. In this case, the amount of collected data differs depending on the used clustering technique, according to the percentage of VMs in the gray area. For each considered period of time (first column) and for each clustering technique, the table shows the amount of data collected and the percentage of VMs in the white area after the fuzzy gray area selection. Let's for example consider the first two rows of the table in the case of PCA-based clustering. The first iteration of the methodology occurs at the end of 1 day of data collection (first row of the table), hence we have monitored with fine granularity all the VMs ($110 \times \overline{K}$ monitored data) and we have 61% of VMs in the white area as a result of the fuzzy gray area selection. During the second day of monitoring (second row of the table) we collect data with fine granularity just on the 39% of the VMs in the gray area ($43 \times \overline{K}$ data), for a total amount of data collected in both days equal to $153 \times \overline{K}$. Moreover, at the end of the second iteration we have the 63% of the VMs in the white area. And so on.

TABLE II: Clustering accuracy and collected data for proposed methodology

Gray area Time Series Length [days]	PCA-based		Bhattacharyya-based	
	Collected Data	White Area VMs [%]	Collected Data	White Area VMs [%]
1	$110 \times \overline{K}$	61	$110 \times \overline{K}$	80
2	$153 \times \overline{K}$	63	$132 \times \overline{K}$	80
3	$194 \times \overline{K}$	71	$154 \times \overline{K}$	81
5	$256 \times \overline{K}$	75	$196 \times \overline{K}$	81
10	$380 \times \overline{K}$	81	$300 \times \overline{K}$	82
15	$480 \times \overline{K}$	83	$400 \times \overline{K}$	84
30	$745 \times \overline{K}$	85	$645 \times \overline{K}$	88

The advantages of the proposed adaptive methodology are twofold. First, as soon as the first iteration is completed, a large percentage of VMs is in the white area: these VMs are correctly classified and a cluster-based monitoring and management can be applied to them without having to cope with misclassification errors. Second, the fine-grained monitoring goes on just on the VMs remaining in the gray area, thus significantly reducing the amount of data collected with respect to existing solutions: at the end of the second day of monitoring, we have a reduction of collected data equal to 30% and to 40% for PCA- and Bhattacharyya-based clustering, respectively; at the end of the fifth day, the data reduction is equal to 53% and 64% for PCA- and Bhattacharyya-based approaches. Clearly the reduction of collected data increases with the length of the gray area time series, up to a reduction of a factor of 5 for 30 days.

V. Related Work

The monitoring of large data centers is a critical topic where several architectures and software solutions have been proposed. Current solutions typically exploit frameworks for periodic collection of system status indicators. A meaningful example of these solutions is Ganglia[1], which supports a hierarchical architecture of data aggregators that can improve the scalability of data collection and monitoring process. Ganglia is widely used to monitor large data centers [10], as well as [11] by collecting time series on physical hosts and VM metrics. Another solution for scalable monitoring is proposed in [12], where data analysis based on the map-reduce paradigm is distributed over a hierarchical architecture. However, all these solutions share the same limitation of considering each monitored object (being it a VM or a host) independent from

[1] http://ganglia.sourceforge.net/

the others. This approach fails to take advantage from the similarities of objects sharing the same behavior.

A more recent approach aiming to exploit cluster-based monitoring have been recently proposed by the authors in [1], [2], [9]. A critical point of these proposals is the clustering of VMs, that exhibit *similar behavior*, to select for each cluster a few representatives which are finely grained monitored. Several approaches have been proposed to represent the VM behavior, to measure the similarity between VMs, and to cluster similar VMs. For example, in [9] the authors use the correlation between the resources usage on each VM to represent VM behavior and k-means algorithm for clustering. A more sophisticated and better performing approach was proposed in [2], where we use Principal Component Analysis [13] to determine the VMs behavior. In [1] we exploit a histogram-based representation of VM behavior, Bhattacharrya distance [14] and spectral clustering [15] to measure VM similarity and to group together VMs, respectively. However, these solutions share the common limit that, even if the clustering accuracy may be high for short time series, we still have to cope with a non-negligible amount of misclassified VMs. Our proposal addresses this issue by separating the VMs clearly belonging to one cluster (in the white area) from the VMs that are undecided (in the gray area), for which further monitoring must be carried out. Our solution provides high clustering accuracy for the VMs in the white area, thus enabling effective cluster-based monitoring and management for these VMs.

Our proposal exploits concepts of fuzzy logic when processing the clustering results. Clustering algorithms based on fuzzy logic [16] are widely adopted in the area of pattern recognition. While specific algorithms such as Fuzzy C-means could be applied in our methodology, we prefer to introduce the fuzzy logic concept of degree of membership to clustering techniques previously proposed and tested for VM clustering. A comprehensive analysis of clustering algorithms including also soft or fuzzy clustering solutions is left as a possible future work.

VI. Conclusions

As cloud is becoming a key enabling technology for the emerging digital society, new scalability issues are emerging for the cloud infrastructures. We focus on techniques that aim to improve the scalability of monitoring operations in IaaS cloud infrastructures by clustering together VMs with similar behaviors. We point out that existing solutions for VM clustering require to monitor VMs for a long time before being able to provide accurate classification. This delay is not compatible with the demands of cloud systems unless we restrict our operations to the case of long term commitments between customers and cloud providers.

We propose a novel approach where, exploiting the principles of fuzzy logic, we adaptively select the length of the time series used for VM clustering purposes. As soon as a VM is detected as clearly belonging to a cluster, we can apply to that VM the existing approaches to improve monitoring scalability. Furthermore, this solution can be used to cope with the inherent dynamic process of deploying and disposing of new VMs typical of cloud scenarios.

Our experiments demonstrate the viability of the proposal and show that it can be successfully applied to different clustering technique. The experimental results show that we can provide 100% clustering accuracy starting with just 1 day of data for a high percentage of the VMs, while the remaining undecided VMs require longer time series to be clustered.

References

[1] 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.

[2] ——, "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.

[3] D. Durkee, "Why cloud computing will never be free," *Queue*, vol. 8, no. 4, pp. 20:20–20:29, Apr. 2010.

[4] Z. Gong and X. Gu, "PAC: Pattern-driven Application Consolidation for Efficient Cloud Computing," in *Proc. of Symposium on Modeling, Analysis, Simulation of Computer and Telecommunication Systems*, Miami Beach, Aug. 2010.

[5] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and gray-box strategies for virtual machine migration," in *Proc. of Conference on Networked systems design and implementation (NSDI)*, Cambridge, Apr. 2007.

[6] D. Ardagna, B. Panicucci, M. Trubian, and L. Zhang, "Energy-Aware Autonomic Resource Allocation in Multitier Virtualized Environments," *IEEE Trans. on Services Computing*, vol. 5, no. 1, pp. 2 –19, Jan. 2012.

[7] T. Setzer and A. Stage, "Decision support for virtual machine reassignments in enterprise data centers," in *Proc. of Network Operations and Management Symposium (NOMS'10)*, Osaka, Japan, Apr. 2010.

[8] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in *OSDI*, M. I. Seltzer and P. J. Leach, Eds. USENIX Association, 1999, pp. 173–186.

[9] C. Canali and R. Lancellotti, "Automated Clustering of VMs for Scalable Cloud Monitoring and Management," in *Proc. of Conference on Software, Telecommunications and Computer Networks (SOFTCOM)*, Split, Croatia, Sept. 2012.

[10] A. N. Naeem, S. Ramadass, and C. Yong, "Controlling Scale Sensor Networks Data Quality in the Ganglia Grid Monitoring Tool," *Communication and Computer*, vol. 7, no. 11, pp. 18–26, Nov. 2010.

[11] C.-Y. Tu, W.-C. Kuo, W.-H. Teng, Y.-T. Wang, and S. Shiau, "A Power-Aware Cloud Architecture with Smart Metering," in *Proc. of 39th International Conference on Parallel Processing Workshops (ICPPW'10)*, San Diego, CA, Sept. 2010.

[12] M. Andreolini, M. Colajanni, and S. Tosi, "A software architecture for the analysis of large sets of data streams in cloud infrastructures," in *Proc. of the 11th IEEE International Conference on Computer and Information Technology (IEEE CIT 2011)*, Cyprus, Aug.-Sept. 2011.

[13] A. Sharma and K. K. Paliwal, "Fast principal component analysis using fixed-point algorithm," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1151–1155, July 2007.

[14] E. Choi and C. Lee, "Feature extraction based on the Bhattacharyya distance," *Pattern Recognition*, vol. 36, no. 8, pp. 1703 – 1709, Aug. 2003.

[15] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 176 – 190, Jan. 2008.

[16] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 29, no. 6, pp. 778–785, 1999.