JAFRC

International Journal of Advance Foundation and Research in Computer (IJAFRC)
Volume 1, Issue 11, November 2014. ISSN 2348 - 4853

Quality of Service Ranking Prediction for Cloud.

Mayura Shelke
JSPM's Bhivarbai Sawant Institute of Technology and Research, Pune
mayura.shelke@gmail.com

A B S T R A C T

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. Building high quality cloud applications is a critical research problem. Quality of service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance. QoS is used to avoid the time-consuming and expensive real-world service invocations. QoS rankings provide valuable information for making optimal cloud service selection from a set of functionally equivalent service candidates. CloudRank framework, which provides personalized QoS ranking prediction for cloud services

Index Terms: QOS, Cloud Computing IAAS, PAAS, SAAS, Cloud Ranking.

I. INTRODUCTION

Cloud computing is Internet-based computing.[1] The shared resources, software and other information are provided to computers and other devices on demand for e.g., Amazon, Google, Microsoft, IBM, etc. Applications deployed in the cloud environment are typically large-scale and complex. The following figure1.1 shows the cloud architecture.[2]

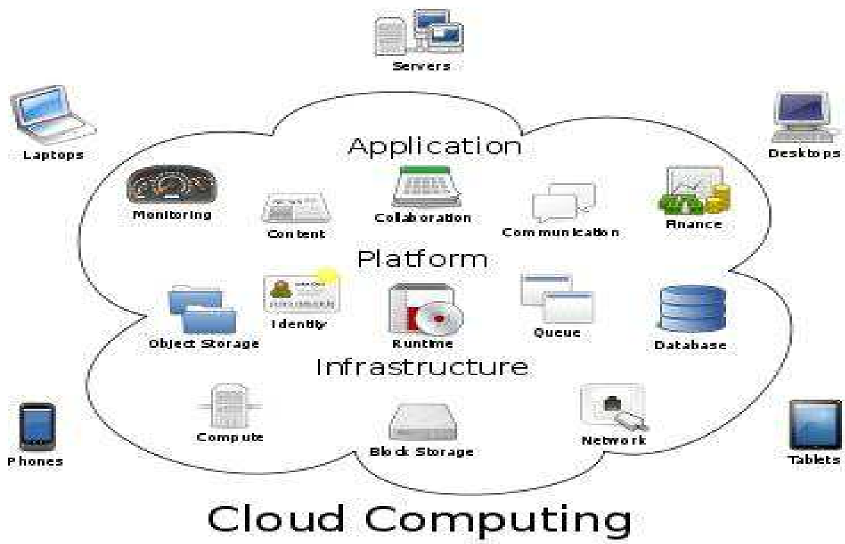

Figure 1: Cloud Architecture

Cloud Computing Service Models

A. Software as a Service (SaaS)

In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud and multiple end users are serviced .On the customers side, there is no need for upfront investment in servers or software licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted and maintained. Today SaaS is offered by companies such as Goggle, Sales-force, Microsoft, Zoho, etc.

B. Platform as a Service (PaaS)

Here, a layer of software or development environment is encapsulated and offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider's infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySql and PHP), restricted J2EE, Ruby etc. Goggles App Engine, Force.com, etc. are some of the popular PaaS examples.

C. Infrastructure as a Service (IaaS)

IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc.are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, Go-grid, 3Tera, etc. The Following figure shows the cloud computing architecture with three service models.

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 Figure 2). These cloud services (can be implemented as Web services) are provided and deployed in the cloud by other companies. These cloud services can also be employed by other cloud applications (e.g., Cloud application 2 and Cloud application 3 in Figure 1). Since there are a number of functionally equivalent services in the cloud, optimal service selection becomes important. In this paper, service users refer to cloud applications that use/invoke 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.

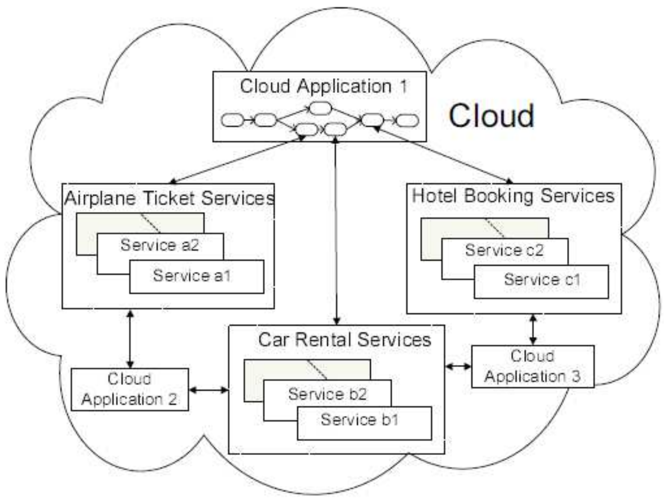

Figure 2. Cloud example

II. QoS CONCEPT

Quality of service (QoS) is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance. QoS criteria are numerous and is highly dependent of the application throughput, Delay, jitter, Loss rate.

1. Loss: probability that a flow's data is lost.
2. Delay: time it takes a packet's flow to get from source to destination.
3. Delay jitter: maximum difference between the delays experienced by two packets of the flow.
4. Bandwidth: maximum rate at which the source can send data.

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 (e.g., Cloud application 1) cannot be transferred directly to another user (e.g., Cloud application 2), since the locations of the cloud applications are quite different. Personalized cloud service QoS ranking is thus required for different cloud applications. When the number of candidate services is large, it is difficult for the cloud application designers to evaluate all the cloud services efficiently.

A. Advantages

1. QoS has been widely employed for presenting the non-functional characteristics of the software systems and services
2. QoS ranking prediction framework for cloud services, which requires no additional service invocations when making QoS ranking.
3. QoS ranking prediction framework for cloud services by taking advantage of the past service usage experiences of other consumers

Literature Review

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. Quality-of-Service (QoS) is usually engaged for describing the non-functional characteristics of Web services. QoS management of Web services refers to the activities in QoS specification, evaluation, prediction, aggregation, and control of resources to meet end-to-end user and application requirements.

A. Ranking-oriented collaborative Filtering

Traditional rating-oriented collaborative filtering focuses on predicting a user's potential ratings on unrated items by utilizing the known ratings associated with similar users or similar items. In this section, we present a ranking-oriented collaborative filtering approach that aims at producing an item ranking for the target user. We first describe a user similarity measure that is based on two users' preferences over the items and then present two methods for ranking items based on the preferences of the set of neighbors of the target user.

B. Kendall Rank Correlation Coefficient

In the ranking-oriented approach, the similarity between users is determined by their preferences over the items, which is reflected by their ranking of the items. Suppose we have a set of three items, to which two users have assigned ratings of {2, 3, 4} and {3, 4, 5} respectively. The rating values on the same items by the two users are clearly different, nevertheless their preferences are very close as the items are ordered in the way based the two user's ratings. The Kendall rank correlation coefficient is a similarity measure between two rankings of the same set of objects:

$$s_{u,v} = 1 - \frac{4 \times \sum_{i,j \in I_u \cap I_v} I^{-}\left((r_{u,i} - r_{u,j})(r_{v,i} - r_{v,j})\right)}{|I_u \cap I_v| \cdot (|I_u \cap I_v| - 1)}$$

where $I^{-}(x)$ is an indicator function defined as:

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

Where a pair of items i and j is disconcordant if i is ranked higher than j in one ranking but lower in the other.

C. Preference Functions

To produce a ranking of the items for a user rather than predicting the rating values, focus on modeling a user's preference function of the form: I×I→R, where Ψ(i, j) > 0 means that item i is more preferable to j for user u and vice versa. The magnitude of this preference function |Ψ (i, j)| indicates the strength of preference and a value of zero means that there is no preference between the two items. We assume that Ψ (i, i) = 0 for all i ε I and that Ψ is anti-symmetric, i.e. Ψ(i, j) = − Ψ(j, i) for all i, j ε I. Note that, however, we do not require Ψ to be transitive, i.e. Ψ(i, j) > 0 ^ Ψ (j,k)> 0 does not imply Ψ (i, k) > 0.

D. Greedy Order Algorithm

Given a preference function Ψ which assigns a score to every pair of Web services i, j ∈ I, want to choose a quality ranking of Web services in I that agrees with the pairwise preferences as much as possible. Let ρ be a ranking of Web services in I such that ρ (i) > ρ (j) if and only if i is ranked higher than j in the ranking ρ. We can define a value function V Ψ (ρ) as follows that measures the consistency of the ranking ρ with the preference function:

$$V\Psi(\rho) = \sum_{i;j:i>j} \Psi(i,j)$$

Goal is to produce a ranking ρ that maximizes the above objective value function. One possible approach to solve the Web service ranking problem is to search through the possible rankings and select the optimal ranking ρ∗ that maximizes the value function defined in above Eq. However, there are n! possible rankings for n Web services. It is impossible to search all the rankings when the value of n is large. To enhance the calculation efficiently, they propose a greedy order algorithm in Algorithm 1 (named as CloudRank) for finding an approximately optimal ranking.

III. ARCHITECTURE

CloudRank is first QoS ranking prediction framework for cloud services. It takes advantage of the past usage experiences of other users for making personalized ranking prediction for the current user. Figure 3 shows the system architecture of our CloudRank framework.

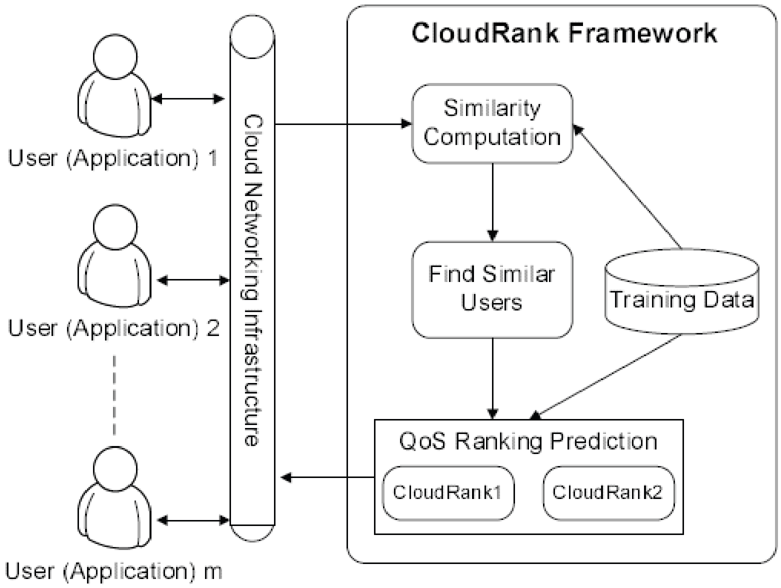

Figure 3. System Architecture

IV. CONCLUSION

Cloud computing is becoming popular. Building high-quality cloud applications is a critical research problem. 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. In this paper, we studied 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, ranking approach identifies and aggregates the preferences between pairs of services to produce a ranking of services.This approach is more efficient for cloud services. In future, we will try to make it more powerful for cloud services.

VII. REFERENCES

[1] Yanbin Lu and Gene Tsudik,"Enhancing Data Privacy in the Cloud",University of California, Irvine.

[2] Pooja Shah, Gopal Pandey, "Keyword Searching Techniques for Encrypted Cloud Data", Department of Information Technology, Shantilal Shah Engineering College, Bhavnagar.

[3] Zibin Zheng, , Xinmaio Wu, Yilei Zhang, Michael R. Lyu, Fellow, and Jianmin Wang "QoS Ranking Prediction for Cloud Services" , University of Posts and Telecommunications, Beijing Wu, X., IEEE Computer Society 03 October 2012.

[4] Z. Zheng, Y. Zhang, and M. R. Lyu, "CloudRank: A QoS-driven component ranking framework for cloud computing," in Proc.Int'l Symposium Reliable Distributed Systems (SRDS'10), 2010, pp.184–193

[5] Yanbin Lu and Gene Tsudik,"Enhancing Data Privacy in the Cloud",University of California, Irvine

[6] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for Web services selection with end-to-end QoS constraints," ACM Trans. the Web, vol. 1, no. 1, pp. 1–26, 2007

