

A Novel Cloud Services Recommendation System Based on Automatic Learning Techniques

Rahma DJIROUN¹, Meriem Amel GUESSOUM¹, Kamel BOUKHALFA¹ and Elhadj BENKHELIFA²

¹Department of Computer Science, LSI/USTHB, Algiers, Algeria.

Email: {[rdjiroun](mailto:rdjiroun@usthb.dz) ; [mguessoum](mailto:mguessoum@usthb.dz) ; [kboukhalfa](mailto:kboukhalfa@usthb.dz)}@usthb.dz

²Cloud Computing and Application Research Lab, Staffordshire University, UK.

Email: e.benkhelifa@staffs.ac.uk

Abstract—The Cloud Computing technology is evolving constantly but essence remains the same that is to offer distinct cost saving opportunities by consolidating and restructuring information technology as a service. With the continuously increasing cloud provisions, cloud consumers start to have difficulties to find the best relevant services that suit their requirements. Therefore, selecting best services by cloud users is becoming a greater challenge. In this paper, we present a framework of services' recommendation system in a Cloud environment, using automatic learning techniques. The system aims at finding the services that suit the interests and preferences of cloud consumers by combining content based and behaviour based recommendations. In this paper, we present, USTHB-CLOUD, a cloud services recommendation prototype evaluated with an experimental study.

Keywords—Cloud Services; Recommendation systems; Automatic learning ; Data mining.Cloud Computing; USTHB-CLOUD

I. INTRODUCTION

In recent years, the emergence of Cloud computing has offered a new computing model, where resources such as computing power, storage, applications and networking infrastructures are provided as 'services' over the internet [1]. Currently, Cloud computing has evolved almost to a mature state and continues to improve in offering distinct cost saving opportunities by deploying advanced solutions to consolidate and restructure the information technology provisioning, to be more dynamic, lightweight, low cost, transparent, reliable and secure.

Nowadays, there exist numerous Cloud services by numerous providers as much as this increase in cloud services and providers is beneficial to the consumer, it poses a new challenge for users. This challenge is posed when these users endeavour to select variety of cloud services relevant best services from a large number of available cloud services in an optimal time. In this situation, manual approaches will be inefficient and cumbersome; hence the need for automated recommendation systems is the way forward to support the user's decision making for best cloud services.

In this context, a recommendation system will be computer-based intelligent solution that is able to solve the problem of finding appropriate services among a large number of available

services [2]. They can be used to efficiently provide personalized services to cloud users by providing suggestions of needed services[3].

A recommendation systems are defined as a software agents that elicit the interests and preferences of consumers [4]. They are considered as a specific form of information filtering to predict the interests or preferences that customers would have attributed to an item or an element not considered before, which saves a lot of time [5].

Several factors are considered in order to categorize recommendation systems such as knowledge of customer's profiles and preferences and knowledge of the items to be recommended. From these factors, several types of recommendations are produced. The most commonly used types are content filtering and collaborative filtering [6].

Several methods in Automatic Learning and Data Mining fields are used in recommendation systems. When a cloud user requires a recommendation for a cloud service, the system determines by an automatic learning the most relevant items based on the users' previous interactions. This technique has been reported by several researchers. other researchers aim to help a user to select best services from different cloud providers , based on the Quality of Services (QoS) and the user's feedback [2], [7]. These approaches are based on the K-means clustering techniques. The user's feedback is provided in a repository, then, the clusters are produced with respect to feedback values given by the user.

In [8] and[9], authors proposed Cloud Service Research and Selection System using the Skyline and an outranking method to select the cloud services that best meet the users' requirements. In the same context authors of [10], used agents, which are trained to observe differences and behave intelligently for service selection.

To the best of the authors knowledge, relate work in literature uses mainly the similarity criteria to determine which Cloud service is the most suitable to the user's requirements. This work is suited only for users who want to find Cloud services that are similar to the ones they already know or use, but does not consider other users who search for best services without having any prior knowledge.

In this paper, we propose a cloud services recommendation approach based on learning techniques and data mining, using clustering methods to give better visibility to users. In addition,

we propose in this approach, to analyze the different user's interactions (consulting, using, unsubscribing, etc.) with the Cloud services in order to define in real time the user's requirements, taking into account any changes of requirements overtime. In this context, we combine two kinds of recommendations approaches. A recommendation based on the content and another one performed on the basis of the user's behavior, which aims to predict user's interests in services done according to the analysis of their traces of other users.

The rest of this paper is structured as follows: Section 2 presents a summary and analysis of the related work. Section 3 describes the architecture of the proposed framework. In section 4, we present the prototype and we name it *USTHB-CLOUD* supported by selected experimentations for evaluation purposes. We conclude our work in Section 5 and present some perspectives and future works.

II. RELATED WORKS

With the increase variety of services in Cloud Computing, the cloud users need to have a system that allows searching and recommending option that best match their requirements, among various Cloud services. There are several work in the literature, which deals with this subject. We can define four classes of works : (1) work focusing on the quality of services (QoS)[2] [7] [10] [8]; (2) work based on collaborative filtering [3] [11]; (3) work focusing on the performance factor [12] [13] and (4) work searching relevance services [14],[15].

In [2], authors propose a Cloud service selection framework in that uses a recommender system. Their proposed system recommends a service based on a network of QoS and Virtual Machine platform factors of different Cloud providers in order to recommend a good combination of Cloud services to consumers.

In order to achieve high quality recommendations authors proposed in [10] an approach using agents that can be trained to observe differences and behave intelligently for service selection. In order to rank different clouds, they proposed a new technique based on artificial intelligence that assigns performance factor for each service provided by a cloud provider and ranks them accordingly.

In [7], authors focused on QoS in cloud service selection method, allowing users to specify their perception of quality criteria. The proposed work is based on data mining technique, clustering. Authors developed a new K-means clustering algorithm that classifies cloud services into different groups based on selected quality attributes and ranks them accordingly. The objective is to assist different type of users for choosing a cloud service without engaging into any financial contract.

In [8], authors improved the previous work on the Cloud Search and Selection System (CSRSS) by introducing seven new dimensions related to QoS constraints. Their work involved designing an agent that uses both the Skyline method and an overflow method, called ELECTREIs Skyline, to determine which Cloud services better meet user's requirements, giving users the ability to specify the values of the QoS attributes they require.

In the same way, authors of [9] proposes a method, which allows Cloud users to find a Cloud service that meets their requirements. The approach is based on the principle of the Skyline. The main contribution of this work is the construction of a Web Agent, using the Skyline method to determine which Cloud services best meet users' needs.

Authors proposed, in [12], a cloud service architecture and service selection algorithm in order to select the optimum service. The cloud users can access cloud services by proxy to select services with lower cost and larger gains (more rapid response, more economic, or more convenient, etc.). The proposed algorithm is scalable and adaptive to different types of environments of services and clients.

In [13], the authors presented a multi-criteria cloud services selection methodology such as user requirement, service and performance. The selection process is based on a method built according to a rigorous mathematical formalization and models.

In [3], a framework for a cloud services selection based on a recommendation process is proposed, in order to lead users to select the most appropriate services from different cloud providers. This system uses collaborative filtering techniques. It is based on a services' ranking established on the basis of the quality of service of a network and virtual machine performance.

A blog recommender system using a cloud computing technology was proposed in [11]. The authors used a collaborative filtering algorithm, which tries to find similarity between several users' information to generate personalized recommendation with high quality in a traditional blog search system. In these work, the recommender system is based on Hadoop distributed file system to store mass blog data and implement the distributed processing of blog crawl and index creation stages.

In [14], the authors proposed a web site recommendation system implemented over a cloud framework. The recommendation is based on user's browsing habits analysis using mining concepts, which analyses the user's web usage statistics and performs clustering algorithm as K-means. The browsing habits of a group of users are analysed and the users are assigned to different clusters in the server such that the users in the same cluster have similar browsing habits to visit similar web sites.

In [15], the authors proposed an approach, which adds a form of communication between the users and the searched cloud services, via requesting the users to enter values about their ideal services. They used clustering methods to set the resulting cloud services in clusters and give a better visibility to users. The approach is based on an algorithm that combines the use of the skyline operator with the ELECTRE IS (ELimination and Choice Expressing REality) method in order to refine the results before applying the clustering algorithm. These techniques allow the system to return to the user a small set of cloud services, which are the closest to the ideal service. The above summarised related work use the similarity criteria to find a Cloud service, which is the most suitable to the user's requirements. The reported research seem to focus only on users who want to find Cloud services that are similar to the ones they already know or use, but neglect users those want to find

services who are not familiar with the existing Cloud services. Indeed, in addition to the user's requirements, a recommendation system must consider multiple attributes: user's interaction in real time, user's feedback, relations between services (composition, use, etc.). Most of the proposed related research do not take all these parameters in consideration at the same time. The approach, which we propose, attempts to offer a solution for this challenge where all these attributes are taken into account when recommending cloud services to different users.

III. THE PROPOSED APPROACH

In this paper, we propose a recommendation approach based on two aspects: analyzing the content and description of services, and analyzing user's behavior i.e. interactions and previous actions with services (consultation, use, unsubscribing of a service or declining a recommended service).

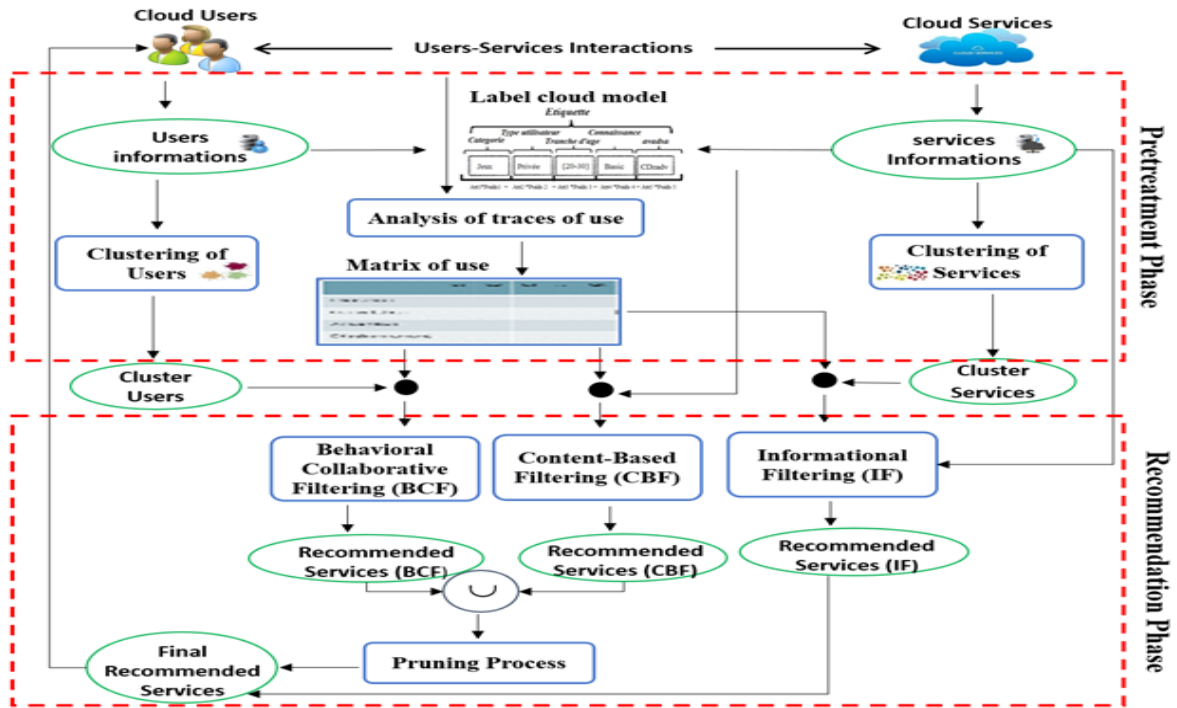


Fig. 1. General architecture of our recommendation system.

Our approach is based on automatic learning techniques, allowing to build user's models in order to generate predictions about future user's interests, concerning services not consulted before or services which the user is not aware of.

In this section, we present the two main phases of our approach, pre-treatment and recommendation. The first phase aims to prepare the necessary elements for the second phase such as user's profile and service's characteristics, structure of the usage matrix and the label cloud model used in the recommendation process. The second phase recommend services to the cloud user based on filtering techniques (performed according to content, user's behaviour and contextual information). The architecture of the proposed approach is depicted in Figure 1 and described in more details below.

A. Pretreatment phase

This phase is based on users' and services' information as well as information on user-service interactions. These features lead us in the recommendation process via applying clustering

algorithms to build both users' and services' clusters. In addition, a label cloud model is defined for users and services, which acts as a descriptor of users' and services' information. From user-service interactions, we build a matrix of use, which gives the current image that summarizes all performed actions by users on services.

1) *Data description*: We structure the information (attributes) of users and characteristics of services in classes. The attributes of users are inspired from the most popular user's profiles of Cloud providers such as Amazon and Google and Azure as id-user, name, age, gender, languages...etc. The characteristics of services are those of the taxonomy (IaaS, PaaS, SaaS) [16], while adding other characteristics collected from a study carried out on the platforms of the suppliers, which we have judged as relevant in the recommendation system.

2) *Users' and services' Clustering*: To build the clusters, we choose the PAM method (K-Medoids) [17] as Non-hierarchical Clustering methods in order to partition a set into subsets represented by their center vector (Centroid) and to minimize the

sensitivity of k-means to mixed (categorical and numerical) values. In our approach, we want to define a single representative for each cluster, and the center responds perfectly to this need.

The cluster's construction is primarily based on the calculation of similarity and proximity between objects. In our approach, the "user" and "service" classes are made up of three types of attributes: numerical, categorical and binary. We opted for three different distance measures such as [18] : (1) Euclidean distance for numerical attributes (formula (1)); (2) the *Leveinstehin* distance for categorical attributes (the minimum number of insertions, deletions and replacements necessary to transform a string A into a string B) and (3) a binary distance is used for the calculation of similarity between binary attributes. For example, the attribute gender can take as values: "Male" and "Female", the distance between two users those have the same value for this attribute is equal to 0 (similar) and is equal to 1 in the opposite case (Dissimilar).

$$EucDis(U_i, U_j) = \sqrt{\sum_{k=1}^n (U_{i_{attrk}} - U_{j_{attrk}})^2} \quad (1)$$

The *Euclidean* and *Leveinstehin* distances are sensitive to the size of the values representing the attribute. Indeed, attributes with large values tend to dominate other values. This incompatibility of the measurement units and the difference of great values can affect the final results. In order to adjust the values, we normalize these last applying a transformation function that makes these values comparable using the formula (2).

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (2)$$

Where x_i represents the value of the distance to be normalized, $x = (x_1, \dots, x_n)$ represents the set of distance values

3) *Description of the labels cloud structure*: In order to recommend the most relevant Cloud services to a user, a similarity calculation between this user and all Cloud services is required. For this, we define a common structure describing the Cloud users and services called *label cloud* constituting a vector, thus, the attributes of each class are described by labels in the cloud.

The structure of a user's label cloud includes some attributes such as: Cloud category, Feature, Domain of Interest, Languages, User Type (Private or Business Entity), User Cloud Knowledge Level, Financial Status, Region, user's age range. The initial values representing the centers of interest are extracted from the user's profile. The problem is the ability that this user evolves and changes his/her interests over time. To address this issue, the label cloud values are updated dynamically when the user interacts with the services. This process will be detailed in the behavioural collaborative filtering.

4) *Trace use Analysis*: In order to have a current image of all users' interactions on services, we construct a matrix of use, resulting from the analysis of users' profiles and their

interactions. This matrix is composed of notes that capture the interests of each user with respect to services. These notes are calculated from the traces of use extracted from the user's profiles, for which we distinguish two profiles: explicit and implicit.

The explicit profile of the users consists of the evaluations explicitly provided by the users. The note given by a user U_i to a service S_j is denoted Note (U_i, S_j).

Concerning the implicit profile, we have constructed a model based on the analysis of the user's behaviour in order to predict his/her tastes and to estimate the interest that he/she can bring to each service. In this work, we estimate the interest of a user for a service by the indicators: duration of consultation of a service, duration of use of a service, the unsubscribe from a service, the decline to a recommended service.

To construct the matrix of use, we take only "consultation" and "use" traces from implicitly collected appreciations and the explicitly provided notes. We propose the formula (3) which estimates the interest that a user U_i has for a service S_j .

$$Interest(U_i, S_j) = \beta * Note(U_i, S_j) + \alpha * UseNote(U_i, S_j) + \delta * ConsultNote(U_i, S_j) \quad (3)$$

Where, α , β , δ , are the weights of note, usage note and consultation note respectively.

We consider that the usage note is more significant than the consultation note and note explicitly expressed by a user is more significant than the usage note, even if the user uses frequently service without ever note it. Thereby, we consider that $\beta=3$, $\alpha=2$ and $\delta=1$.

B. Recommendation phase

This phase aims to identify cloud services to recommend to user. We exploit user's and service's Clusters, matrix of use and the cloud labels structure produced in the pretreatment phase. The recommendation process is based on Content-based filtering and behavioral filtering. A pruning process is applied to a set of recommendable services that may include incompatible relation between services as well as services considered as not interesting by the user.

In order to enrich the recommendation process, we propose to add a contextual filtering process, which produces a set of services that may be of interest the user when he/her consults or uses a given service.

1) *Content based filtering*: this process produces a set of services adapted to the user, based on the matrix of use and the labels cloud structure. It consists of a dynamic build of the user's label cloud, calculating similarity between the active user labels cloud and labels cloud services, and then a decendent sorting on the similarities to identify the recommended services that correspond most to the user. To analyze the content, we use the measure TF*IDF weighting [19] in order to compare the label cloud of a user with that of a service, where we define how a term is considered in: consultation, use, appreciation and unsubscribing. We consider a use of a term five times more important than its consultation. An appreciation of a term is considered more important than its use by adding the grade

given to that term. Finally, unsubscribing a term is discriminated by considering it five times less important than its consultation. For the calculation of similarity between labels clouds, weights are assigned to the attributes of the cloud to express their level of importance in the cloud. It should be noted that all the terms of the same attribute have the same weight. This is applied only on the user labels clouds, because in the case of services there is no order of preference or importance established between the labels of the same attribute.

2) *Behavioral collaborative filtering*: This process seeks to predict user's appreciations about services, based on user's clusters and the matrix of use (interactions of similar users). The content-based filtering process that we have shown presents some disadvantages due to the lack of diversity, because the recommended services are always similar and identical (in terms of content) to the cloud services with which the user has interacted. In order to improve the quality of the recommendation and the diversity, we propose a new recommendation model based on collaborative behavioral filtering centered on the user. This model is inspired by both predictive approaches derived from WUM (Web Usage Mining), and memory-based recommendation approaches such as FCS [20]. The exploitation of the behavior in a recommendation process avoids the problem of lack of explicit notes provided by the users. Indeed, in the case of user's interactions with services, the amount of traces of uses is potentially greater than the amount of explicit available notes. The behavioral collaborative filtering model consists of two steps:

(1) active user's neighborhood identification and (2) active user's interests prediction. In the first step, we aim to generate relevant recommendations using the opinions of the active user's neighborhood. For this, we use users' clusters to reduce the search space. Identifying the neighbors may in particular be based on several evaluation measures users' appreciations of similarities, among which are the *Pearson correlation coefficient*, the *Cosine measure*, and the *Spearman correlation* [20]. In our approach, we use the most widely exploited measure in current recommendation systems such as the *Pearson correlation coefficient* (formula (4)).

$$Pearson(V, W) = \frac{(V-\bar{V}) \cdot (W-\bar{W})}{\sqrt{(V-\bar{V}) \cdot (V-\bar{V}) \times (W-\bar{W}) \cdot (W-\bar{W})}} \quad (4)$$

Pearson correlation between two vectors is a value between -1 and 1. Two identical vectors are perfectly correlated (correlation = 1), while two opposite vectors ($V = -W$) are perfectly anti-correlated (correlation = -1) [20].

If we compare two opposite users U_1 and U_2 , U_1 , where one prefers service S_1 and the other prefers service S_2 . These users are then uncorrelated.

For the prediction interest for an active user step, we exploit the most commonly used method for calculating predictions is the "weighted sum" [40]. It considers the nearest neighbours U (correlated with the active user U_i) that have already interacted with the service S_j to calculate the prediction of the interest of U_i on S_j , denoted $Pred(U_i, S_j)$ [19].

We use the formula (5) to predict the user's rating U_i for a service S_j .

$$Predict(U_i, S_j) = \overline{v(U_i)} + \frac{\sum_{U_k \in U} Similarité(U_i, U_k) \cdot (v(U_k, S_j) - \overline{v(U_k)})}{\sum_{U_k \in U} Similarité(U_i, U_k)} \quad (5)$$

The recommendation process based on behavioral collaborative filtering predicts notes (interests) for the active user on services that he has not explicitly noted or interacted with. These predictions are calculated from the notes given by its closest neighbors.

3) *Informational filtering*: In order to enrich the proposed recommendation, the informational filtering process produces a set of services that may be interesting to the user when this latter consults or uses a service independently of the information concerning him/her.

An approximation technique based on the services' clusters is used in the case of a consultation. Indeed, the set of recommended services correspond to the service's cluster to which this service belongs.

In the case of use of a service, a more complex approach based on traces is used. This technique is based on the method of finding the association rules in order to discover relationships between subscriptions extracted from the trace of uses, making it possible to identify frequent sequences of subscriptions in which the use service belongs. The longest sequence is retained by this method and the services constituting the latter represent the services to be recommended to the active user during his/her use of the concerned service.

To achieve this goal, we used the FP-Growth algorithm to take advantages of its tree structure in order to overcome the performance and response time limits of the two standard algorithms for searching association rules such as the *Apriori* and the *Eclat* algorithms [21]. The FP-Growth algorithm remains among the most efficient and widely used algorithms in the domain [21]. Two steps are necessary to perform this algorithm: the first one is to build the tree structure FP-Tree and the second is the extraction of frequent sequences directly from the tree FP-Tree.

4) *Pruning process*:

Although a large number of cloud services resulting from the recommendation phase is considered as relevant, these services can be non-interesting to the active user due to the nature of cloud computing and some characteristics of cloud services. In order to refine the recommended services to active users from this recommended set of services, we propose to prune the services that are part of three cases: (1) *Incompatible services*: It should be noted that when a cloud service is introduced to a system, the services with which this service is incompatible are provided by the cloud provider. To remedy this, the incompatible service least ranked is pruned of all services recommended; (2) *Recommended services previously declined*: sometimes the active user has already expressed his/her disinterest for a recommended service by removing it from the recommendation list. In this case, if the same declined service appears in the set of services currently recommended, this service is systematically discarded from the set and (3) *Services to which the active user is currently subscribed*:

sometimes the active user subscribes to a service and the service appears in all services currently recommended.

After pruning the services considered as irrelevant, resulting from the three cases mentioned above. The final set of services is more relevant and coherent to recommend to the user.

IV. PROTOTYPE REALIZATION AND EVALUATION

We present in this section, the prototype that we have developed to implement our approach and some experiments to validate it.

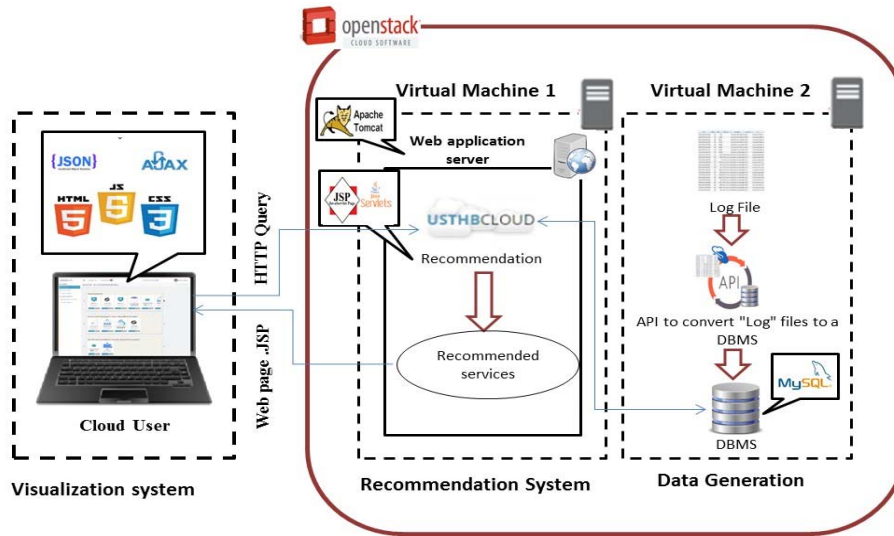


Fig. 2. Technical architecture of our recommendation system

A. Prototype Realisation

To implement our approach, we developed a cloud services recommendation prototype called *USTHB-CLOUD*. In this section, we present the architecture (see Figure 2) of our prototype, where it shows the different tools and development environments used in this realization.

The tool provides two interfaces. The first is for the cloud users, which allow them the registration and authentication on the platform, define their centers of interests by categories of Cloud services and their functionalities, ordered according to the preferences. After the authentication, the user is directed to a main page (Figure3) which is composed of several headings.

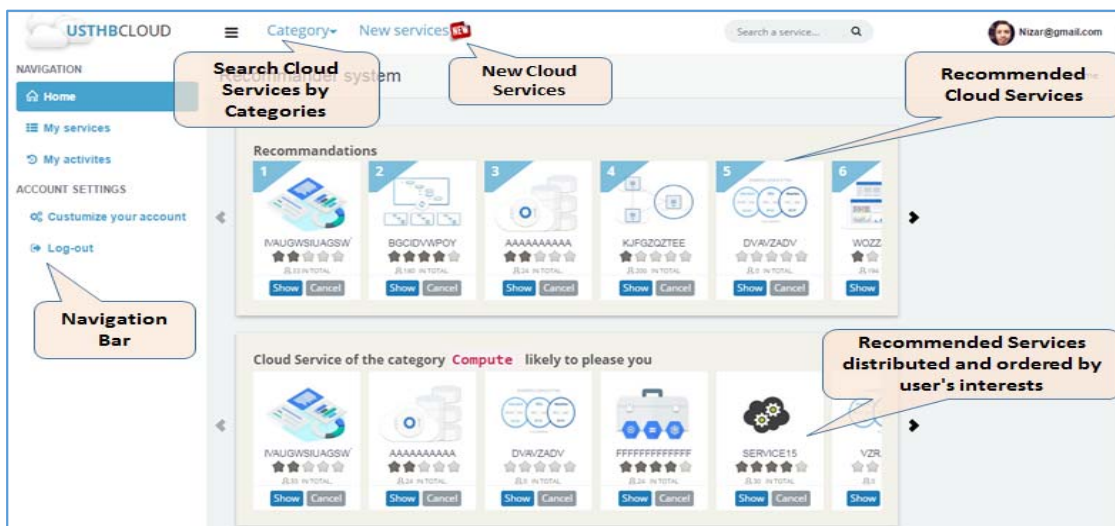


Fig. 3. User interface: Main page of the user

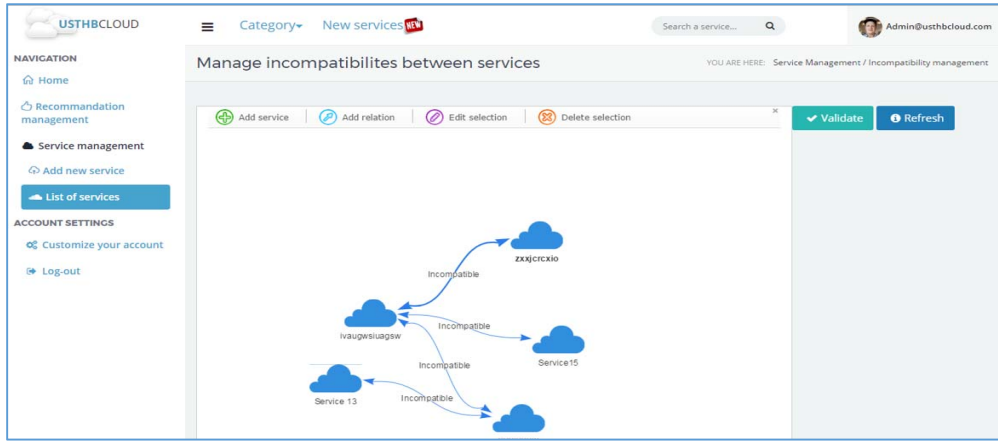


Fig. 4. Administrator Interface/Managing the relationship between services

The interface allows users to consult and use services, it also allows them to see the services of which they are subscribers and to consult their interactions in time through a dashboard (average duration of use of the services, number of subscriptions, etc.).

The second interface is devoted to the administrator for adding services, managing relationships between services (Figure 4), configuring and viewing clustering, and displaying the recommendation details.

B. Evaluation

In order to evaluate the quality of our recommendation system, we propose to analyse two evaluation criteria, relevance and performance.

1) *Relevance*: To evaluate the relevance of our recommendation system, we propose two evaluation measures, the first measure M1 (U_i) (Formula(6)), concerned with calculating the ratio between the number of interactions of the user U_i with services recommended by our system and the total number of its interactions. This means that if a user reacts with a recommended service, this can be considered a good indicator of the relevance of our system.

The second measure M2 (U_i) (formula (7)), calculates the ratio between the number of user interactions with recommended services and the total number of services recommended. This means that more the recommended services are those which the users interact with, more the system is relevant.

$$M1(U_i) = \frac{RSI(U_i)}{RSI(U_i)+IS(U_i)} \quad (6) \quad M2(U_i) = \frac{RSI(U_i)}{RS(U_i)} \quad (7)$$

Where S is the set of all services, RS is the set of recommended services, IS is the set of services with which the user has interacted, and IRS is the set of recommended services and interacted by the user (intersection of IS and RS).

The calculation of the relevance M1, M2 of the recommendation system amounts to calculate the average of relevance of the recommendation for all users. We calculate M1 and M2 from the matrix of use, we obtain the results presented in the following table:

TABLE I. RESULTS OF MEASURES OF RELEVANCE M1 AND M2

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10
M1	0.5	0.5	0.75	0.20	0.66	0.66	0.00	0.33	0.5	0.00
M2	0.25	0.25	0.75	0.25	0.5	0.5	0.00	0.25	0.75	0.00

After calculating the measurements, we obtain the following results: M1=0.37 and M2=0.35.

The interpretation of these results must be scaling to judge the quality of our recommendation system. For this purpose, we propose a scale of relevance corresponding to each result obtained: 0-0.3, 0.3-0.6, and 0.6-1 are bad, good, and very good respectively.

Although the relevance result of the recommendation system is considered good, these results are still non-persuasive because they are obtained from a simulation of user interactions with services.

2) *Performance*: In this section, we evaluate the performance of our recommendation system, calculating response time for each process of our system based on the number of services and users. We consider a dataset composed of 3000 services and 1000 users. Our experiments, were realized on a PC Dell Vostro with Windows 7, having 4 GB of RAM and a processor i3 of 2,4 GHz. In order to analyse the impact of Clustering on the performance of our recommendation system, we compare the response time of the recommendation system based on Behavioural collaborative filtering with and without clustering (K-Medoids algorithm). It should be noted that the execution time of the clustering process is included only once when a new user is added. We calculated the response time of each recommendation system based on the number of users. The results show (see Figure 5) that when the number of users is reduced, the response time of the system with or without clustering is negligible, but the response time without clustering increase from 300 users. This is due to the fact that the correlations between the users are calculated only within the clusters and not between all the users. So, we can say that our system with clustering is clearly better than the naïve

recommendation system without clustering especially when the number of users increases.

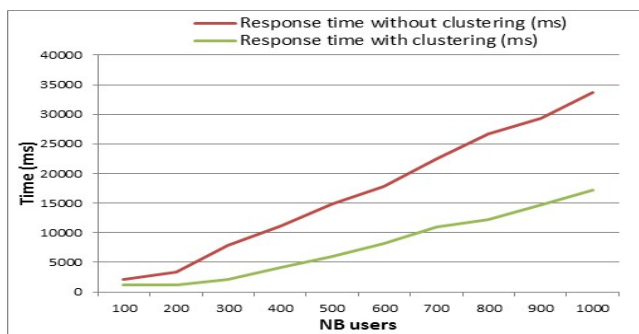


Fig. 5. Collaborative filtering with and without clustering vs number of users

V. CONCLUSION

In the aim to find a solution to select relevant services from a large number existing cloud services, we present in this paper a novel approach for cloud services recommendation system based on learning techniques and data mining. Our proposed solution combines both content filtering and behavioural collaborative filtering techniques. In addition, in order to enrich our recommendation system, we have introduced a recommendation based on informational filtering proposing non-personalized recommendations, independent of the user. A pruning process has been applied to the recommended services resulting from the two recommendation approaches to eliminate services that are not suitable for the user. To implement the proposed recommendation approach, we developed a tool called "USTHB-CLOUD". For the approach's validation, we conducted an experimentation study via assessing the relevance and performance criteria of our proposed system, by performing a series of tests and analysis, where the results obtained were encouraging.

For future works, we plan to improve the quality of our recommendation system by incorporating other information such as SLAs, energy consumption and infrastructure saturation. We also propose to use other recommendation techniques such as the demographic recommendation, as well as other automatic learning techniques such as the use of supervised classification in order to optimize the response time.

Acknowledgment

We thank SID Mohamed Nizar and OUAZAR Mourad, for assistance in the development of USTHB-CLOUD prototype.

References

- [1] M. Armbrust *et al.*, « A view of cloud computing », *Commun. ACM*, vol. 53, n° 4, p. 50–58, 2010.
- [2] S.-M. Han, M. M. Hassan, C.-W. Yoon, H.-W. Lee, et E.-N. Huh, « Efficient Service Recommendation System for Cloud Computing Market », in *Grid and Distributed Computing*, Springer, Berlin, Heidelberg, 2009, p. 117–124.
- [3] G. Adomavicius et A. Tuzhilin, « Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions », *IEEE Trans. Knowl. Data Eng.*, vol. 17, n° 6, p. 734–749, 2005.
- [4] E. Çano, « Cloud-based Recommendation Systems: Applications and Solutions », 2016.
- [5] M. Sharma et S. Mann, « A survey of recommender systems: approaches and limitations », *Int. J. Innov. Eng. Technol.*, vol. 2, n° 2, p. 8–14, 2013.
- [6] P. Melville et V. Sindhvani, « Recommender systems », in *Encyclopedia of machine learning*, Springer, 2011, p. 829–838.
- [7] T. Zain, M. Aslam, M. R. Imran, et A. M. Martinez-Enriquez, « Cloud service recommender system using clustering », in *2014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2014, p. 1–6.
- [8] M. Abourezq et A. Idrissi, « Integration of QoS aspects in the Cloud Computing Research and Selection System », *ArXiv Prepr. ArXiv170204966*, 2015.
- [9] A. Idrissi et M. Abourezq, « Skyline in cloud computing », *J. Theor. Appl. Inf. Technol.*, vol. 60, n° 3, 2014.
- [10] I. M. Rabbani, A. Muhammad, et M. E. A.m, « Intelligent Cloud Service Selection Using Agents », in *The 9th International Conference on Computing and Information Technology (IC2IT2013)*, Springer, Berlin, Heidelberg, 2013, p. 105–114.
- [11] J. Jiang, W. Pang, Y. Deng, K. He, et Z. Gu, « A Blog Personality Recommender System Based on Cloud Computing Infrastructure », in *Proceedings of the 2012 International Joint Conference on Service Sciences*, Washington, DC, USA, 2012, p. 1–5.
- [12] W. Zeng, Y. Zhao, et J. Zeng, « Cloud service and service selection algorithm research », in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 2009, p. 1045–1048.
- [13] Z. ur Rehman, F. K. Hussain, et O. K. Hussain, « Towards Multi-criteria Cloud Service Selection », 2011, p. 44–48.
- [14] S. K. Menon, V. Paul, et M. S. Kumar, « An efficient cloud based framework for Web recommendation systems », *Int. J. Adv. Comput. Sci. Technol.*, vol. 3, n° 2, p. 07–11, 2014.
- [15] Rehioui, A. Idrissi, et M. Abourezq, « The Research and Selection of Ideal Cloud Services using Clustering Techniques: Track: Big Data, Data Mining, Cloud Computing and Remote Sensing », 2016. .
- [16] C. N. Hoefler et G. Karagiannis, « Taxonomy of cloud computing services », in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, 2010, p. 1345–1350.
- [17] J. Han et M. Kamber, *Data mining: concepts and techniques*, 2nd ed. Amsterdam ; Boston : San Francisco, CA: Elsevier ; Morgan Kaufmann, 2006.
- [18] V. M.K et K. K, « A Survey on Similarity Measures in Text Mining », *Mach. Learn. Appl. Int. J.*, vol. 3, n° 1, p. 19–28, mars 2016.
- [19] F. Soualah-Alila, F. Mendes, C. Cruz, et C. Nicolle, « Recommender System For Combination of Learning Elements in Mobile Environment », in *IADIS International Conference Mobile Learning*, 2012, p. 309–314.
- [20] D. Suliema et M. Malek, « Towards Semantic Social recommendation algorithm », in *Seconde conférence sur les Modèles et l'Analyse des Réseaux: Approches Mathématiques et Informatique MARAMI2011.* Du, vol. 19.
- [21] E. I. J. of E. D. and R. Ijedr, « A Survey on frequent pattern mining methods-Apriori, Eclat, FP growth », *Int. J. Eng. Dev. Res. IJEDR ISSN2321-9939*.