A NOVEL FRAMEWORK FOR PREDICTION OF QOS RANKING FOR CLOUD SERVICES

Venkata Murali D[1], Ranjith[2]
[1]M.Tech in SE Dept, [2]Assistant Professor in CSE Dept,
Warangal Institute of Technology and Science, Warangal, Telangana, INDIA

ABSTRACT: Building prime quality cloud applications becomes associate degree directly needed analysis drawback in cloud computing technology. Non-functional performance of cloud services is mostly delineated by Quality-of-Service (QoS). to amass QoS values, real-world usage of services candidates ar typically needed. At this point, there's no framework which will enable users to estimate cloud services and rank them supported their QoS values. This paper intends to framework and a mechanism that measures the standard and ranks cloud services for the users. CloudRank framework by taking the advantage of past service usage experiences of alternative users. thus it will avoid the time overwhelming and dearly-won reality service invocation. this system determines the QoS ranking directly victimisation the 2 customized QoS ranking prediction approach particularly, CloudRank1 and CloudRank2. These algorithms make certain that the active services ar properly stratified. The core determination is ranking prediction of consumer facet QoS properties, that seemingly have completely different values for dissimilar users of identical cloud service. It estimates all the individual services at the user-side and rank the services supported the determined QoS values.

Keywords: cloud services, CloudRank, Quality-of-service, ranking prediction

I. INTRODUCTION

Cloud computing is Internet-based computing, whereby shared configurable resources (e.g., infrastructure, platform, and software) ar provided to computers and alternative devices as services. powerfully promoted by the leading industrial corporations (e.g., Amazon, Google, Microsoft, IBM, etc.), cloud computing is quickly changing into fashionable in recent years. Applications deployed within the cloud surroundings ar usually massive scale and complicated. With the rising quality of cloud computing, the way to build high-quality cloud applications becomes associate desperately needed analysis downside. the same as ancient component-based systems, cloud applications usually involve multiple cloud elements human action with one another over application programming interfaces, like through net services. On-functional performance of cloud services is sometimes delineate by quality-of-service (QoS). QoS is a very important analysis topic in cloud computing. once creating optimum cloud service choice from a group of functionally equivalent services, QoS values of cloud services give valuable info to help deciding. In ancient component-based systems, software system elements ar invoked regionally, whereas in cloud applications, cloud services ar invoked remotely by net connections. Client-side performance of cloud services is so greatly influenced by the unpredictable net connections. Therefore, completely different|completely different} cloud applications might receive different levels of quality for a similar cloud service. In alternative words, the QoS ranking of cloud services for a user can't be transferred on to another user since the locations of the cloud applications ar quite totally different. customized cloud service QoS ranking is so needed for various cloud applications. the foremost easy approach of customized cloud service QoS ranking is to guage all the candidate services at the user-side and rank the services supported the ascertained QoS values. However, this approach is impractical truly, since invocations of cloud services is also charged. although the invocations ar free, death penalty an outsized range of service invocations is time intense and resource intense, and a few service invocations might turn out irreversible effects within the globe. Moreover, once the quantity of candidate services is massive, it's troublesome for the cloud application designers to guage all the cloud services with efficiency. To attack this essential challenge, we have a tendency to propose a personalised ranking prediction framework, named Cloud Rank, to predict the QoS ranking of a group of cloud services while not requiring extra real-world service invocations from the supposed users. Our approach takes advantage of the past usage experiences of alternative users for creating customized ranking prediction for this user. Extended from its preliminary conference version, the contribution of this paper is twofold: This paper identifies the essential downside of customized QoS ranking for cloud services and proposes a QoS ranking prediction framework to handle the matter. To the most effective of our data, Cloud Rank is that the 1st customized QoS ranking prediction framework for cloud services. in depth real-world experiments ar conducted to review the ranking prediction accuracy of our ranking prediction algorithms compared with alternative competitory ranking algorithms. The experimental results show the effectiveness of our approach. we have a tendency to in public unharness our service QoS knowledge set1 for future analysis, that makes our experiments duplicatable. additional correct ranking prediction results will be achieved by providing QoS values on additional cloud services, since the characteristic of the active user will be mined from the provided knowledge. among the Cloud Rank framework, there ar many modules. First, supported the user-provided QoS values, similarities between the active user and coaching users will be calculated. Second, supported the similarity values, a group of comparable users will be known. After that, 2 algorithms ar planned (i.e., Cloud Rank1 and Cloud Rank2) to form customized service ranking by taking blessings of the past service usage experiences of comparable users. Finally,the ranking prediction results ar provided to the active user. The coaching knowledge within the Cloud Rank framework will be obtained from: one. The QoS values provided by other users 2. The QoS values collected by monitoring cloud services. In our previous work, a user-collaborative mechanism is proposed for collecting client-side QoS values of web services from different service users. The observed web service QoS values can be contributed by users by running a client-side web service evaluation application. Different from service-oriented applications, the usage experiences of cloud services are much easier to be obtained in the cloud environment. The cloud applications can invoke and record the client-side QoS performance of the invoked cloud services easily by using monitoring infrastructure services provided by the cloud platform. The cloud provider can collect these client-side QoS values from different cloud applications easily with approval of application owners. The framework can be used at both design time and runtime. At runtime, the cloud application may obtain new QoS values on some cloud services. By providing these values to our Cloud Rank server, new QoS ranking prediction can be obtained. Based on the service QoS ranking, optimal system reconfiguration can be achieved.

II. ARCHITECTURE

The CloudRank framework provides optimal service selection from the more number of equivalent functionalities. Quality-of-service can be measured at the server side or at the client side. Client-side QoS properties provide more realistic measurements of the user usage experience. The generally used client-side QoS properties include response time, throughput, failure probability, etc. the system architecture of, which provides personalized QoS ranking prediction for cloud services.

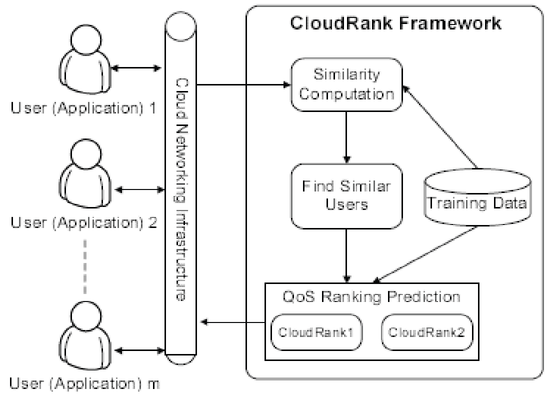


Fig 1: Architerure

III. METHODOLOGY

A. Similarity Computation

Step 1: Initialization:

a) Set n to be the length of s, set m to be the length of t.

b) Construct a matrix containing 0..m rows and 0..n columns.

c) Initialize the first row to 0..n,

d) Initialize the first column to 0..m.

Step2: Processing

a) Examine s (i from 1 to n).

b) Examine t (j from 1 to m).

c) If s[i] equals t[j], the cost is 0.

d) If s[i] doesn't equal t[j], the cost is 1.

Step 3: Result

Step 2 is repeated till the sim(u,v) value is found The user request is consider as a string, then the string will be evaluated by row and column-wise in matrix formation. The row is referred by user request and the column is referred by resides services in the training datasets. S[i] is equal to t[j] means cost is zero, else cost is one. The process is going until d[n] is found. Ranking similarity computations compare users QoS rankings on the commonly invoked services. Suppose we have a set of three cloud services, on which two users have observed response-times (seconds) of {1, 2, 4} and {2, 4, 5}, respectively. The response-time values on these services observed by the two users are clearly different nevertheless; their rankings are very close as the services are ordered in the same way. Given two rankings on the same set of services, the Kendall Rank Correlation Coefficient (KRCC) evaluates the degree of similarity by considering the number of inversions of service pairs which would be needed to transform one rank order into the other. The KRCC value of user's u and v can be calculated by

$$Sim(u,v) = \frac{C - D}{N(N-1)/2},$$

B. Cloud Rank1

A user's preference on a pair of services can be modelled in the form of ,where means that quality of service I is better than j and is thus more preferable for the active user. The value of the preference function indicates the strength of preference and a value of zero means that there is no preference between two services.

Algorithm 1 includes the following steps:

 Step 1 (lines 1-6). Rank the employed cloud services in E based on the observed QoS values. stores the ranking, where t is a cloud service and the function returns the corresponding order of this service. The values of are in the range of [1,|E|], where a smaller value indicates higher quality.

 Step 2 (lines 7-9). For each service in the full service set I, calculate the sum of preference values with all other services by . Since =0 including . in the calculation does not influence the results. Larger value indicates more services are less preferred than i. In other words, service i should be ranked in a higher position.

 Step 3 (lines 10-18). Services are ranked from the highest position to the lowest position by picking the service t that has the maximum value. The selected service is assigned a rank equal to n -|I|+1 so that it will be ranked above all the other remaining services in I. The ranks are in the range of [1,n], where n is the number of services and a smaller value indicates higher quality. The selected service t is then removed from I and the preference sum values of the remaining services are updated to remove the effects of the selected service t.

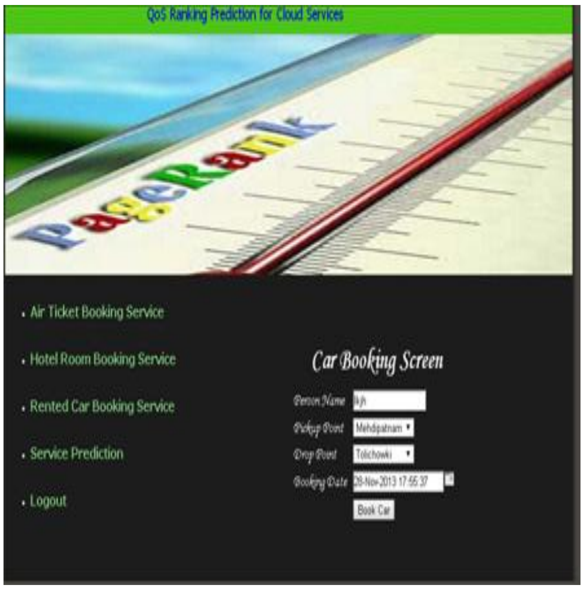
 Step 4 (lines 19-24). Step 3 treats the employed services in E and the nonemployed service in I -E identically which may incorrectly rank the employed services. In this step, the initial service ranking is updated by correcting the rankings of the employed services in E. By replacing the ranking results in with the corresponding correct ranking of , our approach makes sure that the employed services in E are correctly ranked. The preference values in the CloudRank1 algorithm can be obtained explicitly or implicitly. When the active user has QoS values on both the services i and service j, the preference value is obtained explicitly. On the other hand, the preference value is obtained implicitly when employing QoS information of similar users. Assuming there are three cloud services a, b, and c. The active users have invoked service a and service b previously. The list below shows how the preference values of can be obtained explicitly or implicitly : obtained explicitly.

 : obtained implicitly by similar users with similarities of 0.1, 0.2, and 0.3.

 : obtained implicitly by similar users with similarities of 0.7, 0.8, and 0.9.

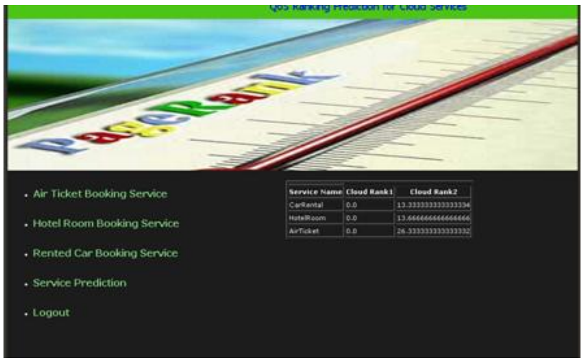
In the above example, we can see that different preference values have different confidence levels. It is clear that C (a,b) >C (b,c) >C (a,c), where C represents the confidence values of different preference values. The confidence value of is higher than , since the similar users of have higher similarities. In the CloudRank1 algorithm, differences in preference values are treated equally, which may hurt the QoS ranking prediction accuracy. By considering the confidence values of different preference values, we propose a QoS ranking prediction algorithm, named CloudRank2, in which confidence values can be calculated by, Where v is a similar user of the current active user u. wv makes sure that a similar user with higher similarity value has greater.

IV. EXPERIMENT RESULTS

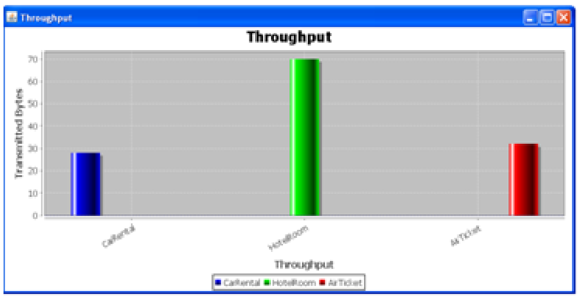

QoS Ranking Prediction for Cloud Services
PageRank
Air Ticket Booking Service
Hotel Room Booking Service
Rented Car Booking Service
Service Prediction
Logout
Book Car

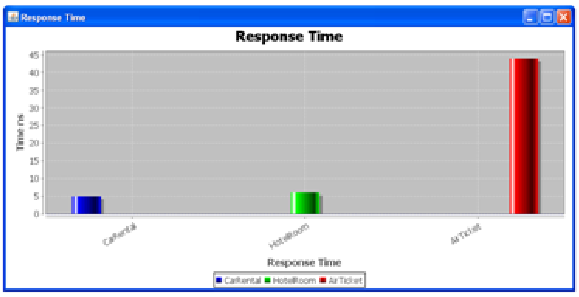
Car Booking Screen

Person Name	kjh
Pickup Point	Mehdipatnam
Drop Point	Tolichowki
Booking Date	28-Nov-2013 17:55:37


QoS Ranking Prediction for Cloud Services
PageRank
Air Ticket Booking Service
Hotel Room Booking Service
Rented Car Booking Service
Service Prediction
Logout

Service Name	Cloud Rank1	Cloud Rank2
CarRental	0.0	13.3333333333333336
HotelRoom	0.0	13.6666666666666666
AirTicket	0.0	26.3333333333333332





V. CONCLUSION

Personalized QoS ranking prediction framework for cloud services, which requires no additional service invocations when making QoS ranking. By taking advantage of the past usage experiences of other users, our ranking approach identifies and aggregates the preferences between pairs of services to produce a ranking of services. We propose two ranking prediction algorithms for computing the service ranking based on the cloud application designer's preferences. Experimental results show that our approaches outperform existing rating-based approaches and the traditional greedy method.

VI. FUTURE WORK

For future work, like to improve the ranking accuracy of our approaches by exploiting additional techniques (e.g., data smoothing, random walk, matrix factorization, utilizing content information, etc.). When a user has multiple invocations of a cloud service at different time, we will explore time-aware QoS ranking prediction approaches for cloud services by employing information of service users, cloud services, and time. As our current approaches only rank different QoS properties independently, we will conduct more investigations on the correlations and combinations of different QoS properties. We will also investigate the combination of rating-based approaches and ranking-based approaches, so that the users can obtain QoS ranking prediction as well as detailed QoS value prediction. Moreover, we will study how to detect and exclude malicious QoS values provided by users.

REFERENCES

[1] M. Deshpande and G. Karypis, “Item-Based Top-n Recommendation,” ACM Trans. Information System, vol. 22, no. 1, pp. 143-177,2004.
[2] R. Jin, J.Y. Chai, and L. Si, “An Automatic Weighting Scheme for Collaborative Filtering,” Proc. 27th Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’04), pp. 337-344, 2004.
[3] Z.zheng, Y.Zhang, and M.R.Lyu,"CloudRank: A QoS-Driven Component Ranking Framework for Cloud Computing,"Proc.Int’l Symp. Reliable Distributed Systems (SRDS ’10), pp. 184-193, 2010.
[4] Z.Zheng and M.r.Lyu,"WS-DREAM: A Distributed Reliability Assessment Mechanism for Web Seervices,"Proc.38th Int'l Conf.Dependable Systems and Networks (DSN '08), pp.392-397, 2008.
[5] R. Burke, “Hybrid Recommender Systems: Survey and Experi- ments,” User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331-370, 2002.
[6] W.W. Cohen, R.E. Schapire, and Y. Singer, “Learning to order things,” J. Artificial Intelligent Research, vol. 10, no. 1, pp. 243-270, 1999.
[7] M. Deshpande and G. Karypis, “Item-Based Top-n Recommenda- tion,” ACM Trans. Information System, vol. 22, no. 1, pp. 143-177, 2004.
[8] P.A. Bonatti and P. Festa, “On Optimal Service Selection,” Proc.14th Int’l Conf. World Wide Web (WWW ’05), pp. 530-538, 2005.
[9] W.W. Cohen, R.E. Schapire, and Y. Singer, “Learning to order things,” J. Artificial Intelligent Research, vol. 10, no. 1, pp. 243-270, 1999.
[10] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “Grouplens: An Open Architecture for Collaborative Filtering of Netnews,”Proc. ACM Conf. Computer Supported Cooperative Work, pp. 175-186, 1994.
[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-Based Collaborative Filtering Recommendation Algorithms,” Proc. 10th Int’l Conf. World Wide Web (WWW ’01), pp. 285-295, 2001.