Analyzing Incoming Workload in Cloud Business Services

Nikola Tanković*, Nikola Bogunović†, Tihana Galinac Grbac‡, Mario Žagar†
*Superius d.o.o., Pula, Croatia
nikola.tankovic@superius.hr
†Faculty of Electrical Engineering and Computing, University of Zagreb
{nikola.bogunovic, mario.zagar}@fer.hr
‡Faculty of Engineering, University of Rijeka, Croatia
tihana.galinac@riteh.hr

***Abstract*—A recent trend, movement of software applications to Cloud, provides among numerous benefits, an important model for infrastructure cost reduction using the pay-as-you-go concept. In our experiments, we noticed that software distribution may significantly influence cost benefits achieved in Cloud. Software distribution optimization requires a continuous information influx on key metrics characterizing incoming workload. In this paper we propose a method for modeling workloads of business applications characterized by nonuniform distribution over the day.**

Two important properties are described: (1) modeling and forecasting repeatable patterns observed in the business context, and (2) modeling the inter-arrival time distribution of service requests. While former is important for constructing automated capacity planning controllers, latter is required for describing the amount of traffic variability. We analyzed these properties on a two-month workload collected from a production business services used by several thousand customers in retail domain in Croatia. Based on this analysis, we propose a high-level design of a quality of service controller applicable to business services in cloud environment.

I. Introduction

Cloud computing is a long-awaited solution in providing computing resources as a utility, such as electricity or water. It appearance elevated industrial and academic progress in supporting on-demand computation services in energy efficient and affordable pay-as-you-grow model [1]. While this is a promising establishment, transition is not to be expected effortless: existing software must be re-engineered in order to migrate on the cloud [2]. Such a transition demands a meticulous and often inconvenient development process due to distributed nature of cloud. One can observe that cloud computing technologies formulated solid foundations for using an almost unlimited amount of computing resources, and it is now up to the software engineering research and practice to come with efficient ways to best employ them.

This paper emphasizes the importance of understanding and correctly modeling the operational profile, or more precisely the incoming workload of a software service in the cloud. A proper model is required not just for improving capacity planning decisions and fine-tuning the load-balancing components [3], but to steer the software structure development decisions [4].

We will analyze the operational profile of an cloud business service used in retail domain. The data used for analysis was collected in a 20 day period, monitoring and recording every incoming request. Data was then analyzed revealing interesting characteristic patterns and distributions steering the design of proposed Quality of Service (QoS) controller.

The remainder of the paper is structured as following. Section II describes the domain context for this study. Important aspects and analysis of incoming workload is given in section III, and section IV gives and high-level proposal of and autonomic QoS controller capable to analyze incoming workload online. We conclude the paper in section V.

II. Business Services

Enterprises are struggling to adapt their business to information and pervasive computing era. Business applications governed by IT departments at large enterprises are now required to function efficiently in small and medium enterprises (SME). SMEs hardly afford such a costly endeavor in software engineering, but being a cloud computing customers, they can benefit from using software in a utility fashion. This work will refer to business applications provided as cloud services as *Business services*.

Definition: A Business service is an interactive business software application adapted for discovery, access and usage on a wide area network, such as Internet.

We will address business services designed on a client-server architecture in which the server part is a distributed,

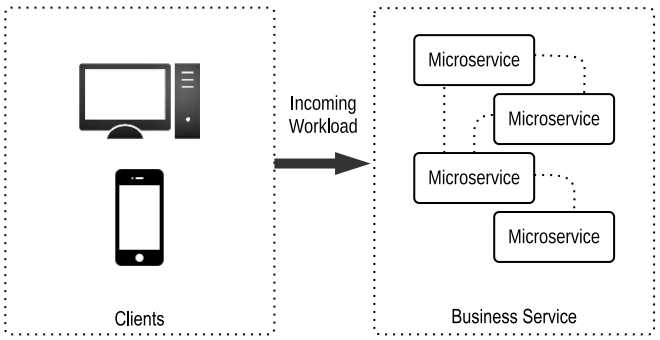

Fig. 1. Business Service as a composition of individual microservices.

interactive, and composable set of web services exposed to desktop, web, and mobile client applications (Fig. 1). In such architecture, service providers are focused around development and provisioning of web services - core building blocks of provided business service. Those services are referred to as Microservices: a small, independent and highly decoupled processes [5] focused on carrying a specific task. For instance, Amazon, Netflix, EBay, and SoundCloud, apply microservice architecture benefiting in scalable development process, easier maintenance and generally more resilient system [6]. The biggest challenge when maintaining such architecture is the decision on optimal boundaries and interfaces between microservices with regards to performance, network traffic amount, resource costs and development efforts. We demonstrated implications of such decisions in [4]. We also demonstrated that by slicing the existing service to many decoupled services introduces a more fine-grained approach to elasticity control, meaning that only the bottleneck services should be scaled when necessary.

III. Incoming Workload Analysis

An important step in building a business service is proper understanding of the amount and type of work it will conduct. Such insights are seldom available up front, meaning an iterative approach is required when building such services. Two important aspects for understanding incoming workload are: (1) workload intensity, and (2) workload distribution. Workload intensity represents the amount of work over the unit of time, and distribution explains the probabilities across different interarrival times: time passed between two consecutive requests.

A. Incoming Workload Definition

Each request submitted by a client encapsulates an individual usage of business service. Requests with indistinguishable resource demands are referred to as *request type* or *request class*. A resource demand measured in units of time or capacity is a measure of consumption of physical or virtual resources required for processing the individual request. We can then define term *workload* as the physical usage of the system across time consisting of series of requests or request classes served by the system. We also introduce the term *time series* (X) as a discrete function that represents measurements $x_i \in R$ for every time point t_i of equally distant time points $t = t_1, t_2, , t_n : X = x_1, x_2, , x_n$ in other words any finite or infinite sequence of observations $X_t : t \in T$ indexed by an order set T representing *time*. For example, a time series of request arrivals is a time series whose values represent $n_i \in N$ unique request arrivals in interval $[t_i, t_{i+1})$.

Since SMEs use their software mostly inside business hours it is expected that mean cloud resource demands will peak during that time window, and remain at lower levels outside. It is also presumable that time series of request arrivals will differentiate across the week, since weekend exhibit different working behaviors. Such patterns are important to take into considerations when designing business cloud software, e.g. for planning the time windows for management tasks.

B. Previous Work

The research in time series analysis is mainly concentrated toward fitting incoming workload data to a model used for forecasting future resource demands. Typical solutions are based on autoregressive (AR), integrated (I), and moving averages (MA) models, combined together in ARIMA family of models [7]. Since most of these methods tend be computationally expensive for a real-time usage in a cloud, numerous faster techniques are being developed instead. Gong et al. [8] apply a Fast Fourier Transform (FFT) analysis for finding patterns in workload on a wider time scale, and then apply discrete-time Markov chain to predict near future demand, like sudden spikes in incoming request rate. Roy et al. [3] apply autoregressive moving average method (ARMA) with a goal to minimize Service Level Agreement violations. Their work is mostly concentrated on large-scale websites. Saripalli et al. [9] use cubic spline interpolation technique in the first phase for predicting the trend, and in second phase a hotspot detection algorithm for forecasting sudden demand peaks. The limitation of their method is that it predicts the load not very far from last observed point. Herbst et al. [10] give an overview on available models and classify them according to computational cost, and preferred use case. They also introduce an online automatic adaptive process that combines most popular forecasting models according to data under analysis. In order to find patterns across different frequencies of time series, Kourentzes et al. [11] decompose data by aggregating it on different time frequencies and then apply best fitted model to each of them. Such an approach is computationally expensive, but can be desirable in domains where many repeating pattern reside.

C. Workload in Business Service Context

In a typical business service one can commonly observe repeatable patterns at weekly and daily levels. Real-time application imposes a requirement for applied method to be unobtrusive, computationally inexpensive, and powerful enough to model observed repeatable patterns. For that reason we apply an approach based on [11], but applied selectively only to several key time frequencies characteristic to SME business applications. Indications for such frequencies arrive from patterns observed on visualization of a weekly workload time-series (Fig. 3). Workload reduction outside business working hours is quite noticeable. The existence of such clear patterns motivates further analysis and application of possible algorithms from time-series analysis.

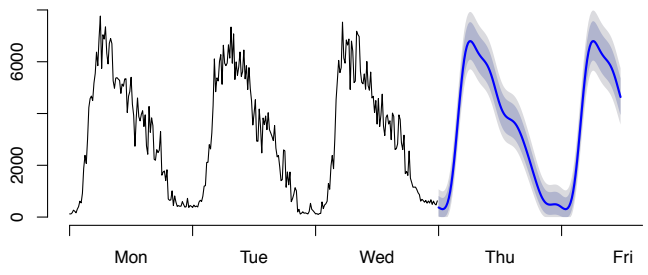

Fig. 2. Forecasting future load using tBATS method

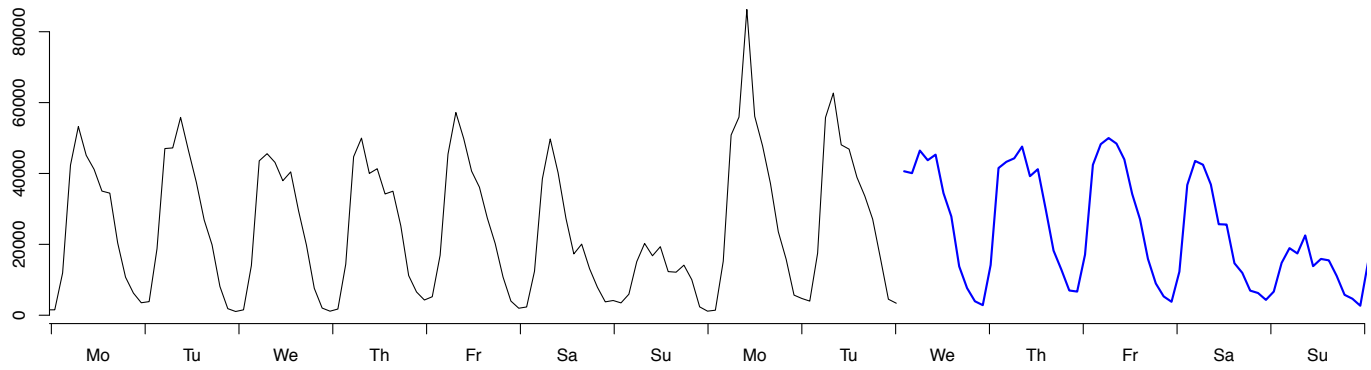

Fig. 3. Total weekly workload

D. Workload Forecasting

We analyzed the workload data using several well-known algorithms: (1) ARIMA, (2) tBATS, and (3) Artificial Neural Networks [12]. We concluded that applying tBATS model [13] and ANN yields positive results regarding speed and precision. ARIMA models required supplying additional parameters regarding orders of seasonal and non-seasonal parameters and applying integer optimization to locate best parameters using least-square method was the slowest method on our test machine.

A key feature of the tBATS model is that it relies on a new method that greatly reduces the computational burden in discovering complex seasonal patterns. Fig. 2 shows graphical results of next-day forecast based on previous three days.

ANN models also proved very efficient by using a Single hidden layer feed forward network that is the most widely used model form for time series modeling and forecast [14]. Their main advantage is faster modeling of non-linear relationships present in data. Fig. 3 show 3-day forecast based on 14-day data. It can be observed that weekend workload is successfully modeled by ANN forecast.

E. Workload Distribution

Another important aspect of incoming workload is the distribution of interarrival times between requests. For that purpose we constructed and analyzed probability distribution functions (PDFs) of incoming traffic. We observed fluctuations of probability distributions among different parts of day. Fig. 4 show the difference between incoming traffic distribution during the transition from non-working business hours to working business hours. We can observe the shift of tail towards higher values due to increase in incoming traffic. All distributions were closely fitted to *log-normal* distribution with different mean (μ) and standard deviation (σ) parameters. We say that a variable X is log-normally distributed when

$$X = e^{\mu+\sigma Z}$$

where Z is a standard normal variable. Parameter μ is also called the location of the distribution, and parameter σ the scale parameter. We constructed and log-normal distribution fit for every hour during the 24-hour period and shown the fluctuation of these parameters on Fig. 5.

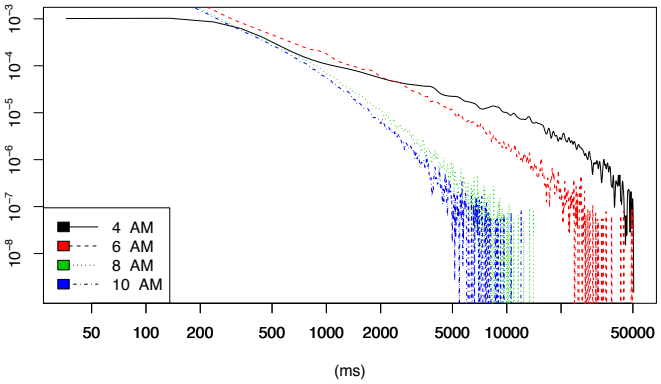

Fig. 4. Distribution (PDF) on a transition between low and high workload

This reveals an important aspect present in business service workload: a non-uniform distribution of incoming workload is present and it is dependent of working schedules of business service clients. In our scenario, the greatest shift in incoming distribution appears at morning when clients begin to use the service. Such transitions are important to model properly and employ them when testing elasticity attributes of business service.

We also constructed an probability distributions for most accessed microservices and discovered yet another difference between incoming workload based on transaction type. Fig. 6 show the difference between four most used services present in our case.

IV. Quality of Service Controller

In order to extract insight and knowledge from incoming traffic time-series and distributions we propose a QoS controller component for business services offered in cloud. Such a component should apply techniques described in previous chapter in online scenario, providing continuous information on incoming workload metrics and using it to maintain service levels. Fig. 7 displays and high-level overview of such solution. Main required components are:

- *Admission Control* - a component for load balancing between available web services placed on leased virtual machines from cloud provider. An solution

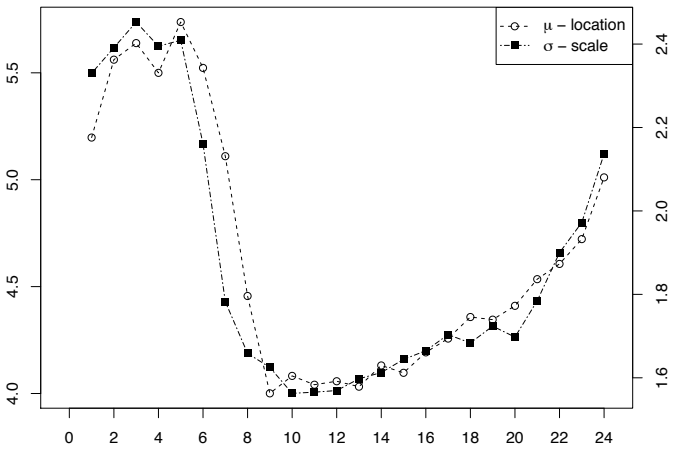

Fig. 5. Distribution variation across workday

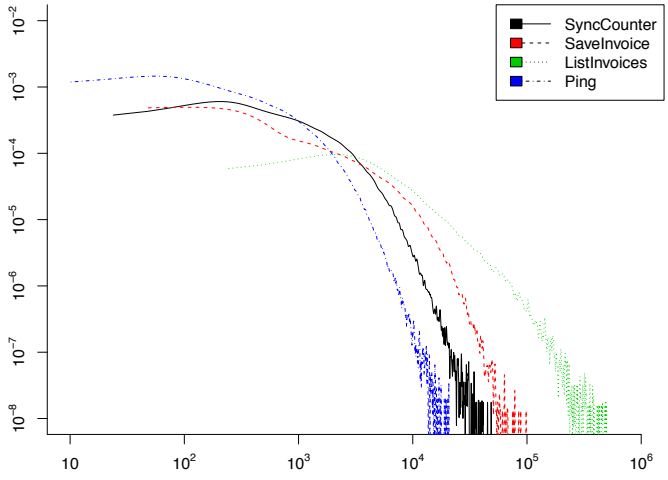

Fig. 6. Distribution (PDF) across transaction type during peak load for different transaction types

such as [15] should be adapted to deal with variable workload distributions as well as variable software deployment structure [4].

- *Workload Analysis and Forecasting* - we emphasize the usage of tBATS algorithm for forecasting daily workload, and using ANN approach for forecasting weekly workloads. By obtaining both predictions, this component updates the knowledge base with forecasts and analyzed distributions. Among incoming traffic, data should also be collected from individual web services so that knowledge can be mined by applying multiple time series analysis. For such a purpose a *Time Series Knowledge Representation* [16] should be implemented as it allows mining the temporal concepts of coincidence and partial order. It provides a speed-up from previously used Allen's Interval representations [17], the only hitherto available system for reasoning about temporal intervals both expressive and computationally effective.
- *Capacity and Deployment Planning* - based on collected knowledge about future workloads, this component will construct upfront schedule of capacity planning decision and control execution over leasing cloud providers infrastructure and deploying services. The main responsibility is to ensure enough cloud resources being employed for a given forecast to ensure a stable balance between infrastructure costs and quality of provided services. In order to deduct the necessary number of underlying infrastructure resources needed for current workload a model of system performance is required. We can obtain such model by mining temporal data from multiple time series: incoming workloadMapping the current request workload with amount of necessary resources can be seen as the bin-packing problem which is NP-hard [18] so research in this area is mainly consisting of approximate models and heuristics [19]. More specifically, an online version of the bin-packing problem is required, due to the fact that not all data is known up front [20]. A Best Fit heuristic is commonly used [15] to solve such problem. Control theory has also proven very powerful when dealing with uncertainty and disturbance by using feedback control [21].

Other components deal with more operational tasks such as communication with cloud provider over specialized APIs, and components monitoring. Projects such as Apache jclouds could be used to centralize communication with multiple cloud providers. For monitoring, a system such as MELA [22] or JCatascopia [23] is required, which supports specifying cloud service topologies and is capable to aggregate monitoring metrics across variable amount of infrastructure resources.

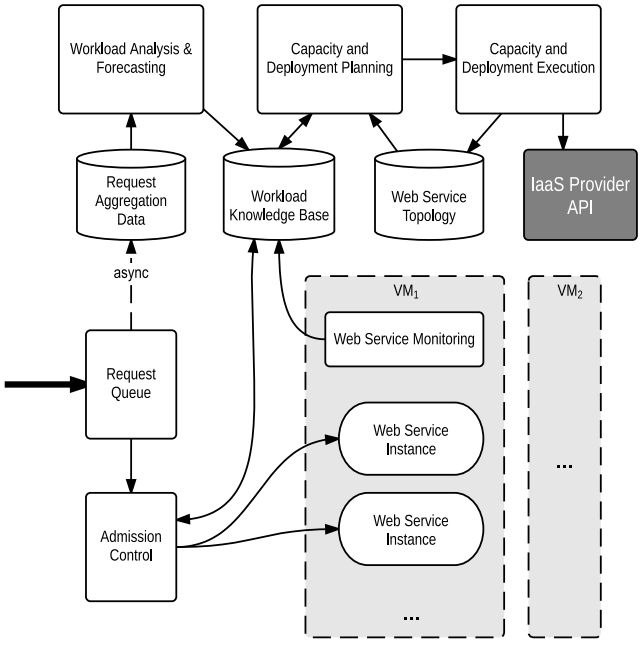

Fig. 7. A high-level overview on proposed mechanism for online analysis of incoming workload.

V. Conclusions

This paper exhibited the importance of proper incoming workload decomposition and analysis in the field of time-

series forecasting and distribution variability. We showed that in the context of business services, one can expect very high fluctuations in incoming workload intensity, typically revolving around business working hours. Furthermore, we showed that distributions of inter-arrival times in the presence of a larger number of customers tend to fit log-normal distribution with different location and scale parameters across the working day and transaction types.

In order to construct a knowledge base around incoming workload analysis we proposed an high-level definition of an QoS controller mechanism constructed for business services in cloud environment. Such a mechanism should be able to automatically predict future workload demands and adapt the software structure according to currently examined log-normal distributions from incoming traffic. We will concentrate our further work on a more detail description of such a mechanism together with a set of algorithms for such a controller.

VI. Acknowledgment

The work presented in this paper is supported by COST action 1304 Autonomous Control for a Reliable Internet of Services (ACROSS) and the research grant 13.09.2.2.16. from the University of Rijeka, Croatia.

References

[1] M. Armbrust, I. Stoica, M. Zaharia, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, p. 50, Apr. 2010. [Online]. Available: http://dl.acm.org/ft_gateway.cfm?id=1721672&type=html

[2] S. G. Saez, V. Andrikopoulos, F. Wessling, and C. C. Marquezan, "Cloud Adaptation and Application (Re-)Distribution: Bridging the Two Perspectives," in *2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*. IEEE, Sep. 2014, pp. 163–172. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6975357

[3] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," *Proceedings - 2011 IEEE 4th International Conference on Cloud Computing, CLOUD 2011*, pp. 500–507, 2011.

[4] N. Tanković, T. Galinac Grbac, H.-l. Truong, and S. Dustdar, "Transforming Vertical Web Applications Into Elastic Cloud Applications," in *International Conference on Cloud Engineering (IC2E 2015)*. IEEE, Mar. 2015, pp. 135–144. [Online]. Available: Accepted. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=7092911

[5] J. Thönes, "Microservices," *IEEE Software*, vol. 32, no. 1, pp. 116–116, Jan. 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7030212

[6] S. Newman, *Building Microservices*. "O'Reilly Media, Inc.", 2015. [Online]. Available: https://www.google.hr/books?hl=en&lr=&id=jjl4BgAAQBAJ&pgis=1

[7] C. Chatfield, *The Analysis of Time Series: An Introduction, Sixth Edition*. CRC Press, 2013, vol. 19. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=qKzyAbdaDFAC&pgis=1

[8] Z. Gong, X. Gu, and J. Wilkes, "PRESS: PRedictive Elastic reSource Scaling for cloud systems," *Proceedings of the 2010 International Conference on Network and Service Management, CNSM 2010*, no. Vm, pp. 9–16, 2010.

[9] P. Saripalli, G. V. R. Kiran, R. R. Shankar, H. Narware, and N. Bindal, "Load prediction and hot spot detection models for autonomic cloud computing," *Proceedings - 2011 4th IEEE International Conference on Utility and Cloud Computing, UCC 2011*, pp. 397–402, 2011.

[10] N. R. Herbst, N. Huber, S. Kounev, and E. Amrehn, "Self-Adaptive Workload Classification and Forecasting for Proactive Resource Provisioning," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 12, pp. 2053–2078, 2014.

[11] N. Kourentzes, F. Petropoulos, and J. R. Trapero, "Improving forecasting by estimating time series structural components across multiple frequencies," *International Journal of Forecasting*, vol. 30, no. 2, pp. 291–302, 2014.

[12] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[13] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, Dec. 2011. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1198/jasa.2011.tm09771 http://www.buseco.monash.edu.au/ebs/pubs/wpapers/2009/wp9-09.pdf

[14] G. P. Zhang, E. B. Patuwo, and H. Michael Y., "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169207097000447

[15] P. Leitner, W. Hummer, B. Satzger, C. Inzinger, and S. Dustdar, "Cost-efficient and application SLA-aware client side request scheduling in an infrastructure-as-a-service cloud," *Proceedings - 2012 IEEE 5th International Conference on Cloud Computing, CLOUD 2012*, pp. 213–220, 2012.

[16] F. Moerchen, "Algorithms for time series knowledge mining," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, no. 2. New York, New York, USA: ACM Press, Aug. 2006, p. 668. [Online]. Available: http://dl.acm.org/citation.cfm?id=1150402.1150485

[17] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[18] M. E. Frîncu, "Scheduling highly available applications on cloud environments," *Future Generation Computer Systems*, vol. 32, pp. 138–153, Mar. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X12001136

[19] L. Wu, S. Kumar Garg, and R. Buyya, "SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1280–1299, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jcss.2011.12.014

[20] M. P. Renault, A. Rosén, and R. van Stee, "Online Algorithms with Advice for Bin Packing and Scheduling Problems," p. 15, Nov. 2013. [Online]. Available: http://arxiv.org/abs/1311.7589

[21] M. Maggio, H. Hoffmann, M. D. Santambrogio, A. Agarwal, and A. Leva, "Decision making in autonomic computing systems," *8th ACM international conference on Autonomic computing*, p. 201, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1998629

[22] D. Moldovan and G. Copil, "Mela: Monitoring and analyzing elasticity of cloud services," in *International Conference on Cloud Computing*, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6753781

[23] D. Trihinas, G. Pallis, and M. D. Dikaiakos, "JCatascopia: Monitoring Elastically Adaptive Applications in the Cloud," *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 226–235, May 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6846458