2013 IEEE Sixth International Conference on Cloud Computing

Scale-Space Filtering for Workload Analysis and Forecast

Gustavo A. C. Santos, José G. R. Maia, Leonardo O. Moreira, Flávio R. C. Sousa, Javam C. Machado
Departament of Computer Science
Federal University of Ceara
Fortaleza, Brazil
{gsantos, gilvanm, leoomoreira, sousa, javam}@ufc.br

***Abstract*—Dynamic resource provisioning poses a major challenge for infrastructure providers because it is necessary to both forecast resource consumption and react to recent surges on demand for maintaining a tradeoff between quality of service and cost. However, approaches to workload analysis and forecast are affected due to noise in observed data, specially in forecast models. Moreover, most studies do not consider different prediction horizons may be necessary in order to take action before an SLA violation occurs. This paper presents an approach based in the scale-space theory to assist the dynamic resource provisioning. This theory is capable of eliminating the presence of irrelevant information from a signal that can potentially induce wrong or late decision making. In order to evaluate our approach, some experiments are presented considering both reactive and proactive strategies. The results confirm that our approach improves the workload analysis and forecast.**

***Keywords*-Scale-Space, Modeling, Prediction, Workload.**

I. Introduction

Cloud computing provides on-demand services with pay-as-you-go model. One major benefit claimed for this computing model is elasticity, which adjusts the system's capacity at runtime by adding and removing resources without service interruption in order to handle the workload variation. This characteristic makes it an attractive paradigm potentially able to meet the most strict Quality of Service (QoS) levels stipulated in a Service Level Agreement (SLA). However it is still a challenging issue for dynamic application providers to efficiently tackle both situations of gradual load variations and load peaks from the workloads seen by dynamic applications, which are highly dynamic and, as such, very unpredictable [1], in order not to violate SLA requirements so as to maximize their revenues.

Resource provisioning plays a key role in ensuring that the service providers adequately accomplish their obligation to costumers while maximizing the utilization of underlying infrastructure [2]. In this scenario, it is clear that one needs to avoid both under-provisioning, which leads to application slowdown, and over-provisioning, which leads to unnecessary resource costs [3]. Service providers can reduce the inefficiency caused by these situations by optimizing the number of active servers to support a given user base [4].

Dynamic provisioning techniques are designed to handle workload fluctuations [5] so that SLA violations and their associated contractual penalties can be avoided or limited, and reduce cost. These techniques usually take actions based on workload observation and can be classified as either reactive or proactive. Proactive solutions apply sophisticated system models for prediction [6] and use resulting forecasts for triggering allocations in advance of expected need. In contrast, reactive approaches do not use prediction, but rather detect and react to existing resource bottlenecks by means of predefined thresholds for application-level or system metrics. In the last years, researchers have been employing both these techniques in order to deal with the resource provisioning issue [7][5][8][9][10][11][12].

Even though releasing resources is usually not such a complex task, acquiring resources incurs performance overheads [13], specifically their setup time, but also, in case of the database tier, the time needed to handle replication and synchronization of the associated disk state of the database in order to preserve data integrity [1]. Once these actions take certain time to be effectively performed, it is suitable to tackle the resource demand problem from the time series analysis perspective, in which data are represented in a line graph that records the values of a given set of variables (datapoints) that describes the system's state within a period of time. This representation plays a central role in the development of resource provisioning approaches.

Thus, independently on which approach is used, reactive or proactive, being able to determine the degree of relevance of an observed datapoint and, consequently, also being able to eliminate the influence of irrelevant ones are desirable features. In our context, being a relevant datapoint means whether it is in a peak or in a depression in the observed workload signal that lasts a period of time long enough to justify the addition or removal of resources. If it is not the case, then it is irrelevant and might misguide system's decisions if kept for consideration. This may lead to the addition (removal) of resources that will most likely be removed (added) in a short period of time, thus causing unecessary overheads. An important observation is that irrelevant datapoints also include noisy data, which are often indicative either of measurement error or some system instability that may have occurred during monitoring. Instabilities in the studied phenomena may also happen due to a monitoring strategy that interferes with the system's normal operation stressing or altering it.

As such, in this paper, we apply scale-space theory [14] to

978-0-7695-5028-2/13 $26.00 © 2013 IEEE
DOI 10.1109/CLOUD.2013.119

IEEE computer society

assist the dynamic resource provisioning approach. This theory is a qualitative signal description in multi-scale measurement [15], thus enabling multiple interpretations of the data. Also, it allows us to derive such multi-scale representations in a mathematically sound way [16]. By means of choosing the right scale, it is a powerful technique for eliminating, or considerably diminishing, the presence of irrelevant information from a signal that can potentially induce wrong or late decision making by both proactive and reactive methods. More importantly, it does so with mathematical guarantees that no additional structures, i.e. new peaks and depressions, are introduced in the process as the original sampling is kept unchanged over time. This is a significant difference between this approach and similar representations usually applied elsewhere.

The contributions of our paper are:

- we specifically carry out time-series analysis in order to eliminate irrelevant information and thus obtain a signal that better explains the underlying behavior of interest;
- we apply scale-space theory in the context of dynamic provisioning of resources, which, to the best of our knowledge, has not yet been experimented. Thus, information provided might be used in decision-making for elasticity in the cloud environment;
- we show that the Support Vector Machine for Regression (SVR) ϵ-SVR [17] and the Autoregressive Integrated Moving Average (ARIMA) [18] forecasting models benefit from our approach and yield reasonable results.

The appoach is robust to noise and may be used together with both reactive and proactive solutions. This time-series analysis constitutes the core of our work and we can easily extend it to other time-series of interest even in the multivariate context.

The remainder of this paper is structured as follows: In section 2, we discuss related work. In section 3, we present our approach. Evaluation of the proposed technique is presented in section 4. Finally, section 5 concludes the paper.

II. Related Work

Among works that adopt reactive solutions, in [7] authors present RepliC, an approach to database replication in the cloud with quality of service, elasticity, and support to multi-tenancy. In order to cope with elasticity, the rule for their reactive approach was defined taking CPU utilization as a threshold. Their decisions are triggered after an observation timespan of two minutes. The monitoring strategy they adopt consists of storing two CPU utilization values over a two minutes timespan obtained by median and standard deviation calculations. The two medians with lower deviation are selected as the final values to be stored. To these values, it is applied an exponentially weighted moving average. Sakr and Liu [5] develop a framework whose aim is to facilitate adaptive and dynamic provisioning of the database tier. In their work, both CPU utilization and percentage of the SLA satisfaction of the workload transactions are considered and decisions are based on observations for a period of five minutes. In [9] streaming applications in a cloud environment are considered. The goal is to adjust virtualized CPU allocation so that the processing rate can meet the data arrival rate, while optimizing the computing costs based on a reactive statistical approach.

Other works adopt a prediction strategy to act before the occurrence of SLA violation and in time so that they can be avoided. In [19], authors employ a forecast technique based on regression analysis and Markov state diagram to predict load in the near future. In [10] auto-regressive linear prediction and neural network prediction methods are examined in order to forecast the future load demand profiles. The authors of [11] employs a load prediction algorithm based on ARIMA model to gradually adapt resource according to the future demand. Authors in [20] adopt the ARMA autoregressive model in order to capture both trends and seasonal patterns in workload variations. They also propose predictive data grouping and placement methods based on the access history and load prediction model, thus allowing for agile scaling up and down in cases that cannot be foreseen by the predictive model.

These works deal with specific situations and observations are taken over a fixed and empirically obtained time scale. Also the impact caused by a noisy dataset is not discussed. Our approach naturally allows an analysis in any time-scale deemed viable by the application to perform resource provisioning. For example, in [20] training data is observed for a period of two weeks, so no forecasting is possible before day 15. Moreover, training data does not contain workload variation at minute or hour, but days.

Matsunaga and Fortes [21] present a comparative evaluation of various machine learning techniques for predicting time and resources consumed by applications, arguing that such techniques are promising because they are capable of taking a large number of attributes in account. These authors also extend the Predicting Query Runtime in order to choose/pick a specific regression method that is best suited for dealing with certain data patterns, then concluding that different algorithms perform better depending on the situation although fine parameter tuning and training time pose as a challenging problem. However, [21] does not include a comparison against other techniques well-known in literature, such as Bayesian methods, autoregressive models and Markov chains.

III. Our Approach

Both reactive and proactive techniques might be sensitive to the highly dynamic nature observed in dynamic applications' workloads. As such, our approach aims to be a helpful tool in order to avoid the necessity to monitor those changes that are not significant to an application's context but which can equally be able to trigger decisions that will most likely need to be readily undone.

An overview of our approach is depicted in *Figure 1*. Its first component consists in obtaining a higher level description of a set of metrics (signals) $f_i(x)$ by applying the scale-space filter. This allows us to select a scale or a set of scales by looking at how the interpretation of the data changes as the scale is varied. The obtained signal is ready to be used by a reactive provisioning algorithm.

If it is desired to adopt a proactive approach, the prediction module, which wraps the training and prediction steps using specific forecasting models, might receive signals which are not preprocessed or the signals in their new scale after application of scale-space. Forecasts are then performed in order to be used by a proactive provisioning algorithm.

SLA metrics define the items to be monitored, and, in our case, we made use of the metric *workload* as $f(x)$ and we propose a proactive approach based on ϵ-SVR and ARIMA at the prediction module. To focus on the provisioning algorithms is not in the scope of this paper. In future work we will define strategies to cope with this task.

The reason we use *workload* to perform our analysis is that this metric plays an important role in the cloud environment since it is very dynamic and any information about it may only be obtained at runtime. Although not linearly, other metrics' behaviors are related to it, and we believe that if we are able to improve our analysis over the *workload* signal we might also take more informed decisions in future multivariate studies. Still, we highlight that any metric or set of metrics may be used, as may be any reactive or proactive approach.

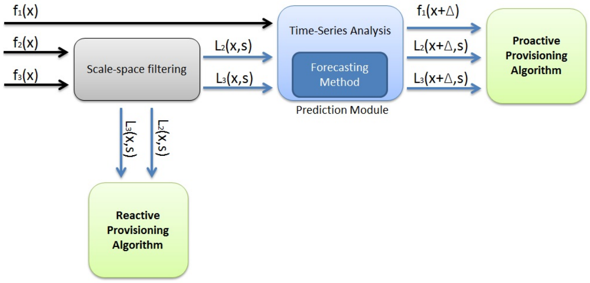

Fig. 1. Our approach. Scale-space filtering is optionally applied to a signal $f_i(x)$, which generates another signal $L_i(x, s)$ in a new scale. The obtained signal might be used by a reactive provisioning algorithm. Also it can be used at the prediction module to generate a forecast that may be used by a proactive provisioning algorithm.

A. Scale-space

As stated in [22], the main idea of creating a multi-scale representation of a signal, as shown in *Figure 2*, is by generating a family of one-parameter (scale) derived signals, each of them based on the original one and presenting a decreasing level of detail as the scale increases. As a result, unnecessary features and noise are removed or strongly attenuated at a wider scale so the signal processing may be concentrated over features shown at that scale.

Given a signal $f : R \to R$, the scale-space [14] representation $L : R \times R_+ \to R$ is defined such that the representation at "zero scale" is equal to the original signal $L(\cdot; 0) = f(\cdot)$ and the representations at coarser scales are given by convolution of the given signal with Gaussian kernels of successively increasing width.

The Gaussian function is used because it is the unique kernel satisfying the *Scaling Theorem* [23]. It states that Gaussian is the only kernel for which local maxima either remains the same or decrease and local minima either remains the same or increase as the bandwidth of the filter *'s'* is increased, i.e., with increasing scale. The reverse is also verified: if a convolution kernel never introduces additional structure, then it must be gaussian. Therefore, no additional structures are introduced by this process [24]. Moreover, most of the structures in the signal with a characteristic length less than *'s'* are removed after convolving a signal by $g(\cdot; s)$. Hence, the bandwidth of the filter controls the type of high frequency information we want to eliminate from the signal.

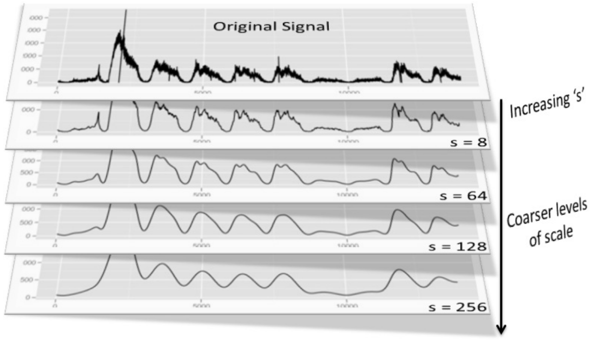

Fig. 2. A multi-scale representation of a signal.

Consequently, we can also minimize the effect of data points that, although are not noisy objects or outliers, are still irrelevant or only weakly-relevant [25] to the underlying data analysis. In our context, such datapoints are extrema whose duration might be considered irrelevant to be taken into account by provisioning strategies.

However, as we apply higher values of *'s'*, the interval between the original and the obtained extrema increases. Although the resulting signal does capture the system's behavior, it becomes less useful in expressing the system's real demands, since the observed values will be somewhere below the expected ones, which should be as close as possible to the original value. Such behavior is expected, given that for small values of 's' the Gaussian convolution approaches the un-smoothed signal and for large 's' the signal's mean [14].

We overcome this problem by using linear interpolation [26], which allows us to restore the amplitudes of the smoothed signal to their original values. This is depicted in *Figure 3*.

With the application of both scale-space smoothing on the signal that represents the monitored metrics and interpolation on the obtained smoothed signal we end up with a dataset that is either free of noise or contains an easily treatable amount of it and in which the relevant extrema correspond to their counterparts in the original signal.

IV. Evaluation

A. Experimental Setup

In the following, two types of experiments are conducted. The first one aims to corroborate the hypothesis presented in the previous sections about the advantages of using scale-space technique together with reactive solutions in order to deal

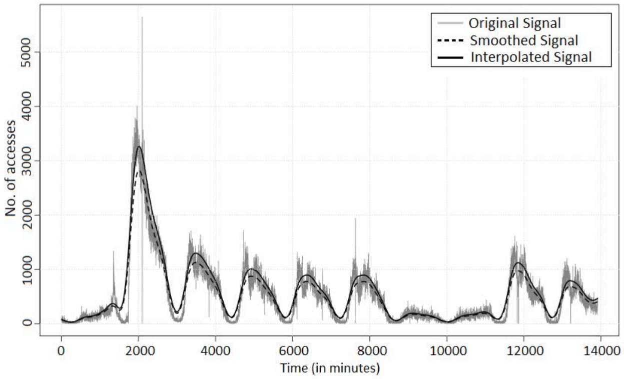

Fig. 3. Interpolating the signal obtained by a high value of *'s'* in the scale-space algorithm with the original one helps bringing the extrema close to their original amplitude. This is very helpful, since the detection of extrema plays a key role in resource provisioning.

with sudden workload variations. We show that by analyzing the signal in a scale that disregards regions of peaks and depressions considered irrelevant by an application, one can readily take provisioning actions without the need to wait for a specified period of time to determine whether such actions should be triggered. Since this observation time is not expended, we can diminish SLA violations.

In the second group of experiments, we use SVR and ARIMA over both a scale-space preprocessed signal and the original one to show how this technique increases prediction accuracy because the noise that often masks observations in data-driven modeling approaches is removed or greatly reduced and because, since the irrelevant datapoints are not considered, obtained forecasts will most likely reflect the system's real needs for resources.

B. Implementation

We implemented the scale-space algorithm in C language. It receives the Gaussian's standard deviation $s = \sqrt{2t}$ as parameter and interpolation is supported. The prediction module was implemented in Python and uses a binding for LIBSVM [27]. In our implementation of the scale-space algorithm, we pass as input parameters the original signal and *'s'*, which explicitly represents a threshold in time units under which peaks and depressions are considered irrelevant.

ARIMA and SVR are run, the latter with the Radial Basis Function (RBF) kernel, over both the the transformed signal and the original one.

C. Workload

We gathered a workload trace from our institution's academic management system during the enrollment period of the undergraduate students for the first semester of 2012. It happened from $01/22/2012$ to $01/31/2012$. The number of accesses during this period is presented in intervals of one minute, totalizing ten days of collected data. This can be seen in *Figure 4*

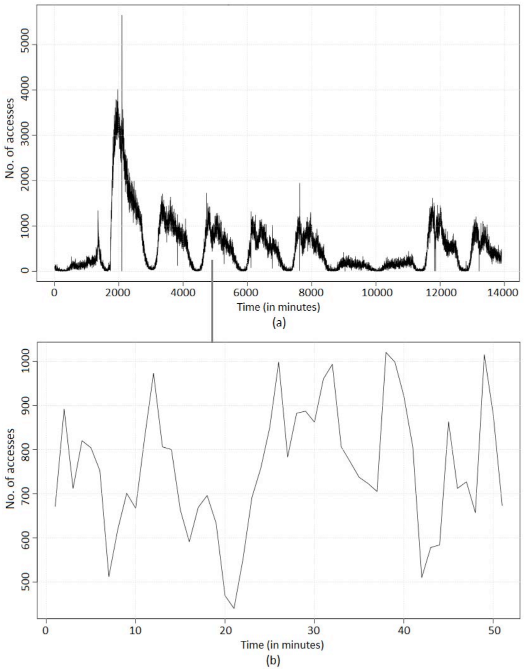

Fig. 4. Workload trace during the enrollment period is shown in (a), which happened from 01/22/2012 to 01/31/2012. In (b), a workload trace of 50min, from the pointed region in (b), is presented.

The reason we chose this workload for our tests is that it resembles the type of workload seen in typical web-based dynamic applications [1][28]. They are usually characterized by dynamically varying workloads that contain both long-term and short-term fluctuations. The first type, from which one can normally extract a pattern or pinpoint factors that influence it, includes variations that can be affected by the season, time of the day, or day of the week, to name a few. These are usually predictable, but may also trick a forecast method due to external influences like last hour schedule modifications, delays or changes to certain functionalities, which happen in our system from time to time. The latter type is normally due to flash crowds and can be very unpredictable . In *Figure 4* we can see both these types of fluctuations in (a) and (b) respectively.

Even though in this work we focus in a time-scale of minutes and perform short-term predictions, our workload also allows us to perform analysis in the different time scales (hours, days or weeks for example).

D. Experiments

In this section, we present our experimental results on the use of scale-space for improving resource provisioning techniques in dynamic environments.

1) Experiment 1: In order to conduct this experiment, we must first define a time interval under which all information should be considered irrelevant to a provisioning strategy. Based on the time of observation of some related work, we choose this time interval to be of five minutes and, as such, we run the scale-space algorithm in our original signal with $s = 5$. This is because, as we mentioned before, most of the structures in the signal with a characteristic length less than *'s'* are removed after convolving a signal by $g(\cdot; s)$. As such, we are able to concentrate on the features shown at this specific scale, which prevents us from detaining our attention on irrelevant information. Thus, in other words, all peaks and depressions over this interval of five minutes are to be considered as irrelevant.

In *Figure 5* we represent data obtained from Wednesday, January $25th$. In (a) we ran the scale-space algorithm over the interval from 12:27 PM to 1:17 PM. Observe that those peaks and depressions that last less than five minutes are no longer represented in the obtained signal. Also notice in (b), which represent data from 12:27 PM to 12:58 PM, that the behavior of the obtained curve does not change as we execute the algorithm, with the same parameter, over a shorter range of a specific region. This behavior has been observed in every experiment and it guarantees that, if we run the algorithm on-line, over the original signal, irrelevant datapoints will not appear in the obtained scale over which we are focusing our observations. Thus, any variation that appears will be meaningful and prompt action should be taken by a reactive solution.

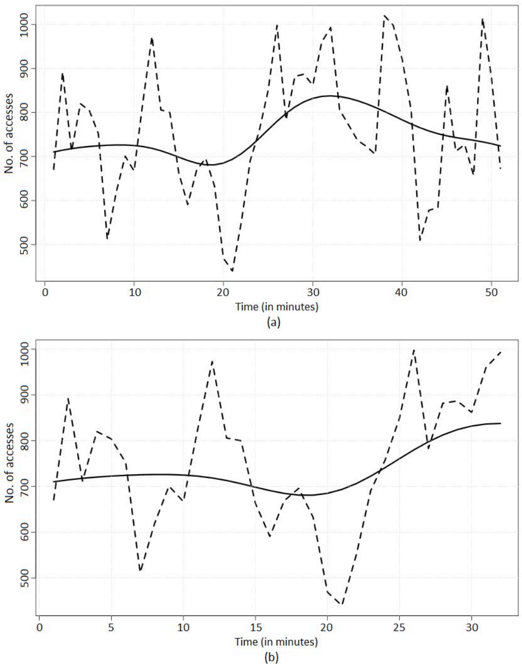

Fig. 5. Application of scale-space algorithm over a region of the workload (a) and a shorter range of it (b). The shorter timespan in (b) is obtained by removing samples from the right of the interval shown in (a).

If we apply another smoothing technique, like the moving average, in the same situation presented in *Figure 5*, we may also notice that the behavior of the curve does not change in the shorter region, as we keep the same window size of five in both of them. This is depicted in *Figure 6*.

Notice, however, two interesting situations. By choosing a window size equal to five in the moving average, the same value used as the scale-space *'s'* parameter, we are not able to conclude anything related to the type of information that is removed from the signal. Thus, there is no relationship between its parameter and a unit of time that may lead us to have considerable control over the type of information we are supressing from the signal, as we do with scale-space. Besides, and most importantly, notice that there are some regions where peaks (depressions) appear where before there was a depression (peak). Thus, structures are being added to the obtained signal, and that might compromise resource provisioning strategies. With scale-space, however, due to the *Scaling Theorem*, as we mentioned before, this never happens in scale-space smoothing.

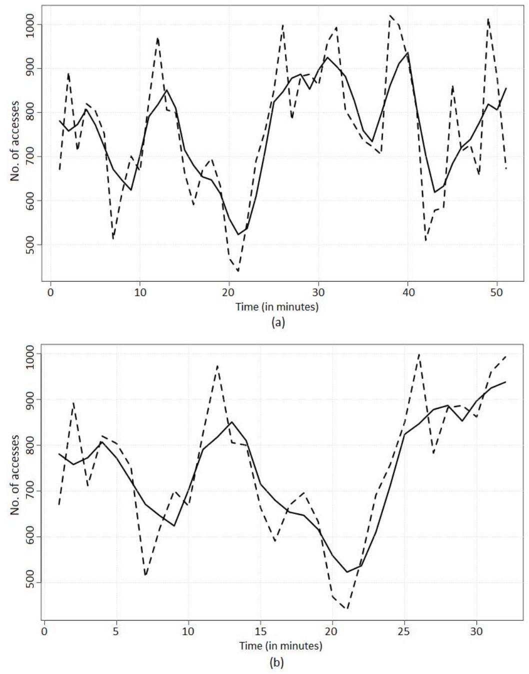

Fig. 6. Application of moving average algorithm over a region of the workload (a) and a shorter range of it (b). The shorter timespan in (b) is obtained by removing samples from the right of the interval shown in (a).

Figure 7 clearly shows the difference in the behavior of obtained signals in different scales after applying the scale-space algorithm to the interval from Wednesday, $25th$, 12:27 PM to 12:39 PM. If we change the value of *'s'*, from (b) $s = 5$ to (a) $s = 1.5$, the obtained signal will consider relevant those peaks and depressions with more than one and a half minutes.

This shows us that the mathematical properties of scale-space algorithm gives us a strict control over the type of information we want a signal to preserve, by means of the standard deviation parameter, with the guarantee, by the *Scaling Theorem*, that no additional information will be added. Also, the scale-space filtering, by definition, is the convolution of the input curve (length n) with a Gaussian filter of a chosen scale (size s). Thus, the complexity for computing each scale is

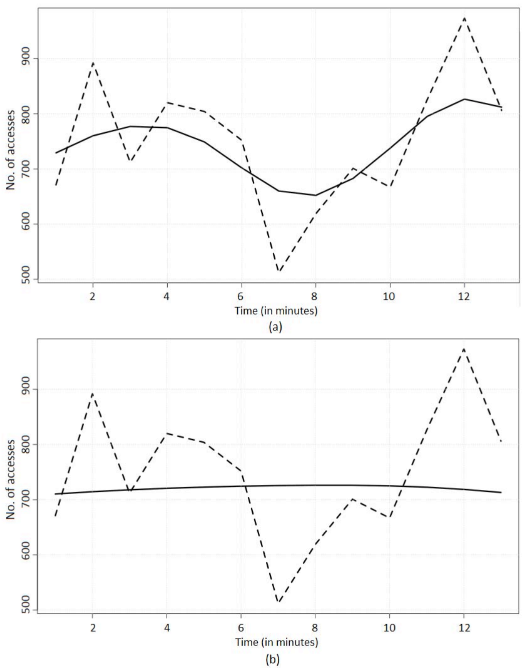

Fig. 7. Application of scale-space algorithm with different values of *'s'* for the same signal. In (a) we have $s = 1.5$ and in (b) $s = 5.0$.

in the order of O(ns), which is not computationally expensive.

Also, we emphasize that we can easily perform the time-series analysis on different time-scales, as depicted in *Figure 8*. There we apply $s = 120$ and interpolation, thus making our signal suitable, for example, to a daily based analysis. With this value of *'s'*, we elimante information that may be considered irrelevant when we want to predict the behavior of the signal on a specific day based on the observation of previous ones. In these cases, performing subsampling is admissible given that the error introduced by this process is not very significant in the approximation of the original signal. Thus, as the number of samples is reduced, calculations are also simplified and the overall method performs faster. Notice as well in this figure that the signal is smoother but reflects very well the behavior of the system in a daily basis.

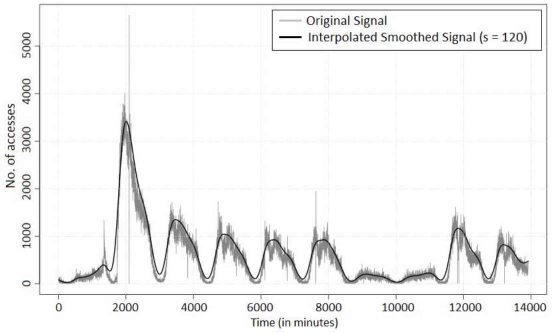

Fig. 8. Application of scale-space algorithm with $s = 120.0$

This easiness to change from one time-scale to another can be useful to applications that might want to vary the granularity level of the analysis they perform from a finer level to a coarser one and back forth.

2) Experiment 2: For this experiment, we applied SVR and ARIMA regression over both the original signal and the result of scale-space filtering in order to evaluate the benefits of our approach in combination with proactive strategies.

The main issue in SVR generalization resides in the complex relationship among its hyper-parameters [29] because it is very hard to obtain values that perform better for different training sets given a specific problem. Moreover, computationally expensive situations may arise both from model training and parameter selection. Thus, we empirically determined a set of values to those hyper-parameters which led to obtain acceptable generalization levels. ARIMA, on the other hand, is executed by fitting a model to each observed region by parameter selection since this computation is not so expensive.

A comparison of obtained forecasting results in the signal after application of scale-space is presented in *Figure 9*. As in the previous experiment, we chose to keep $s = 5$, since at this point we intended to perform short range predictions. Both SVR and ARIMA show varied accuracy levels. Still, notice that in (a) they are capable to deal with local extrema and keep performing well after this kind of event in the signal. In (b) we have a typical situation in which none is accurate due to the occurrence of unforeseen behavior. About the size of both the observation window and the prediction horizon, it was observed that changes do not necessarily affect the quality of the model, although it does happen in many cases. In some cases, reducing the observation window from 30min to 20min does not significantly affect forecasting quality. On the other hand, increasing its size in some cases lead to worse models.

A curious behavior was observed when SVR performs erroneous forecasts: it exceeds the value actually observed in the prediction horizon, but this growth in the workload is observed a little ahead. Thus SVR would lead to a pessimist proactive provisioning algorithm that typically will avoid SLA violations using a bit more resources than necessary.

In *Figure 10* we have the same signal as in *Figure 9* (a), in its original form, i.e., before applying scale-space smoothing. It is important to notice that we are not deliberately smoothing a signal with the specific goal to obtain better forecastings. We are making controlled choices of the scale-space *'s'* parameter which will eliminate unwanted information. This process allows us to focus our analysis on signals of practical interest. So information is lost in the process, but only that which is of no concern to us either because it is noisy or because it is irrelevant to be considered. This contributes to minimize unnecessary additions and removals and also provides us with a more significant signal to perform time-series analysis since meaningful information becomes more evident.

We intend to perform this experiment with lower values of *'s'*. The objective is to try to find the scale that best emphasizes the underlying signal behavior to allow us to better

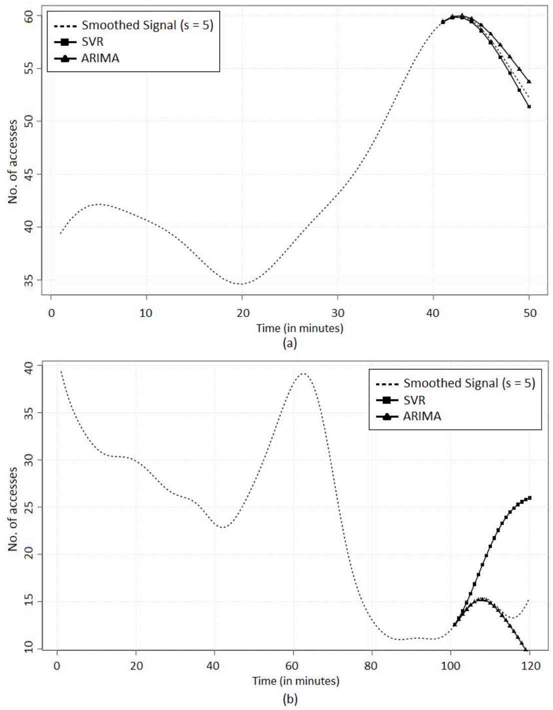

Fig. 9. Forecasts with both SVR and ARIMA over the scale-space preprocessed signal. In (a) both methods perform equally well. The observation window here is $40min$ long, with a prediction horizon of $10min$. In (b) we present a situation in which both methods are not accurate. In this case, observation window is $100min$ long, and the prediction horizon, $20min$.

capture meaningful patterns on the signal, thus improving our forecastings and allowing to obtain more flexible observation and prediction windows. Notice that, in order to do so, without losing peaks and depressions information, the value of *'s'* should not be greater than the amount of time necessary to allocate or deallocate resources in the cloud.

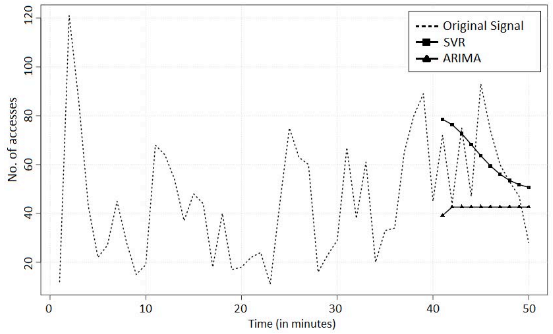

Fig. 10. Forecasts with both SVR (blue) and ARIMA (red) over the original signal.

We chose three metrics to evaluate the accuracy of our predictions: *root-mean-square error (RMSE)*, *Mean Absolute Percentage Error (MAPE)* and *PRED(25)* [30]. Lower values of RMSE and MAPE indicate superior prediction accuracy. The measure PRED(25) is defined as the percentage of observations whose prediction accuracy falls within 25% of the actual value. For this metric, values closer to 1 (100%) indicate a better fit of the prediction model.

We applyed these metrics in a set of predictions over the period of one day, on January 24th, in order to evaluate the suitability of each SVR and ARIMA to the observed workload over both the original signal and the scale-space smoothed one. These predictions were run every ten minutes, using the last 30 obtained samples, thus representing 30min each, in order to predict the following 10min. The values are presented in *Table I*. With the application of scale-space, we were able to reduce the error observed in our predictions, as we can observe with the RMSE and MAPE metrics. As a result, by the observation of the PRED(25) metric, we were able to increase the accuracy of our short range predictions.

	SVR			ARIMA		
	RMSE	MAPE	PRED(25)	RMSE	MAPE	PRED(25)
OS	128.71	0,203	0,738	129,45	0,197	0,762
S-S	34,76	0,040	0,994	28,51	0,021	0,997

TABLE I
COMPARISON SVR X ARIMA IN BOTH THE ORIGINAL SIGNAL (OS) AND SCALE-SPACE SMOOTHED SIGNAL (S-S)

It also worths noticing that predictions in the preprocessed signal are also qualitatively better given that, once the noise is removed, the obtained forecasts are more likely to capture the real system's behavior that is masked by the noise present in the original signal. Going one step ahead and eliminating, in a controlled manner, information that is considered irrelevant to the underlying study proves to be a very valuable tool. It improves forecasts as a consequence of a well-founded sequence of steps rather than as an unjustified pursuit to an end that will show up to be only quantitatively better, thus giving a false idea of quality.

In *Figure 11* we present forecasting results described in *Table I* with both SVR and ARIMA over the whole scale-space signal in the considered region. The results over the original signal are not presented due to space issues.

Since SVR hyper-parameters are powerful means for regularization and adaptation to the noise in training data, one might argue that by properly tunning these hyper-parameters, it is possible to obtain a curve that is already able to disregard both noise and the irrelevant information from the forecasting without the need to apply scale-space to the original signal. Although it might be possible, recall that we mentioned that it is not straightforward to select SVR hyper-parameters properly given the complex interdependence relationship among them. As such, allowing SVR to deal with datapoint relevance can be an exhaustive work of trial and error that presents no guarantees as those provided by scale-space. In an environment that requires fast decisions, this would turn to be unfruitful.

V. CONCLUSION AND FUTURE WORK

In this paper we presented an approach in order to use time series analysis for resource usage prediction in a dynamic

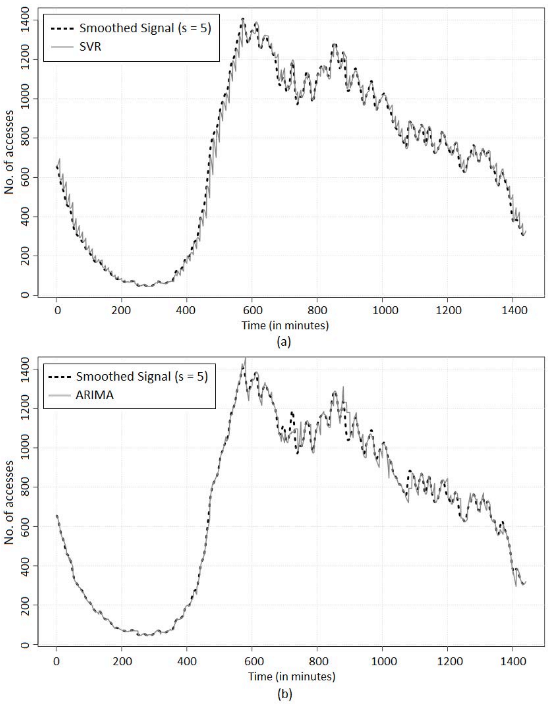

Fig. 11. Predictions obtained with (a) SVR and (b) ARIMA over the time-series obtained from January 24th after scale-space was applied with $s = 5$. Forecastings were gathered for the whole day and were run every ten minutes, using 30 samples representing 30min in order to predict the following 10min.

environment and with varying workloads. We compared performance obtained by two widely used forecasting methods, SVR and ARIMA, which are representative, respectively, from machine learning and from statistical time series analysis. Further, we evaluated proposed approach under the light of a real-world workload. From the results, our approach is suitable for both reactive and proactive strategies. Moreover, it can be used with other types of time-series. In our experiments, we use a workload of dynamic characteristic, having a strong correlation with the demand for resources in such environments.

As future works, we wish to: test workloads with different characteristics, in order to cover various types of applications of varying contexts; incorporate to our solution other variables such as CPU, memory, response time; categorize workloads by, for example, transactions' type. Also, we intend to integrate our approach to a number of provisioning systems. Finally, to study on the influence of the granularity of time on the forecast performance, increased efficiency and cost reduction are other directions to follow to improve our solution.

REFERENCES

[1] E. Cecchet, R. Singh, U. Sharma, and P. Shenoy, "Dolly: virtualization-driven database provisioning for the cloud," in *VEE '11*, 2011, pp. 51–62.

[2] S. K. Garg, S. K. Gopalaiyengar, and R. Buyya, "Sla-based resource provisioning for heterogeneous workloads in a virtualized cloud data-center," in *ICA3PP (1)*, 2011, pp. 371–384.

[3] S. Vijayakumar, Q. Zhu, and G. Agrawal, "Dynamic resource provisioning for data streaming applications in a cloud environment," in *CLOUDCOM '10*, 2010, pp. 441–448.

[4] Accenture, "Cloud computing and sustainability: the environmental benefits of moving to the cloud," november 2010. [Online]. Available: http://www.accenture.com/

[5] S. Sakr and A. Liu, "Sla-based and consumer-centric dynamic provisioning for cloud databases," in *IEEE CLOUD*, 2012, pp. 360 –367.

[6] S. Ghanbari, *Cost-aware Dynamic Provisioning for Performance and Power Management*, ser. Canadian theses. University of Toronto, 2008. [Online]. Available: http://books.google.com.br/books?id=pXBPOgAACAAJ

[7] F. R. C. Sousa and J. C. Machado, "Towards elastic multi-tenant database replication with quality of service," in *UCC '12*, 2012.

[8] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Sandpiper: Black-box and gray-box resource management for virtual machines," *Comput. Netw.*, vol. 53, no. 17, pp. 2923–2938, 2009.

[9] S. Vijayakumar, Q. Zhu, and G. Agrawal, in *CloudCom*. IEEE, 2010, pp. 441–448.

[10] J. Prevost, K. Nagothu, B. Kelley, and M. Jamshidi, "Prediction of cloud data center networks loads using stochastic and neural models," in *SoSE*, 2011, pp. 276 –281.

[11] W. Fang, Z. Lu, J. Wu, and Z. Cao, "Rpps: A novel resource prediction and provisioning scheme in cloud data center," in *SCC '12*, 2012, pp. 609–616.

[12] O. Niehoerster and A. Brinkmann, "Autonomic resource management handling delayed configuration effects," in *CloudCom*, 2011, pp. 138 –145.

[13] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in *IEEE CLOUD*, july 2011, pp. 500 –507.

[14] A. P. Witkin, "Scale-space filtering," in *IJCAI'83 - Volume 2*, 1983, pp. 1019–1022.

[15] M. Sato, T. Wada, and H. Kawarada, "A hierarchical representation of random waveforms by scale-space filtering," in *IEEE ICASSP '87.*, vol. 12, apr 1987, pp. 273 – 276.

[16] T. M. Sezgin and R. Davis, "Scale-space based feature point detection for digital ink," in *ACM SIGGRAPH '06 Courses*, 2006.

[17] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[18] G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Control*, ser. Wiley Series in Probability and Statistics. Wiley, 2008.

[19] B. Guenter, N. Jain, and C. Williams, in *INFOCOM*, 2011, pp. 1332–1340.

[20] J. M. Tirado, D. Higuero, F. Isaila, and J. Carretero, "Predictive Data Grouping and Placement for Cloud-Based Elastic Server Infrastructures," in *CCGrid*, 2011, pp. 285–294.

[21] A. M. Matsunaga and J. A. B. Fortes, "On the use of machine learning to predict the time and resources consumed by applications." in *CCGrid*, 2010, pp. 495–504.

[22] T. Lindeberg, *Scale-Space Theory in Computer Vision*. Norwell, MA, USA: Kluwer Academic Publishers, 1994.

[23] L. Wu and Z. Xie, "Scaling theorems for zero-crossings," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 1, pp. 46 –54, jan 1990.

[24] R. L. Allen and D. Mills, *Signal Analysis: Time, Frequency, Scale, and Structure*. Wiley-IEEE Press, 2004. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0471234419

[25] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, "Enhancing data analysis with noise removal," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 3, pp. 304 – 319, march 2006.

[26] G. Phillips, *Interpolation and Approximation by Polynomials*, ser. CMS Books in Mathematics. Springer, 2011. [Online]. Available: http://books.google.com.br/books?id=UrwCkgAACAAJ

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[28] B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, and T. Wood, "Agile dynamic provisioning of multi-tier internet applications," *ACM Trans. Auton. Adapt. Syst.*, vol. 3, no. 1, pp. 1:1–1:39, Mar. 2008.

[29] A. d. Monteiro, "Interest rate curve estimation: a financial application for support vector regression," Princeton University, Tech. Rep., 2004.

[30] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Comp. Syst.*, vol. 28, no. 1, pp. 155–162, 2012.