Ranking Prediction of Cloud Services based on BPR

Aswathi Vandana P
PG Scholar
Department of CSE (PG)
Sri Ramakrishna Engineering College, Tamilnadu, India

Bhaggiaraj S
Assistant Professor (Sr. Gr.)
Department of IT
Sri Ramakrishna Engineering College Tamilnadu, India

ABSTRACT

Cloud is an advancing technology where the the concept of service oriented architecture, distributed, autonomic, and utility computing is being utilized by various service providers. The current changing world is highly challengeable to build high-quality cloud application. Since QoS ranking that provide beneficial information for optimal cloud service selection is time consuming and expensive, this paper focuses on QoS ranking prediction framework for selecting optimal cloud services based on Bayesian Personalized Ranking. Our proposed framework is able to perform well with no additional invocations of cloud services while predicting appropriate services. QoS Ranking Prediction can be done by two approaches. The simulation results show that the approaches are outperforming other competing approaches.

Keywords

Cloud computing, Ranking prediction, Bayesian Personalized Ranking

1. INTRODUCTION

Cloud computing [NIST] is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. All the resources in the cloud are provisioned as services. The popularity Cloud computing has rapidly increased nowadays only because for its flexibility, scalability and reduced cost. Thus, many leading organizations use cloud environment for deploying their applications. Cloud applications include lots of complex components and those components communicate over application programming interfaces. Thus building of high-quality cloud applications is becoming essential.

As said, each of the components performs a specified functionality and there are lots of equally functional services available for a particular cloud application. It's the cloud user's responsibility to select an appropriate service for their cloud application. Thus the complexity of cloud user is increasing according to the number of Cloud providers. But unfortunately, the quality of the cloud applications is still far from perfect in reality.

At present, the demand for high quality cloud applications is becoming extremely strong. Quality-of-service (QoS) describes non functional performance of cloud services. Also QoS assist in decision making. The software components in traditional component-based system are invoked locally, where in cloud applications, cloud services are invoked remotely through internet. Due to unpredictable internet connections the client-side performance may obtain different level of quality for same cloud service. Since the locations of cloud applications are relatively different it is important to proceed for personalized cloud service ranking.

QoS ranking [11] is used to evaluate all the client-side candidate services and ranking the services based on the observed QoS values is an impractical approach since the cloud services may be charged or sometimes execution of such invocations may be time consuming and resource consuming and also it may produce irreversible effects in reality. Also it is difficult for the designer to evaluate all the cloud services efficiently whenever the number of services is more and large.

Thus to predict the QoS rankings for a set of cloud services without any additional invocations from the intended users, we propose a personalized ranking prediction framework, named BPRRank based on QoS rankings Bayesian Prediction Ranking [9]. Our proposed approach has an advantage that ranking prediction of current user can be made by past usage experiences.

The remainder section of this paper is organized as follows: Section 2 is the related work, Section 3 is the proposed architecture, Section 4 describes BPR, Section 5 describes QoSRank framework Section 6 provides the results and analysis and Section 7 concludes the paper.

2. RELATED WORK

As the popularity of Cloud computing as is increasing an attractive alternative to classic information processing systems has increased the importance of reliable and fault tolerant processing.

Andrzej Goscinski, Michael Brock, 2010 [1] proposed the application of the Resources Via Web Services framework (RVWS) to offer higher level abstraction of clouds in the form of a new technology that makes possible the provision of service publication, discovery and selection based on dynamic attributes.

Jose Luis Lucas-Simarro et al. , 2012 [3], presents a modular broker architecture that can work with different scheduling strategies for optimal deployment of virtual services across multiple clouds, based on different optimization criteria, user constraints, and environmental conditions.

Linlin Wu, Saurabh Kumar Garg, Rajkumar Buyya, 2011 [4] proposes innovative admission control and scheduling algorithms for SaaS providers to effectively utilize public cloud resources to maximize profit by minimizing cost and improving customer satisfaction level.

Michael Armbrust et al, 2010 [5] provides simple figures to quantify comparisons between of cloud and conventional computing, and identifying the top technical and non-technical obstacles and opportunities of cloud computing.

Pawel Czarnul, 2012[6] proposes a technique for filtering measured data, in particular to avoid vendor lock-in issues. Also provides a design and results from an engine for simulation of various ranking algorithms in response to streams of prices from various providers.

Qinyi Wu, Arun Iyengar et.al, 2009 [7] proposes ServiceRank framework which considers quality of service aspects as well as social perspectives of services for qualitatively ranking the services.

Saurabh Kumar Garg, Steve Versteeg, Rajkumar Buyya, 2012 [8] proposes, The Service Measurement Index Cloud framework- SMICloud has been proposed based on Service Measurement Index (SMI) helps Cloud customers to find the most suitable Cloud provider and services based on Quality of Service requirements and ranking and therefore can initiate SLAs.

Yilei Zhang, Zibin Zheng and Michael R. Lyu, 2011 [10] proposes BFTCloud is a Byzantine Fault Tolerance framework for building robust systems in voluntary-resource cloud environments. BFTCloud employs replication techniques for overwhelming failures. BFTCloud can also be integrated into cloud nodes as a middleware.

3. PROPOSED ARCHITECTURE

QoS properties always provide good indications of the cloud service capacitiess. Quality of service can be measured at both the server side and client side, where client side QoS properties provide more accurate measurements of the past usage experience which includes response time, throughput, failure rate, etc. This paper mainly focuses on ranking prediction of QoS properties on the client side, that may differ for the user application for same service. Fig. 1 shows the system architecture, which provides personalized cloud ranking based on QoS and BPR. QoSRank is made for the cloud applications which is entirely used for optimal service selection. The user here is called as active user, where he/she 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. The results may be more accurate if the results are achieved by providing QoS values and personalization on cloud services, since the characteristic of the active user can be obtained from the data provided.

Personalization is supportive for both providers and for customers. Here we focus on item recommendation. Item recommendation is used to create a user-specific ranking for a set of items. The user's past interaction with the system helps them in decision making.

Continuing with the similarity computation using QoS and BPR, there are several modules in QoSRank framework. First, based on the user-provided QoS values, similarities can be calculated between the active user and training users. Second, a set of similar users can be identified based on the similarity values. Later, two algorithms are proposed (i.e., QoSRank1 and QoSRank2) to make personalized service ranking by the past usage experiences of similar users. The ranking prediction results are returned to the active user finally. The training data in the QoSRank framework can be obtained from: 1) the QoS values provided by other users; and 2) the QoS values collected by monitoring cloud services. The QoSRank framework is a user-collaborative mechanism incorporating Bayesian Personalized Ranking, the performance of the current system will get improved. The usage experiences can be easily obtained in the cloud environment. Also client-side QoS performance of the invoked cloud services can be made easily by using monitoring infrastructure. The cloud provider is responsible for collecting the client-side QoS values from different user applications easily.

4. SIMILARITY COMPUTATION

In Cloud environment, an application proceeds with the cloudlet creation. A simulation toolkit is used that enables modelling and simulation of Cloud computing systems and application provisioning environments.

4.1 QoS Ranking

The objective of QoS ranking is to obtain quality service for a user application out of numerous services. This section presents the QoSRank framework for cloud services. First, we calculate the similarity of the active user with training data. Thus the similar users are identified. Here we calculate the similarity and identifies similar users.

Ranking compare users QoS rankings on the commonly invoked services. Given two rankings on the same set of services, the Kendall Rank Correlation Coefficient (KRCC) [2] 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

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

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. An indicator function defined as

$$I(x) = \begin{cases} 1\ if\ x < 0 \\ 0\ otherwise \end{cases} \qquad (2)$$

From the definition given above, 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*.

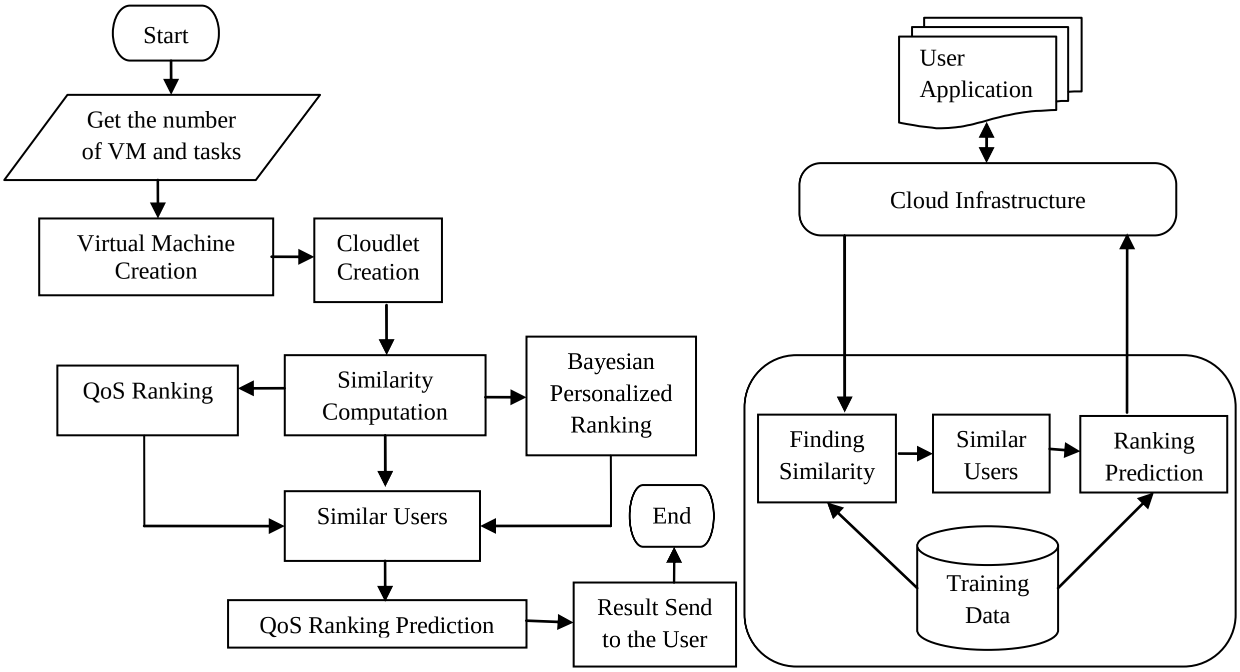

Fig 1: QoSRank Architecture

Thus the similar users are being identified by the similarity values obtained for active user with the training users. Previous approaches have a disadvantage that the dissimilar users may roughly affect the accuracy of ranking. Hence we exclude the negative similarity values for making prediction.

4.2 BPR-Opt

Item recommendation is the task of predicting a personalized ranking on a set of items, here items refers to cloud services. Implicit feedback is a technique used for personalized item recommendation. Here, we use a generic optimization criterion BPR-Opt for personalized ranking that is derived from a Bayesian analysis of the problem.

Recommender system is a crucial area of research. The most important topic in today's scenario is regarding explicit feedback provided by users, e.g. in terms of ratings. However, most of the feedback in real world is implicit rather than explicit. Implicit feedback such as like monitoring clicks, view times, purchases, etc. is tracked automatically and also it is much easier to collect, because the user need not made to express their choice explicitly. Frankly speaking, implicit feedback is readily available in almost any information system. Here we present a generic method for learning models for personalized ranking. The contributions of this work are:

1. The generic optimization criterion BPR-Opt derived from the maximum posterior estimator for optimal personalized ranking.
2. We use the generic learning algorithm LearnBPR that is based on stochastic gradient descent with bootstrap sampling of training triples.

4.2.1 BPR Learning Algorithm

The ranked list of items is provided to the user by personalized ranking called as item recommendation. This is also called item recommendation. For example, an online shop that wants to recommend a personalized ranked list of items to the user. Here the ranking has to be deduced from the implicit behavior (e.g. past experiences) of the user. Only positive observations are available for implicit feedback systems. The non-observed user-item pairs {e.g. a user has not bought an item yet} are a mixture of real negative feedback (the user is not interested in buying the item) and missing values.

Gradient descent based algorithms are an obvious choice for maximization as the criterion is differentiable. But as we will see, standard gradient descent is not the right choice for the problem. To solve this issue we propose LearnBPR, a stochastic gradient-descent algorithm based on bootstrap sampling of training triples.

```
procedure LearnBPR($D_S$, $\Theta$)
    initialize $\Theta$
     repeat
        draw (u, i, j) from DS
            $\Theta \leftarrow \Theta + \alpha \left( \frac{e^{-x'uij}}{1+e^{-x'uij}} \cdot \frac{\partial}{\partial\Theta} x'_{uij} + \lambda_{\Theta}.\Theta \right)$
        until convergence
    return $\Theta$'
end procedure
```

The above procedure, optimize models for BPR with bootstrapping based stochastic gradient descent with learning rate α and regularization λ_{Θ}.

5. RANKING PREDICTION

To resolve the problem of less accuracy or incorrect ranking, we propose two algorithms, QoSRank1 and QoSRank2 in the following. These two ranking approaches predict the QoS ranking without predicting the corresponding QoS values.

5.1 QoSRank1

A preference function for the user according to their choice as ψ: $I \times I \rightarrow \mathbb{R}$, where $\Psi(i,j)$>0 means that quality of service *i* is better than service *j* and is most preferable for the active user. The value of preference function specifies the strength of the user's preference. Likewise, if the value is zero, it means that there is no preference between two services.

Preference function assigns a score to every pair of services *i, j* ϵ *I*, in which we want to choose a quality ranking of services in *I*. Let *ρ* be a ranking of services in *I* such that *ρ(i)> ρ(j)*, if and only if *i* is ranked higher than *j*. The calculation made efficient by a greedy algorithm named as QoSRank1 for optimal rank prediction. It includes the following steps:

Algorithm 1: QoSRank1
Input: an employed service set E, a full service set I, preference function ψ
Output: a service ranking *ρ'*

```
$F=E;$
while $F \neq \emptyset$ do
    $t = arg\ max_{i\,\epsilon\,F}\, q_i;$
    $\rho_c(t) = |E| - |F| + 1;$
    $F = F - \{t\};$
end
for each $i \in I$ do
    $\pi(i) = \Sigma_{j\,\epsilon\,I}\ \psi(i,j);$
end
$n = |I|$
while $I \neq \emptyset$ do
    $t = \arg\max_{i\,\epsilon\,I} \pi(i)$
    $\rho'(t) = n - |I| + 1;$
    $I = I - \{t\};$
    for each $i \in I$ do
        $\pi(i) = \pi(i) - \psi(i,t);$
    end
end
while $E \neq \emptyset$ do
    $e = arg\ min_{i\,\epsilon\,E}\, \rho_e\, i;$
    $index = min_{i\,\epsilon\,E}\, \rho'(i);$
    $\rho'(e) = index;$
    $E = E - \{e\};$
end
```

Here the cloud services are ranked on the basis of observed QoS values in which $\rho_c\,(t)$ stores the ranking, where *t* is the service and *ρ* is the ranking order. Thus the smaller *ρ* value indicates the high quality. Then we calculate the preference function value. Later the services are being ranked from the highest position to the lowest position by π (*t*) at the maximum. The smaller value of ranked π (*t*) indicates high quality. Hence the remaining services are updated by replacing the ranking results in *ρ'*.

5.2 QoSRank2

As in QoSRank1, preference value is calculated explicitly or implicitly. But in the QoSRank1 algorithm, all the differences in the preference values are treated evenly which may affect the accuracy of the ranking result. Thus we use another algorithm, QoSRank2 by considering confidence value of the different preference values by using the following rules:

- If a user is having the QoS value of two cloud services i and j, then confidence value is 1.
- For similar users, the confidence value is determined by similarities of them as:

$$C(i,j) = \sum_{v\,\epsilon\,N(u)^{ij}} w_v Sim(u,v) \qquad (3)$$

Where *v* is a similar user of the current user *u*, $N(u)^{ij}$ is a subset of similar users and w_v is a weighting factor which can be calculated by

$$w(v) = \frac{Sim(u,v)}{\sum_{v\,\epsilon\,N(u)^{ij}} Sim(u,v)} \qquad (4)$$

Algorithm 1: QoSRank2
Input: an employed service set E, a full service set I, preference function ψ, confidence value C
Output: a service ranking *ρ'*

```
$F=E;$
while $F \neq \emptyset$ do
    $t = arg\ max_{i\,\epsilon\,F}\, q_i;$
    $\rho_c(t) = |E| - |F| + 1;$
    $F = F - \{t\};$
end
for each $i \in I$ do
    $\pi(i) = \Sigma_{j\,\epsilon\,I}\ C(i,j) \times \psi(i,j);$
end
$n = |I|$
while $\neq \emptyset$ do
    $t = \arg\max_{i\,\epsilon\,I} \pi(i);$
    $\rho'(t) = n - |I| + 1;$
    $I = I - \{t\};$
    for each $i \in I$ do
        $\pi(i) = \pi(i) - C(i,j) \times \psi(i,j);$
    end
end
while $E \neq \emptyset$ do
    $e = arg\ min_{i\,\epsilon\,E}\, \rho_e\, i;$
    $index = min_{i\,\epsilon\,E}\, \rho'(i);$
    $\rho'(e) = index;$
    $E = E - \{e\};$
end
```

The above mentioned algorithm consider both the preference value and confidence value and produce the accurate result than QoSRank1.

6. RESULTS AND ANALYSIS

A simulation tool named CloudSim has been used for simulating 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 QoSRank.

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.

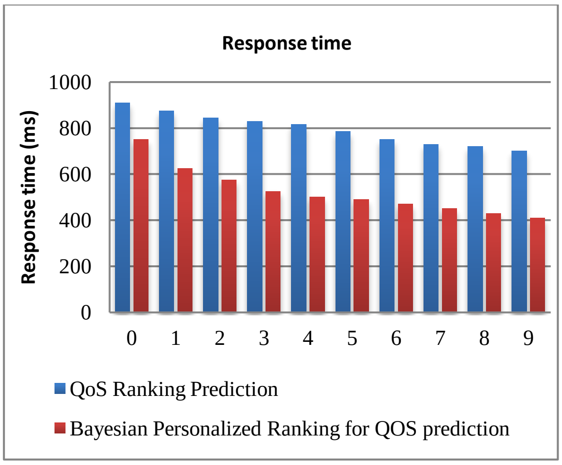

Fig 2: Response time Comparison

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.

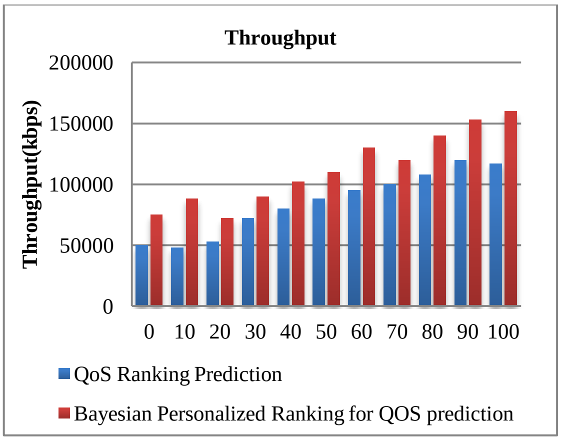

Fig 3: Throughput Comparison

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.

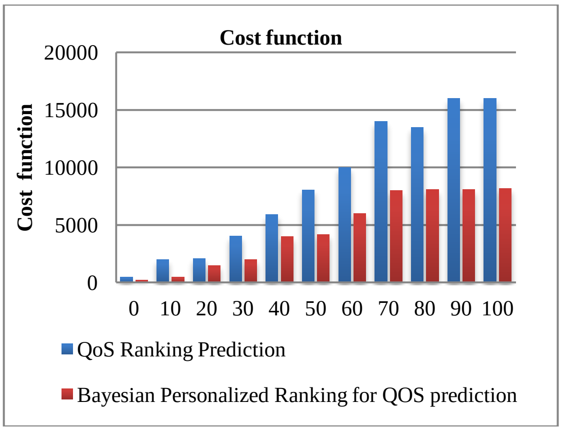

Fig 4: Cost Comparison

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.

7. CONCLUSION

Concluding, QoSRank, the QoS ranking prediction framework selecting optimal cloud services based on Bayesian Personalized Ranking. The result shows that proposed framework is able to perform well with no additional invocations of cloud services while predicting appropriate services for the user application. The simulation results show that these approaches are outperforming other competing approaches. The experimental results show that QoSRank1 and QoSRank2 approaches outperform other approaches and the Kendall Rank Correlation Coefficient along with Bayesian Personalized Rank provides the exact result as expected. The results proves that algorithms used for the ranking prediction outperforms the existing systems.

In the future, we would like to extend this work with enhancement in additional techniques used for ranking or by means of providing security for the services.

8. REFERENCES

[1] Andrzej Goscinski, Michael Brock, 2010,“Toward dynamic and attribute based publication, discovery and selection for cloud computing”, *Future Generation Computer Systems*, pp. 947-970.

[2] J. Marden, 1995, “Analyzing and Modeling Ranking Data”, *Chapman & Hall*.

[3] Jose Luis Lucas-Simarro, Rafael Moreno-Vozmediano, Ruben S. Montero, Ignacio M. Llorente, 2012, “Scheduling strategies for optimal service deployment acrossmultiple clouds”, *Future Generation Computer Systems*, pp.1431–1441.

[4] Linlin Wu, Saurabh Kumar Garg, Rajkumar Buyya, 2011, “SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments ”, *Journal of Computer and System Sciences*, pp. 1280–1299.

[5] Michael Armbrust et al., 2010, “A View of Cloud Computing,” *Comm. ACM*, vol. 53, no. 4, pp. 50-58,.

[6] Pawel Czarnul, 2012,” An Evaluation Engine for Dynamic Ranking of Cloud Providers”, *Informatica 37*, pp. 123–130,.

[7] Qinyi Wu, Arun Iyengar et.al, 2009, “Combining Quality of Service and Social Information for Ranking Services”, *7th International Joint Conference, ICSOC-ServiceWave, Sweden*, pp. 561-575.

[8] Saurabh Kumar Garg, Steve Versteeg, Rajkumar Buyya, 2012, “A framework for ranking of cloud computing services”, *Future Generation Computer Systems*, pp.1012–1023.

[9] Rendle et. al., 2009, “BPR: Bayesian Personalized Ranking from Implicit Feedback”, *UAI*, pp. 452-461.

[10] Yilei Zhang, Zibin Zheng and Michael R. Lyu, 2011, “BFTCloud: A Byzantine Fault Tolerance Framework for Voluntary-Resource Cloud Computing”, *2011 IEEE 4th International Conference on Cloud Computing*, pp. 444-451.

[11] Zibin Zheng et al., 2012, “QoS Ranking Prediction for Cloud Services”, *IEEE Transactions On Parallel And Distributed Systems*, Vol. -24, No. 6, June 2013, pp. 1213-1222.