International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Efficient Algorithm for Predicting QOS in Cloud Services

Sangeeta R. Alagi[1], Srinu Dharavath[2]

[1]ME Computer Engineering, G. S. MOZE College of Engineering Balewadi, Pune, Maharashtra – India

[2]Assistant Professor, Computer Engineering, G. S. MOZE College of Engineering Balewadi, Pune, Maharashtra – India

Abstract: ***Cloud computing model enables accessing to information resources in request time. On the other hand, there are different cloud Service providers which present services with different qualitative characteristics. Determining the best cloud computing service for a specific application is a serious problem for users. Ranking compares the different services offered by different providers based on quality of services, in order to select the most appropriate service. In this paper, the existing approaches for ranking cloud computing services are analyzed. The overall performance of each method is presented by reviewing and comparing of them. The essential features of an efficient rating mechanism are indicated. Cloud computing is becoming valuable now a days. Make high-performance based cloud applications is a critical research problem. QoS rankings provide essential information for making optimal cloud service selection from a set of functionally equivalent service candidates. For getting QoS values, real-world invocations on the service candidates are usually essential. To avoid the time-consuming factors and expensive real-world services invocations, this paper develop a QoS ranking prediction architecture for cloud services by taking advantage of the past service usage experiences of other consumers. Our proposed architecture requires no additional invocations of cloud services when making QoS ranking prediction. Here we develop two personalized QoS ranking prediction approaches are proposed to predict the QoS rankings directly. Comprehensive experiments are conducted for performance analysis. The experimental results show that our approach performance is efficient than other competing approaches.***

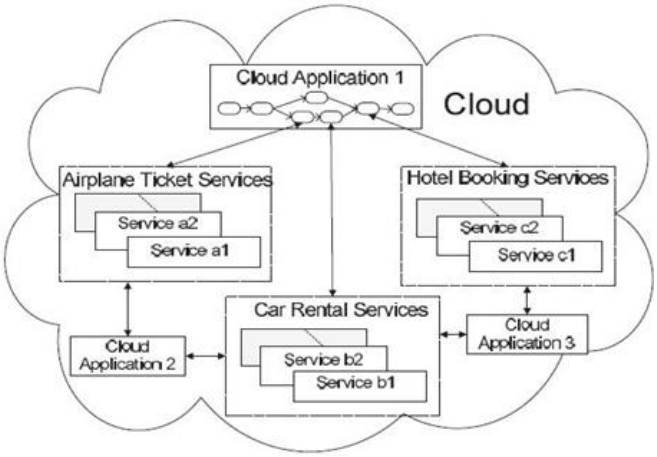
Keywords: Quality-of-Service, Cloud Service, Ranking Prediction, Personalization, Cloud Computing, Cloud Service Provider, Cloud Architecture.

1. Introduction

Cloud Computing approach provides services and delivers on demand resources request on time. It can be considered as the next utility required for human in cloud computing world. In this cloud computing environment each user has its own unique requirement and demand. That's why, selecting the excellent service that fulfills user's application requirements is an important research challenge now a days. The usage of service usually determines the success of its application infrastructure in cloud computing era. The provider's capability and performance cannot be fully utilized by selecting wrong services. The quality of service (QoS) information is required in service comparison with other methods. This information can be measured by providers or a third party vendor. Some attributes like response time, delay, usability, privacy and availability are defined for preparing quality of service information in cloud computing. The value of these attributes represents degree and performance of quality of services. The objective of ranking of services is helping users to evaluate and compare different services with others. So, users can select the most appropriate service that satisfies their requirement on demand. This paper describes service rankings in two ways. Firs part describe evaluation and comparison of services and the second part evaluate service ranking in cloud computing .Here we compare the existing computing models that has a dedicated infrastructure, cloud computing has the benefit of saving money and time. The cloud users need not to bother about the large cost of purchasing hardware, installing and to see through peak times of resource on demand and as per requirements. The main benefit of cloud computing is the method of paying only the amount of service used and resources can also be scaled according to the need and requirement. There is no any problem to the customer for the maintenance of servers and other components installation. The companies that are providing cloud services which are Rackspace, Microsoft, Amazon, and Google. Different providers offer a large variety of services and quality which offers services at different price, different level of performance and different features. Different providers also have different rates for the same services in the market. Each provider has its own different personal way of performing and measurement of prices thus makes it difficult for the customer to find out the best one in the market [2]. ution of this paper is very important this paper identifies the critical problem of personalized QoS ranking for cloud services and proposes a QoS ranking prediction model to address the problem. To the best of our knowledge, Cloud Rank is the first personalized QoS ranking prediction framework for cloud services now days. Extensive real-world experiments are conducted to study the ranking prediction accuracy of our ranking prediction algorithms compared with other competing ranking algorithms for getting better results and better performance. The experimental results show the effectiveness of our approach in detail manner.

Quality-of-service can be measured at the server side or at the client side both ways. While server-side QoS properties provide good indications of the cloud service capacities, client-side QoS properties provide more realistic results 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 of the same cloud service.

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 in better way.



This paper identifies the critical problem of personalized QoS ranking for cloud services and proposes a QoS ranking prediction framework to address the problem of the cloud computing. To the best of our knowledge, CloudRank is the first personalized QoS ranking prediction approach 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 in the cloud computing. The experimental results show the effectiveness of our model.

System Architecture

QoS properties provide excellent indications of the cloud service capacities. QoS can be measured at both the server side and client side, where client side QoS properties provide more correct results of the past usage experience which includes response time, throughput, failure rate, etc. This paper mainly emphasis on ranking prediction of QoS properties on the client side that may differ for the user application for same service. QoSRank is made for the cloud applications which are entirely used for optimal service selection and to achieve the correct result. The user is called as active user, where user requests the ranking prediction from the QoSRank framework. User can obtain service ranking prediction of all accessible cloud services from the QoSRank framework by providing observed QoS values of some cloud services now days. The results may be more accurate if the results are achieved by providing QoS values and personalization on cloud services its very valuable.The characteristic of the active user can be obtained from the data provided. Quality-of-service can be achieved at the server side or at the client side in both ways.. Within the CloudRank model, 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 in cloud, a set of similar users can be identified. After then, two algorithms are proposed (i.e., CloudRank1 and CloudRank2) to make personalized service ranking by taking advantages of the past service usage experiences of similar users for achieve better results. Finally, the ranking prediction results are provided to the active user. The training data in the CloudRank framework can be obtained from: 1) the QoS values provided by other users; and 2) the QoS values achieved 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 in cloud computing. The observed web service QoS values can be contributed by users by running a client-side web service evaluation application now days. Different from service-oriented applications, the usage experiences of cloud services are much easier to be obtained in the cloud environment in very efficient way. The cloud applications can 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 gather these client-side QoS values from different cloud applications easily with approval of application owners.

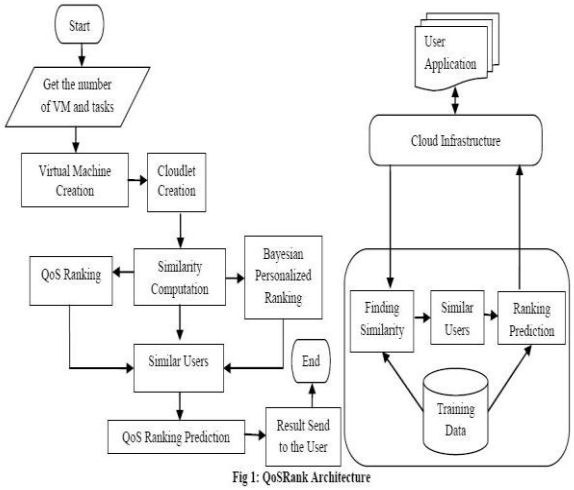


Fig 1: QoSRank Architecture

QoS Ranking prediction

CloudRank QoS ranking prediction model for cloud services. This Section calculates the similarity of the active user with training users based on their rankings on the generally invoked cloud services. It identifies a set of similar users. It presents two QoS ranking prediction algorithms, named CloudRank1 and CloudRank2, respectively. It analyzes the computational complexity and Similarity Computation.

Ranking similarity computations compare users' QoS rankings on the commonly invoked services in the cloud. Suppose we have a set of three cloud services, on which two users have measure 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, 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) [14] evaluates the degree of similarity by considering the number of inversions of service pairs which would be ne transform one rank order into the other.

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

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

$$Sim(u,v) = 1 - \frac{4 \times \sum_{i,j \in I_u \cap I_v} \tilde{I}((q_{u,i} - q_{u,j})(q_{v,i} - q_{v,j}))}{|I_u \cap I_v| \times (|I_u \cap I_v| - 1)}, \quad (2)$$

where $I_u \cap I_v$ is the subset of cloud services commonly invoked by users u and v, $q_{u,i}$ is the QoS value (e.g., response time, throughput, etc.) of service i observed by user u, and $\tilde{I}(x)$ is an indicator function defined as

$$\tilde{I}(x) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

From above definition, the ranking similarity between two rankings is in the interval of $[-1, 1]$, where -1 is obtained when the order of user u is the exact reverse of user v, and 1 is obtained when order of user u is equal to the order of user v. Since KRCC compares service pairs, the intersection between two users has to be at least 2 ($|I_u \cap I_v| \geq 2$) for making similarity computation.

Find Similar Users

By calculating similarity values between the current active users with other training users, the similar users can be-

Expected Response time: (2, 3, 5) on (s1, s2, s3)
Prediction 1: (3, 2, 4): MAE = (|2−3| + |3−2| + |5−4|)/3 = 1
Prediction 2: (1, 2, 3): MAE = (|2−1| + |3−2| + |5−3|)/3 = 1.3

Expected Ranking: s1 < s2 < s3
Prediction 1: s2 < s1 < s3
Prediction 2: s1 < s2 < s3

Fig. 3. Rating-oriented versus ranking oriented.

identified. Previous approaches [12], [18] 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. To address this problem, we exclude the users with negative correlations (negative similarity values) and only employ the Top-K similar users for making QoS ranking prediction. In our approach, a set of similar users $S(u)$ is identified for the active user u by

$$N(u) = \{v | v \in T_u, Sim(u,v) > 0, v \neq u\}, \quad (4)$$

where T_u is a set of the Top-K similar users to the user u and $Sim(u,v) > 0$ excludes the dissimilar users with negative similarity values. The value of $Sim(u,v)$ in 4 is calculated by (2).

QoS Ranking Predictions

Rating-oriented collaborative filtering approaches first predict the missing QoS values before making QoS ranking. The aim of rating-oriented approaches is to predict QoS values as accurate as possible. The accurate QoS value prediction may not lead to accurate QoS ranking prediction. There are two predictions using rating-oriented approaches: (3, 2, 4) and (1, 2, 3). Prediction 1 is better than Prediction 2, since it has a small MAE value (MAE refers to Mean Absolute Error, which is an evaluation metric for rating-oriented prediction results. Details of MAE will be introduced. However, from the ranking-oriented perspective, Prediction 1 is worse than Prediction 2 since the former leads to incorrect ranking based on the predicted QoS values. To solve this problem, we develop two ranking oriented approaches, named as CloudRank1 and CloudRank2. Our ranking-oriented approaches predict the QoS ranking directly without predicting the corresponding QoS values in cloud computing.

Cloud Rank Algorithm

We spend more valuable time to improving our online systems mechanism of cloud computing we believe in the right mix of great content and great technology to definitely change how you learn and how you get inside the cloud computing era. Look, this is not only about data, actually our Clod Rank team is working on several factors. We constantly change and improve it to serve better results to our users.

Cloud Rank1:

User's preference on a pair of services can be modeled in the form where v>0 means that quality of service i is better than service j and is thus more preferable for the active user and vice versa. The value of the preference function indicates the strength of preference and a value of zero means that there is no Given the user-observed QoS values on two cloud services, the preference between these two services can be easily derived by comparing the QoS values. To obtain the preference values regarding pairs of services that have not been invoked or observed by the current user, the preference values of similar users are employed. The basic idea is that the more often the similar users in observe service i as higher quality than service j, the stronger the evidence is of vj>0 for the current user. This leads to the following formula for estimating the value of the preference function, where service i and service j are not explicitly observed by the current user u where v is a similar user of the current u, is a subset of similar users, who obtain QoS values of both services i and j, and wv is a weighting factor of the similar user v, which can be calculated by Our goal is to produce a ranking that maximizes the above objective value function. One possible solution is to search through the possible rankings and select the optimal ranking that maximizes the value function defined in (7). However, there are n! possible rankings for n services, and the optimal ranking search problem is NP-Complete [6]. To enhance the calculation efficiently, we propose a greedy based algorithm in Algorithm 1 (named as CloudRank1) for finding an approximately optimal ranking. 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 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 including in the calculation does not influence the results. Larger value indicates more services are less preferred than 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 1 so that it will be ranked above all the other remaining services in I. The ranks are in the range of 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 e, our approach makes sure that the employed services in E are correctly ranked.

Algorithm 1: CloudRank1

Input: an employed service set E, a full service set I, a preference function Ψ
Output: a service ranking $\hat{\rho}$

1 $F = E$;
2 **while** $F \neq \emptyset$ **do**
3 $\quad t = \arg\max_{i \in F} q_i$;
4 $\quad \rho_e(t) = |E| - |F| + 1$;
5 $\quad F = F - \{t\}$;
6 **end**
7 **foreach** $i \in I$ **do**
8 $\quad \pi(i) = \sum_{j \in I} \Psi(i,j)$;
9 **end**
10 $n = |I|$;
11 **while** $I \neq \emptyset$ **do**
12 $\quad t = \arg\max_{i \in I} \pi(i)$;
13 $\quad \hat{\rho}(t) = n - |I| + 1$;
14 $\quad I = I - \{t\}$;
15 $\quad$ **foreach** $i \in I$ **do**
16 $\quad\quad \pi(i) = \pi(i) - \Psi(i,t)$
17 $\quad$ **end**
18 **end**
19 **while** $E \neq \emptyset$ **do**
20 $\quad e = \arg\min_{i \in E} \rho_e i$;
21 $\quad index = \min_{i \in E} \hat{\rho}(i)$;
22 $\quad \hat{\rho}(e) = index$;
23 $\quad E = E - \{e\}$;
24 **end**

Cloud Rank2:

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 and can be obtained explicitly or implicitly

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, which uses the following rules to calculate the confidence values: If the user has QoS values of these two services i and j. The confidence of the preference value is 1. When employing similar users for the preference value prediction, the confidence is determined by similarities of similar users where v is a similar user of the current active user u, is a subset of similar users, who obtain QoS values of both services i and j, and wv is a weighting factor of the similar user v, which can be calculated by wv makes sure that a similar user with higher similarity value has greater impact on the confidence calculation. Equation guarantees that similar users with higher similarities will generate higher confidence values. Algorithm 2 shows the details of the CloudRank2 algorithm, which considers the confidence values of different preference values when calculating the preference sum. In this way, more accurate ranking prediction can be achieved.

Algorithm 2: CloudRank2

Input: an employed service set E, a full service set I, a preference function Ψ, confidence values C
Output: a service ranking $\hat{\rho}$

1 $F = E$;
2 **while** $F \neq \emptyset$ **do**
3 $\quad t = \arg\max_{i \in F} q_i$;
4 $\quad \rho_e(t) = |E| - |F| + 1$;
5 $\quad F = F - \{t\}$;
6 **end**
7 **foreach** $i \in I$ **do**
8 $\quad \pi(i) = \sum_{j \in I} C(i,j) \times \Psi(i,j)$;
9 **end**
10 $n = |I|$;
11 **while** $I \neq \emptyset$ **do**
12 $\quad t = \arg\max_{i \in I} \pi(i)$;
13 $\quad \hat{\rho}(t) = n - |I| + 1$;
14 $\quad I = I - \{t\}$;
15 $\quad$ **foreach** $i \in I$ **do**
16 $\quad\quad \pi(i) = \pi(i) - C(i,j) \times \Psi(i,t)$
17 $\quad$ **end**
18 **end**
19 **while** $E \neq \emptyset$ **do**
20 $\quad e = \arg\min_{i \in E} \rho_e i$;
21 $\quad index = \min_{i \in E} \hat{\rho}(i)$;
22 $\quad \hat{\rho}(e) = index$;
23 $\quad E = E - \{e\}$;
24 **end**

Mathematical Model

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; 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) [14] 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 users u and v can be calculated by

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

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 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 defined a

$$\tilde{I}(x) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{otherwise.} \end{cases}$$

From above definition, the ranking similarity between two rankings is in the interval of [-1,-1] where1 is obtained when the order of user u is the exact reverse of user v, and 1 is obtained when order of user u is equal to the order of user v. Since KRCC compares service pairs, the intersection between two users has to be at least 2 (|Iu^ Iv>=2) for making similarity computation.

2. Related work and Discussion

Cloud computing is becoming popular. A number of works have been carried out on cloud computing [8], [10], including performance analysis, market-oriented cloud computing, management tool, workload balance, dynamic selection, etc. Quality-of-service has been widely employed for presenting the nonfunctional characteristics of the software systems and services. QoS of cloud services can be measured from either the client side (e.g., response time, throughput, etc.) or at the server side (e.g., price, availability, etc.). Based on the service QoS measures, various approaches have been proposed for service selection [3], which enables optimal service to be identified from a set of functionally similar or equivalent candidates, our work provides a comprehensive study of how to provide accurate QoS ranking for cloud services, which is a new and urgently-required research problem. Currently, our CloudRank framework is mainly designed for cloud applications, because: 1) client-side QoS values of different users can be easily obtained in the cloud environment; and 2) there are a lot of redundant services abundantly available in the cloud, QoS ranking of candidate services becomes important when building cloud applications. The CloudRank framework can also be extended to other component-based applications, in case that the components are used by a number of users, and the past usage experiences of different users can be obtained.

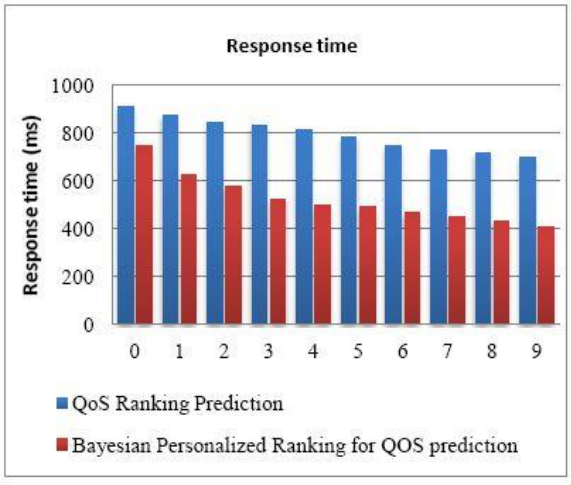
3. Results and Analysis

Java Technology is used for developing this project. The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. QoS of cloud services can be measured from either the client side based on the response time for each service. Response-time refers to the time duration between the user sending out a request to a service and receiving a response. The following table, Table 1 shows the performance of QoS Rank.

Table 1. Performance comparison by response time

Number of Tasks	QoS	BPR
0	910	750
1	875	625
2	845	575
3	830	525
4	817	500
5	785	490

Fig. 2 shows the result of QoS response time for QoS ranking prediction and Bayesian Personalized Ranking method for QoS Prediction, it shows that proposed Bayesian Personalized Ranking method for QoS Prediction methods response quick than the QoS ranking prediction methods for services.



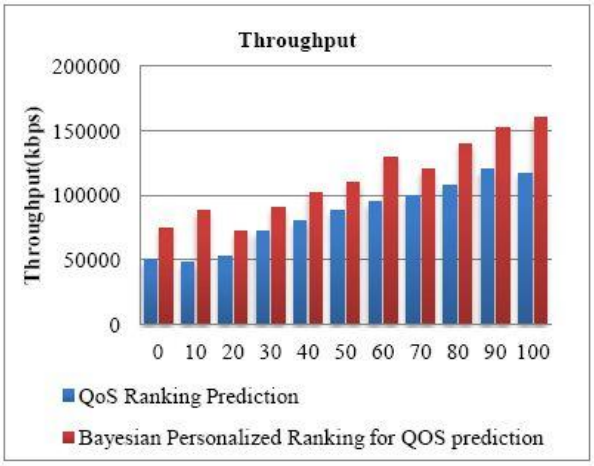
QoS of cloud services can be measured from either the client side based on the Throughput for each service. Throughput represents the data transfer rate over the network. Fig. 2 and

Table 2 shows the result of QoS Throughput for QoS ranking prediction and Bayesian Personalized Ranking method for QoS Prediction, it shows that proposed Bayesian Personalized Ranking method for QoS Prediction methods achieves higher throughput result than QoS ranking prediction methods for services.

Table 2. Performance comparison by throughput

Number of Tasks	QoS	BPR
0	50000	75000
10	48000	88000
20	53000	72000
30	72000	90000
40	80000	102000
50	88000	110000
60	95000	130000
70	100000	120000
80	108000	140000
90	120000	153000
100	117000	160000

It shows that proposed system transfer highest data transfer than existing methods. The throughput values also exhibit a great variation.



QoS of cloud services can be measured from either the server side based on the cost function for each service. Cost function represents the quick time to process QoS services.

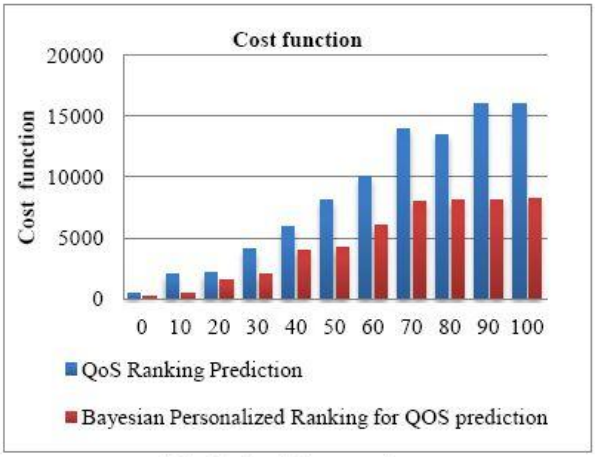


Fig 4 includes Table 3 shows the result of QoS cost function result for QoS ranking prediction and Bayesian Personalized Ranking method for QoS Prediction, it shows that proposed Bayesian Personalized Ranking method for QoS Prediction methods achieves higher cost result than QoS ranking prediction methods for services.

Table 2. Performance comparison by cost

Number of Tasks	QoS	BPR
0	500	200
10	2000	500
20	2100	1500
30	4050	2000
40	5900	4000
50	8075	4200
60	10000	6000
70	14000	8000
80	13500	8100
90	16000	8100
100	16000	8200

It shows that proposed system achieves less time to perform process QoS prediction results than existing methods.

4.4 Impact of Similarity Computation

There are different types of similarity computation methods. Rating similarity computation methods compare the QoS values of the commonly invoked cloud services for the computation, while ranking similarity computation methods employ QoS rankings of services for calculating the similarities. Well-known rating similarity computation methods include VS and PCC, while well-known ranking similarity computation methods include KRCC. To compare the performance of different similarity computation methods, we implement three versions of our CloudRank1 and CloudRank2 algorithms, using the KRCC, PCC, and VS similarity computation methods, respectively. We change the matrix density from 5 to 50 percent with a step value of 5 percent. We set Top-K to 5 in this experiment. CloudRank1 and CloudRank2 algorithms with different similarity computation methods are compared in this experiment. Fig. 5 shows the experimental results, where Figs. 5a and 5b show the NDCG100 results of response time of CloudRank1 (labeled as CR1 in the Figure) and CloudRank2 (labeled as CR2 in the Figure), respectively. Figs. 5c and 5d show the NDCG100 results of throughput of CloudRank1 and CloudRank2, respectively.

4. Conclusion

In this paper, we develop a 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.

5. Future Work

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 Experiments," 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 Recommendation," ACM Trans. Information System, vol. 22, no. 1, pp. 143-177, 2004.
[8] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing," IEEE Trans. Parallel Distributed System, vol. 22, no. 6, pp. 931-945, June 2011.
[9] 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.
[10] H. Khazaei, J. Misic, and V.B. Misic, "Performance Analysis of Cloud Computing Centers Using m/g/m/m+r Queuing Systems," IEEE Trans. Parallel Distributed System, vol. 23, no. 5, pp. 936-943, May 2012.
[11] G. Linden, B. Smith, and J. York, "Amazon.Com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan./Feb. 2003.
[12] N.N. Liu and Q. Yang, "Eigenrank: A Ranking-Oriented Approach to Collaborative Filtering," Proc. 31st Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '08), pp. 83-90, 2008.
[13] H. Ma, I. King, and M.R. Lyu, "Effective Missing Data Prediction for Collaborative Filtering," Proc. 30th Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '07), pp. 39-46, 2007.
[14] J. Marden, Analyzing and Modeling Ranking Data. Chapman & Hall, 1995.
[15] 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, 2009
[16] 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.
[17] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. Zhou, and Z. Wu, "Predicting QoS for Service Selection by Neighborhood-Based Collaborative Filtering," IEEE Trans. System, Man, and Cybernetics, Part A, to appear.
[18] C. Yang, B. Wei, J. Wu, Y. Zhang, and L. Zhang, "Cares: A Ranking-Oriented Cadal Recommender System," Proc. Ninth ACM/IEEE-CS Joint Conf. Digital Libraries (JCDL '09), pp. 203-212, 2009.
[19] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints," ACM Trans. Web, vol. 1, no. 1, pp. 1-26, 2007.
[20] Z. Zheng, H. Ma, M.R. Lyu, and I. King, "WSRec: A Collaborative Filtering Based Web Service Recommender System," Proc. Seventh Int'l Conf. Web Services (ICWS '09), pp. 437-444, 2009.

Author Profile

Sangeeta Alagi studying M.E Computer Engineering, received b.tech degree in Computer Science And Engineering from BKEC college of engineering, Basavakalyan, Dist Bidar, Karnataka in 2011.