A Ranking Oriented Approach to the Data in Cloud Service Provider and Cloud Services

B. Karthika[#1], V. M. Suresh[#2]
#1 PG Student, E.G.S.Pillay Engineering College, Nagapattinam, Tamilnadu, India.
#2 Department of Computers, E.G.S.Pillay Engineering College, Nagapattinam, Tamilnadu, India.

Abstract—**Cloud computing is becoming popular. Building high-quality cloud applications is a grave research problem. QoS ranking provide valuable information for making ideal 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 invocations, this paper proposes a QoS ranking prediction framework for cloud services by taking advantage of the past service usage experiences of other consumers. Our proposed framework requires no additional invocations of cloud services when making QoS ranking prediction. Two personalized QoS ranking prediction approaches are proposed to predict the QoS rankings directly. The proposed results show that our approaches outperform other competing approaches.**

Keywords—Quality-of-service, cloud service, ranking prediction, personalization, comparison methods

I. INTRODUCTION

The CLOUD computing is Internet-based computing, where by shared configurable resources (e.g., infrastructure, platform, and software) are provided to computers and other devices as services [1]. 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 cloud environment (named cloud applications in this paper) 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. Fig. 1 shows an example of cloud applications. As shown in the figure, Cloud application 1 is a tourism Website deployed in the cloud, providing various types of tourism services to customers. The business process of this cloud application is composed by a number of software components, where each component fulfills a specified functionality.

To outsource part of business to other companies, some of these components invoke other cloud services (e.g., airplane ticket services, car rental services, and hotel booking services in Fig. 1). These cloud services (can be implemented as web services) are provided and deployed in the cloud by other

other cloud applications (e.g., cloud application 2 and cloud application 3 in Fig. 1).

The service users refer to cloud applications that use invokes the cloud services. In the context of a service invocation, the user-side (or client side) refers to the cloud applications and server side refers to the cloud services. The most straightforward approach of personalized cloud QoS ranking is to evaluate all the candidate services at the user-side and rank the services based on the observed QoS values.

II. SYSTEM ARCHITECTURE

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.

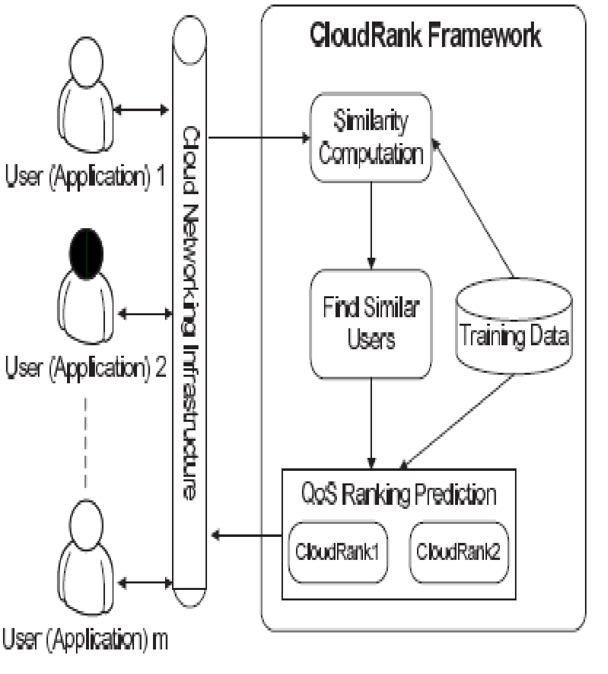

Fig. 1 System Architecture

The ranking prediction of client-side QoS properties, which likely have different values for different users (or user applications) of the same cloud service. The system architecture of our Cloud Rank framework, which provides personalized QoS ranking prediction for cloud services.

The target users of the CloudRank framework are the cloud applications which need personalized cloud service ranking for making optimal service selection Within the CloudRank framework, there are several modules. First, based on the user-provided QoS values, similarities between the active user and training users can be calculated.

III. QOS RANKING PREDICTION

This section presents our CloudRankQoS ranking prediction framework for cloud services.

A) Similarity 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.

B) 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. 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.

C) QoS Ranking Prediction

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. 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.
To address this problem, we propose two ranking-oriented approaches, named as CloudRank1 and Clou-dRank2, in the following. Our ranking-oriented approaches predict the QoS ranking directly without predicting the corresponding QoS values..

1) CloudRank1

Quality of service i is better than service j and is thus more preferable for the active user and vice versa
Goal is to produce a ranking that maximizes the above objective value through the possible rankings and select the optimal ranking that maximizes function. One possible solution is to search the value function defined. However, there are n! possible rankings for n services, and the optimal ranking search problem is NP-Complete . To enhance the calculation efficiently, propose a greedy-based algorithm in Algorithm 1 (named as CloudRank1).

Algorithm 1: CloudRank1

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

```
F = E;
while F ≠ ∅ do
    t = arg max_{i∈F} q_i;
    ρ_e(t) = |E| − |F| + 1;
    F = F − {t};
end
foreach i ∈ I do
    π(i) = Σ_{j∈I} Ψ(i, j);
end
n = |I|;
while I ≠ ∅ do
    t = arg max_{i∈I} π(i);
    ρ̂(t) = n − |I| + 1;
    I = I − {t};
    foreach i ∈ I do
        π(i) = π(i) − Ψ(i, t)
    end
end
while E ≠ ∅ do
    e = arg min_{i∈E} ρ_e i;
    index = min_{i∈E} ρ̂(i);
    ρ̂(e) = index;
    E = E − {e};
end
```

2) 3.3.2 CloudRank2

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.

D) Computational Complexity Analysis

Assuming there are n cloud services and m users, this section analyzes the worst case computational complexity of the CloudRank1 and CloudRank2 algorithms, respectively. Based on the preference values, the CloudRank1 algorithm and CloudRank2 algorithm make QoS ranking prediction. As shown in Algorithms 1 and 2, the computational complexities of CloudRank1 and CloudRank2 are both equal.

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 $\hat{\rho}$

$F = E$;
while $F \neq \emptyset$ **do**
    $t = \arg\max_{i \in F} q_i$;
    $\rho_e(t) = |E| - |F| + 1$;
    $F = F - \{t\}$;
end
foreach $i \in I$ **do**
    $\pi(i) = \sum_{j \in I} C(i, j) \times \Psi(i, j)$;
end
$n = |I|$;
while $I \neq \emptyset$ **do**
    $t = \arg\max_{i \in I} \pi(i)$;
    $\hat{\rho}(t) = n - |I| + 1$;
    $I = I - \{t\}$;
    foreach $i \in I$ **do**
        $\pi(i) = \pi(i) - C(i, j) \times \Psi(i, t)$
    end
end
while $E \neq \emptyset$ **do**
    $e = \arg\min_{i \in E} \rho_e i$;
    $index = \min_{i \in E} \hat{\rho}(i)$;
    $\hat{\rho}(e) = index$;
    $E = E - \{e\}$;
end

IV. EXPERIMENTS

A) Data Set Description

To evaluate the QoS ranking prediction accuracy, we conduct a large-scale real-world web service evaluation to collect QoS values on real-world web services. We have collected addresses of 500 real-world web services from the Internet. To collect QoS values of these web services, first, we generated web service invocation codes by Axis2, Java-based open-source package for web services.

B) Evaluation Metric

Rating-oriented approaches must predict QoS values as accurate as possible. Therefore, differences between the predicted values and the true values are usually employed to evaluate the prediction accuracy. Mean Absolute Error and Root-Mean Square Error (RMSE) metrics are two widely adopted evaluation metrics for rating-oriented approaches.

C) Performance Comparison

User-based collaborative filtering method using Vector Similarity (UVS). This method employs vector similarity for calculating the user similarities and engages the similar users for the QoS value prediction.
Item-based collaborative filtering method using Vector Similarity (IVS). This method employs vector similarity for computing the item (cloud services) similarities when making QoS value prediction.

User-based and Item-based collaborative filtering using Vector Similarity (UIVS). This method combines the user-based and item-based collaborative filtering approaches and employs the vector similarity for the similarity computation for users and items.

User-based collaborative filtering method using Pearson Correlation Coefficient (UPCC). This is a classical method. It employs PCC for calculating the user similarities and engages the similar users for the QoS value prediction [4].

Item-based collaborative filtering method using Pearson Correlation Coefficient (IPCC). This method is widely used in industry company like Amazon. It employs PCC for the similarity computation and employs similar items (cloud services) for the QoS value prediction.

User-based and item-based Collaborative filtering using Pearson Correlation Coefficient (UIPCC). This method combines the user-based and item-based collaborative filtering approaches and employs PCC for the similarity computation.

D) Impact of Similarity Computation

There are different types of similarity computation methods. Rating similarity computation methods compare the QoSvalues of the commonly invoked cloud services for the computation, while ranking similarity computation methods employ QoS rankings of services for calculating the similarities.

V. Related Work And Discussion

A number of works have been carried out on cloud computing [2], [3], including performance analysis, market-oriented cloud computing, management tool, workload balance, dynamic selection, etc. 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.

VI. Conclusion And Future Work

Propose 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.

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.

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, 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.