IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. VIII (Mar-Apr. 2014), PP 59-62
www.iosrjournals.org

Ranking of Cloud Services Using Dynamic Qos

Priyanka V[#1], Sabari venkatesh G[#1], Prithiviraj S[#1,] Christopher Paul A[#2]
[#1]*UG Scholar, Department of CSE, SNS College of Technology,Coimbatore, India*
[#2]*Asst. Professor, Department of CSE, SNS College of Technology,Coimbatore, India.*

Abstract: *Cloud computing is an internet based computing and is becoming popular. QoS rankings provide valuable information for making optimal cloud service selection from a set of functionally equivalent service candidates. To obtain QoS values, real-world invocations on the service candidates are usually required. To avoid the time-consuming and expensive real-world service invocation QoS ranking prediction framework is used. This framework requires no additional invocations of cloud services when making QoS ranking prediction. Two personalized QoS ranking prediction approaches are used to predict the QoS rankings directly. Comprehensive experiments are conducted employing real-world QoS data, including 300 distributed users and 500 real-world web services all over the world. The proposed uses modernized ranking approach which uses different QoS parameters to predict the ranking more accurately. Different QoS parameters like latency, availability, failure probability can be used to improve the ranking.*

Keyword: *Quality-of-service, cloud service, ranking prediction, personalization*

I. Introduction

Cloud computing is Internet-based computing, whereby shared configurable resources (e.g., infrastructure, platform, and software) are provided to computers and other devices as services. Strongly promoted by the leading industrial companies (e.g., Amazon, Google, Microsoft, IBM, etc.), cloud computing is quickly becoming popular in recent years. Applications deployed in the cloud environment are typically large scale and complex. With the rising popularity of cloud computing, how to build high-quality cloud applications becomes an urgently required research problem. Similar to traditional component-based systems, cloud applications typically involve multiple cloud components communicating with each other over application programming interfaces, such as through web services. On-functional performance of cloud services is usually described by quality-of-service (QoS). QoS is an important research topic in cloud computing. When making optimal cloud service selection from a set of functionally equivalent services, QoS values of cloud services provide valuable information to assist decision making. In traditional component-based systems, software components are invoked locally, while in cloud applications, cloud services are invoked remotely by Internet connections.

Client-side performance of cloud services is thus greatly influenced by the unpredictable Internet connections. Therefore, different cloud applications may receive different levels of quality for the same cloud service. In other words, the QoS ranking of cloud services for a user cannot be transferred directly to another user since the locations of the cloud applications are quite different. Personalized cloud service QoS ranking is thus required for different cloud applications. The most straightforward approach of personalized cloud service QoS ranking is to evaluate all the candidate services at the user-side and rank the services based on the observed QoS values. However, this approach is impractical in reality, since invocations of cloud services may be charged. Even if the invocations are free, executing a large number of service invocations is time consuming and resource consuming, and some service invocations may produce irreversible effects in the real world. Moreover, when the number of candidate services is large, it is difficult for the cloud application designers to evaluate all the cloud services efficiently. To attack this critical challenge, we propose a personalized ranking prediction framework, named Cloud Rank, to predict the QoS ranking of a set of cloud services without requiring additional real-world service invocations from the intended users. Our approach takes advantage of the past usage experiences of other users for making personalized ranking prediction for the current user. Extended from its preliminary conference version, the contribution of this paper is twofold: This paper identifies the critical problem of personalized QoS ranking for cloud services and proposes a QoS ranking prediction framework to address the problem. To the best of our knowledge, Cloud Rank is the first personalized QoS ranking prediction framework for cloud services. Extensive real-world experiments are conducted to study the ranking prediction accuracy of our ranking prediction algorithms compared with other competing ranking algorithms. The experimental results show the effectiveness of our approach. We publicly release our service QoS data set1 for future research, which makes our experiments reproducible.

Quality-of-service can be measured at the server side or at the client side. While server-side QoS properties provide good indications of the cloud service capacities, client-side QoS properties provide more realistic measurements of the user usage experience. The commonly used client-side QoS properties include

response time, throughput, failure probability, etc. This paper mainly focuses on ranking prediction of client-side QoS properties, which likely have different values for different users (or user applications) of the same cloud service. Fig. 1 shows the system architecture of our Cloud Rank framework, which provides personalized QoS ranking prediction for cloud services. The target users of the Cloud Rank framework are the cloud applications, which need personalized cloud service ranking for making optimal service selection. A user is called active user if he/ she is requesting ranking prediction from the Cloud Rank framework. As shown in Fig. 1, a user can obtain service ranking prediction of all available cloud services from the Cloud Rank frame- work by providing observed QoS values of some cloud services.

More accurate ranking prediction results can be achieved by providing QoS values on more cloud services, since the characteristic of the active user can be mined from the provided data. Within the Cloud Rank framework, there are several modules. First, based on the user-provided QoS values, similarities between the active user and training users can be calculated. Second, based on the similarity values, a set of similar users can be identified. After that, two algorithms are proposed (i.e., Cloud Rank1 and Cloud Rank2) to make personalized service ranking by taking advantages of the past service usage experiences of similar users. Finally, the ranking prediction results are provided to the active user. The training data in the Cloud Rank framework can be obtained from: 1.

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. System Architecture

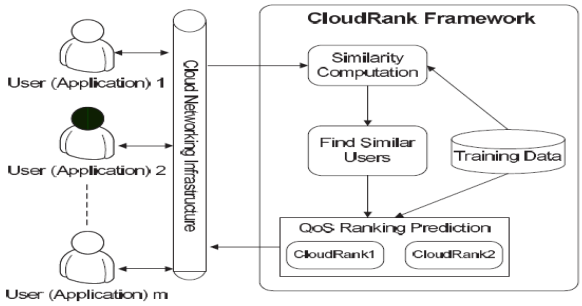

Fig.1 System architecture of cloud Rank

III. Qos RANKING PREDICTION

This section presents our Cloud Rank QoS ranking prediction framework for cloud services. Section 3.1 calculates the similarity of the active user with training users based on their rankings on the commonly invoked cloud services identifies a set of similar users presents two QoS ranking prediction algorithms, named Cloud Rank1 and Cloud Rank2, respectively analyses the computational complexity.

3.1Similarity Computation

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

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

Where N is the number of services, C is the number of concordant pairs between two lists, D is the number of discordant pairs, and there are totally N (N-1)/2 pairs for N cloud services. Since C=N (N-1)/2-D, (1) is equal to

Sim (u, v) =1-(4D/N (N-1)).Employing KRCC, the similarity between two service rankings can be calculated by,

$$Sim(u,v) = 1 - \frac{4 \times \sum_{i,j \in I_u \cap I_v} \tilde{I}((q_{u,i} - q_{u,j})(q_{v,i} - q_{v,j}))}{|I_u \cap I_v| \times (|I_u \cap I_v| - 1)},$$

Where Iu Iv is the subset of cloud services commonly invoked by users u and v, qu,i is the QoS value (e.g., response time, throughput, etc.) of service i observed by user u, and I(x) is an indicator function

3.2 Find Similar Users

By calculating similarity values between the current active user with other training users, the similar users can be identified. Previous approaches usually employ information of all the users for making ranking prediction of the current user, which may include dissimilar users. However, employing QoS values of dissimilar users will greatly influence the prediction accuracy.

3.3 QoS Ranking Prediction

Rating-oriented collaborative filtering approaches first predict the missing QoS values before making QoS ranking. The target of rating-oriented approaches is to predict QoS values as accurate as possible. However, accurate QoS value prediction may not lead to accurate QoS ranking prediction. For example, assume the expected response times of three services are 2, 3, and 5 seconds, respectively. There are two predictions using rating-oriented approaches: (3, 2, 4) and (1, 2, 3). Since rating-oriented approaches try to predict the QoS value as accurate as possible, Prediction 1 is better than Prediction 2, since it has a smaller MAE value. However, from the ranking-oriented perspective, Prediction1 is worse than Prediction 2 since the former leads to incorrect ranking based on the predicted QoS values. To address this problem, we propose two ranking- oriented approaches, named as Cloud Rank1 and CloudRank2, in the following. Our ranking-oriented approaches predict the QoS ranking directly without predicting the corresponding QoS values.

3.4 Cloud Rank1

A user's preference on a pair of services can be modelled in the form of $\varphi: I * I \rightarrow IR$,where $\varphi(i,j) > 0$ 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 $\varphi(i,j)$ 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.$\rho e(t)$stores the ranking, where t is a cloud service and the function $\rho e(t)$returns the corresponding order of this service. The values of $\rho e(t)$ 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 $\pi(i) = \varphi(i.j)$. Since $\varphi(i.j)$=0 including $\varphi(i.i) = 0$. in the calculation does not influence the results. Larger π(i) 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 π(t) 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 φ(i) 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 ρe, our approach makes sure that the employed services in E are correctly ranked.

The preference values φ(i, j) 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 φ(a, b), $\varphi(a,c), \varphi(b,c)$ can be obtained explicitly or implicitly

- $\varphi(\mathrm{a},\mathrm{b})$: obtained explicitly.
- $\varphi(a,c)$: obtained implicitly by similar users with similarities of 0.1, 0.2, and 0.3.
- $\varphi(b,c)$: 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 $\varphi(b,c)$ is higher than$\varphi(a,c)$, since the similar users of $\varphi(b,c)$ 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,

$$C(i,j) = wvSim(u,v)$$

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. Conclusion And Future Work

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.

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. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report EECS-2009-28, Univ. California, Berkeley, 2009.

[2] K.J. arvelin and J. Kekalainen, "Cumulated Gain-Based Evaluation of IR Techniques," ACM Trans. Information Systems, vol. 20, no. 4, pp. 422-446, 2002.

[3] P.A. Bonatti and P. Festa, "On Optimal Service Selection," Proc. 14th Int'l Conf. World Wide Web (WWW '05), pp. 530-538, 2005.

[4] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proc. 14th Ann. Conf. Uncertainty in Artificial Intelligence (UAI '98), pp. 43-52, 1998.

[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.