IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 9 (September. 2018), ||V (IV) || PP 38-43

User Preference Selection Mechanism using Markov Chain for a Cloud Service Provider

M. Satheesh[1], M. Aramudhan[2]
[1.] *Department of Computer Science, Don Bosco College Arts and Science*
[2.] *Department of Information Technology, Perunthalaivar Kamarajar Institute of Engineering and Technology(PKIET)*
Corresponding Author: M. Satheesh

Abstract: Cloud Service selection has become a challenging decision for many organization due to the increasing number of services and user constantly changes his preferences based in the required and expected level of satisfaction of the experienced cloud services. Hence we propose the work to overcome this drawback and to develop a cloud broker architecture for cloud service selection by finding a pattern of the changing priorities of the User Preferences (UPs). To achieve this goal we have adapted the Markov chain to find the changing patter. The identified pattern is then connected to the Quality of Service (QoS) and the Best Worst Method (BWM) is used to rank the service from the available cloud services. The proposed methodology provides a prioritized list of the services based on the pattern of changing Ups. The findings are validated through a real quality of service performance data of Amazon EC2 cloud Services.

Keywords: Cloud Service Provider(CSP), Markov chain, Service selection, AHP Method, Best Worst method(BWM)

Date of Submission: 13-09-2018 Date of acceptance: 28-09-2018

I. INTRODUCTION

Cloud computing is defined a s a model that enables ubiquitous, convenient and on-demand network access to a shared pool of configurable computing resources. As the number of cloud services is constantly growing, a user is exposed to many choices to select the right service from the right cloud provider at the right time. A proposed clod broker Architecture gives a ranked list of potential cloud services in order to reduce the complexity of service selection for the cloud user. For instance, Equinix, Amazon, etc.. which provides different types of cloud services employing various cloud service selection algorithms, yet, non of these take into account the changing needs of the cloud user for best service recommendation.

Traditionally, cloud service selection is considered the process of finding the most suitable cloud service by matching the functional and quality of service(QoS) requirements of a user, later on focusing on the criteria to be considered as performance, cost and efficiency. But there is some uncertainties which hinders the selection of the service providers due to the uncertainty of the User Preferences which solely depends on the user requirement and level of satisfaction with the service. The level of satisfaction may change over a period of time during service consumption.

To address this problem we can collect the changing preferences from the users to understand their satisfaction before and after using the recommended service. Then by grouping the users with the similar service requirements, their level of satisfaction can be analyzed to determine any pattern. This would help in recommending the selection of the most appropriate cloud service for other users having similar service requirements.

II. METHODOLOGY

A special cloud broker architecture taking into account the changing Ups over time has been introduced. Using a Markov chain to capture and track the changes in UPs and find transition patterns. The identified patterns are then used for recommendation and selection of the suitable services. Finally, the Best Worst Method(BWM) which is a multi criteria Decision Making(MCDM) method is used to rank the services.
The Main emphasis of this method is

1. By using a Markov chain, capture the model in changes to find the transitioning patterns
2. Using Markov chain combined with BWM to rank the services based on changing user preferences.
3. We evaluate our proposed model on real cloud services data for grater consistency

The paper is organized in different sections as follows. In section 2 we discuss the related works and present an overview of some of these studies. In section 3 we develop an Algorithm to implement the Markova chain method for finding the best services. Section 4 deals with implementation of the proposed method. In section 5 deals with evaluation and finally section 6 concludes with notable observations.

III. RELATED WORKS

In recent years a number of research efforts have been on cloud service selection based on QoS. The Quality based approach consider the criteria for decision making. Cloud Service Selection proposed by Rehman, Hussain – Making use of the QoS history over different time periods, ranks the service using Multi Criteria Decision Making(MCDM) in each time period and aggregates the results to determine the overall rank for service available options Han, Yoon, Lee and Huh – the QoS is considered by assigning weights using logistic decay function. This method selects the best combination of services from different cloud providers by maintaining a record of all available resources in the market and ranks on the basis of QoS values.

Zeng and Zhao designed a cloud service selection algorithm using the maximized-gain and minimized-cost approach. Based on the user request, the service selection algorithm aggregates the gain and cost values by a weighted sum of relative importance of involved factors. Garg, Versteeg and Buyya proposed SMI Cloud which is based on Service Measurement Index(SMI) for comparing and ranking cloud services on the SMI criteria. It measures all the QoS attributes in SMI and then uses Analytic

Hierarchy Process (AHP) using QoS criteria such as usability, functionality, Scalability, cost, vendor Reputation etc. Ghosh and Members presented a framework for selection based on the Risk estimation based on trustworthiness and competence.

The non-Quality of Service based approaches define and measure different attributes and metrics for cloud service selection. LI Yang, Kandula and Zhang discussed the cloud service selection problem and identified the basic attributes for each type of cloud services(such as IaaS, SaaS) that must be taken into consideration when comparing one cloud service with another. Kang and Sim developed a semantic based cloud service search engine called Cloudle using a cloud ontology. The proposed system maintains a record of all the service in a database. The user search query is sent to an engine that performs similarity reasoning between the query and the concepts in the database using cloud ontology. The output of the Cloudle search engine is an ordered list of cloud services based on concept of similarity, price, and cost utility.

Wang, Cao and Ziang proposed a dynamic cloud service selection(DCS) strategy, which consists of a set of dynamic service selection algorithms, uses an adaptive learning mechanism that comprises the incentive, forgetting and degenerate functions. Nie, She and Chen proposed an evaluation system of cloud service selection using AHP that calculates the weights of attributes for service evaluation. They also presented a number of qualitative models for decision making in cloud service selection. Goash and members presented a framework to facilitate cloud service selection that calculates the risk estimation based on trustworthiness and competence. Capuano, Chiclana, Fujita and Loia proposed a model to consider the real preference of an expert whom is influenced by the opinion of other experts. In an unassumable preferences they employee a user friendly fuzzy ranking model to obtain the preferences. Zhanga, Dong divided decision makers into two different clusters. Individual preference vectors are obtained and a feedback adjustment process is utilized to help decision makers adjust their preferences

It is in this frequent and continuous change of user preferences this paper performs that through a combined application of Markovian Model and BWM, makes an attempt to find a best solution in finding best cloud services. We use a general version of Markov chain which is applicable to trace changing priorities of cloud customers.

IV. PROPOSED METHOD

The Markov chain generates a pattern of the changing priorities of user preferences. This pattern is then used as the input for BWM to find out the priorities of QoS criteria. The QoS priorities are then used to rank the services. First we need to obtain the initial Ups through service queries on the cloud broker. The changes can be traced using a Markov chain to predict a pattern of customer needs. The MCDM method utilizes the Markov decision process, where if it finds a chaining user preferences, then next step would be to utilize the captured information for cloud service selection. In this selection, alternatives are ranked again the criteria by using MCDM method. The Markov chain does not directly make the decision, instead the power of the Markov chain is leveraged to help BWM to connect the importance of these elements to cloud service specifications and rank the alternatives

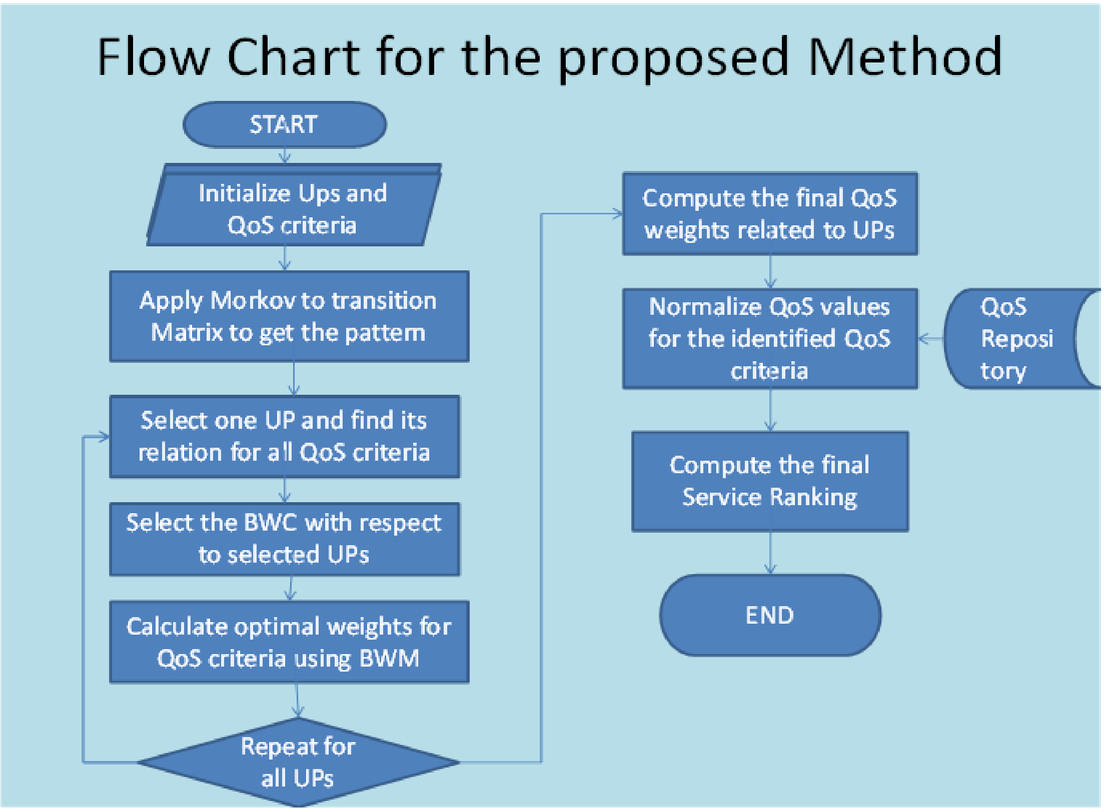
The Markov chain addresses the problem of Ups being discrete events and finds a pattern in them. Although the Markov chain finds the pattern to be utilized instead of the initial preferences this does not mean that users may not change priorities. Every user may keep changing their priorities. However, if these changes are traced in terms of the whole system including many other users for similar services, a pattern for the required specifications can be observed. In such pattern there will be a certain number of users with each of the preferences. For example some users may change to low priority for a certain preference while other users may change to high priority for the same preference. Those numbers are computable using the Markov chain as Follows.

4.1 Algorithm

STEP 1- Get the required transition Matrix of initial User Preferences and QoS Criteria
STEP 2 - Apply Markov chain to transition Matrix and initial Ups to get the pattern of Changing Ups.
STEP 3 - For all 1 to N user preferences
Select one UP and find its relation with all the QoS criteria
STEP 4 – Select the best and worst QoS criteria with respect to selected UP
STEP 5 – Calculate Optimal weights for QoS criteria using BWM
STEP 6 – Repeat until N Ups
STEP 7 – Compute the final QoS weights considering their relation to Ups
STEP 8 – Normalize QoS values for the identified QoS for the QoS Repository
STEP 9 – Compute the final service Ranks
STEP10- END

NOTE:

In this selection, alternatives are ranked again the criteria by using MCDM method. The markov chain does not directly make the decision, instead the power of the Markov chain is leveraged to help BWM to connect the importance of these elements to cloud service specifications and rank the alternatives



4.2 Algorithm -2 Best Worst Method

It allows decision maker to compute the weights when the comparisons are either fully consistent or partially consistent. If the comparisons are inconsistent, it allows decision maker to re-adjust the comparisons. For partially consistent, we propse using a threshold value between 0 and max ξ that allows decision maker to adjust the consistency according to the problem.

Algorithm -2 Best Worst Method
BWM(UP,Q)
1. C1 = Consistency Index Table;
2. Choose Best Criterian abb where abb € Q
3. Chose worst criterion aww € Q
4. J=1;
5. Set aBw = x where x ϵ [1,9]
6. While (j≤count(Q))
7. Do
8. Set aBj = x where x ϵ [1,9];
9. Set ajw = x where x ϵ [1,9];
10. While (aBw = aBj and aBw = ajw);
11. AB. add(aBj);
12. Aw. add(ajw);
13. **End While**
14. **For** all *j in Q*
15. Form equation of the form|*WB – Wj *aBj*|where aBj ϵ AB;
16. Form equation of the form|*Wj – Ww *ajw*|where aBj ϵ AB;
17. **End for**
18. Form equation Σj **wj**=1
19. Compute **wj** for UP(row) in Matrix Wup –Qos by solving system of linear equations
20. Return Wup –Qos ;

NOTE:
In this selection, alternatives are ranked again the criteria by using MCDM method. The markov chain does not directly make the decision, instead the power of the Markov chain is leveraged to help BWM to connect the importance of these elements to cloud service specifications and rank the alternatives.

4.3 SAMPLE ANALYSIS
Let us assume that the initial priority list of Ups is obtained through interviews and normalized in matrix(1)

$$W_{up} = \begin{matrix} s1 \\ s2 \\ : \\ sn \end{matrix} \begin{bmatrix} a1 \\ a2 \\ \vdots \\ an \end{bmatrix}$$

Consider a time set, T = { t_1 ... t_m}. There is always a likelihood of changing the preference from one UP to another after a period of time. For example, at time t_1 , a_i is greater a_j than , which means more users prefer and select (UP compared with (UP) as the most important requirement. At time , UPs may change and may not remain more than . The time interval depends on how regularly users utilize the service and this varies for different service categories. Observing user behavior over time, we can see the proportion of users who have UP1 as their most important need and so wish to stay with UP1 or shift to other UPs (e.g. UP2, UP3...). Figure 2 is an example showing the transitions where there are four UPs (S1 to S4).

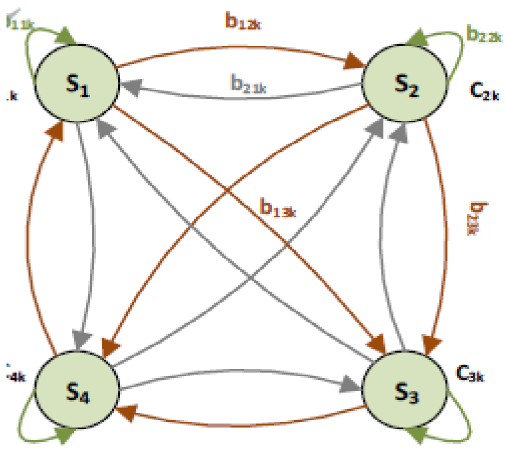


Figure 2. Transitioning between states at kth period

In the above figure, C_{ik} stands for the number of users that are with S_i at period k (S_{ik}), and b_{ijk} represents the number of users that transition from S_i to S_j (changing from one UP to another as their priority). In order to compute the transition matrix, we need to calculate the probability of transitioning from to at kth period for every UP. The probabilities are computed as follows. If there are users who prefer S1 at time k, and are those who prefer to transition to S2, then the probability of this transition is computed as:

More generally, the probabilities are computed as follows. If, after a sequence of periods, a P_{ij} can be estimated for which the following condition exists, then the value of in the transition matrix is equal to (note that stands for a small value). In many cases, finding a that stays within the narrow interval of ( ) for a large number of successive periods may not be applicable. In such cases, different amounts for ***Pi*** can instead be found and used for a reasonable number of periods. To cope with such a situation, a number should be set by decision makers.

The transition matrix is computed as follows

S1 S2 ---- Sn

$$P = \begin{bmatrix} P11 & P12 & P1n \\ P21 & P22 & P2n \\ Pn1 & \cdots & Pnn \end{bmatrix}$$

By frequent multiplication of the transpose of matrix by the transition matrix, P, a set of is obtained, where i is the number of the multiplications. **wj** for UP(row) in Matrix Wup –Qos.

A generalized form of the formula is presented in Form equation form $|WB – Wj *aBj|$ where aBj ϵ AB Considering the inherent convergence of the stochastic matrices, we expect that the matrices become the same after three to five times of multiplication. Since the adjusted priorities of UPs are independent of the initial state [5], this method stands independent of the initial priorities of UPs. Thus, rather than the identification of the users‟ initial (instant) preferences, there should be a focus on forming a transitional matrix.

When the number of successive periods that stays in the interval and goes beyond the number, the transition matrix is computed and will be in use until a new trend appears. Techniques and charts of statistical quality control (SQC) can be utilized for the purpose of recognition of the trends, detection of the points of shifting, and monitoring probabilities in the transition matrix. The Matrix below is a standard form since the absorbing states A and B precede the non-absorbing state C. The general standard form matrix P is listed on the right in which the matrix P is partitioned into four sub-matrices I,O,R and Q where I is an identity matrix and O is the zero Matrix.

$$P = \begin{bmatrix} Performance & 1 & 0 \\ Reliability & 1.5 & 1 \\ \text{Price} & 1 & 0.5 \end{bmatrix}$$

1	0	0
0	1	0
0.1	0.4	0.5

where $P = \begin{bmatrix} I & 0 \\ R & Q \end{bmatrix}$

The Transistion matrix value can be arrived at $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.134 & 0.45 & 0.523 \end{bmatrix}$

In summary, when the sequence is relatively close to , the value of is estimated by . Since the

transition matrix includes values presented by , we do not expect a considerable effect for minor changes. In the study presented in the paper, we use a threshold of 0.1. If the absolute variation of the probabilities in the transition matrix divided by its order goes beyond 10 percent, calculation of the transition matrix is required.

Thus based on QoS criteria of the available services and the relevant computations, it turn out that taking the feedback of previous users into account for this service category, the most appropriate secice. The trasition matrix is the study transforms the user behavious for most preferred criteria from performance to Reliability overtime. That means the Ups change from Performance to Reliability overtime for services in the selection category.

V. CONCLUSION

In this paper work we have discussed the cloud service selection problem in an environment where the priorities of users keep changing. We proposed a framework that finds a pattern of changing Ups using a Markov chain independent of the initial user preferences. The pattern is then linked to QoS criteria of all available services to find weights using the BWM. The weights of criteria of all services are then used to determine the overall rank of options for cloud service selection. The study of the test samples show that utilizing the previous users' experience and feedback produces more suitable service recommendation and selection for future cloud services usage. In the future it can me further examined and developed the applicability of other methods in combination with BWM to address the concerns of uncertainty in the decision process in the cloud service selection.

REFERENCES

IOSR Journal of Engineering (IOSRJEN) is UGC approved Journal with Sl. No. 3240, Journal no. 48995.

M. Satheesh1" User Preference Selection Mechanism using Markov Chain for a Cloud Service Provider" IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 9, 2018, pp. 38-43.