

A Collaborative Quality Ranking Framework for Cloud Components

Zibin Zheng and Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
{zbxheng,lyu}@cse.cuhk.edu.hk

Abstract

The rising popularity of cloud computing makes building high quality cloud applications an urgently-required research problem. Component quality ranking provides valuable information for the designers of cloud applications. This paper proposes a collaborative component ranking framework for cloud applications which requires no additional invocations on the cloud components. The extensive experimental results show the effectiveness of our approach.

1. Introduction

Cloud computing is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like a public utility. Cloud applications, which include a number of distributed cloud components, are usually large-scale and very complex. With the rising popularity of cloud computing, building high-quality cloud applications becomes an urgently-required research problem. In the cloud environment, cloud components are usually invoked remotely by communication links. Influenced by the unpredictable communication links, different cloud applications will receive different levels of quality of the same reusable cloud component. Personalized component quality ranking is thus a crucial task for building high-quality cloud applications.

The major challenge for making quality ranking of cloud components is that the component quality ranking of a user (i.e., designer of a cloud application) cannot be transferred directly to another one influenced by the locations of cloud applications. The most straightforward approach to achieve personalized cloud component ranking is to evaluate all the components at the user-side and rank the components based on the observed QoS performance. However, this approach is impractical in reality, since there is a huge number of components in the cloud. Conducting evaluation on all these components is time-consuming and resource-consuming.

To attack this critical challenge, we propose a collabo-

orative quality ranking framework in Section 2 and conduct extensive experiments in Section 3 to show the effectiveness of our approach.

2. Collaborative Quality Ranking Framework

Given two rankings on the same set of components, the Kendall Rank Correlation Coefficient (KRCC) [3] evaluates the degree of similarity by considering the number of inversions of component pairs which would be needed to transform one rank order into the other. The KRCC value of user a and user u can be calculated by:

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

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

$$\tilde{I}(x) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

A user's preference on a pair of components can be modeled in the form of $\Psi : I \times I \rightarrow \mathbb{R}$, where $\Psi(i, j) > 0$ means that quality of component i is higher than component j and vice versa [2]. The value of the preference function $\Psi(i, j)$ indicates the strength of preference. Given the user-observed QoS values on two cloud components, the preference value can be calculated by $\Psi(i, j) = q_i - q_j$. To obtain preference information regarding pairs of components that have not both been invoked by the current user, the QoS values of similar users $S(u)$ is employed:

$$\Psi(i, j) = \sum_{v \in N(u)^{ij}} \frac{Sim(u, v)}{\sum_{v \in N(u)^{ij}} Sim(u, v)} (q_{v,i} - q_{v,j}), \quad (3)$$

where v is a similar user and $N(u)^{ij}$ is a subset of similar users who have QoS values of both component i and j .

Algorithm 1: CloudRank Algorithm

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

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

Given a preference function Ψ , we want to choose a quality ranking of components in I that agrees with the pairwise preferences as much as possible. Let ρ be a ranking of components in I such that $\rho(i) > \rho(j)$ if and only if i is ranked higher than j in the ranking ρ . We can define a value function $V^\Psi(\rho)$ as follow that measures the consistency of the ranking ρ with the preference function:

$$V^\Psi(\rho) = \sum_{i, j: \rho(i) > \rho(j)} \Psi(i, j). \quad (4)$$

To produce a ranking ρ^* that maximizes the objective function in Eq. (4), we propose Algorithm 1 for finding an approximately optimal ranking. Algorithm 1 first rank the cloud components in E which have been employed by the user (line 1-6). Then, for each component in the full component set I , the preference sum is calculated by $\pi(i) = \sum_{j \in I} \Psi(i, j)$ (line 7-9). After that, the components in I are ranked from the highest position to the lowest position by picking the component t that has the maximum $\pi(t)$ value (line 10-18). Finally, the initial component ranking $\hat{\rho}(i)$ is updated by correcting the rankings of the employed components in E (line 19-24). By these steps, our approach finds an approximately optimal ranking and makes sure that the employed components in E are correctly ranked.

3. Experiments

We evaluate the ranking performance using our WS-DREAM¹ QoS dataset [4], which includes QoS perfor-

¹<http://www.wsdream.net>

Table 1. NDCG of Response Time

Methods	NDCG3	NDCG10	NDCG100
UVS:	0.9491	0.9104	0.9514
UPCC:	0.9347	0.8968	0.9414
IVS:	0.9710	0.9308	0.9637
IPCC:	0.9737	0.9359	0.9656
Greedy	0.9789	0.9523	0.9755
CloudRank	0.9792	0.9532	0.9763

mance of about 1.5 million Web service invocations from 150 distributed users on 100 service components. In our experiments, the *response-time* QoS property is employed to rank the components. To evaluate the ranking performance, we employ the Normalized Discounted Cumulative Gain (NDCG) [1] metric, where larger value indicates better performance. Table 1 show the NDCG values of different ranking approaches employing 10% density user-item matrix. In the first row of the table, NDCG3 indicates that the ranking accuracy of the top 3 components is investigated. The first four methods in the table are well-known rating-oriented collaborative filtering methods, while the last two methods are ranking-oriented methods. For each column in the Tables, we have highlighted the best performer among all methods. Among all the ranking methods, our approach (named as CloudRank) obtains better ranking performance under all the experimental settings consistently.

4. Conclusion and Future Work

In this paper, we propose a collaborative quality ranking framework for cloud components. The experimental results show that our approach outperforms existing rating-based collaborative filtering approaches and the traditional greedy method. We would like to investigate various techniques for improving the ranking accuracy and study more QoS properties in our future work.

Acknowledgement

The work described in this paper was fully supported by a grant (Project No. CUHK4154/09E) from the Research Grants Council of Hong Kong, China.

References

- [1] K. J. arvelin and J. Kekalainen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.
- [2] N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *SIGIR*, pages 83–90, 2008.
- [3] J. Marden. *Analyzing and Modeling Ranking Data*. Chapman & Hall, New York, 1995.
- [4] Z. Zheng and M. R. Lyu. Collaborative reliability prediction for service-oriented systems. In *ICSE*, 2010.