




A Quality of Experience Illustrator User Interface for Cloud Provider Recommendations

Panagiotis Kokkinos^{1,2} , Dionisis Margarīs² , and Dimitris Spiliotopoulos³ 

¹ Institute of Communication and Computer Systems, NTUA, Athens, Greece
p.kokkinos@uop.gr

² Department of Digital Systems, University of the Peloponnese, Sparta, Greece
margaris@uop.gr

³ Department of Management Science and Technology, University of the Peloponnese, Tripoli, Greece
dspiliot@uop.gr

Abstract. Cloud infrastructures handle processing and storage options for a multitude of applications and services. Expert users are tasked to verify assigned resources and select optimal combinations to accommodate the infrastructure operations. For the technical users (engineers) in this specialised environment, user intent is not modelled in the traditional HCI application sense, but rather by intentionally combining the functional and non-functional requirements of the infrastructure through provider recommendations that are used as features. This work reports on the design, development and evaluation of a user interface that enable intent transfer from the specialised technical level of the expert user to the provider recommendation evaluation by the same users.

Keywords: User interface · User evaluation · Quality of experience · Usability · Cloud provider services · User intent · Recommendations

1 Introduction

Centralised cloud computing infrastructures are currently handling the processing and storage workload of most applications and services, rendering cloud computing a key component of modern economy. There is a plethora of computing and storage resource offerings by multiple cloud providers, such as Google, Microsoft Azure and Amazon Web Services (AWS) and smaller ones like Vultr, UpCloud, and Linode. The resources offered differ in terms of computing, networking, storage, and memory capacity, while targeting different use cases. These also differ in terms of cost, availability, security, region of operation and other parameters of interest. Offerings also include multi-cloud services, incorporating multiple resources from multiple cloud providers, so the customers may deploy their workloads and store their data in a (semi-)transparent manner. Recently, edge computing has also emerged offering computation and storage at the very edge of the network where data is produced, in order to reduce latency and limit the load that is carried to higher layers of the infrastructure hierarchy.

Edge, together with the traditional cloud resources, form an edge-cloud continuum [1] that offers better quality of services and lower monetary and energy costs. Tasks and data are assigned respectively: ephemeral storage and low-latency required computations on-device or/and on the edge, permanent storage and complex computations at the cloud (Fig. 1).

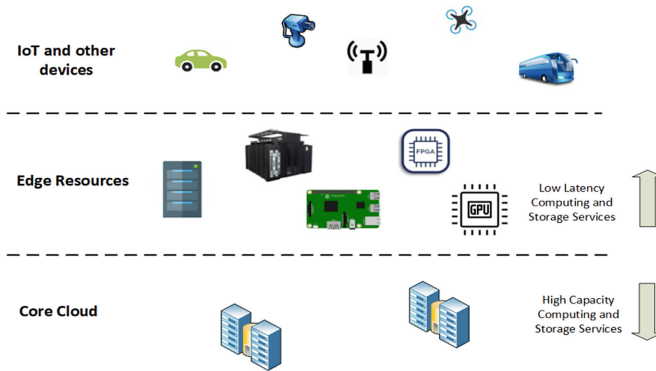


Fig. 1. The on-device-edge-cloud continuum.

In a multi-cloud and multi-edge infrastructure environment the service of a particular workload and storage request may be served with multiple ways, by allocating different combinations of the available resources. Orchestrator entities are responsible for receiving application requests, in terms of computational and storage workload, and for deciding their efficient deployment based on the objectives and constraints (performance, energy, cost, security, etc.) set. However, it is not always possible for users to identify quantitative measures for their submitted workload, as it is required by an orchestration mechanism. In addition, direct user interaction, as part of the decision process, may increase the trust of the users to the provided infrastructure services and their quality of experience (QoE) [2].

In this work, we (a) propose the use of a prototype user interface (UI) that enables users and applications to specify requirements through generic intents, and (b) report on the user satisfaction from the utilised resources.

The rest of the paper is structured as follows: Sect. 2 overviews related work on cloud recommendation challenges, while Sect. 3 presents the proposed intent-driven recommendation. Section 4 presents the UI prototype and the pilot user evaluation. Finally, Sect. 5 concludes the paper and outlines the future work.

2 Cloud Service Recommendation

In the cloud-era, users have to select from a variety of cloud services and cloud-based application programming interfaces (API), with similar functionalities but different quality of service (QoS) characteristics. This makes the cloud service selection process a challenging task for users or applications balancing between the satisfaction of functional and quality requirements.

A number of service recommendation approaches have been proposed to assist cloud service selection, deciding a ranked list of services based on their QoS values. The recommendation methodologies utilised are based on state-of-the-art approaches of recommender systems, appropriately extended for the cloud-domain. In general, the recommender systems can be classified into the following different categories, based on the techniques adopted:

- a. Content-based recommendation systems recommend similar items to a user, based on the items the user liked in the past, either explicitly (rating) or implicitly (clicking on a link, selecting a resource).
- b. In collaborative filtering, users are placed into groups of users of similar interests and a user's potential preferences for items are based on the other group's members know preferences. For more details on collaborative filtering, the interested reader is referred to [3, 4].
- c. Recommendation methods based on association rules count the relationships in which different rules appear based on historical data.
- d. Knowledge-based methods recommend items to a user in an interactive manner.
- e. Hybrid recommendation methods combine multiple methods [5].

Also, since QoS values are not always available for service recommendation, being either too expensive/difficult to collect or lost over time, the service ranking approaches adapt their proposition using partial data set and predictions [6].

A number of works provide a survey of the cloud services recommendation research activities [7, 8]. The basic parameters and characteristics considered by cloud service selection approaches include security, performance, accessibility, usability, scalability, resource distribution, and cost. In [9], a collaborative filtering approach for personalized cloud services recommendation is proposed. Users are grouped into communities, based on their similarities (geographic proximity, rating history and interest) and then a ranked list of services is recommended with the best predicted ratings. [10] proposes a Recommendation-as-a-Service concept and develops a cloud recommendation platform to recommend cloud configurations based on estimated cloud platform performance and users' budgets. [11] designs a user-centric recommendation framework of cloud services, which uses a collaborative filtering approach, focusing on users' personalised preference and experiences on cloud QoS. [12] proposes a recommendation approach that improves the efficiency of QoS-aware service selection for multi-tenant SaaS. The search space of the service selection is reduced by selecting representative candidate services based on the users' QoS requirements.

The quality of service (QoS) of cloud services changes frequently over time. A number of existing service recommendation approaches [13–16] attempt to address this property, considering the effect of user preference change over time for cloud service API recommendation. The proposed methodology tracks changes in user preferences through the temporal behaviour-aware information and combines the results of preference drift detection with cloud service API recommendation to generate recommendation results. [17] describes adaptive recommendations for VMs in the edge–cloud environment to

serve various IoT workloads according to multiple purposes. [18] presents a recommendation system for Infrastructure-as-a-Service for cloud offerings that enables users to define multiple design-time and real-time QoS constraints or requirements.

This work presents a UI design where user intent, user satisfaction and service recommendations can be efficiently communicated between users and cloud providers.

3 Intent-Driven Operations and Recommendations

Recently, relative intent-based operations have been proposed by various actors (standardization organizations, providers, academia) as a declarative approach for applications and users to specify their requirements on an infrastructure. Different actors may have different perspectives for intent driven operations in the networking or the cloud area, e.g., who can use them and how these can be used. For example, one approach is to include technical details that requires some level of expertise, while another approach is to shield users from technological details. A number of principles that can be common among intents and the different usage scenario have been identified in the literature: (i) intents should be declarative, (ii) an easy-to-use interface should be provided for their definition, (iii) intents should be technology independent and portable across similar systems.

This work proposes an interface for users to provide their intents regarding their submitted workload and data in cloud and edge infrastructures. The intent-driven paradigm of federated cloud infrastructures enables applications and users to express their high-level requirements in an infrastructure agnostic manner. Hence, applications and users can experience transparent, adaptive, and efficient access to heterogeneous processing and storage resources in the cloud and in the Edge, that also belong to different providers (multi-cloud). In a multi-cloud and multi-edge infrastructure environment the service of a particular workload and storage request may be served with multiple ways by allocating different combinations of the available resources (Fig. 2).

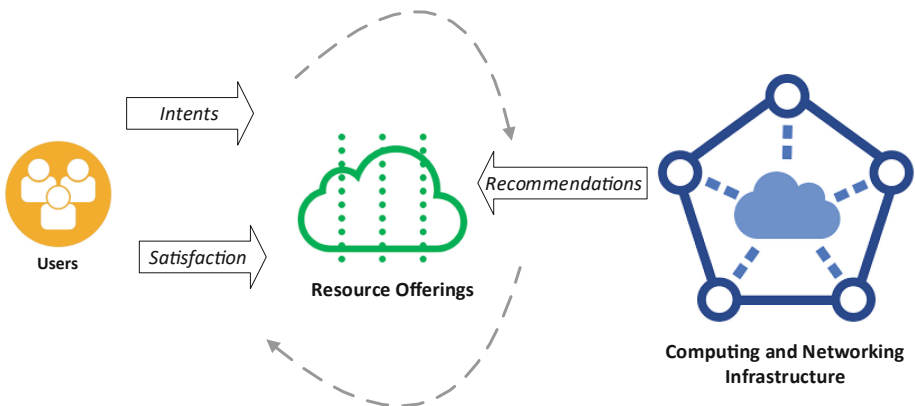


Fig. 2. A closed-loop operation involving user intents and satisfaction and infrastructure-service recommendations.

The UI can utilise the intents provided by the users to narrow down the available options and better match the user and application preferences.

The following steps are embedded into the design:

1. The user sets the demands to be served along with his/her intentions.
2. The system provides a set of available options/recommendations regarding the resources to be utilized.
3. The user selects and submits the workload.
4. After job completion, the user provides quality of experience feedback.
5. The above is fed in the recommendation system that provides resource recommendations for users in the next round of demand submissions.

The intent-driven paradigm enables applications and users to express their high-level requirements in an infrastructure agnostic manner. Hence, applications and users can experience transparent, adaptive and efficient access to heterogeneous processing and storage resources in the cloud and in the edge that may also belong to different providers (multi-cloud). Intents can then be translated to specific infrastructure-aware parameters served to the recommendation and the orchestration sub-systems. The provided intents can be application-domain related, indicating the reason for submitting the respective workload.

In what follows, we provide some examples of such intents. For example, a Fintech related intent can be the following: “Execute an X number of market analysis operations, for the period of Y months, for Z investment portfolios”. This intent indicates the amount of workload that will be submitted, and that the type of workload has low processing latency, high security, and low availability requirements. An intent for a storage provider can be as follows: “Store X amount of old medical records”. This intent indicates that these data will be stored for a long term, there is not any need to have low retrieval latency, while there are high security and reliability requirements. Alternative, an intent like: “Upload X number of photos from the party”, indicates that these data will be accessed immediately by many people for a short period of time. This means that reliability, availability, and storage costs are less important, instead the access latency is a more significant parameter. As a result, selecting edge storage resources instead of cloud ones can better serve this scenario. These intents can be specified in the UI by providing application domain specific menus, avoiding the use of natural language.

The above intents can be then translated to the following generic parameters before submitted to the recommendation and orchestration subsystems (Table 1). Other parameters can also be specified.

4 User Interface Design and Evaluation

The UI design for this work aims to provide a feedback-rich visual representation of the QoE selections of combinations and setups. This required comparative visualisations of selected setups, visualising the advantages and disadvantages of each selection, to allow user adjustments and re-scoring (Fig. 3).

For the evaluation, six experienced developers were asked to evaluate the UI prototype, specifically interfacing with randomised evaluation-driven scenarios for QoE,

Table 1. Parameters derived from analysis of intent.

id	name	values	Description
1	GeoLocation	0: near user 1: anywhere 2: specific	Specify whether the allocated resources will be geographically close or not, to the user
2	Availability	0: always on 1: highest possible 2: best effort 3: any	This specifies the availability requirements depending on the criticality of the submitted workload and data
3	Cost (computing, storage, access)	0: minimum 1: best effort 2: any	The cost the user or application is willing to pay for utilizing the respective resources
4	Latency	0: minimum 1: best effort 2: any	The latency requirements of the submitted workload for processing, for data storage or data retrieval
5	Security	0: None 1: Best effort 2: Storage only 3: Computational only 4: Storage and Computational	The security requirements

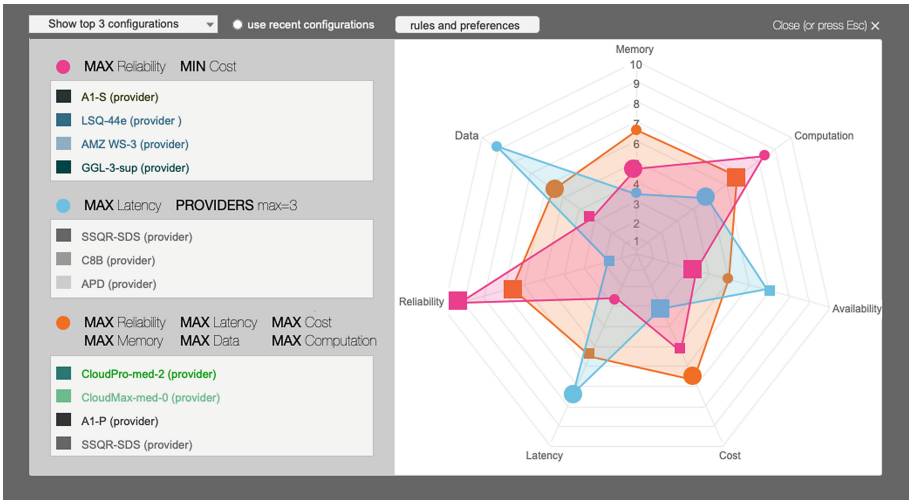


Fig. 3. User Interface for the exploration of recommended configurations.

and reported on the usability (accuracy, ease of use, acceptance). More specifically, the proposed UI was designed to implement the following functionalities:

- Default values and selections for multiple provider recommendation candidates, allowing the user to either accept or enable/disable or adjust the parameters of individual candidates.
- Real-time score visualisation for default and adjusted values.
- User-initialised value recast, as a user-in-the-loop recommendation formulation.
- QoE exploration through system recommendation.

The users reported that three was the optimal number of setup feedback configurations. The parameter setup process was more accurate, and the visual feedback was found to be very useful. Recommendations for improvement included exporting capabilities for selected parameter setups, visualisation of intent through matching to already constructed setups, and explainable information via user interaction triggering on the visualised comparison charts.

5 Conclusion

This work presented a UI prototype that which enables users and applications to specify cloud provider requirements through generic intents. This UI incorporates functionalities to provide a feedback-rich visual representation of the QoE selections of combinations and setups. Experienced developers participated in an experiment through which the UI was evaluated. The evaluation process showed very high levels of accuracy, ease of use and acceptance. Our future work will focus on considering social media data for search enrichment and recommendation accuracy.

Acknowledgements. The work was supported by the EU research project SERRANO, under grant agreement No 101017168.

References

1. Kretsis, A., et al.: SERRANO: transparent application deployment in a secure, accelerated and cognitive cloud continuum. In: 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom). pp. 55–60. IEEE, Athens, Greece (2021). <https://doi.org/10.1109/MeditCom49071.2021.9647689>
2. Spiliotopoulos, D., Margaris, D., Vassilakis, C.: Data-assisted persona construction using social media data. *Big Data Cogn. Comput.* **4**, 21 (2020). <https://doi.org/10.3390/bdcc4030021>
3. Margaris, D., Spiliotopoulos, D., Vassilakis, C.: Social Relations versus near neighbours: reliable recommenders in limited information social network collaborative filtering for online advertising. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2019). pp. 1160–1167. ACM, Vancouver, B.C., Canada (2019). <https://doi.org/10.1145/3341161.3345620>

4. Margaris, D., Kobusinska, A., Spiliotopoulos, D., Vassilakis, C.: An adaptive social network-aware collaborative filtering algorithm for improved rating prediction accuracy. *IEEE Access* **8**, 68301–68310 (2020). <https://doi.org/10.1109/ACCESS.2020.2981567>
5. Aivazoglou, M., et al.: A fine-grained social network recommender system. *Soc. Netw. Anal. Min.* **10**(1), 1–18 (2019). <https://doi.org/10.1007/s13278-019-0621-7>
6. Margaris, D., Spiliotopoulos, D., Vassilakis, C., Karagiorgos, G.: A user interface for personalized web service selection in business processes. In: Stephanidis, C., et al. (eds.) *HCI 2020*. LNCS, vol. 12427, pp. 560–573. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60152-2_41
7. Sun, L., Dong, H., Hussain, F.K., Hussain, O.K., Chang, E.: Cloud service selection: state-of-the-art and future research directions. *J. Netw. Comput. Appl.* **45**, 134–150 (2014). <https://doi.org/10.1016/j.jnca.2014.07.019>
8. Aznoli, F., Navimipour, N.J.: Cloud services recommendation: Reviewing the recent advances and suggesting the future research directions. *J. Netw. Comput. Appl.* **77**, 73–86 (2017). <https://doi.org/10.1016/j.jnca.2016.10.009>
9. Afify, Y.M., Moawad, I.F., Badr, N.L., Tolba, M.F.: Enhanced similarity measure for personalized cloud services recommendation: enhanced similarity measure for personalized cloud services recommendation. *Concurr. Computat. Pract. Exper.* **29**, e4020 (2017). <https://doi.org/10.1002/cpe.4020>
10. Jung, G., Mukherjee, T., Kunde, S., Kim, H., Sharma, N., Goetz, F.: CloudAdvisor: a recommendation-as-a-service platform for cloud configuration and Pricing. In: 2013 IEEE Ninth World Congress on Services, pp. 456–463. IEEE, Santa Clara, CA, USA (2013). <https://doi.org/10.1109/SERVICES.2013.55>
11. Yu, Q.: CloudRec: a framework for personalized service recommendation in the cloud. *Knowl. Inf. Syst.* **43**(2), 417–443 (2014). <https://doi.org/10.1007/s10115-013-0723-x>
12. Wang, Y., He, Q., Yang, Y.: QoS-aware service recommendation for multi-tenant saas on the cloud. In: 2015 IEEE International Conference on Services Computing, pp. 178–185. IEEE, New York City, NY, USA (2015). <https://doi.org/10.1109/SCC.2015.33>
13. Li, S., Wen, J., Luo, F., Ranzi, G.: Time-aware QoS prediction for cloud service recommendation based on matrix factorization. *IEEE Access* **6**, 77716–77724 (2018). <https://doi.org/10.1109/ACCESS.2018.2883939>
14. Ding, S., Li, Y., Wu, D., Zhang, Y., Yang, S.: Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and ARIMA model. *Decis. Supp. Syst.* **107**, 103–115 (2018). <https://doi.org/10.1016/j.dss.2017.12.012>
15. Meng, S., et al.: A Temporal-aware hybrid collaborative recommendation method for cloud service. In: 2016 IEEE International Conference on Web Services (ICWS), pp. 252–259. IEEE, San Francisco, CA, USA (2016). <https://doi.org/10.1109/ICWS.2016.40>
16. Wang, L., Zhang, Y., Zhu, X.: Concept drift-aware temporal cloud service APIs recommendation for building composite cloud systems. *J. Syst. Softw.* **174**, 110902 (2021). <https://doi.org/10.1016/j.jss.2020.110902>
17. Xu, Y., Li, J., Lu, Z., Wu, J., Hung, P.C.K., Alelaiwi, A.: ARVMEC: adaptive recommendation of virtual machines for IoT in edge-cloud environment. *J. Parallel Distrib. Comput.* **141**, 23–34 (2020). <https://doi.org/10.1016/j.jpdc.2020.03.006>
18. Zhang, M., et al.: An Infrastructure service recommendation system for cloud applications with real-time QoS requirement constraints. *IEEE Syst. J.* **11**, 2960–2970 (2017). <https://doi.org/10.1109/JSYST.2015.2427338>