GENERATING RANKING FOR CLOUD SERVICES USING CLOUD RANK FRAMEWORK

1 CHITHALURI RANJITH KUMAR, 2 G.MANOJ KUMAR

1M.Tech, Department of CSE, Malla reddy College Of Engineering and Technology , Maisammaguda village, Medchal Mandal, Rangareddy District, A.P, India.

2 Assistant Professor, Department of CSE, Malla reddy College Of Engineering and Technology, Maisammaguda village, Medchal Mandal, Rangareddy District, A.P, India.

Abstract:
QoS (Quality-of-Service) and Building high quality cloud applications becomes an urgently required is an important topic in cloud computing. It is very difficult to choose which service is best one to use and what the criteria for their selection are. Quality of service parameters can be used to select the best possible service provider. The providers can be ranked according to the Quality of services. Two QoS ranking prediction algorithms are used to predict the QoS rankings. In the proposed work the feedback method is used to predict the ranking based on the likes given by the end users. In order to avoid this scenario we propose an approach which directly predicts the QoS ranking by taking past QoS data of other consumers without checking for the subsequent QoS values of other web services.

I. INTRODUCTION

Cloud computing is an incipient paradigm for distributing ondemand resources (e.g., infrastructure, platform, software, etc.) for customers akin to other utilities (e.g., dihydrogen monoxide, electricity and gas). The current Cloud computing architecture enables three layers of accommodations. The cloud abstracts the desideratum for you to be in the same physical location as the hardware that stores your data. There are number of functionally equiollent accommodations in the cloud Due to unreliable internet connections different cloud applications may receive different calibers of quality for same cloud accommodations so that we require to cull the optimal accommodations.

The introduction of the paper should explain the nature of the problem, previous work, purpose, and the contribution of the paper. Recent days the cloud computing technology is popular because it is an attracting technology in the pasture of computer science. Cloud computing is internet based computing that generally referred the shared configurable resources (e.g., infrastructure, platform, and software) is provided with computers and other devices as services. Cloud computing entrusts services with a consumer’s data, software and computation ended a network. The customer of the cloud can get the services during the network. In additional words, users are using or buying computing services from others. Cloud can provide
Anything-as-a-Service (AaaS). In Cloud technology, the QoS based service selection is an essential research topic. While lots of services suggest similar functionality, QoS values show a critical role for separating the optimal service for that particular task [5]. Because many number of cloud services are available. Since the customer points of view, it is not easy to choose the best service and what mechanism used to select their services [10]. QoS models are associated with End-Users and providers.

A service application is decomposed into a set of software components/services running on VEEs on the same or different VEEs within a site or across different sites. However, Reservoir architecture does not allow a component/service to run on its duplicates on different VEEs; Moreover, computing resources are abstracted as hosting service which might not be necessarily true for all clouds. In [6], a software platform for .NET based cloud computing named Aneka was introduced. Aneka is a customizable and extensible service oriented runtime environment that enables developers to build .NET applications with the supports of APIs and multiple programming models. Aneka is a service-oriented, pure PaaS cloud solution. In [7], Rajkumar and his colleagues explained a market-oriented cloud architecture in detail used by Aneka, which regulates the supply and demand of cloud resources to achieve market equilibrium, adds economic incentives for both cloud consumers and providers, and promotes QoS-based resource allocation mechanisms that differentiates service request based on their utility. The key component of this architecture is SLA (Service Level Agreement) Resource Allocator which is consisted of Service Request Examiner and Access Control, VM (Virtual Machines) monitor, Service Request Monitor, and Request Dispatcher. Based on the feedback from VM and Service Request monitors, the dispatcher routes the requests from users/brokers to the cloud resources that can fulfill their QoS requirements. In [8], Huang and her colleagues from IBM described a service oriented cloud computing platform that enables web-delivery of application-based services with a set of common business and operational services. The platform
supports multi-tenancy feature by utilizing single application instance model. The isolation among tenants is taken care by the underline design. Other services include subscription management, federated ID management, application firewall, etc.

2.1 QoS Ranking Prediction on Cloud Services

Since this work explores the issue of building high quality cloud applications. Quality-of-Service (QoS) is usually employed for describing the non-functional characteristics of Web services and employed as an important differentiating point of different Web services. Users in different geographic locations collaborate with each other to evaluate the target Web services and share their observed Web service QoS information. In optimal Service Selection [2] proposed Exact and approximated algorithms for optimal service selection based on a given set of service requests (such as the activities occurring in a workflow), a set of service users (the available services), the result of the matchmaking process (that associates each request to the set of users that can satisfy it), and a numeric preference measure. It identified the Service Selection Problem (SSP). We show that the high computational complexity of the service selection problem is caused by the one-time costs associated with service users (e.g., Initialization and registration costs). In the absence of one-time costs, the optimal selection problem can be solved in polynomial time by applying a greedy approach. The heuristic algorithm seems to be faster, but it has no guarantees on the quality of the solution. Collaborative filtering algorithms [3] proposed Memory-based algorithm and Model-based algorithm that predicts the utility of items to a particular user (the active user) based on a database of user votes from a sample or population of other users (the user database). We use two basic classes of evaluation metrics. The first characterizes accuracy over a set of individual predictions in terms of average absolute deviation. The second estimates the utility of a ranked list of suggested items. Bayesian networks typically have smaller memory requirements and allow for faster predictions than a memory-based technique such as correlation but Bayesian methods examined here require a learning phase that can take up to several hours and results are reflected in the recommendations. Item-Based Top-N Recommendation Algorithms [4] determines the similarities between the various items from the set of items to be recommended.

Fig.2 System architecture of cloud Rank

The key steps in this class of algorithms are (i) the method used to compute the similarity between the items, and (ii) the method used to combine these similarities in order to compute the similarity between a basket of items and a candidate recommender item. The goal of top-N recommendation algorithm was to classify the items purchased by an individual user into two classes: like and dislike. This algorithm is faster than the traditional user-neighborhood based recommender
systems and it provides recommendations with comparable or better quality. The proposed algorithms are independent of the size of the user–item matrix. Recommendation Algorithm [6] determines a set of customers whose purchased and rated items overlap the user’s purchased and rated items. The algorithm aggregates items from these similar customers, eliminates items the user has already purchased or rated, and recommends the remaining items to the user. It generates high quality recommendations and the algorithm must respond immediately to new information. It is used to personalize the online store for each customer but it needs to apply recommendation algorithms for targeted marketing, both online and offline. Collaborative filtering approach [7] addresses the item ranking problem directly by modelling user preferences derived from the ratings. It performs ranking items based on the preferences of similar users. CloudRank approach [11] proposed greedy algorithm and it rank the component instead of service but this algorithm. It is used to rank a set of items, which treats the explicitly rated items and the unrated items equally. It does not guarantee that the explicitly rated items will be ranked correctly.

QoS-Aware Web Service by Collaborative Filtering [12] proposed Hybrid collaborative filtering method that To improve performance of Recommender System. It includes a user-contribution mechanism for Web service QoS information collection and an effective and novel hybrid collaborative filtering algorithm for Web service QoS value prediction. It is used to collect systematic QoS information and it provides better feasibility of WSRec (Web service recommender system) but it needs to monitor more realworld Web services and it needs to investigate more QoS properties of Web services.

3. 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. The KRCC value of user’s u and v can be calculated by

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

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,

\[ S_{\text{sim}}(u,v) = 1 - \frac{4 \times \sum_{i,j \in I_u \cap I_v} I(q_{u,i} - q_{v,i})(q_{u,j} - q_{v,j})}{|I_u \cap I_v| \times (|I_u \cap I_v| - 1)}, \]

Where \(I_u I_v\) 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 \(I(x)\) is an indicator function.
4. CONCLUSION

In this paper, we have compared different ranking frameworks and different architectures for culling an accommodation according to the rank of the accommodation. The main goal of the comparison is to compare the frameworks and giving information about the technique the framework is predicated on and the circumscriptions of the different frameworks, so that the utilizer can cull cloud accommodation according to the requisites. The framework used are very subsidiary in guiding customers about culling the optimal accommodation which are predicated on the quality of accommodation parameters.

5. REFERENCES


