User Behavior and Application Modeling in Decentralized Edge Cloud Infrastructures

John Violos[1(✉)], Vinicius Monteiro de Lira[2], Patrizio Dazzi[2], Jörn Altmann[3],
Baseem Al-Athwari[3], Antonia Schwichtenberg[4], Young-Woo Jung[5],
Theodora Varvarigou[1], and Konstantinos Tserpes[1,6]

[1] School of Electrical and Computer Engineering,
National Technical University of Athens, 15771 Zografou, Greece
violos@mail.ntua.gr,dora@telecom.ntua.gr

[2] Institute of Information Science and Technologies,
National Research Council of Italy (CNR), Pisa, Italy
{vinicius.monteirodelira,patrizio.dazzi}@isti.cnr.it

[3] Technology Management, Economics, and Policy Program, College of Engineering,
Seoul National University, Seoul, Republic of Korea
{altmann,baseem}@snu.ac.kr

[4] CAS Software AG, Karlsruhe, Germany
Antonia.Schwichtenberg@cas.de

[5] Cloud Computing Research Department,
Electronics and Telecommunications Research Institute (ETRI),
Daejeon, Republic of Korea
jungyw@etri.re.kr

[6] Department of Informatics and Telematcs,
Harokopio University of Athens, Tavros, Greece
tserpes@hua.gr

Abstract. Edge computing has emerged as a solution that can accommodate complex application requirements by shifting data and computation to infrastructure elements that are more suitable to manage them given the current circumstances. The BASMATI Knowledge Extractor is a component that facilitates the modeling of the resource utilization by providing tools to analyze application usage together with user behavior. This is particularly relevant in the case of mobile applications where user context and activity are tightly coupled to the application performance.

Keywords: Performance modeling · Resource utilization · User behavior modeling · Application usage modeling · BASMATI project

1 Introduction

The rate at which mobile applications expand to take over a large share in the software market is unprecedented. The supporting infrastructures are undergoing a continuous development in order to cope with the demanding setting. The advent of decentralized edge clouds has managed to stick out among the viable

C. Pham et al. (Eds.): GECON 2017, LNCS 10537, pp. 193–203, 2017.
DOI: 10.1007/978-3-319-68066-8_15

solutions and has been quickly adopted by application developers. This type of infrastructure assumes a dynamic orchestration of multiple computing and storage elements from which it may be comprised.

The BASMATI Knowledge Extractor (BKE) contributes towards this direction by providing decision support services to federated cloud brokers and job offloading managers. It does so by investigating the contribution of the user behavior (and in particular mobility since we are referring to mobile applications), of the application model, as well as their combination, to the resource utilization patterns.

This investigation is relying on data analysis for building predictive models for user behavior and application usage. The work suggests a number of possible approaches that yield different results under various conditions. The BKE can be seen as a toolkit that can contribute in the preservation of QoS through a more efficient resource utilization.

This work describes the design of such a component in the context of a federated, decentralized edge cloud platform supporting mobile applications. It continues with details about the data management and concludes with the tools that are to enable the knowledge acquisition.

2 Functionality and Architecture of the Knowledge Extractor

The general architecture of the BKE component is presented in Fig. 1. It is comprised of three knowledge acquisition subcomponents and an auxiliary data preprocessing subcomponent. The three knowledge acquisition components are the User Mobility Behavior, Application Usage Modeling and the Situation Knowledge Acquisition. These subcomponents rely on prediction techniques that are described in Sect. 4 regarding the knowledge acquisition. The preprocessing of data takes place in the Unified Representation subcomponent and uses data fusion and feature engineering techniques described in Sect. 3.

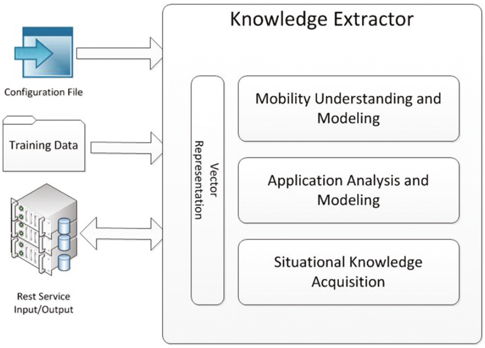

Knowledge Extractor
Configuration File
Training Data
Rest Service Input/Output
Vector Representation
Mobility Understanding and Modeling
Application Analysis and Modeling
Situational Knowledge Acquisition

Fig. 1. Knowledge extractor component.

Figure 2 depicts the formalization of the problem that resolves the BKE. Given the available data from various data resources, the BKE refines and fuses them in a unified representation upon which it then applies a prediction technique in combination with the Knowledge base to produce the outcome predictions. The outcome predictions can be defined in terms of user mobility or applications and sessions demands on resources. With the term "session" we refer to an application session running at a specific time for a specific user.

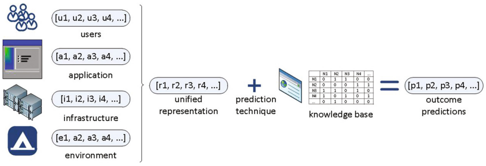


Fig. 2. Flow of logic of the knowledge extractor process.

2.1 User Mobility Behavior Modeling

The Mobility understanding and modeling is the subcomponent of the BKE that analyzes and predicts the behavior of the mobile app users. It is application-dependent and operates on the assumption that the user behavior affects the provision of application services and the utilization of the federated resources. For instance, for location-based mobile applications supported by decentralized edge cloud infrastructures, the resource utilization may vary based on the mobility patterns of the end users. As such, analysis of semantic trajectories (that is, trajectories enhanced with e.g. event metadata) may assist in dynamically balancing the load among the various infrastructure elements and thus preserving the QoS guarantees. The extraction of these trajectories requires the fusion and analysis of multiple data sources including textual, geo-spatial and other types of data.

The actions for a Basmati end user to adapt the mobility modeling to a new specific given application are the following two: firstly, a compatible dataset based on previous observations should be passed to the BKE in order to construct the representation knowledge base that will be used from the supervised machine learning techniques. Afterwards, in the configuration file should be declared which prediction technique will be used and the type of the input data.

2.2 Application Usage Modeling

Based on the application usage modeling, predictions related to resource demands came be made, this time from the application's perspective. Different applications pose different resource demands under the same load. These predictions in combination with the QoS should be examined by the resource broker for the optimal resource management and the avoidance of bottlenecks. Some of the resource demand parameters that are modeled are CPU, memory, bandwidth, average file size, application duration usage and time interval between two application requests.

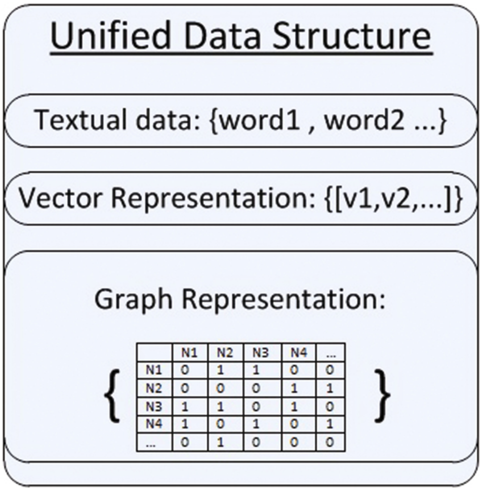
2.3 Situational Knowledge Acquisition

The Situational Knowledge Acquisition is the third main subcomponent of the BKE. This subcomponent makes predictions of the resource demands that are generated for sessions between the users and the applications.

2.4 Unified Representation

The unified representation introduces an intermediate layer between the sources of incoming data and the three abovementioned main BKE subcomponents. Its purpose is to unify and transform the input data in a form compatible and readable by the predictive algorithms.

Heterogeneous data sources will constantly feed the BKE with observations. They include human trajectories, web-based data, user contextual data, resource utilization per application, etc. Feature engineering techniques are applied to gauge the best feature representation of the observations. Then, the users behavior, application model and combinations of user-application can be represented with a composite data structure as illustrated in Fig. 3.


Unified Data Structure
Textual data: {word1 , word2 ...}
Vector Representation: {[v1,v2,...]}
Graph Representation:
{
}

	N1	N2	N3	N4	...
N1	0	1	1	0	0
N2	0	0	0	1	1
N3	1	1	0	1	0
N4	1	0	1	0	1
...	0	1	0	0	0

Fig. 3. Unified data structure.

Respectively each prediction model can process different kind of data such as textual data, vectors or graphs and retrieve them from the compatible field of the unified data structure.

2.5 Configuration

The BKE components can be used on demand, based on the particular application scenario requirements. Each case can have its own specific data as input and expect its own specific data as outcome predictions. The form of the input data, the expected outcomes, and the selection of the predictor techniques can be passed to the BKE through a configuration service. This configuration service orchestrates how the BKE works. Technically, it comprises a REST endpoint that receives the configurations in the form of an XML or JSON document, allowing for the implementation of a usable web interface for its creation.

2.6 Training Data

According to the supervised machine learning techniques, the three main subcomponents use training data to build their internal knowledge representations. Initially these training data can be provided by files in the hosted servers of the components.

2.7 Input/Output

The input and the output of the BKE are provided and stored in a local relational database, accessed through a RESTful API. Similarly, user context and application data can be persisted to the database through the API so they can then be transformed to the form expected by the unified data structure.

3 Data Preprocessing

The data preprocessing is carried out in order to fit the data to the unified representation structure's schema involving three consequent steps. The data fusion that gathers the data from the data sources performing operations to join them, the feature engineering that produce the most representative features and the normalization of the value features. The aim of the data preprocessing stage is to combine relevant information from various sources into a single structure that provides a more accurate and flexible description in contrast with the individual data sources.

The varying input instances are mapped in the unified representation structures. The knowledge base files are loaded and used by the prediction techniques in combination with the representation of the input instances so as to produce the requested predictions. There is no need to store and retrieve the input data in any kind of database. The unified representation subcomponent is responsible to handle the varying input data and the extra effort of a database should be avoided. Furthermore, the knowledge base files are stored in files such as JSON and arff. The prediction techniques do not need specific parts of the stored data and specific queries on them, but they need entire length of the knowledge base to carry out their processes (Fig. 4).

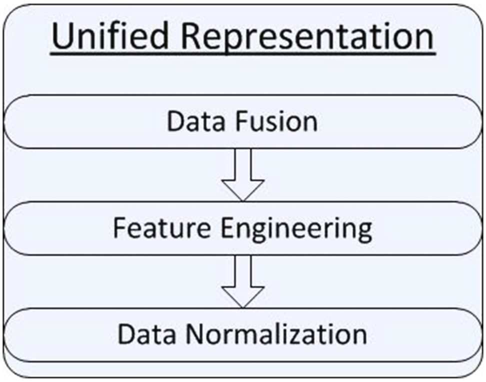


Fig. 4. Unified representation.

3.1 Data Fusion

The fusion of data is a very powerful tool and enriches the prediction methods. For the BKE component the data is comprised of user profile data and application, service and resource usage data. These data are integrated, using any of the two following models according to the frequency and the size of the data load from the data sources.

- The Multi-sensor integration fusion model [11] follows a hierarchical fashion to combine data in fusion centers. The first fusion center retrieves data from two peer sensors. The following fusion centers combine the data from a new data sensor with the output of the previous Fusion center. The last fusion center outputs the integrated data in a unified structure. This method is used in case of a heavy data workload.
- The second option for the data fusion from multiple sensors is the Behaviour knowledge-based model [12]. This model consists of a series of stages. The first stage retrieves the data from all the data sensors. In the next stage a feature vector is extracted from the retrieved data. The third stage associates a data structure to the predefined needs. In the last stage, a set of rules is applied according to formalism of the representation. This method will be used in case of a frequent data update.

3.2 Feature Engineering

The intrinsic structure of the available data and the needs of the end users may constitute a supervised learning problem that involves features that do not have a positive contribution to the method accuracy. The concept the more data the better results is not applied in the prediction methods and large amounts of data may produce low accuracy and performance in data analytic applications.

Two different set of techniques are used to mitigate the issue of the redundant features of the datasets: the feature extraction and the feature selection. Both of them reduce the data representation using fewer features. Features, attributes, variables, terms, dimensions are interchangeable notions for the needs of BKE. Feature extraction methods represent features in a new dimensional space making a fusion or a transformation of the features. On the contrary, feature selection methods do not transform the dimensions. They select the dimensions that contain more bits of information based on a certain objective function.

Feature Extraction. The Feature extraction methods introduce a new lower feature-dimensional space that combines the initial data features. The derived features should satisfy the following three properties. They should be informative and non-redundant, facilitate the machine learning and predictions methods in which they will be used, and in some case to provide a better human understanding of the problem. Two feature extraction techniques are presented below the technique that satisfy the aforementioned criteria in a better way will be implemented in the unified representation subcomponent.

A common dimensionality reduction method is the Principal Component Analysis (PCA) [8]. PCA is a statistical method that orthogonally transforms the original dimensions into a new component set of dimensions. The new set of dimensions is called principal components, it is smaller and it retains most of the information. The basic idea is to convert the correlated features into a new set of features that are linearly uncorrelated. The PCA is an iterative process. The first principal component should have the largest variance and the following components should have the highest variance under the restriction to be orthogonal to the previous components.

The Independent Component Analysis (ICA) [6] is a statistical and computation feature extraction method that detects the latent features that underline sets of random observations. ICA is based on a generative model for multivariate data. The instances can be linear or nonlinear mixtures of the unknown latent variables while how they are mixed is unknown. The latent variables that will be found by ICA are assumed to be non-Gaussian, linear and mutually independent and they are called the independent components of the observed data. The feature extraction process is not reversible because some information is lost in the process of transformation.

Feature Selection. Feature selection is the process of choosing a subset of features from a set of candidate features based on a statistical score such as variation of variable correlation. The features that will be selected are the most important and representative of the available features. These methods require a good understanding of the aspects of the prediction problem that have to be resolved.

The three main feature selection approaches are wrapper method, filter method and the embedded method. The wrapper and the embedded Feature selection techniques cannot be applied to the KE because of the computation demands and their inability to discriminate the feature selection stage with the prediction stage. On the other hand, the filter methods produce good results with low computation demands. Other filter techniques are Information gain, Chi-square, Mutual Information, Fisher score, Low variance criterion.

3.3 Data Normalization

The data sources may provide values in a different scale than the internal representation of the predefined knowledge of the BKE. A standardization and normalization process bridges this gap reproducing the feature values to lie between a specified minimum and a maximum value.

4 Knowledge Acquisition

The BKE predictions can be carried out by classification, clustering or regression methods. The decision of what prediction approach is used depends on the type and the amount of provided data and the parameters to be predicted which vary in each use case. The following proposed method can be enhanced to take into consideration the time evolution of the observations and the predicted parameters.

4.1 Natural Language Processing

Predictions based on the textual data can be carried out using a Graph representation model in combination with Graph similarity metrics. The Graph model has been used for classification [2] and clustering purposes [16] and it can be also applied for regression analysis. The graph model has been used with N-Grams being represented as nodes yet there is the option to use words instead. In the following description of the Graph model we use the term implying a word or N-gram.

4.2 Vector Processing

The classification, clustering and regression of vectors have been extensively researched and applied in many fields. We use two of the main vector prediction methods as the baseline models of the BKE: the Support Vector Machine (SVM) and Bayes Classification. The SVM [14] model represents the instances as points in an N-dimensional space. Then a planar is gauged to divide the instances that belong to different categories. The gap between instances that belong to different categories should be as wide as possible. The Gaussian Bayes classifier [7] uses a conditional probability model in which the values of the vector are the independent variables and the category is the dependent variable y.

4.3 Graph Partitioning

To identify categories of users or the correlation between users and applications a graph partition method can be applied. Typically the graph partitioning problems are NP-hard. Heuristic and approximation algorithms have been proposed that produce sufficient results. The KernighanLin algorithm [3] is a graph partition algorithm that performs well in a dense graph with less than 10000 nodes. It uses a technique that exchanges nodes between the partitions using the betweenness metric of internal and external cost. Girvan and Newman [5] inctroduced the K-Means partition algorithm. K-Means produce good results with the limitation that the observations should be linear. In this method each observation is assigned in the cluster with the nearest mean value.

4.4 Artificial Neural Network Method

Literacy research [10,15] suggests that another option that achieves the same goal is to employ artificial neural networks (ANN). ANN may provide an accurate model of the application deployment allowing to estimate the computational requirements of a given user.

5 Implementation of Knowledge Extractor

The implementation of the BKE took place using the Java programming language and it complies with the specifications of Open Cloud Computing Interface (OCCI) [4]. The BKE uses a variety of prediction techniques, some of them are implemented based on available java libraries such as Weka [17], deeplearning4java [1] and some others such as NgramsGraphs NLP classification and Markov chains developed from the researchers and developers of the Basmati project. In order to provide more prediction techniques that are based on the scikit-learn library [13] of the python programming language we have examined the use of wrappers such as Jython [9].

The BKE interfaces with the other components of Basmati through a restful API using JSON for data representation. The Decision Maker (DM) is the component of the Basmati platform that requests the predictions of the users mobility and the application resources. The Application Monitoring is the component that is responsible to store and retrieve the data that can be used to update the knowledge base.

6 Conclusions

Predicting the resource requirements in an edge cloud platform supporting mobile applications calls for the development of a complex model. This model maps user behavior and application usage to resource utilization. The BKE provides an architecture that facilitates a number of machine learning techniques that can be adapted to the given data and provide estimations of the resource utilization.

Acknowledgements. BASMATI (http://basmati.cloud) has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement no. 723131 and from ICT R&D program of Korean Ministry of Science, ICT and Future Planning no. R0115-16-0001.

References

1. Deeplearning4j: Open-source distributed deep learning for the jvm. https://deeplearning4j.org. Accessed 17 July 17
2. Aisopos, F., Tzannetos, D., Violos, J., Varvarigou, T.A.: Using n-gram graphs for sentiment analysis: an extended study on twitter. In: Second IEEE International Conference on Big Data Computing Service and Applications, BigDataService 2016, Oxford, United Kingdom, March 29 - April 1, 2016, pp. 44–51 (2016). http://dx.doi.org/10.1109/BigDataService.2016.13
3. Dutt, S.: New faster kernighan-lin-type graph-partitioning algorithms. In: Proceedings of 1993 International Conference on Computer Aided Design (ICCAD), pp. 370–377, November 1993
4. Edmonds, A., Metsch, T., Papaspyrou, A.: Open cloud computing interface in data management-related setups. In: Fiore, S., Aloisio, G. (eds.) Grid and Cloud Database Management, pp. 23–48. Springer, Heidelberg (2011). doi:10.1007/978-3-642-20045-8_2
5. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. **99**(12), 7821–7826 (2002)
6. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. Neural Netw. **13**(4–5), 411–430 (2000)
7. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, UAI 1995, pp. 338–345. Morgan Kaufmann Publishers Inc., San Francisco (1995)
8. Jolliffe, I.: Principal Component Analysis. Wiley, New York (2014)
9. Juneau, J., Baker, J., Wierzbicki, F., Muoz, L.S., Ng, V., Ng, A., Baker, D.L.: The Definitive Guide to Jython: Python for the Java Platform. Apress, Berkeley (2010)
10. Kousiouris, G., Menychtas, A., Kyriazis, D., Gogouvitis, S., Varvarigou, T.: Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in cloud platforms. Future Gener. Comput. Syst. **32**, 27–40 (2014)
11. Luo, R.C., Kay, M.G.: Multisensor integration and fusion in intelligent systems. IEEE Trans. Syst. Man Cybern. **19**(5), 901–931 (1989)
12. Pau, L.F.: Sensor data fusion. J. Intell. Robot. Syst. **1**(2), 103–116 (1988)
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
14. Singla, A., Patra, S., Bruzzone, L.: A novel classification technique based on progressive transductive SVM learning. Pattern Recogn. Lett. **42**, 101–106 (2014)
15. Tserpes, K., Kyriazis, D., Menychtas, A., Varvarigou, T.: A novel mechanism for provisioning of high-level quality of service information in grid environments. Eur. J. Oper. Res. **191**(3), 1113–1131 (2008)

16. Violos, J., Tserpes, K., Papaoikonomou, A., Kardara, M., Varvarigou, T.A.: Clustering documents using the 3-gram graph representation model. In: 18th Panhellenic Conference on Informatics, PCI 2014, Athens, Greece, October 2–4, 2014, pp. 29:1–29:5 (2014). http://doi.acm.org/10.1145/2645791.2645812
17. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: PractIcal Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2016)