Modeling and predicting dynamics of heterogeneous workloads for cloud environments

Maria Carla Calzarossa*, Marco L. Della Vedova†, Luisa Massari*, Giuseppe Nebbione*, Daniele Tessera†
*Dept. of Electrical, Computer and Biomedical Engineering
Università degli Studi di Pavia
Via Ferrata 5, I-27100 Pavia, Italy
{mcc, luisa.massari}@unipv.it, giuseppe.nebbione01@ateneopv.it
†Dept. of Mathematics and Physics
Università Cattolica del Sacro Cuore
Via Musei 41, I-25121 Brescia, Italy
{marco.dellavedova, daniele.tessera}@unicatt.it

***Abstract*—The services and applications deployed nowadays in cloud environments are characterized by variable intensity and resource requirements. The variability of these workloads coupled with their heterogeneity affects the cost associated with the cloud infrastructure and the performance levels that can be satisfied. In these complex scenarios, resource provisioning policies have to take into account the actual workloads being processed and pro-actively anticipate in a timely manner the changes in workload intensity and characteristics. To support this decision process, we propose an integrated approach – that combines various workload characterization techniques – for modeling and predicting workload access patterns. The application of this approach has shown the importance of identifying models that specifically capture and reproduce the dynamics of these patterns and consider at the same time their peculiarities.**

***Index Terms*—predictive models, workload characterization, clustering techniques, time series analysis, cloud computing.**

I. Introduction

The performance of the services and applications being deployed nowadays on the Internet is affected by a mix of technological, sociological and psychological factors, including, among the others, the interactions of the users with the services, the number and mix of the service requests being concurrently processed as well as the characteristics of the cloud infrastructure [1].

Since most services are on demand, their usage patterns vary – depending on the time of the day and the day of the week – and experience the so-called "network effect". These patterns are also influenced by the pervasive use of mobile devices and by the presence of a large variety of bots and IoT devices that automatically and continuously generate requests. Moreover, despite the flexibility and elasticity offered by cloud environments, the use of virtualized resources and the co-location of heterogeneous workloads on a physical machine might cause contentions and performance fluctuations [2], [3].

In these complex and uncertain scenarios, it is quite challenging to choose effective resource allocation plans, that is, to provision the amount of resources of minimum cost able to satisfy the SLA requirements and the desired performance levels [4]. More precisely, decisions about provisioning and management of cloud and data center resources have to take into account the actual workloads being processed and to pro-actively anticipate in a timely manner the changes in workload intensity and characteristics. In fact, to better exploit their cloud resources, service providers deploy heterogeneous workloads with different SLA requirements (e.g., time sensitive applications, data and I/O intensive analytics tasks, computation intensive tasks). Therefore, the dynamic decision process has to be supported by accurate models based on the analysis of historical workload data.

Although workload has been extensively studied in the past, in recent years it did not receive the attention it deserves especially in the framework of predictions for interactive cloud services. In this paper, we study the workload with the objective of modeling and forecasting the access patterns of service requests, that is, when requests will be issued and how many will be issued at the same time. For this purpose, we propose an integrated approach based on a combined application of various workload characterization and time series analysis and forecasting techniques. As an application of this approach, we analyze properly anonymized HTTP trace files of the official web site of the University of Pavia. In particular, starting from some data analytics, we investigate the main characteristics of the requests processed by the server and their similarities. We then identify models that capture and reproduce the dynamics of the temporal patterns of these requests and consider at the same time their peculiarities.

The main contributions of this work are summarized as:

- integrated approach for modeling and predicting workload access patterns to be used for resource provisioning in cloud environments;
- identification of multiclass models of heterogeneous workloads;
- application of the proposed approach to real workload data.

The layout of the paper is as follows. Section II briefly discusses the state of the art in the domain of workload forecasting for cloud environments. The workload prediction approach is described in Section III. An application of the proposed approach is presented in Section IV. Finally, Section V summarizes the paper and outlines some future research directions.

II. Related work

Several papers acknowledge that satisfying the desired performance levels with a cost-effective amount of resources is very challenging because cloud workloads are quite dynamic and might even experience sudden variations over time [5]. Hence, to cope with these challenges, accurate predictions of workloads and of their resource demands are compelling.

Workload predictions have been addressed under different perspectives. Some papers focus on time series analysis and forecasting, while some others rely on machine learning techniques. Roy et al. [6] estimate the incoming workload of a system for future time periods by means of an Autoregressive Moving Average model that takes into account the workload patterns up to the current time period. Similarly, Calheiros et al. [7] address the prediction of workloads characterized by a seasonal behavior using an Autoregressive Integrated Moving Average model. The model – based on historical workload data – is updated on the run by applying feedback from latest observed loads. The predicted load is then used to dynamically provision cloud resources.

Islam et al. [8] propose an evolutionary approach for developing models able to make predictions to be used for adaptive resource provisioning in the cloud. In particular, by applying machine learning algorithms (i.e., error correction neural network and linear regression) and the sliding window technique, they show the superior prediction of the neural network models with an optimal window size (i.e., the Mean Absolute Percentage Error is 0.195 compared to 0.364 for linear regression).

Yang et al. [9] present a cost-aware auto-scaling approach that integrates a workload predictor based on a simple linear regression model, while in [10] the predictions rely on a recurrent neural network model.

In the framework of data center workload forecasting, Kumar and Singh [11] propose a prediction approach based on three layer neural network trained using a self adaptive differential evolution algorithm able to explore the solution space in multiple directions. A cyclic window learning approach is applied in [12] to predict the probability distribution parameters of the number of task arrivals to a data center during every predetermined period, while a Seasonal ARIMA model is proposed in [13].

Starting from the idea that workloads exhibit different change patterns, Liu et al. [14] present an adaptive approach for workload forecasting. According to this approach, different models – based on linear regression and support vector machine – are associated with different workload classes. The Mean Relative Percentage Error of this approach (i.e., 0.4677) is lower than errors obtained with other methods (e.g., linear regression, ARIMA).

Similarly to this work, we subdivide the workload in classes according to its characteristics and we study the behavior of the arrival patterns of each class separately. In particular, we represent the workload arrivals as a time series, whose analysis allows us to identify models to be used for predictions. These models are also very useful in all studies, e.g., simulation, that require the definition of multiclass synthetic workloads.

III. Workload prediction approach

As already pointed out, workload models represent the basis for implementing provisioning policies that dynamically adapt cloud resources to workload changes. In this section we present an integrated approach to identify models able to accurately capture and reproduce workload dynamics and at the same time take into account the workload peculiarities.

The proposed approach – whose workflow is summarized in Figure 1 – includes several steps aimed at discovering the properties of the workload data and identifying the corresponding models. In particular, the preprocessing of historical workload data (i.e., measurements collected on the infrastructure under investigation) provides some preliminary insights into their behavior and statistical properties. More precisely, the exploratory analysis of the parameters describing each workload component – coupled with the detection of potential outliers – works well for this purpose.

The exploratory analysis is based on the application of statistical and visualization techniques. Outlier detection relies on the computation of the percentiles of the parameter distributions. We note that outliers refer to the workload components characterized by an "anomalous" behavior with respect to one or multiple parameters. Therefore, once identified, outliers are usually removed not to perturb the following analysis.

To further investigate and summarize the behavior of the workload, the proposed approach focuses on the identification of classes of components with similar properties. This step is very important especially in the case of heterogeneous workloads in that it allows the individual characteristics of these workloads to be specifically taken into account and represented into these classes.

In detail, cluster analysis applied to the principal components – obtained as a result of the Principal Component Analysis (PCA) – provides these classes. Let us remark that the PCA application is advisable to remove potential correlations among parameters and reduce data dimensionality.

A fundamental step of the proposed approach deals with time series analysis aimed at formulating models of workload dynamics and predicting their future behavior. Although the techniques for time series analysis and forecasting are well defined, their application is not usually straightforward and requires particular care [15], [16].

More specifically, a preliminary visual inspection of the temporal behavior of the time series highlights recognizable patterns (e.g., trend, seasonality). The trend refers to slowly varying patterns over quite long periods of time, while the seasonality corresponds to a behavior that repeats over short periods of time (e.g., day, week, month, year).

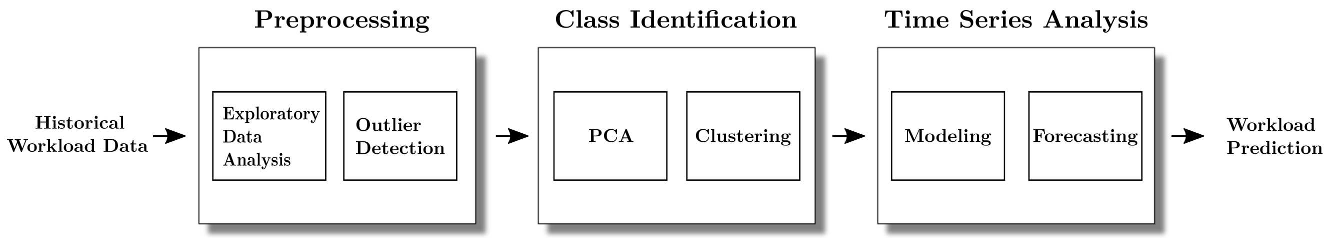


Fig. 1: Workflow of the proposed approach.

Therefore, to analytically detect periodic behaviors spectral analysis is applied because it represents the data in the frequency domain.

To better understand the properties of the time series, exploring its behavior over time and improve forecasts, a classical decomposition based on an additive model – which includes deterministic and stochastic components – is used [17]. This type of decomposition is particularly useful because the models of the deterministic components (i.e., trend and seasonal) can be associated with long-term planning of cloud resources, while the model of the stochastic component (i.e., the remainder of the time series) can be associated with short-term planning (see Figure 2).

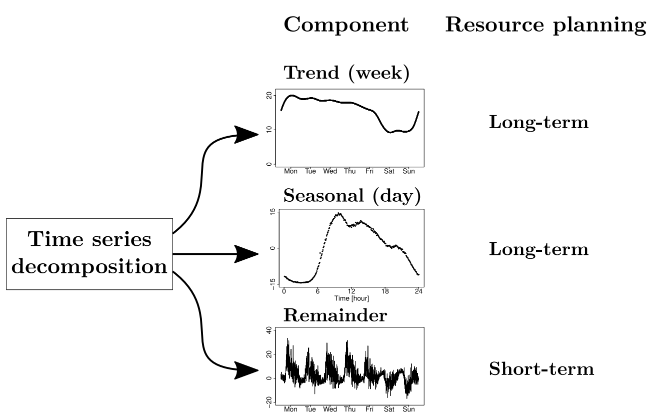


Fig. 2: Decomposition of the time series into its underlying components.

The estimation of the deterministic components is obtained by fitting appropriate models to the data, while techniques, such as moving average, auto regressive, Holt-Winters, Box and Jenkins [18], are applied to estimate the stochastic component.

These models are the basis for making forecasts. For the deterministic components, future values are obtained by extrapolation, while approaches, such as Box and Jenkins, are applied for predicting the stochastic component.

IV. Experimental results

In this section we describe the application of the proposed approach to analyze properly anonymized HTTP trace files collected on the web server hosting the official web site of the University of Pavia.

A. Dataset description

The trace files – graciously provided for this study – account for 33 GB of workload data that refer to more than 124M requests received and processed by the web server in one year. The traffic generated by the server for the responses of these requests accounts for 7.5 TB (see Table I for details).

TABLE I: Characteristics of workload data collected by the web server hosting the official web site of the University of Pavia.

Measurement Period	16 July 2017 – 17 July 2018
Trace file size	33 GB
Number of requests	124,168,736
Unique IP addresses	1,994,015
Unique User-Agent strings	106,296
Total traffic	7.569 TB

To accurately characterize and model this workload we introduce the concepts of user and session. In particular, a "user" is defined by the pair consisting of the anonymized IP address and User-Agent string – identifying the software agent – stored in the trace files and associated with each request. Although a pair could correspond to multiple users behind proxy servers, we believe this is a good approximation to distinguish users. Of course, a user accessing the web site with different devices in general will correspond to multiple pairs.

Moreover, we define a session as the set of requests issued by a user and characterized by an interarrival time (i.e., the time between consecutive requests) within a predefined threshold (set to five minutes in our study).

B. Data analytics

Out of the 2M unique IP addresses and slightly more than 100,000 unique User-Agent strings (see Table I), we identify about 3.3M users – either humans or automated software agents, such as web robots or watchdogs – and 9.8M sessions. In particular, 35% of the sessions are "singleton" consisting of one request only and there is another 32% of sessions with a number of requests ranging from two to nine. All these sessions account for only 13.5% (i.e., 16M) of the requests processed by the server over the entire measurement period.

Moreover, only about one fourth of the 3.3M users revisit the web site (i.e., issue more than one session).

In what follows we focus on sessions – independently of the user – with at least ten requests each, that is, 3,181,164 sessions. In fact, these sessions – that account for most of the requests and the majority of the users – heavily influence server performance. Moreover, it might be difficult to rely on "short" sessions for making predictions about resources to be provisioned.

To obtain some preliminary insights on the properties of the sessions we describe each session in terms of parameters that quantify its load on the server, namely, number of requests, session duration, average and standard deviation of the request interarrival times, traffic generated per session and per request. Table II summarizes basic statistics of some of these parameters. We notice a significant heterogeneity among sessions. In general, the standard deviation is at least one order magnitude larger than the corresponding average. Moreover, some sessions (i.e., 4%) are characterized by both duration and interarrival times equal to zero. This may be due to the one-second temporal resolution used in the trace files coupled with the structural properties of the web site – whose pages consist of many embedded objects stored on the same server.

TABLE II: Main properties of the user sessions.

Parameter	Average	St. Dev.	Min.	Max.
Number of requests	34.3	474.7	10	701,077
Duration [s]	171.0	13,935.7	0	21,035,250
Avg. interarrival time [s]	4.5	13.4	0	289
Session traffic [MB]	1.9	25.6	0	13,870

Additionally, to investigate the relationships among the parameters describing each session, we compute their correlations. As expected, the session duration is highly correlated (i.e., 0.98) with the number of requests in a session, thus in what follows we consider only one of these two parameters (i.e., number of requests).

A closer look at the distributions of the parameters suggests the presence of sessions that can be considered as outliers with respect to number of requests and traffic per session and per request. By trimming their distributions at the 99th percentile, we identify and remove from the dataset 65,752 outliers (corresponding to 0.2% of the sessions).

The temporal behavior of the session arrival process is another important aspect considered to investigate the overall properties of the workload. Figure 3 shows the number of sessions initiated every five minutes over two weeks. We clearly distinguish weekdays characterized by a much larger number of arrivals, from weekend days. Moreover, we notice a typical diurnal pattern, with a smaller number of new sessions over night than during the day. In particular, in the morning hours of weekdays there are up to 390 sessions in five minutes, while in weekend days only 188. On the contrary, night hours are characterized by much fewer sessions (i.e., as few as 15 in five minutes) independently of the day of the week. In addition, there are some spikes that denote sudden and brief increases in the number of new sessions. This is for example the case of the spike occurring on Sunday Oct. 15 at 8:30pm, when the number of sessions goes in five minutes from 83 to 188.

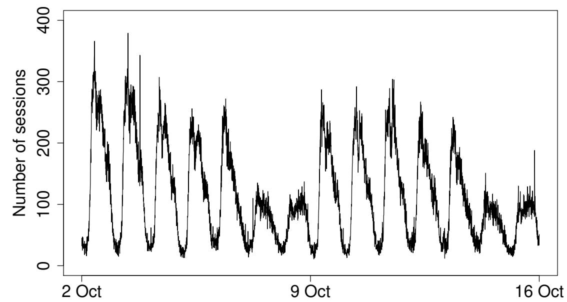


Fig. 3: Number of sessions initiated every five minutes over two weeks starting Monday, October 2, 2017.

Figure 4 shows the behavior of the session arrival process over the entire measurement period. The daily and weekly patterns have been removed by computing the moving average with a window size of one week. We notice a peak of new sessions in September, corresponding to the beginning of the academic year, and two periods characterized by very few sessions, corresponding to summer and Christmas holidays, respectively.

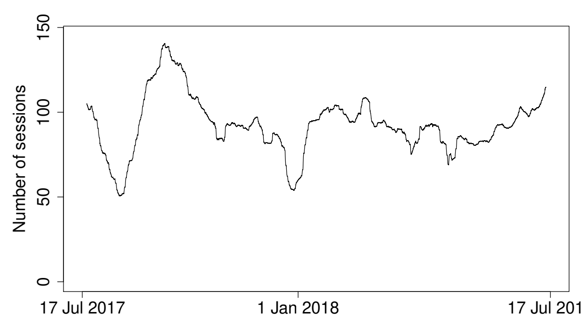


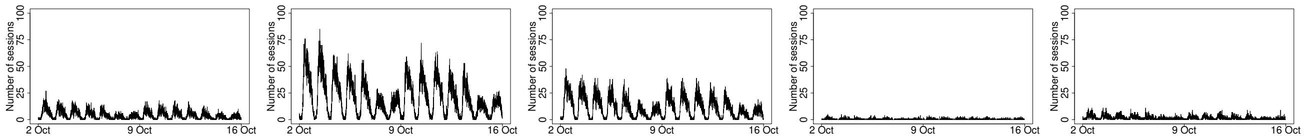
Fig. 4: Behavior of the session arrival process over one year.

In summary, data analytics suggests the need to properly take into account the variability and heterogeneity of the session characteristics and of their arrival patterns.

C. Class identification

To identify classes of sessions with similar characteristics, we apply clustering techniques, namely, the Lloyd's implementation of the k-means algorithm [19], [20]. The sum of squared distances within clusters is the metric used to assess the optimal number of clusters.

Clustering is applied to three out of the five principal components obtained as a result of the application of the PCA to the parameters describing the sessions (see Sect. IV-B).



Class 1 Class 2 Class 3 Class 4 Class 5

Fig. 5: Breakdown into five classes of the session arrivals shown in Figure 3.

Note that these three principal components explain a very large fraction of the total variance of the data, i.e., 94%.

As a result of the combined application of PCA and clustering, we obtain an optimal partition of the sessions into five classes. The centroids of these classes – in the space of session parameters – are presented in Table III. As can

TABLE III: Centroids in the space of the session parameters and composition of the five classes of sessions.

Class ID	Number of requests	Interarrival time avg. [s]	Interarrival time st. dev. [s]	Traffic per request [MB]	Traffic per session [MB]	Session %
1	59.5	5.5	20.0	0.036	2.14	10
2	23.9	1.8	5.6	0.022	0.54	55
3	34.1	1.4	4.9	0.065	2.08	30
4	31.7	13.8	24.3	0.304	9.12	1
5	16.7	54.2	68.6	0.060	0.99	4

be seen, these classes nicely capture the heterogeneity of the sessions. For example, class 1 groups sessions with the largest number of requests whose average interarrival time is equal to 5.5 seconds. On the contrary, sessions belonging to class 5 have a much smaller number of requests but their average interarrival time is an order of magnitude larger. Moreover, 55% of the sessions are grouped in one class, i.e., class 2. These are "light" sessions in terms of traffic per session and per request.

Another interesting result of the analysis of class composition refers to the identification of sessions associated with automated software agents, i.e., web robots. This analysis reveals that the majority of web robot sessions – identified by means of their User-Agent string – are grouped into two classes, namely, 4 and 5. This result can be very useful to assign different service levels to different workload classes.

D. Time Series Analysis

The classes previously identified are the basis for modeling and predicting the arrival process of the sessions. The breakdown into the five classes of the patterns of Figure 3 is shown in Figure 5. As this figure suggests, even though in general the arrival process exhibits regular weekly and diurnal behaviors, the number of new sessions and their patterns are quite different, thus requiring individual models.

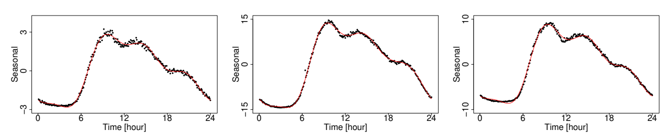
To formulate models of the session dynamics and predict their behavior, we study the arrival process of each class in the domain of time series analysis. More precisely, we split the measurement period in two sub-periods, namely, we use the data of the first 44 weeks for identifying the models and the data of the remaining eight weeks for validating the forecasting accuracy of the models.

As expected, spectral analysis highlights daily and weekly periodic patterns for all classes. On the contrary, the additive time series decomposition – into seasonal (i.e., daily pattern), periodic trend (i.e., weekly pattern) and irregular (i.e., the time series remainder) components – suggests that classes are characterized by different behaviors.

In detail, the time series decomposition clearly identifies daily and weekly periodic patterns for deterministic components of the first three classes, whereas for classes 4 and 5 the contribution of deterministic components is negligible. Therefore, to avoid overfitting, we directly describe the arrival process of these two classes by ARIMA models of $(1, 1, 1)$ and $(0, 1, 1)$ orders, respectively (see Fig. 8).

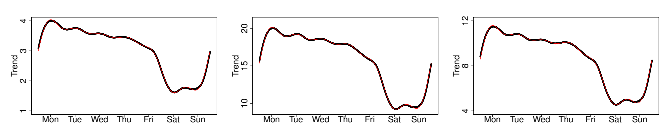
To model the daily and weekly components for the first three classes we use trigonometric polynomials. We identify their optimal degree and parameters by applying numerical fitting techniques coupled with goodness of fit tests, analysis of variance and backward stepwise regression techniques.

Figures 6 and 7 show the deterministic components of the time series and their models. The seasonal components of



Class 1 Class 2 Class 3

Fig. 6: Seasonal (daily) components of the time series representing the arrival process of three classes (dots) and corresponding models (solid lines).



Class 1 Class 2 Class 3

Fig. 7: Trend (weekly) components of the time series representing the arrival process of three classes (dots) and corresponding models (solid lines).

these classes are modeled by trigonometric polynomials of degree 4, 5 and 4, respectively. On the other hand, trigonometric polynomials of degree 6, 6 and 5 best fit the trend components.

Once the models of the deterministic components have been identified, we investigate the properties (e.g., stationarity, autocorrelation) of the stochastic component corresponding to the time series remainder. In particular, we use ARIMA models in that they allow us to accurately capture and forecast time series behaviors. Therefore, we describe the remainders of the three classes by ARIMA models with $(1, 1, 1)$, $(0, 1, 1)$, and $(0, 1, 1)$ orders, respectively. Finally, we compute the overall model of the session arrival process for each class as the sum of the models previously identified for the deterministic and stochastic components.

Figure 8 depicts the arrival process together with the corresponding model and the one-step ahead forecasts for all five classes. As can be seen, our models nicely cope with the arrival process of the various classes. More specifically, we compute the Mean Absolute Error (MAE) to assess the accuracy of the models and the corresponding forecasts. As Table IV suggests, the accuracy of the forecasts is good and the corresponding errors are comparable with the errors introduced in the models.

TABLE IV: Accuracy of models and forecasts in terms of MAE.

	Class 1	Class 2	Class 3	Class 4	Class 5
Model	1.33	3.26	2.29	0.39	0.88
Forecast	1.33	3.49	2.38	0.31	0.79

In summary, these results show the benefits of explicitly considering the workload heterogeneity in the models.

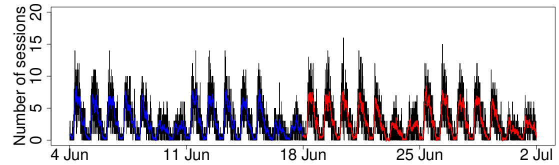
V. Conclusion

Workloads deployed in cloud environments are typically characterized by variable intensity and resource requirements. The variability of these workloads and their heterogeneity affect the performance perceived by the users and makes resource provisioning decisions rather challenging. Hence, to cope with these complex and uncertain scenarios, these decisions have to rely on accurate workload models.

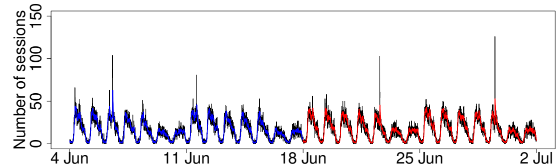
In this paper, we presented an integrated approach for identifying models able to capture and reproduce workload dynamics and at the same time take account of the workload peculiarities. The approach is based on a combined application of various workload characterization techniques. In particular, clustering techniques summarize the behavior of the workload and identify classes of homogeneous components. Time series analysis and forecasting identify models of the workload dynamics and predict its future behavior. The application of the proposed approach to anonymized HTTP trace files confirmed the heterogeneity of this workload and the importance of modeling the arrival process of each class separately. In addition, in case of significant changes in the workload dynamics (e.g., structural breaks) it might be necessary to refit the models.

We outline that the proposed approach is general enough to be applied for modeling and predicting the dynamics of any type of workload and it can be used in all studies that require the definition of multiclass synthetic workloads.

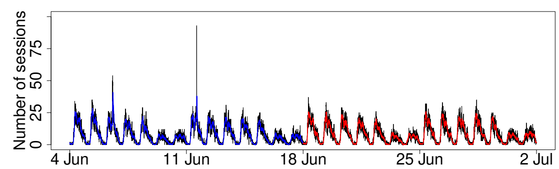
As future research activities, we plan to investigate the benefits of the proposed approach in proactive autoscaling policies for cloud environments. Moreover, we will apply our approach to measurements collected in different scenarios and application domains (e.g., IoT, social media).



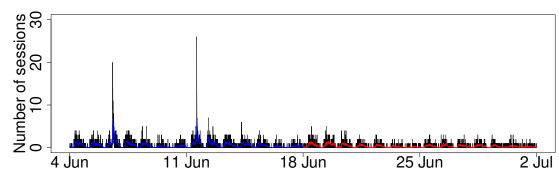
Class 1



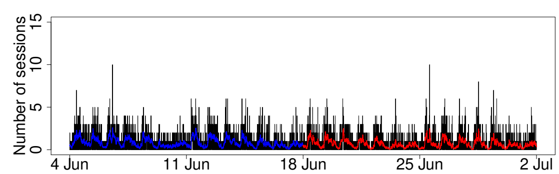
Class 2



Class 3



Class 4



Class 5

Fig. 8: Arrival process over four weeks (black), model (blue) and one-step ahead forecasts (red) for the five classes identified from historical workload data.

REFERENCES

[1] M. Calzarossa, L. Massari, and D. Tessera, "Workload characterization: A survey revisited," *ACM Computing Surveys*, vol. 48, no. 3, pp. 48:1–48:43, 2016.

[2] N. Herbst, A. Bauer, S. Kounev, G. Oikonomou, E. V. Eyk, G. Kousiouris, A. Evangelinou, R. Krebs, T. Brecht, C. L. Abad, and A. Iosup, "Quantifying cloud performance and dependability: Taxonomy, metric design, and emerging challenges," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 3, no. 4, pp. 19:1–19:36, 2018.

[3] S. Ristov, R. Mathá, and R. Prodan, "Analysing the Performance Instability Correlation with Various Workflow and Cloud Parameters," in *Proc. of 25th Euromicro Int. Conf. on Parallel, Distributed and Network-based Processing - PDP*, 2017, pp. 446–453.

[4] M. C. Calzarossa, M. L. Della Vedova, and D. Tessera, "A methodological framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty," *Future Generation Computer Systems*, vol. 93, pp. 212–223, 2019.

[5] M. C. Calzarossa, M. L. Della Vedova, L. Massari, D. Petcu, M. I. M. Tabash, and D. Tessera, "Workloads in the Clouds," in *Principles of Performance and Reliability Modeling and Evaluation*, ser. Springer Series in Reliability Engineering, L. Fiondella and A. Puliafito, Eds. Springer, 2016, pp. 525–550.

[6] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in *Proc. Int. Conf. on Cloud Computing*. IEEE, 2011, pp. 500–507.

[7] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS," *IEEE Trans. Cloud Computing*, vol. 3, no. 4, pp. 449–458, 2015.

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

[9] J. Yang, C. Liu, Y. Shang, B. Cheng, Z. Mao, C. Liu, L. Niu, and J. Chen, "A cost-aware auto-scaling approach using the workload prediction in service clouds," *Information Systems Frontiers*, vol. 16, no. 1, pp. 7–18, 2014.

[10] Y.-C. Chang, R.-S. Chang, and F.-W. Chuang, "A predictive method for workload forecasting in the cloud environment," in *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*, Y.-M. Huang, H.-C. Chao, D.-J. Deng, and J. J. J. H. Park, Eds. Springer, 2014, pp. 577–585.

[11] J. Kumar and A. K. Singh, "Workload prediction in cloud using artificial neural network and adaptive differential evolution," *Future Generation Computer Systems*, vol. 81, pp. 41–52, 2018.

[12] M. S. Yoon, A. E. Kamal, and Z. Zhu, "Adaptive data center activation with user request prediction," *Computer Networks*, vol. 122, pp. 191–204, 2017.

[13] V. G. Tran, V. Debusschere, and S. Bacha, "Hourly server workload forecasting up to 168 hours ahead using Seasonal ARIMA model," in *Proc. of the IEEE Int. Conf. on Industrial Technology*, 2012, pp. 1127–1131.

[14] C. Liu, C. Liu, Y. Shang, S. Chen, B. Cheng, and J. Chen, "An adaptive prediction approach based on workload pattern discrimination in the cloud," *Journal of Network and Computer Applications*, vol. 80, pp. 35–44, 2017.

[15] M. Calzarossa and D. Tessera, "Analysis and Forecasting of Web Content Dynamics," in *Proc. of the 32nd Int. Conf. on Advanced Information Networking and Applications Workshops*. IEEE Computer Society, 2018, pp. 12–17.

[16] M. Calzarossa, M. Della Vedova, L. Massari, G. Nebbione, and D. Tessera, "A methodological approach for time series analysis and forecasting of web dynamics," *Transactions on Computational Collective Intelligence*, 2019, (to appear).

[17] R. Cleveland, W. Cleveland, J. McRae, and I. Terpenning, "STL: A Seasonal-Trend Decomposition procedure based on Loess (with discussion)," *Journal of Official Statistics*, vol. 6, pp. 3–73, 1990.

[18] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Wiley, 2015.

[19] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[20] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Systems with Applications*, vol. 40, no. 1, pp. 200–210, 2013.